FIFTH EDITION

COMPUTER NETWORKS

TANENBAUM | WETHERALL

This page intentionally left blank

COMPUTER NETWORKS

FIFTH EDITION

This page intentionally left blank

COMPUTER NETWORKS

FIFTH EDITION

ANDREW S. TANENBAUM

Vrije Universiteit
Amsterdam, The Netherlands

DAVID J. WETHERALL

University of Washington
Seattle, WA

PRENTICE HALL

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Paris Montreal Toronto
Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Tapei Tokyo

Editorial Director: Marcia Horton
Editor-in-Chief: Michael Hirsch
Executive Editor: Tracy Dunkelberger
Assistant Editor: Melinda Haggerty
Editorial Assistant: Allison Michael
Vice President, Marketing: Patrice Jones
Marketing Manager: Yezan Alayan
Marketing Coordinator: Kathryn Ferranti
Vice President, Production: Vince O'Brien
Managing Editor: Jeff Holcomb
Senior Operations Supervisor: Alan Fischer
Manufacturing Buyer: Lisa McDowell
Cover Direction: Andrew S. Tanenbaum, David J. Wetherall, Tracy Dunkelberger
Art Director: Linda Knowles
Cover Designer: Susan Paradise
Cover Illustration: Jason Consalvo
Interior Design: Andrew S. Tanenbaum
AV Production Project Manager: Gregory L. Dulles
Interior Illustrations: Laserwords, Inc.
Media Editor: Daniel Sandin
Composition: Andrew S. Tanenbaum
Copyeditor: Rachel Head
Proofreader: Joe Ruddick
Printer/Binder: Courier/Westford
Cover Printer: Lehigh-Phoenix Color/ Hagerstown

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on appropriate page within text.

Library of Congress Cataloging-in-Publication Data

Tanenbaum, Andrew S., 1944-
Computer networks / Andrew S. Tanenbaum, David J. Wetherall. -- 5th ed.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-13-212695-3 (alk. paper)
ISBN-10: 0-13-212695-8 (alk. paper)
1. Computer networks. I. Wetherall, D. (David) II. Title.
TK5105.5.T36 2011
004.6--dc22

2010034366

10 9 8 7 6 5 4 3 2 1—CRW—14 13 12 11 10

PEARSON

To Suzanne, Barbara, Daniel, Aron, Marvin, Matilde, and the memory of Bram, and Sweetie π (AST)

To Katrin, Lucy, and Pepper (DJW)

CONTENTS

5 THE NETWORK LAYER 355

This page intentionally left blank

PREFACE

This book is now in its fifth edition. Each edition has corresponded to a different phase in the way computer networks were used. When the first edition appeared in 1980, networks were an academic curiosity. When the second edition appeared in 1988, networks were used by universities and large businesses. When the third edition appeared in 1996, computer networks, especially the Internet, had become a daily reality for millions of people. By the fourth edition, in 2003, wireless networks and mobile computers had become commonplace for accessing the Web and the Internet. Now, in the fifth edition, networks are about content distribution (especially videos using CDNs and peer-to-peer networks) and mobile phones are small computers on the Internet.

New in the Fifth Edition

Among the many changes in this book, the most important one is the addition of Prof. David J. Wetherall as a co-author. David brings a rich background in networking, having cut his teeth designing metropolitan-area networks more than 20 years ago. He has worked with the Internet and wireless networks ever since and is a professor at the University of Washington, where he has been teaching and doing research on computer networks and related topics for the past decade.

Of course, the book also has many changes to keep up with the: ever-changing world of computer networks. Among these are revised and new material on

- Wireless networks (802.12 and 802.16)
- The 3G networks used by smart phones
- RFID and sensor networks
- Content distribution using CDNs
- Peer-to-peer networks
- Real-time media (from stored, streaming, and live sources)
- Internet telephony (voice over IP)
- Delay-tolerant networks

A more detailed chapter-by-chapter list follows.

Chapter 1 has the same introductory function as in the fourth edition, but the contents have been revised and brought up to date. The Internet, mobile phone networks, 802.11, and RFID and sensor networks are discussed as examples of computer networks. Material on the original Ethernet—with its vampire taps—has been removed, along with the material on ATM.

Chapter 2, which covers the physical layer, has expanded coverage of digital modulation (including OFDM as widely used in wireless networks) and 3G networks (based on CDMA). New technologies are discussed, including Fiber to the Home and power-line networking.

Chapter 3, on point-to-point links, has been improved in two ways. The material on codes for error detection and correction has been updated, and also includes a brief description of the modern codes that are important in practice (e.g., convolutional and LDPC codes). The examples of protocols now use Packet over SONET and ADSL. Sadly, the material on protocol verification has been removed as it is little used.

In Chapter 4, on the MAC sublayer, the principles are timeless but the technologies have changed. Sections on the example networks have been redone accordingly, including gigabit Ethernet, 802.11, 802.16, Bluetooth, and RFID. Also updated is the coverage of LAN switching, including VLANs.

Chapter 5, on the network layer, covers the same ground as in the fourth edition. The revisions have been to update material and add depth, particularly for quality of service (relevant for real-time media) and internetworking. The sections on BGP, OSPF and CIDR have been expanded, as has the treatment of multicast routing. Anycast routing is now included.

Chapter 6, on the transport layer, has had material added, revised, and removed. New material describes delay-tolerant networking and congestion control in general. The revised material updates and expands the coverage of TCP congestion control. The material removed described connection-oriented network layers, something rarely seen any more.

Chapter 7, on applications, has also been updated and enlarged. While material on DNS and email is similar to that in the fourth edition, in the past few years there have been many developments in the use of the Web, streaming media and content delivery. Accordingly, sections on the Web and streaming media have been brought up to date. A new section covers content distribution, including CDNs and peer-to-peer networks.

Chapter 8, on security, still covers both symmetric and public-key cryptography for confidentiality and authenticity. Material on the techniques used in practice, including firewalls and VPNs, has been updated, with new material on 802.11 security and Kerberos V5 added.

Chapter 9 contains a renewed list of suggested readings and a comprehensive bibliography of over 300 citations to the current literature. More than half of these are to papers and books written in 2000 or later, and the rest are citations to classic papers.

List of Acronyms

Computer books are full of acronyms. This one is no exception. By the time you are finished reading this one, the following should ring a bell: ADSL, AES, AJAX, AODV, AP, ARP, ARQ, AS, BGP, BOC, CDMA, CDN, CGI, CIDR, CRL, CSMA, CSS, DCT, DES, DHCP, DHT, DIFS, DMCA, DMT, DMZ, DNS, DOCSIS, DOM, DSLAM, DTN, FCFS, FDD, FDDI, FDM, FEC, FIFO, FSK, FTP, GPRS, GSM, HDTV, HFC, HMAC, HTTP, IAB, ICANN, ICMP, IDEA, IETF, IMAP, IMP, IP, IPTV, IRTF, ISO, ISP, ITU, JPEG, JSP, JVM, LAN, LATA, LEC, LEO, LLC, LSR, LTE, MAN, MFJ, MIME, MPEG, MPLS, MSC, MTSO, MTU, NAP, NAT, NRZ, NSAP, OFDM, OSI, OSPF, PAWS, PCM, PGP, PIM, PKI, POP, POTS, PPP, PSTN, QAM, QPSK, RED, RFC, RFID, RPC, RSA, RTSP, SHA, SIP, SMTP, SNR, SOAP, SONET, SPE, SSL, TCP, TDD, TDM, TSAP, UDP, UMTS, URL, VLAN, VSAT, WAN, WDM, and XML. But don't worry. Each will appear in **boldface type** and be carefully defined before it is used. As a fun test, see how many you can identify *before* reading the book, write the number in the margin, then try again *after* reading the book.

How to Use the Book

To help instructors use this book as a text for courses ranging in length from quarters to semesters, we have structured the chapters into core and optional material. The sections marked with a "*" in the table of contents are the optional ones. If a major section (e.g., 2.7) is so marked, all of its subsections are optional. They provide material on network technologies that is useful but can be omitted from a short course without loss of continuity. Of course, students should be encouraged to read those sections as well, to the extent they have time, as all the material is up to date and of value.

Instructors' Resource Materials

The following protected instructors' resource materials are available on the publisher's Web site at *www.pearsonhighered.com/tanenbaum*. For a username and password, please contact your local Pearson representative.

- Solutions manual
- PowerPoint lecture slides

Students' Resource Materials

Resources for students are available through the open-access Companion Web site link on *www.pearsonhighered.com/tanenbaum*, including

- Web resources, links to tutorials, organizations, FAQs, and more
- Figures, tables, and programs from the book
- Steganography demo
- Protocol simulators

Acknowledgements

Many people helped us during the course of the fifth edition. We would especially like to thank Emmanuel Agu (Worcester Polytechnic Institute), Yoris Au (University of Texas at Antonio), Nikhil Bhargava (Aircom International, Inc.), Michael Buettner (University of Washington), John Day (Boston University), Kevin Fall (Intel Labs), Ronald Fulle (Rochester Institute of Technology), Ben Greenstein (Intel Labs), Daniel Halperin (University of Washington), Bob Kinicki (Worcester Polytechnic Institute), Tadayoshi Kohno (University of Washington), Sarvish Kulkarni (Villanova University), Hank Levy (University of Washington), Ratul Mahajan (Microsoft Research), Craig Partridge (BBN), Michael Piatek (University of Washington), Joshua Smith (Intel Labs), Neil Spring (University of Maryland), David Teneyuca (University of Texas at Antonio), Tammy VanDegrift (University of Portland), and Bo Yuan (Rochester Institute of Technology), for providing ideas and feedback. Melody Kadenko and Julie Svendsen provided administrative support to David.

Shivakant Mishra (University of Colorado at Boulder) and Paul Nagin (Chimborazo Publishing, Inc.) thought of many new and challenging end-of-chapter problems. Our editor at Pearson, Tracy Dunkelberger, was her usual helpful self in many ways large and small. Melinda Haggerty and Jeff Holcomb did a good job of keeping things running smoothly. Steve Armstrong (LeTourneau University) prepared the PowerPoint slides. Stephen Turner (University of Michigan at Flint) artfully revised the Web resources and the simulators that accompany the text. Our copyeditor, Rachel Head, is an odd hybrid: she has the eye of an eagle and the memory of an elephant. After reading all her corrections, both of us wondered how we ever made it past third grade.

Finally, we come to the most important people. Suzanne has been through this 19 times now and still has endless patience and love. Barbara and Marvin now know the difference between good textbooks and bad ones and are always an inspiration to produce good ones. Daniel and Matilde are welcome additions to our family. Aron is unlikely to read this book soon, but he likes the nice pictures on page 866 (AST). Katrin and Lucy provided endless support and always managed to keep a smile on my face. Thank you (DJW).

ANDREW S. TANENBAUM

DAVID J. WETHERALL

1

INTRODUCTION

Each of the past three centuries was dominated by a single new technology. The 18th century was the era of the great mechanical systems accompanying the Industrial Revolution. The 19th century was the age of the steam engine. During the 20th century, the key technology was information gathering, processing, and distribution. Among other developments, we saw the installation of worldwide telephone networks, the invention of radio and television, the birth and unprecedented growth of the computer industry, the launching of communication satellites, and, of course, the Internet.

As a result of rapid technological progress, these areas are rapidly converging in the 21st century and the differences between collecting, transporting, storing, and processing information are quickly disappearing. Organizations with hundreds of offices spread over a wide geographical area routinely expect to be able to examine the current status of even their most remote outpost at the push of a button. As our ability to gather, process, and distribute information grows, the demand for ever more sophisticated information processing grows even faster.

Although the computer industry is still young compared to other industries (e.g., automobiles and air transportation), computers have made spectacular progress in a short time. During the first two decades of their existence, computer systems were highly centralized, usually within a single large room. Not infrequently, this room had glass walls, through which visitors could gawk at the great electronic wonder inside. A medium-sized company or university might have had one or two computers, while very large institutions had at most a few dozen. The idea that within forty years vastly more powerful computers smaller than postage stamps would be mass produced by the billions was pure science fiction.

The merging of computers and communications has had a profound influence on the way computer systems are organized. The once-dominant concept of the "computer center" as a room with a large computer to which users bring their work for processing is now totally obsolete (although data centers holding thousands of Internet servers are becoming common). The old model of a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called **computer networks**. The design and organization of these networks are the subjects of this book.

Throughout the book we will use the term "computer network" to mean a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information. The connection need not be via a copper wire; fiber optics, microwaves, infrared, and communication satellites can also be used. Networks come in many sizes, shapes and forms, as we will see later. They are usually connected together to make larger networks, with the **Internet** being the most well-known example of a network of networks.

There is considerable confusion in the literature between a computer network and a **distributed system**. The key distinction is that in a distributed system, a collection of independent computers appears to its users as a single coherent system. Usually, it has a single model or paradigm that it presents to the users. Often a layer of software on top of the operating system, called **middleware**, is responsible for implementing this model. A well-known example of a distributed system is the **World Wide Web**. It runs on top of the Internet and presents a model in which everything looks like a document (Web page).

In a computer network, this coherence, model, and software are absent. Users are exposed to the actual machines, without any attempt by the system to make the machines look and act in a coherent way. If the machines have different hardware and different operating systems, that is fully visible to the users. If a user wants to run a program on a remote machine, he† has to log onto that machine and run it there.

In effect, a distributed system is a software system built on top of a network. The software gives it a high degree of cohesiveness and transparency. Thus, the distinction between a network and a distributed system lies with the software (especially the operating system), rather than with the hardware.

Nevertheless, there is considerable overlap between the two subjects. For example, both distributed systems and computer networks need to move files around. The difference lies in who invokes the movement, the system or the user.

† "He" should be read as "he or she" throughout this book.

Although this book primarily focuses on networks, many of the topics are also important in distributed systems. For more information about distributed systems, see Tanenbaum and Van Steen (2007).

1.1 USES OF COMPUTER NETWORKS

Before we start to examine the technical issues in detail, it is worth devoting some time to pointing out why people are interested in computer networks and what they can be used for. After all, if nobody were interested in computer networks, few of them would be built. We will start with traditional uses at companies, then move on to home networking and recent developments regarding mobile users, and finish with social issues.

1.1.1 Business Applications

Most companies have a substantial number of computers. For example, a company may have a computer for each worker and use them to design products, write brochures, and do the payroll. Initially, some of these computers may have worked in isolation from the others, but at some point, management may have decided to connect them to be able to distribute information throughout the company.

Put in slightly more general form, the issue here is **resource sharing**. The goal is to make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource or the user. An obvious and widespread example is having a group of office workers share a common printer. None of the individuals really needs a private printer, and a high-volume networked printer is often cheaper, faster, and easier to maintain than a large collection of individual printers.

However, probably even more important than sharing physical resources such as printers, and tape backup systems, is sharing information. Companies small and large are vitally dependent on computerized information. Most companies have customer records, product information, inventories, financial statements, tax information, and much more online. If all of its computers suddenly went down, a bank could not last more than five minutes. A modern manufacturing plant, with a computer-controlled assembly line, would not last even 5 seconds. Even a small travel agency or three-person law firm is now highly dependent on computer networks for allowing employees to access relevant information and documents instantly.

For smaller companies, all the computers are likely to be in a single office or perhaps a single building, but for larger ones, the computers and employees may be scattered over dozens of offices and plants in many countries. Nevertheless, a sales person in New York might sometimes need access to a product inventory database in Singapore. Networks called **VPNs** (**Virtual Private Networks**) may be used to join the individual networks at different sites into one extended network. In other words, the mere fact that a user happens to be 15,000 km away from his data should not prevent him from using the data as though they were local. This goal may be summarized by saying that it is an attempt to end the "tyranny of geography."

In the simplest of terms, one can imagine a company's information system as consisting of one or more databases with company information and some number of employees who need to access them remotely. In this model, the data are stored on powerful computers called **servers**. Often these are centrally housed and maintained by a system administrator. In contrast, the employees have simpler machines, called **clients**, on their desks, with which they access remote data, for example, to include in spreadsheets they are constructing. (Sometimes we will refer to the human user of the client machine as the "client," but it should be clear from the context whether we mean the computer or its user.) The client and server machines are connected by a network, as illustrated in Fig. 1-1. Note that we have shown the network as a simple oval, without any detail. We will use this form when we mean a network in the most abstract sense. When more detail is required, it will be provided.

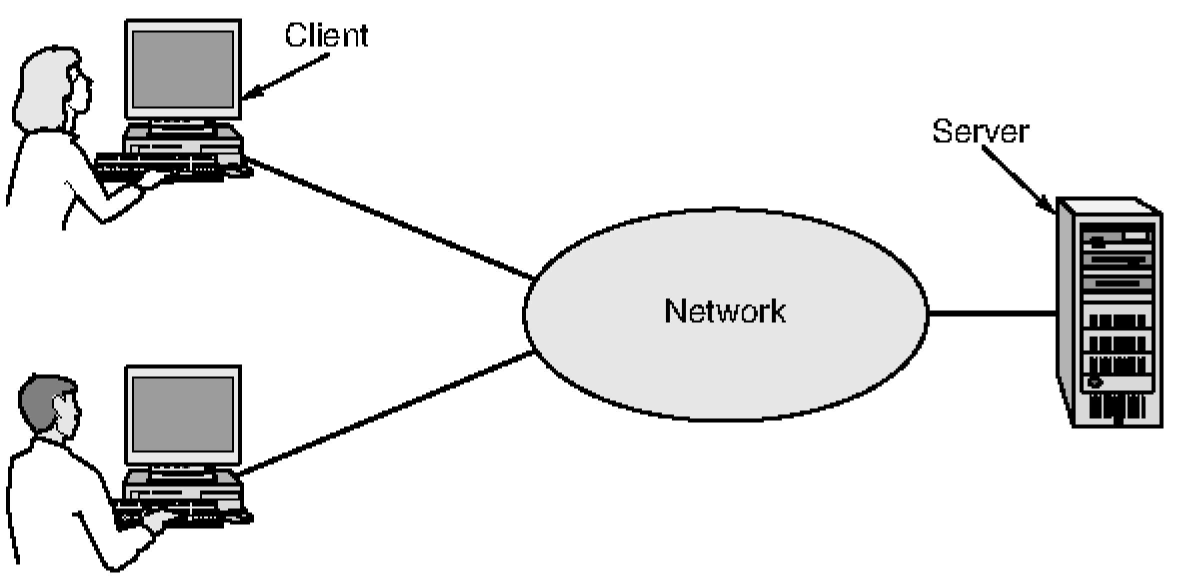


Figure 1-1. A network with two clients and one server.

This whole arrangement is called the **client-server model**. It is widely used and forms the basis of much network usage. The most popular realization is that of a **Web application**, in which the server generates Web pages based on its database in response to client requests that may update the database. The client-server model is applicable when the client and server are both in the same building (and belong to the same company), but also when they are far apart. For example, when a person at home accesses a page on the World Wide Web, the same model is employed, with the remote Web server being the server and the user's personal computer being the client. Under most conditions, one server can handle a large number (hundreds or thousands) of clients simultaneously.

If we look at the client-server model in detail, we see that two processes (i.e., running programs) are involved, one on the client machine and one on the server machine. Communication takes the form of the client process sending a message over the network to the server process. The client process then waits for a reply message. When the server process gets the request, it performs the requested work or looks up the requested data and sends back a reply. These messages are shown in Fig. 1-2.

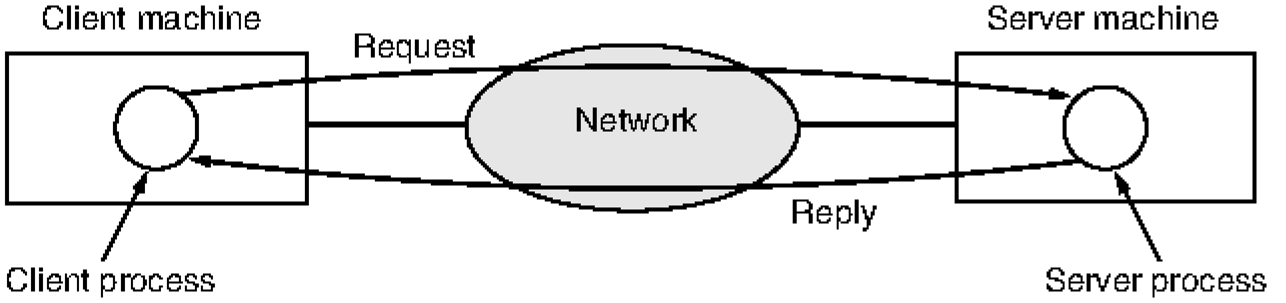


Figure 1-2. The client-server model involves requests and replies.

A second goal of setting up a computer network has to do with people rather than information or even computers. A computer network can provide a powerful **communication medium** among employees. Virtually every company that has two or more computers now has **email** (**electronic mail**), which employees generally use for a great deal of daily communication. In fact, a common gripe around the water cooler is how much email everyone has to deal with, much of it quite meaningless because bosses have discovered that they can send the same (often content-free) message to all their subordinates at the push of a button.

Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called **IP telephony** or **Voice over IP** (**VoIP**) when Internet technology is used. The microphone and speaker at each end may belong to a VoIP-enabled phone or the employee's computer. Companies find this a wonderful way to save on their telephone bills.

Other, richer forms of communication are made possible by computer networks. Video can be added to audio so that employees at distant locations can see and hear each other as they hold a meeting. This technique is a powerful tool for eliminating the cost and time previously devoted to travel. **Desktop sharing** lets remote workers see and interact with a graphical computer screen. This makes it easy for two or more people who work far apart to read and write a shared blackboard or write a report together. When one worker makes a change to an online document, the others can see the change immediately, instead of waiting several days for a letter. Such a speedup makes cooperation among far-flung groups of people easy where it previously had been impossible. More ambitious forms of remote coordination such as telemedicine are only now starting to be used (e.g., remote patient monitoring) but may become much more important. It is sometimes said that communication and transportation are having a race, and whichever wins will make the other obsolete.

A third goal for many companies is doing business electronically, especially with customers and suppliers. This new model is called **e-commerce** (**electronic commerce**) and it has grown rapidly in recent years. Airlines, bookstores, and other retailers have discovered that many customers like the convenience of shopping from home. Consequently, many companies provide catalogs of their goods and services online and take orders online. Manufacturers of automobiles, aircraft, and computers, among others, buy subsystems from a variety of suppliers and then assemble the parts. Using computer networks, manufacturers can place orders electronically as needed. This reduces the need for large inventories and enhances efficiency.

1.1.2 Home Applications

In 1977, Ken Olsen was president of the Digital Equipment Corporation, then the number two computer vendor in the world (after IBM). When asked why Digital was not going after the personal computer market in a big way, he said: "There is no reason for any individual to have a computer in his home." History showed otherwise and Digital no longer exists. People initially bought computers for word processing and games. Recently, the biggest reason to buy a home computer was probably for Internet access. Now, many consumer electronic devices, such as set-top boxes, game consoles, and clock radios, come with embedded computers and computer networks, especially wireless networks, and home networks are broadly used for entertainment, including listening to, looking at, and creating music, photos, and videos.

Internet access provides home users with **connectivity** to remote computers. As with companies, home users can access information, communicate with other people, and buy products and services with e-commerce. The main benefit now comes from connecting outside of the home. Bob Metcalfe, the inventor of Ethernet, hypothesized that the value of a network is proportional to the square of the number of users because this is roughly the number of different connections that may be made (Gilder, 1993). This hypothesis is known as "Metcalfe's law." It helps to explain how the tremendous popularity of the Internet comes from its size.

Access to remote information comes in many forms. It can be surfing the World Wide Web for information or just for fun. Information available includes the arts, business, cooking, government, health, history, hobbies, recreation, science, sports, travel, and many others. Fun comes in too many ways to mention, plus some ways that are better left unmentioned.

Many newspapers have gone online and can be personalized. For example, it is sometimes possible to tell a newspaper that you want everything about corrupt politicians, big fires, scandals involving celebrities, and epidemics, but no football, thank you. Sometimes it is possible to have the selected articles downloaded to your computer while you sleep. As this trend continues, it will cause massive unemployment among 12-year-old paperboys, but newspapers like it because distribution has always been the weakest link in the whole production chain. Of course, to make this model work, they will first have to figure out how to make money in this new world, something not entirely obvious since Internet users expect everything to be free.

The next step beyond newspapers (plus magazines and scientific journals) is the online digital library. Many professional organizations, such as the ACM (*www.acm.org*) and the IEEE Computer Society (*www.computer.org*), already have all their journals and conference proceedings online. Electronic book readers and online libraries may make printed books obsolete. Skeptics should take note of the effect the printing press had on the medieval illuminated manuscript.

Much of this information is accessed using the client-server model, but there is different, popular model for accessing information that goes by the name of **peer-to-peer** communication (Parameswaran et al., 2001). In this form, individuals who form a loose group can communicate with others in the group, as shown in Fig. 1-3. Every person can, in principle, communicate with one or more other people; there is no fixed division into clients and servers.

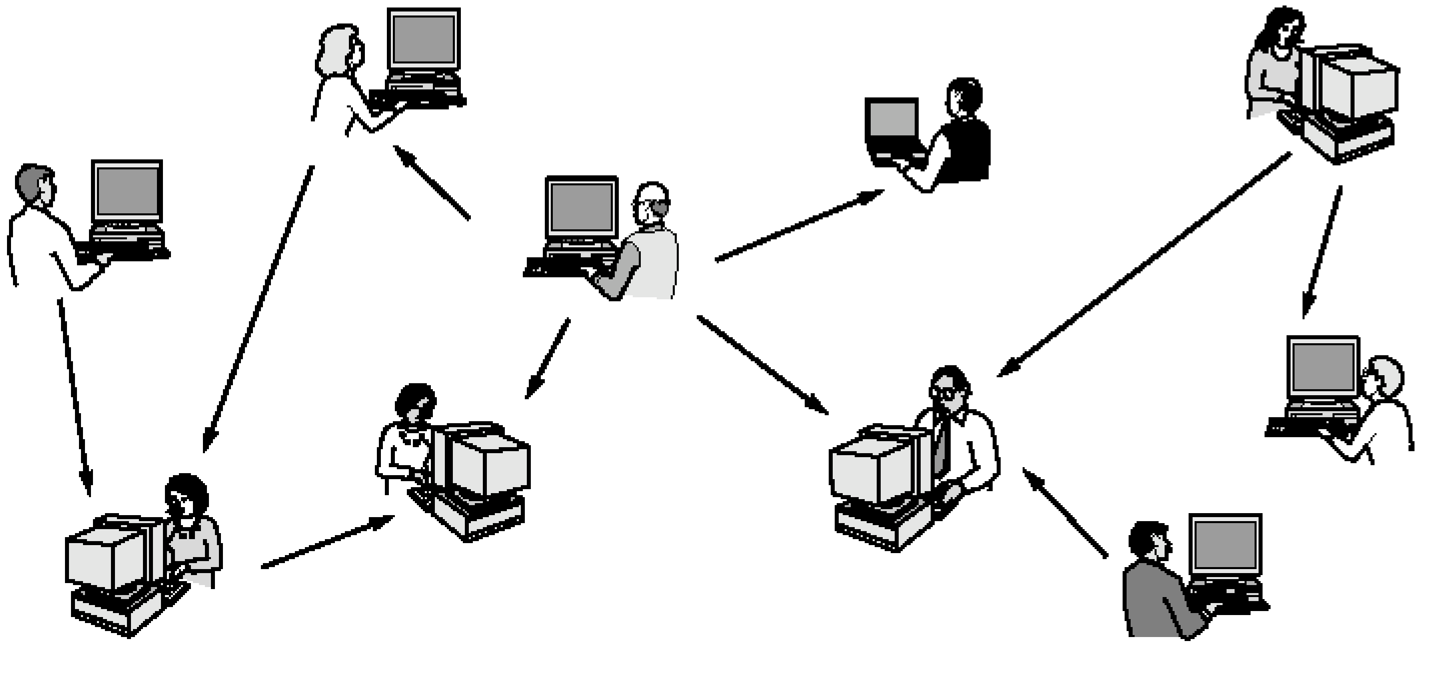


Figure 1-3. In a peer-to-peer system there are no fixed clients and servers.

Many peer-to-peer systems, such BitTorrent (Cohen, 2003), do not have any central database of content. Instead, each user maintains his own database locally and provides a list of other nearby people who are members of the system. A new user can then go to any existing member to see what he has and get the names of other members to inspect for more content and more names. This lookup process can be repeated indefinitely to build up a large local database of what is out there. It is an activity that would get tedious for people but computers excel at it.

Peer-to-peer communication is often used to share music and videos. It really hit the big time around 2000 with a music sharing service called Napster that was shut down after what was probably the biggest copyright infringement case in all of recorded history (Lam and Tan, 2001; and Macedonia, 2000). Legal applications for peer-to-peer communication also exist. These include fans sharing public domain music, families sharing photos and movies, and users downloading public software packages. In fact, one of the most popular Internet applications of all, email, is inherently peer-to-peer. This form of communication is likely to grow considerably in the future.

All of the above applications involve interactions between a person and a remote database full of information. The second broad category of network use is person-to-person communication, basically the 21st century's answer to the 19th century's telephone. E-mail is already used on a daily basis by millions of people all over the world and its use is growing rapidly. It already routinely contains audio and video as well as text and pictures. Smell may take a while.

Any teenager worth his or her salt is addicted to **instant messaging**. This facility, derived from the UNIX *talk* program in use since around 1970, allows two people to type messages at each other in real time. There are multi-person messaging services too, such as the **Twitter** service that lets people send short text messages called "tweets" to their circle of friends or other willing audiences.

The Internet can be used by applications to carry audio (e.g., Internet radio stations) and video (e.g., YouTube). Besides being a cheap way to call to distant friends, these applications can provide rich experiences such as telelearning, meaning attending 8 A.M. classes without the inconvenience of having to get out of bed first. In the long run, the use of networks to enhance human-to-human communication may prove more important than any of the others. It may become hugely important to people who are geographically challenged, giving them the same access to services as people living in the middle of a big city.

Between person-to-person communications and accessing information are **social network** applications. Here, the flow of information is driven by the relationships that people declare between each other. One of the most popular social networking sites is **Facebook**. It lets people update their personal profiles and shares the updates with other people who they have declared to be their friends. Other social networking applications can make introductions via friends of friends, send news messages to friends such as Twitter above, and much more.

Even more loosely, groups of people can work together to create content. A **wiki**, for example, is a collaborative Web site that the members of a community edit. The most famous wiki is the **Wikipedia**, an encyclopedia anyone can edit, but there are thousands of other wikis.

Our third category is electronic commerce in the broadest sense of the term. Home shopping is already popular and enables users to inspect the online catalogs of thousands of companies. Some of these catalogs are interactive, showing products from different viewpoints and in configurations that can be personalized.

After the customer buys a product electronically but cannot figure out how to use it, online technical support may be consulted.

Another area in which e-commerce is widely used is access to financial institutions. Many people already pay their bills, manage their bank accounts, and handle their investments electronically. This trend will surely continue as networks become more secure.

One area that virtually nobody foresaw is electronic flea markets (e-flea?). Online auctions of second-hand goods have become a massive industry. Unlike traditional e-commerce, which follows the client-server model, online auctions are peer-to-peer in the sense that consumers can act as both buyers and sellers.

Some of these forms of e-commerce have acquired cute little tags based on the fact that "to" and "2" are pronounced the same. The most popular ones are listed in Fig. 1-4.

Tag	Full name	Example
B2C	Business-to-consumer	Ordering books online
B2B	Business-to-business	Car manufacturer ordering tires from supplier
G2C	Government-to-consumer	Government distributing tax forms electronically
C2C	Consumer-to-consumer	Auctioning second-hand products online
P2P	Peer-to-peer	Music sharing

Figure 1-4. Some forms of e-commerce.

Our fourth category is entertainment. This has made huge strides in the home in recent years, with the distribution of music, radio and television programs, and movies over the Internet beginning to rival that of traditional mechanisms. Users can find, buy, and download MP3 songs and DVD-quality movies and add them to their personal collection. TV shows now reach many homes via **IPTV** (**IP TeleVision**) systems that are based on IP technology instead of cable TV or radio transmissions. Media streaming applications let users tune into Internet radio stations or watch recent episodes of their favorite TV shows. Naturally, all of this content can be moved around your house between different devices, displays and speakers, usually with a wireless network.

Soon, it may be possible to search for any movie or television program ever made, in any country, and have it displayed on your screen instantly. New films may become interactive, where the user is occasionally prompted for the story direction (should Macbeth murder Duncan or just bide his time?) with alternative scenarios provided for all cases. Live television may also become interactive, with the audience participating in quiz shows, choosing among contestants, and so on.

Another form of entertainment is game playing. Already we have multiperson real-time simulation games, like hide-and-seek in a virtual dungeon, and flight simulators with the players on one team trying to shoot down the players on the opposing team. Virtual worlds provide a persistent setting in which thousands of users can experience a shared reality with three-dimensional graphics.

Our last category is **ubiquitous computing**, in which computing is embedded into everyday life, as in the vision of Mark Weiser (1991). Many homes are already wired with security systems that include door and window sensors, and there are many more sensors that can be folded in to a smart home monitor, such as energy consumption. Your electricity, gas and water meters could also report usage over the network. This would save money as there would be no need to send out meter readers. And your smoke detectors could call the fire department instead of making a big noise (which has little value if no one is home). As the cost of sensing and communication drops, more and more measurement and reporting will be done with networks.

Increasingly, consumer electronic devices are networked. For example, some high-end cameras already have a wireless network capability and use it to send photos to a nearby display for viewing. Professional sports photographers can also send their photos to their editors in real-time, first wirelessly to an access point then over the Internet. Devices such as televisions that plug into the wall can use **power-line networks** to send information throughout the house over the wires that carry electricity. It may not be very surprising to have these objects on the network, but objects that we do not think of as computers may sense and communicate information too. For example, your shower may record water usage, give you visual feedback while you lather up, and report to a home environmental monitoring application when you are done to help save on your water bill.

A technology called **RFID** (**Radio Frequency IDentification**) will push this idea even further in the future. RFID tags are passive (i.e., have no battery) chips the size of stamps and they can already be affixed to books, passports, pets, credit cards, and other items in the home and out. This lets RFID readers locate and communicate with the items over a distance of up to several meters, depending on the kind of RFID. Originally, RFID was commercialized to replace barcodes. It has not succeeded yet because barcodes are free and RFID tags cost a few cents. Of course, RFID tags offer much more and their price is rapidly declining. They may turn the real world into the Internet of things (ITU, 2005).

1.1.3 Mobile Users

Mobile computers, such as laptop and handheld computers, are one of the fastest-growing segments of the computer industry. Their sales have already overtaken those of desktop computers. Why would anyone want one? People on the go often want to use their mobile devices to read and send email, tweet, watch movies, download music, play games, or simply to surf the Web for information. They want to do all of the things they do at home and in the office. Naturally, they want to do them from anywhere on land, sea or in the air.

Connectivity to the Internet enables many of these mobile uses. Since having a wired connection is impossible in cars, boats, and airplanes, there is a lot of interest in wireless networks. Cellular networks operated by the telephone companies are one familiar kind of wireless network that blankets us with coverage for mobile phones. Wireless **hotspots** based on the 802.11 standard are another kind of wireless network for mobile computers. They have sprung up everywhere that people go, resulting in a patchwork of coverage at cafes, hotels, airports, schools, trains and planes. Anyone with a laptop computer and a wireless modem can just turn on their computer on and be connected to the Internet through the hotspot, as though the computer were plugged into a wired network.

Wireless networks are of great value to fleets of trucks, taxis, delivery vehicles, and repairpersons for keeping in contact with their home base. For example, in many cities, taxi drivers are independent businessmen, rather than being employees of a taxi company. In some of these cities, the taxis have a display the driver can see. When a customer calls up, a central dispatcher types in the pickup and destination points. This information is displayed on the drivers' displays and a beep sounds. The first driver to hit a button on the display gets the call.

Wireless networks are also important to the military. If you have to be able to fight a war anywhere on Earth at short notice, counting on using the local networking infrastructure is probably not a good idea. It is better to bring your own.

Although wireless networking and mobile computing are often related, they are not identical, as Fig. 1-5 shows. Here we see a distinction between **fixed wireless** and **mobile wireless** networks. Even notebook computers are sometimes wired. For example, if a traveler plugs a notebook computer into the wired network jack in a hotel room, he has mobility without a wireless network.

Wireless	Mobile	Typical applications
No	No	Desktop computers in offices
No	Yes	A notebook computer used in a hotel room
Yes	No	Networks in unwired buildings
Yes	Yes	Store inventory with a handheld computer

Figure 1-5. Combinations of wireless networks and mobile computing.

Conversely, some wireless computers are not mobile. In the home, and in offices or hotels that lack suitable cabling, it can be more convenient to connect desktop computers or media players wirelessly than to install wires. Installing a wireless network may require little more than buying a small box with some electronics in it, unpacking it, and plugging it in. This solution may be far cheaper than having workmen put in cable ducts to wire the building.

Finally, there are also true mobile, wireless applications, such as people walking around stores with a handheld computers recording inventory. At many busy airports, car rental return clerks work in the parking lot with wireless mobile computers. They scan the barcodes or RFID chips of returning cars, and their mobile device, which has a built-in printer, calls the main computer, gets the rental information, and prints out the bill on the spot.

Perhaps the key driver of mobile, wireless applications is the mobile phone. **Text messaging** or **texting** is tremendously popular. It lets a mobile phone user type a short message that is then delivered by the cellular network to another mobile subscriber. Few people would have predicted ten years ago that having teenagers tediously typing short text messages on mobile phones would be an immense money maker for telephone companies. But texting (or **Short Message Service** as it is known outside the U.S.) is very profitable since it costs the carrier but a tiny fraction of one cent to relay a text message, a service for which they charge far more.

The long-awaited convergence of telephones and the Internet has finally arrived, and it will accelerate the growth of mobile applications. **Smart phones**, such as the popular iPhone, combine aspects of mobile phones and mobile computers. The (3G and 4G) cellular networks to which they connect can provide fast data services for using the Internet as well as handling phone calls. Many advanced phones connect to wireless hotspots too, and automatically switch between networks to choose the best option for the user.

Other consumer electronics devices can also use cellular and hotspot networks to stay connected to remote computers. Electronic book readers can download a newly purchased book or the next edition of a magazine or today's newspaper wherever they roam. Electronic picture frames can update their displays on cue with fresh images.

Since mobile phones know their locations, often because they are equipped with **GPS** (**Global Positioning System**) receivers, some services are intentionally location dependent. Mobile maps and directions are an obvious candidate as your GPS-enabled phone and car probably have a better idea of where you are than you do. So, too, are searches for a nearby bookstore or Chinese restaurant, or a local weather forecast. Other services may record location, such as annotating photos and videos with the place at which they were made. This annotation is known as "geo-tagging."

An area in which mobile phones are now starting to be used is **m-commerce** (**mobile-commerce**) (Senn, 2000). Short text messages from the mobile are used to authorize payments for food in vending machines, movie tickets, and other small items instead of cash and credit cards. The charge then appears on the mobile phone bill. When equipped with **NFC** (**Near Field Communication**) technology the mobile can act as an RFID smartcard and interact with a nearby reader for payment. The driving forces behind this phenomenon are the mobile device makers and network operators, who are trying hard to figure out how to get a piece of the e-commerce pie. From the store's point of view, this scheme may save them most of the credit card company's fee, which can be several percent.

Of course, this plan may backfire, since customers in a store might use the RFID or barcode readers on their mobile devices to check out competitors' prices before buying and use them to get a detailed report on where else an item can be purchased nearby and at what price.

One huge thing that m-commerce has going for it is that mobile phone users are accustomed to paying for everything (in contrast to Internet users, who expect everything to be free). If an Internet Web site charged a fee to allow its customers to pay by credit card, there would be an immense howling noise from the users. If, however, a mobile phone operator its customers to pay for items in a store by waving the phone at the cash register and then tacked on a fee for this convenience, it would probably be accepted as normal. Time will tell.

No doubt the uses of mobile and wireless computers will grow rapidly in the future as the size of computers shrinks, probably in ways no one can now foresee. Let us take a quick look at some possibilities. **Sensor networks** are made up of nodes that gather and wirelessly relay information they sense about the state of the physical world. The nodes may be part of familiar items such as cars or phones, or they may be small separate devices. For example, your car might gather data on its location, speed, vibration, and fuel efficiency from its on-board diagnostic system and upload this information to a database (Hull et al., 2006). Those data can help find potholes, plan trips around congested roads, and tell you if you are a "gas guzzler" compared to other drivers on the same stretch of road.

Sensor networks are revolutionizing science by providing a wealth of data on behavior that could not previously be observed. One example is tracking the migration of individual zebras by placing a small sensor on each animal (Juang et al., 2002). Researchers have packed a wireless computer into a cube 1 mm on edge (Warneke et al., 2001). With mobile computers this small, even small birds, rodents, and insects can be tracked.

Even mundane uses, such as in parking meters, can be significant because they make use of data that were not previously available. Wireless parking meters can accept credit or debit card payments with instant verification over the wireless link. They can also report when they are in use over the wireless network. This would let drivers download a recent parking map to their car so they can find an available spot more easily. Of course, when a meter expires, it might also check for the presence of a car (by bouncing a signal off it) and report the expiration to parking enforcement. It has been estimated that city governments in the U.S. alone could collect an additional $10 billion this way (Harte et al., 2000).

Wearable computers are another promising application. Smart watches with radios have been part of our mental space since their appearance in the Dick Tracy comic strip in 1946; now you can buy them. Other such devices may be implanted, such as pacemakers and insulin pumps. Some of these can be controlled over a wireless network. This lets doctors test and reconfigure them more easily. It could also lead to some nasty problems if the devices are as insecure as the average PC and can be hacked easily (Halperin et al., 2008).

1.1.4 Social Issues

Computer networks, like the printing press 500 years ago, allow ordinary citizens to distribute and view content in ways that were not previously possible. But along with the good comes the bad, as this new-found freedom brings with it many unsolved social, political, and ethical issues. Let us just briefly mention a few of them; a thorough study would require a full book, at least.

Social networks, message boards, content sharing sites, and a host of other applications allow people to share their views with like-minded individuals. As long as the subjects are restricted to technical topics or hobbies like gardening, not too many problems will arise.

The trouble comes with topics that people actually care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Worse yet, they may not be politically correct. Furthermore, opinions need not be limited to text; high-resolution color photographs and video clips are easily shared over computer networks. Some people take a live-and-let-live view, but others feel that posting certain material (e.g., verbal attacks on particular countries or religions, pornography, etc.) is simply unacceptable and that such content must be censored. Different countries have different and conflicting laws in this area. Thus, the debate rages.

In the past, people have sued network operators, claiming that they are responsible for the contents of what they carry, just as newspapers and magazines are. The inevitable response is that a network is like a telephone company or the post office and cannot be expected to police what its users say.

It should now come only as a slight surprise to learn that some network operators block content for their own reasons. Some users of peer-to-peer applications had their network service cut off because the network operators did not find it profitable to carry the large amounts of traffic sent by those applications. Those same operators would probably like to treat different companies differently. If you are a big company and pay well then you get good service, but if you are a small-time player, you get poor service. Opponents of this practice argue that peer-to-peer and other content should be treated in the same way because they are all just bits to the network. This argument for communications that are not differentiated by their content or source or who is providing the content is known as **network neutrality** (Wu, 2003). It is probably safe to say that this debate will go on for a while.

Many other parties are involved in the tussle over content. For instance, pirated music and movies fueled the massive growth of peer-to-peer networks, which did not please the copyright holders, who have threatened (and sometimes taken) legal action. There are now automated systems that search peer-to-peer networks and fire off warnings to network operators and users who are suspected of infringing copyright. In the United States, these warnings are known as **DMCA takedown notices** after the **Digital Millennium Copyright Act**. This search is an arms' race because it is hard to reliably catch copyright infringement. Even your printer might be mistaken for a culprit (Piatek et al., 2008).

Computer networks make it very easy to communicate. They also make it easy for the people who run the network to snoop on the traffic. This sets up conflicts over issues such as employee rights versus employer rights. Many people read and write email at work. Many employers have claimed the right to read and possibly censor employee messages, including messages sent from a home computer outside working hours. Not all employees agree with this, especially the latter part.

Another conflict is centered around government versus citizen's rights. The FBI has installed systems at many Internet service providers to snoop on all incoming and outgoing email for nuggets of interest. One early system was originally called Carnivore, but bad publicity caused it to be renamed to the more innocent-sounding DCS1000 (Blaze and Bellovin, 2000; Sobel, 2001; and Zacks, 2001). The goal of such systems is to spy on millions of people in the hope of perhaps finding information about illegal activities. Unfortunately for the spies, the Fourth Amendment to the U.S. Constitution prohibits government searches without a search warrant, but the government often ignores it.

Of course, the government does not have a monopoly on threatening people's privacy. The private sector does its bit too by **profiling** users. For example, small files called **cookies** that Web browsers store on users' computers allow companies to track users' activities in cyberspace and may also allow credit card numbers, social security numbers, and other confidential information to leak all over the Internet (Berghel, 2001). Companies that provide Web-based services may maintain large amounts of personal information about their users that allows them to study user activities directly. For example, Google can read your email and show you advertisements based on your interests if you use its email service, **Gmail**.

A new twist with mobile devices is location privacy (Beresford and Stajano, 2003). As part of the process of providing service to your mobile device the network operators learn where you are at different times of day. This allows them to track your movements. They may know which nightclub you frequent and which medical center you visit.

Computer networks also offer the potential to increase privacy by sending anonymous messages. In some situations, this capability may be desirable. Beyond preventing companies from learning your habits, it provides, for example, a way for students, soldiers, employees, and citizens to blow the whistle on illegal behavior on the part of professors, officers, superiors, and politicians without fear of reprisals. On the other hand, in the United States and most other democracies, the law specifically permits an accused person the right to confront and challenge his accuser in court so anonymous accusations cannot be used as evidence.

The Internet makes it possible to find information quickly, but a great deal of it is ill considered, misleading, or downright wrong. That medical advice you plucked from the Internet about the pain in your chest may have come from a Nobel Prize winner or from a high-school dropout.

Other information is frequently unwanted. Electronic junk mail (spam) has become a part of life because spammers have collected millions of email addresses and would-be marketers can cheaply send computer-generated messages to them. The resulting flood of spam rivals the flow messages from real people. Fortunately, filtering software is able to read and discard the spam generated by other computers, with lesser or greater degrees of success.

Still other content is intended for criminal behavior. Web pages and email messages containing active content (basically, programs or macros that execute on the receiver's machine) can contain viruses that take over your computer. They might be used to steal your bank account passwords, or to have your computer send spam as part of a **botnet** or pool of compromised machines.

Phishing messages masquerade as originating from a trustworthy party, for example, your bank, to try to trick you into revealing sensitive information, for example, credit card numbers. Identity theft is becoming a serious problem as thieves collect enough information about a victim to obtain credit cards and other documents in the victim's name.

It can be difficult to prevent computers from impersonating people on the Internet. This problem has led to the development of **CAPTCHAs**, in which a computer asks a person to solve a short recognition task, for example, typing in the letters shown in a distorted image, to show that they are human (von Ahn, 2001). This process is a variation on the famous Turing test in which a person asks questions over a network to judge whether the entity responding is human.

A lot of these problems could be solved if the computer industry took computer security seriously. If all messages were encrypted and authenticated, it would be harder to commit mischief. Such technology is well established and we will study it in detail in Chap. 8. The problem is that hardware and software vendors know that putting in security features costs money and their customers are not demanding such features. In addition, a substantial number of the problems are caused by buggy software, which occurs because vendors keep adding more and more features to their programs, which inevitably means more code and thus more bugs. A tax on new features might help, but that might be a tough sell in some quarters. A refund for defective software might be nice, except it would bankrupt the entire software industry in the first year.

Computer networks raise new legal problems when they interact with old laws. Electronic gambling provides an example. Computers have been simulating things for decades, so why not simulate slot machines, roulette wheels, blackjack dealers, and more gambling equipment? Well, because it is illegal in a lot of places. The trouble is, gambling is legal in a lot of other places (England, for example) and casino owners there have grasped the potential for Internet gambling. What happens if the gambler, the casino, and the server are all in different countries, with conflicting laws? Good question.

1.2 NETWORK HARDWARE

It is now time to turn our attention from the applications and social aspects of networking (the dessert) to the technical issues involved in network design (the spinach). There is no generally accepted taxonomy into which all computer networks fit, but two dimensions stand out as important: transmission technology and scale. We will now examine each of these in turn.

Broadly speaking, there are two types of transmission technology that are in widespread use: **broadcast** links and **point-to-point** links.

Point-to-point links connect individual pairs of machines. To go from the source to the destination on a network made up of point-to-point links, short messages, called **packets** in certain contexts, may have to first visit one or more intermediate machines. Often multiple routes, of different lengths, are possible, so finding good ones is important in point-to-point networks. Point-to-point transmission with exactly one sender and exactly one receiver is sometimes called **unicasting**.

In contrast, on a broadcast network, the communication channel is shared by all the machines on the network; packets sent by any machine are received by all the others. An address field within each packet specifies the intended recipient. Upon receiving a packet, a machine checks the address field. If the packet is intended for the receiving machine, that machine processes the packet; if the packet is intended for some other machine, it is just ignored.

A wireless network is a common example of a broadcast link, with communication shared over a coverage region that depends on the wireless channel and the transmitting machine. As an analogy, consider someone standing in a meeting room and shouting "Watson, come here. I want you." Although the packet may actually be received (heard) by many people, only Watson will respond; the others just ignore it.

Broadcast systems usually also allow the possibility of addressing a packet to *all* destinations by using a special code in the address field. When a packet with this code is transmitted, it is received and processed by every machine on the network. This mode of operation is called **broadcasting**. Some broadcast systems also support transmission to a subset of the machines, which known as **multicasting**.

An alternative criterion for classifying networks is by scale. Distance is important as a classification metric because different technologies are used at different scales.

In Fig. 1-6 we classify multiple processor systems by their rough physical size. At the top are the personal area networks, networks that are meant for one person. Beyond these come longer-range networks. These can be divided into local, metropolitan, and wide area networks, each with increasing scale. Finally, the connection of two or more networks is called an internetwork. The worldwide Internet is certainly the best-known (but not the only) example of an internetwork.

Soon we will have even larger internetworks with the **Interplanetary Internet** that connects networks across space (Burleigh et al., 2003).

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	Local area network
1 km	Campus	Local area network
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	Wide area network
10,000 km	Planet	The Internet

Figure 1-6. Classification of interconnected processors by scale.

In this book we will be concerned with networks at all these scales. In the following sections, we give a brief introduction to network hardware by scale.

1.2.1 Personal Area Networks

PANs (**Personal Area Networks**) let devices communicate over the range of a person. A common example is a wireless network that connects a computer with its peripherals. Almost every computer has an attached monitor, keyboard, mouse, and printer. Without using wireless, this connection must be done with cables. So many new users have a hard time finding the right cables and plugging them into the right little holes (even though they are usually color coded) that most computer vendors offer the option of sending a technician to the user's home to do it. To help these users, some companies got together to design a short-range wireless network called **Bluetooth** to connect these components without wires. The idea is that if your devices have Bluetooth, then you need no cables. You just put them down, turn them on, and they work together. For many people, this ease of operation is a big plus.

In the simplest form, Bluetooth networks use the master-slave paradigm of Fig. 1-7. The system unit (the PC) is normally the master, talking to the mouse, keyboard, etc., as slaves. The master tells the slaves what addresses to use, when they can broadcast, how long they can transmit, what frequencies they can use, and so on.

Bluetooth can be used in other settings, too. It is often used to connect a headset to a mobile phone without cords and it can allow your digital music player

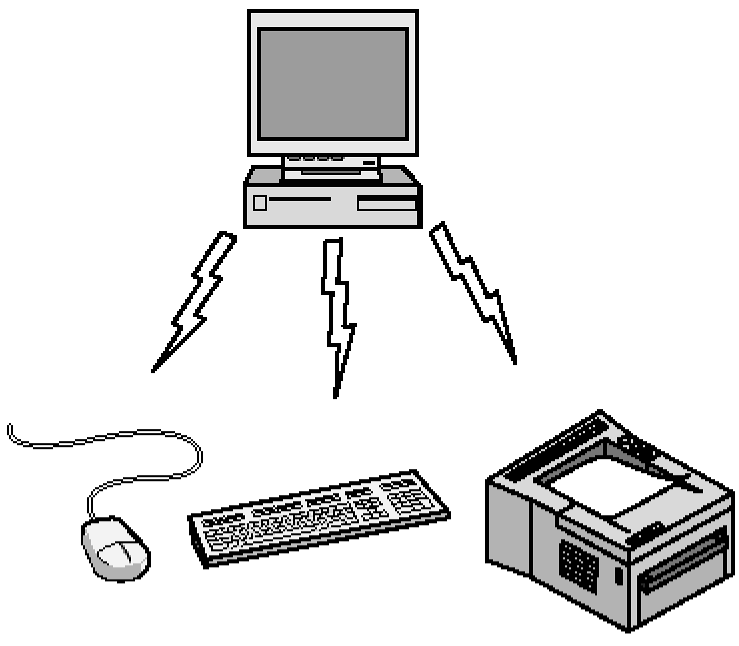


Figure 1-7. Bluetooth PAN configuration.

to connect to your car merely being brought within range. A completely different kind of PAN is formed when an embedded medical device such as a pacemaker, insulin pump, or hearing aid talks to a user-operated remote control. We will discuss Bluetooth in more detail in Chap. 4.

PANs can also be built with other technologies that communicate over short ranges, such as RFID on smartcards and library books. We will study RFID in Chap. 4.

1.2.2 Local Area Networks

The next step up is the **LAN** (**Local Area Network**). A LAN is a privately owned network that operates within and nearby a single building like a home, office or factory. LANs are widely used to connect personal computers and consumer electronics to let them share resources (e.g., printers) and exchange information. When LANs are used by companies, they are called **enterprise networks**.

Wireless LANs are very popular these days, especially in homes, older office buildings, cafeterias, and other places where it is too much trouble to install cables. In these systems, every computer has a radio modem and an antenna that it uses to communicate with other computers. In most cases, each computer talks to a device in the ceiling as shown in Fig. 1-8(a). This device, called an **AP** (**Access Point**), **wireless router**, or **base station**, relays packets between the wireless computers and also between them and the Internet. Being the AP is like being the popular kid as school because everyone wants to talk to you. However, if other computers are close enough, they can communicate directly with one another in a peer-to-peer configuration.

There is a standard for wireless LANs called **IEEE 802.11**, popularly known as **WiFi**, which has become very widespread. It runs at speeds anywhere from 11

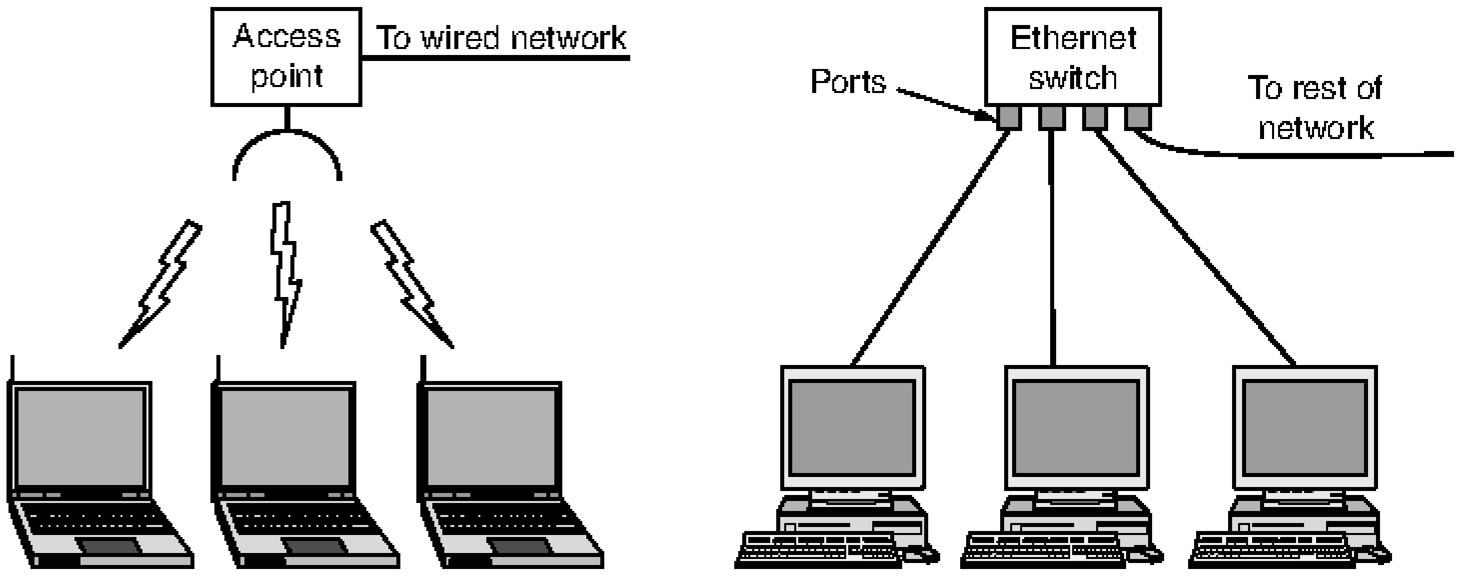


Figure 1-8. Wireless and wired LANs. (a) 802.11. (b) Switched Ethernet.

to hundreds of Mbps. (In this book we will adhere to tradition and measure line speeds in megabits/sec, where 1 Mbps is 1,000,000 bits/sec, and gigabits/sec, where 1 Gbps is 1,000,000,000 bits/sec.) We will discuss 802.11 in Chap. 4.

Wired LANs use a range of different transmission technologies. Most of them use copper wires, but some use optical fiber. LANs are restricted in size, which means that the worst-case transmission time is bounded and known in advance. Knowing these bounds helps with the task of designing network protocols. Typically, wired LANs run at speeds of 100 Mbps to 1 Gbps, have low delay (microseconds or nanoseconds), and make very few errors. Newer LANs can operate at up to 10 Gbps. Compared to wireless networks, wired LANs exceed them in all dimensions of performance. It is just easier to send signals over a wire or through a fiber than through the air.

The topology of many wired LANs is built from point-to-point links. IEEE 802.3, popularly called **Ethernet**, is, by far, the most common type of wired LAN. Fig. 1-8(b) shows a sample topology of **switched Ethernet**. Each computer speaks the Ethernet protocol and connects to a box called a **switch** with a point-to-point link. Hence the name. A switch has multiple **ports**, each of which can connect to one computer. The job of the switch is to relay packets between computers that are attached to it, using the address in each packet to determine which computer to send it to.

To build larger LANs, switches can be plugged into each other using their ports. What happens if you plug them together in a loop? Will the network still work? Luckily, the designers thought of this case. It is the job of the protocol to sort out what paths packets should travel to safely reach the intended computer. We will see how this works in Chap. 4.

It is also possible to divide one large physical LAN into two smaller logical LANs. You might wonder why this would be useful. Sometimes, the layout of the network equipment does not match the organization's structure. For example, the engineering and finance departments of a company might have computers on the same physical LAN because they are in the same wing of the building but it might be easier to manage the system if engineering and finance logically each had its own network **Virtual LAN** or **VLAN**. In this design each port is tagged with a "color," say green for engineering and red for finance. The switch then forwards packets so that computers attached to the green ports are separated from the computers attached to the red ports. Broadcast packets sent on a red port, for example, will not be received on a green port, just as though there were two different LANs. We will cover VLANs at the end of Chap. 4.

There are other wired LAN topologies too. In fact, switched Ethernet is a modern version of the original Ethernet design that broadcast all the packets over a single linear cable. At most one machine could successfully transmit at a time, and a distributed arbitration mechanism was used to resolve conflicts. It used a simple algorithm: computers could transmit whenever the cable was idle. If two or more packets collided, each computer just waited a random time and tried later. We will call that version **classic Ethernet** for clarity, and as you suspected, you will learn about it in Chap. 4.

Both wireless and wired broadcast networks can be divided into static and dynamic designs, depending on how the channel is allocated. A typical static allocation would be to divide time into discrete intervals and use a round-robin algorithm, allowing each machine to broadcast only when its time slot comes up. Static allocation wastes channel capacity when a machine has nothing to say during its allocated slot, so most systems attempt to allocate the channel dynamically (i.e., on demand).

Dynamic allocation methods for a common channel are either centralized or decentralized. In the centralized channel allocation method, there is a single entity, for example, the base station in cellular networks, which determines who goes next. It might do this by accepting multiple packets and prioritizing them according to some internal algorithm. In the decentralized channel allocation method, there is no central entity; each machine must decide for itself whether to transmit. You might think that this approach would lead to chaos, but it does not. Later we will study many algorithms designed to bring order out of the potential chaos.

It is worth spending a little more time discussing LANs in the home. In the future, it is likely that every appliance in the home will be capable of communicating with every other appliance, and all of them will be accessible over the Internet. This development is likely to be one of those visionary concepts that nobody asked for (like TV remote controls or mobile phones), but once they arrived nobody can imagine how they lived without them.

Many devices are already capable of being networked. These include computers, entertainment devices such as TVs and DVDs, phones and other consumer electronics such as cameras, appliances like clock radios, and infrastructure like utility meters and thermostats. This trend will only continue. For instance, the average home probably has a dozen clocks (e.g., in appliances), all of which could adjust to daylight savings time automatically if the clocks were on the Internet. Remote monitoring of the home is a likely winner, as many grown children would be willing to spend some money to help their aging parents live safely in their own homes.

While we could think of the home network as just another LAN, it is more likely to have different properties than other networks. First, the networked devices have to be very easy to install. Wireless routers are the most returned consumer electronic item. People buy one because they want a wireless network at home, find that it does not work "out of the box," and then return it rather than listen to elevator music while on hold on the technical helpline.

Second, the network and devices have to be foolproof in operation. Air conditioners used to have one knob with four settings: OFF, LOW, MEDIUM, and HIGH. Now they have 30-page manuals. Once they are networked, expect the chapter on security alone to be 30 pages. This is a problem because only computer users are accustomed to putting up with products that do not work; the car-, television-, and refrigerator-buying public is far less tolerant. They expect products to work 100% without the need to hire a geek.

Third, low price is essential for success. People will not pay a $50 premium for an Internet thermostat because few people regard monitoring their home temperature from work that important. For $5 extra, though, it might sell.

Fourth, it must be possible to start out with one or two devices and expand the reach of the network gradually. This means no format wars. Telling consumers to buy peripherals with IEEE 1394 (FireWire) interfaces and a few years later retracting that and saying USB 2.0 is the interface-of-the-month and then switching that to 802.11g—oops, no, make that 802.11n—I mean 802.16 (different wireless networks)—is going to make consumers very skittish. The network interface will have to remain stable for decades, like the television broadcasting standards.

Fifth, security and reliability will be very important. Losing a few files to an email virus is one thing; having a burglar disarm your security system from his mobile computer and then plunder your house is something quite different.

An interesting question is whether home networks will be wired or wireless. Convenience and cost favors wireless networking because there are no wires to fit, or worse, retrofit. Security favors wired networking because the radio waves that wireless networks use are quite good at going through walls. Not everyone is overjoyed at the thought of having the neighbors piggybacking on their Internet connection and reading their email. In Chap. 8 we will study how encryption can be used to provide security, but it is easier said than done with inexperienced users.

A third option that may be appealing is to reuse the networks that are already in the home. The obvious candidate is the electric wires that are installed throughout the house. **Power-line networks** let devices that plug into outlets broadcast information throughout the house. You have to plug in the TV anyway, and this way it can get Internet connectivity at the same time. The difficulty is how to carry both power and data signals at the same time. Part of the answer is that they use different frequency bands.

In short, home LANs offer many opportunities and challenges. Most of the latter relate to the need for the networks to be easy to manage, dependable, and secure, especially in the hands of nontechnical users, as well as low cost.

1.2.3 Metropolitan Area Networks

A **MAN (Metropolitan Area Network)** covers a city. The best-known examples of MANs are the cable television networks available in many cities. These systems grew from earlier community antenna systems used in areas with poor over-the-air television reception. In those early systems, a large antenna was placed on top of a nearby hill and a signal was then piped to the subscribers' houses.

At first, these were locally designed, ad hoc systems. Then companies began jumping into the business, getting contracts from local governments to wire up entire cities. The next step was television programming and even entire channels designed for cable only. Often these channels were highly specialized, such as all news, all sports, all cooking, all gardening, and so on. But from their inception until the late 1990s, they were intended for television reception only.

When the Internet began attracting a mass audience, the cable TV network operators began to realize that with some changes to the system, they could provide two-way Internet service in unused parts of the spectrum. At that point, the cable TV system began to morph from simply a way to distribute television to a metropolitan area network. To a first approximation, a MAN might look something like the system shown in Fig. 1-9. In this figure we see both television signals and Internet being fed into the centralized **cable headend** for subsequent distribution to people's homes. We will come back to this subject in detail in Chap. 2.

Cable television is not the only MAN, though. Recent developments in high-speed wireless Internet access have resulted in another MAN, which has been standardized as IEEE 802.16 and is popularly known as **WiMAX**. We will look at it in Chap. 4.

1.2.4 Wide Area Networks

A **WAN (Wide Area Network)** spans a large geographical area, often a country or continent. We will begin our discussion with wired WANs, using the example of a company with branch offices in different cities.

The WAN in Fig. 1-10 is a network that connects offices in Perth, Melbourne, and Brisbane. Each of these offices contains computers intended for running user (i.e., application) programs. We will follow traditional usage and call these machines **hosts**. The rest of the network that connects these hosts is then called the

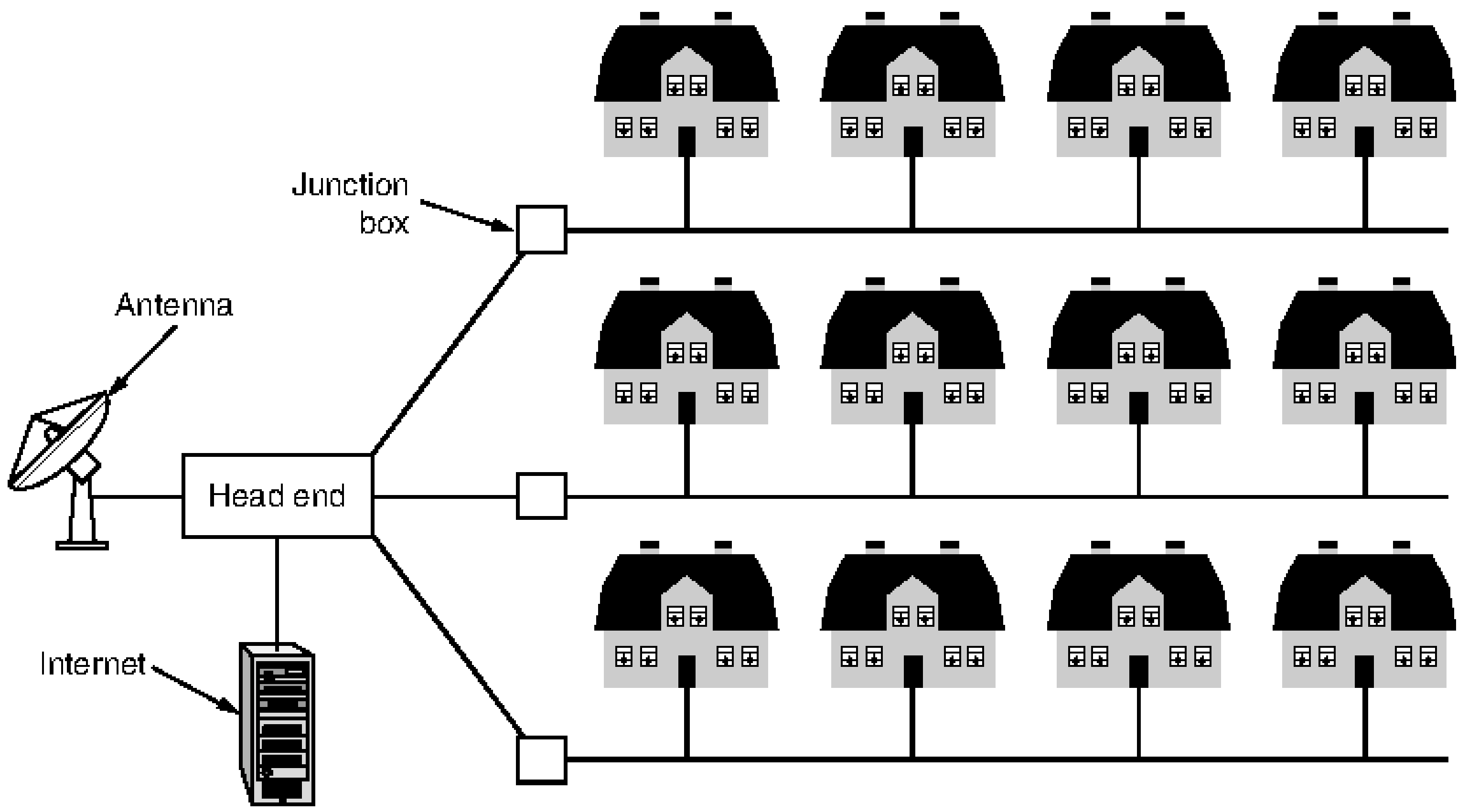


Figure 1-9. A metropolitan area network based on cable TV.

communication subnet, or just **subnet** for short. The job of the subnet is to carry messages from host to host, just as the telephone system carries words (really just sounds) from speaker to listener.

In most WANs, the subnet consists of two distinct components: transmission lines and switching elements. **Transmission lines** move bits between machines. They can be made of copper wire, optical fiber, or even radio links. Most companies do not have transmission lines lying about, so instead they lease the lines from a telecommunications company. **Switching elements**, or just **switches**, are specialized computers that connect two or more transmission lines. When data arrive on an incoming line, the switching element must choose an outgoing line on which to forward them. These switching computers have been called by various names in the past; the name **router** is now most commonly used. Unfortunately, some people pronounce it "rooter" while others have it rhyme with "doubter." Determining the correct pronunciation will be left as an exercise for the reader. (Note: the perceived correct answer may depend on where you live.)

A short comment about the term "subnet" is in order here. Originally, its **only** meaning was the collection of routers and communication lines that moved packets from the source host to the destination host. Readers should be aware that it has acquired a second, more recent meaning in conjunction with network addressing. We will discuss that meaning in Chap. 5 and stick with the original meaning (a collection of lines and routers) until then.

The WAN as we have described it looks similar to a large wired LAN, but there are some important differences that go beyond long wires. Usually in a WAN, the hosts and subnet are owned and operated by different people. In our

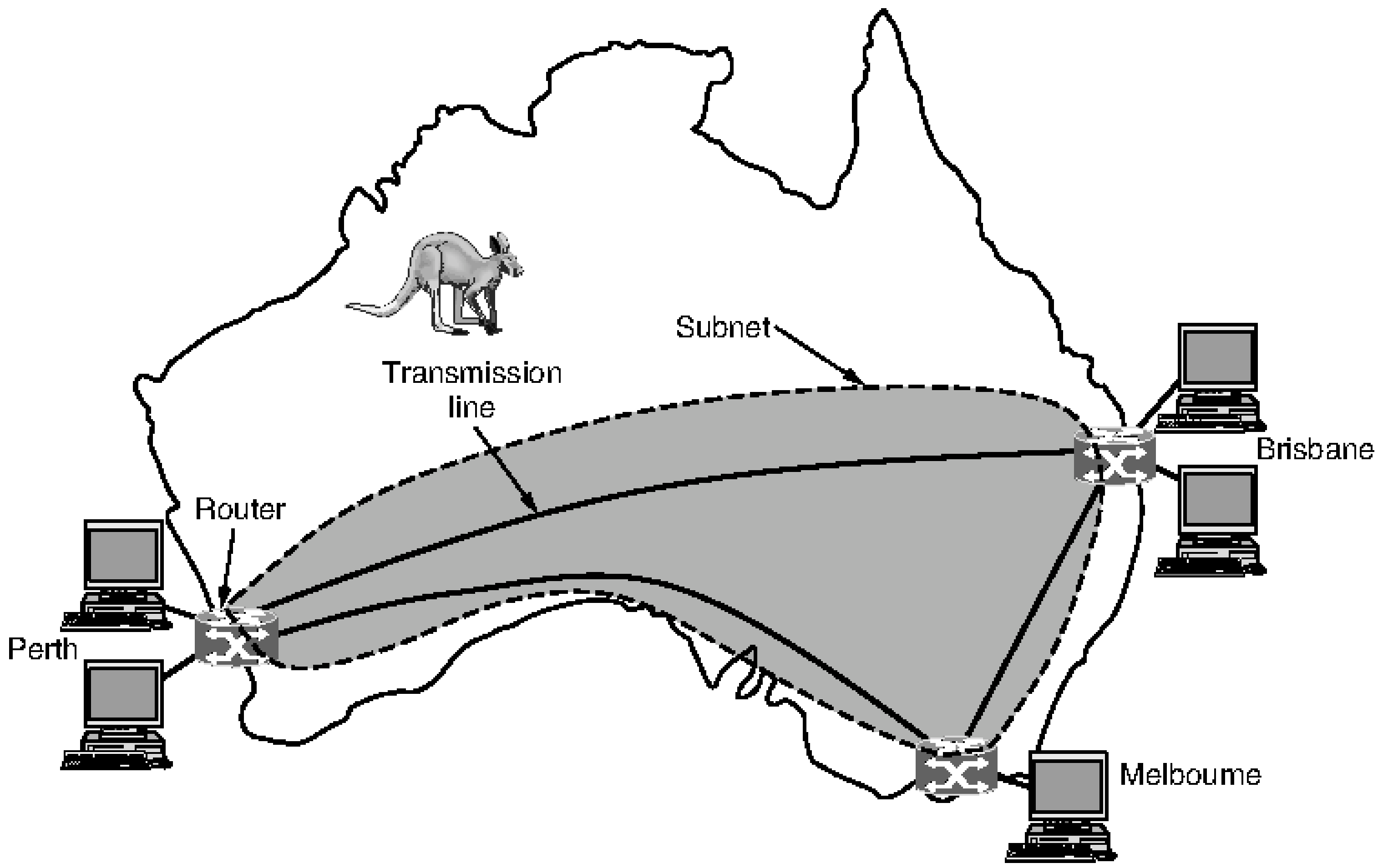


Figure 1-10. WAN that connects three branch offices in Australia.

example, the employees might be responsible for their own computers, while the company's IT department is in charge of the rest of the network. We will see clearer boundaries in the coming examples, in which the network provider or telephone company operates the subnet. Separation of the pure communication aspects of the network (the subnet) from the application aspects (the hosts) greatly simplifies the overall network design.

A second difference is that the routers will usually connect different kinds of networking technology. The networks inside the offices may be switched Ethernet, for example, while the long-distance transmission lines may be SONET links (which we will cover in Chap. 2). Some device needs to join them. The astute reader will notice that this goes beyond our definition of a network. This means that many WANs will in fact be **internetworks**, or composite networks that are made up of more than one network. We will have more to say about internetworks in the next section.

A final difference is in what is connected to the subnet. This could be individual computers, as was the case for connecting to LANs, or it could be entire LANs. This is how larger networks are built from smaller ones. As far as the subnet is concerned, it does the same job.

We are now in a position to look at two other varieties of WANs. First, rather than lease dedicated transmission lines, a company might connect its offices to the Internet This allows connections to be made between the offices as virtual links that use the underlying capacity of the Internet. This arrangement, shown in Fig. 1-11, is called a **VPN** (**Virtual Private Network**). Compared to the dedicated arrangement, a VPN has the usual advantage of virtualization, which is that it provides flexible reuse of a resource (Internet connectivity). Consider how easy it is to add a fourth office to see this. A VPN also has the usual disadvantage of virtualization, which is a lack of control over the underlying resources. With a dedicated line, the capacity is clear. With a VPN your mileage may vary with your Internet service.

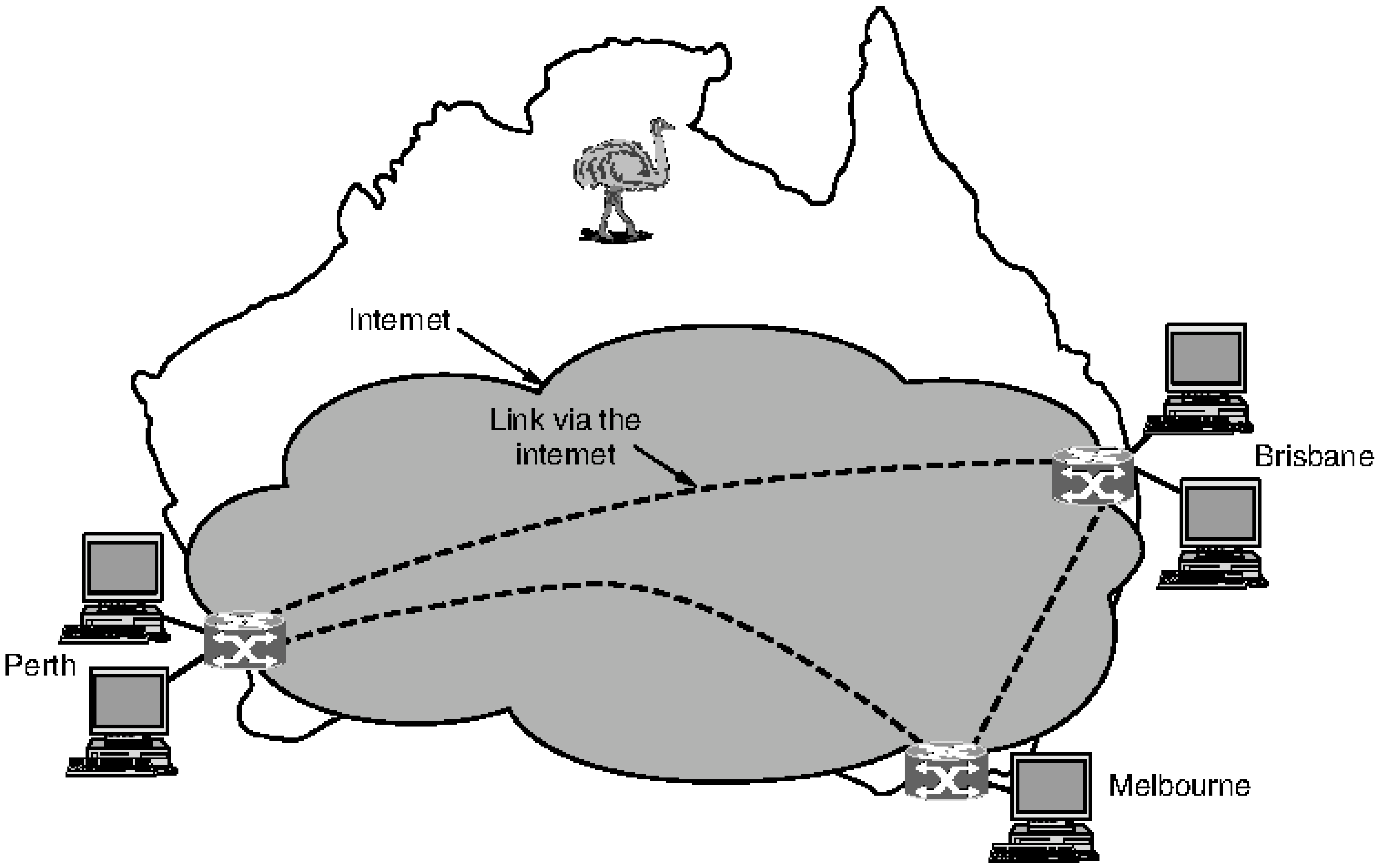


Figure 1-11. WAN using a virtual private network.

The second variation is that the subnet may be run by a different company. The subnet operator is known as a **network service provider** and the offices are its customers. This structure is shown in Fig. 1-12. The subnet operator will connect to other customers too, as long as they can pay and it can provide service. Since it would be a disappointing network service if the customers could only send packets to each other, the subnet operator will also connect to other networks that are part of the Internet. Such a subnet operator is called an **ISP** (**Internet Service Provider**) and the subnet is an **ISP network**. Its customers who connect to the ISP receive Internet service.

We can use the ISP network to preview some key issues that we will study in later chapters. In most WANs, the network contains many transmission lines, each connecting a pair of routers. If two routers that do not share a transmission line wish to communicate, they must do this indirectly, via other routers. There

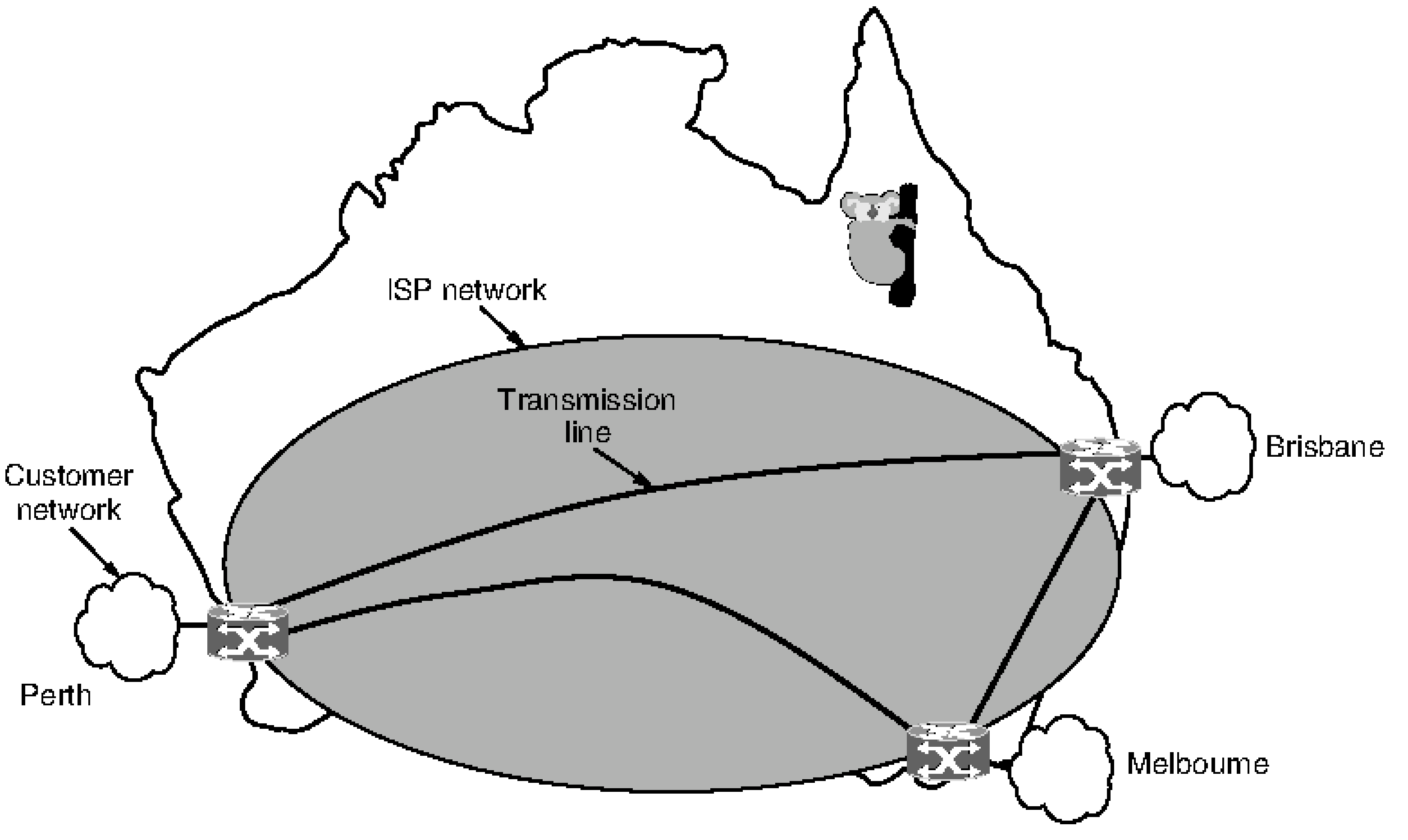


Figure 1-12. WAN using an ISP network.

may be many paths in the network that connect these two routers. How the network makes the decision as to which path to use is called the **routing algorithm**. Many such algorithms exist. How each router makes the decision as to where to send a packet next is called the **forwarding algorithm**. Many of them exist too. We will study some of both types in detail in Chap. 5.

Other kinds of WANs make heavy use of wireless technologies. In satellite systems, each computer on the ground has an antenna through which it can send data to and receive data from to a satellite in orbit. All computers can hear the output *from* the satellite, and in some cases they can also hear the upward transmissions of their fellow computers *to* the satellite as well. Satellite networks are inherently broadcast and are most useful when the broadcast property is important.

The cellular telephone network is another example of a WAN that uses wireless technology. This system has already gone through three generations and a fourth one is on the horizon. The first generation was analog and for voice only. The second generation was digital and for voice only. The third generation is digital and is for both voice and data. Each cellular base station covers a distance much larger than a wireless LAN, with a range measured in kilometers rather than tens of meters. The base stations are connected to each other by a backbone network that is usually wired. The data rates of cellular networks are often on the order of 1 Mbps, much smaller than a wireless LAN that can range up to on the order of 100 Mbps. We will have a lot to say about these networks in Chap. 2.

1.2.5 Internetworks

Many networks exist in the world, often with different hardware and software. People connected to one network often want to communicate with people attached to a different one. The fulfillment of this desire requires that different, and frequently incompatible, networks be connected. A collection of interconnected networks is called an **internetwork** or **internet**. These terms will be used in a generic sense, in contrast to the worldwide Internet (which is one specific internet), which we will always capitalize. The Internet uses ISP networks to connect enterprise networks, home networks, and many other networks. We will look at the Internet in great detail later in this book.

Subnets, networks, and internetworks are often confused. The term "subnet" makes the most sense in the context of a wide area network, where it refers to the collection of routers and communication lines owned by the network operator. As an analogy, the telephone system consists of telephone switching offices connected to one another by high-speed lines, and to houses and businesses by low-speed lines. These lines and equipment, owned and managed by the telephone company, form the subnet of the telephone system. The telephones themselves (the hosts in this analogy) are not part of the subnet.

A network is formed by the combination of a subnet and its hosts. However, the word "network" is often used in a loose sense as well. A subnet might be described as a network, as in the case of the "ISP network" of Fig. 1-12. An internetwork might also be described as a network, as in the case of the WAN in Fig. 1-10. We will follow similar practice, and if we are distinguishing a network from other arrangements, we will stick with our original definition of a collection of computers interconnected by a single technology.

Let us say more about what constitutes an internetwork. We know that an internet is formed when distinct networks are interconnected. In our view, connecting a LAN and a WAN or connecting two LANs is the usual way to form an internetwork, but there is little agreement in the industry over terminology in this area. There are two rules of thumb that are useful. First, if different organizations have paid to construct different parts of the network and each maintains its part, we have an internetwork rather than a single network. Second, if the underlying technology is different in different parts (e.g., broadcast versus point-to-point and wired versus wireless), we probably have an internetwork.

To go deeper, we need to talk about how two different networks can be connected. The general name for a machine that makes a connection between two or more networks and provides the necessary translation, both in terms of hardware and software, is a **gateway**. Gateways are distinguished by the layer at which they operate in the protocol hierarchy. We will have much more to say about layers and protocol hierarchies starting in the next section, but for now imagine that higher layers are more tied to applications, such as the Web, and lower layers are more tied to transmission links, such as Ethernet.

Since the benefit of forming an internet is to connect computers across networks, we do not want to use too low-level a gateway or we will be unable to make connections between different kinds of networks. We do not want to use too high-level a gateway either, or the connection will only work for particular applications. The level in the middle that is "just right" is often called the network layer, and a router is a gateway that switches packets at the network layer. We can now spot an internet by finding a network that has routers.

1.3 NETWORK SOFTWARE

The first computer networks were designed with the hardware as the main concern and the software as an afterthought. This strategy no longer works. Network software is now highly structured. In the following sections we examine the software structuring technique in some detail. The approach described here forms the keystone of the entire book and will occur repeatedly later on.

1.3.1 Protocol Hierarchies

To reduce their design complexity, most networks are organized as a stack of **layers** or **levels**, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. The purpose of each layer is to offer certain services to the higher layers while shielding those layers from the details of how the offered services are actually implemented. In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.

This concept is actually a familiar one and is used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

When layer n on one machine carries on a conversation with layer n on another machine, the rules and conventions used in this conversation are collectively known as the layer n protocol. Basically, a **protocol** is an agreement between the communicating parties on how communication is to proceed. As an analogy, when a woman is introduced to a man, she may choose to stick out her hand. He, in turn, may decide to either shake it or kiss it, depending, for example, on whether she is an American lawyer at a business meeting or a European princess at a formal ball. Violating the protocol will make communication more difficult, if not completely impossible.

A five-layer network is illustrated in Fig. 1-13. The entities comprising the corresponding layers on different machines are called **peers**. The peers may be software processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol to talk to each other.

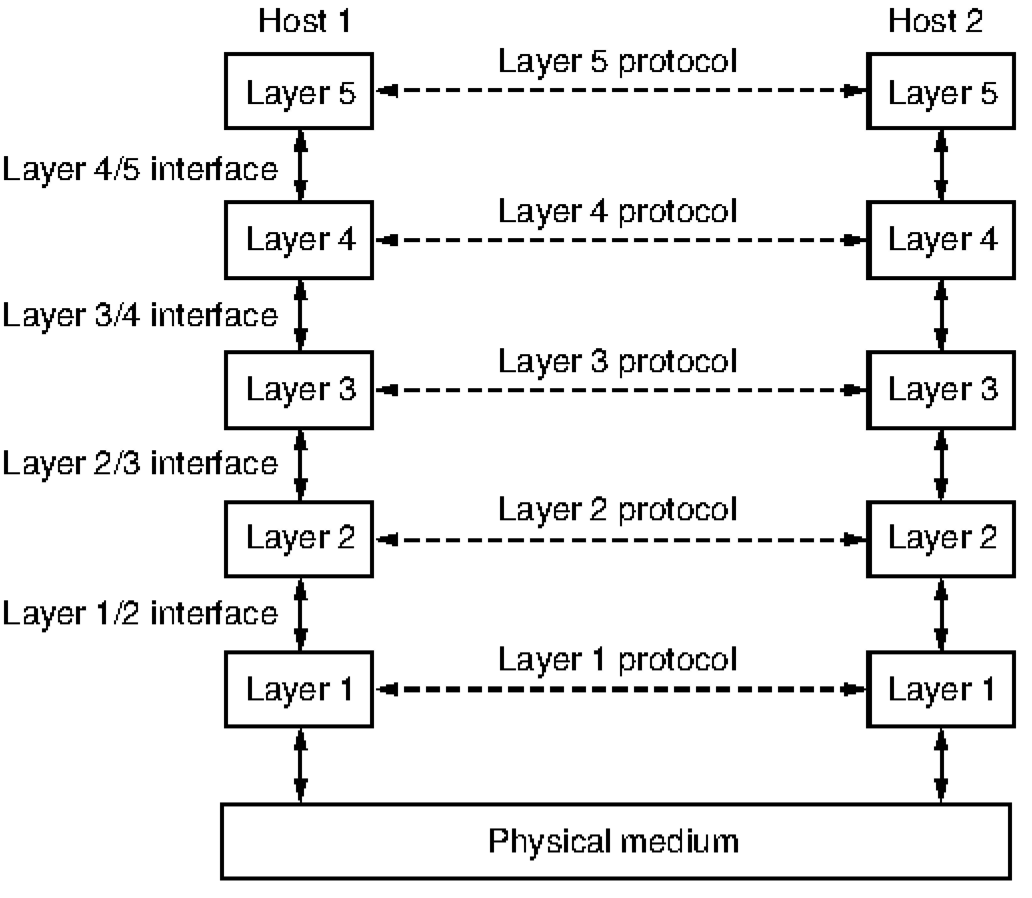


Figure 1-13. Layers, protocols, and interfaces.

In reality, no data are directly transferred from layer *n* on one machine to layer *n* on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the **physical medium** through which actual communication occurs. In Fig. 1-13, virtual communication is shown by dotted lines and physical communication by solid lines.

Between each pair of adjacent layers is an **interface**. The interface defines which primitive operations and services the lower layer makes available to the upper one. When network designers decide how many layers to include in a network and what each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so, in turn, requires that each layer perform a specific collection of well-understood functions. In addition to minimizing the amount of information that must be passed between layers, clear-cut interfaces also make it simpler to replace one layer with a completely different protocol or implementation (e.g., replacing all the telephone lines by satellite channels) because all that is required of the new protocol or implementation is that it offer exactly the same set of services to its upstairs neighbor as the old one did. It is common that different hosts use different implementations of the same protocol (often written by different companies). In fact, the protocol itself can change in some layer without the layers above and below it even noticing.

A set of layers and protocols is called a **network architecture**. The specification of an architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Neither the details of the implementation nor the specification of the interfaces is part of the architecture because these are hidden away inside the machines and not visible from the outside. It is not even necessary that the interfaces on all machines in a network be the same, provided that each machine can correctly use all the protocols. A list of the protocols used by a certain system, one protocol per layer, is called a **protocol stack**. Network architectures, protocol stacks, and the protocols themselves are the principal subjects of this book.

An analogy may help explain the idea of multilayer communication. Imagine two philosophers (peer processes in layer 3), one of whom speaks Urdu and English and one of whom speaks Chinese and French. Since they have no common language, they each engage a translator (peer processes at layer 2), each of whom in turn contacts a secretary (peer processes in layer 1). Philosopher 1 wishes to convey his affection for *oryctolagus cuniculus* to his peer. To do so, he passes a message (in English) across the 2/3 interface to his translator, saying "I like rabbits," as illustrated in Fig. 1-14. The translators have agreed on a neutral language known to both of them, Dutch, so the message is converted to "Ik vind konijnen leuk." The choice of the language is the layer 2 protocol and is up to the layer 2 peer processes.

The translator then gives the message to a secretary for transmission, for example, by email (the layer 1 protocol). When the message arrives at the other secretary, it is passed to the local translator, who translates it into French and passes it across the 2/3 interface to the second philosopher. Note that each protocol is completely independent of the other ones as long as the interfaces are not changed. The translators can switch from Dutch to, say, Finnish, at will, provided that they both agree and neither changes his interface with either layer 1 or layer 3. Similarly, the secretaries can switch from email to telephone without disturbing (or even informing) the other layers. Each process may add some information intended only for its peer. This information is not passed up to the layer above.

Now consider a more technical example: how to provide communication to the top layer of the five-layer network in Fig. 1-15. A message, M, is produced by an application process running in layer 5 and given to layer 4 for transmission. Layer 4 puts a **header** in front of the message to identify the message and passes the result to layer 3. The header includes control information, such as addresses, to allow layer 4 on the destination machine to deliver the message. Other examples of control information used in some layers are sequence numbers (in case the lower layer does not preserve message order), sizes, and times.

In many networks, no limit is placed on the size of messages transmitted in the layer 4 protocol but there is nearly always a limit imposed by the layer 3 protocol. Consequently, layer 3 must break up the incoming messages into smaller

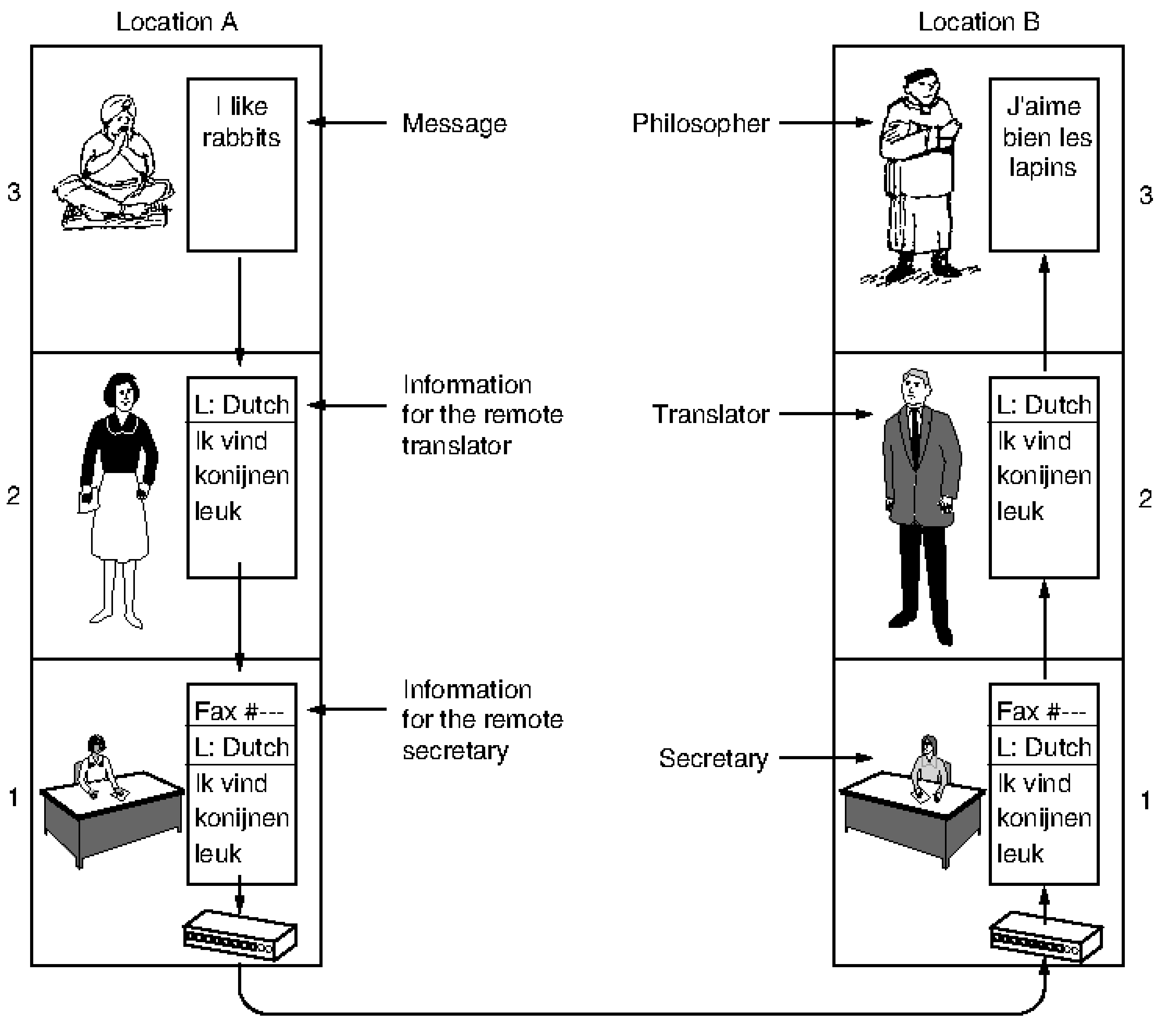


Figure 1-14. The philosopher-translator-secretary architecture.

units, packets, prepending a layer 3 header to each packet. In this example, M is split into two parts, M_1 and M_2, that will be transmitted separately.

Layer 3 decides which of the outgoing lines to use and passes the packets to layer 2. Layer 2 adds to each piece not only a header but also a trailer, and gives the resulting unit to layer 1 for physical transmission. At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses. None of the headers for layers below n are passed up to layer n.

The important thing to understand about Fig. 1-15 is the relation between the virtual and actual communication and the difference between protocols and interfaces. The peer processes in layer 4, for example, conceptually think of their communication as being "horizontal," using the layer 4 protocol. Each one is likely to have procedures called something like *SendToOtherSide* and *GetFromOtherSide*, even though these procedures actually communicate with lower layers across the 3/4 interface, and not with the other side.

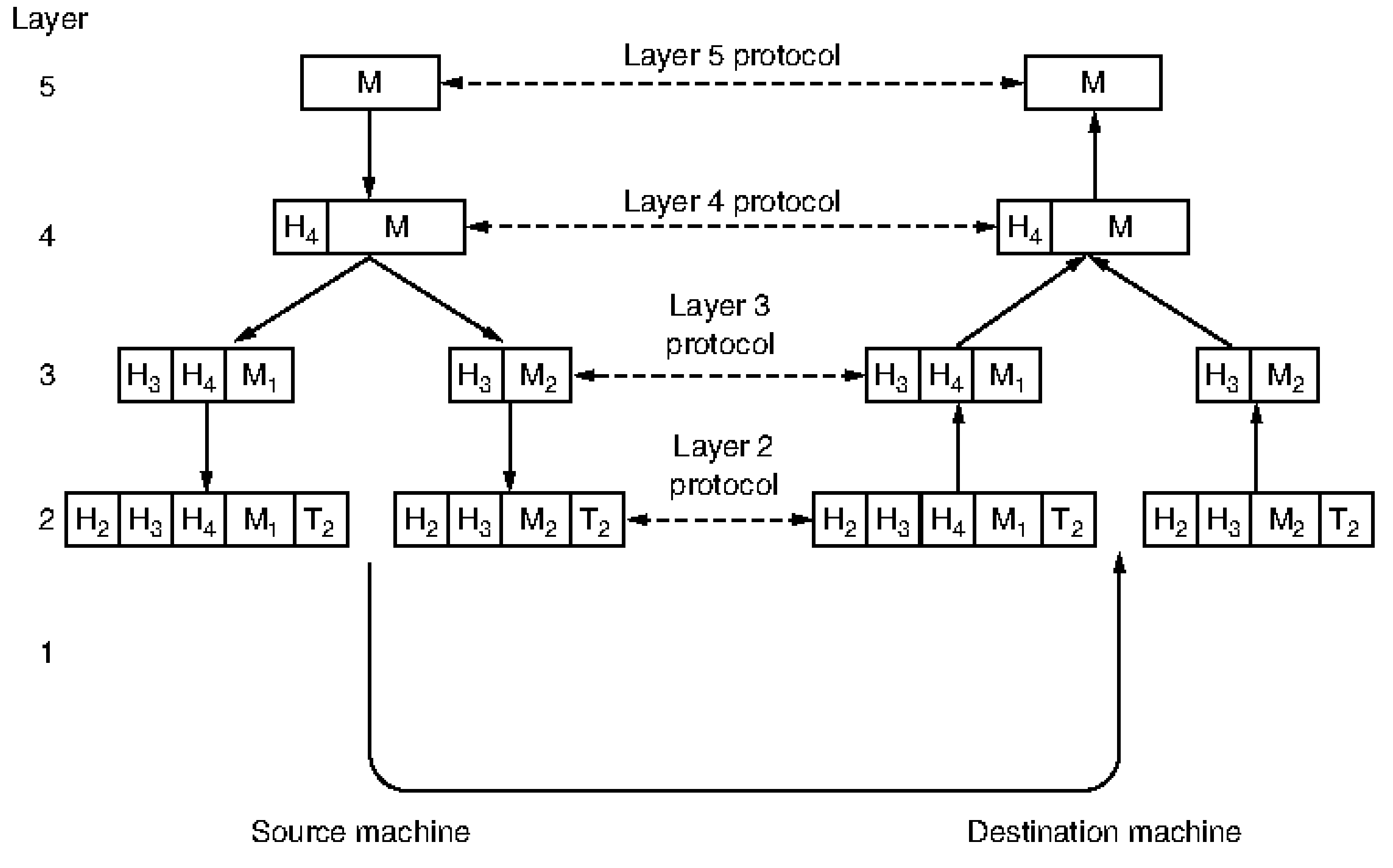


Figure 1-15. Example information flow supporting virtual communication in layer 5.

The peer process abstraction is crucial to all network design. Using it, the unmanageable task of designing the complete network can be broken into several smaller, manageable design problems, namely, the design of the individual layers.

Although Sec. 1.3 is called "Network Software," it is worth pointing out that the lower layers of a protocol hierarchy are frequently implemented in hardware or firmware. Nevertheless, complex protocol algorithms are involved, even if they are embedded (in whole or in part) in hardware.

1.3.2 Design Issues for the Layers

Some of the key design issues that occur in computer networks will come up in layer after layer. Below, we will briefly mention the more important ones.

Reliability is the design issue of making a network that operates correctly even though it is made up of a collection of components that are themselves unreliable. Think about the bits of a packet traveling through the network. There is a chance that some of these bits will be received damaged (inverted) due to fluke electrical noise, random wireless signals, hardware flaws, software bugs and so on. How is it possible that we find and fix these errors?

One mechanism for finding errors in received information uses codes for **error detection**. Information that is incorrectly received can then be retransmitted until it is received correctly. More powerful codes allow for **error correction**, where the correct message is recovered from the possibly incorrect bits that were originally received. Both of these mechanisms work by adding redundant information. They are used at low layers, to protect packets sent over individual links, and high layers, to check that the right contents were received.

Another reliability issue is finding a working path through a network. Often there are multiple paths between a source and destination, and in a large network, there may be some links or routers that are broken. Suppose that the network is down in Germany. Packets sent from London to Rome via Germany will not get through, but we could instead send packets from London to Rome via Paris. The network should automatically make this decision. This topic is called **routing**.

A second design issue concerns the evolution of the network. Over time, networks grow larger and new designs emerge that need to be connected to the existing network. We have recently seen the key structuring mechanism used to support change by dividing the overall problem and hiding implementation details: **protocol layering**. There are many other strategies as well.

Since there are many computers on the network, every layer needs a mechanism for identifying the senders and receivers that are involved in a particular message. This mechanism is called **addressing** or **naming**, in the low and high layers, respectively.

An aspect of growth is that different network technologies often have different limitations. For example, not all communication channels preserve the order of messages sent on them, leading to solutions that number messages. Another example is differences in the maximum size of a message that the networks can transmit. This leads to mechanisms for disassembling, transmitting, and then reassembling messages. This overall topic is called **internetworking**.

When networks get large, new problems arise. Cities can have traffic jams, a shortage of telephone numbers, and it is easy to get lost. Not many people have these problems in their own neighborhood, but citywide they may be a big issue. Designs that continue to work well when the network gets large are said to be **scalable**.

A third design issue is resource allocation. Networks provide a service to hosts from their underlying resources, such as the capacity of transmission lines. To do this well, they need mechanisms that divide their resources so that one host does not interfere with another too much.

Many designs share network bandwidth dynamically, according to the short-term needs of hosts, rather than by giving each host a fixed fraction of the bandwidth that it may or may not use. This design is called **statistical multiplexing**, meaning sharing based on the statistics of demand. It can be applied at low layers for a single link, or at high layers for a network or even applications that use the network.

An allocation problem that occurs at every level is how to keep a fast sender from swamping a slow receiver with data. Feedback from the receiver to the sender is often used. This subject is called **flow control**. Sometimes the problem is that the network is oversubscribed because too many computers want to send too much traffic, and the network cannot deliver it all. This overloading of the network is called **congestion**. One strategy is for each computer to reduce its demand when it experiences congestion. It, too, can be used in all layers.

It is interesting to observe that the network has more resources to offer than simply bandwidth. For uses such as carrying live video, the timeliness of delivery matters a great deal. Most networks must provide service to applications that want this **real-time** delivery at the same time that they provide service to applications that want high throughput. **Quality of service** is the name given to mechanisms that reconcile these competing demands.

The last major design issue is to secure the network by defending it against different kinds of threats. One of the threats we have mentioned previously is that of eavesdropping on communications. Mechanisms that provide **confidentiality** defend against this threat, and they are used in multiple layers. Mechanisms for **authentication** prevent someone from impersonating someone else. They might be used to tell fake banking Web sites from the real one, or to let the cellular network check that a call is really coming from your phone so that you will pay the bill. Other mechanisms for **integrity** prevent surreptitious changes to messages, such as altering "debit my account $10" to "debit my account $1000." All of these designs are based on cryptography, which we shall study in Chap. 8.

1.3.3 Connection-Oriented Versus Connectionless Service

Layers can offer two different types of service to the layers above them: connection-oriented and connectionless. In this section we will look at these two types and examine the differences between them.

Connection-oriented service is modeled after the telephone system. To talk to someone, you pick up the phone, dial the number, talk, and then hang up. Similarly, to use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at one end, and the receiver takes them out at the other end. In most cases the order is preserved so that the bits arrive in the order they were sent.

In some cases when a connection is established, the sender, receiver, and subnet conduct a **negotiation** about the parameters to be used, such as maximum message size, quality of service required, and other issues. Typically, one side makes a proposal and the other side can accept it, reject it, or make a counter-proposal. A **circuit** is another name for a connection with associated resources, such as a fixed bandwidth. This dates from the telephone network in which a circuit was a path over copper wire that carried a phone conversation.

In contrast to connection-oriented service, **connectionless** service is modeled after the postal system. Each message (letter) carries the full destination address, and each one is routed through the intermediate nodes inside the system independent of all the subsequent messages. There are different names for messages in different contexts; a **packet** is a message at the network layer. When the intermediate nodes receive a message in full before sending it on to the next node, this is called **store-and-forward switching**. The alternative, in which the onward transmission of a message at a node starts before it is completely received by the node, is called **cut-through switching**. Normally, when two messages are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first.

Each kind of service can further be characterized by its reliability. Some services are reliable in the sense that they never lose data. Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message so the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable.

A typical situation in which a reliable connection-oriented service is appropriate is file transfer. The owner of the file wants to be sure that all the bits arrive correctly and in the same order they were sent. Very few file transfer customers would prefer a service that occasionally scrambles or loses a few bits, even if it is much faster.

Reliable connection-oriented service has two minor variations: message sequences and byte streams. In the former variant, the message boundaries are preserved. When two 1024-byte messages are sent, they arrive as two distinct 1024-byte messages, never as one 2048-byte message. In the latter, the connection is simply a stream of bytes, with no message boundaries. When 2048 bytes arrive at the receiver, there is no way to tell if they were sent as one 2048-byte message, two 1024-byte messages, or 2048 1-byte messages. If the pages of a book are sent over a network to a phototypesetter as separate messages, it might be important to preserve the message boundaries. On the other hand, to download a DVD movie, a byte stream from the server to the user's computer is all that is needed. Message boundaries within the movie are not relevant.

For some applications, the transit delays introduced by acknowledgements are unacceptable. One such application is digitized voice traffic for **voice over IP**. It is less disruptive for telephone users to hear a bit of noise on the line from time to time than to experience a delay waiting for acknowledgements. Similarly, when transmitting a video conference, having a few pixels wrong is no problem, but having the image jerk along as the flow stops and starts to correct errors is irritating.

Not all applications require connections. For example, spammers send electronic junk-mail to many recipients. The spammer probably does not want to go to the trouble of setting up and later tearing down a connection to a recipient just to send them one item. Nor is 100 percent reliable delivery essential, especially if it costs more. All that is needed is a way to send a single message that has a high probability of arrival, but no guarantee. Unreliable (meaning not acknowledged) connectionless service is often called **datagram** service, in analogy with telegram service, which also does not return an acknowledgement to the sender. Despite it being unreliable, it is the dominant form in most networks for reasons that will become clear later

In other situations, the convenience of not having to establish a connection to send one message is desired, but reliability is essential. The **acknowledged datagram** service can be provided for these applications. It is like sending a registered letter and requesting a return receipt. When the receipt comes back, the sender is absolutely sure that the letter was delivered to the intended party and not lost along the way. Text messaging on mobile phones is an example.

Still another service is the **request-reply** service. In this service the sender transmits a single datagram containing a request; the reply contains the answer. Request-reply is commonly used to implement communication in the client-server model: the client issues a request and the server responds to it. For example, a mobile phone client might send a query to a map server to retrieve the map data for the current location. Figure 1-16 summarizes the types of services discussed above.

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Movie download
	Unreliable connection	Voice over IP
Connection-less	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Text messaging
	Request-reply	Database query

Figure 1-16. Six different types of service.

The concept of using unreliable communication may be confusing at first. After all, why would anyone actually prefer unreliable communication to reliable communication? First of all, reliable communication (in our sense, that is, acknowledged) may not be available in a given layer. For example, Ethernet does not provide reliable communication. Packets can occasionally be damaged in transit. It is up to higher protocol levels to recover from this problem. In particular, many reliable services are built on top of an unreliable datagram service. Second, the delays inherent in providing a reliable service may be unacceptable, especially in real-time applications such as multimedia. For these reasons, both reliable and unreliable communication coexist.

1.3.4 Service Primitives

A service is formally specified by a set of **primitives** (operations) available to user processes to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets.

The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connectionless service. As a minimal example of the service primitives that might provide a reliable byte stream, consider the primitives listed in Fig. 1-17. They will be familiar to fans of the Berkeley socket interface, as the primitives are a simplified version of that interface.

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Figure 1-17. Six service primitives that provide a simple connection-oriented service.

These primitives might be used for a request-reply interaction in a client-server environment. To illustrate how, We sketch a simple protocol that implements the service using acknowledged datagrams.

First, the server executes LISTEN to indicate that it is prepared to accept incoming connections. A common way to implement LISTEN is to make it a blocking system call. After executing the primitive, the server process is blocked until a request for connection appears.

Next, the client process executes CONNECT to establish a connection with the server. The CONNECT call needs to specify who to connect to, so it might have a parameter giving the server's address. The operating system then typically sends a packet to the peer asking it to connect, as shown by (1) in Fig. 1-18. The client process is suspended until there is a response.

When the packet arrives at the server, the operating system sees that the packet is requesting a connection. It checks to see if there is a listener, and if so it unblocks the listener. The server process can then establish the connection with the ACCEPT call. This sends a response (2) back to the client process to accept the

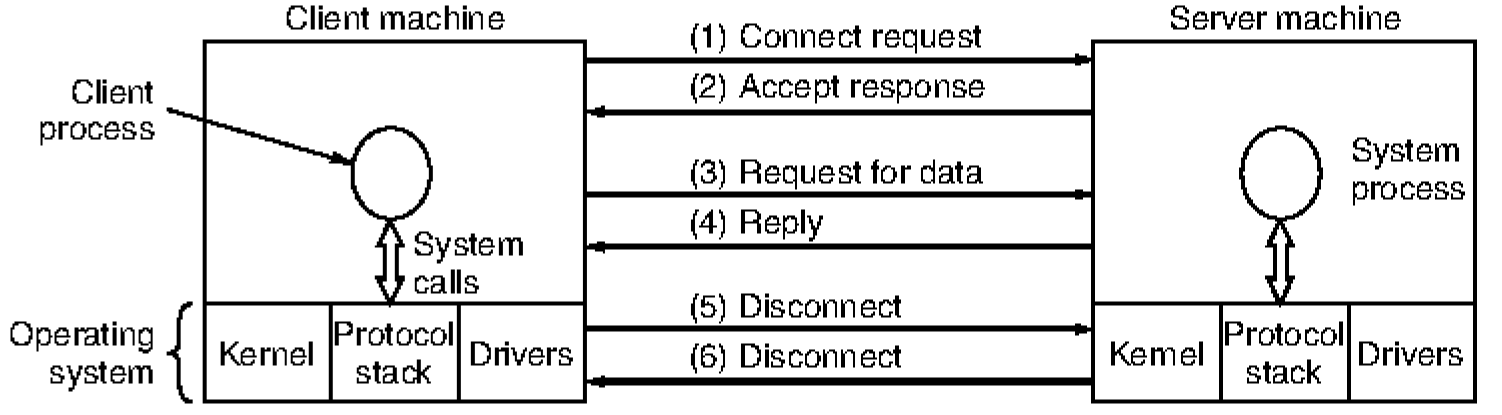


Figure 1-18. A simple client-server interaction using acknowledged datagrams.

connection. The arrival of this response then releases the client. At this point the client and server are both running and they have a connection established.

The obvious analogy between this protocol and real life is a customer (client) calling a company's customer service manager. At the start of the day, the service manager sits next to his telephone in case it rings. Later, a client places a call. When the manager picks up the phone, the connection is established.

The next step is for the server to execute RECEIVE to prepare to accept the first request. Normally, the server does this immediately upon being released from the LISTEN, before the acknowledgement can get back to the client. The RECEIVE call blocks the server.

Then the client executes SEND to transmit its request (3) followed by the execution of RECEIVE to get the reply. The arrival of the request packet at the server machine unblocks the server so it can handle the request. After it has done the work, the server uses SEND to return the answer to the client (4). The arrival of this packet unblocks the client, which can now inspect the answer. If the client has additional requests, it can make them now.

When the client is done, it executes DISCONNECT to terminate the connection (5). Usually, an initial DISCONNECT is a blocking call, suspending the client and sending a packet to the server saying that the connection is no longer needed. When the server gets the packet, it also issues a DISCONNECT of its own, acknowledging the client and releasing the connection (6). When the server's packet gets back to the client machine, the client process is released and the connection is broken. In a nutshell, this is how connection-oriented communication works.

Of course, life is not so simple. Many things can go wrong here. The timing can be wrong (e.g., the CONNECT is done before the LISTEN), packets can get lost, and much more. We will look at these issues in great detail later, but for the moment, Fig. 1-18 briefly summarizes how client-server communication might work with acknowledged datagrams so that we can ignore lost packets.

Given that six packets are required to complete this protocol, one might wonder why a connectionless protocol is not used instead. The answer is that in a perfect world it could be, in which case only two packets would be needed: one for the request and one for the reply. However, in the face of large messages in either direction (e.g., a megabyte file), transmission errors, and lost packets, the situation changes. If the reply consisted of hundreds of packets, some of which could be lost during transmission, how would the client know if some pieces were missing? How would the client know whether the last packet actually received was really the last packet sent? Suppose the client wanted a second file. How could it tell packet 1 from the second file from a lost packet 1 from the first file that suddenly found its way to the client? In short, in the real world, a simple request-reply protocol over an unreliable network is often inadequate. In Chap. 3 we will study a variety of protocols in detail that overcome these and other problems. For the moment, suffice it to say that having a reliable, ordered byte stream between processes is sometimes very convenient.

1.3.5 The Relationship of Services to Protocols

Services and protocols are distinct concepts. This distinction is so important that we emphasize it again here. A *service* is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

A *protocol*, in contrast, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols to implement their service definitions. They are free to change their protocols at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled. This is a key concept that any network designer should understand well.

To repeat this crucial point, services relate to the interfaces between layers, as illustrated in Fig. 1-19. In contrast, protocols relate to the packets sent between peer entities on different machines. It is very important not to confuse the two concepts.

An analogy with programming languages is worth making. A service is like an abstract data type or an object in an object-oriented language. It defines operations that can be performed on an object but does not specify how these operations are implemented. In contrast, a protocol relates to the *implementation* of the service and as such is not visible to the user of the service.

Many older protocols did not distinguish the service from the protocol. In effect, a typical layer might have had a service primitive SEND PACKET with the user providing a pointer to a fully assembled packet. This arrangement meant that all changes to the protocol were immediately visible to the users. Most network designers now regard such a design as a serious blunder.

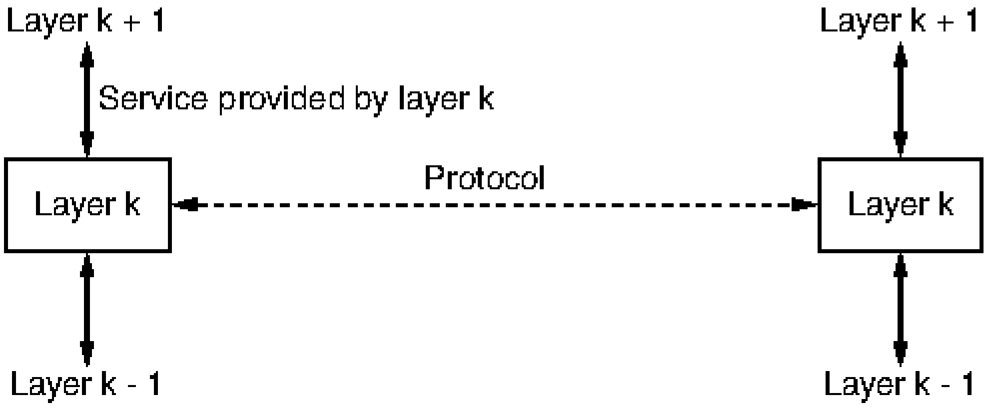


Figure 1-19. The relationship between a service and a protocol.

1.4 REFERENCE MODELS

Now that we have discussed layered networks in the abstract, it is time to look at some examples. We will discuss two important network architectures: the OSI reference model and the TCP/IP reference model. Although the *protocols* associated with the OSI model are not used any more, the *model* itself is actually quite general and still valid, and the features discussed at each layer are still very important. The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used. For this reason we will look at both of them in detail. Also, sometimes you can learn more from failures than from successes.

1.4.1 The OSI Reference Model

The OSI model (minus the physical medium) is shown in Fig. 1-20. This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995). The model is called the ISO **OSI** (**Open Systems Interconnection**) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will just call it the **OSI model** for short.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.

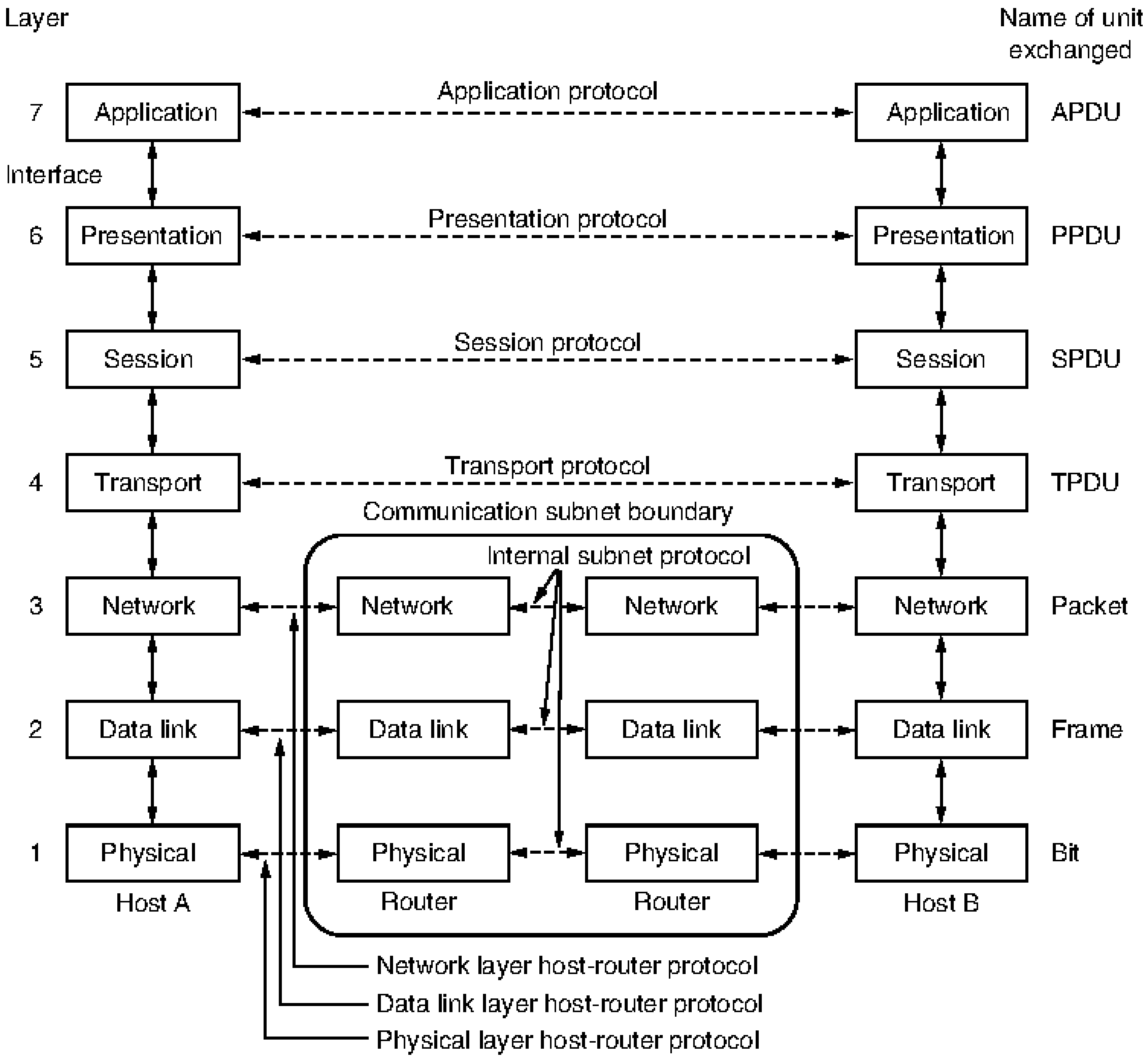


Figure 1-20. The OSI reference model.

4. The layer boundaries should be chosen to minimize the information flow across the interfaces.

5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

Below we will discuss each layer of the model in turn, starting at the bottom layer. Note that the OSI model itself is not a network architecture because it does not specify the exact services and protocols to be used in each layer. It just tells what each layer should do. However, ISO has also produced standards for all the layers, although these are not part of the reference model itself. Each one has been published as a separate international standard. The *model* (in part) is widely used although the associated protocols have been long forgotten.

The Physical Layer

The **physical layer** is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are what electrical signals should be used to represent a 1 and a 0, how many nanoseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established, how it is torn down when both sides are finished, how many pins the network connector has, and what each pin is used for. These design issues largely deal with mechanical, electrical, and timing interfaces, as well as the physical transmission medium, which lies below the physical layer.

The Data Link Layer

The main task of the **data link layer** is to transform a raw transmission facility into a line that appears free of undetected transmission errors. It does so by masking the real errors so the network layer does not see them. It accomplishes this task by having the sender break up the input data into **data frames** (typically a few hundred or a few thousand bytes) and transmit the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an **acknowledgement frame**.

Another issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism may be needed to let the transmitter know when the receiver can accept more data.

Broadcast networks have an additional issue in the data link layer: how to control access to the shared channel. A special sublayer of the data link layer, the **medium access control** sublayer, deals with this problem.

The Network Layer

The **network layer** controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into" the network and rarely changed, or more often they can be updated automatically to avoid failed components. They can also be determined at the start of each conversation, for example, a terminal session, such as a login to a remote machine. Finally, they can be highly dynamic, being determined anew for each packet to reflect the current network load.

If too many packets are present in the subnet at the same time, they will get in one another's way, forming bottlenecks. Handling congestion is also a responsibility of the network layer, in conjunction with higher layers that adapt the load they place on the network. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from that used by the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected.

In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

The Transport Layer

The basic function of the **transport layer** is to accept data from above it, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology over the course of time.

The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service exist, such as the transporting of isolated messages with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established. (As an aside, an error-free channel is completely impossible to achieve; what people really mean by this term is that the error rate is low enough to ignore in practice.)

The transport layer is a true end-to-end layer; it carries data all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, each protocols is between a machine and its immediate neighbors, and not between the ultimate source and destination machines, which may be separated by many routers. The difference between layers 1 through 3, which are chained, and layers 4 through 7, which are end-to-end, is illustrated in Fig. 1-20.

The Session Layer

The session layer allows users on different machines to establish **sessions** between them. Sessions offer various services, including **dialog control** (keeping track of whose turn it is to transmit), **token management** (preventing two parties from attempting the same critical operation simultaneously), and **synchronization** (checkpointing long transmissions to allow them to pick up from where they left off in the event of a crash and subsequent recovery).

The Presentation Layer

Unlike the lower layers, which are mostly concerned with moving bits around, the **presentation layer** is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different internal data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used "on the wire." The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records) to be defined and exchanged.

The Application Layer

The **application layer** contains a variety of protocols that are commonly needed by users. One widely used application protocol is **HTTP** (**HyperText Transfer Protocol**), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server hosting the page using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

1.4.2 The TCP/IP Reference Model

Let us now turn from the OSI reference model to the reference model used in the grandparent of all wide area computer networks, the ARPANET, and its successor, the worldwide Internet. Although we will give a brief history of the ARPANET later, it is useful to mention a few key aspects of it now. The ARPANET was a research network sponsored by the DoD (U.S. Department of Defense). It eventually connected hundreds of universities and government installations, using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so a new reference architecture was needed. Thus, from nearly the beginning, the ability to connect multiple networks in a seamless way was one of the major design goals. This architecture later became known as the **TCP/IP Reference Model**, after its two primary protocols. It was first described by Cerf and Kahn (1974), and later refined and defined as a standard in the Internet community (Braden, 1989). The design philosophy behind the model is discussed by Clark (1988).

Given the DoD's worry that some of its precious hosts, routers, and internetwork gateways might get blown to pieces at a moment's notice by an attack from the Soviet Union, another major goal was that the network be able to survive loss of subnet hardware, without existing conversations being broken off. In other words, the DoD wanted connections to remain intact as long as the source and destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation. Furthermore, since applications with divergent requirements were envisioned, ranging from transferring files to real-time speech transmission, a flexible architecture was needed.

The Link Layer

All these requirements led to the choice of a packet-switching network based on a connectionless layer that runs across different networks. The lowest layer in the model, the **link layer** describes what links such as serial lines and classic Ethernet must do to meet the needs of this connectionless internet layer. It is not really a layer at all, in the normal sense of the term, but rather an interface between hosts and transmission links. Early material on the TCP/IP model has little to say about it.

The Internet Layer

The **internet layer** is the linchpin that holds the whole architecture together. It is shown in Fig. 1-21 as corresponding roughly to the OSI network layer. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even arrive in a completely different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet.

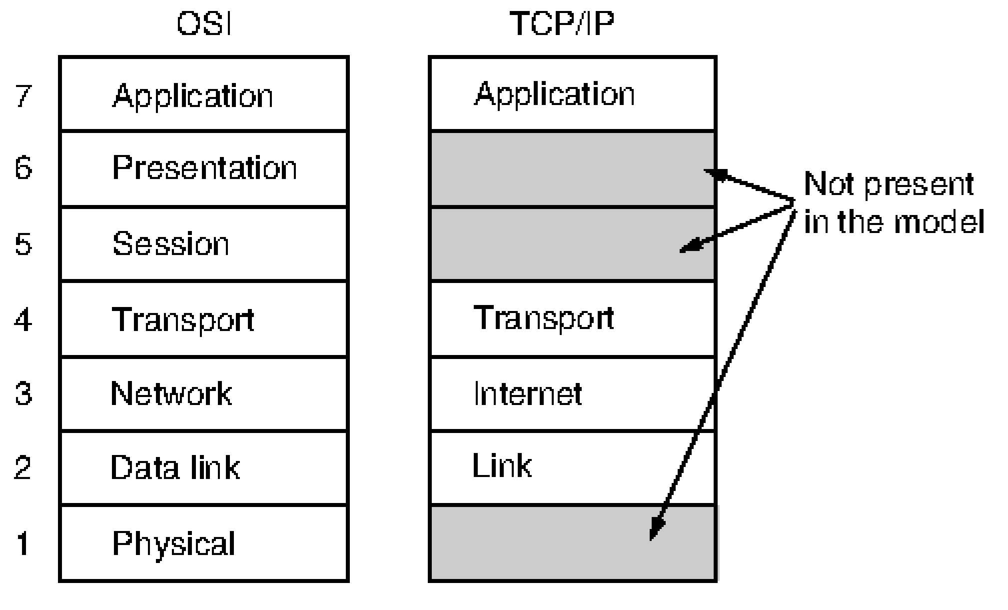


Figure 1-21. The TCP/IP reference model.

The analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mailbox in one country, and with a little luck, most of them will be delivered to the correct address in the destination country. The letters will probably travel through one or more international mail gateways along the way, but this is transparent to the users. Furthermore, that each country (i.e., each network) has its own stamps, preferred envelope sizes, and delivery rules is hidden from the users.

The internet layer defines an official packet format and protocol called **IP** (**Internet Protocol**), plus a companion protocol called **ICMP** (**Internet Control Message Protocol**) that helps it function. The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly a major issue here, as is congestion (though IP has not proven effective at avoiding congestion).

The Transport Layer

The layer above the internet layer in the TCP/IP model is now usually called the **transport layer**. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here. The first one, **TCP** (**Transmission Control Protocol**), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It segments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

The second protocol in this layer, **UDP** (**User Datagram Protocol**), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Fig. 1-22. Since the model was developed, IP has been implemented on many other networks.

The Application Layer

The TCP/IP model does not have session or presentation layers. No need for them was perceived. Instead, applications simply include any session and presentation functions that they require. Experience with the OSI model has proven this view correct: these layers are of little use to most applications.

On top of the transport layer is the **application layer**. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP). Many other protocols have been added to these over the years. Some important ones that we will study, shown in Fig. 1-22,

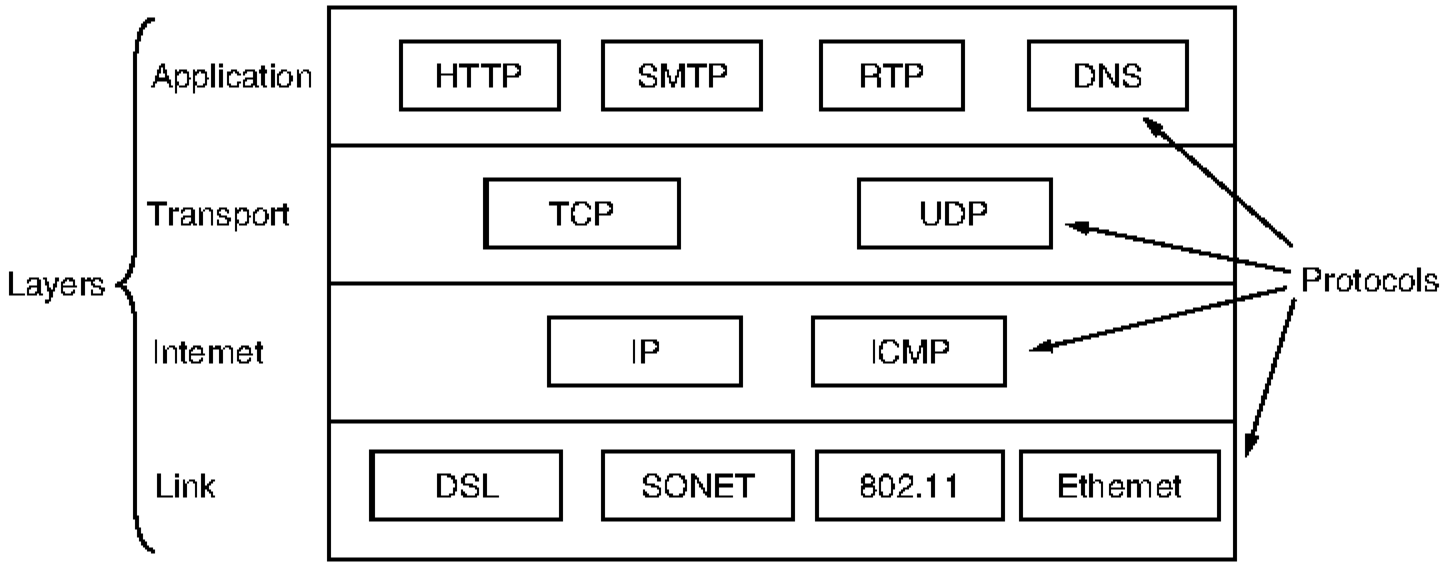


Figure 1-22. The TCP/IP model with some protocols we will study.

include the Domain Name System (DNS), for mapping host names onto their network addresses, HTTP, the protocol for fetching pages on the World Wide Web, and RTP, the protocol for delivering real-time media such as voice or movies.

1.4.3 The Model Used in This Book

As mentioned earlier, the strength of the OSI reference model is the *model* itself (minus the presentation and session layers), which has proven to be exceptionally useful for discussing computer networks. In contrast, the strength of the TCP/IP reference model is the *protocols*, which have been widely used for many years. Since computer scientists like to have their cake and eat it, too, we will use the hybrid model of Fig. 1-23 as the framework for this book.

5	Application
4	Transport
3	Network
2	Link
1	Physical

Figure 1-23. The reference model used in this book.

This model has five layers, running from the physical layer up through the link, network and transport layers to the application layer. The physical layer specifies how to transmit bits across different kinds of media as electrical (or other analog) signals. The link layer is concerned with how to send finite-length messages between directly connected computers with specified levels of reliability. Ethernet and 802.11 are examples of link layer protocols.

The network layer deals with how to combine multiple links into networks, and networks of networks, into internetworks so that we can send packets between distant computers. This includes the task of finding the path along which to send the packets. IP is the main example protocol we will study for this layer. The transport layer strengthens the delivery guarantees of the Network layer, usually with increased reliability, and provide delivery abstractions, such as a reliable byte stream, that match the needs of different applications. TCP is an important example of a transport layer protocol.

Finally, the application layer contains programs that make use of the network. Many, but not all, networked applications have user interfaces, such as a Web browser. Our concern, however, is with the portion of the program that uses the network. This is the HTTP protocol in the case of the Web browser. There are also important support programs in the application layer, such as the DNS, that are used by many applications.

Our chapter sequence is based on this model. In this way, we retain the value of the OSI model for understanding network architectures, but concentrate primarily on protocols that are important in practice, from TCP/IP and related protocols to newer ones such as 802.11, SONET, and Bluetooth.

1.4.4 A Comparison of the OSI and TCP/IP Reference Models

The OSI and TCP/IP reference models have much in common. Both are based on the concept of a stack of independent protocols. Also, the functionality of the layers is roughly similar. For example, in both models the layers up through and including the transport layer are there to provide an end-to-end, network-independent transport service to processes wishing to communicate. These layers form the transport provider. Again in both models, the layers above transport are application-oriented users of the transport service.

Despite these fundamental similarities, the two models also have many differences. In this section we will focus on the key differences between the two reference models. It is important to note that we are comparing the *reference models* here, not the corresponding *protocol stacks*. The protocols themselves will be discussed later. For an entire book comparing and contrasting TCP/IP and OSI, see Piscitello and Chapin (1993).

Three concepts are central to the OSI model:

1. Services.
2. Interfaces.
3. Protocols.

Probably the biggest contribution of the OSI model is that it makes the distinction between these three concepts explicit. Each layer performs some *services* for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics.

A layer's *interface* tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside.

Finally, the peer *protocols* used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers.

These ideas fit very nicely with modern ideas about object-oriented programming. An object, like a layer, has a set of methods (operations) that processes outside the object can invoke. The semantics of these methods define the set of services that the object offers. The methods' parameters and results form the object's interface. The code internal to the object is its protocol and is not visible or of any concern outside the object.

The TCP/IP model did not originally clearly distinguish between services, interfaces, and protocols, although people have tried to retrofit it after the fact to make it more OSI-like. For example, the only real services offered by the internet layer are SEND IP PACKET and RECEIVE IP PACKET. As a consequence, the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes. Being able to make such changes transparently is one of the main purposes of having layered protocols in the first place.

The OSI reference model was devised *before* the corresponding protocols were invented. This ordering meant that the model was not biased toward one particular set of protocols, a fact that made it quite general. The downside of this ordering was that the designers did not have much experience with the subject and did not have a good idea of which functionality to put in which layer.

For example, the data link layer originally dealt only with point-to-point networks. When broadcast networks came around, a new sublayer had to be hacked into the model. Furthermore, when people started to build real networks using the OSI model and existing protocols, it was discovered that these networks did not match the required service specifications (wonder of wonders), so convergence sublayers had to be grafted onto the model to provide a place for papering over the differences. Finally, the committee originally expected that each country would have one network, run by the government and using the OSI protocols, so no thought was given to internetworking. To make a long story short, things did not turn out that way.

With TCP/IP the reverse was true: the protocols came first, and the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. They fit perfectly. The only trouble was that the *model* did not fit any other protocol stacks. Consequently, it was not especially useful for describing other, non-TCP/IP networks.

Turning from philosophical matters to more specific ones, an obvious difference between the two models is the number of layers: the OSI model has seven layers and the TCP/IP model has four. Both have (inter)network, transport, and application layers, but the other layers are different.

Another difference is in the area of connectionless versus connection-oriented communication. The OSI model supports both connectionless and connection-oriented communication in the network layer, but only connection-oriented communication in the transport layer, where it counts (because the transport service is visible to the users). The TCP/IP model supports only one mode in the network layer (connectionless) but both in the transport layer, giving the users a choice. This choice is especially important for simple request-response protocols.

1.4.5 A Critique of the OSI Model and Protocols

Neither the OSI model and its protocols nor the TCP/IP model and its protocols are perfect. Quite a bit of criticism can be, and has been, directed at both of them. In this section and the next one, we will look at some of these criticisms. We will begin with OSI and examine TCP/IP afterward.

At the time the second edition of this book was published (1989), it appeared to many experts in the field that the OSI model and its protocols were going to take over the world and push everything else out of their way. This did not happen. Why? A look back at some of the reasons may be useful. They can be summarized as:

1. Bad timing.
2. Bad technology.
3. Bad implementations.
4. Bad politics.

Bad Timing

First let us look at reason one: bad timing. The time at which a standard is established is absolutely critical to its success. David Clark of M.I.T. has a theory of standards that he calls the *apocalypse of the two elephants*, which is illustrated in Fig. 1-24.

This figure shows the amount of activity surrounding a new subject. When the subject is first discovered, there is a burst of research activity in the form of discussions, papers, and meetings. After a while this activity subsides, corporations discover the subject, and the billion-dollar wave of investment hits.

It is essential that the standards be written in the trough in between the two "elephants." If they are written too early (before the research results are well

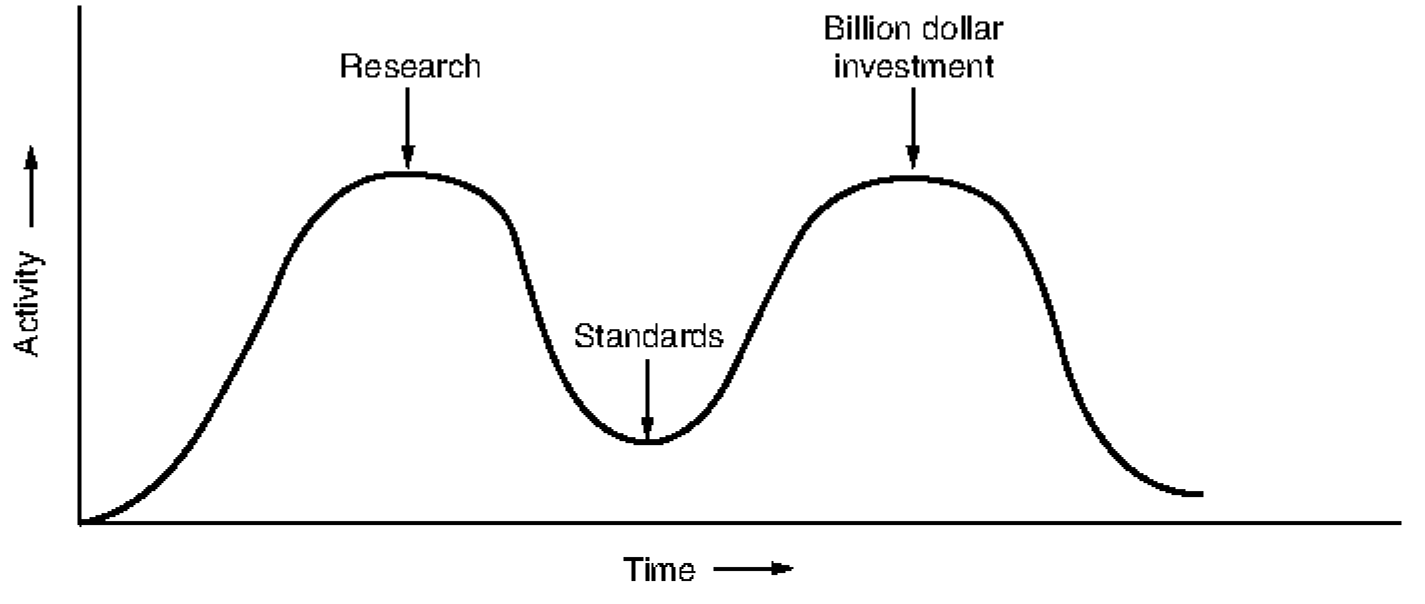


Figure 1-24. The apocalypse of the two elephants.

established), the subject may still be poorly understood; the result is a bad standard. If they are written too late, so many companies may have already made major investments in different ways of doing things that the standards are effectively ignored. If the interval between the two elephants is very short (because everyone is in a hurry to get started), the people developing the standards may get crushed.

It now appears that the standard OSI protocols got crushed. The competing TCP/IP protocols were already in widespread use by research universities by the time the OSI protocols appeared. While the billion-dollar wave of investment had not yet hit, the academic market was large enough that many vendors had begun cautiously offering TCP/IP products. When OSI came around, they did not want to support a second protocol stack until they were forced to, so there were no initial offerings. With every company waiting for every other company to go first, no company went first and OSI never happened.

Bad Technology

The second reason that OSI never caught on is that both the model and the protocols are flawed. The choice of seven layers was more political than technical, and two of the layers (session and presentation) are nearly empty, whereas two other ones (data link and network) are overfull.

The OSI model, along with its associated service definitions and protocols, is extraordinarily complex. When piled up, the printed standards occupy a significant fraction of a meter of paper. They are also difficult to implement and inefficient in operation. In this context, a riddle posed by Paul Mockapetris and cited by Rose (1993) comes to mind:

Q: What do you get when you cross a mobster with an international standard?
A: Someone who makes you an offer you can't understand.

In addition to being incomprehensible, another problem with OSI is that some functions, such as addressing, flow control, and error control, reappear again and again in each layer. Saltzer et al. (1984), for example, have pointed out that to be effective, error control must be done in the highest layer, so that repeating it over and over in each of the lower layers is often unnecessary and inefficient.

Bad Implementations

Given the enormous complexity of the model and the protocols, it will come as no surprise that the initial implementations were huge, unwieldy, and slow. Everyone who tried them got burned. It did not take long for people to associate "OSI" with "poor quality." Although the products improved in the course of time, the image stuck.

In contrast, one of the first implementations of TCP/IP was part of Berkeley UNIX and was quite good (not to mention, free). People began using it quickly, which led to a large user community, which led to improvements, which led to an even larger community. Here the spiral was upward instead of downward.

Bad Politics

On account of the initial implementation, many people, especially in academia, thought of TCP/IP as part of UNIX, and UNIX in the 1980s in academia was not unlike parenthood (then incorrectly called motherhood) and apple pie.

OSI, on the other hand, was widely thought to be the creature of the European telecommunication ministries, the European Community, and later the U.S. Government. This belief was only partly true, but the very idea of a bunch of government bureaucrats trying to shove a technically inferior standard down the throats of the poor researchers and programmers down in the trenches actually developing computer networks did not aid OSI's cause. Some people viewed this development in the same light as IBM announcing in the 1960s that PL/I was the language of the future, or the DoD correcting this later by announcing that it was actually Ada.

1.4.6 A Critique of the TCP/IP Reference Model

The TCP/IP model and protocols have their problems too. First, the model does not clearly distinguish the concepts of services, interfaces, and protocols. Good software engineering practice requires differentiating between the specification and the implementation, something that OSI does very carefully, but TCP/IP does not. Consequently, the TCP/IP model is not much of a guide for designing new networks using new technologies.

Second, the TCP/IP model is not at all general and is poorly suited to describing any protocol stack other than TCP/IP. Trying to use the TCP/IP model to describe Bluetooth, for example, is completely impossible.

Third, the link layer is not really a layer at all in the normal sense of the term as used in the context of layered protocols. It is an interface (between the network and data link layers). The distinction between an interface and a layer is crucial, and one should not be sloppy about it.

Fourth, the TCP/IP model does not distinguish between the physical and data link layers. These are completely different. The physical layer has to do with the transmission characteristics of copper wire, fiber optics, and wireless communication. The data link layer's job is to delimit the start and end of frames and get them from one side to the other with the desired degree of reliability. A proper model should include both as separate layers. The TCP/IP model does not do this.

Finally, although the IP and TCP protocols were carefully thought out and well implemented, many of the other protocols were ad hoc, generally produced by a couple of graduate students hacking away until they got tired. The protocol implementations were then distributed free, which resulted in their becoming widely used, deeply entrenched, and thus hard to replace. Some of them are a bit of an embarrassment now. The virtual terminal protocol, TELNET, for example, was designed for a ten-character-per-second mechanical Teletype terminal. It knows nothing of graphical user interfaces and mice. Nevertheless, it is still in use some 30 years later.

1.5 EXAMPLE NETWORKS

The subject of computer networking covers many different kinds of networks, large and small, well known and less well known. They have different goals, scales, and technologies. In the following sections, we will look at some examples, to get an idea of the variety one finds in the area of computer networking.

We will start with the Internet, probably the best known network, and look at its history, evolution, and technology. Then we will consider the mobile phone network. Technically, it is quite different from the Internet, contrasting nicely with it. Next we will introduce IEEE 802.11, the dominant standard for wireless LANs. Finally, we will look at RFID and sensor networks, technologies that extend the reach of the network to include the physical world and everyday objects.

1.5.1 The Internet

The Internet is not really a network at all, but a vast collection of different networks that use certain common protocols and provide certain common services. It is an unusual system in that it was not planned by anyone and is not controlled by anyone. To better understand it, let us start from the beginning and see how it has developed and why. For a wonderful history of the Internet, John Naughton's (2000) book is highly recommended. It is one of those rare books that is not only fun to read, but also has 20 pages of *ibid.*'s and *op. cit.*'s for the serious historian. Some of the material in this section is based on this book.

Of course, countless technical books have been written about the Internet and its protocols as well. For more information, see, for example, Maufer (1999).

The ARPANET

The story begins in the late 1950s. At the height of the Cold War, the U.S. DoD wanted a command-and-control network that could survive a nuclear war. At that time, all military communications used the public telephone network, which was considered vulnerable. The reason for this belief can be gleaned from Fig. 1-25(a). Here the black dots represent telephone switching offices, each of which was connected to thousands of telephones. These switching offices were, in turn, connected to higher-level switching offices (toll offices), to form a national hierarchy with only a small amount of redundancy. The vulnerability of the system was that the destruction of a few key toll offices could fragment it into many isolated islands.

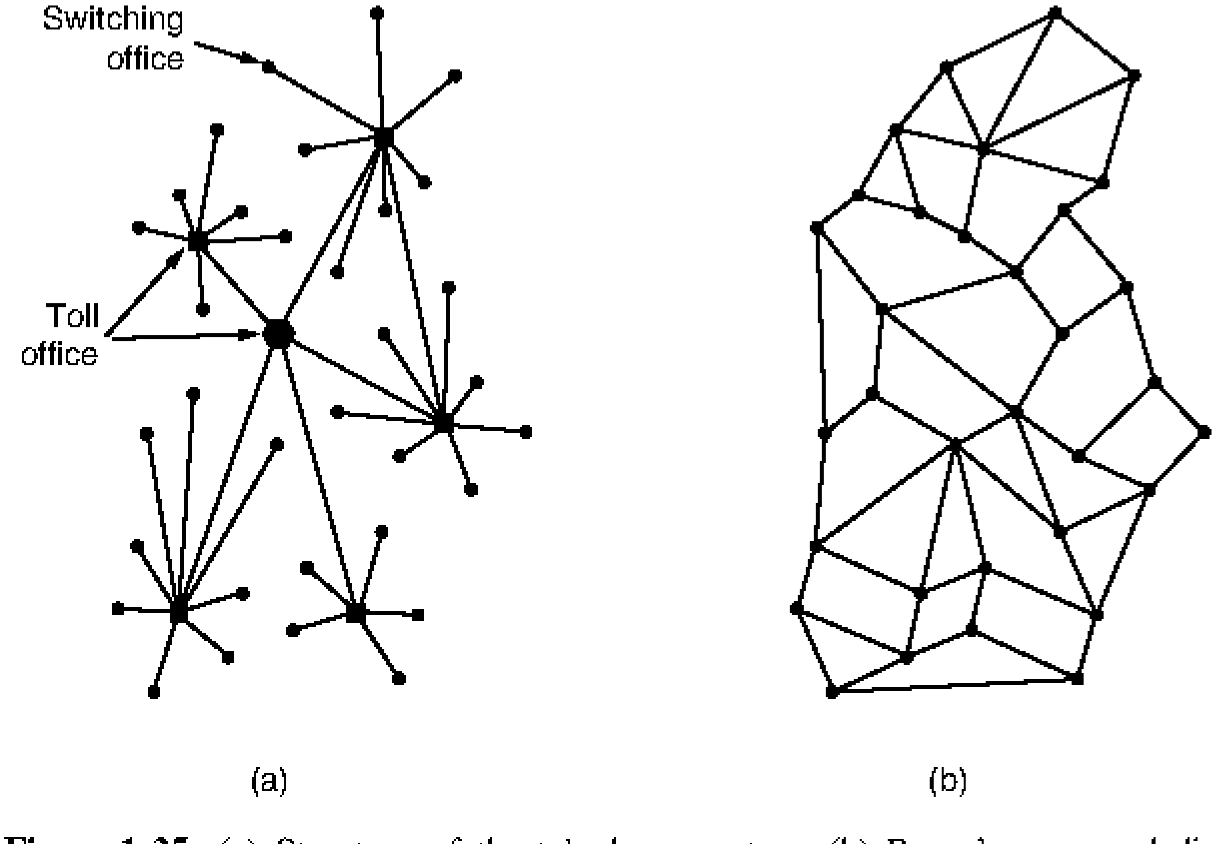


Figure 1-25. (a) Structure of the telephone system. (b) Baran's proposed distributed switching system.

Around 1960, the DoD awarded a contract to the RAND Corporation to find a solution. One of its employees, Paul Baran, came up with the highly distributed and fault-tolerant design of Fig. 1-25(b). Since the paths between any two switching offices were now much longer than analog signals could travel without distortion, Baran proposed using digital packet-switching technology. Baran wrote several reports for the DoD describing his ideas in detail (Baran, 1964). Officials at the Pentagon liked the concept and asked AT&T, then the U.S.' national telephone monopoly, to build a prototype. AT&T dismissed Baran's ideas out of hand. The biggest and richest corporation in the world was not about to allow some young whippersnapper tell it how to build a telephone system. They said Baran's network could not be built and the idea was killed.

Several years went by and still the DoD did not have a better command-and-control system. To understand what happened next, we have to go back all the way to October 1957, when the Soviet Union beat the U.S. into space with the launch of the first artificial satellite, Sputnik. When President Eisenhower tried to find out who was asleep at the switch, he was appalled to find the Army, Navy, and Air Force squabbling over the Pentagon's research budget. His immediate response was to create a single defense research organization, **ARPA**, the **Advanced Research Projects Agency**. ARPA had no scientists or laboratories; in fact, it had nothing more than an office and a small (by Pentagon standards) budget. It did its work by issuing grants and contracts to universities and companies whose ideas looked promising to it.

For the first few years, ARPA tried to figure out what its mission should be. In 1967, the attention of Larry Roberts, a program manager at ARPA who was trying to figure out how to provide remote access to computers, turned to networking. He contacted various experts to decide what to do. One of them, Wesley Clark, suggested building a packet-switched subnet, connecting each host to its own router.

After some initial skepticism, Roberts bought the idea and presented a somewhat vague paper about it at the ACM SIGOPS Symposium on Operating System Principles held in Gatlinburg, Tennessee in late 1967 (Roberts, 1967). Much to Roberts' surprise, another paper at the conference described a similar system that had not only been designed but actually fully implemented under the direction of Donald Davies at the National Physical Laboratory in England. The NPL system was not a national system (it just connected several computers on the NPL campus), but it demonstrated that packet switching could be made to work. Furthermore, it cited Baran's now discarded earlier work. Roberts came away from Gatlinburg determined to build what later became known as the **ARPANET**.

The subnet would consist of minicomputers called **IMPs** (**Interface Message Processors**) connected by 56-kbps transmission lines. For high reliability, each IMP would be connected to at least two other IMPs. The subnet was to be a datagram subnet, so if some lines and IMPs were destroyed, messages could be automatically rerouted along alternative paths.

Each node of the network was to consist of an IMP and a host, in the same room, connected by a short wire. A host could send messages of up to 8063 bits to its IMP, which would then break these up into packets of at most 1008 bits and forward them independently toward the destination. Each packet was received in its entirety before being forwarded, so the subnet was the first electronic store-and-forward packet-switching network.

ARPA then put out a tender for building the subnet. Twelve companies bid for it. After evaluating all the proposals, ARPA selected BBN, a consulting firm based in Cambridge, Massachusetts, and in December 1968 awarded it a contract to build the subnet and write the subnet software. BBN chose to use specially modified Honeywell DDP-316 minicomputers with 12K 16-bit words of core memory as the IMPs. The IMPs did not have disks, since moving parts were considered unreliable. The IMPs were interconnected by 56-kbps lines leased from telephone companies. Although 56 kbps is now the choice of teenagers who cannot afford DSL or cable, it was then the best money could buy.

The software was split into two parts: subnet and host. The subnet software consisted of the IMP end of the host-IMP connection, the IMP-IMP protocol, and a source IMP to destination IMP protocol designed to improve reliability. The original ARPANET design is shown in Fig. 1-26.

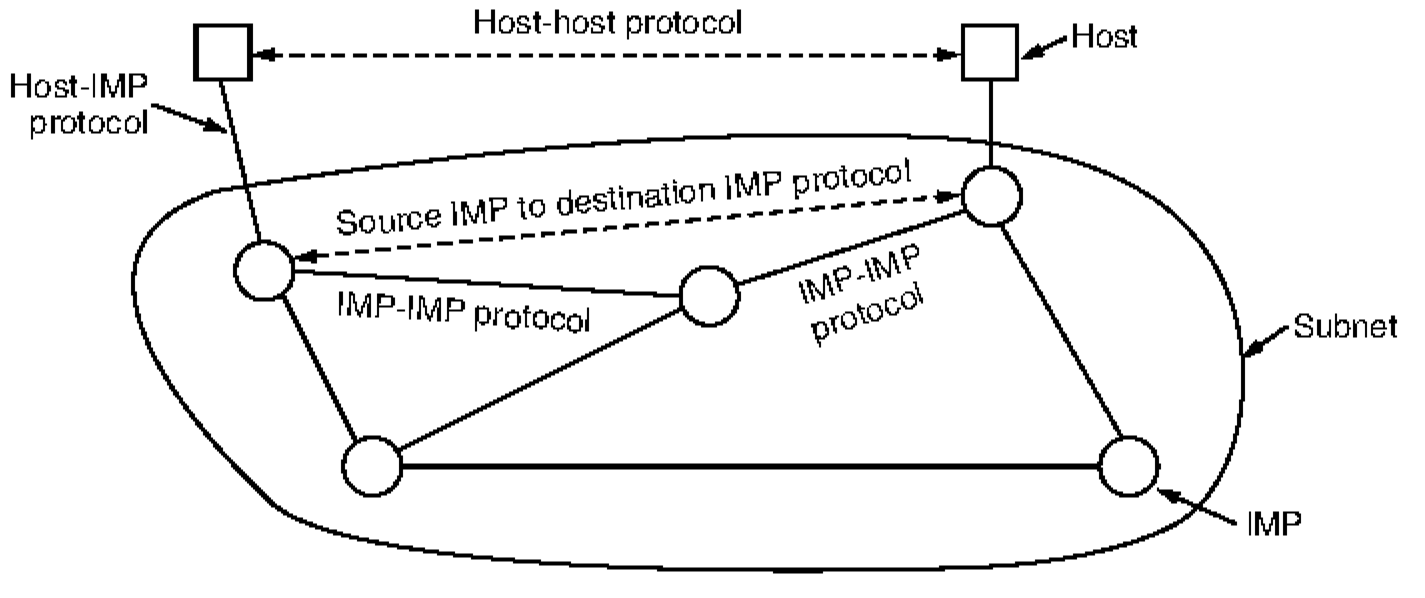


Figure 1-26. The original ARPANET design.

Outside the subnet, software was also needed, namely, the host end of the host-IMP connection, the host-host protocol, and the application software. It soon became clear that BBN was of the opinion that when it had accepted a message on a host-IMP wire and placed it on the host-IMP wire at the destination, its job was done.

Roberts had a problem, though: the hosts needed software too. To deal with it, he convened a meeting of network researchers, mostly graduate students, at Snowbird, Utah, in the summer of 1969. The graduate students expected some network expert to explain the grand design of the network and its software to them and then assign each of them the job of writing part of it. They were astounded when there was no network expert and no grand design. They had to figure out what to do on their own.

Nevertheless, somehow an experimental network went online in December 1969 with four nodes: at UCLA, UCSB, SRI, and the University of Utah. These four were chosen because all had a large number of ARPA contracts, and all had different and completely incompatible host computers (just to make it more fun). The first host-to-host message had been sent two months earlier from the UCLA node by a team led by Len Kleinrock (a pioneer of the theory of packet switching) to the SRI node. The network grew quickly as more IMPs were delivered and installed; it soon spanned the United States. Figure 1-27 shows how rapidly the ARPANET grew in the first 3 years.

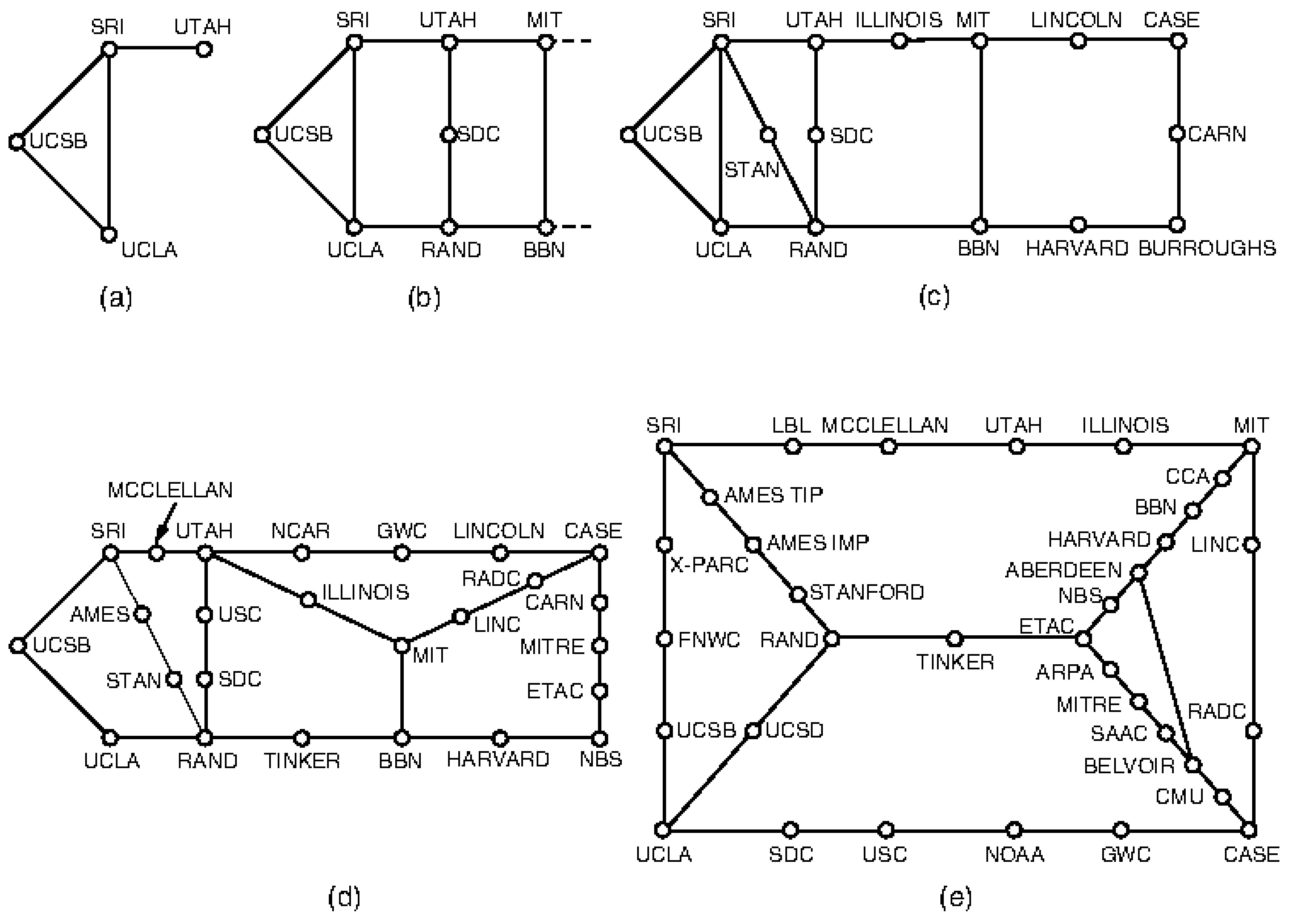


Figure 1-27. Growth of the ARPANET. (a) December 1969. (b) July 1970. (c) March 1971. (d) April 1972. (e) September 1972.

In addition to helping the fledgling ARPANET grow, ARPA also funded research on the use of satellite networks and mobile packet radio networks. In one now famous demonstration, a truck driving around in California used the packet radio network to send messages to SRI, which were then forwarded over the ARPANET to the East Coast, where they were shipped to University College in London over the satellite network. This allowed a researcher in the truck to use a computer in London while driving around in California.

This experiment also demonstrated that the existing ARPANET protocols were not suitable for running over different networks. This observation led to more research on protocols, culminating with the invention of the TCP/IP model and protocols (Cerf and Kahn, 1974). TCP/IP was specifically designed to handle communication over internetworks, something becoming increasingly important as more and more networks were hooked up to the ARPANET.

To encourage adoption of these new protocols, ARPA awarded several contracts to implement TCP/IP on different computer platforms, including IBM, DEC, and HP systems, as well as for Berkeley UNIX. Researchers at the University of California at Berkeley rewrote TCP/IP with a new programming interface called **sockets** for the upcoming 4.2BSD release of Berkeley UNIX. They also wrote many application, utility, and management programs to show how convenient it was to use the network with sockets.

The timing was perfect. Many universities had just acquired a second or third VAX computer and a LAN to connect them, but they had no networking software. When 4.2BSD came along, with TCP/IP, sockets, and many network utilities, the complete package was adopted immediately. Furthermore, with TCP/IP, it was easy for the LANs to connect to the ARPANET, and many did.

During the 1980s, additional networks, especially LANs, were connected to the ARPANET. As the scale increased, finding hosts became increasingly expensive, so **DNS** (**Domain Name System**) was created to organize machines into domains and map host names onto IP addresses. Since then, DNS has become a generalized, distributed database system for storing a variety of information related to naming. We will study it in detail in Chap. 7.

NSFNET

By the late 1970s, NSF (the U.S. National Science Foundation) saw the enormous impact the ARPANET was having on university research, allowing scientists across the country to share data and collaborate on research projects. However, to get on the ARPANET a university had to have a research contract with the DoD. Many did not have a contract. NSF's initial response was to fund the Computer Science Network (**CSNET**) in 1981. It connected computer science departments and industrial research labs to the ARPANET via dial-up and leased lines. In the late 1980s, the NSF went further and decided to design a successor to the ARPANET that would be open to all university research groups.

To have something concrete to start with, NSF decided to build a backbone network to connect its six supercomputer centers, in San Diego, Boulder, Champaign, Pittsburgh, Ithaca, and Princeton. Each supercomputer was given a little brother, consisting of an LSI-11 microcomputer called a **fuzzball**. The fuzzballs were connected with 56-kbps leased lines and formed the subnet, the same hardware technology the ARPANET used. The software technology was different however: the fuzzballs spoke TCP/IP right from the start, making it the first TCP/IP WAN.

NSF also funded some (eventually about 20) regional networks that connected to the backbone to allow users at thousands of universities, research labs, libraries, and museums to access any of the supercomputers and to communicate with one another. The complete network, including backbone and the regional networks, was called **NSFNET**. It connected to the ARPANET through a link between an

IMP and a fuzzball in the Carnegie-Mellon machine room. The first NSFNET backbone is illustrated in Fig. 1-28 superimposed on a map of the U.S.

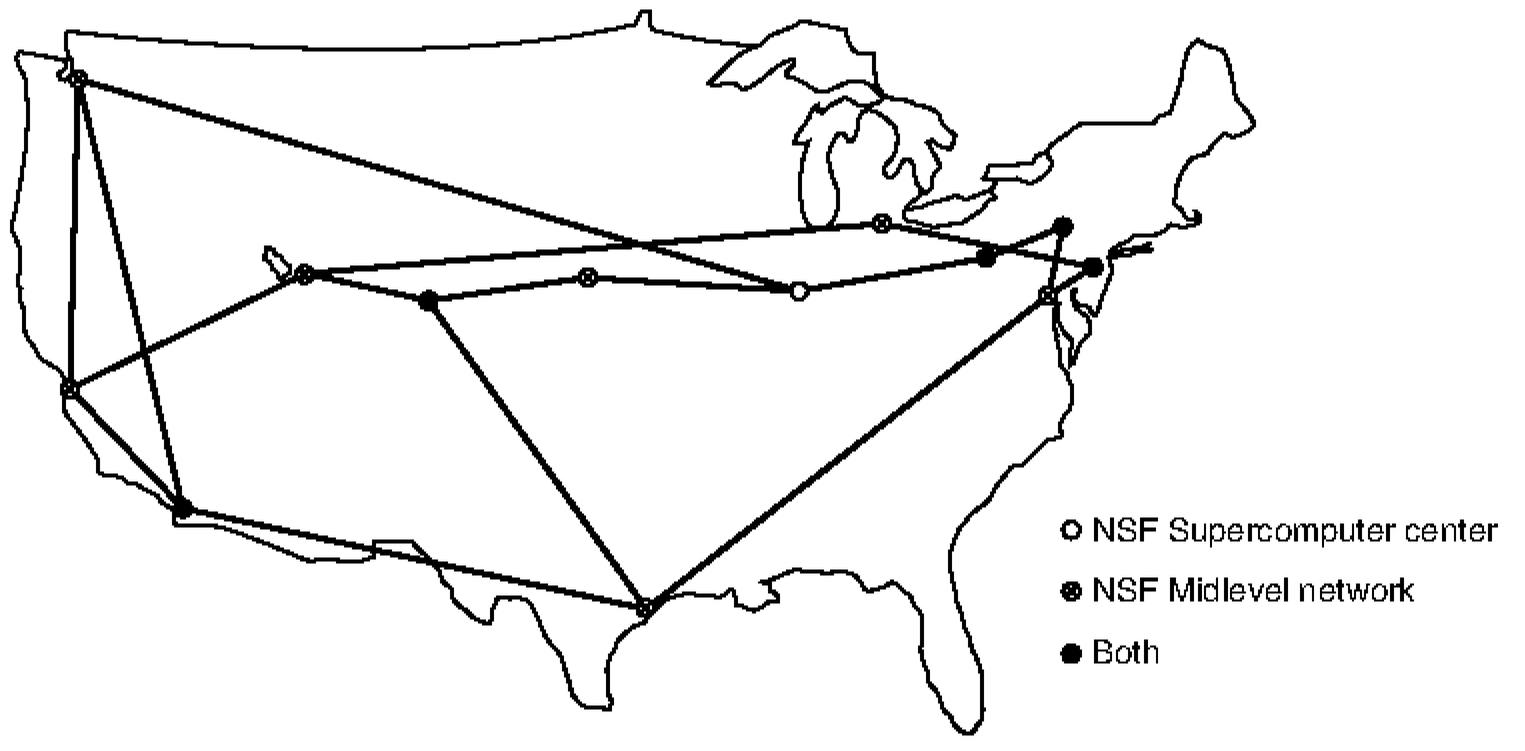


Figure 1-28. The NSFNET backbone in 1988.

NSFNET was an instantaneous success and was overloaded from the word go. NSF immediately began planning its successor and awarded a contract to the Michigan-based MERIT consortium to run it. Fiber optic channels at 448 kbps were leased from MCI (since merged with WorldCom) to provide the version 2 backbone. IBM PC-RTs were used as routers. This, too, was soon overwhelmed, and by 1990, the second backbone was upgraded to 1.5 Mbps.

As growth continued, NSF realized that the government could not continue financing networking forever. Furthermore, commercial organizations wanted to join but were forbidden by NSF's charter from using networks NSF paid for. Consequently, NSF encouraged MERIT, MCI, and IBM to form a nonprofit corporation, **ANS** (**Advanced Networks and Services**), as the first step along the road to commercialization. In 1990, ANS took over NSFNET and upgraded the 1.5-Mbps links to 45 Mbps to form **ANSNET**. This network operated for 5 years and was then sold to America Online. But by then, various companies were offering commercial IP service and it was clear the government should now get out of the networking business.

To ease the transition and make sure every regional network could communicate with every other regional network, NSF awarded contracts to four different network operators to establish a **NAP** (**Network Access Point**). These operators were PacBell (San Francisco), Ameritech (Chicago), MFS (Washington, D.C.), and Sprint (New York City, where for NAP purposes, Pennsauken, New Jersey counts as New York City). Every network operator that wanted to provide backbone service to the NSF regional networks had to connect to all the NAPs.

This arrangement meant that a packet originating on any regional network had a choice of backbone carriers to get from its NAP to the destination's NAP. Consequently, the backbone carriers were forced to compete for the regional networks' business on the basis of service and price, which was the idea, of course. As a result, the concept of a single default backbone was replaced by a commercially driven competitive infrastructure. Many people like to criticize the Federal Government for not being innovative, but in the area of networking, it was DoD and NSF that created the infrastructure that formed the basis for the Internet and then handed it over to industry to operate.

During the 1990s, many other countries and regions also built national research networks, often patterned on the ARPANET and NSFNET. These included EuropaNET and EBONE in Europe, which started out with 2-Mbps lines and then upgraded to 34-Mbps lines. Eventually, the network infrastructure in Europe was handed over to industry as well.

The Internet has changed a great deal since those early days. It exploded in size with the emergence of the World Wide Web (WWW) in the early 1990s. Recent data from the Internet Systems Consortium puts the number of visible Internet hosts at over 600 million. This guess is only a low-ball estimate, but it far exceeds the few million hosts that were around when the first conference on the WWW was held at CERN in 1994.

The way we use the Internet has also changed radically. Initially, applications such as email-for-academics, newsgroups, remote login, and file transfer dominated. Later it switched to email-for-everyman, then the Web and peer-to-peer content distribution, such as the now-shuttered Napster. Now real-time media distribution, social networks (e.g., Facebook), and microblogging (e.g., Twitter) are taking off. These switches brought richer kinds of media to the Internet and hence much more traffic. In fact, the dominant traffic on the Internet seems to change with some regularity as, for example, new and better ways to work with music or movies can become very popular very quickly.

Architecture of the Internet

The architecture of the Internet has also changed a great deal as it has grown explosively. In this section, we will attempt to give a brief overview of what it looks like today. The picture is complicated by continuous upheavals in the businesses of telephone companies (telcos), cable companies and ISPs that often make it hard to tell who is doing what. One driver of these upheavals is telecommunications convergence, in which one network is used for previously different uses. For example, in a "triple play" one company sells you telephony, TV, and Internet service over the same network connection on the assumption that this will save you money. Consequently, the description given here will be of necessity somewhat simpler than reality. And what is true today may not be true tomorrow.

The big picture is shown in Fig. 1-29. Let us examine this figure piece by piece, starting with a computer at home (at the edges of the figure). To join the Internet, the computer is connected to an **Internet Service Provider**, or simply **ISP**, from who the user purchases **Internet access** or **connectivity**. This lets the computer exchange packets with all of the other accessible hosts on the Internet. The user might send packets to surf the Web or for any of a thousand other uses, it does not matter. There are many kinds of Internet access, and they are usually distinguished by how much bandwidth they provide and how much they cost, but the most important attribute is connectivity.

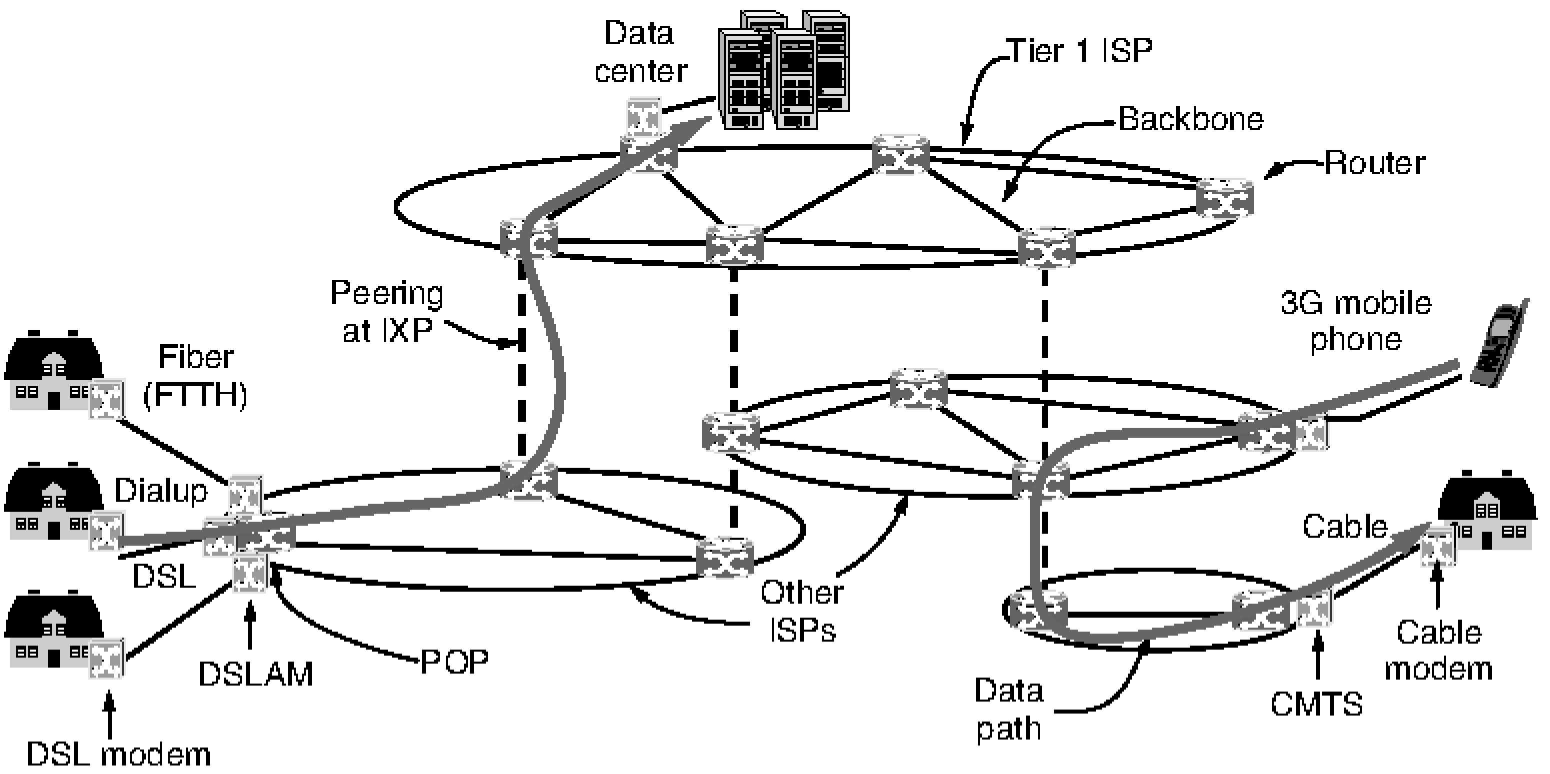


Figure 1-29. Overview of the Internet architecture.

A common way to connect to an ISP is to use the phone line to your house, in which case your phone company is your ISP. **DSL**, short for **Digital Subscriber Line**, reuses the telephone line that connects to your house for digital data transmission. The computer is connected to a device called a **DSL modem** that converts between digital packets and analog signals that can pass unhindered over the telephone line. At the other end, a device called a **DSLAM** (**Digital Subscriber Line Access Multiplexer**) converts between signals and packets.

Several other popular ways to connect to an ISP are shown in Fig. 1-29. DSL is a higher-bandwidth way to use the local telephone line than to send bits over a traditional telephone call instead of a voice conversation. That is called **dial-up** and done with a different kind of modem at both ends. The word **modem** is short for "*mo*dulator *dem*odulator" and refers to any device that converts between digital bits and analog signals.

Another method is to send signals over the cable TV system. Like DSL, this is a way to reuse existing infrastructure, in this case otherwise unused cable TV channels. The device at the home end is called a **cable modem** and the device at the **cable headend** is called the **CMTS** (**Cable Modem Termination System**).

DSL and cable provide Internet access at rates from a small fraction of a megabit/sec to multiple megabit/sec, depending on the system. These rates are much greater than dial-up rates, which are limited to 56 kbps because of the narrow bandwidth used for voice calls. Internet access at much greater than dial-up speeds is called **broadband**. The name refers to the broader bandwidth that is used for faster networks, rather than any particular speed.

The access methods mentioned so far are limited by the bandwidth of the "last mile" or last leg of transmission. By running optical fiber to residences, faster Internet access can be provided at rates on the order of 10 to 100 Mbps. This design is called **FTTH** (**Fiber to the Home**). For businesses in commercial areas, it may make sense to lease a high-speed transmission line from the offices to the nearest ISP. For example, in North America, a T3 line runs at roughly 45 Mbps.

Wireless is used for Internet access too. An example we will explore shortly is that of 3G mobile phone networks. They can provide data delivery at rates of 1 Mbps or higher to mobile phones and fixed subscribers in the coverage area.

We can now move packets between the home and the ISP. We call the location at which customer packets enter the ISP network for service the ISP's **POP** (**Point of Presence**). We will next explain how packets are moved between the POPs of different ISPs. From this point on, the system is fully digital and packet switched.

ISP networks may be regional, national, or international in scope. We have already seen that their architecture is made up of long-distance transmission lines that interconnect routers at POPs in the different cities that the ISPs serve. This equipment is called the **backbone** of the ISP. If a packet is destined for a host served directly by the ISP, that packet is routed over the backbone and delivered to the host. Otherwise, it must be handed over to another ISP.

ISPs connect their networks to exchange traffic at **IXPs** (**Internet eXchange Points**). The connected ISPs are said to **peer** with each other. There are many IXPs in cities around the world. They are drawn vertically in Fig. 1-29 because ISP networks overlap geographically. Basically, an IXP is a room full of routers, at least one per ISP. A LAN in the room connects all the routers, so packets can be forwarded from any ISP backbone to any other ISP backbone. IXPs can be large and independently owned facilities. One of the largest is the Amsterdam Internet Exchange, to which hundreds of ISPs connect and through which they exchange hundreds of gigabits/sec of traffic.

The peering that happens at IXPs depends on the business relationships between ISPs. There are many possible relationships. For example, a small ISP might pay a larger ISP for Internet connectivity to reach distant hosts, much as a customer purchases service from an Internet provider. In this case, the small ISP is said to pay for **transit**. Alternatively, two large ISPs might decide to exchange traffic so that each ISP can deliver some traffic to the other ISP without having to pay for transit. One of the many paradoxes of the Internet is that ISPs who publicly compete with one another for customers often privately cooperate to do peering (Metz, 2001).

The path a packet takes through the Internet depends on the peering choices of the ISPs. If the ISP delivering a packet peers with the destination ISP, it might deliver the packet directly to its peer. Otherwise, it might route the packet to the nearest place at which it connects to a paid transit provider so that provider can deliver the packet. Two example paths across ISPs are drawn in Fig. 1-29. Often, the path a packet takes will not be the shortest path through the Internet.

At the top of the food chain are a small handful of companies, like AT&T and Sprint, that operate large international backbone networks with thousands of routers connected by high-bandwidth fiber optic links. These ISPs do not pay for transit. They are usually called **tier 1** ISPs and are said to form the backbone of the Internet, since everyone else must connect to them to be able to reach the entire Internet.

Companies that provide lots of content, such as Google and Yahoo!, locate their computers in **data centers** that are well connected to the rest of the Internet. These data centers are designed for computers, not humans, and may be filled with rack upon rack of machines called a **server farm**. **Colocation** or **hosting** data centers let customers put equipment such as servers at ISP POPs so that short, fast connections can be made between the servers and the ISP backbones. The Internet hosting industry has become increasingly virtualized so that it is now common to rent a virtual machine that is run on a server farm instead of installing a physical computer. These data centers are so large (tens or hundreds of thousands of machines) that electricity is a major cost, so data centers are sometimes built in areas where electricity is cheap.

This ends our quick tour of the Internet. We will have a great deal to say about the individual components and their design, algorithms, and protocols in subsequent chapters. One further point worth mentioning here is that what it means to be on the Internet is changing. It used to be that a machine was on the Internet if it: (1) ran the TCP/IP protocol stack; (2) had an IP address; and (3) could send IP packets to all the other machines on the Internet. However, ISPs often reuse IP addresses depending on which computers are in use at the moment, and home networks often share one IP address between multiple computers. This practice undermines the second condition. Security measures such as firewalls can also partly block computers from receiving packets, undermining the third condition. Despite these difficulties, it makes sense to regard such machines as being on the Internet while they are connected to their ISPs.

Also worth mentioning in passing is that some companies have interconnected all their existing internal networks, often using the same technology as the Internet. These **intranets** are typically accessible only on company premises or from company notebooks but otherwise work the same way as the Internet.

1.5.2 Third-Generation Mobile Phone Networks

People love to talk on the phone even more than they like to surf the Internet, and this has made the mobile phone network the most successful network in the world. It has more than four billion subscribers worldwide. To put this number in perspective, it is roughly 60% of the world's population and more than the number of Internet hosts and fixed telephone lines combined (ITU, 2009).

The architecture of the mobile phone network has changed greatly over the past 40 years along with its tremendous growth. First-generation mobile phone systems transmitted voice calls as continuously varying (analog) signals rather than sequences of (digital) bits. **AMPS** (**Advanced Mobile Phone System**), which was deployed in the United States in 1982, was a widely used first-generation system. Second-generation mobile phone systems switched to transmitting voice calls in digital form to increase capacity, improve security, and offer text messaging. **GSM** (**Global System for Mobile communications**), which was deployed starting in 1991 and has become the most widely used mobile phone system in the world, is a 2G system.

The third generation, or 3G, systems were initially deployed in 2001 and offer both digital voice and broadband digital data services. They also come with a lot of jargon and many different standards to choose from. 3G is loosely defined by the ITU (an international standards body we will discuss in the next section) as providing rates of at least 2 Mbps for stationary or walking users and 384 kbps in a moving vehicle. **UMTS** (**Universal Mobile Telecommunications System**), also called **WCDMA** (**Wideband Code Division Multiple Access**), is the main 3G system that is being rapidly deployed worldwide. It can provide up to 14 Mbps on the downlink and almost 6 Mbps on the uplink. Future releases will use multiple antennas and radios to provide even greater speeds for users.

The scarce resource in 3G systems, as in 2G and 1G systems before them, is radio spectrum. Governments license the right to use parts of the spectrum to the mobile phone network operators, often using a spectrum auction in which network operators submit bids. Having a piece of licensed spectrum makes it easier to design and operate systems, since no one else is allowed transmit on that spectrum, but it often costs a serious amount of money. In the UK in 2000, for example, five 3G licenses were auctioned for a total of about $40 billion.

It is the scarcity of spectrum that led to the **cellular network** design shown in Fig. 1-30 that is now used for mobile phone networks. To manage the radio interference between users, the coverage area is divided into cells. Within a cell, users are assigned channels that do not interfere with each other and do not cause too much interference for adjacent cells. This allows for good reuse of the spectrum, or **frequency reuse**, in the neighboring cells, which increases the capacity of the network. In 1G systems, which carried each voice call on a specific frequency band, the frequencies were carefully chosen so that they did not conflict with neighboring cells. In this way, a given frequency might only be reused once in several cells. Modern 3G systems allow each cell to use all frequencies, but in a way that results in a tolerable level of interference to the neighboring cells. There are variations on the cellular design, including the use of directional or sectored antennas on cell towers to further reduce interference, but the basic idea is the same.

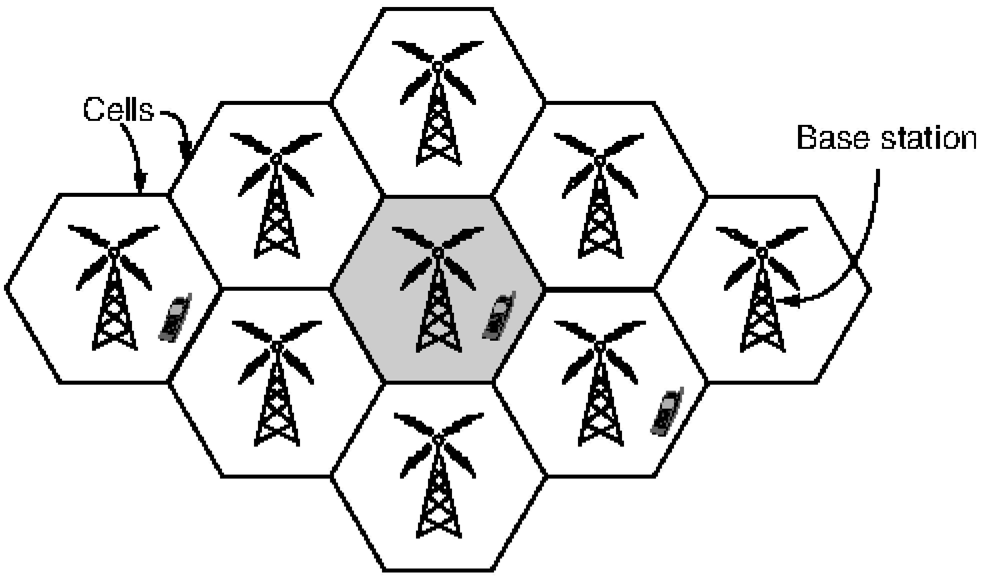


Figure 1-30. Cellular design of mobile phone networks.

The architecture of the mobile phone network is very different than that of the Internet. It has several parts, as shown in the simplified version of the UMTS architecture in Fig. 1-31. First, there is the **air interface**. This term is a fancy name for the radio communication protocol that is used over the air between the mobile device (e.g., the cell phone) and the **cellular base station**. Advances in the air interface over the past decades have greatly increased wireless data rates. The UMTS air interface is based on **Code Division Multiple Access** (**CDMA**), a technique that we will study in Chap. 2.

The cellular base station together with its controller forms the **radio access network**. This part is the wireless side of the mobile phone network. The controller node or **RNC** (**Radio Network Controller**) controls how the spectrum is used. The base station implements the air interface. It is called **Node B**, a temporary label that stuck.

The rest of the mobile phone network carries the traffic for the radio access network. It is called the **core network**. The UMTS core network evolved from the core network used for the 2G GSM system that came before it. However, something surprising is happening in the UMTS core network.

Since the beginning of networking, a war has been going on between the people who support packet networks (i.e., connectionless subnets) and the people who support circuit networks (i.e., connection-oriented subnets). The main proponents of packets come from the Internet community. In a connectionless design, every packet is routed independently of every other packet. As a consequence, if some routers go down during a session, no harm will be done as long as the system can

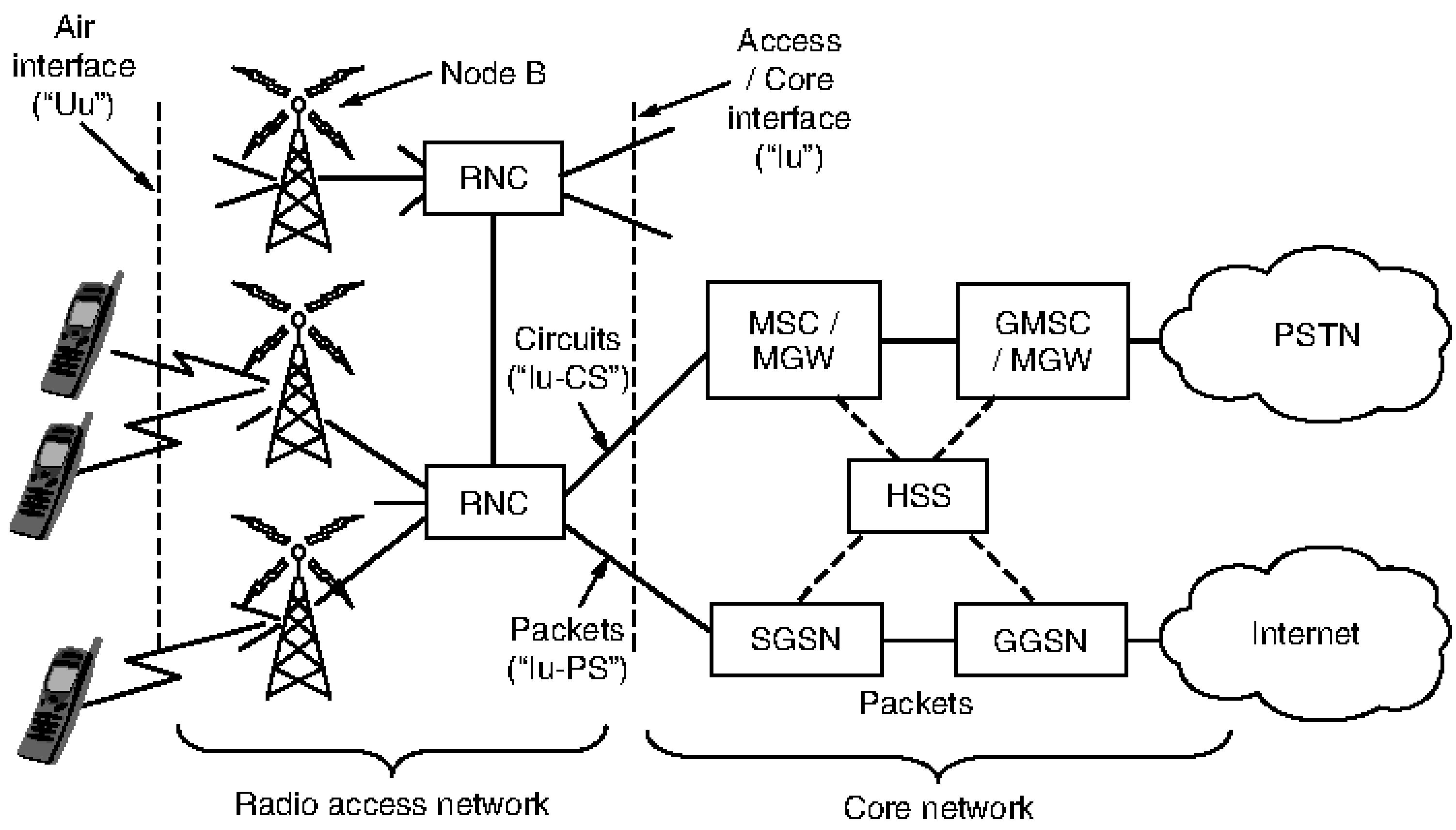


Figure 1-31. Architecture of the UMTS 3G mobile phone network.

dynamically reconfigure itself so that subsequent packets can find some route to the destination, even if it is different from that which previous packets used.

The circuit camp comes from the world of telephone companies. In the telephone system, a caller must dial the called party's number and wait for a connection before talking or sending data. This connection setup establishes a route through the telephone system that is maintained until the call is terminated. All words or packets follow the same route. If a line or switch on the path goes down, the call is aborted, making it less fault tolerant than a connectionless design.

The advantage of circuits is that they can support quality of service more easily. By setting up a connection in advance, the subnet can reserve resources such as link bandwidth, switch buffer space, and CPU. If an attempt is made to set up a call and insufficient resources are available, the call is rejected and the caller gets a kind of busy signal. In this way, once a connection has been set up, the connection will get good service.

With a connectionless network, if too many packets arrive at the same router at the same moment, the router will choke and probably lose packets. The sender will eventually notice this and resend them, but the quality of service will be jerky and unsuitable for audio or video unless the network is lightly loaded. Needless to say, providing adequate audio quality is something telephone companies care about very much, hence their preference for connections.

The surprise in Fig. 1-31 is that there is both packet and circuit switched equipment in the core network. This shows the mobile phone network in transition, with mobile phone companies able to implement one or sometimes both of the alternatives. Older mobile phone networks used a circuit-switched core in the style of the traditional phone network to carry voice calls. This legacy is seen in the UMTS network with the **MSC** (**Mobile Switching Center**), **GMSC** (**Gateway Mobile Switching Center**), and **MGW** (**Media Gateway**) elements that set up connections over a circuit-switched core network such as the **PSTN** (**Public Switched Telephone Network**).

Data services have become a much more important part of the mobile phone network than they used to be, starting with text messaging and early packet data services such as **GPRS** (**General Packet Radio Service**) in the GSM system. These older data services ran at tens of kbps, but users wanted more. Newer mobile phone networks carry packet data at rates of multiple Mbps. For comparison, a voice call is carried at a rate of 64 kbps, typically 3–4x less with compression.

To carry all this data, the UMTS core network nodes connect directly to a packet-switched network. The **SGSN** (**Serving GPRS Support Node**) and the **GGSN** (**Gateway GPRS Support Node**) deliver data packets to and from mobiles and interface to external packet networks such as the Internet.

This transition is set to continue in the mobile phone networks that are now being planned and deployed. Internet protocols are even used on mobiles to set up connections for voice calls over a packet data network, in the manner of voice-over-IP. IP and packets are used all the way from the radio access through to the core network. Of course, the way that IP networks are designed is also changing to support better quality of service. If it did not, then problems with chopped-up audio and jerky video would not impress paying customers. We will return to this subject in Chap. 5.

Another difference between mobile phone networks and the traditional Internet is mobility. When a user moves out of the range of one cellular base station and into the range of another one, the flow of data must be re-routed from the old to the new cell base station. This technique is known as **handover** or **handoff**, and it is illustrated in Fig. 1-32.

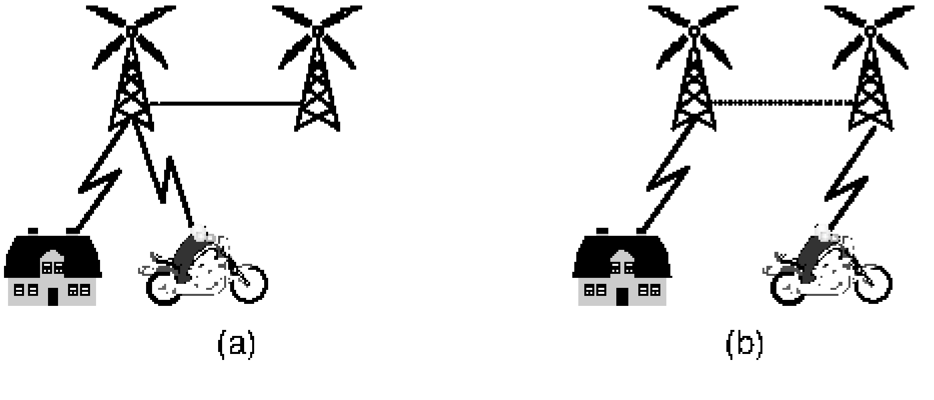


Figure 1-32. Mobile phone handover (a) before, (b) after.

Either the mobile device or the base station may request a handover when the quality of the signal drops. In some cell networks, usually those based on CDMA technology, it is possible to connect to the new base station before disconnecting from the old base station. This improves the connection quality for the mobile because there is no break in service; the mobile is actually connected to two base stations for a short while. This way of doing a handover is called a **soft handover** to distinguish it from a **hard handover**, in which the mobile disconnects from the old base station before connecting to the new one.

A related issue is how to find a mobile in the first place when there is an incoming call. Each mobile phone network has a **HSS** (**Home Subscriber Server**) in the core network that knows the location of each subscriber, as well as other profile information that is used for authentication and authorization. In this way, each mobile can be found by contacting the HSS.

A final area to discuss is security. Historically, phone companies have taken security much more seriously than Internet companies for a long time because of the need to bill for service and avoid (payment) fraud. Unfortunately that is not saying much. Nevertheless, in the evolution from 1G through 3G technologies, mobile phone companies have been able to roll out some basic security mechanisms for mobiles.

Starting with the 2G GSM system, the mobile phone was divided into a handset and a removable chip containing the subscriber's identity and account information. The chip is informally called a **SIM card**, short for **Subscriber Identity Module**. SIM cards can be switched to different handsets to activate them, and they provide a basis for security. When GSM customers travel to other countries on vacation or business, they often bring their handsets but buy a new SIM card for few dollars upon arrival in order to make local calls with no roaming charges.

To reduce fraud, information on SIM cards is also used by the mobile phone network to authenticate subscribers and check that they are allowed to use the network. With UMTS, the mobile also uses the information on the SIM card to check that it is talking to a legitimate network.

Another aspect of security is privacy. Wireless signals are broadcast to all nearby receivers, so to make it difficult to eavesdrop on conversations, cryptographic keys on the SIM card are used to encrypt transmissions. This approach provides much better privacy than in 1G systems, which were easily tapped, but is not a panacea due to weaknesses in the encryption schemes.

Mobile phone networks are destined to play a central role in future networks. They are now more about mobile broadband applications than voice calls, and this has major implications for the air interfaces, core network architecture, and security of future networks. 4G technologies that are faster and better are on the drawing board under the name of **LTE** (**Long Term Evolution**), even as 3G design and deployment continues. Other wireless technologies also offer broadband Internet access to fixed and mobile clients, notably 802.16 networks under the common name of **WiMAX**. It is entirely possible that LTE and WiMAX are on a collision course with each other and it is hard to predict what will happen to them.

1.5.3 Wireless LANs: 802.11

Almost as soon as laptop computers appeared, many people had a dream of walking into an office and magically having their laptop computer be connected to the Internet. Consequently, various groups began working on ways to accomplish this goal. The most practical approach is to equip both the office and the laptop computers with short-range radio transmitters and receivers to allow them to talk.

Work in this field rapidly led to wireless LANs being marketed by a variety of companies. The trouble was that no two of them were compatible. The proliferation of standards meant that a computer equipped with a brand *X* radio would not work in a room equipped with a brand *Y* base station. In the mid 1990s, the industry decided that a wireless LAN standard might be a good idea, so the IEEE committee that had standardized wired LANs was given the task of drawing up a wireless LAN standard.

The first decision was the easiest: what to call it. All the other LAN standards had numbers like 802.1, 802.2, and 802.3, up to 802.10, so the wireless LAN standard was dubbed 802.11. A common slang name for it is **WiFi** but it is an important standard and deserves respect, so we will call it by its proper name, 802.11.

The rest was harder. The first problem was to find a suitable frequency band that was available, preferably worldwide. The approach taken was the opposite of that used in mobile phone networks. Instead of expensive, licensed spectrum, 802.11 systems operate in unlicensed bands such as the **ISM** (**Industrial, Scientific, and Medical**) bands defined by ITU-R (e.g., 902-928 MHz, 2.4-2.5 GHz, 5.725-5.825 GHz). All devices are allowed to use this spectrum provided that they limit their transmit power to let different devices coexist. Of course, this means that 802.11 radios may find themselves competing with cordless phones, garage door openers, and microwave ovens.

802.11 networks are made up of clients, such as laptops and mobile phones, and infrastructure called **APs** (**access points**) that is installed in buildings. Access points are sometimes called **base stations**. The access points connect to the wired network, and all communication between clients goes through an access point. It is also possible for clients that are in radio range to talk directly, such as two computers in an office without an access point. This arrangement is called an **ad hoc network**. It is used much less often than the access point mode. Both modes are shown in Fig. 1-33.

802.11 transmission is complicated by wireless conditions that vary with even small changes in the environment. At the frequencies used for 802.11, radio signals can be reflected off solid objects so that multiple echoes of a transmission may reach a receiver along different paths. The echoes can cancel or reinforce each other, causing the received signal to fluctuate greatly. This phenomenon is called **multipath fading**, and it is shown in Fig. 1-34.

The key idea for overcoming variable wireless conditions is **path diversity**, or the sending of information along multiple, independent paths. In this way, the

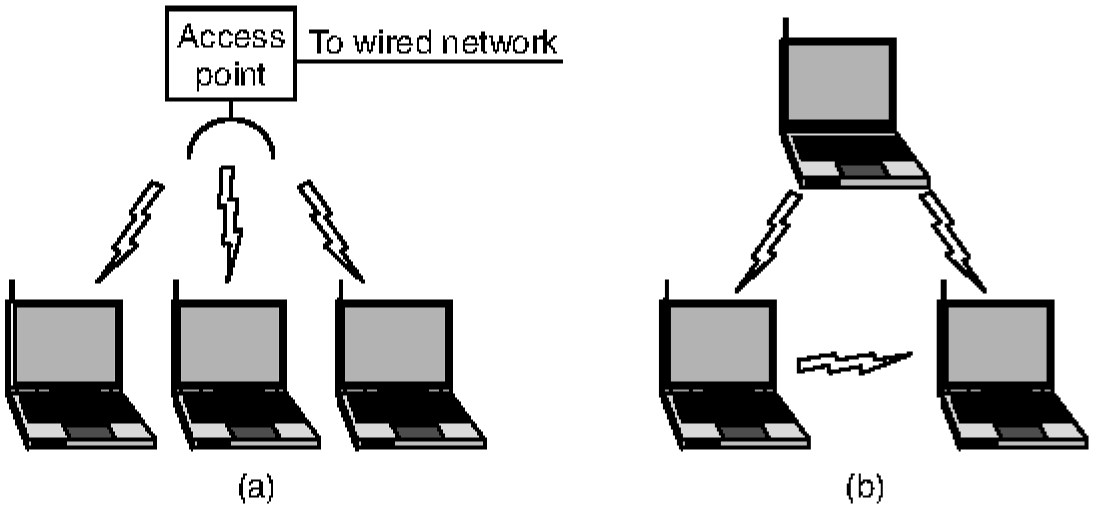


Figure 1-33. (a) Wireless network with an access point. (b) Ad hoc network.

information is likely to be received even if one of the paths happens to be poor due to a fade. These independent paths are typically built into the digital modulation scheme at the physical layer. Options include using different frequencies across the allowed band, following different spatial paths between different pairs of antennas, or repeating bits over different periods of time.

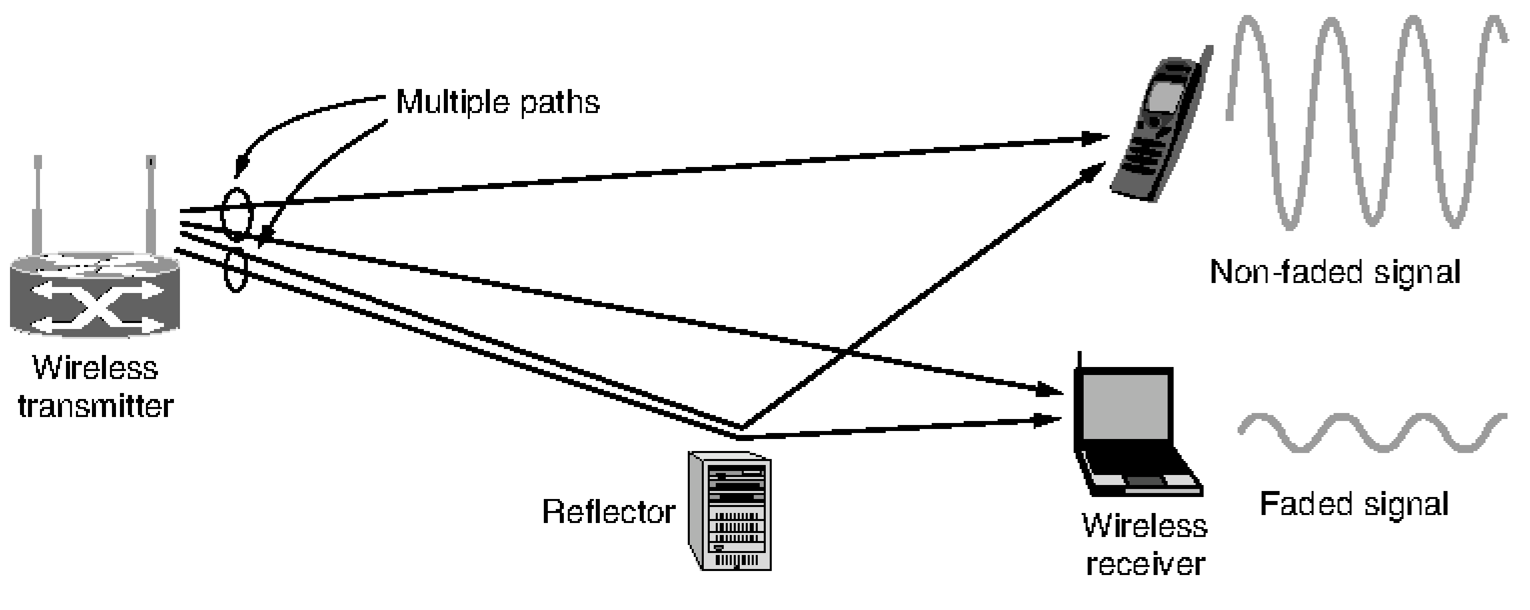


Figure 1-34. Multipath fading.

Different versions of 802.11 have used all of these techniques. The initial (1997) standard defined a wireless LAN that ran at either 1 Mbps or 2 Mbps by hopping between frequencies or spreading the signal across the allowed spectrum. Almost immediately, people complained that it was too slow, so work began on faster standards. The spread spectrum design was extended and became the (1999) 802.11b standard running at rates up to 11 Mbps. The 802.11a (1999) and 802.11g (2003) standards switched to a different modulation scheme called **OFDM** (**Orthogonal Frequency Division Multiplexing**). It divides a wide band of spectrum into many narrow slices over which different bits are sent in parallel. This improved scheme, which we will study in Chap. 2, boosted the 802.11a/g bit rates up to 54 Mbps. That is a significant increase, but people still wanted more throughput to support more demanding uses. The latest version is 802.11n (2009). It uses wider frequency bands and up to four antennas per computer to achieve rates up to 450 Mbps.

Since wireless is inherently a broadcast medium, 802.11 radios also have to deal with the problem that multiple transmissions that are sent at the same time will collide, which may interfere with reception. To handle this problem, 802.11 uses a **CSMA** (**Carrier Sense Multiple Access**) scheme that draws on ideas from classic wired Ethernet, which, ironically, drew from an early wireless network developed in Hawaii and called **ALOHA**. Computers wait for a short random interval before transmitting, and defer their transmissions if they hear that someone else is already transmitting. This scheme makes it less likely that two computers will send at the same time. It does not work as well as in the case of wired networks, though. To see why, examine Fig. 1-35. Suppose that computer *A* is transmitting to computer *B*, but the radio range of *A*'s transmitter is too short to reach computer *C*. If *C* wants to transmit to *B* it can listen before starting, but the fact that it does not hear anything does not mean that its transmission will succeed. The inability of *C* to hear *A* before starting causes some collisions to occur. After any collision, the sender then waits another, longer, random delay and retransmits the packet. Despite this and some other issues, the scheme works well enough in practice.

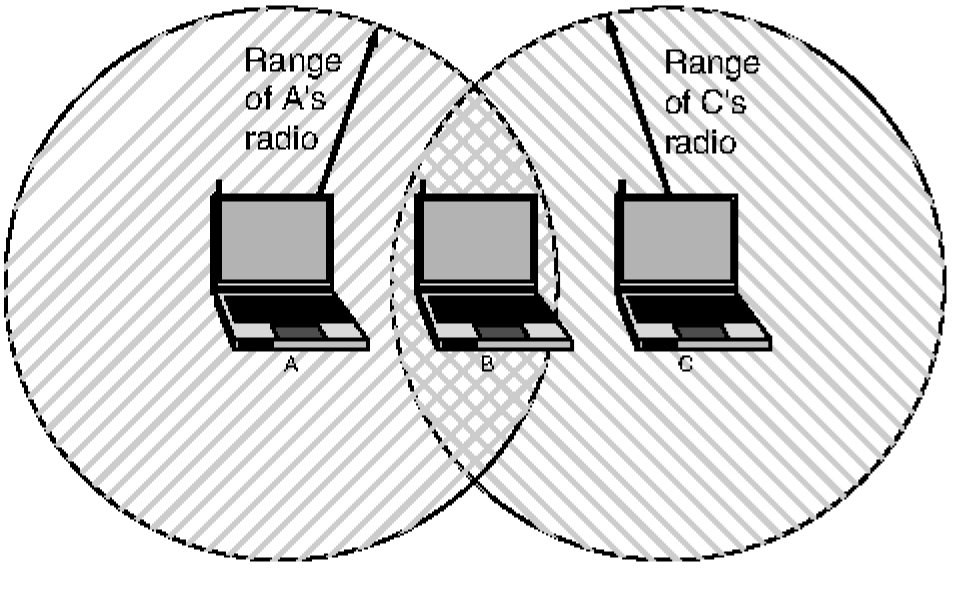


Figure 1-35. The range of a single radio may not cover the entire system.

Another problem is that of mobility. If a mobile client is moved away from the access point it is using and into the range of a different access point, some way of handing it off is needed. The solution is that an 802.11 network can consist of multiple cells, each with its own access point, and a distribution system that connects the cells. The distribution system is often switched Ethernet, but it can use any technology. As the clients move, they may find another access point with a better signal than the one they are currently using and change their association. From the outside, the entire system looks like a single wired LAN.

That said, mobility in 802.11 has been of limited value so far compared to mobility in the mobile phone network. Typically, 802.11 is used by nomadic clients that go from one fixed location to another, rather than being used on-the-go. Mobility is not really needed for nomadic usage. Even when 802.11 mobility is used, it extends over a single 802.11 network, which might cover at most a large building. Future schemes will need to provide mobility across different networks and across different technologies (e.g., 802.21).

Finally, there is the problem of security. Since wireless transmissions are broadcast, it is easy for nearby computers to receive packets of information that were not intended for them. To prevent this, the 802.11 standard included an encryption scheme known as **WEP** (**Wired Equivalent Privacy**). The idea was to make wireless security like that of wired security. It is a good idea, but unfortunately the scheme was flawed and soon broken (Borisov et al., 2001). It has since been replaced with newer schemes that have different cryptographic details in the 802.11i standard, also called **WiFi Protected Access**, initially called **WPA** but now replaced by **WPA2**.

802.11 has caused a revolution in wireless networking that is set to continue. Beyond buildings, it is starting to be installed in trains, planes, boats, and automobiles so that people can surf the Internet wherever they go. Mobile phones and all manner of consumer electronics, from game consoles to digital cameras, can communicate with it. We will come back to it in detail in Chap. 4.

1.5.4 RFID and Sensor Networks

The networks we have studied so far are made up of computing devices that are easy to recognize, from computers to mobile phones. With **Radio Frequency IDentification** (**RFID**), everyday objects can also be part of a computer network.

An RFID tag looks like a postage stamp-sized sticker that can be affixed to (or embedded in) an object so that it can be tracked. The object might be a cow, a passport, a book or a shipping pallet. The tag consists of a small microchip with a unique identifier and an antenna that receives radio transmissions. RFID readers installed at tracking points find tags when they come into range and interrogate them for their information as shown in Fig. 1-36. Applications include checking identities, managing the supply chain, timing races, and replacing barcodes.

There are many kinds of RFID, each with different properties, but perhaps the most fascinating aspect of RFID technology is that most RFID tags have neither an electric plug nor a battery. Instead, all of the energy needed to operate them is supplied in the form of radio waves by RFID readers. This technology is called **passive RFID** to distinguish it from the (less common) **active RFID** in which there is a power source on the tag.

One common form of RFID is **UHF RFID** (**Ultra-High Frequency RFID**). It is used on shipping pallets and some drivers licenses. Readers send signals in

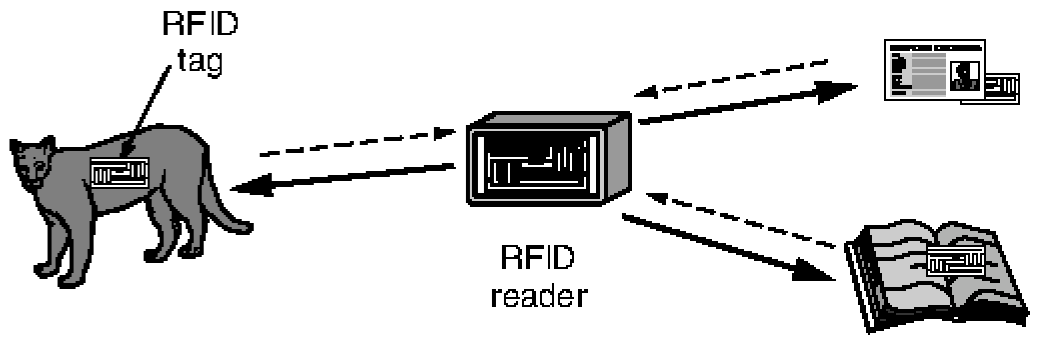


Figure 1-36. RFID used to network everyday objects.

the 902-928 MHz band in the United States. Tags communicate at distances of several meters by changing the way they reflect the reader signals; the reader is able to pick up these reflections. This way of operating is called **backscatter**.

Another popular kind of RFID is **HF RFID** (**High Frequency RFID**). It operates at 13.56 MHz and is likely to be in your passport, credit cards, books, and noncontact payment systems. HF RFID has a short range, typically a meter or less, because the physical mechanism is based on induction rather than backscatter. There are also other forms of RFID using other frequencies, such as **LF RFID** (**Low Frequency RFID**), which was developed before HF RFID and used for animal tracking. It is the kind of RFID likely to be in your cat.

RFID readers must somehow solve the problem of dealing with multiple tags within reading range. This means that a tag cannot simply respond when it hears a reader, or the signals from multiple tags may collide. The solution is similar to the approach taken in 802.11: tags wait for a short random interval before responding with their identification, which allows the reader to narrow down individual tags and interrogate them further.

Security is another problem. The ability of RFID readers to easily track an object, and hence the person who uses it, can be an invasion of privacy. Unfortunately, it is difficult to secure RFID tags because they lack the computation and communication power to run strong cryptographic algorithms. Instead, weak measures like passwords (which can easily be cracked) are used. If an identity card can be remotely read by an official at a border, what is to stop the same card from being tracked by other people without your knowledge? Not much.

RFID tags started as identification chips, but are rapidly turning into full-fledged computers. For example, many tags have memory that can be updated and later queried, so that information about what has happened to the tagged object can be stored with it. Rieback et al. (2006) demonstrated that this means that all of the usual problems of computer malware apply, only now your cat or your passport might be used to spread an RFID virus.

A step up in capability from RFID is the **sensor network**. Sensor networks are deployed to monitor aspects of the physical world. So far, they have mostly been used for scientific experimentation, such as monitoring bird habitats, volcanic activity, and zebra migration, but business applications including healthcare, monitoring equipment for vibration, and tracking of frozen, refrigerated, or otherwise perishable goods cannot be too far behind.

Sensor nodes are small computers, often the size of a key fob, that have temperature, vibration, and other sensors. Many nodes are placed in the environment that is to be monitored. Typically, they have batteries, though they may scavenge energy from vibrations or the sun. As with RFID, having enough energy is a key challenge, and the nodes must communicate carefully to be able to deliver their sensor information to an external collection point. A common strategy is for the nodes to self-organize to relay messages for each other, as shown in Fig. 1-37. This design is called a **multihop network**.

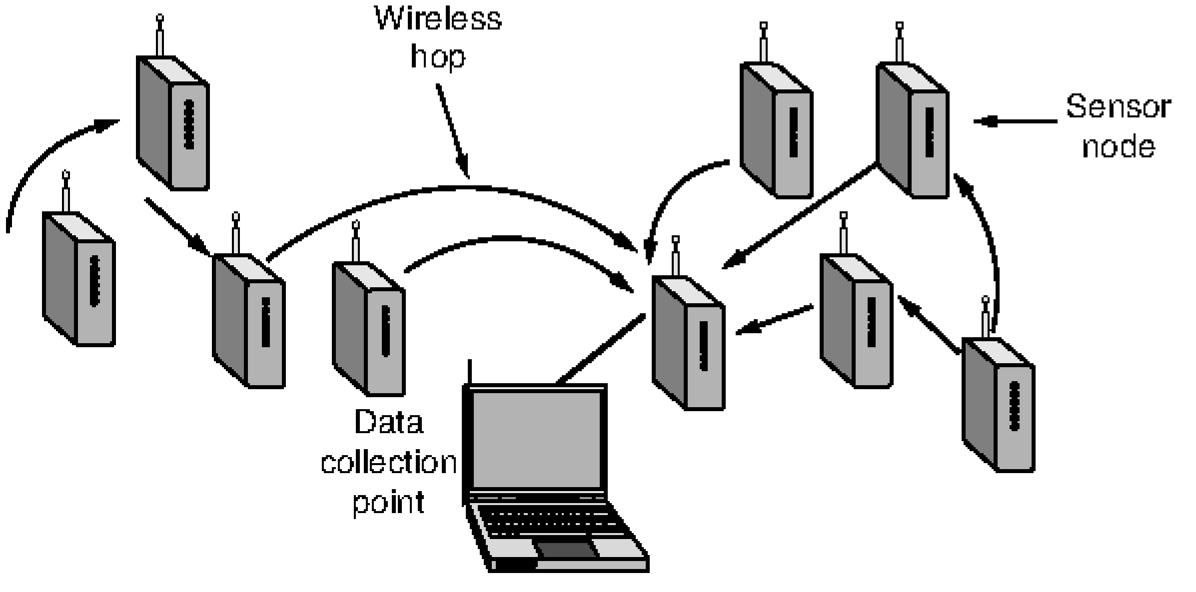


Figure 1-37. Multihop topology of a sensor network.

RFID and sensor networks are likely to become much more capable and pervasive in the future. Researchers have already combined the best of both technologies by prototyping programmable RFID tags with light, movement, and other sensors (Sample et al., 2008).

1.6 NETWORK STANDARDIZATION

Many network vendors and suppliers exist, each with its own ideas of how things should be done. Without coordination, there would be complete chaos, and users would get nothing done. The only way out is to agree on some network standards. Not only do good standards allow different computers to communicate, but they also increase the market for products adhering to the standards. A larger market leads to mass production, economies of scale in manufacturing, better implementations, and other benefits that decrease price and further increase acceptance.

In this section we will take a quick look at the important but little-known, world of international standardization. But let us first discuss what belongs in a standard. A reasonable person might assume that a standard tells you how a protocol should work so that you can do a good job of implementing it. That person would be wrong.

Standards define what is needed for interoperability: no more, no less. That lets the larger market emerge and also lets companies compete on the basis of how good their products are. For example, the 802.11 standard defines many transmission rates but does not say when a sender should use which rate, which is a key factor in good performance. That is up to whoever makes the product. Often getting to interoperability this way is difficult, since there are many implementation choices and standards usually define many options. For 802.11, there were so many problems that, in a strategy that has become common practice, a trade group called the **WiFi Alliance** was started to work on interoperability within the 802.11 standard.

Similarly, a protocol standard defines the protocol over the wire but not the service interface inside the box, except to help explain the protocol. Real service interfaces are often proprietary. For example, the way TCP interfaces to IP within a computer does not matter for talking to a remote host. It only matters that the remote host speaks TCP/IP. In fact, TCP and IP are commonly implemented together without any distinct interface. That said, good service interfaces, like good APIs, are valuable for getting protocols used, and the best ones (such as Berkeley sockets) can become very popular.

Standards fall into two categories: de facto and de jure. **De facto** (Latin for "from the fact") standards are those that have just happened, without any formal plan. HTTP, the protocol on which the Web runs, started life as a de facto standard. It was part of early WWW browsers developed by Tim Berners-Lee at CERN, and its use took off with the growth of the Web. Bluetooth is another example. It was originally developed by Ericsson but now everyone is using it.

De jure (Latin for "by law") standards, in contrast, are adopted through the rules of some formal standardization body. International standardization authorities are generally divided into two classes: those established by treaty among national governments, and those comprising voluntary, nontreaty organizations. In the area of computer network standards, there are several organizations of each type, notably ITU, ISO, IETF and IEEE, all of which we will discuss below.

In practice, the relationships between standards, companies, and standardization bodies are complicated. De facto standards often evolve into de jure standards, especially if they are successful. This happened in the case of HTTP, which was quickly picked up by IETF. Standards bodies often ratify each others' standards, in what looks like patting one another on the back, to increase the market for a technology. These days, many ad hoc business alliances that are formed around particular technologies also play a significant role in developing and refining network standards. For example, **3GPP** (**Third Generation Partnership Project**) is a collaboration between telecommunications associations that drives the UMTS 3G mobile phone standards.

1.6.1 Who's Who in the Telecommunications World

The legal status of the world's telephone companies varies considerably from country to country. At one extreme is the United States, which has over 2000 separate, (mostly very small) privately owned telephone companies. A few more were added with the breakup of AT&T in 1984 (which was then the world's largest corporation, providing telephone service to about 80 percent of America's telephones), and the Telecommunications Act of 1996 that overhauled regulation to foster competition.

At the other extreme are countries in which the national government has a complete monopoly on all communication, including the mail, telegraph, telephone, and often radio and television. Much of the world falls into this category. In some cases the telecommunication authority is a nationalized company, and in others it is simply a branch of the government, usually known as the **PTT** (**Post, Telegraph & Telephone** administration). Worldwide, the trend is toward liberalization and competition and away from government monopoly. Most European countries have now (partially) privatized their PTTs, but elsewhere the process is still only slowly gaining steam.

With all these different suppliers of services, there is clearly a need to provide compatibility on a worldwide scale to ensure that people (and computers) in one country can call their counterparts in another one. Actually, this need has existed for a long time. In 1865, representatives from many European governments met to form the predecessor to today's **ITU** (**International Telecommunication Union**). Its job was to standardize international telecommunications, which in those days meant telegraphy. Even then it was clear that if half the countries used Morse code and the other half used some other code, there was going to be a problem. When the telephone was put into international service, ITU took over the job of standardizing telephony (pronounced te-LEF-ony) as well. In 1947, ITU became an agency of the United Nations.

ITU has about 200 governmental members, including almost every member of the United Nations. Since the United States does not have a PTT, somebody else had to represent it in ITU. This task fell to the State Department, probably on the grounds that ITU had to do with foreign countries, the State Department's specialty. ITU also has more than 700 sector and associate members. They include telephone companies (e.g., AT&T, Vodafone, Sprint), telecom equipment manufacturers (e.g., Cisco, Nokia, Nortel), computer vendors (e.g., Microsoft, Agilent, Toshiba), chip manufacturers (e.g., Intel, Motorola, TI), and other interested companies (e.g., Boeing, CBS, VeriSign).

ITU has three main sectors. We will focus primarily on **ITU-T**, the Telecommunications Standardization Sector, which is concerned with telephone and data communication systems. Before 1993, this sector was called **CCITT**, which is an acronym for its French name, Comité Consultatif International Télégraphique et Téléphonique. **ITU-R**, the Radiocommunications Sector, is concerned with coordinating the use by competing interest groups of radio frequencies worldwide. The other sector is ITU-D, the Development Sector. It promotes the development of information and communication technologies to narrow the "digital divide" between countries with effective access to the information technologies and countries with limited access.

ITU-T's task is to make technical recommendations about telephone, telegraph, and data communication interfaces. These often become internationally recognized standards, though technically the recommendations are only suggestions that governments can adopt or ignore, as they wish (because governments are like 13-year-old boys—they do not take kindly to being given orders). In practice, a country that wishes to adopt a telephone standard different from that used by the rest of the world is free to do so, but at the price of cutting itself off from everyone else. This might work for North Korea, but elsewhere it would be a real problem.

The real work of ITU-T is done in its Study Groups. There are currently 10 Study Groups, often as large as 400 people, that cover topics ranging from telephone billing to multimedia services to security. SG 15, for example, standardizes the DSL technologies popularly used to connect to the Internet. In order to make it possible to get anything at all done, the Study Groups are divided into Working Parties, which are in turn divided into Expert Teams, which are in turn divided into ad hoc groups. Once a bureaucracy, always a bureaucracy.

Despite all this, ITU-T actually does get things done. Since its inception, it has produced more than 3000 recommendations, many of which are widely used in practice. For example, Recommendation H.264 (also an ISO standard known as MPEG-4 AVC) is widely used for video compression, and X.509 public key certificates are used for secure Web browsing and digitally signed email.

As the field of telecommunications completes the transition started in the 1980s from being entirely national to being entirely global, standards will become increasingly important, and more and more organizations will want to become involved in setting them. For more information about ITU, see Irmer (1994).

1.6.2 Who's Who in the International Standards World

International standards are produced and published by **ISO** (**International Standards Organization**†), a voluntary nontreaty organization founded in 1946. Its members are the national standards organizations of the 157 member countries. These members include ANSI (U.S.), BSI (Great Britain), AFNOR (France), DIN (Germany), and 153 others.

ISO issues standards on a truly vast number of subjects, ranging from nuts and bolts (literally) to telephone pole coatings [not to mention cocoa beans (ISO 2451), fishing nets (ISO 1530), women's underwear (ISO 4416) and quite a few

† For the purist, ISO's true name is the International Organization for Standardization.

other subjects one might not think were subject to standardization]. On issues of telecommunication standards, ISO and ITU-T often cooperate (ISO is a member of ITU-T) to avoid the irony of two official and mutually incompatible international standards.

Over 17,000 standards have been issued, including the OSI standards. ISO has over 200 Technical Committees (TCs), numbered in the order of their creation, each dealing with a specific subject. TC1 deals with the nuts and bolts (standardizing screw thread pitches). JTC1 deals with information technology, including networks, computers, and software. It is the first (and so far only) Joint Technical Committee, created in 1987 by merging TC97 with activities in IEC, yet another standardization body. Each TC has subcommittees (SCs) divided into working groups (WGs).

The real work is done largely in the WGs by over 100,000 volunteers worldwide. Many of these "volunteers" are assigned to work on ISO matters by their employers, whose products are being standardized. Others are government officials keen on having their country's way of doing things become the international standard. Academic experts also are active in many of the WGs.

The procedure used by ISO for adopting standards has been designed to achieve as broad a consensus as possible. The process begins when one of the national standards organizations feels the need for an international standard in some area. A working group is then formed to come up with a **CD** (**Committee Draft**). The CD is then circulated to all the member bodies, which get 6 months to criticize it. If a substantial majority approves, a revised document, called a **DIS** (**Draft International Standard**) is produced and circulated for comments and voting. Based on the results of this round, the final text of the **IS** (**International Standard**) is prepared, approved, and published. In areas of great controversy, a CD or DIS may have to go through several versions before acquiring enough votes, and the whole process can take years.

NIST (**National Institute of Standards and Technology**) is part of the U.S. Department of Commerce. It used to be called the National Bureau of Standards. It issues standards that are mandatory for purchases made by the U.S. Government, except for those of the Department of Defense, which defines its own standards.

Another major player in the standards world is **IEEE** (**Institute of Electrical and Electronics Engineers**), the largest professional organization in the world. In addition to publishing scores of journals and running hundreds of conferences each year, IEEE has a standardization group that develops standards in the area of electrical engineering and computing. IEEE's 802 committee has standardized many kinds of LANs. We will study some of its output later in this book. The actual work is done by a collection of working groups, which are listed in Fig. 1-38. The success rate of the various 802 working groups has been low; having an 802.x number is no guarantee of success. Still, the impact of the success stories (especially 802.3 and 802.11) on the industry and the world has been enormous.

Number	Topic
802.1	Overview and architecture of LANs
802.2 ↓	Logical link control
802.3 *	Ethernet
802.4 ↓	Token bus (was briefly used in manufacturing plants)
802.5	Token ring (IBM's entry into the LAN world)
802.6 ↓	Dual queue dual bus (early metropolitan area network)
802.7 ↓	Technical advisory group on broadband technologies
802.8 †	Technical advisory group on fiber optic technologies
802.9 ↓	Isochronous LANs (for real-time applications)
802.10 ↓	Virtual LANs and security
802.11 *	Wireless LANs (WiFi)
802.12 ↓	Demand priority (Hewlett-Packard's AnyLAN)
802.13	Unlucky number; nobody wanted it
802.14 ↓	Cable modems (defunct: an industry consortium got there first)
802.15 *	Personal area networks (Bluetooth, Zigbee)
802.16 *	Broadband wireless (WiMAX)
802.17	Resilient packet ring
802.18	Technical advisory group on radio regulatory issues
802.19	Technical advisory group on coexistence of all these standards
802.20	Mobile broadband wireless (similar to 802.16e)
802.21	Media independent handoff (for roaming over technologies)
802.22	Wireless regional area network

Figure 1-38. The 802 working groups. The important ones are marked with *. The ones marked with ↓ are hibernating. The one marked with † gave up and disbanded itself.

1.6.3 Who's Who in the Internet Standards World

The worldwide Internet has its own standardization mechanisms, very different from those of ITU-T and ISO. The difference can be crudely summed up by saying that the people who come to ITU or ISO standardization meetings wear suits, while the people who come to Internet standardization meetings wear jeans (except when they meet in San Diego, when they wear shorts and T-shirts).

ITU-T and ISO meetings are populated by corporate officials and government civil servants for whom standardization is their job. They regard standardization as a Good Thing and devote their lives to it. Internet people, on the other hand, prefer anarchy as a matter of principle. However, with hundreds of millions of people all doing their own thing, little communication can occur. Thus, standards, however regrettable, are sometimes needed. In this context, David Clark of M.I.T. once made a now-famous remark about Internet standardization consisting of "rough consensus and running code."

When the ARPANET was set up, DoD created an informal committee to oversee it. In 1983, the committee was renamed the **IAB** (**Internet Activities Board**) and was given a slighter broader mission, namely, to keep the researchers involved with the ARPANET and the Internet pointed more or less in the same direction, an activity not unlike herding cats. The meaning of the acronym "IAB" was later changed to **Internet Architecture Board**.

Each of the approximately ten members of the IAB headed a task force on some issue of importance. The IAB met several times a year to discuss results and to give feedback to the DoD and NSF, which were providing most of the funding at this time. When a standard was needed (e.g., a new routing algorithm), the IAB members would thrash it out and then announce the change so the graduate students who were the heart of the software effort could implement it. Communication was done by a series of technical reports called **RFCs** (**Request For Comments**). RFCs are stored online and can be fetched by anyone interested in them from *www.ietf.org/rfc*. They are numbered in chronological order of creation. Over 5000 now exist. We will refer to many RFCs in this book.

By 1989, the Internet had grown so large that this highly informal style no longer worked. Many vendors by then offered TCP/IP products and did not want to change them just because ten researchers had thought of a better idea. In the summer of 1989, the IAB was reorganized again. The researchers were moved to the **IRTF** (**Internet Research Task Force**), which was made subsidiary to IAB, along with the **IETF** (**Internet Engineering Task Force**). The IAB was repopulated with people representing a broader range of organizations than just the research community. It was initially a self-perpetuating group, with members serving for a 2-year term and new members being appointed by the old ones. Later, the **Internet Society** was created, populated by people interested in the Internet. The Internet Society is thus in a sense comparable to ACM or IEEE. It is governed by elected trustees who appoint the IAB's members.

The idea of this split was to have the IRTF concentrate on long-term research while the IETF dealt with short-term engineering issues. The IETF was divided up into working groups, each with a specific problem to solve. The chairmen of these working groups initially met as a steering committee to direct the engineering effort. The working group topics include new applications, user information, OSI integration, routing and addressing, security, network management, and standards. Eventually, so many working groups were formed (more than 70) that they were grouped into areas and the area chairmen met as the steering committee.

In addition, a more formal standardization process was adopted, patterned after ISOs. To become a **Proposed Standard**, the basic idea must be explained in an RFC and have sufficient interest in the community to warrant consideration.

To advance to the **Draft Standard** stage, a working implementation must have been rigorously tested by at least two independent sites for at least 4 months. If the IAB is convinced that the idea is sound and the software works, it can declare the RFC to be an **Internet Standard**. Some Internet Standards have become DoD standards (MIL-STD), making them mandatory for DoD suppliers.

For Web standards, the **World Wide Web Consortium** (**W3C**) develops protocols and guidelines to facilitate the long-term growth of the Web. It is an industry consortium led by Tim Berners-Lee and set up in 1994 as the Web really begun to take off. W3C now has more than 300 members from around the world and has produced more than 100 W3C Recommendations, as its standards are called, covering topics such as HTML and Web privacy.

1.7 METRIC UNITS

To avoid any confusion, it is worth stating explicitly that in this book, as in computer science in general, metric units are used instead of traditional English units (the furlong-stone-fortnight system). The principal metric prefixes are listed in Fig. 1-39. The prefixes are typically abbreviated by their first letters, with the units greater than 1 capitalized (KB, MB, etc.). One exception (for historical reasons) is kbps for kilobits/sec. Thus, a 1-Mbps communication line transmits 10^6 bits/sec and a 100-psec (or 100-ps) clock ticks every 10^{-10} seconds. Since milli and micro both begin with the letter "m," a choice had to be made. Normally, "m" is used for milli and "μ" (the Greek letter mu) is used for micro.

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.0000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.0000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.0000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Figure 1-39. The principal metric prefixes.

It is also worth pointing out that for measuring memory, disk, file, and database sizes, in common industry practice, the units have slightly different meanings. There, kilo means 2^{10} (1024) rather than 10^3 (1000) because memories are always a power of two. Thus, a 1-KB memory contains 1024 bytes, not 1000 bytes. Note also the capital "B" in that usage to mean "bytes" (units of eight bits), instead of a lowercase "b" that means "bits." Similarly, a 1-MB memory contains 2^{20} (1,048,576) bytes, a 1-GB memory contains 2^{30} (1,073,741,824) bytes, and a 1-TB database contains 2^{40} (1,099,511,627,776) bytes. However, a 1-kbps communication line transmits 1000 bits per second and a 10-Mbps LAN runs at 10,000,000 bits/sec because these speeds are not powers of two. Unfortunately, many people tend to mix up these two systems, especially for disk sizes. To avoid ambiguity, in this book, we will use the symbols KB, MB, GB, and TB for 2^{10}, 2^{20}, 2^{30}, and 2^{40} bytes, respectively, and the symbols kbps, Mbps, Gbps, and Tbps for 10^3, 10^6, 10^9, and 10^{12} bits/sec, respectively.

1.8 OUTLINE OF THE REST OF THE BOOK

This book discusses both the principles and practice of computer networking. Most chapters start with a discussion of the relevant principles, followed by a number of examples that illustrate these principles. These examples are usually taken from the Internet and wireless networks such as the mobile phone network since these are both important and very different. Other examples will be given where relevant.

The book is structured according to the hybrid model of Fig. 1-23. Starting with Chap. 2, we begin working our way up the protocol hierarchy beginning at the bottom. We provide some background in the field of data communication that covers both wired and wireless transmission systems. This material is concerned with how to deliver information over physical channels, although we cover only the architectural rather than the hardware aspects. Several examples of the physical layer, such as the public switched telephone network, the mobile telephone network, and the cable television network are also discussed.

Chapters 3 and 4 discuss the data link layer in two parts. Chap. 3 looks at the problem of how to send packets across a link, including error detection and correction. We look at DSL (used for broadband Internet access over phone lines) as a real-world example of a data link protocol.

In Chap. 4, we examine the medium access sublayer. This is the part of the data link layer that deals with how to share a channel between multiple computers. The examples we look at include wireless, such as 802.11 and RFID, and wired LANs such as classic Ethernet. Link layer switches that connect LANs, such as switched Ethernet, are also discussed here.

Chapter 5 deals with the network layer, especially routing. Many routing algorithms, both static and dynamic, are covered. Even with good routing algorithms, though, if more traffic is offered than the network can handle, some packets will be delayed or discarded. We discuss this issue from how to prevent congestion to how to guarantee a certain quality of service. Connecting heterogeneous networks to form internetworks also leads to numerous problems that are discussed here. The network layer in the Internet is given extensive coverage.

Chapter 6 deals with the transport layer. Much of the emphasis is on connection-oriented protocols and reliability, since many applications need these. Both Internet transport protocols, UDP and TCP, are covered in detail, as are their performance issues.

Chapter 7 deals with the application layer, its protocols, and its applications. The first topic is DNS, which is the Internet's telephone book. Next comes email, including a discussion of its protocols. Then we move on to the Web, with detailed discussions of static and dynamic content, and what happens on the client and server sides. We follow this with a look at networked multimedia, including streaming audio and video. Finally, we discuss content-delivery networks, including peer-to-peer technology.

Chapter 8 is about network security. This topic has aspects that relate to all layers, so it is easiest to treat it after all the layers have been thoroughly explained. The chapter starts with an introduction to cryptography. Later, it shows how cryptography can be used to secure communication, email, and the Web. The chapter ends with a discussion of some areas in which security collides with privacy, freedom of speech, censorship, and other social issues.

Chapter 9 contains an annotated list of suggested readings arranged by chapter. It is intended to help those readers who would like to pursue their study of networking further. The chapter also has an alphabetical bibliography of all the references cited in this book.

The authors' Web site at Pearson:

http://www.pearsonhighered.com/tanenbaum

has a page with links to many tutorials, FAQs, companies, industry consortia, professional organizations, standards organizations, technologies, papers, and more.

1.9 SUMMARY

Computer networks have many uses, both for companies and for individuals, in the home and while on the move. Companies use networks of computers to share corporate information, typically using the client-server model with employee desktops acting as clients accessing powerful servers in the machine room. For individuals, networks offer access to a variety of information and entertainment resources, as well as a way to buy and sell products and services. Individuals often access the Internet via their phone or cable providers at home, though increasingly wireless access is used for laptops and phones. Technology advances are enabling new kinds of mobile applications and networks with computers embedded in appliances and other consumer devices. The same advances raise social issues such as privacy concerns.

Roughly speaking, networks can be divided into LANs, MANs, WANs, and internetworks. LANs typical cover a building and operate at high speeds. MANs usually cover a city. An example is the cable television system, which is now used by many people to access the Internet. WANs may cover a country or a continent. Some of the technologies used to build these networks are point-to-point (e.g., a cable) while others are broadcast (e.g.,wireless). Networks can be interconnected with routers to form internetworks, of which the Internet is the largest and best known example. Wireless networks, for example 802.11 LANs and 3G mobile telephony, are also becoming extremely popular.

Network software is built around protocols, which are rules by which processes communicate. Most networks support protocol hierarchies, with each layer providing services to the layer above it and insulating them from the details of the protocols used in the lower layers. Protocol stacks are typically based either on the OSI model or on the TCP/IP model. Both have link, network, transport, and application layers, but they differ on the other layers. Design issues include reliability, resource allocation, growth, security, and more. Much of this book deals with protocols and their design.

Networks provide various services to their users. These services can range from connectionless best-efforts packet delivery to connection-oriented guaranteed delivery. In some networks, connectionless service is provided in one layer and connection-oriented service is provided in the layer above it.

Well-known networks include the Internet, the 3G mobile telephone network, and 802.11 LANs. The Internet evolved from the ARPANET, to which other networks were added to form an internetwork. The present-day Internet is actually a collection of many thousands of networks that use the TCP/IP protocol stack. The 3G mobile telephone network provides wireless and mobile access to the Internet at speeds of multiple Mbps, and, of course, carries voice calls as well. Wireless LANs based on the IEEE 802.11 standard are deployed in many homes and cafes and can provide connectivity at rates in excess of 100 Mbps. New kinds of networks are emerging too, such as embedded sensor networks and networks based on RFID technology.

Enabling multiple computers to talk to each other requires a large amount of standardization, both in the hardware and software. Organizations such as ITU-T, ISO, IEEE, and IAB manage different parts of the standardization process.

PROBLEMS

1. Imagine that you have trained your St. Bernard, Bernie, to carry a box of three 8-mm tapes instead of a flask of brandy. (When your disk fills up, you consider that an emergency.) These tapes each contain 7 gigabytes. The dog can travel to your side, wherever you may be, at 18 km/hour. For what range of distances does Bernie have a higher data rate than a transmission line whose data rate (excluding overhead) is 150 Mbps? How does your answer change if (i) Bernie's speed is doubled; (ii) each tape capacity is doubled; (iii) the data rate of the transmission line is doubled.

2. An alternative to a LAN is simply a big timesharing system with terminals for all users. Give two advantages of a client-server system using a LAN.

3. The performance of a client-server system is strongly influenced by two major network characteristics: the bandwidth of the network (that is, how many bits/sec it can transport) and the latency (that is, how many seconds it takes for the first bit to get from the client to the server). Give an example of a network that exhibits high bandwidth but also high latency. Then give an example of one that has both low bandwidth and low latency.

4. Besides bandwidth and latency, what other parameter is needed to give a good characterization of the quality of service offered by a network used for (i) digitized voice traffic? (ii) video traffic? (iii) financial transaction traffic?

5. A factor in the delay of a store-and-forward packet-switching system is how long it takes to store and forward a packet through a switch. If switching time is 10 μsec, is this likely to be a major factor in the response of a client-server system where the client is in New York and the server is in California? Assume the propagation speed in copper and fiber to be 2/3 the speed of light in vacuum.

6. A client-server system uses a satellite network, with the satellite at a height of 40,000 km. What is the best-case delay in response to a request?

7. In the future, when everyone has a home terminal connected to a computer network, instant public referendums on important pending legislation will become possible. Ultimately, existing legislatures could be eliminated, to let the will of the people be expressed directly. The positive aspects of such a direct democracy are fairly obvious; discuss some of the negative aspects.

8. Five routers are to be connected in a point-to-point subnet. Between each pair of routers, the designers may put a high-speed line, a medium-speed line, a low-speed line, or no line. If it takes 100 ms of computer time to generate and inspect each topology, how long will it take to inspect all of them?

9. A disadvantage of a broadcast subnet is the capacity wasted when multiple hosts attempt to access the channel at the same time. As a simplistic example, suppose that time is divided into discrete slots, with each of the n hosts attempting to use the channel with probability p during each slot. What fraction of the slots will be wasted due to collisions?

10. What are two reasons for using layered protocols? What is one possible disadvantage of using layered protocols?

11. The president of the Specialty Paint Corp. gets the idea to work with a local beer brewer to produce an invisible beer can (as an anti-litter measure). The president tells her legal department to look into it, and they in turn ask engineering for help. As a result, the chief engineer calls his counterpart at the brewery to discuss the technical aspects of the project. The engineers then report back to their respective legal departments, which then confer by telephone to arrange the legal aspects. Finally, the two corporate presidents discuss the financial side of the deal. What principle of a multilayer protocol in the sense of the OSI model does this communication mechanism violate?

12. Two networks each provide reliable connection-oriented service. One of them offers a reliable byte stream and the other offers a reliable message stream. Are these identical? If so, why is the distinction made? If not, give an example of how they differ.

13. What does "negotiation" mean when discussing network protocols? Give an example.

14. In Fig. 1-19, a service is shown. Are any other services implicit in this figure? If so, where? If not, why not?

15. In some networks, the data link layer handles transmission errors by requesting that damaged frames be retransmitted. If the probability of a frame's being damaged is p, what is the mean number of transmissions required to send a frame? Assume that acknowledgements are never lost.

16. A system has an n-layer protocol hierarchy. Applications generate messages of length M bytes. At each of the layers, an h-byte header is added. What fraction of the network bandwidth is filled with headers?

17. What is the main difference between TCP and UDP?

18. The subnet of Fig. 1-25(b) was designed to withstand a nuclear war. How many bombs would it take to partition the nodes into two disconnected sets? Assume that any bomb wipes out a node and all of the links connected to it.

19. The Internet is roughly doubling in size every 18 months. Although no one really knows for sure, one estimate put the number of hosts on it at 600 million in 2009. Use these data to compute the expected number of Internet hosts in the year 2018. Do you believe this? Explain why or why not.

20. When a file is transferred between two computers, two acknowledgement strategies are possible. In the first one, the file is chopped up into packets, which are individually acknowledged by the receiver, but the file transfer as a whole is not acknowledged. In the second one, the packets are not acknowledged individually, but the entire file is acknowledged when it arrives. Discuss these two approaches.

21. Mobile phone network operators need to know where their subscribers' mobile phones (hence their users) are located. Explain why this is bad for users. Now give reasons why this is good for users.

22. How long was a bit in the original 802.3 standard in meters? Use a transmission speed of 10 Mbps and assume the propagation speed in coax is 2/3 the speed of light in vacuum.

23. An image is 1600×1200 pixels with 3 bytes/pixel. Assume the image is uncompressed. How long does it take to transmit it over a 56-kbps modem channel? Over a 1-Mbps cable modem? Over a 10-Mbps Ethernet? Over 100-Mbps Ethernet? Over gigabit Ethernet?

24. Ethernet and wireless networks have some similarities and some differences. One property of Ethernet is that only one frame at a time can be transmitted on an Ethernet. Does 802.11 share this property with Ethernet? Discuss your answer.

25. List two advantages and two disadvantages of having international standards for network protocols.

26. When a system has a permanent part and a removable part (such as a CD-ROM drive and the CD-ROM), it is important that the system be standardized, so that different companies can make both the permanent and removable parts and everything still works together. Give three examples outside the computer industry where such international standards exist. Now give three areas outside the computer industry where they do not exist.

27. Suppose the algorithms used to implement the operations at layer k is changed. How does this impact operations at layers $k-1$ and $k+1$?

28. Suppose there is a change in the service (set of operations) provided by layer k. How does this impact services at layers k-1 and k+1?

29. Provide a list of reasons for why the response time of a client may be larger than the best-case delay.

30. What are the disadvantages of using small, fixed-length cells in ATM?

31. Make a list of activities that you do every day in which computer networks are used. How would your life be altered if these networks were suddenly switched off?

32. Find out what networks are used at your school or place of work. Describe the network types, topologies, and switching methods used there.

33. The *ping* program allows you to send a test packet to a given location and see how long it takes to get there and back. Try using *ping* to see how long it takes to get from your location to several known locations. From these data, plot the one-way transit time over the Internet as a function of distance. It is best to use universities since the location of their servers is known very accurately. For example, *berkeley.edu* is in Berkeley, California; *mit.edu* is in Cambridge, Massachusetts; *vu.nl* is in Amsterdam; The Netherlands; *www.usyd.edu.au* is in Sydney, Australia; and *www.uct.ac.za* is in Cape Town, South Africa.

34. Go to IETF's Web site, *www.ietf.org*, to see what they are doing. Pick a project you like and write a half-page report on the problem and the proposed solution.

35. The Internet is made up of a large number of networks. Their arrangement determines the topology of the Internet. A considerable amount of information about the Internet topology is available on line. Use a search engine to find out more about the Internet topology and write a short report summarizing your findings.

36. Search the Internet to find out some of the important peering points used for routing packets in the Internet at present.

37. Write a program that implements message flow from the top layer to the bottom layer of the 7-layer protocol model. Your program should include a separate protocol function for each layer. Protocol headers are sequence up to 64 characters. Each protocol function has two parameters: a message passed from the higher layer protocol (a char buffer) and the size of the message. This function attaches its header in front of the message, prints the new message on the standard output, and then invokes the protocol function of the lower-layer protocol. Program input is an application message (a sequence of 80 characters or less).

2

THE PHYSICAL LAYER

In this chapter we will look at the lowest layer in our protocol model, the physical layer. It defines the electrical, timing and other interfaces by which bits are sent as signals over channels. The physical layer is the foundation on which the network is built. The properties of different kinds of physical channels determine the performance (e.g., throughput, latency, and error rate) so it is a good place to start our journey into networkland.

We will begin with a theoretical analysis of data transmission, only to discover that Mother (Parent?) Nature puts some limits on what can be sent over a channel. Then we will cover three kinds of transmission media: guided (copper wire and fiber optics), wireless (terrestrial radio), and satellite. Each of these technologies has different properties that affect the design and performance of the networks that use them. This material will provide background information on the key transmission technologies used in modern networks.

Next comes digital modulation, which is all about how analog signals are converted into digital bits and back again. After that we will look at multiplexing schemes, exploring how multiple conversations can be put on the same transmission medium at the same time without interfering with one another.

Finally, we will look at three examples of communication systems used in practice for wide area computer networks: the (fixed) telephone system, the mobile phone system, and the cable television system. Each of these is important in practice, so we will devote a fair amount of space to each one.

2.1 THE THEORETICAL BASIS FOR DATA COMMUNICATION

Information can be transmitted on wires by varying some physical property such as voltage or current. By representing the value of this voltage or current as a single-valued function of time, $f(t)$, we can model the behavior of the signal and analyze it mathematically. This analysis is the subject of the following sections.

2.1.1 Fourier Analysis

In the early 19th century, the French mathematician Jean-Baptiste Fourier proved that any reasonably behaved periodic function, $g(t)$ with period T, can be constructed as the sum of a (possibly infinite) number of sines and cosines:

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft) \qquad (2\text{-}1)$$

where $f = 1/T$ is the fundamental frequency, a_n and b_n are the sine and cosine amplitudes of the nth **harmonics** (terms), and c is a constant. Such a decomposition is called a **Fourier series**. From the Fourier series, the function can be reconstructed. That is, if the period, T, is known and the amplitudes are given, the original function of time can be found by performing the sums of Eq. (2-1).

A data signal that has a finite duration, which all of them do, can be handled by just imagining that it repeats the entire pattern over and over forever (i.e., the interval from T to $2T$ is the same as from 0 to T, etc.).

The a_n amplitudes can be computed for any given $g(t)$ by multiplying both sides of Eq. (2-1) by $\sin(2\pi kft)$ and then integrating from 0 to T. Since

$$\int_0^T \sin(2\pi kft)\sin(2\pi nft)\,dt = \begin{cases} 0 \text{ for } k \neq n \\ T/2 \text{ for } k = n \end{cases}$$

only one term of the summation survives: a_n. The b_n summation vanishes completely. Similarly, by multiplying Eq. (2-1) by $\cos(2\pi kft)$ and integrating between 0 and T, we can derive b_n. By just integrating both sides of the equation as it stands, we can find c. The results of performing these operations are as follows:

$$a_n = \frac{2}{T}\int_0^T g(t)\sin(2\pi nft)\,dt \qquad b_n = \frac{2}{T}\int_0^T g(t)\cos(2\pi nft)\,dt \qquad c = \frac{2}{T}\int_0^T g(t)\,dt$$

2.1.2 Bandwidth-Limited Signals

The relevance of all of this to data communication is that real channels affect different frequency signals differently. Let us consider a specific example: the transmission of the ASCII character "b" encoded in an 8-bit byte. The bit pattern that is to be transmitted is 01100010. The left-hand part of Fig. 2-1(a) shows the voltage output by the transmitting computer. The Fourier analysis of this signal yields the coefficients:

$$a_n = \frac{1}{\pi n}[\cos(\pi n/4) - \cos(3\pi n/4) + \cos(6\pi n/4) - \cos(7\pi n/4)]$$

$$b_n = \frac{1}{\pi n}[\sin(3\pi n/4) - \sin(\pi n/4) + \sin(7\pi n/4) - \sin(6\pi n/4)]$$

$$c = 3/4$$

The root-mean-square amplitudes, $\sqrt{a_n^2 + b_n^2}$, for the first few terms are shown on the right-hand side of Fig. 2-1(a). These values are of interest because their squares are proportional to the energy transmitted at the corresponding frequency.

No transmission facility can transmit signals without losing some power in the process. If all the Fourier components were equally diminished, the resulting signal would be reduced in amplitude but not distorted [i.e., it would have the same nice squared-off shape as Fig. 2-1(a)]. Unfortunately, all transmission facilities diminish different Fourier components by different amounts, thus introducing distortion. Usually, for a wire, the amplitudes are transmitted mostly undiminished from 0 up to some frequency f_c [measured in cycles/sec or Hertz (Hz)], with all frequencies above this cutoff frequency attenuated. The width of the frequency range transmitted without being strongly attenuated is called the **bandwidth**. In practice, the cutoff is not really sharp, so often the quoted bandwidth is from 0 to the frequency at which the received power has fallen by half.

The bandwidth is a physical property of the transmission medium that depends on, for example, the construction, thickness, and length of a wire or fiber. Filters are often used to further limit the bandwidth of a signal. 802.11 wireless channels are allowed to use up to roughly 20 MHz, for example, so 802.11 radios filter the signal bandwidth to this size. As another example, traditional (analog) television channels occupy 6 MHz each, on a wire or over the air. This filtering lets more signals share a given region of spectrum, which improves the overall efficiency of the system. It means that the frequency range for some signals will not start at zero, but this does not matter. The bandwidth is still the width of the band of frequencies that are passed, and the information that can be carried depends only on this width and not on the starting and ending frequencies. Signals that run from 0 up to a maximum frequency are called **baseband** signals. Signals that are shifted to occupy a higher range of frequencies, as is the case for all wireless transmissions, are called **passband** signals.

Now let us consider how the signal of Fig. 2-1(a) would look if the bandwidth were so low that only the lowest frequencies were transmitted [i.e., if the function were being approximated by the first few terms of Eq. (2-1)]. Figure 2-1(b) shows the signal that results from a channel that allows only the first harmonic

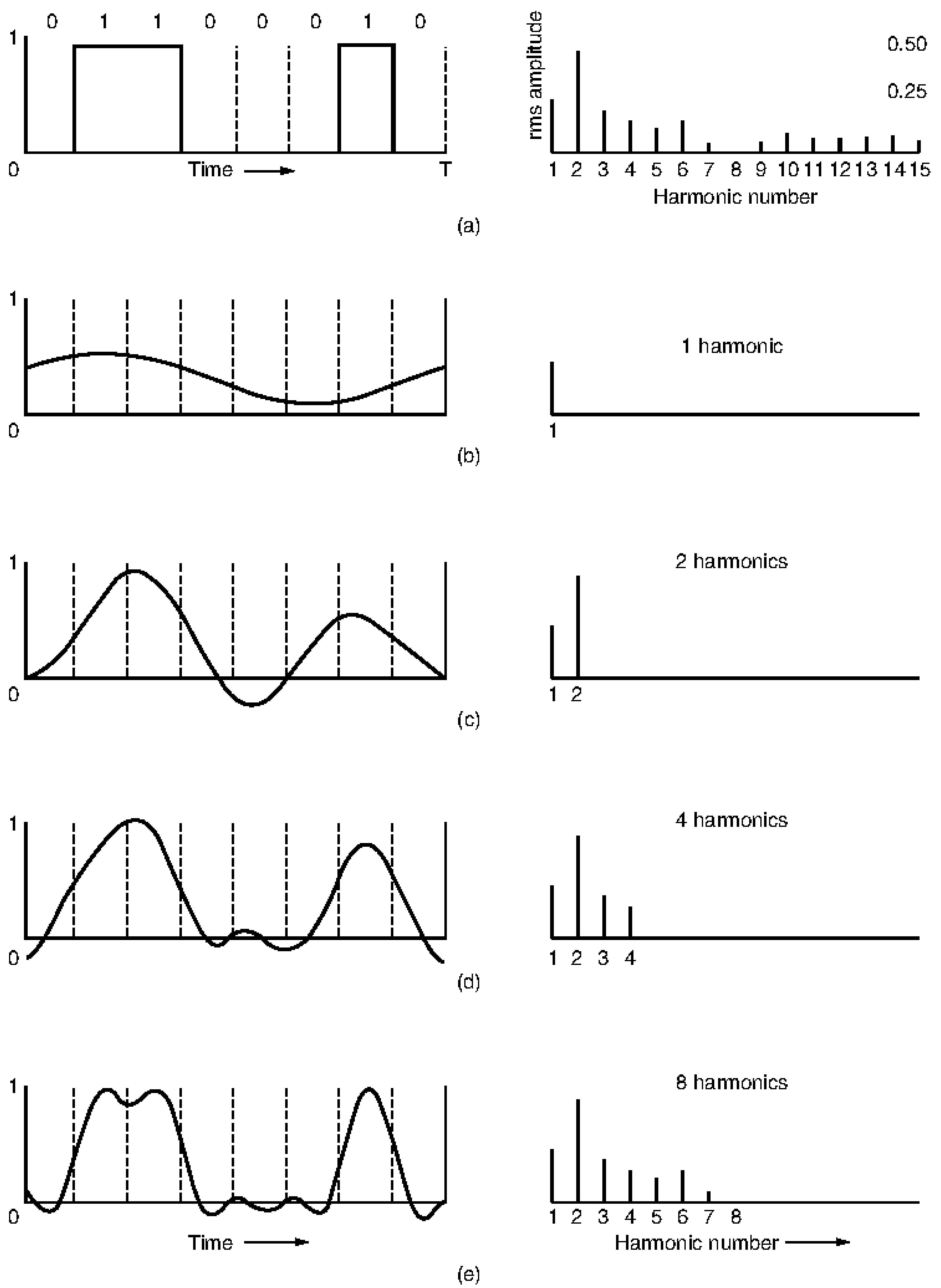


Figure 2-1. (a) A binary signal and its root-mean-square Fourier amplitudes. (b)–(e) Successive approximations to the original signal.

(the fundamental, f) to pass through. Similarly, Fig. 2-1(c)–(e) show the spectra and reconstructed functions for higher-bandwidth channels. For digital transmission, the goal is to receive a signal with just enough fidelity to reconstruct the sequence of bits that was sent. We can already do this easily in Fig. 2-1(e), so it is wasteful to use more harmonics to receive a more accurate replica.

Given a bit rate of b bits/sec, the time required to send the 8 bits in our example 1 bit at a time is $8/b$ sec, so the frequency of the first harmonic of this signal is $b/8$ Hz. An ordinary telephone line, often called a **voice-grade line**, has an artificially introduced cutoff frequency just above 3000 Hz. The presence of this restriction means that the number of the highest harmonic passed through is roughly $3000/(b/8)$, or $24{,}000/b$ (the cutoff is not sharp).

For some data rates, the numbers work out as shown in Fig. 2-2. From these numbers, it is clear that trying to send at 9600 bps over a voice-grade telephone line will transform Fig. 2-1(a) into something looking like Fig. 2-1(c), making accurate reception of the original binary bit stream tricky. It should be obvious that at data rates much higher than 38.4 kbps, there is no hope at all for *binary* signals, even if the transmission facility is completely noiseless. In other words, limiting the bandwidth limits the data rate, even for perfect channels. However, coding schemes that make use of several voltage levels do exist and can achieve higher data rates. We will discuss these later in this chapter.

Bps	T (msec)	First harmonic (Hz)	# Harmonics sent
300	26.67	37.5	80
600	13.33	75	40
1200	6.67	150	20
2400	3.33	300	10
4800	1.67	600	5
9600	0.83	1200	2
19200	0.42	2400	1
38400	0.21	4800	0

Figure 2-2. Relation between data rate and harmonics for our example.

There is much confusion about bandwidth because it means different things to electrical engineers and to computer scientists. To electrical engineers, (analog) bandwidth is (as we have described above) a quantity measured in Hz. To computer scientists, (digital) bandwidth is the maximum data rate of a channel, a quantity measured in bits/sec. That data rate is the end result of using the analog bandwidth of a physical channel for digital transmission, and the two are related, as we discuss next. In this book, it will be clear from the context whether we mean analog bandwidth (Hz) or digital bandwidth (bits/sec).

2.1.3 The Maximum Data Rate of a Channel

As early as 1924, an AT&T engineer, Henry Nyquist, realized that even a perfect channel has a finite transmission capacity. He derived an equation expressing the maximum data rate for a finite-bandwidth noiseless channel. In 1948, Claude Shannon carried Nyquist's work further and extended it to the case of a channel subject to random (that is, thermodynamic) noise (Shannon, 1948). This paper is the most important paper in all of information theory. We will just briefly summarize their now classical results here.

Nyquist proved that if an arbitrary signal has been run through a low-pass filter of bandwidth B, the filtered signal can be completely reconstructed by making only $2B$ (exact) samples per second. Sampling the line faster than $2B$ times per second is pointless because the higher-frequency components that such sampling could recover have already been filtered out. If the signal consists of V discrete levels, Nyquist's theorem states:

$$\text{maximum data rate} = 2B \log_2 V \text{ bits/sec} \tag{2-2}$$

For example, a noiseless 3-kHz channel cannot transmit binary (i.e., two-level) signals at a rate exceeding 6000 bps.

So far we have considered only noiseless channels. If random noise is present, the situation deteriorates rapidly. And there is always random (thermal) noise present due to the motion of the molecules in the system. The amount of thermal noise present is measured by the ratio of the signal power to the noise power, called the **SNR** (**Signal-to-Noise Ratio**). If we denote the signal power by S and the noise power by N, the signal-to-noise ratio is S/N. Usually, the ratio is expressed on a log scale as the quantity $10 \log_{10} S/N$ because it can vary over a tremendous range. The units of this log scale are called **decibels** (**dB**), with "deci" meaning 10 and "bel" chosen to honor Alexander Graham Bell, who invented the telephone. An S/N ratio of 10 is 10 dB, a ratio of 100 is 20 dB, a ratio of 1000 is 30 dB, and so on. The manufacturers of stereo amplifiers often characterize the bandwidth (frequency range) over which their products are linear by giving the 3-dB frequency on each end. These are the points at which the amplification factor has been approximately halved (because $10 \log_{10} 0.5 \approx -3$).

Shannon's major result is that the maximum data rate or **capacity** of a noisy channel whose bandwidth is B Hz and whose signal-to-noise ratio is S/N, is given by:

$$\text{maximum number of bits/sec} = B \log_2 (1 + S/N) \tag{2-3}$$

This tells us the best capacities that real channels can have. For example, ADSL (Asymmetric Digital Subscriber Line), which provides Internet access over normal telephone lines, uses a bandwidth of around 1 MHz. The SNR depends strongly on the distance of the home from the telephone exchange, and an SNR of around 40 dB for short lines of 1 to 2 km is very good. With these characteristics, the channel can never transmit much more than 13 Mbps, no matter how many or how few signal levels are used and no matter how often or how infrequently samples are taken. In practice, ADSL is specified up to 12 Mbps, though users often see lower rates. This data rate is actually very good, with over 60 years of communications techniques having greatly reduced the gap between the Shannon capacity and the capacity of real systems.

Shannon's result was derived from information-theory arguments and applies to any channel subject to thermal noise. Counterexamples should be treated in the same category as perpetual motion machines. For ADSL to exceed 13 Mbps, it must either improve the SNR (for example by inserting digital repeaters in the lines closer to the customers) or use more bandwidth, as is done with the evolution to ASDL2+.

2.2 GUIDED TRANSMISSION MEDIA

The purpose of the physical layer is to transport bits from one machine to another. Various physical media can be used for the actual transmission. Each one has its own niche in terms of bandwidth, delay, cost, and ease of installation and maintenance. Media are roughly grouped into guided media, such as copper wire and fiber optics, and unguided media, such as terrestrial wireless, satellite, and lasers through the air. We will look at guided media in this section, and unguided media in the next sections.

2.2.1 Magnetic Media

One of the most common ways to transport data from one computer to another is to write them onto magnetic tape or removable media (e.g., recordable DVDs), physically transport the tape or disks to the destination machine, and read them back in again. Although this method is not as sophisticated as using a geosynchronous communication satellite, it is often more cost effective, especially for applications in which high bandwidth or cost per bit transported is the key factor.

A simple calculation will make this point clear. An industry-standard Ultrium tape can hold 800 gigabytes. A box $60 \times 60 \times 60$ cm can hold about 1000 of these tapes, for a total capacity of 800 terabytes, or 6400 terabits (6.4 petabits). A box of tapes can be delivered anywhere in the United States in 24 hours by Federal Express and other companies. The effective bandwidth of this transmission is 6400 terabits/86,400 sec, or a bit over 70 Gbps. If the destination is only an hour away by road, the bandwidth is increased to over 1700 Gbps. No computer network can even approach this. Of course, networks are getting faster, but tape densities are increasing, too.

If we now look at cost, we get a similar picture. The cost of an Ultrium tape is around \$40 when bought in bulk. A tape can be reused at least 10 times, so the tape cost is maybe $4000 per box per usage. Add to this another $1000 for shipping (probably much less), and we have a cost of roughly $5000 to ship 800 TB. This amounts to shipping a gigabyte for a little over half a cent. No network can beat that. The moral of the story is:

Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.

2.2.2 Twisted Pairs

Although the bandwidth characteristics of magnetic tape are excellent, the delay characteristics are poor. Transmission time is measured in minutes or hours, not milliseconds. For many applications an online connection is needed. One of the oldest and still most common transmission media is **twisted pair**. A twisted pair consists of two insulated copper wires, typically about 1 mm thick. The wires are twisted together in a helical form, just like a DNA molecule. Twisting is done because two parallel wires constitute a fine antenna. When the wires are twisted, the waves from different twists cancel out, so the wire radiates less effectively. A signal is usually carried as the difference in voltage between the two wires in the pair. This provides better immunity to external noise because the noise tends to affect both wires the same, leaving the differential unchanged.

The most common application of the twisted pair is the telephone system. Nearly all telephones are connected to the telephone company (telco) office by a twisted pair. Both telephone calls and ADSL Internet access run over these lines. Twisted pairs can run several kilometers without amplification, but for longer distances the signal becomes too attenuated and repeaters are needed. When many twisted pairs run in parallel for a substantial distance, such as all the wires coming from an apartment building to the telephone company office, they are bundled together and encased in a protective sheath. The pairs in these bundles would interfere with one another if it were not for the twisting. In parts of the world where telephone lines run on poles above ground, it is common to see bundles several centimeters in diameter.

Twisted pairs can be used for transmitting either analog or digital information. The bandwidth depends on the thickness of the wire and the distance traveled, but several megabits/sec can be achieved for a few kilometers in many cases. Due to their adequate performance and low cost, twisted pairs are widely used and are likely to remain so for years to come.

Twisted-pair cabling comes in several varieties. The garden variety deployed in many office buildings is called **Category 5** cabling, or "Cat 5." A category 5 twisted pair consists of two insulated wires gently twisted together. Four such pairs are typically grouped in a plastic sheath to protect the wires and keep them together. This arrangement is shown in Fig. 2-3.

Different LAN standards may use the twisted pairs differently. For example, 100-Mbps Ethernet uses two (out of the four) pairs, one pair for each direction.

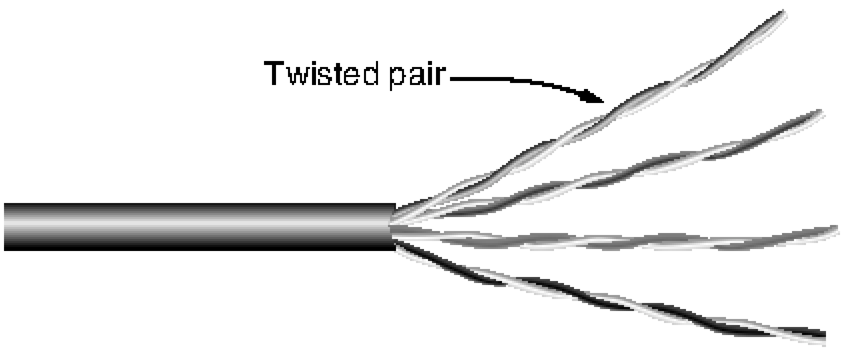


Figure 2-3. Category 5 UTP cable with four twisted pairs.

To reach higher speeds, 1-Gbps Ethernet uses all four pairs in both directions simultaneously; this requires the receiver to factor out the signal that is transmitted locally.

Some general terminology is now in order. Links that can be used in both directions at the same time, like a two-lane road, are called **full-duplex** links. In contrast, links that can be used in either direction, but only one way at a time, like a single-track railroad line. are called **half-duplex** links. A third category consists of links that allow traffic in only one direction, like a one-way street. They are called **simplex** links.

Returning to twisted pair, Cat 5 replaced earlier **Category 3** cables with a similar cable that uses the same connector, but has more twists per meter. More twists result in less crosstalk and a better-quality signal over longer distances, making the cables more suitable for high-speed computer communication, especially 100-Mbps and 1-Gbps Ethernet LANs.

New wiring is more likely to be **Category 6** or even **Category 7**. These categories has more stringent specifications to handle signals with greater bandwidths. Some cables in Category 6 and above are rated for signals of 500 MHz and can support the 10-Gbps links that will soon be deployed.

Through Category 6, these wiring types are referred to as **UTP** (**Unshielded Twisted Pair**) as they consist simply of wires and insulators. In contrast to these, Category 7 cables have shielding on the individual twisted pairs, as well as around the entire cable (but inside the plastic protective sheath). Shielding reduces the susceptibility to external interference and crosstalk with other nearby cables to meet demanding performance specifications. The cables are reminiscent of the high-quality, but bulky and expensive shielded twisted pair cables that IBM introduced in the early 1980s, but which did not prove popular outside of IBM installations. Evidently, it is time to try again.

2.2.3 Coaxial Cable

Another common transmission medium is the **coaxial cable** (known to its many friends as just “coax” and pronounced “co-ax”). It has better shielding and greater bandwidth than unshielded twisted pairs, so it can span longer distances at higher speeds. Two kinds of coaxial cable are widely used. One kind, 50-ohm cable, is commonly used when it is intended for digital transmission from the start. The other kind, 75-ohm cable, is commonly used for analog transmission and cable television. This distinction is based on historical, rather than technical, factors (e.g., early dipole antennas had an impedance of 300 ohms, and it was easy to use existing 4:1 impedance-matching transformers). Starting in the mid-1990s, cable TV operators began to provide Internet access over cable, which has made 75-ohm cable more important for data communication.

A coaxial cable consists of a stiff copper wire as the core, surrounded by an insulating material. The insulator is encased by a cylindrical conductor, often as a closely woven braided mesh. The outer conductor is covered in a protective plastic sheath. A cutaway view of a coaxial cable is shown in Fig. 2-4.

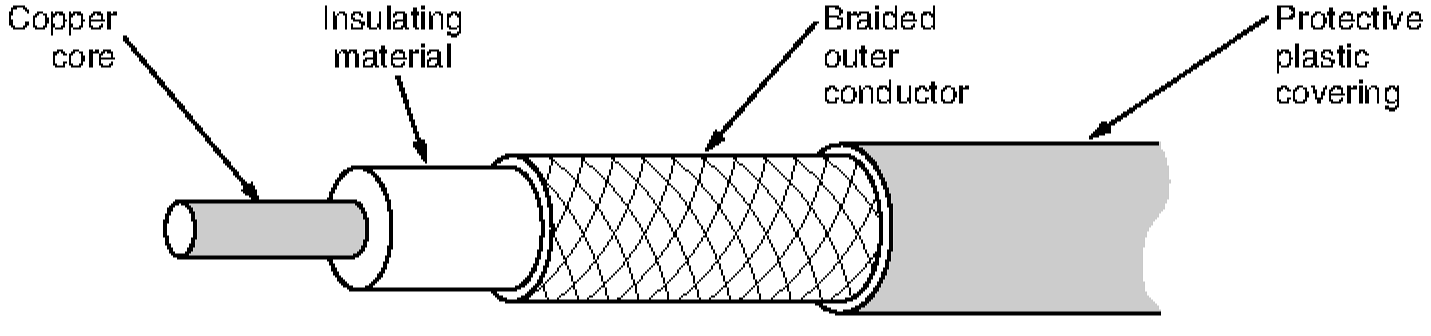


Figure 2-4. A coaxial cable.

The construction and shielding of the coaxial cable give it a good combination of high bandwidth and excellent noise immunity. The bandwidth possible depends on the cable quality and length. Modern cables have a bandwidth of up to a few GHz. Coaxial cables used to be widely used within the telephone system for long-distance lines but have now largely been replaced by fiber optics on long-haul routes. Coax is still widely used for cable television and metropolitan area networks, however.

2.2.4 Power Lines

The telephone and cable television networks are not the only sources of wiring that can be reused for data communication. There is a yet more common kind of wiring: electrical power lines. Power lines deliver electrical power to houses, and electrical wiring within houses distributes the power to electrical outlets.

The use of power lines for data communication is an old idea. Power lines have been used by electricity companies for low-rate communication such as remote metering for many years, as well in the home to control devices (e.g., the X10 standard). In recent years there has been renewed interest in high-rate communication over these lines, both inside the home as a LAN and outside the home for broadband Internet access. We will concentrate on the most common scenario: using electrical wires inside the home.

The convenience of using power lines for networking should be clear. Simply plug a TV and a receiver into the wall, which you must do anyway because they need power, and they can send and receive movies over the electrical wiring. This configuration is shown in Fig. 2-5. There is no other plug or radio. The data signal is superimposed on the low-frequency power signal (on the active or "hot" wire) as both signals use the wiring at the same time.

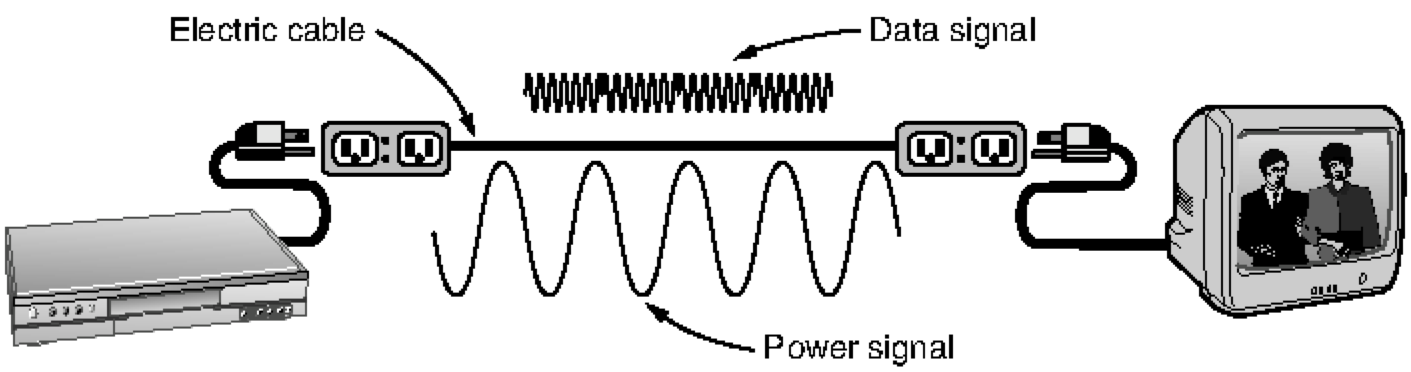


Figure 2-5. A network that uses household electrical wiring.

The difficulty with using household electrical wiring for a network is that it was designed to distribute power signals. This task is quite different than distributing data signals, at which household wiring does a horrible job. Electrical signals are sent at 50–60 Hz and the wiring attenuates the much higher frequency (MHz) signals needed for high-rate data communication. The electrical properties of the wiring vary from one house to the next and change as appliances are turned on and off, which causes data signals to bounce around the wiring. Transient currents when appliances switch on and off create electrical noise over a wide range of frequencies. And without the careful twisting of twisted pairs, electrical wiring acts as a fine antenna, picking up external signals and radiating signals of its own. This behavior means that to meet regulatory requirements, the data signal must exclude licensed frequencies such as the amateur radio bands.

Despite these difficulties, it is practical to send at least 100 Mbps over typical household electrical wiring by using communication schemes that resist impaired frequencies and bursts of errors. Many products use various proprietary standards for power-line networking, so international standards are actively under development.

2.2.5 Fiber Optics

Many people in the computer industry take enormous pride in how fast computer technology is improving as it follows Moore's law, which predicts a doubling of the number of transistors per chip roughly every two years (Schaller, 1997). The original (1981) IBM PC ran at a clock speed of 4.77 MHz. Twenty-eight years later, PCs could run a four-core CPU at 3 GHz. This increase is a gain of a factor of around 2500, or 16 per decade. Impressive.

In the same period, wide area communication links went from 45 Mbps (a T3 line in the telephone system) to 100 Gbps (a modern long distance line). This gain is similarly impressive, more than a factor of 2000 and close to 16 per decade, while at the same time the error rate went from 10^{-5} per bit to almost zero. Furthermore, single CPUs are beginning to approach physical limits, which is why it is now the number of CPUs that is being increased per chip. In contrast, the achievable bandwidth with fiber technology is in excess of 50,000 Gbps (50 Tbps) and we are nowhere near reaching these limits. The current practical limit of around 100 Gbps is due to our inability to convert between electrical and optical signals any faster. To build higher-capacity links, many channels are simply carried in parallel over a single fiber.

In this section we will study fiber optics to learn how that transmission technology works. In the ongoing race between computing and communication, communication may yet win because of fiber optic networks. The implication of this would be essentially infinite bandwidth and a new conventional wisdom that computers are hopelessly slow so that networks should try to avoid computation at all costs, no matter how much bandwidth that wastes. This change will take a while to sink in to a generation of computer scientists and engineers taught to think in terms of the low Shannon limits imposed by copper.

Of course, this scenario does not tell the whole story because it does not include cost. The cost to install fiber over the last mile to reach consumers and bypass the low bandwidth of wires and limited availability of spectrum is tremendous. It also costs more energy to move bits than to compute. We may always have islands of inequities where either computation or communication is essentially free. For example, at the edge of the Internet we throw computation and storage at the problem of compressing and caching content, all to make better use of Internet access links. Within the Internet, we may do the reverse, with companies such as Google moving huge amounts of data across the network to where it is cheaper to store or compute on it.

Fiber optics are used for long-haul transmission in network backbones, high-speed LANs (although so far, copper has always managed catch up eventually), and high-speed Internet access such as **FttH** (**Fiber to the Home**). An optical transmission system has three key components: the light source, the transmission medium, and the detector. Conventionally, a pulse of light indicates a 1 bit and the absence of light indicates a 0 bit. The transmission medium is an ultra-thin fiber of glass. The detector generates an electrical pulse when light falls on it. By attaching a light source to one end of an optical fiber and a detector to the other, we have a unidirectional data transmission system that accepts an electrical signal, converts and transmits it by light pulses, and then reconverts the output to an electrical signal at the receiving end.

This transmission system would leak light and be useless in practice were it not for an interesting principle of physics. When a light ray passes from one medium to another—for example, from fused silica to air—the ray is refracted (bent) at the silica/air boundary, as shown in Fig. 2-6(a). Here we see a light ray incident on the boundary at an angle α_1 emerging at an angle β_1. The amount of refraction depends on the properties of the two media (in particular, their indices of refraction). For angles of incidence above a certain critical value, the light is refracted back into the silica; none of it escapes into the air. Thus, a light ray incident at or above the critical angle is trapped inside the fiber, as shown in Fig. 2-6(b), and can propagate for many kilometers with virtually no loss.

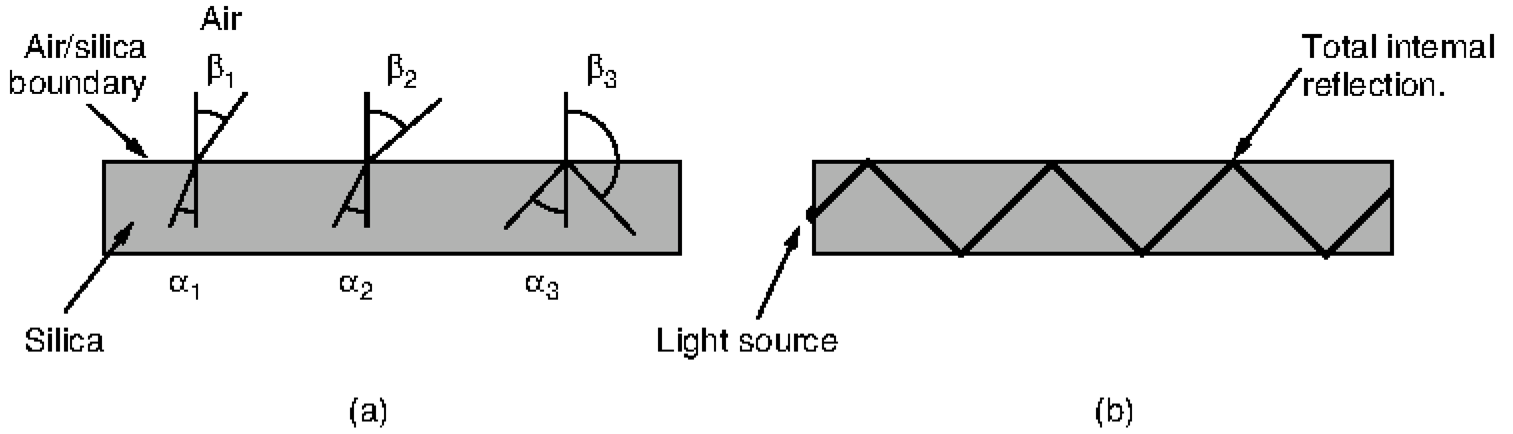


Figure 2-6. (a) Three examples of a light ray from inside a silica fiber impinging on the air/silica boundary at different angles. (b) Light trapped by total internal reflection.

The sketch of Fig. 2-6(b) shows only one trapped ray, but since any light ray incident on the boundary above the critical angle will be reflected internally, many different rays will be bouncing around at different angles. Each ray is said to have a different mode, so a fiber having this property is called a **multimode fiber**.

However, if the fiber's diameter is reduced to a few wavelengths of light the fiber acts like a wave guide and the light can propagate only in a straight line, without bouncing, yielding a **single-mode fiber**. Single-mode fibers are more expensive but are widely used for longer distances. Currently available single-mode fibers can transmit data at 100 Gbps for 100 km without amplification. Even higher data rates have been achieved in the laboratory for shorter distances.

Transmission of Light Through Fiber

Optical fibers are made of glass, which, in turn, is made from sand, an inexpensive raw material available in unlimited amounts. Glassmaking was known to the ancient Egyptians, but their glass had to be no more than 1 mm thick or the light could not shine through. Glass transparent enough to be useful for windows was developed during the Renaissance. The glass used for modern optical fibers is so transparent that if the oceans were full of it instead of water, the seabed would be as visible from the surface as the ground is from an airplane on a clear day.

The attenuation of light through glass depends on the wavelength of the light (as well as on some physical properties of the glass). It is defined as the ratio of input to output signal power. For the kind of glass used in fibers, the attenuation is shown in Fig. 2-7 in units of decibels per linear kilometer of fiber. For example, a factor of two loss of signal power gives an attenuation of $10 \log_{10} 2 = 3$ dB. The figure shows the near-infrared part of the spectrum, which is what is used in practice. Visible light has slightly shorter wavelengths, from 0.4 to 0.7 microns. (1 micron is 10^{-6} meters.) The true metric purist would refer to these wavelengths as 400 nm to 700 nm, but we will stick with traditional usage.

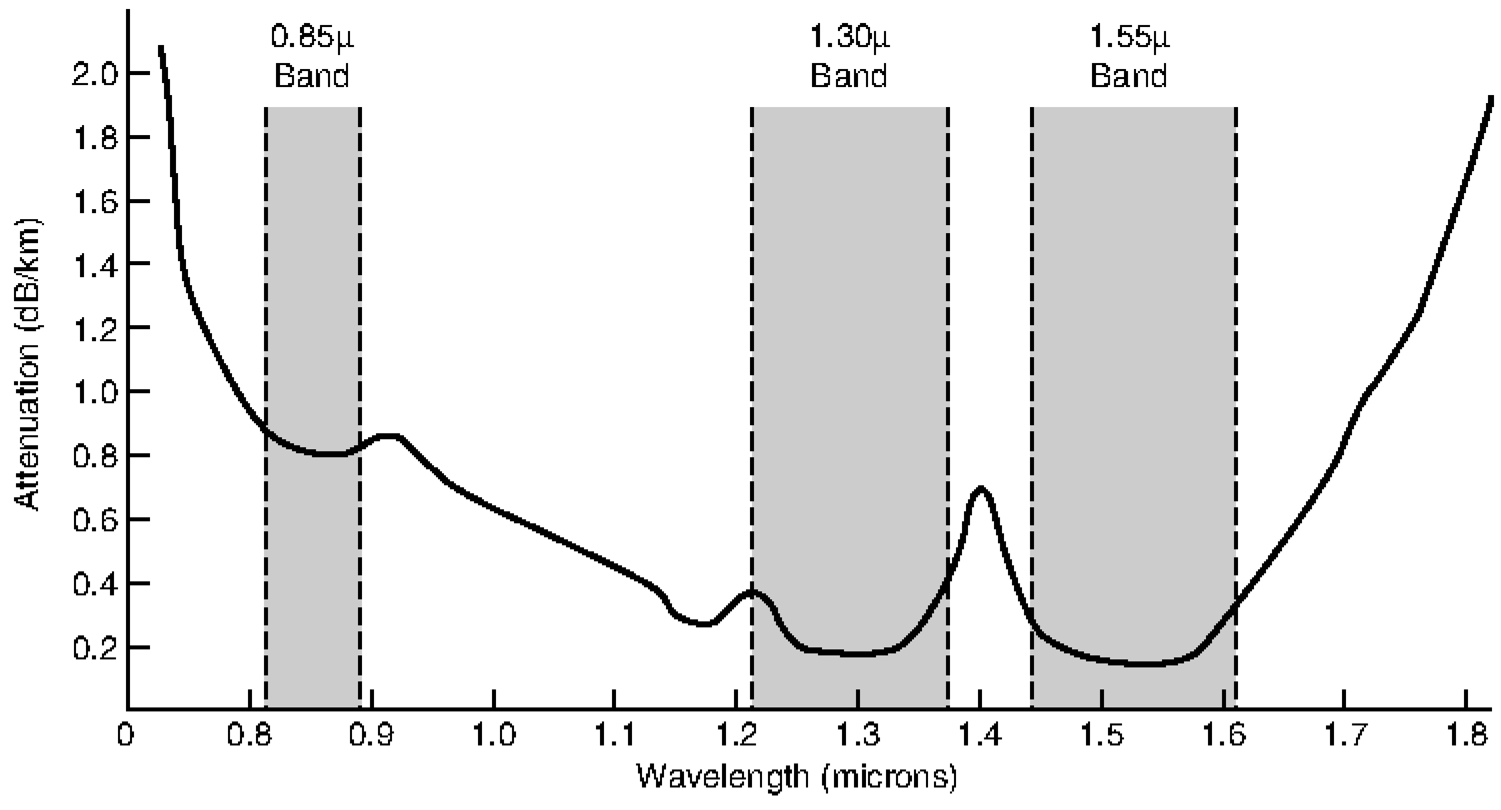


Figure 2-7. Attenuation of light through fiber in the infrared region.

Three wavelength bands are most commonly used at present for optical communication. They are centered at 0.85, 1.30, and 1.55 microns, respectively. All three bands are 25,000 to 30,000 GHz wide. The 0.85-micron band was used first. It has higher attenuation and so is used for shorter distances, but at that wavelength the lasers and electronics could be made from the same material (gallium arsenide). The last two bands have good attenuation properties (less than 5% loss per kilometer). The 1.55-micron band is now widely used with erbium-doped amplifiers that work directly in the optical domain.

Light pulses sent down a fiber spread out in length as they propagate. This spreading is called **chromatic dispersion**. The amount of it is wavelength dependent. One way to keep these spread-out pulses from overlapping is to increase the distance between them, but this can be done only by reducing the signaling rate. Fortunately, it has been discovered that making the pulses in a special shape related to the reciprocal of the hyperbolic cosine causes nearly all the dispersion effects cancel out, so it is possible to send pulses for thousands of kilometers without appreciable shape distortion. These pulses are called **solitons**. A considerable amount of research is going on to take solitons out of the lab and into the field.

Fiber Cables

Fiber optic cables are similar to coax, except without the braid. Figure 2-8(a) shows a single fiber viewed from the side. At the center is the glass core through which the light propagates. In multimode fibers, the core is typically 50 microns in diameter, about the thickness of a human hair. In single-mode fibers, the core is 8 to 10 microns.

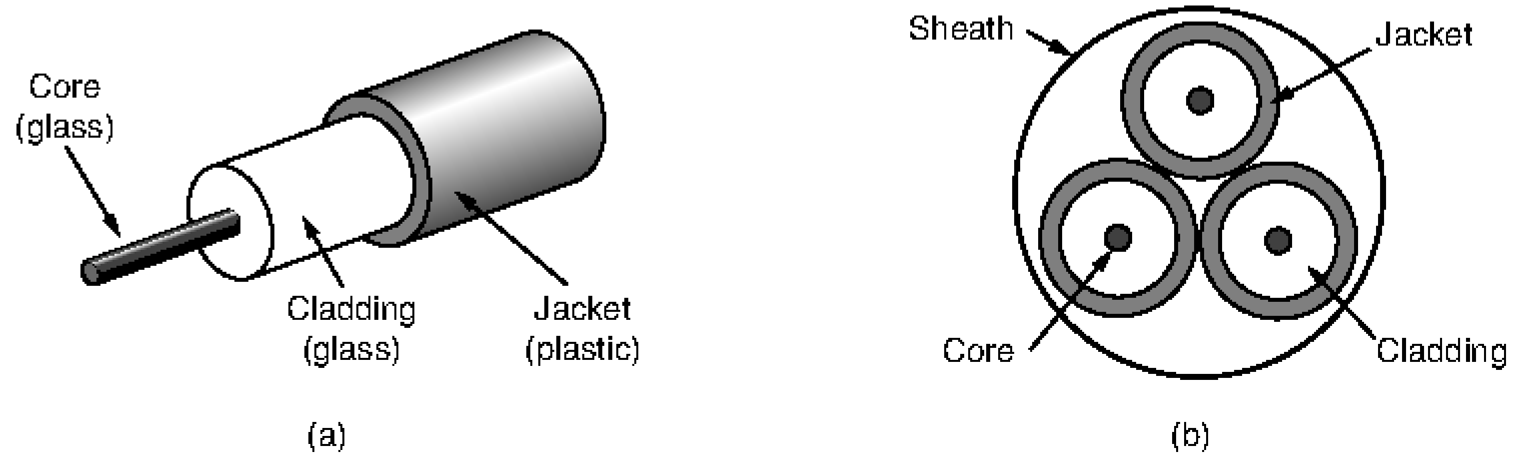


Figure 2-8. (a) Side view of a single fiber. (b) End view of a sheath with three fibers.

The core is surrounded by a glass cladding with a lower index of refraction than the core, to keep all the light in the core. Next comes a thin plastic jacket to protect the cladding. Fibers are typically grouped in bundles, protected by an outer sheath. Figure 2-8(b) shows a sheath with three fibers.

Terrestrial fiber sheaths are normally laid in the ground within a meter of the surface, where they are occasionally subject to attacks by backhoes or gophers. Near the shore, transoceanic fiber sheaths are buried in trenches by a kind of seaplow. In deep water, they just lie on the bottom, where they can be snagged by fishing trawlers or attacked by giant squid.

Fibers can be connected in three different ways. First, they can terminate in connectors and be plugged into fiber sockets. Connectors lose about 10 to 20% of the light, but they make it easy to reconfigure systems.

Second, they can be spliced mechanically. Mechanical splices just lay the two carefully cut ends next to each other in a special sleeve and clamp them in place. Alignment can be improved by passing light through the junction and then making small adjustments to maximize the signal. Mechanical splices take trained personnel about 5 minutes and result in a 10% light loss.

Third, two pieces of fiber can be fused (melted) to form a solid connection. A fusion splice is almost as good as a single drawn fiber, but even here, a small amount of attenuation occurs.

For all three kinds of splices, reflections can occur at the point of the splice, and the reflected energy can interfere with the signal.

Two kinds of light sources are typically used to do the signaling. These are **LEDs** (**Light Emitting Diodes**) and semiconductor lasers. They have different properties, as shown in Fig. 2-9. They can be tuned in wavelength by inserting Fabry-Perot or Mach-Zehnder interferometers between the source and the fiber. Fabry-Perot interferometers are simple resonant cavities consisting of two parallel mirrors. The light is incident perpendicular to the mirrors. The length of the cavity selects out those wavelengths that fit inside an integral number of times. Mach-Zehnder interferometers separate the light into two beams. The two beams travel slightly different distances. They are recombined at the end and are in phase for only certain wavelengths.

Item	LED	Semiconductor laser
Data rate	Low	High
Fiber type	Multi-mode	Multi-mode or single-mode
Distance	Short	Long
Lifetime	Long life	Short life
Temperature sensitivity	Minor	Substantial
Cost	Low cost	Expensive

Figure 2-9. A comparison of semiconductor diodes and LEDs as light sources.

The receiving end of an optical fiber consists of a photodiode, which gives off an electrical pulse when struck by light. The response time of photodiodes, which convert the signal from the optical to the electrical domain, limits data rates to about 100 Gbps. Thermal noise is also an issue, so a pulse of light must carry enough energy to be detected. By making the pulses powerful enough, the error rate can be made arbitrarily small.

Comparison of Fiber Optics and Copper Wire

It is instructive to compare fiber to copper. Fiber has many advantages. To start with, it can handle much higher bandwidths than copper. This alone would require its use in high-end networks. Due to the low attenuation, repeaters are needed only about every 50 km on long lines, versus about every 5 km for copper, resulting in a big cost saving. Fiber also has the advantage of not being affected by power surges, electromagnetic interference, or power failures. Nor is it affected by corrosive chemicals in the air, important for harsh factory environments.

Oddly enough, telephone companies like fiber for a different reason: it is thin and lightweight. Many existing cable ducts are completely full, so there is no room to add new capacity. Removing all the copper and replacing it with fiber empties the ducts, and the copper has excellent resale value to copper refiners who see it as very high-grade ore. Also, fiber is much lighter than copper. One thousand twisted pairs 1 km long weigh 8000 kg. Two fibers have more capacity and weigh only 100 kg, which reduces the need for expensive mechanical support systems that must be maintained. For new routes, fiber wins hands down due to its much lower installation cost. Finally, fibers do not leak light and are difficult to tap. These properties give fiber good security against potential wiretappers.

On the downside, fiber is a less familiar technology requiring skills not all engineers have, and fibers can be damaged easily by being bent too much. Since optical transmission is inherently unidirectional, two-way communication requires either two fibers or two frequency bands on one fiber. Finally, fiber interfaces cost more than electrical interfaces. Nevertheless, the future of all fixed data communication over more than short distances is clearly with fiber. For a discussion of all aspects of fiber optics and their networks, see Hecht (2005).

2.3 WIRELESS TRANSMISSION

Our age has given rise to information junkies: people who need to be online all the time. For these mobile users, twisted pair, coax, and fiber optics are of no use. They need to get their "hits" of data for their laptop, notebook, shirt pocket, palmtop, or wristwatch computers without being tethered to the terrestrial communication infrastructure. For these users, wireless communication is the answer.

In the following sections, we will look at wireless communication in general. It has many other important applications besides providing connectivity to users who want to surf the Web from the beach. Wireless has advantages for even fixed devices in some circumstances. For example, if running a fiber to a building is difficult due to the terrain (mountains, jungles, swamps, etc.), wireless may be better. It is noteworthy that modern wireless digital communication began in the Hawaiian Islands, where large chunks of Pacific Ocean separated the users from their computer center and the telephone system was inadequate.

2.3.1 The Electromagnetic Spectrum

When electrons move, they create electromagnetic waves that can propagate through space (even in a vacuum). These waves were predicted by the British physicist James Clerk Maxwell in 1865 and first observed by the German physicist Heinrich Hertz in 1887. The number of oscillations per second of a wave is called its **frequency**, f, and is measured in **Hz** (in honor of Heinrich Hertz). The distance between two consecutive maxima (or minima) is called the **wavelength**, which is universally designated by the Greek letter λ (lambda).

When an antenna of the appropriate size is attached to an electrical circuit, the electromagnetic waves can be broadcast efficiently and received by a receiver some distance away. All wireless communication is based on this principle.

In a vacuum, all electromagnetic waves travel at the same speed, no matter what their frequency. This speed, usually called the **speed of light**, c, is approximately 3×10^8 m/sec, or about 1 foot (30 cm) per nanosecond. (A case could be made for redefining the foot as the distance light travels in a vacuum in 1 nsec rather than basing it on the shoe size of some long-dead king.) In copper or fiber the speed slows to about 2/3 of this value and becomes slightly frequency dependent. The speed of light is the ultimate speed limit. No object or signal can ever move faster than it.

The fundamental relation between f, λ, and c (in a vacuum) is

$$\lambda f = c \tag{2-4}$$

Since c is a constant, if we know f, we can find λ, and vice versa. As a rule of thumb, when λ is in meters and f is in MHz, $\lambda f \approx 300$. For example, 100-MHz waves are about 3 meters long, 1000-MHz waves are 0.3 meters long, and 0.1-meter waves have a frequency of 3000 MHz.

The electromagnetic spectrum is shown in Fig. 2-10. The radio, microwave, infrared, and visible light portions of the spectrum can all be used for transmitting information by modulating the amplitude, frequency, or phase of the waves. Ultraviolet light, X-rays, and gamma rays would be even better, due to their higher frequencies, but they are hard to produce and modulate, do not propagate well through buildings, and are dangerous to living things. The bands listed at the bottom of Fig. 2-10 are the official ITU (International Telecommunication Union) names and are based on the wavelengths, so the LF band goes from 1 km to 10 km (approximately 30 kHz to 300 kHz). The terms LF, MF, and HF refer to Low, Medium, and High Frequency, respectively. Clearly, when the names were assigned nobody expected to go above 10 MHz, so the higher bands were later named the Very, Ultra, Super, Extremely, and Tremendously High Frequency bands. Beyond that there are no names, but Incredibly, Astonishingly, and Prodigiously High Frequency (IHF, AHF, and PHF) would sound nice.

We know from Shannon [Eq. (2-3)] that the amount of information that a signal such as an electromagnetic wave can carry depends on the received power and is proportional to its bandwidth. From Fig. 2-10 it should now be obvious why networking people like fiber optics so much. Many GHz of bandwidth are available to tap for data transmission in the microwave band, and even more in fiber because it is further to the right in our logarithmic scale. As an example, consider the 1.30-micron band of Fig. 2-7, which has a width of 0.17 microns. If we use

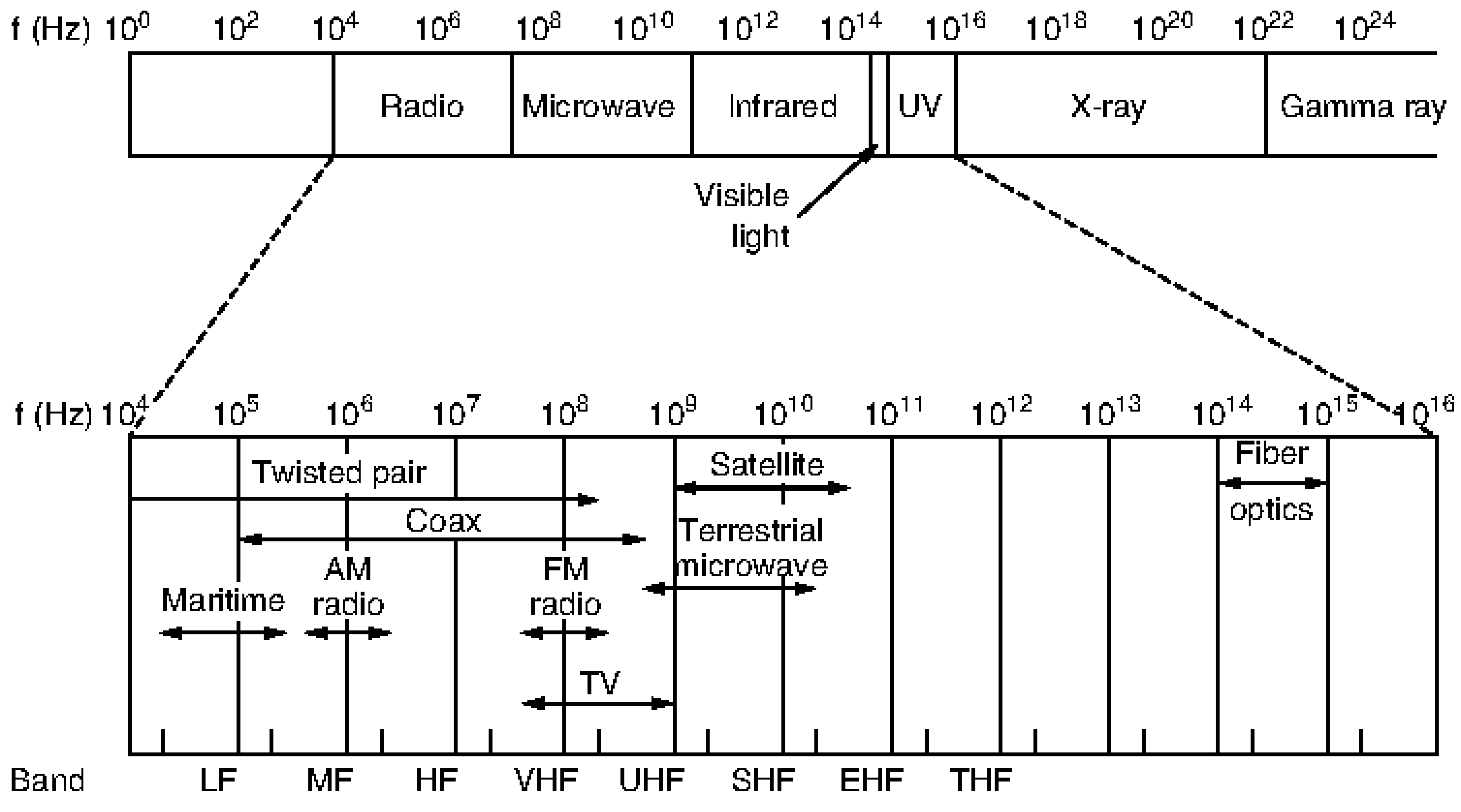


Figure 2-10. The electromagnetic spectrum and its uses for communication.

Eq. (2-4) to find the start and end frequencies from the start and end wavelengths, we find the frequency range to be about 30,000 GHz. With a reasonable signal-to-noise ratio of 10 dB, this is 300 Tbps.

Most transmissions use a relatively narrow frequency band (i.e., $\Delta f / f \ll 1$). They concentrate their signals in this narrow band to use the spectrum efficiently and obtain reasonable data rates by transmitting with enough power. However, in some cases, a wider band is used, with three variations. In **frequency hopping spread spectrum**, the transmitter hops from frequency to frequency hundreds of times per second. It is popular for military communication because it makes transmissions hard to detect and next to impossible to jam. It also offers good resistance to multipath fading and narrowband interference because the receiver will not be stuck on an impaired frequency for long enough to shut down communication. This robustness makes it useful for crowded parts of the spectrum, such as the ISM bands we will describe shortly. This technique is used commercially, for example, in Bluetooth and older versions of 802.11.

As a curious footnote, the technique was coinvented by the Austrian-born sex goddess Hedy Lamarr, the first woman to appear nude in a motion picture (the 1933 Czech film *Extase*). Her first husband was an armaments manufacturer who told her how easy it was to block the radio signals then used to control torpedoes. When she discovered that he was selling weapons to Hitler, she was horrified, disguised herself as a maid to escape him, and fled to Hollywood to continue her career as a movie actress. In her spare time, she invented frequency hopping to help the Allied war effort. Her scheme used 88 frequencies, the number of keys (and frequencies) on the piano. For their invention, she and her friend, the musical composer George Antheil, received U.S. patent 2,292,387. However, they were unable to convince the U.S. Navy that their invention had any practical use and never received any royalties. Only years after the patent expired did it become popular.

A second form of spread spectrum, **direct sequence spread spectrum**, uses a code sequence to spread the data signal over a wider frequency band. It is widely used commercially as a spectrally efficient way to let multiple signals share the same frequency band. These signals can be given different codes, a method called **CDMA** (**Code Division Multiple Access**) that we will return to later in this chapter. This method is shown in contrast with frequency hopping in Fig. 2-11. It forms the basis of 3G mobile phone networks and is also used in GPS (Global Positioning System). Even without different codes, direct sequence spread spectrum, like frequency hopping spread spectrum, can tolerate narrowband interference and multipath fading because only a fraction of the desired signal is lost. It is used in this role in older 802.11b wireless LANs. For a fascinating and detailed history of spread spectrum communication, see Scholtz (1982).

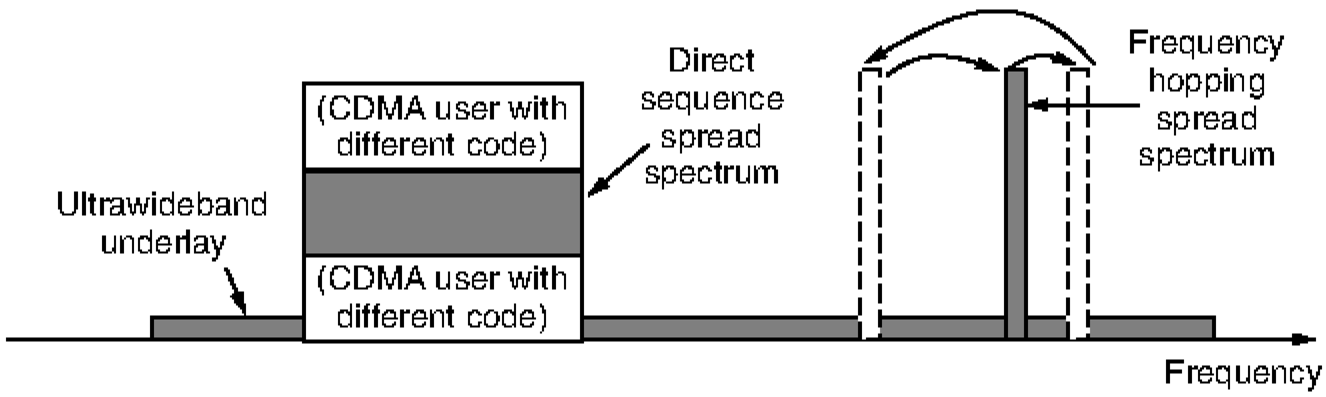


Figure 2-11. Spread spectrum and ultra-wideband (UWB) communication.

A third method of communication with a wider band is **UWB** (**Ultra-WideBand**) communication. UWB sends a series of rapid pulses, varying their positions to communicate information. The rapid transitions lead to a signal that is spread thinly over a very wide frequency band. UWB is defined as signals that have a bandwidth of at least 500 MHz or at least 20% of the center frequency of their frequency band. UWB is also shown in Fig. 2-11. With this much bandwidth, UWB has the potential to communicate at high rates. Because it is spread across a wide band of frequencies, it can tolerate a substantial amount of relatively strong interference from other narrowband signals. Just as importantly, since UWB has very little energy at any given frequency when used for short-range transmission, it does not cause harmful interference to those other narrowband radio signals. It is said to **underlay** the other signals. This peaceful coexistence has led to its application in wireless PANs that run at up to 1 Gbps, although commercial success has been mixed. It can also be used for imaging through solid objects (ground, walls, and bodies) or as part of precise location systems.

We will now discuss how the various parts of the electromagnetic spectrum of Fig. 2-11 are used, starting with radio. We will assume that all transmissions use a narrow frequency band unless otherwise stated.

2.3.2 Radio Transmission

Radio frequency (RF) waves are easy to generate, can travel long distances, and can penetrate buildings easily, so they are widely used for communication, both indoors and outdoors. Radio waves also are omnidirectional, meaning that they travel in all directions from the source, so the transmitter and receiver do not have to be carefully aligned physically.

Sometimes omnidirectional radio is good, but sometimes it is bad. In the 1970s, General Motors decided to equip all its new Cadillacs with computer-controlled antilock brakes. When the driver stepped on the brake pedal, the computer pulsed the brakes on and off instead of locking them on hard. One fine day an Ohio Highway Patrolman began using his new mobile radio to call headquarters, and suddenly the Cadillac next to him began behaving like a bucking bronco. When the officer pulled the car over, the driver claimed that he had done nothing and that the car had gone crazy.

Eventually, a pattern began to emerge: Cadillacs would sometimes go berserk, but only on major highways in Ohio and then only when the Highway Patrol was watching. For a long, long time General Motors could not understand why Cadillacs worked fine in all the other states and also on minor roads in Ohio. Only after much searching did they discover that the Cadillac's wiring made a fine antenna for the frequency used by the Ohio Highway Patrol's new radio system.

The properties of radio waves are frequency dependent. At low frequencies, radio waves pass through obstacles well, but the power falls off sharply with distance from the source—at least as fast as $1/r^2$ in air—as the signal energy is spread more thinly over a larger surface. This attenuation is called **path loss**. At high frequencies, radio waves tend to travel in straight lines and bounce off obstacles. Path loss still reduces power, though the received signal can depend strongly on reflections as well. High-frequency radio waves are also absorbed by rain and other obstacles to a larger extent than are low-frequency ones. At all frequencies, radio waves are subject to interference from motors and other electrical equipment.

It is interesting to compare the attenuation of radio waves to that of signals in guided media. With fiber, coax and twisted pair, the signal drops by the same fraction per unit distance, for example 20 dB per 100m for twisted pair. With radio, the signal drops by the same fraction as the distance doubles, for example 6 dB per doubling in free space. This behavior means that radio waves can travel long distances, and interference between users is a problem. For this reason, all governments tightly regulate the use of radio transmitters, with few notable exceptions, which are discussed later in this chapter.

In the VLF, LF, and MF bands, radio waves follow the ground, as illustrated in Fig. 2-12(a). These waves can be detected for perhaps 1000 km at the lower frequencies, less at the higher ones. AM radio broadcasting uses the MF band, which is why the ground waves from Boston AM radio stations cannot be heard easily in New York. Radio waves in these bands pass through buildings easily, which is why portable radios work indoors. The main problem with using these bands for data communication is their low bandwidth [see Eq. (2-4)].

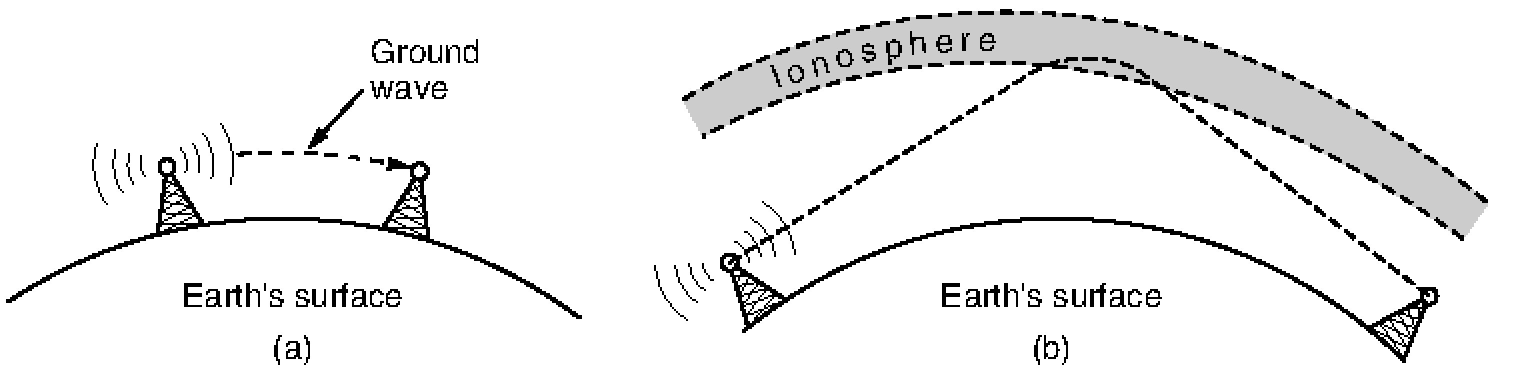


Figure 2-12. (a) In the VLF, LF, and MF bands, radio waves follow the curvature of the earth. (b) In the HF band, they bounce off the ionosphere.

In the HF and VHF bands, the ground waves tend to be absorbed by the earth. However, the waves that reach the ionosphere, a layer of charged particles circling the earth at a height of 100 to 500 km, are refracted by it and sent back to earth, as shown in Fig. 2-12(b). Under certain atmospheric conditions, the signals can bounce several times. Amateur radio operators (hams) use these bands to talk long distance. The military also communicate in the HF and VHF bands.

2.3.3 Microwave Transmission

Above 100 MHz, the waves travel in nearly straight lines and can therefore be narrowly focused. Concentrating all the energy into a small beam by means of a parabolic antenna (like the familiar satellite TV dish) gives a much higher signal-to-noise ratio, but the transmitting and receiving antennas must be accurately aligned with each other. In addition, this directionality allows multiple transmitters lined up in a row to communicate with multiple receivers in a row without interference, provided some minimum spacing rules are observed. Before fiber optics, for decades these microwaves formed the heart of the long-distance telephone transmission system. In fact, MCI, one of AT&T's first competitors after it was deregulated, built its entire system with microwave communications passing between towers tens of kilometers apart. Even the company's name reflected this (MCI stood for Microwave Communications, Inc.). MCI has since gone over to fiber and through a long series of corporate mergers and bankruptcies in the telecommunications shuffle has become part of Verizon.

Microwaves travel in a straight line, so if the towers are too far apart, the earth will get in the way (think about a Seattle-to-Amsterdam link). Thus, repeaters are needed periodically. The higher the towers are, the farther apart they can be. The distance between repeaters goes up very roughly with the square root of the tower height. For 100-meter-high towers, repeaters can be 80 km apart.

Unlike radio waves at lower frequencies, microwaves do not pass through buildings well. In addition, even though the beam may be well focused at the transmitter, there is still some divergence in space. Some waves may be refracted off low-lying atmospheric layers and may take slightly longer to arrive than the direct waves. The delayed waves may arrive out of phase with the direct wave and thus cancel the signal. This effect is called **multipath fading** and is often a serious problem. It is weather and frequency dependent. Some operators keep 10% of their channels idle as spares to switch on when multipath fading temporarily wipes out some frequency band.

The demand for more and more spectrum drives operators to yet higher frequencies. Bands up to 10 GHz are now in routine use, but at about 4 GHz a new problem sets in: absorption by water. These waves are only a few centimeters long and are absorbed by rain. This effect would be fine if one were planning to build a huge outdoor microwave oven for roasting passing birds, but for communication it is a severe problem. As with multipath fading, the only solution is to shut off links that are being rained on and route around them.

In summary, microwave communication is so widely used for long-distance telephone communication, mobile phones, television distribution, and other purposes that a severe shortage of spectrum has developed. It has several key advantages over fiber. The main one is that no right of way is needed to lay down cables. By buying a small plot of ground every 50 km and putting a microwave tower on it, one can bypass the telephone system entirely. This is how MCI managed to get started as a new long-distance telephone company so quickly. (Sprint, another early competitor to the deregulated AT&T, went a completely different route: it was formed by the Southern Pacific Railroad, which already owned a large amount of right of way and just buried fiber next to the tracks.)

Microwave is also relatively inexpensive. Putting up two simple towers (which can be just big poles with four guy wires) and putting antennas on each one may be cheaper than burying 50 km of fiber through a congested urban area or up over a mountain, and it may also be cheaper than leasing the telephone company's fiber, especially if the telephone company has not yet even fully paid for the copper it ripped out when it put in the fiber.

The Politics of the Electromagnetic Spectrum

To prevent total chaos, there are national and international agreements about who gets to use which frequencies. Since everyone wants a higher data rate, everyone wants more spectrum. National governments allocate spectrum for AM and FM radio, television, and mobile phones, as well as for telephone companies, police, maritime, navigation, military, government, and many other competing users. Worldwide, an agency of ITU-R (WRC) tries to coordinate this allocation so devices that work in multiple countries can be manufactured. However, countries are not bound by ITU-R's recommendations, and the FCC (Federal Communication Commission), which does the allocation for the United States, has occasionally rejected ITU-R's recommendations (usually because they required some politically powerful group to give up some piece of the spectrum).

Even when a piece of spectrum has been allocated to some use, such as mobile phones, there is the additional issue of which carrier is allowed to use which frequencies. Three algorithms were widely used in the past. The oldest algorithm, often called the **beauty contest**, requires each carrier to explain why its proposal serves the public interest best. Government officials then decide which of the nice stories they enjoy most. Having some government official award property worth billions of dollars to his favorite company often leads to bribery, corruption, nepotism, and worse. Furthermore, even a scrupulously honest government official who thought that a foreign company could do a better job than any of the national companies would have a lot of explaining to do.

This observation led to algorithm 2, holding a **lottery** among the interested companies. The problem with that idea is that companies with no interest in using the spectrum can enter the lottery. If, say, a fast food restaurant or shoe store chain wins, it can resell the spectrum to a carrier at a huge profit and with no risk.

Bestowing huge windfalls on alert but otherwise random companies has been severely criticized by many, which led to algorithm 3: **auction** off the bandwidth to the highest bidder. When the British government auctioned off the frequencies needed for third-generation mobile systems in 2000, it expected to get about $4 billion. It actually received about $40 billion because the carriers got into a feeding frenzy, scared to death of missing the mobile boat. This event switched on nearby governments' greedy bits and inspired them to hold their own auctions. It worked, but it also left some of the carriers with so much debt that they are close to bankruptcy. Even in the best cases, it will take many years to recoup the licensing fee.

A completely different approach to allocating frequencies is to not allocate them at all. Instead, let everyone transmit at will, but regulate the power used so that stations have such a short range that they do not interfere with each other. Accordingly, most governments have set aside some frequency bands, called the **ISM** (**Industrial, Scientific, Medical**) bands for unlicensed usage. Garage door openers, cordless phones, radio-controlled toys, wireless mice, and numerous other wireless household devices use the ISM bands. To minimize interference between these uncoordinated devices, the FCC mandates that all devices in the ISM bands limit their transmit power (e.g., to 1 watt) and use other techniques to spread their signals over a range of frequencies. Devices may also need to take care to avoid interference with radar installations.

The location of these bands varies somewhat from country to country. In the United States, for example, the bands that networking devices use in practice without requiring a FCC license are shown in Fig. 2-13. The 900-MHz band was used for early versions of 802.11, but it is crowded. The 2.4-GHz band is available in most countries and widely used for 802.11b/g and Bluetooth, though it is subject to interference from microwave ovens and radar installations. The 5-GHz part of the spectrum includes **U-NII (Unlicensed National Information Infrastructure)** bands. The 5-GHz bands are relatively undeveloped but, since they have the most bandwidth and are used by 802.11a, they are quickly gaining in popularity.

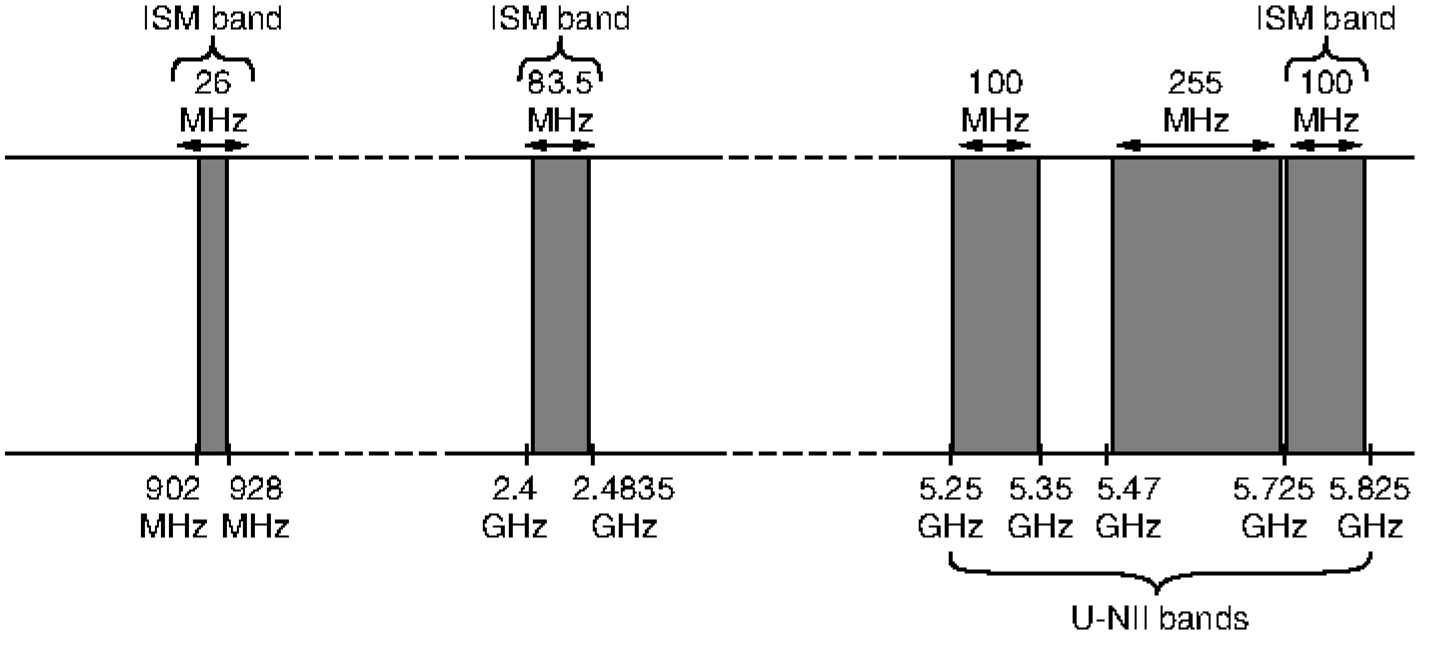


Figure 2-13. ISM and U-NII bands used in the United States by wireless devices.

The unlicensed bands have been a roaring success over the past decade. The ability to use the spectrum freely has unleashed a huge amount of innovation in wireless LANs and PANs, evidenced by the widespread deployment of technologies such as 802.11 and Bluetooth. To continue this innovation, more spectrum is needed. One exciting development in the U.S. is the FCC decision in 2009 to allow unlicensed use of **white spaces** around 700 MHz. White spaces are frequency bands that have been allocated but are not being used locally. The transition from analog to all-digital television broadcasts in the U.S. in 2010 freed up white spaces around 700 MHz. The only difficulty is that, to use the white spaces, unlicensed devices must be able to detect any nearby licensed transmitters, including wireless microphones, that have first rights to use the frequency band.

Another flurry of activity is happening around the 60-GHz band. The FCC opened 57 GHz to 64 GHz for unlicensed operation in 2001. This range is an enormous portion of spectrum, more than all the other ISM bands combined, so it can support the kind of high-speed networks that would be needed to stream high-definition TV through the air across your living room. At 60 GHz, radio waves are absorbed by oxygen. This means that signals do not propagate far, making them well suited to short-range networks. The high frequencies (60 GHz is in the Extremely High Frequency or "millimeter" band, just below infrared radiation) posed an initial challenge for equipment makers, but products are now on the market.

2.3.4 Infrared Transmission

Unguided infrared waves are widely used for short-range communication. The remote controls used for televisions, VCRs, and stereos all use infrared communication. They are relatively directional, cheap, and easy to build but have a major drawback: they do not pass through solid objects. (Try standing between your remote control and your television and see if it still works.) In general, as we go from long-wave radio toward visible light, the waves behave more and more like light and less and less like radio.

On the other hand, the fact that infrared waves do not pass through solid walls well is also a plus. It means that an infrared system in one room of a building will not interfere with a similar system in adjacent rooms or buildings: you cannot control your neighbor's television with your remote control. Furthermore, security of infrared systems against eavesdropping is better than that of radio systems precisely for this reason. Therefore, no government license is needed to operate an infrared system, in contrast to radio systems, which must be licensed outside the ISM bands. Infrared communication has a limited use on the desktop, for example, to connect notebook computers and printers with the **IrDA** (**Infrared Data Association**) standard, but it is not a major player in the communication game.

2.3.5 Light Transmission

Unguided optical signaling or **free-space optics** has been in use for centuries. Paul Revere used binary optical signaling from the Old North Church just prior to his famous ride. A more modern application is to connect the LANs in two buildings via lasers mounted on their rooftops. Optical signaling using lasers is inherently unidirectional, so each end needs its own laser and its own photodetector. This scheme offers very high bandwidth at very low cost and is relatively secure because it is difficult to tap a narrow laser beam. It is also relatively easy to install and, unlike microwave transmission, does not require an FCC license.

The laser's strength, a very narrow beam, is also its weakness here. Aiming a laser beam 1 mm wide at a target the size of a pin head 500 meters away requires the marksmanship of a latter-day Annie Oakley. Usually, lenses are put into the system to defocus the beam slightly. To add to the difficulty, wind and temperature changes can distort the beam and laser beams also cannot penetrate rain or thick fog, although they normally work well on sunny days. However, many of these factors are not an issue when the use is to connect two spacecraft.

One of the authors (AST) once attended a conference at a modern hotel in Europe at which the conference organizers thoughtfully provided a room full of terminals to allow the attendees to read their email during boring presentations. Since the local PTT was unwilling to install a large number of telephone lines for just 3 days, the organizers put a laser on the roof and aimed it at their university's computer science building a few kilometers away. They tested it the night before the conference and it worked perfectly. At 9 A.M. on a bright, sunny day, the link failed completely and stayed down all day. The pattern repeated itself the next two days. It was not until after the conference that the organizers discovered the problem: heat from the sun during the daytime caused convection currents to rise up from the roof of the building, as shown in Fig. 2-14. This turbulent air diverted the beam and made it dance around the detector, much like a shimmering road on a hot day. The lesson here is that to work well in difficult conditions as well as good conditions, unguided optical links need to be engineered with a sufficient margin of error.

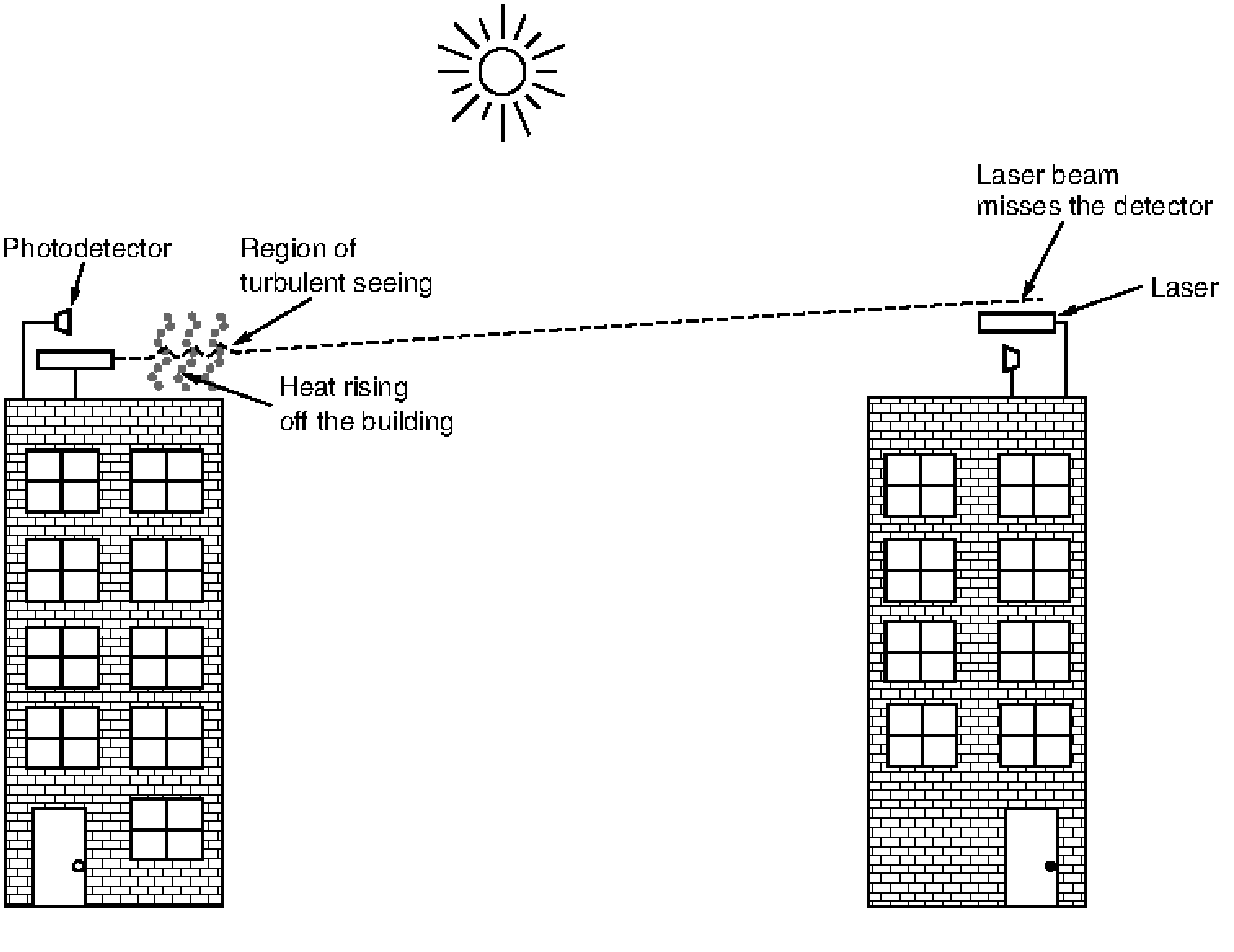


Figure 2-14. Convection currents can interfere with laser communication systems. A bidirectional system with two lasers is pictured here.

Unguided optical communication may seem like an exotic networking technology today, but it might soon become much more prevalent. We are surrounded by cameras (that sense light) and displays (that emit light using LEDs and other technology). Data communication can be layered on top of these displays by encoding information in the pattern at which LEDs turn on and off that is below the threshold of human perception. Communicating with visible light in this way is inherently safe and creates a low-speed network in the immediate vicinity of the display. This could enable all sorts of fanciful ubiquitous computing scenarios. The flashing lights on emergency vehicles might alert nearby traffic lights and vehicles to help clear a path. Informational signs might broadcast maps. Even festive lights might broadcast songs that are synchronized with their display.

2.4 COMMUNICATION SATELLITES

In the 1950s and early 1960s, people tried to set up communication systems by bouncing signals off metallized weather balloons. Unfortunately, the received signals were too weak to be of any practical use. Then the U.S. Navy noticed a kind of permanent weather balloon in the sky—the moon—and built an operational system for ship-to-shore communication by bouncing signals off it.

Further progress in the celestial communication field had to wait until the first communication satellite was launched. The key difference between an artificial satellite and a real one is that the artificial one can amplify the signals before sending them back, turning a strange curiosity into a powerful communication system.

Communication satellites have some interesting properties that make them attractive for many applications. In its simplest form, a communication satellite can be thought of as a big microwave repeater in the sky. It contains several **transponders**, each of which listens to some portion of the spectrum, amplifies the incoming signal, and then rebroadcasts it at another frequency to avoid interference with the incoming signal. This mode of operation is known as a **bent pipe**. Digital processing can be added to separately manipulate or redirect data streams in the overall band, or digital information can even be received by the satellite and rebroadcast. Regenerating signals in this way improves performance compared to a bent pipe because the satellite does not amplify noise in the upward signal. The downward beams can be broad, covering a substantial fraction of the earth's surface, or narrow, covering an area only hundreds of kilometers in diameter.

According to Kepler's law, the orbital period of a satellite varies as the radius of the orbit to the 3/2 power. The higher the satellite, the longer the period. Near the surface of the earth, the period is about 90 minutes. Consequently, low-orbit satellites pass out of view fairly quickly, so many of them are needed to provide continuous coverage and ground antennas must track them. At an altitude of about 35,800 km, the period is 24 hours. At an altitude of 384,000 km, the period is about one month, as anyone who has observed the moon regularly can testify.

A satellite's period is important, but it is not the only issue in determining where to place it. Another issue is the presence of the Van Allen belts, layers of highly charged particles trapped by the earth's magnetic field. Any satellite flying within them would be destroyed fairly quickly by the particles. These factors lead to three regions in which satellites can be placed safely. These regions and some of their properties are illustrated in Fig. 2-15. Below we will briefly describe the satellites that inhabit each of these regions.

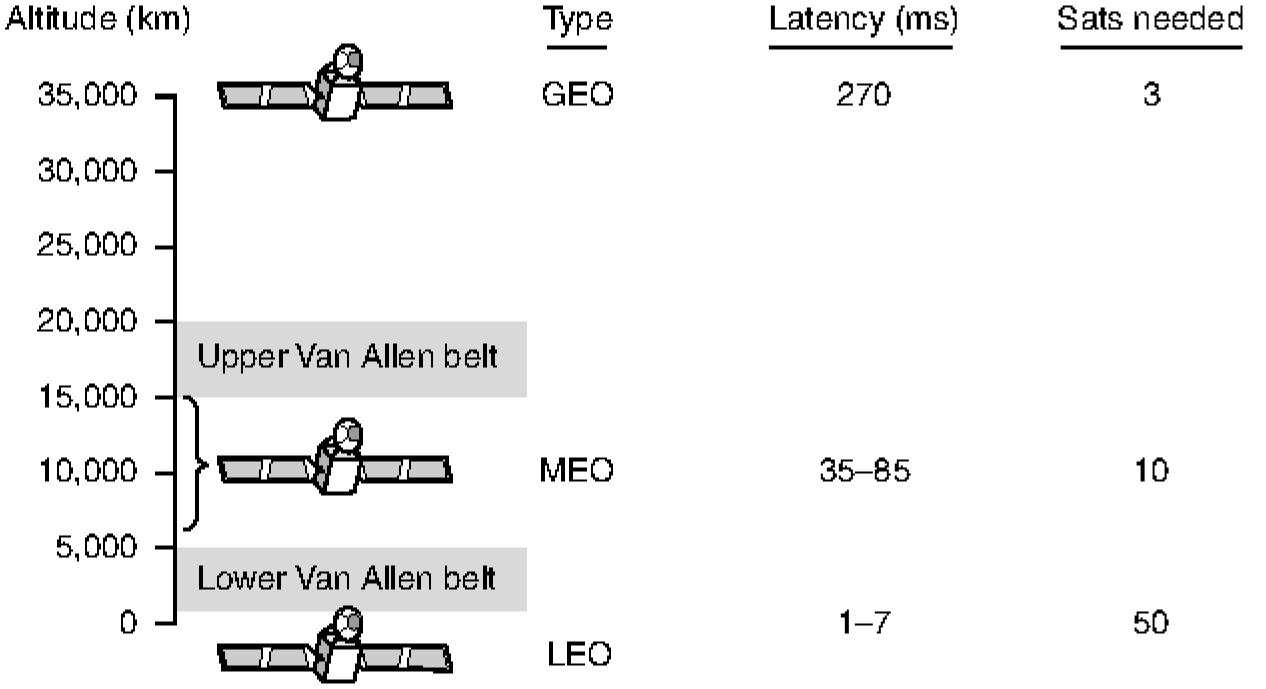


Figure 2-15. Communication satellites and some of their properties, including altitude above the earth, round-trip delay time, and number of satellites needed for global coverage.

2.4.1 Geostationary Satellites

In 1945, the science fiction writer Arthur C. Clarke calculated that a satellite at an altitude of 35,800 km in a circular equatorial orbit would appear to remain motionless in the sky, so it would not need to be tracked (Clarke, 1945). He went on to describe a complete communication system that used these (manned) **geostationary satellites**, including the orbits, solar panels, radio frequencies, and launch procedures. Unfortunately, he concluded that satellites were impractical due to the impossibility of putting power-hungry, fragile vacuum tube amplifiers into orbit, so he never pursued this idea further, although he wrote some science fiction stories about it.

The invention of the transistor changed all that, and the first artificial communication satellite, Telstar, was launched in July 1962. Since then, communication satellites have become a multibillion dollar business and the only aspect of outer space that has become highly profitable. These high-flying satellites are often called **GEO** (**Geostationary Earth Orbit**) satellites.

With current technology, it is unwise to have geostationary satellites spaced much closer than 2 degrees in the 360-degree equatorial plane, to avoid interference. With a spacing of 2 degrees, there can only be 360/2 = 180 of these satellites in the sky at once. However, each transponder can use multiple frequencies and polarizations to increase the available bandwidth.

To prevent total chaos in the sky, orbit slot allocation is done by ITU. This process is highly political, with countries barely out of the stone age demanding "their" orbit slots (for the purpose of leasing them to the highest bidder). Other countries, however, maintain that national property rights do not extend up to the moon and that no country has a legal right to the orbit slots above its territory. To add to the fight, commercial telecommunication is not the only application. Television broadcasters, governments, and the military also want a piece of the orbiting pie.

Modern satellites can be quite large, weighing over 5000 kg and consuming several kilowatts of electric power produced by the solar panels. The effects of solar, lunar, and planetary gravity tend to move them away from their assigned orbit slots and orientations, an effect countered by on-board rocket motors. This fine-tuning activity is called **station keeping**. However, when the fuel for the motors has been exhausted (typically after about 10 years) the satellite drifts and tumbles helplessly, so it has to be turned off. Eventually, the orbit decays and the satellite reenters the atmosphere and burns up (or very rarely crashes to earth).

Orbit slots are not the only bone of contention. Frequencies are an issue, too, because the downlink transmissions interfere with existing microwave users. Consequently, ITU has allocated certain frequency bands to satellite users. The main ones are listed in Fig. 2-16. The C band was the first to be designated for commercial satellite traffic. Two frequency ranges are assigned in it, the lower one for downlink traffic (from the satellite) and the upper one for uplink traffic (to the satellite). To allow traffic to go both ways at the same time, two channels are required. These channels are already overcrowded because they are also used by the common carriers for terrestrial microwave links. The L and S bands were added by international agreement in 2000. However, they are narrow and also crowded.

Band	Downlink	Uplink	Bandwidth	Problems
L	1.5 GHz	1.6 GHz	15 MHz	Low bandwidth; crowded
S	1.9 GHz	2.2 GHz	70 MHz	Low bandwidth; crowded
C	4.0 GHz	6.0 GHz	500 MHz	Terrestrial interference
Ku	11 GHz	14 GHz	500 MHz	Rain
Ka	20 GHz	30 GHz	3500 MHz	Rain, equipment cost

Figure 2-16. The principal satellite bands.

The next-highest band available to commercial telecommunication carriers is the Ku (K under) band. This band is not (yet) congested, and at its higher frequencies, satellites can be spaced as close as 1 degree. However, another problem exists: rain. Water absorbs these short microwaves well. Fortunately, heavy storms are usually localized, so using several widely separated ground stations instead of just one circumvents the problem, but at the price of extra antennas, extra cables, and extra electronics to enable rapid switching between stations. Bandwidth has also been allocated in the Ka (K above) band for commercial satellite traffic, but the equipment needed to use it is expensive. In addition to these commercial bands, many government and military bands also exist.

A modern satellite has around 40 transponders, most often with a 36-MHz bandwidth. Usually, each transponder operates as a bent pipe, but recent satellites have some on-board processing capacity, allowing more sophisticated operation. In the earliest satellites, the division of the transponders into channels was static: the bandwidth was simply split up into fixed frequency bands. Nowadays, each transponder beam is divided into time slots, with various users taking turns. We will study these two techniques (frequency division multiplexing and time division multiplexing) in detail later in this chapter.

The first geostationary satellites had a single spatial beam that illuminated about 1/3 of the earth's surface, called its **footprint**. With the enormous decline in the price, size, and power requirements of microelectronics, a much more sophisticated broadcasting strategy has become possible. Each satellite is equipped with multiple antennas and multiple transponders. Each downward beam can be focused on a small geographical area, so multiple upward and downward transmissions can take place simultaneously. Typically, these so-called **spot beams** are elliptically shaped, and can be as small as a few hundred km in diameter. A communication satellite for the United States typically has one wide beam for the contiguous 48 states, plus spot beams for Alaska and Hawaii.

A recent development in the communication satellite world is the development of low-cost microstations, sometimes called **VSATs** (**Very Small Aperture Terminals**) (Abramson, 2000). These tiny terminals have 1-meter or smaller antennas (versus 10 m for a standard GEO antenna) and can put out about 1 watt of power. The uplink is generally good for up to 1 Mbps, but the downlink is often up to several megabits/sec. Direct broadcast satellite television uses this technology for one-way transmission.

In many VSAT systems, the microstations do not have enough power to communicate directly with one another (via the satellite, of course). Instead, a special ground station, the **hub**, with a large, high-gain antenna is needed to relay traffic between VSATs, as shown in Fig. 2-17. In this mode of operation, either the sender or the receiver has a large antenna and a powerful amplifier. The trade-off is a longer delay in return for having cheaper end-user stations.

VSATs have great potential in rural areas. It is not widely appreciated, but over half the world's population lives more than hour's walk from the nearest

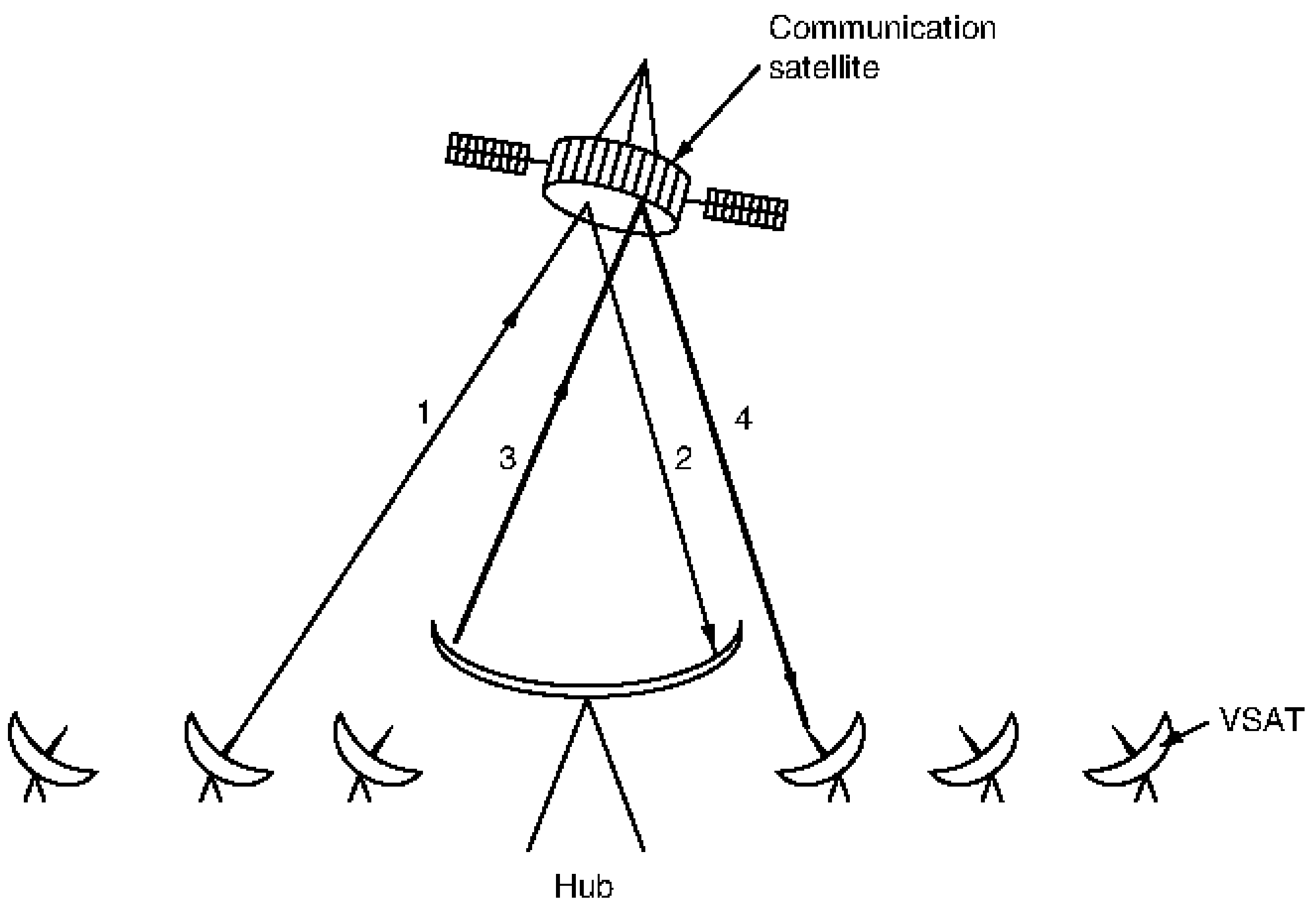


Figure 2-17. VSATs using a hub.

telephone. Stringing telephone wires to thousands of small villages is far beyond the budgets of most Third World governments, but installing 1-meter VSAT dishes powered by solar cells is often feasible. VSATs provide the technology that will wire the world.

Communication satellites have several properties that are radically different from terrestrial point-to-point links. To begin with, even though signals to and from a satellite travel at the speed of light (nearly 300,000 km/sec), the long round-trip distance introduces a substantial delay for GEO satellites. Depending on the distance between the user and the ground station and the elevation of the satellite above the horizon, the end-to-end transit time is between 250 and 300 msec. A typical value is 270 msec (540 msec for a VSAT system with a hub).

For comparison purposes, terrestrial microwave links have a propagation delay of roughly 3 μsec/km, and coaxial cable or fiber optic links have a delay of approximately 5 μsec/km. The latter are slower than the former because electromagnetic signals travel faster in air than in solid materials.

Another important property of satellites is that they are inherently broadcast media. It does not cost more to send a message to thousands of stations within a transponder's footprint than it does to send to one. For some applications, this property is very useful. For example, one could imagine a satellite broadcasting popular Web pages to the caches of a large number of computers spread over a wide area. Even when broadcasting can be simulated with point-to-point lines, satellite broadcasting may be much cheaper. On the other hand, from a privacy point of view, satellites are a complete disaster: everybody can hear everything. Encryption is essential when security is required.

Satellites also have the property that the cost of transmitting a message is independent of the distance traversed. A call across the ocean costs no more to service than a call across the street. Satellites also have excellent error rates and can be deployed almost instantly, a major consideration for disaster response and military communication.

2.4.2 Medium-Earth Orbit Satellites

At much lower altitudes, between the two Van Allen belts, we find the **MEO** (**Medium-Earth Orbit**) satellites. As viewed from the earth, these drift slowly in longitude, taking something like 6 hours to circle the earth. Accordingly, they must be tracked as they move through the sky. Because they are lower than the GEOs, they have a smaller footprint on the ground and require less powerful transmitters to reach them. Currently they are used for navigation systems rather than telecommunications, so we will not examine them further here. The constellation of roughly 30 **GPS** (**Global Positioning System**) satellites orbiting at about 20,200 km are examples of MEO satellites.

2.4.3 Low-Earth Orbit Satellites

Moving down in altitude, we come to the **LEO** (**Low-Earth Orbit**) satellites. Due to their rapid motion, large numbers of them are needed for a complete system. On the other hand, because the satellites are so close to the earth, the ground stations do not need much power, and the round-trip delay is only a few milliseconds. The launch cost is substantially cheaper too. In this section we will examine two examples of satellite constellations for voice service, Iridium and Globalstar.

For the first 30 years of the satellite era, low-orbit satellites were rarely used because they zip into and out of view so quickly. In 1990, Motorola broke new ground by filing an application with the FCC asking for permission to launch 77 low-orbit satellites for the **Iridium** project (element 77 is iridium). The plan was later revised to use only 66 satellites, so the project should have been renamed Dysprosium (element 66), but that probably sounded too much like a disease. The idea was that as soon as one satellite went out of view, another would replace it. This proposal set off a feeding frenzy among other communication companies. All of a sudden, everyone wanted to launch a chain of low-orbit satellites.

After seven years of cobbling together partners and financing, communication service began in November 1998. Unfortunately, the commercial demand for large, heavy satellite telephones was negligible because the mobile phone network had grown in a spectacular way since 1990. As a consequence, Iridium was not profitable and was forced into bankruptcy in August 1999 in one of the most spectacular corporate fiascos in history. The satellites and other assets (worth $5 billion) were later purchased by an investor for $25 million at a kind of extraterrestrial garage sale. Other satellite business ventures promptly followed suit.

The Iridium service restarted in March 2001 and has been growing ever since. It provides voice, data, paging, fax, and navigation service everywhere on land, air, and sea, via hand-held devices that communicate directly with the Iridium satellites. Customers include the maritime, aviation, and oil exploration industries, as well as people traveling in parts of the world lacking a telecom infrastructure (e.g., deserts, mountains, the South Pole, and some Third World countries).

The Iridium satellites are positioned at an altitude of 750 km, in circular polar orbits. They are arranged in north-south necklaces, with one satellite every 32 degrees of latitude, as shown in Fig. 2-18. Each satellite has a maximum of 48 cells (spot beams) and a capacity of 3840 channels, some of which are used for paging and navigation, while others are used for data and voice.

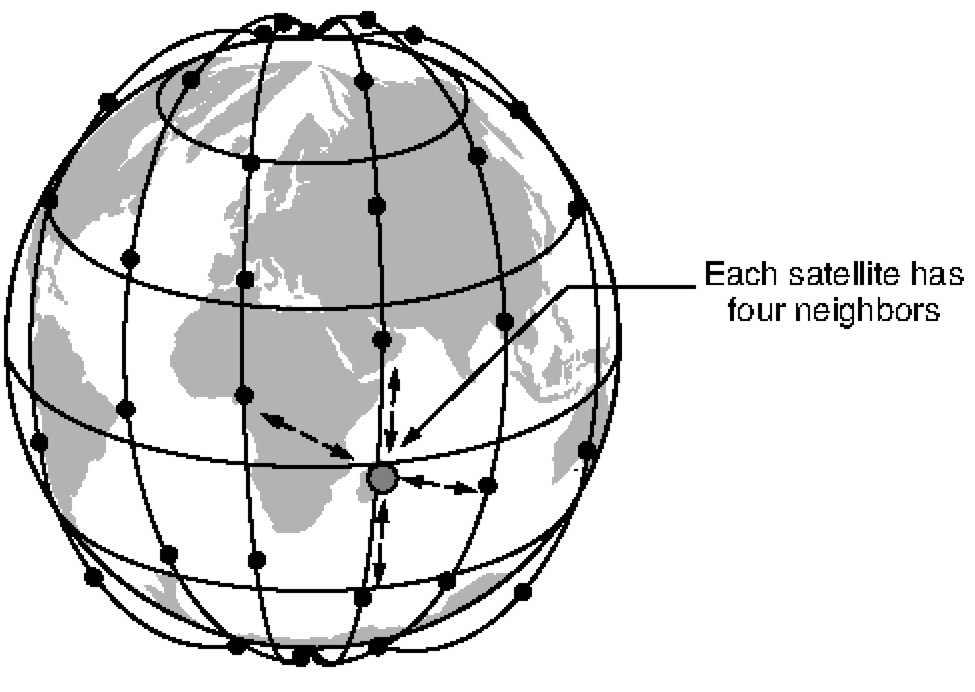


Figure 2-18. The Iridium satellites form six necklaces around the earth.

With six satellite necklaces the entire earth is covered, as suggested by Fig. 2-18. An interesting property of Iridium is that communication between distant customers takes place in space, as shown in Fig. 2-19(a). Here we see a caller at the North Pole contacting a satellite directly overhead. Each satellite has four neighbors with which it can communicate, two in the same necklace (shown) and two in adjacent necklaces (not shown). The satellites relay the call across this grid until it is finally sent down to the callee at the South Pole.

An alternative design to Iridium is **Globalstar**. It is based on 48 LEO satellites but uses a different switching scheme than that of Iridium. Whereas Iridium relays calls from satellite to satellite, which requires sophisticated switching equipment in the satellites, Globalstar uses a traditional bent-pipe design. The call originating at the North Pole in Fig. 2-19(b) is sent back to earth and picked

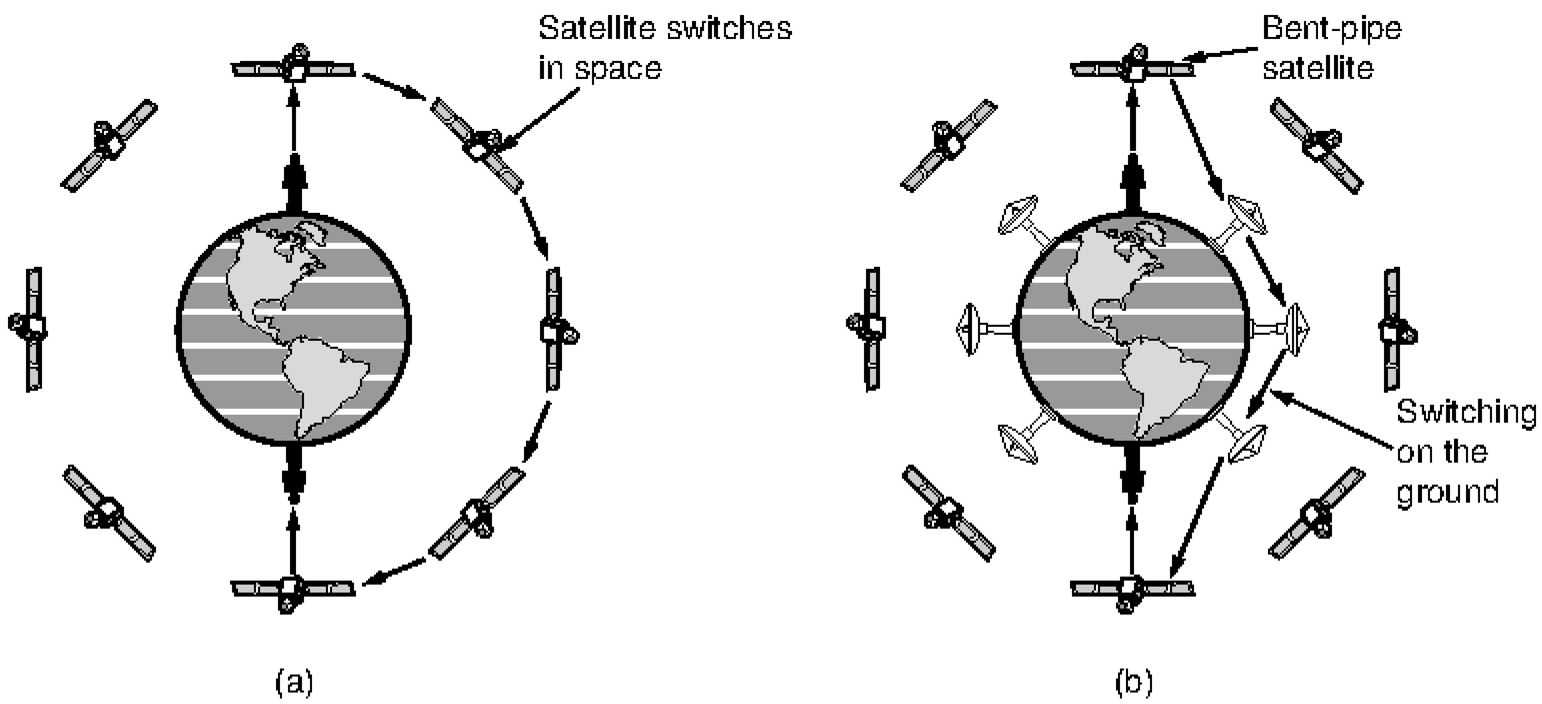


Figure 2-19. (a) Relaying in space. (b) Relaying on the ground.

up by the large ground station at Santa's Workshop. The call is then routed via a terrestrial network to the ground station nearest the callee and delivered by a bent-pipe connection as shown. The advantage of this scheme is that it puts much of the complexity on the ground, where it is easier to manage. Also, the use of large ground station antennas that can put out a powerful signal and receive a weak one means that lower-powered telephones can be used. After all, the telephone puts out only a few milliwatts of power, so the signal that gets back to the ground station is fairly weak, even after having been amplified by the satellite.

Satellites continue to be launched at a rate of around 20 per year, including ever-larger satellites that now weigh over 5000 kilograms. But there are also very small satellites for the more budget-conscious organization. To make space research more accessible, academics from Cal Poly and Stanford got together in 1999 to define a standard for miniature satellites and an associated launcher that would greatly lower launch costs (Nugent et al., 2008). **CubeSats** are satellites in units of 10 cm × 10 cm × 10 cm cubes, each weighing no more than 1 kilogram, that can be launched for as little as $40,000 each. The launcher flies as a secondary payload on commercial space missions. It is basically a tube that takes up to three units of cubesats and uses springs to release them into orbit. Roughly 20 cubesats have launched so far, with many more in the works. Most of them communicate with ground stations on the UHF and VHF bands.

2.4.4 Satellites Versus Fiber

A comparison between satellite communication and terrestrial communication is instructive. As recently as 25 years ago, a case could be made that the future of communication lay with communication satellites. After all, the telephone system had changed little in the previous 100 years and showed no signs of changing in the next 100 years. This glacial movement was caused in no small part by the regulatory environment in which the telephone companies were expected to provide good voice service at reasonable prices (which they did), and in return got a guaranteed profit on their investment. For people with data to transmit, 1200-bps modems were available. That was pretty much all there was.

The introduction of competition in 1984 in the United States and somewhat later in Europe changed all that radically. Telephone companies began replacing their long-haul networks with fiber and introduced high-bandwidth services like ADSL (Asymmetric Digital Subscriber Line). They also stopped their long-time practice of charging artificially high prices to long-distance users to subsidize local service. All of a sudden, terrestrial fiber connections looked like the winner.

Nevertheless, communication satellites have some major niche markets that fiber does not (and, sometimes, cannot) address. First, when rapid deployment is critical, satellites win easily. A quick response is useful for military communication systems in times of war and disaster response in times of peace. Following the massive December 2004 Sumatra earthquake and subsequent tsunami, for example, communications satellites were able to restore communications to first responders within 24 hours. This rapid response was possible because there is a developed satellite service provider market in which large players, such as Intelsat with over 50 satellites, can rent out capacity pretty much anywhere it is needed. For customers served by existing satellite networks, a VSAT can be set up easily and quickly to provide a megabit/sec link to elsewhere in the world.

A second niche is for communication in places where the terrestrial infrastructure is poorly developed. Many people nowadays want to communicate everywhere they go. Mobile phone networks cover those locations with good population density, but do not do an adequate job in other places (e.g., at sea or in the desert). Conversely, Iridium provides voice service everywhere on Earth, even at the South Pole. Terrestrial infrastructure can also be expensive to install, depending on the terrain and necessary rights of way. Indonesia, for example, has its own satellite for domestic telephone traffic. Launching one satellite was cheaper than stringing thousands of undersea cables among the 13,677 islands in the archipelago.

A third niche is when broadcasting is essential. A message sent by satellite can be received by thousands of ground stations at once. Satellites are used to distribute much network TV programming to local stations for this reason. There is now a large market for satellite broadcasts of digital TV and radio directly to end users with satellite receivers in their homes and cars. All sorts of other content can be broadcast too. For example, an organization transmitting a stream of stock, bond, or commodity prices to thousands of dealers might find a satellite system to be much cheaper than simulating broadcasting on the ground.

In short, it looks like the mainstream communication of the future will be terrestrial fiber optics combined with cellular radio, but for some specialized uses, satellites are better. However, there is one caveat that applies to all of this: economics. Although fiber offers more bandwidth, it is conceivable that terrestrial and satellite communication could compete aggressively on price. If advances in technology radically cut the cost of deploying a satellite (e.g., if some future space vehicle can toss out dozens of satellites on one launch) or low-orbit satellites catch on in a big way, it is not certain that fiber will win all markets.

2.5 DIGITAL MODULATION AND MULTIPLEXING

Now that we have studied the properties of wired and wireless channels, we turn our attention to the problem of sending digital information. Wires and wireless channels carry analog signals such as continuously varying voltage, light intensity, or sound intensity. To send digital information, we must devise analog signals to represent bits. The process of converting between bits and signals that represent them is called **digital modulation**.

We will start with schemes that directly convert bits into a signal. These schemes result in **baseband transmission**, in which the signal occupies frequencies from zero up to a maximum that depends on the signaling rate. It is common for wires. Then we will consider schemes that regulate the amplitude, phase, or frequency of a carrier signal to convey bits. These schemes result in **passband transmission**, in which the signal occupies a band of frequencies around the frequency of the carrier signal. It is common for wireless and optical channels for which the signals must reside in a given frequency band.

Channels are often shared by multiple signals. After all, it is much more convenient to use a single wire to carry several signals than to install a wire for every signal. This kind of sharing is called **multiplexing**. It can be accomplished in several different ways. We will present methods for time, frequency, and code division multiplexing.

The modulation and multiplexing techniques we describe in this section are all widely used for wires, fiber, terrestrial wireless, and satellite channels. In the following sections, we will look at examples of networks to see them in action.

2.5.1 Baseband Transmission

The most straightforward form of digital modulation is to use a positive voltage to represent a 1 and a negative voltage to represent a 0. For an optical fiber, the presence of light might represent a 1 and the absence of light might represent a 0. This scheme is called **NRZ** (**Non-Return-to-Zero**). The odd name is for historical reasons, and simply means that the signal follows the data. An example is shown in Fig. 2-20(b).

Once sent, the NRZ signal propagates down the wire. At the other end, the receiver converts it into bits by sampling the signal at regular intervals of time.

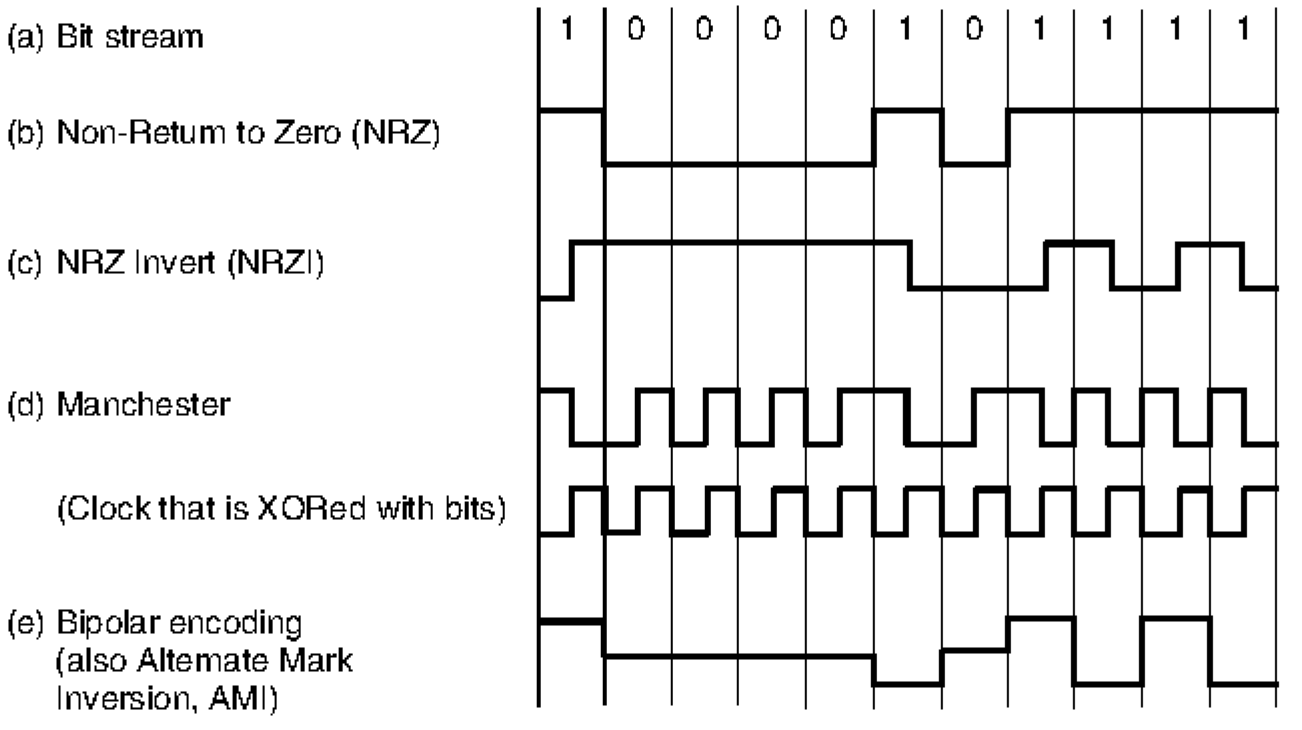


Figure 2-20. Line codes: (a) Bits, (b) NRZ, (c) NRZI, (d) Manchester, (e) Bipolar or AMI.

This signal will not look exactly like the signal that was sent. It will be attenuated and distorted by the channel and noise at the receiver. To decode the bits, the receiver maps the signal samples to the closest symbols. For NRZ, a positive voltage will be taken to indicate that a 1 was sent and a negative voltage will be taken to indicate that a 0 was sent.

NRZ is a good starting point for our studies because it is simple, but it is seldom used by itself in practice. More complex schemes can convert bits to signals that better meet engineering considerations. These schemes are called **line codes**. Below, we describe line codes that help with bandwidth efficiency, clock recovery, and DC balance.

Bandwidth Efficiency

With NRZ, the signal may cycle between the positive and negative levels up to every 2 bits (in the case of alternating 1s and 0s). This means that we need a bandwidth of at least $B/2$ Hz when the bit rate is B bits/sec. This relation comes from the Nyquist rate [Eq. (2-2)]. It is a fundamental limit, so we cannot run NRZ faster without using more bandwidth. Bandwidth is often a limited resource, even for wired channels, Higher-frequency signals are increasingly attenuated, making them less useful, and higher-frequency signals also require faster electronics.

One strategy for using limited bandwidth more efficiently is to use more than two signaling levels. By using four voltages, for instance, we can send 2 bits at once as a single **symbol**. This design will work as long as the signal at the receiver is sufficiently strong to distinguish the four levels. The rate at which the signal changes is then half the bit rate, so the needed bandwidth has been reduced.

We call the rate at which the signal changes the **symbol rate** to distinguish it from the **bit rate**. The bit rate is the symbol rate multiplied by the number of bits per symbol. An older name for the symbol rate, particularly in the context of devices called telephone modems that convey digital data over telephone lines, is the **baud rate**. In the literature, the terms "bit rate" and "baud rate" are often used incorrectly.

Note that the number of signal levels does not need to be a power of two. Often it is not, with some of the levels used for protecting against errors and simplifying the design of the receiver.

Clock Recovery

For all schemes that encode bits into symbols, the receiver must know when one symbol ends and the next symbol begins to correctly decode the bits. With NRZ, in which the symbols are simply voltage levels, a long run of 0s or 1s leaves the signal unchanged. After a while it is hard to tell the bits apart, as 15 zeros look much like 16 zeros unless you have a very accurate clock.

Accurate clocks would help with this problem, but they are an expensive solution for commodity equipment. Remember, we are timing bits on links that run at many megabits/sec, so the clock would have to drift less than a fraction of a microsecond over the longest permitted run. This might be reasonable for slow links or short messages, but it is not a general solution.

One strategy is to send a separate clock signal to the receiver. Another clock line is no big deal for computer buses or short cables in which there are many lines in parallel, but it is wasteful for most network links since if we had another line to send a signal we could use it to send data. A clever trick here is to mix the clock signal with the data signal by XORing them together so that no extra line is needed. The results are shown in Fig. 2-20(d). The clock makes a clock transition in every bit time, so it runs at twice the bit rate. When it is XORed with the 0 level it makes a low-to-high transition that is simply the clock. This transition is a logical 0. When it is XORed with the 1 level it is inverted and makes a high-to-low transition. This transition is a logical 1. This scheme is called **Manchester** encoding and was used for classic Ethernet.

The downside of Manchester encoding is that it requires twice as much bandwidth as NRZ because of the clock, and we have learned that bandwidth often matters. A different strategy is based on the idea that we should code the data to ensure that there are enough transitions in the signal. Consider that NRZ will have clock recovery problems only for long runs of 0s and 1s. If there are frequent transitions, it will be easy for the receiver to stay synchronized with the incoming stream of symbols.

As a step in the right direction, we can simplify the situation by coding a 1 as a transition and a 0 as no transition, or vice versa. This coding is called **NRZI** (**Non-Return-to-Zero Inverted**), a twist on NRZ. An example is shown in

Fig. 2-20(c). The popular **USB (Universal Serial Bus)** standard for connecting computer peripherals uses NRZI. With it, long runs of 1s do not cause a problem.

Of course, long runs of 0s still cause a problem that we must fix. If we were the telephone company, we might simply require that the sender not transmit too many 0s. Older digital telephone lines in the U.S., called **T1 lines**, did in fact require that no more than 15 consecutive 0s be sent for them to work correctly. To really fix the problem we can break up runs of 0s by mapping small groups of bits to be transmitted so that groups with successive 0s are mapped to slightly longer patterns that do not have too many consecutive 0s.

A well-known code to do this is called **4B/5B**. Every 4 bits is mapped into a5-bit pattern with a fixed translation table. The five bit patterns are chosen so that there will never be a run of more than three consecutive 0s. The mapping is shown in Fig. 2-21. This scheme adds 25% overhead, which is better than the 100% overhead of Manchester encoding. Since there are 16 input combinations and 32 output combinations, some of the output combinations are not used. Putting aside the combinations with too many successive 0s, there are still some codes left. As a bonus, we can use these nondata codes to represent physical layer control signals. For example, in some uses "11111" represents an idle line and "11000" represents the start of a frame.

Data (4B)	Codeword (5B)	Data (4B)	Codeword (5B)
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Figure 2-21. 4B/5B mapping.

An alternative approach is to make the data look random, known as scrambling. In this case it is very likely that there will be frequent transitions. A **scrambler** works by XORing the data with a pseudorandom sequence before it is transmitted. This mixing will make the data as random as the pseudorandom sequence (assuming it is independent of the pseudorandom sequence). The receiver then XORs the incoming bits with the same pseudorandom sequence to recover the real data. For this to be practical, the pseudorandom sequence must be easy to create. It is commonly given as the seed to a simple random number generator.

Scrambling is attractive because it adds no bandwidth or time overhead. In fact, it often helps to condition the signal so that it does not have its energy in dominant frequency components (caused by repetitive data patterns) that might radiate electromagnetic interference. Scrambling helps because random signals tend to be "white," or have energy spread across the frequency components.

However, scrambling does not guarantee that there will be no long runs. It is possible to get unlucky occasionally. If the data are the same as the pseudorandom sequence, they will XOR to all 0s. This outcome does not generally occur with a long pseudorandom sequence that is difficult to predict. However, with a short or predictable sequence, it might be possible for malicious users to send bit patterns that cause long runs of 0s after scrambling and cause links to fail. Early versions of the standards for sending IP packets over SONET links in the telephone system had this defect (Malis and Simpson, 1999). It was possible for users to send certain "killer packets" that were guaranteed to cause problems.

Balanced Signals

Signals that have as much positive voltage as negative voltage even over short periods of time are called **balanced signals**. They average to zero, which means that they have no DC electrical component. The lack of a DC component is an advantage because some channels, such as coaxial cable or lines with transformers, strongly attenuate a DC component due to their physical properties. Also, one method of connecting the receiver to the channel called **capacitive coupling** passes only the AC portion of a signal. In either case, if we send a signal whose average is not zero, we waste energy as the DC component will be filtered out.

Balancing helps to provide transitions for clock recovery since there is a mix of positive and negative voltages. It also provides a simple way to calibrate receivers because the average of the signal can be measured and used as a decision threshold to decode symbols. With unbalanced signals, the average may be drift away from the true decision level due to a density of 1s, for example, which would cause more symbols to be decoded with errors.

A straightforward way to construct a balanced code is to use two voltage levels to represent a logical 1, (say +1 V or −1 V) with 0 V representing a logical zero. To send a 1, the transmitter alternates between the +1 V and −1 V levels so that they always average out. This scheme is called **bipolar encoding**. In telephone networks it is called **AMI** (**Alternate Mark Inversion**), building on old terminology in which a 1 is called a "mark" and a 0 is called a "space." An example is given in Fig. 2-20(e).

Bipolar encoding adds a voltage level to achieve balance. Alternatively we can use a mapping like 4B/5B to achieve balance (as well as transitions for clock recovery). An example of this kind of balanced code is the **8B/10B** line code. It maps 8 bits of input to 10 bits of output, so it is 80% efficient, just like the 4B/5B line code. The 8 bits are split into a group of 5 bits, which is mapped to 6 bits, and a group of 3 bits, which is mapped to 4 bits. The 6-bit and 4-bit symbols are then concatenated. In each group, some input patterns can be mapped to balanced output patterns that have the same number of 0s and 1s. For example, "001" is mapped to "1001," which is balanced. But there are not enough combinations for all output patterns to be balanced. For these cases, each input pattern is mapped to two output patterns. One will have an extra 1 and the alternate will have an extra 0. For example, "000" is mapped to both "1011" and its complement "0100." As input bits are mapped to output bits, the encoder remembers the **disparity** from the previous symbol. The disparity is the total number of 0s or 1s by which the signal is out of balance. The encoder then selects either an output pattern or its alternate to reduce the disparity. With 8B/10B, the disparity will be at most 2 bits. Thus, the signal will never be far from balanced. There will also never be more than five consecutive 1s or 0s, to help with clock recovery.

2.5.2 Passband Transmission

Often, we want to use a range of frequencies that does not start at zero to send information across a channel. For wireless channels, it is not practical to send very low frequency signals because the size of the antenna needs to be a fraction of the signal wavelength, which becomes large. In any case, regulatory constraints and the need to avoid interference usually dictate the choice of frequencies. Even for wires, placing a signal in a given frequency band is useful to let different kinds of signals coexist on the channel. This kind of transmission is called passband transmission because an arbitrary band of frequencies is used to pass the signal.

Fortunately, our fundamental results from earlier in the chapter are all in terms of bandwidth, or the width of the frequency band. The absolute frequency values do not matter for capacity. This means that we can take a **baseband** signal that occupies 0 to B Hz and shift it up to occupy a **passband** of S to $S+B$ Hz without changing the amount of information that it can carry, even though the signal will look different. To process a signal at the receiver, we can shift it back down to baseband, where it is more convenient to detect symbols.

Digital modulation is accomplished with passband transmission by regulating or modulating a carrier signal that sits in the passband. We can modulate the amplitude, frequency, or phase of the carrier signal. Each of these methods has a corresponding name. In **ASK (Amplitude Shift Keying)**, two different amplitudes are used to represent 0 and 1. An example with a nonzero and a zero level is shown in Fig. 2-22(b). More than two levels can be used to represent more symbols. Similarly, with **FSK (Frequency Shift Keying)**, two or more different tones are used. The example in Fig. 2-21(c) uses just two frequencies. In the simplest form of **PSK (Phase Shift Keying)**, the carrier wave is systematically shifted 0 or 180 degrees at each symbol period. Because there are two phases, it is called **BPSK (Binary Phase Shift Keying)**. "Binary" here refers to the two symbols, not that the symbols represent 2 bits. An example is shown in Fig. 2-22(c). A better scheme that uses the channel bandwidth more efficiently is to use four shifts, e.g., 45, 135, 225, or 315 degrees, to transmit 2 bits of information per symbol. This version is called **QPSK** (**Quadrature Phase Shift Keying**).

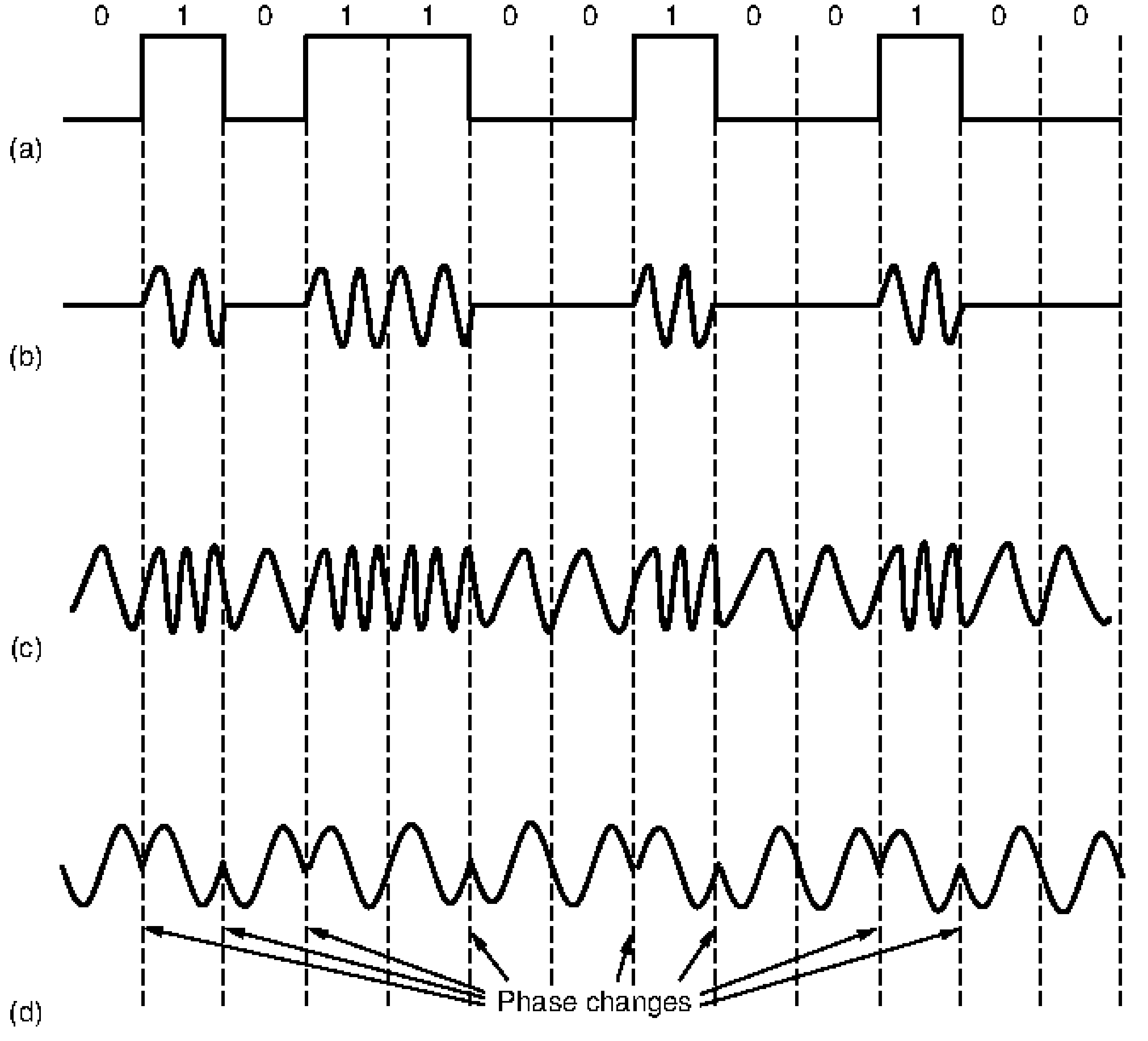


Figure 2-22. (a) A binary signal. (b) Amplitude shift keying. (c) Frequency shift keying. (d) Phase shift keying.

We can combine these schemes and use more levels to transmit more bits per symbol. Only one of frequency and phase can be modulated at a time because they are related, with frequency being the rate of change of phase over time. Usually, amplitude and phase are modulated in combination. Three examples are shown in Fig. 2-23. In each example, the points give the legal amplitude and phase combinations of each symbol. In Fig. 2-23(a), we see equidistant dots at 45, 135, 225, and 315 degrees. The phase of a dot is indicated by the angle a line from it to the origin makes with the positive x-axis. The amplitude of a dot is the distance from the origin. This figure is a representation of QPSK.

This kind of diagram is called a **constellation diagram**. In Fig. 2-23(b) we see a modulation scheme with a denser constellation. Sixteen combinations of amplitudes and phase are used, so the modulation scheme can be used to transmit

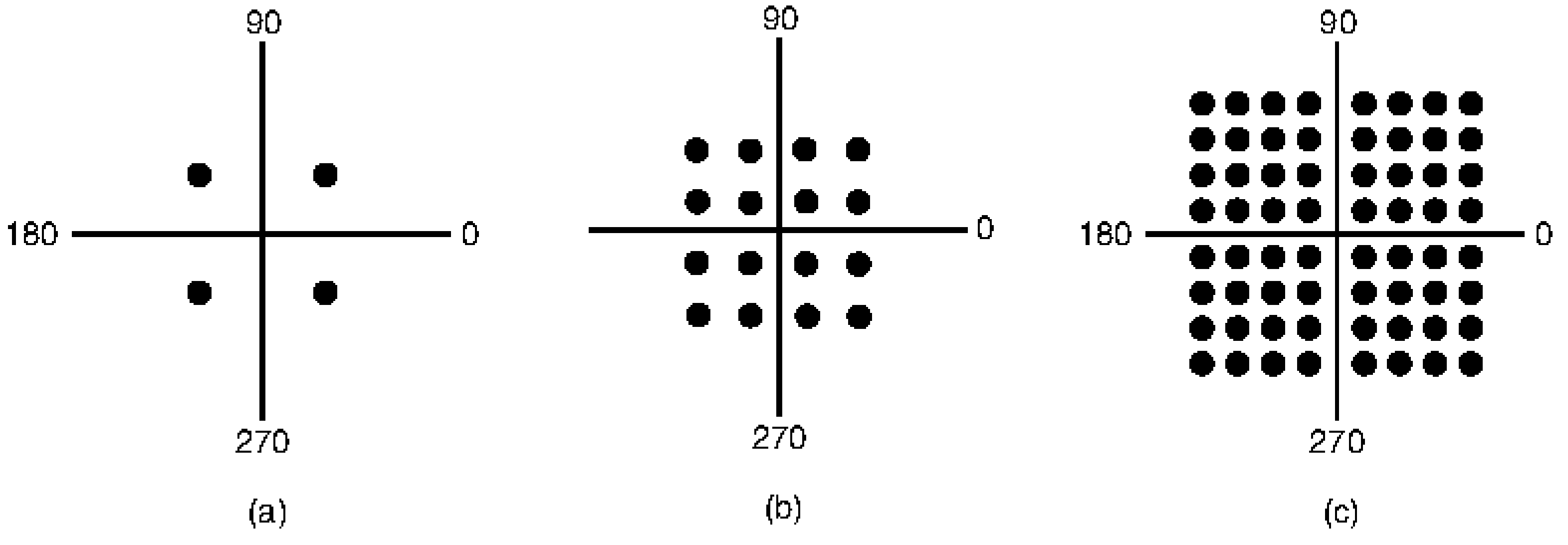


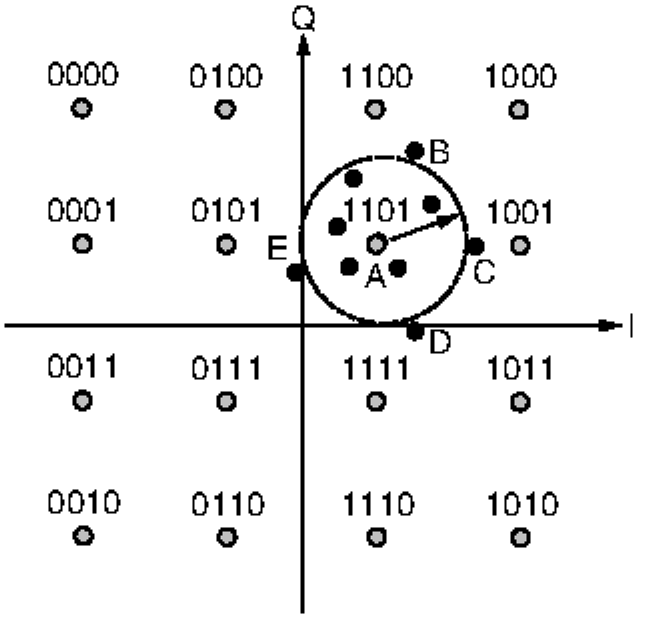
Figure 2-23. (a) QPSK. (b) QAM-16. (c) QAM-64.

4 bits per symbol. It is called **QAM-16**, where QAM stands for **Quadrature Amplitude Modulation**. Figure 2-23(c) is a still denser modulation scheme with 64 different combinations, so 6 bits can be transmitted per symbol. It is called **QAM-64**. Even higher-order QAMs are used too. As you might suspect from these constellations, it is easier to build electronics to produce symbols as a combination of values on each axis than as a combination of amplitude and phase values. That is why the patterns look like squares rather than concentric circles.

The constellations we have seen so far do not show how bits are assigned to symbols. When making the assignment, an important consideration is that a small burst of noise at the receiver not lead to many bit errors. This might happen if we assigned consecutive bit values to adjacent symbols. With QAM-16, for example, if one symbol stood for 0111 and the neighboring symbol stood for 1000, if the receiver mistakenly picks the adjacent symbol it will cause all of the bits to be wrong. A better solution is to map bits to symbols so that adjacent symbols differ in only 1 bit position. This mapping is called a **Gray code**. Fig. 2-24 shows a QAM-16 constellation that has been Gray coded. Now if the receiver decodes the symbol in error, it will make only a single bit error in the expected case that the decoded symbol is close to the transmitted symbol.

2.5.3 Frequency Division Multiplexing

The modulation schemes we have seen let us send one signal to convey bits along a wired or wireless link. However, economies of scale play an important role in how we use networks. It costs essentially the same amount of money to install and maintain a high-bandwidth transmission line as a low-bandwidth line between two different offices (i.e., the costs come from having to dig the trench and not from what kind of cable or fiber goes into it). Consequently, multiplexing schemes have been developed to share lines among many signals.



When 1101 is sent:

Point	Decodes as	Bit errors
A	1101	0
B	1100	1
C	1001	1
D	1111	1
E	0101	1

Figure 2-24. Gray-coded QAM-16.

FDM (Frequency Division Multiplexing) takes advantage of passband transmission to share a channel. It divides the spectrum into frequency bands, with each user having exclusive possession of some band in which to send their signal. AM radio broadcasting illustrates FDM. The allocated spectrum is about 1 MHz, roughly 500 to 1500 kHz. Different frequencies are allocated to different logical channels (stations), each operating in a portion of the spectrum, with the interchannel separation great enough to prevent interference.

For a more detailed example, in Fig. 2-25 we show three voice-grade telephone channels multiplexed using FDM. Filters limit the usable bandwidth to about 3100 Hz per voice-grade channel. When many channels are multiplexed together, 4000 Hz is allocated per channel. The excess is called a **guard band**. It keeps the channels well separated. First the voice channels are raised in frequency, each by a different amount. Then they can be combined because no two channels now occupy the same portion of the spectrum. Notice that even though there are gaps between the channels thanks to the guard bands, there is some overlap between adjacent channels. The overlap is there because real filters do not have ideal sharp edges. This means that a strong spike at the edge of one channel will be felt in the adjacent one as nonthermal noise.

This scheme has been used to multiplex calls in the telephone system for many years, but multiplexing in time is now preferred instead. However, FDM continues to be used in telephone networks, as well as cellular, terrestrial wireless, and satellite networks at a higher level of granularity.

When sending digital data, it is possible to divide the spectrum efficiently without using guard bands. In **OFDM (Orthogonal Frequency Division Multiplexing)**, the channel bandwidth is divided into many subcarriers that independently send data (e.g., with QAM). The subcarriers are packed tightly together in the frequency domain. Thus, signals from each subcarrier extend into adjacent ones. However, as seen in Fig. 2-26, the frequency response of each subcarrier is

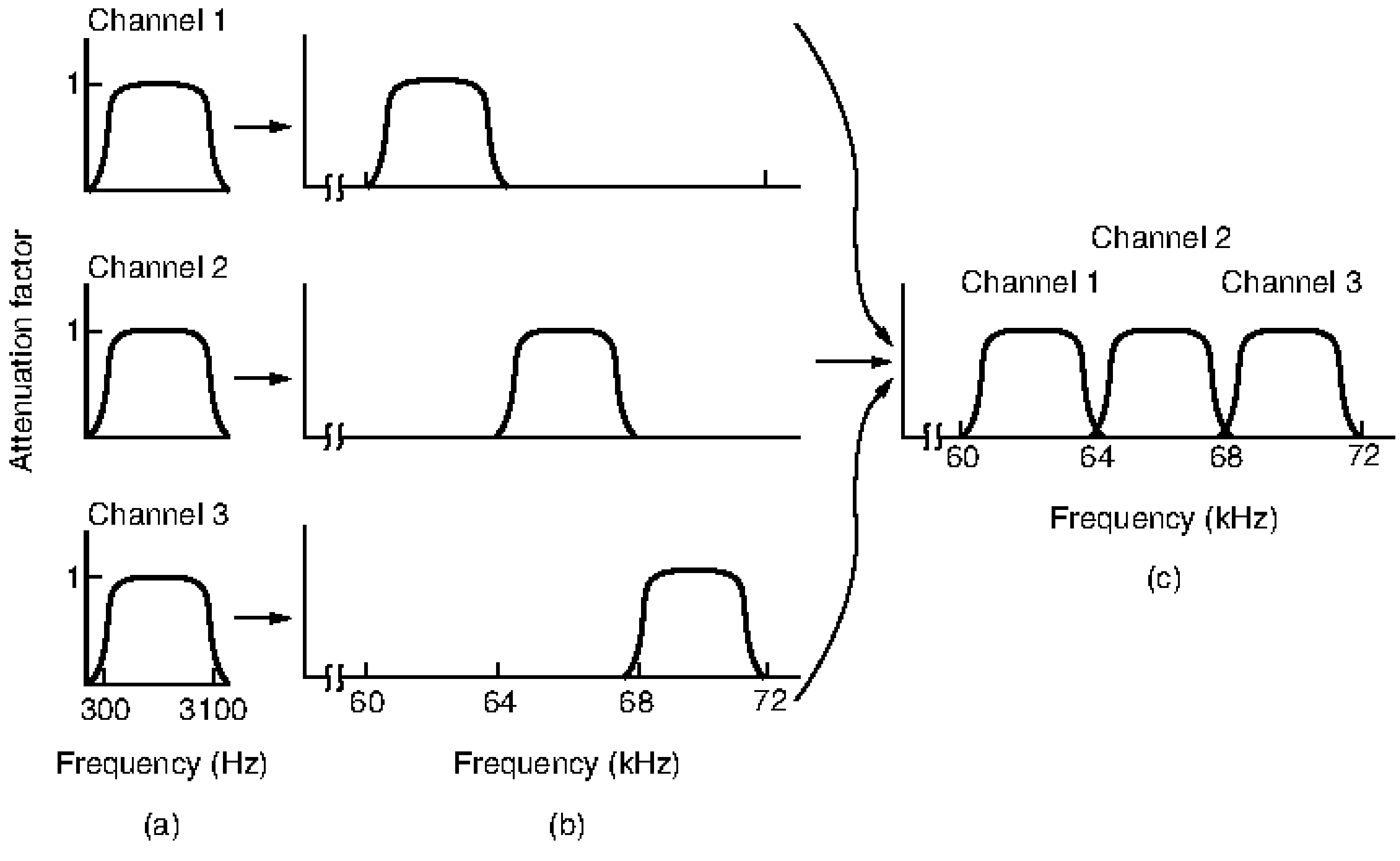


Figure 2-25. Frequency division multiplexing. (a) The original bandwidths. (b) The bandwidths raised in frequency. (c) The multiplexed channel.

designed so that it is zero at the center of the adjacent subcarriers. The subcarriers can therefore be sampled at their center frequencies without interference from their neighbors. To make this work, a guard time is needed to repeat a portion of the symbol signals in time so that they have the desired frequency response. However, this overhead is much less than is needed for many guard bands.

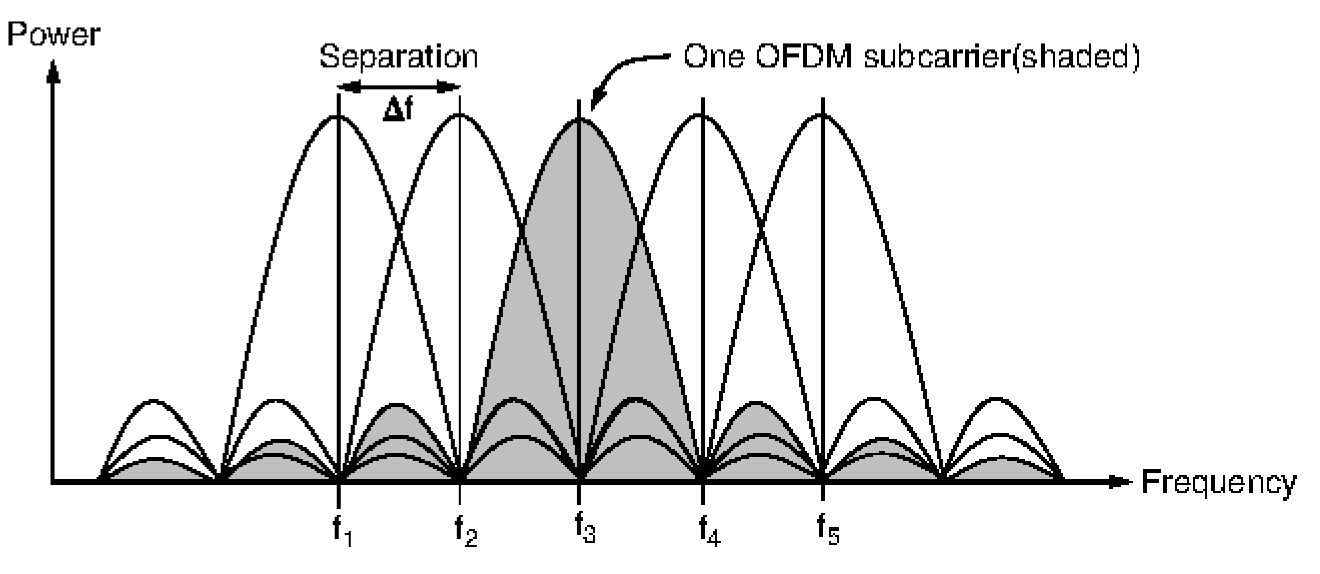


Figure 2-26. Orthogonal frequency division multiplexing (OFDM).

The idea of OFDM has been around for a long time, but it is only in the last decade that it has been widely adopted, following the realization that it is possible to implement OFDM efficiently in terms of a Fourier transform of digital data over all subcarriers (instead of separately modulating each subcarrier). OFDM is used in 802.11, cable networks and power line networking, and is planned for fourth-generation cellular systems. Usually, one high-rate stream of digital information is split into many low-rate streams that are transmitted on the subcarriers in parallel. This division is valuable because degradations of the channel are easier to cope with at the subcarrier level; some subcarriers may be very degraded and excluded in favor of subcarriers that are received well.

2.5.4 Time Division Multiplexing

An alternative to FDM is **TDM** (**Time Division Multiplexing**). Here, the users take turns (in a round-robin fashion), each one periodically getting the entire bandwidth for a little burst of time. An example of three streams being multiplexed with TDM is shown in Fig. 2-27. Bits from each input stream are taken in a fixed **time slot** and output to the aggregate stream. This stream runs at the sum rate of the individual streams. For this to work, the streams must be synchronized in time. Small intervals of **guard time** analogous to a frequency guard band may be added to accommodate small timing variations.

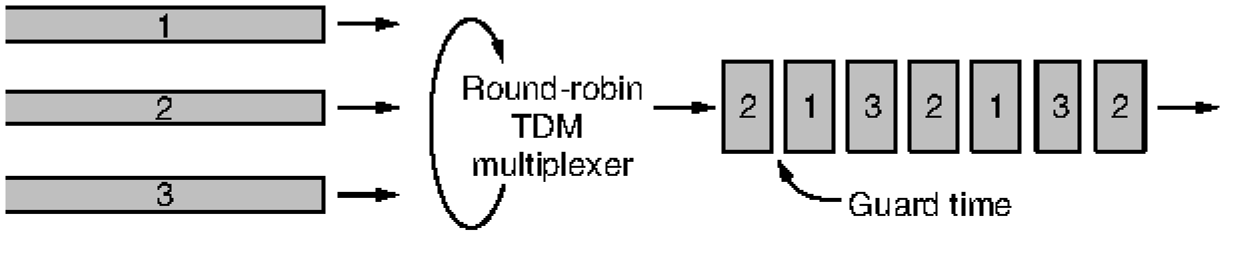


Figure 2-27. Time Division Multiplexing (TDM).

TDM is used widely as part of the telephone and cellular networks. To avoid one point of confusion, let us be clear that it is quite different from the alternative **STDM** (**Statistical Time Division Multiplexing**). The prefix "statistical" is added to indicate that the individual streams contribute to the multiplexed stream *not* on a fixed schedule, but according to the statistics of their demand. STDM is packet switching by another name.

2.5.5 Code Division Multiplexing

There is a third kind of multiplexing that works in a completely different way than FDM and TDM. **CDM** (**Code Division Multiplexing**) is a form of **spread spectrum** communication in which a narrowband signal is spread out over a wider frequency band. This can make it more tolerant of interference, as well as allowing multiple signals from different users to share the same frequency band. Because code division multiplexing is mostly used for the latter purpose it is commonly called **CDMA** (**Code Division Multiple Access**).

CDMA allows each station to transmit over the entire frequency spectrum all the time. Multiple simultaneous transmissions are separated using coding theory. Before getting into the algorithm, let us consider an analogy: an airport lounge with many pairs of people conversing. TDM is comparable to pairs of people in the room taking turns speaking. FDM is comparable to the pairs of people speaking at different pitches, some high-pitched and some low-pitched such that each pair can hold its own conversation at the same time as but independently of the others. CDMA is comparable to each pair of people talking at once, but in a different language. The French-speaking couple just hones in on the French, rejecting everything that is not French as noise. Thus, the key to CDMA is to be able to extract the desired signal while rejecting everything else as random noise. A somewhat simplified description of CDMA follows.

In CDMA, each bit time is subdivided into m short intervals called **chips**. Typically, there are 64 or 128 chips per bit, but in the example given here we will use 8 chips/bit for simplicity. Each station is assigned a unique m-bit code called a **chip sequence**. For pedagogical purposes, it is convenient to use a bipolar notation to write these codes as sequences of −1 and +1. We will show chip sequences in parentheses.

To transmit a 1 bit, a station sends its chip sequence. To transmit a 0 bit, it sends the negation of its chip sequence. No other patterns are permitted. Thus, for $m = 8$, if station A is assigned the chip sequence (−1 −1 −1 +1 +1 −1 +1 +1), it can send a 1 bit by transmiting the chip sequence and a 0 by transmitting (+1 +1 +1 −1 −1 +1 −1 −1). It is really signals with these voltage levels that are sent, but it is sufficient for us to think in terms of the sequences.

Increasing the amount of information to be sent from b bits/sec to mb chips/sec for each station means that the bandwidth needed for CDMA is greater by a factor of m than the bandwidth needed for a station not using CDMA (assuming no changes in the modulation or encoding techniques). If we have a 1-MHz band available for 100 stations, with FDM each one would have 10 kHz and could send at 10 kbps (assuming 1 bit per Hz). With CDMA, each station uses the full 1 MHz, so the chip rate is 100 chips per bit to spread the station's bit rate of 10 kbps across the channel.

In Fig. 2-28(a) and (b) we show the chip sequences assigned to four example stations and the signals that they represent. Each station has its own unique chip sequence. Let us use the symbol $\mathbf{S}$ to indicate the m-chip vector for station S, and $\overline{\mathbf{S}}$ for its negation. All chip sequences are pairwise **orthogonal**, by which we mean that the normalized inner product of any two distinct chip sequences, $\mathbf{S}$ and $\mathbf{T}$ (written as $\mathbf{S}\bullet\mathbf{T}$), is 0. It is known how to generate such orthogonal chip sequences using a method known as **Walsh codes**. In mathematical terms, orthogonality of the chip sequences can be expressed as follows:

$$\mathbf{S}\bullet\mathbf{T} \equiv \frac{1}{m}\sum_{i=1}^{m} S_i T_i = 0 \qquad (2\text{-}5)$$

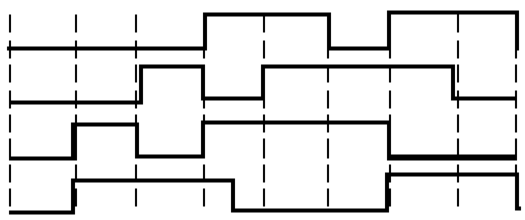
In plain English, as many pairs are the same as are different. This orthogonality property will prove crucial later. Note that if $\mathbf{S} \bullet \mathbf{T} = 0$, then $\mathbf{S} \bullet \overline{\mathbf{T}}$ is also 0. The normalized inner product of any chip sequence with itself is 1:

$$\mathbf{S} \bullet \mathbf{S} = \frac{1}{m} \sum_{i=1}^{m} S_i S_i = \frac{1}{m} \sum_{i=1}^{m} S_i^2 = \frac{1}{m} \sum_{i=1}^{m} (\pm 1)^2 = 1$$

This follows because each of the m terms in the inner product is 1, so the sum is m. Also note that $\mathbf{S} \bullet \overline{\mathbf{S}} = -1$.

A = (−1 −1 −1 +1 +1 −1 +1 +1)
B = (−1 −1 +1 −1 +1 +1 +1 −1)
C = (−1 +1 −1 +1 +1 +1 −1 −1)
D = (−1 +1 −1 −1 −1 −1 +1 −1)

(a)



(b)

$S_1 = C$ = (−1 +1 −1 +1 +1 +1 −1 −1)
$S_2 = B+C$ = (−2 0 0 0 +2 +2 0 −2)
$S_3 = A+\overline{B}$ = (0 0 −2 +2 0 −2 0 +2)
$S_4 = A+\overline{B}+C$ = (−1 +1 −3 +3 +1 −1 −1 +1)
$S_5 = A+B+C+D$ = (−4 0 −2 0 +2 0 +2 −2)
$S_6 = A+B+\overline{C}+D$ = (−2 −2 0 −2 0 −2 +4 0)

(c)

$S_1 \bullet C$ = [1+1−1+1+1+1−1−1]/8 = 1
$S_2 \bullet C$ = [2+0+0+0+2+2+0+2]/8 = 1
$S_3 \bullet C$ = [0+0+2+2+0−2+0−2]/8 = 0
$S_4 \bullet C$ = [1+1+3+3+1−1+1−1]/8 = 1
$S_5 \bullet C$ = [4+0+2+0+2+0−2+2]/8 = 1
$S_6 \bullet C$ = [2−2+0−2+0−2−4+0]/8 = −1

(d)

Figure 2-28. (a) Chip sequences for four stations. (b) Signals the sequences represent (c) Six examples of transmissions. (d) Recovery of station C's signal.

During each bit time, a station can transmit a 1 (by sending its chip sequence), it can transmit a 0 (by sending the negative of its chip sequence), or it can be silent and transmit nothing. We assume for now that all stations are synchronized in time, so all chip sequences begin at the same instant. When two or more stations transmit simultaneously, their bipolar sequences add linearly. For example, if in one chip period three stations output +1 and one station outputs −1, +2 will be received. One can think of this as signals that add as voltages superimposed on the channel: three stations output +1 V and one station outputs −1 V, so that 2 V is received. For instance, in Fig. 2-28(c) we see six examples of one or more stations transmitting 1 bit at the same time. In the first example, *C* transmits a 1 bit, so we just get *C*'s chip sequence. In the second example, both *B* and *C* transmit 1 bits, so we get the sum of their bipolar chip sequences, namely:

$$(-1\ -1\ +1\ -1\ +1\ +1\ +1\ -1) + (-1\ +1\ -1\ +1\ +1\ +1\ -1\ -1) = (-2\ \ 0\ \ 0\ \ 0\ +2\ +2\ \ 0\ -2)$$

To recover the bit stream of an individual station, the receiver must know that station's chip sequence in advance. It does the recovery by computing the normalized inner product of the received chip sequence and the chip sequence of the station whose bit stream it is trying to recover. If the received chip sequence is **S** and the receiver is trying to listen to a station whose chip sequence is **C**, it just computes the normalized inner product, **S**•**C**.

To see why this works, just imagine that two stations, A and C, both transmit a 1 bit at the same time that B transmits a 0 bit, as is the case in the third example. The receiver sees the sum, $\mathbf{S} = \mathbf{A} + \overline{\mathbf{B}} + \mathbf{C}$, and computes

$$\mathbf{S} \bullet \mathbf{C} = (\mathbf{A} + \overline{\mathbf{B}} + \mathbf{C}) \bullet \mathbf{C} = \mathbf{A} \bullet \mathbf{C} + \overline{\mathbf{B}} \bullet \mathbf{C} + \mathbf{C} \bullet \mathbf{C} = 0 + 0 + 1 = 1$$

The first two terms vanish because all pairs of chip sequences have been carefully chosen to be orthogonal, as shown in Eq. (2-5). Now it should be clear why this property must be imposed on the chip sequences.

To make the decoding process more concrete, we show six examples in Fig. 2-28(d). Suppose that the receiver is interested in extracting the bit sent by station C from each of the six signals S_1 through S_6. It calculates the bit by summing the pairwise products of the received $\mathbf{S}$ and the $\mathbf{C}$ vector of Fig. 2-28(a) and then taking 1/8 of the result (since $m = 8$ here). The examples include cases where C is silent, sends a 1 bit, and sends a 0 bit, individually and in combination with other transmissions. As shown, the correct bit is decoded each time. It is just like speaking French.

In principle, given enough computing capacity, the receiver can listen to all the senders at once by running the decoding algorithm for each of them in parallel. In real life, suffice it to say that this is easier said than done, and it is useful to know which senders might be transmitting.

In the ideal, noiseless CDMA system we have studied here, the number of stations that send concurrently can be made arbitrarily large by using longer chip sequences. For 2^n stations, Walsh codes can provide 2^n orthogonal chip sequences of length 2^n. However, one significant limitation is that we have assumed that all the chips are synchronized in time at the receiver. This synchronization is not even approximately true in some applications, such as cellular networks (in which CDMA has been widely deployed starting in the 1990s). It leads to different designs. We will return to this topic later in the chapter and describe how asynchronous CDMA differs from synchronous CDMA.

As well as cellular networks, CDMA is used by satellites and cable networks. We have glossed over many complicating factors in this brief introduction. Engineers who want to gain a deep understanding of CDMA should read Viterbi (1995) and Lee and Miller (1998). These references require quite a bit of background in communication engineering, however.

2.6 THE PUBLIC SWITCHED TELEPHONE NETWORK

When two computers owned by the same company or organization and located close to each other need to communicate, it is often easiest just to run a cable between them. LANs work this way. However, when the distances are large or there are many computers or the cables have to pass through a public road or other public right of way, the costs of running private cables are usually prohibitive.

Furthermore, in just about every country in the world, stringing private transmission lines across (or underneath) public property is also illegal. Consequently, the network designers must rely on the existing telecommunication facilities.

These facilities, especially the **PSTN** (**Public Switched Telephone Network**), were usually designed many years ago, with a completely different goal in mind: transmitting the human voice in a more-or-less recognizable form. Their suitability for use in computer-computer communication is often marginal at best. To see the size of the problem, consider that a cheap commodity cable running between two computers can transfer data at 1 Gbps or more. In contrast, typical ADSL, the blazingly fast alternative to a telephone modem, runs at around 1 Mbps. The difference between the two is the difference between cruising in an airplane and taking a leisurely stroll.

Nonetheless, the telephone system is tightly intertwined with (wide area) computer networks, so it is worth devoting some time to study it in detail. The limiting factor for networking purposes turns out to be the "last mile" over which customers connect, not the trunks and switches inside the telephone network. This situation is changing with the gradual rollout of fiber and digital technology at the edge of the network, but it will take time and money. During the long wait, computer systems designers used to working with systems that give at least three orders of magnitude better performance have devoted much time and effort to figure out how to use the telephone network efficiently.

In the following sections we will describe the telephone system and show how it works. For additional information about the innards of the telephone system see Bellamy (2000).

2.6.1 Structure of the Telephone System

Soon after Alexander Graham Bell patented the telephone in 1876 (just a few hours ahead of his rival, Elisha Gray), there was an enormous demand for his new invention. The initial market was for the sale of telephones, which came in pairs. It was up to the customer to string a single wire between them. If a telephone owner wanted to talk to n other telephone owners, separate wires had to be strung to all n houses. Within a year, the cities were covered with wires passing over houses and trees in a wild jumble. It became immediately obvious that the model of connecting every telephone to every other telephone, as shown in Fig. 2-29(a), was not going to work.

To his credit, Bell saw this problem early on and formed the Bell Telephone Company, which opened its first switching office (in New Haven, Connecticut) in 1878. The company ran a wire to each customer's house or office. To make a call, the customer would crank the phone to make a ringing sound in the telephone company office to attract the attention of an operator, who would then manually connect the caller to the callee by using a short jumper cable to connect the caller to the callee. The model of a single switching office is illustrated in Fig. 2-29(b).

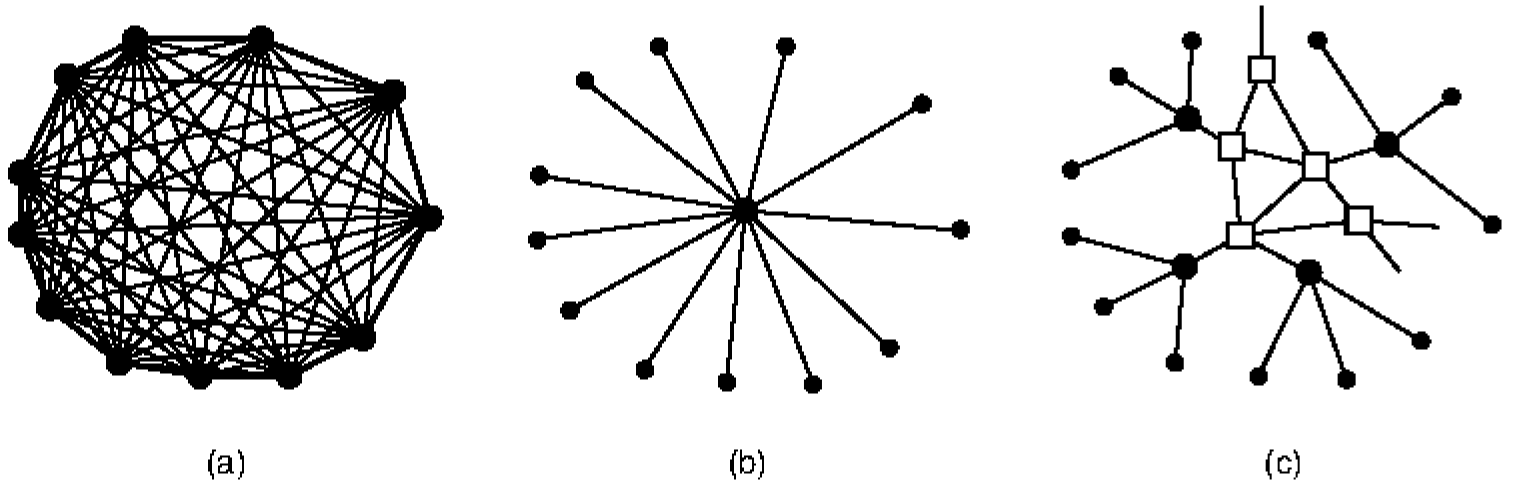


Figure 2-29. (a) Fully interconnected network. (b) Centralized switch. (c) Two-level hierarchy.

Pretty soon, Bell System switching offices were springing up everywhere and people wanted to make long-distance calls between cities, so the Bell System began to connect the switching offices. The original problem soon returned: to connect every switching office to every other switching office by means of a wire between them quickly became unmanageable, so second-level switching offices were invented. After a while, multiple second-level offices were needed, as illustrated in Fig. 2-29(c). Eventually, the hierarchy grew to five levels.

By 1890, the three major parts of the telephone system were in place: the switching offices, the wires between the customers and the switching offices (by now balanced, insulated, twisted pairs instead of open wires with an earth return), and the long-distance connections between the switching offices. For a short technical history of the telephone system, see Hawley (1991).

While there have been improvements in all three areas since then, the basic Bell System model has remained essentially intact for over 100 years. The following description is highly simplified but gives the essential flavor nevertheless. Each telephone has two copper wires coming out of it that go directly to the telephone company's nearest **end office** (also called a **local central office**). The distance is typically 1 to 10 km, being shorter in cities than in rural areas. In the United States alone there are about 22,000 end offices. The two-wire connections between each subscriber's telephone and the end office are known in the trade as the **local loop**. If the world's local loops were stretched out end to end, they would extend to the moon and back 1000 times.

At one time, 80% of AT&T's capital value was the copper in the local loops. AT&T was then, in effect, the world's largest copper mine. Fortunately, this fact was not well known in the investment community. Had it been known, some corporate raider might have bought AT&T, ended all telephone service in the United States, ripped out all the wire, and sold it to a copper refiner for a quick payback.

If a subscriber attached to a given end office calls another subscriber attached to the same end office, the switching mechanism within the office sets up a direct electrical connection between the two local loops. This connection remains intact for the duration of the call.

If the called telephone is attached to another end office, a different procedure has to be used. Each end office has a number of outgoing lines to one or more nearby switching centers, called **toll offices** (or, if they are within the same local area, **tandem offices**). These lines are called **toll connecting trunks**. The number of different kinds of switching centers and their topology varies from country to country depending on the country's telephone density.

If both the caller's and callee's end offices happen to have a toll connecting trunk to the same toll office (a likely occurrence if they are relatively close by), the connection may be established within the toll office. A telephone network consisting only of telephones (the small dots), end offices (the large dots), and toll offices (the squares) is shown in Fig. 2-29(c).

If the caller and callee do not have a toll office in common, a path will have to be established between two toll offices. The toll offices communicate with each other via high-bandwidth **intertoll trunks** (also called **interoffice trunks**). Prior to the 1984 breakup of AT&T, the U.S. telephone system used hierarchical routing to find a path, going to higher levels of the hierarchy until there was a switching office in common. This was then replaced with more flexible, nonhierarchical routing. Figure 2-30 shows how a long-distance connection might be routed.

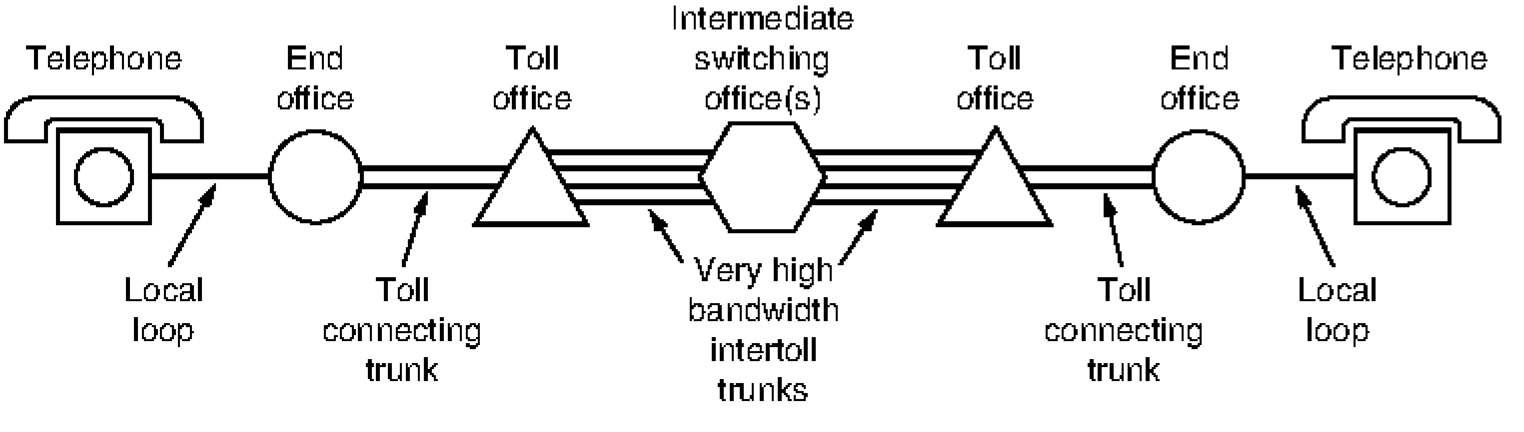


Figure 2-30. A typical circuit route for a long-distance call.

A variety of transmission media are used for telecommunication. Unlike modern office buildings, where the wiring is commonly Category 5, local loops to homes mostly consist of Category 3 twisted pairs, with fiber just starting to appear. Between switching offices, coaxial cables, microwaves, and especially fiber optics are widely used.

In the past, transmission throughout the telephone system was analog, with the actual voice signal being transmitted as an electrical voltage from source to destination. With the advent of fiber optics, digital electronics, and computers, all the trunks and switches are now digital, leaving the local loop as the last piece of analog technology in the system. Digital transmission is preferred because it is not necessary to accurately reproduce an analog waveform after it has passed through many amplifiers on a long call. Being able to correctly distinguish a 0 from a 1 is enough. This property makes digital transmission more reliable than analog. It is also cheaper and easier to maintain.

In summary, the telephone system consists of three major components:

1. Local loops (analog twisted pairs going to houses and businesses).
2. Trunks (digital fiber optic links connecting the switching offices).
3. Switching offices (where calls are moved from one trunk to another).

After a short digression on the politics of telephones, we will come back to each of these three components in some detail. The local loops provide everyone access to the whole system, so they are critical. Unfortunately, they are also the weakest link in the system. For the long-haul trunks, the main issue is how to collect multiple calls together and send them out over the same fiber. This calls for multiplexing, and we apply FDM and TDM to do it. Finally, there are two fundamentally different ways of doing switching; we will look at both.

2.6.2 The Politics of Telephones

For decades prior to 1984, the Bell System provided both local and long-distance service throughout most of the United States. In the 1970s, the U.S. Federal Government came to believe that this was an illegal monopoly and sued to break it up. The government won, and on January 1, 1984, AT&T was broken up into AT&T Long Lines, 23 **BOCs** (**Bell Operating Companies**), and a few other pieces. The 23 BOCs were grouped into seven regional BOCs (RBOCs) to make them economically viable. The entire nature of telecommunication in the United States was changed overnight by court order (*not* by an act of Congress).

The exact specifications of the divestiture were described in the so-called **MFJ** (**Modified Final Judgment**), an oxymoron if ever there was one—if the judgment could be modified, it clearly was not final. This event led to increased competition, better service, and lower long-distance rates for consumers and businesses. However, prices for local service rose as the cross subsidies from long-distance calling were eliminated and local service had to become self supporting. Many other countries have now introduced competition along similar lines.

Of direct relevance to our studies is that the new competitive framework caused a key technical feature to be added to the architecture of the telephone network. To make it clear who could do what, the United States was divided up into 164 **LATAs** (**Local Access and Transport Areas**). Very roughly, a LATA is about as big as the area covered by one area code. Within each LATA, there was one **LEC** (**Local Exchange Carrier**) with a monopoly on traditional telephone service within its area. The most important LECs were the BOCs, although some LATAs contained one or more of the 1500 independent telephone companies operating as LECs.

The new feature was that all inter-LATA traffic was handled by a different kind of company, an **IXC** (**IntereXchange Carrier**). Originally, AT&T Long Lines was the only serious IXC, but now there are well-established competitors such as Verizon and Sprint in the IXC business. One of the concerns at the breakup was to ensure that all the IXCs would be treated equally in terms of line quality, tariffs, and the number of digits their customers would have to dial to use them. The way this is handled is illustrated in Fig. 2-31. Here we see three example LATAs, each with several end offices. LATAs 2 and 3 also have a small hierarchy with tandem offices (intra-LATA toll offices).

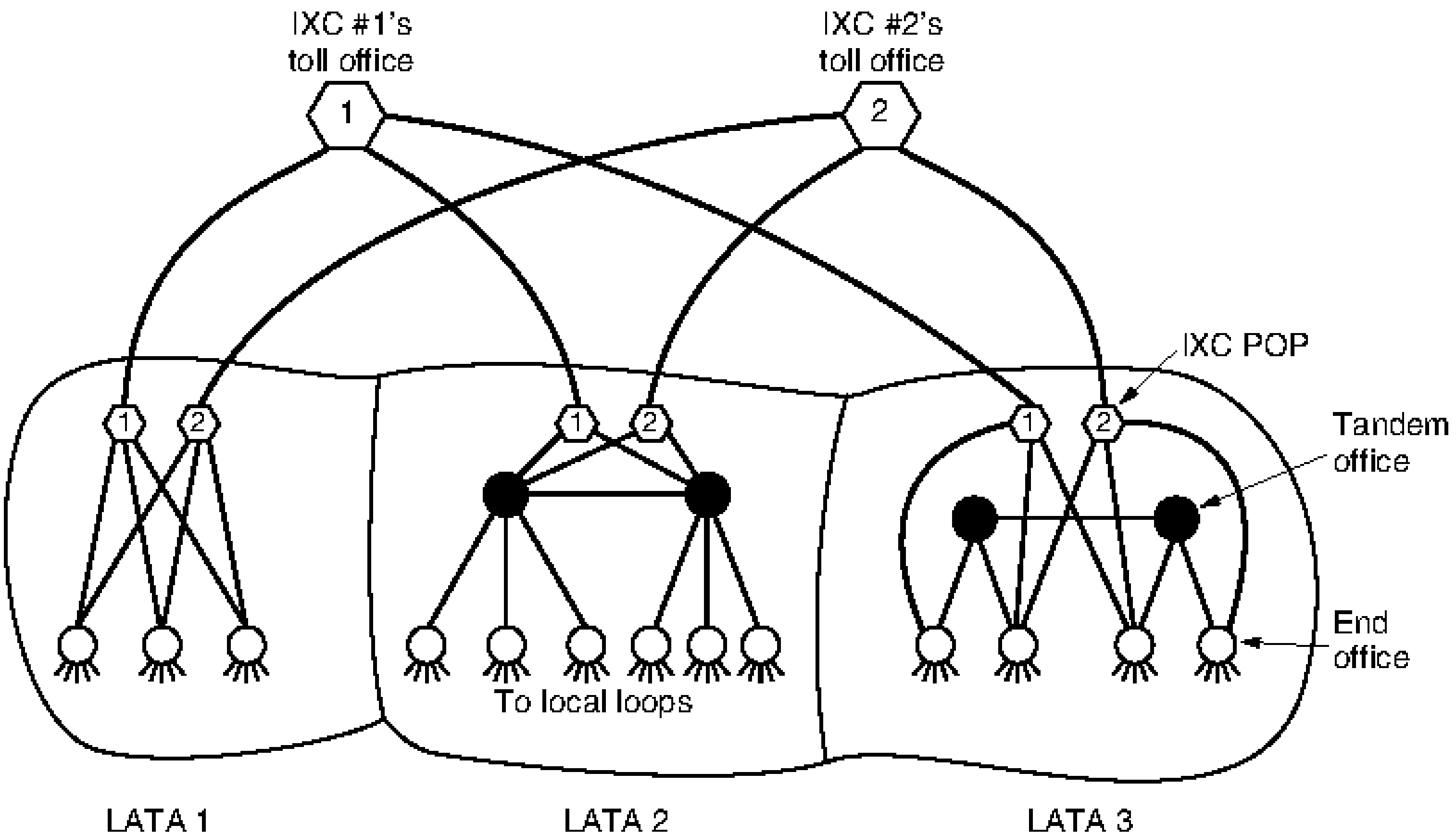


Figure 2-31. The relationship of LATAs, LECs, and IXCs. All the circles are LEC switching offices. Each hexagon belongs to the IXC whose number is in it.

Any IXC that wishes to handle calls originating in a LATA can build a switching office called a **POP** (**Point of Presence**) there. The LEC is required to connect each IXC to every end office, either directly, as in LATAs 1 and 3, or indirectly, as in LATA 2. Furthermore, the terms of the connection, both technical and financial, must be identical for all IXCs. This requirement enables, a subscriber in, say, LATA 1, to choose which IXC to use for calling subscribers in LATA 3.

As part of the MFJ, the IXCs were forbidden to offer local telephone service and the LECs were forbidden to offer inter-LATA telephone service, although both were free to enter any other business, such as operating fried chicken restaurants. In 1984, that was a fairly unambiguous statement. Unfortunately, technology has a funny way of making the law obsolete. Neither cable television nor mobile phones were covered by the agreement. As cable television went from one way to two way and mobile phones exploded in popularity, both LECs and IXCs began buying up or merging with cable and mobile operators.

By 1995, Congress saw that trying to maintain a distinction between the various kinds of companies was no longer tenable and drafted a bill to preserve accessibility for competition but allow cable TV companies, local telephone companies, long-distance carriers, and mobile operators to enter one another's businesses. The idea was that any company could then offer its customers a single integrated package containing cable TV, telephone, and information services and that different companies would compete on service and price. The bill was enacted into law in February 1996 as a major overhaul of telecommunications regulation. As a result, some BOCs became IXCs and some other companies, such as cable television operators, began offering local telephone service in competition with the LECs.

One interesting property of the 1996 law is the requirement that LECs implement **local number portability**. This means that a customer can change local telephone companies without having to get a new telephone number. Portability for mobile phone numbers (and between fixed and mobile lines) followed suit in 2003. These provisions removed a huge hurdle for many people, making them much more inclined to switch LECs. As a result, the U.S. telecommunications landscape became much more competitive, and other countries have followed suit. Often other countries wait to see how this kind of experiment works out in the U.S. If it works well, they do the same thing; if it works badly, they try something else.

2.6.3 The Local Loop: Modems, ADSL, and Fiber

It is now time to start our detailed study of how the telephone system works. Let us begin with the part that most people are familiar with: the two-wire local loop coming from a telephone company end office into houses. The local loop is also frequently referred to as the "last mile," although the length can be up to several miles. It has carried analog information for over 100 years and is likely to continue doing so for some years to come, due to the high cost of converting to digital.

Much effort has been devoted to squeezing data networking out of the copper local loops that are already deployed. Telephone modems send digital data between computers over the narrow channel the telephone network provides for a voice call. They were once widely used, but have been largely displaced by broadband technologies such as ADSL that. reuse the local loop to send digital data from a customer to the end office, where they are siphoned off to the Internet.

Both modems and ADSL must deal with the limitations of old local loops: relatively narrow bandwidth, attenuation and distortion of signals, and susceptibility to electrical noise such as crosstalk.

In some places, the local loop has been modernized by installing optical fiber to (or very close to) the home. Fiber is the way of the future. These installations support computer networks from the ground up, with the local loop having ample bandwidth for data services. The limiting factor is what people will pay, not the physics of the local loop.

In this section we will study the local loop, both old and new. We will cover telephone modems, ADSL, and fiber to the home.

Telephone Modems

To send bits over the local loop, or any other physical channel for that matter, they must be converted to analog signals that can be transmitted over the channel. This conversion is accomplished using the methods for digital modulation that we studied in the previous section. At the other end of the channel, the analog signal is converted back to bits.

A device that converts between a stream of digital bits and an analog signal that represents the bits is called a **modem**, which is short for "*mo*dulator *dem*odulator." Modems come in many varieties: telephone modems, DSL modems, cable modems, wireless modems, etc. The modem may be built into the computer (which is now common for telephone modems) or be a separate box (which is common for DSL and cable modems). Logically, the modem is inserted between the (digital) computer and the (analog) telephone system, as seen in Fig. 2-32.

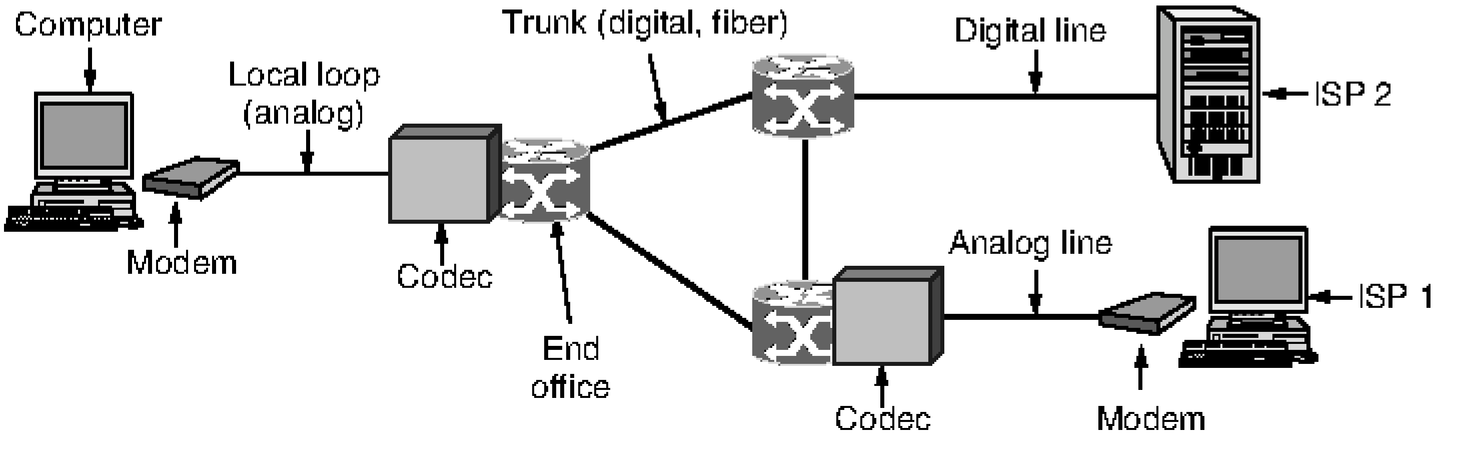


Figure 2-32. The use of both analog and digital transmission for a computer-to-computer call. Conversion is done by the modems and codecs.

Telephone modems are used to send bits between two computers over a voice-grade telephone line, in place of the conversation that usually fills the line. The main difficulty in doing so is that a voice-grade telephone line is limited to 3100 Hz, about what is sufficient to carry a conversation. This bandwidth is more than four orders of magnitude less than the bandwidth that is used for Ethernet or

802.11 (WiFi). Unsurprisingly, the data rates of telephone modems are also four orders of magnitude less than that of Ethernet and 802.11.

Let us run the numbers to see why this is the case. The Nyquist theorem tells us that even with a perfect 3000-Hz line (which a telephone line is decidedly not), there is no point in sending symbols at a rate faster than 6000 baud. In practice, most modems send at a rate of 2400 symbols/sec, or 2400 baud, and focus on getting multiple bits per symbol while allowing traffic in both directions at the same time (by using different frequencies for different directions).

The humble 2400-bps modem uses 0 volts for a logical 0 and 1 volt for a logical 1, with 1 bit per symbol. One step up, it can use four different symbols, as in the four phases of QPSK, so with 2 bits/symbol it can get a data rate of 4800 bps.

A long progression of higher rates has been achieved as technology has improved. Higher rates require a larger set of symbols or **constellation**. With many symbols, even a small amount of noise in the detected amplitude or phase can result in an error. To reduce the chance of errors, standards for the higher-speed modems use some of the symbols for error correction. The schemes are known as **TCM** (**Trellis Coded Modulation**) (Ungerboeck, 1987).

The **V.32** modem standard uses 32 constellation points to transmit 4 data bits and 1 check bit per symbol at 2400 baud to achieve 9600 bps with error correction. The next step above 9600 bps is 14,400 bps. It is called **V.32 bis** and transmits 6 data bits and 1 check bit per symbol at 2400 baud. Then comes **V.34**, which achieves 28,800 bps by transmitting 12 data bits/symbol at 2400 baud. The constellation now has thousands of points. The final modem in this series is **V.34 bis** which uses 14 data bits/symbol at 2400 baud to achieve 33,600 bps.

Why stop here? The reason that standard modems stop at 33,600 is that the Shannon limit for the telephone system is about 35 kbps based on the average length of local loops and the quality of these lines. Going faster than this would violate the laws of physics (department of thermodynamics).

However, there is one way we can change the situation. At the telephone company end office, the data are converted to digital form for transmission within the telephone network (the core of the telephone network converted from analog to digital long ago). The 35-kbps limit is for the situation in which there are two local loops, one at each end. Each of these adds noise to the signal. If we could get rid of one of these local loops, we would increase the SNR and the maximum rate would be doubled.

This approach is how 56-kbps modems are made to work. One end, typically an ISP, gets a high-quality digital feed from the nearest end office. Thus, when one end of the connection is a high-quality signal, as it is with most ISPs now, the maximum data rate can be as high as 70 kbps. Between two home users with modems and analog lines, the maximum is still 33.6 kbps.

The reason that 56-kbps modems (rather than 70-kbps modems) are in use has to do with the Nyquist theorem. A telephone channel is carried inside the telephone system as digital samples. Each telephone channel is 4000 Hz wide when the guard bands are included. The number of samples per second needed to reconstruct it is thus 8000. The number of bits per sample in the U.S. is 8, one of which may be used for control purposes, allowing 56,000 bits/sec of user data. In Europe, all 8 bits are available to users, so 64,000-bit/sec modems could have been used, but to get international agreement on a standard, 56,000 was chosen.

The end result is the **V.90** and **V.92** modem standards. They provide for a 56-kbps downstream channel (ISP to user) and a 33.6-kbps and 48-kbps upstream channel (user to ISP), respectively. The asymmetry is because there is usually more data transported from the ISP to the user than the other way. It also means that more of the limited bandwidth can be allocated to the downstream channel to increase the chances of it actually working at 56 kbps.

Digital Subscriber Lines

When the telephone industry finally got to 56 kbps, it patted itself on the back for a job well done. Meanwhile, the cable TV industry was offering speeds up to 10 Mbps on shared cables. As Internet access became an increasingly important part of their business, the telephone companies (LECs) began to realize they needed a more competitive product. Their answer was to offer new digital services over the local loop.

Initially, there were many overlapping high-speed offerings, all under the general name of **xDSL** (**Digital Subscriber Line**), for various *x*. Services with more bandwidth than standard telephone service are sometimes called **broadband**, although the term really is more of a marketing concept than a specific technical concept. Later, we will discuss what has become the most popular of these services, **ADSL** (**Asymmetric DSL**). We will also use the term DSL or xDSL as shorthand for all flavors.

The reason that modems are so slow is that telephones were invented for carrying the human voice and the entire system has been carefully optimized for this purpose. Data have always been stepchildren. At the point where each local loop terminates in the end office, the wire runs through a filter that attenuates all frequencies below 300 Hz and above 3400 Hz. The cutoff is not sharp—300 Hz and 3400 Hz are the 3-dB points—so the bandwidth is usually quoted as 4000 Hz even though the distance between the 3 dB points is 3100 Hz. Data on the wire are thus also restricted to this narrow band.

The trick that makes xDSL work is that when a customer subscribes to it, the incoming line is connected to a different kind of switch, one that does not have this filter, thus making the entire capacity of the local loop available. The limiting factor then becomes the physics of the local loop, which supports roughly 1 MHz, not the artificial 3100 Hz bandwidth created by the filter.

Unfortunately, the capacity of the local loop falls rather quickly with distance from the end office as the signal is increasingly degraded along the wire. It also depends on the thickness and general quality of the twisted pair. A plot of the potential bandwidth as a function of distance is given in Fig. 2-33. This figure assumes that all the other factors are optimal (new wires, modest bundles, etc.).

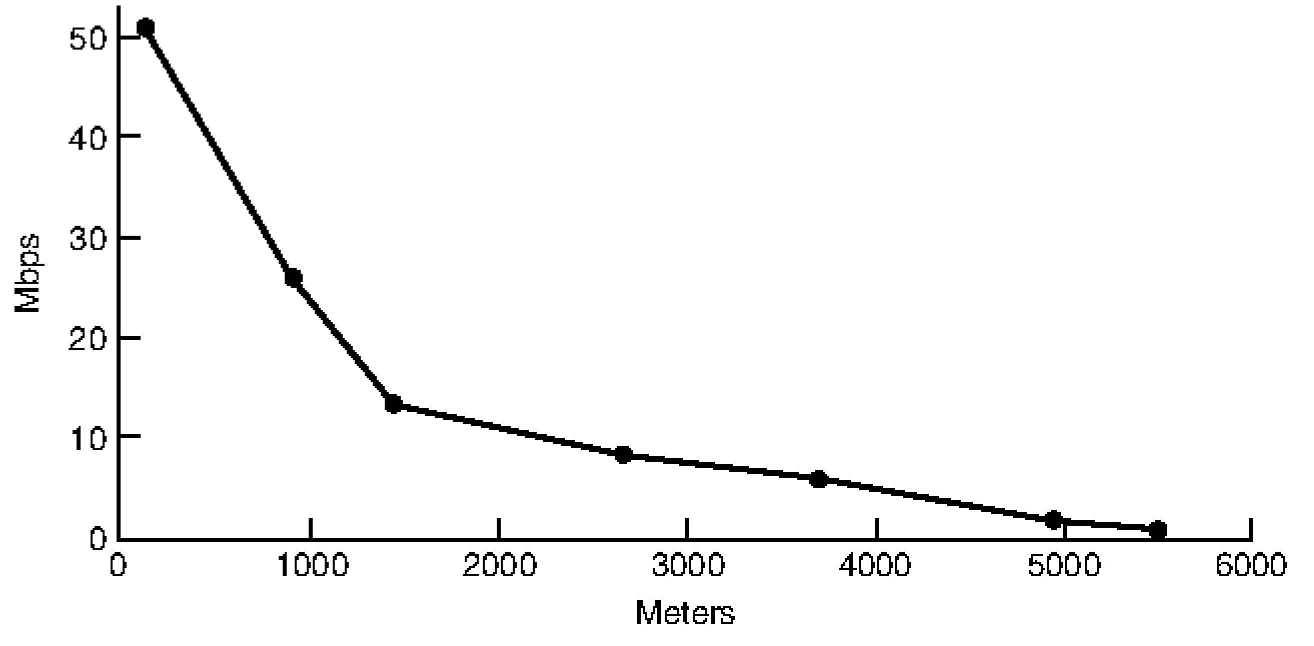


Figure 2-33. Bandwidth versus distance over Category 3 UTP for DSL.

The implication of this figure creates a problem for the telephone company. When it picks a speed to offer, it is simultaneously picking a radius from its end offices beyond which the service cannot be offered. This means that when distant customers try to sign up for the service, they may be told "Thanks a lot for your interest, but you live 100 meters too far from the nearest end office to get this service. Could you please move?" The lower the chosen speed is, the larger the radius and the more customers are covered. But the lower the speed, the less attractive the service is and the fewer the people who will be willing to pay for it. This is where business meets technology.

The xDSL services have all been designed with certain goals in mind. First, the services must work over the existing Category 3 twisted pair local loops. Second, they must not affect customers' existing telephones and fax machines. Third, they must be much faster than 56 kbps. Fourth, they should be always on, with just a monthly charge and no per-minute charge.

To meet the technical goals, the available 1.1 MHz spectrum on the local loop is divided into 256 independent channels of 4312.5 Hz each. This arrangement is shown in Fig. 2-34. The OFDM scheme, which we saw in the previous section, is used to send data over these channels, though it is often called **DMT** (**Discrete MultiTone**) in the context of ADSL. Channel 0 is used for **POTS** (**Plain Old Telephone Service**). Channels 1–5 are not used, to keep the voice and data signals from interfering with each other. Of the remaining 250 channels, one is used for upstream control and one is used for downstream control. The rest are available for user data.

In principle, each of the remaining channels can be used for a full-duplex data stream, but harmonics, crosstalk, and other effects keep practical systems well

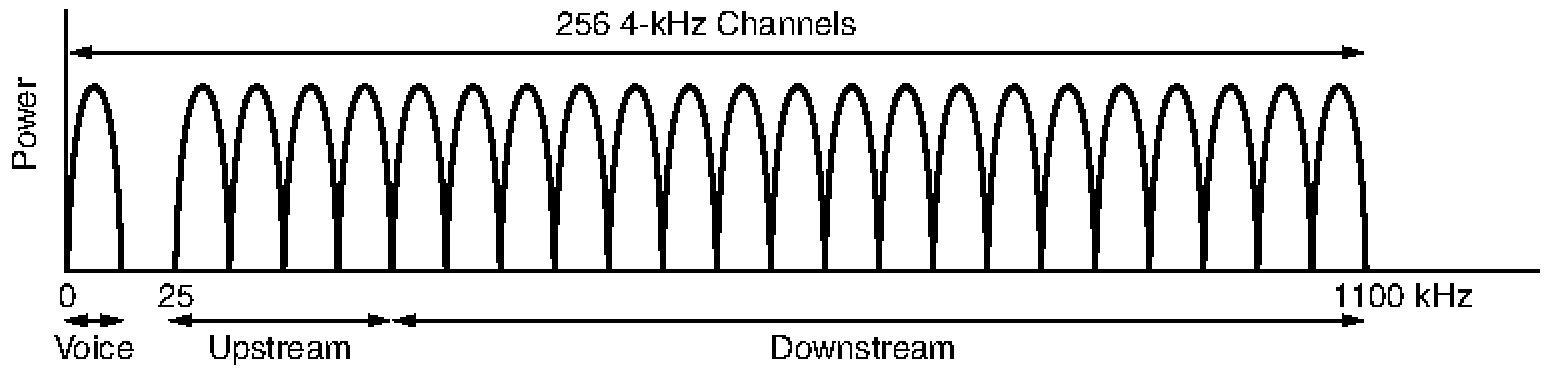


Figure 2-34. Operation of ADSL using discrete multitone modulation.

below the theoretical limit. It is up to the provider to determine how many channels are used for upstream and how many for downstream. A 50/50 mix of upstream and downstream is technically possible, but most providers allocate something like 80–90% of the bandwidth to the downstream channel since most users download more data than they upload. This choice gives rise to the "A" in ADSL. A common split is 32 channels for upstream and the rest downstream. It is also possible to have a few of the highest upstream channels be bidirectional for increased bandwidth, although making this optimization requires adding a special circuit to cancel echoes.

The international ADSL standard, known as **G.dmt**, was approved in 1999. It allows speeds of as much as 8 Mbps downstream and 1 Mbps upstream. It was superseded by a second generation in 2002, called ADSL2, with various improvements to allow speeds of as much as 12 Mbps downstream and 1 Mbps upstream. Now we have ADSL2+, which doubles the downstream speed to 24 Mbps by doubling the bandwidth to use 2.2 MHz over the twisted pair.

However, the numbers quoted here are best-case speeds for good lines close (within 1 to 2 km) to the exchange. Few lines support these rates, and few providers offer these speeds. Typically, providers offer something like 1 Mbps downstream and 256 kbps upstream (standard service), 4 Mbps downstream and 1 Mbps upstream (improved service), and 8 Mbps downstream and 2 Mbps upstream (premium service).

Within each channel, QAM modulation is used at a rate of roughly 4000 symbols/sec. The line quality in each channel is constantly monitored and the data rate is adjusted by using a larger or smaller constellation, like those in Fig. 2-23. Different channels may have different data rates, with up to 15 bits per symbol sent on a channel with a high SNR, and down to 2, 1, or no bits per symbol sent on a channel with a low SNR depending on the standard.

A typical ADSL arrangement is shown in Fig. 2-35. In this scheme, a telephone company technician must install a **NID** (**Network Interface Device**) on the customer's premises. This small plastic box marks the end of the telephone company's property and the start of the customer's property. Close to the NID (or sometimes combined with it) is a **splitter**, an analog filter that separates the

0–4000-Hz band used by POTS from the data. The POTS signal is routed to the existing telephone or fax machine. The data signal is routed to an ADSL modem, which uses digital signal processing to implement OFDM. Since most ADSL modems are external, the computer must be connected to them at high speed. Usually, this is done using Ethernet, a USB cable, or 802.11.

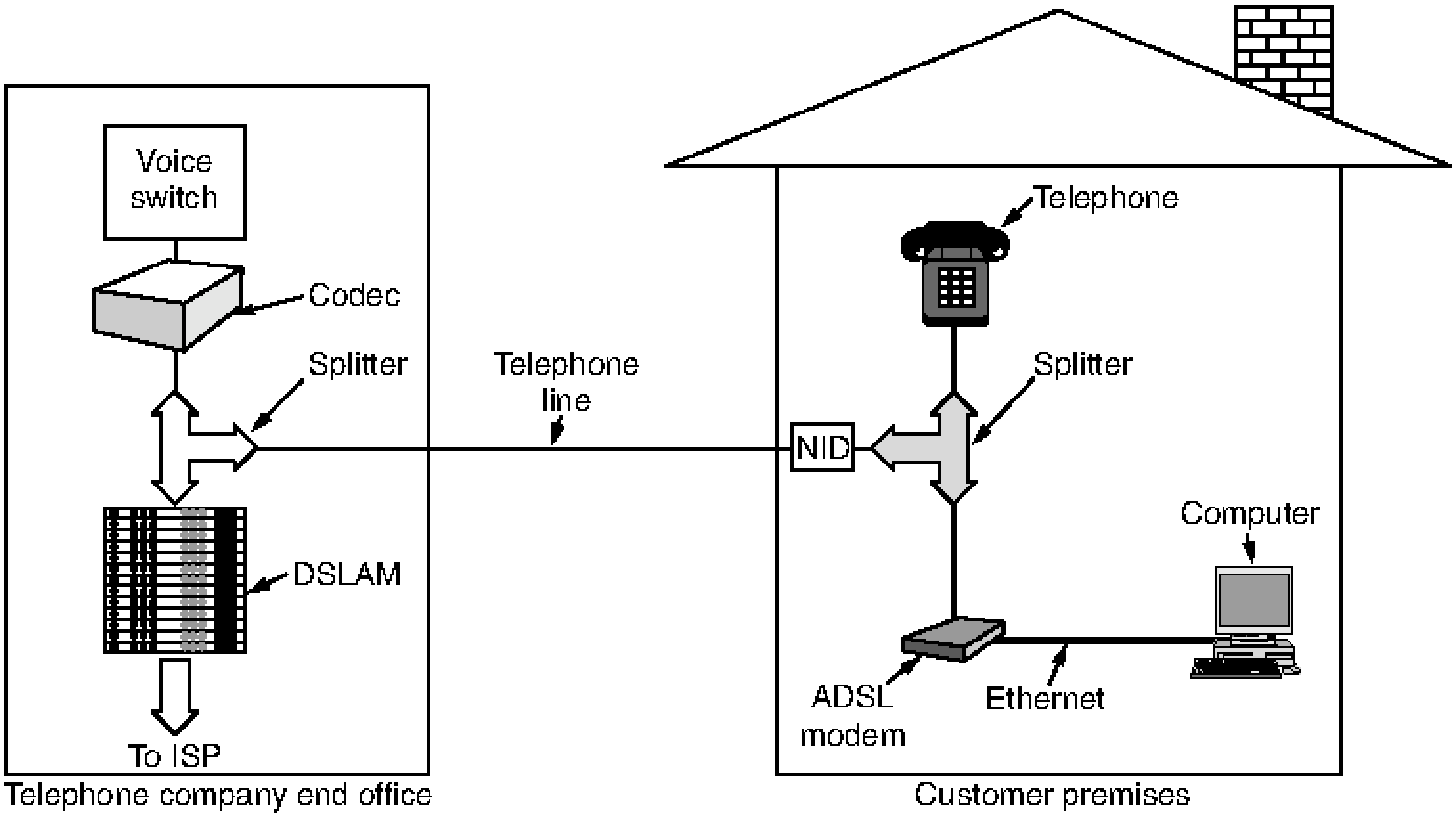


Figure 2-35. A typical ADSL equipment configuration.

At the other end of the wire, on the end office side, a corresponding splitter is installed. Here, the voice portion of the signal is filtered out and sent to the normal voice switch. The signal above 26 kHz is routed to a new kind of device called a **DSLAM** (**Digital Subscriber Line Access Multiplexer**), which contains the same kind of digital signal processor as the ADSL modem. Once the bits have been recovered from the signal, packets are formed and sent off to the ISP.

This complete separation between the voice system and ADSL makes it relatively easy for a telephone company to deploy ADSL. All that is needed is buying a DSLAM and splitter and attaching the ADSL subscribers to the splitter. Other high-bandwidth services (e.g., ISDN) require much greater changes to the existing switching equipment.

One disadvantage of the design of Fig. 2-35 is the need for a NID and splitter on the customer's premises. Installing these can only be done by a telephone company technician, necessitating an expensive "truck roll" (i.e., sending a technician to the customer's premises). Therefore, an alternative, splitterless design, informally called **G.lite**, has also been standardized. It is the same as Fig. 2-35 but without the customer's splitter. The existing telephone line is used as is. The only difference is that a microfilter has to be inserted into each telephone jack between the telephone or ADSL modem and the wire. The microfilter for the telephone is a low-pass filter eliminating frequencies above 3400 Hz; the microfilter for the ADSL modem is a high-pass filter eliminating frequencies below 26 kHz. However, this system is not as reliable as having a splitter, so G.lite can be used only up to 1.5 Mbps (versus 8 Mbps for ADSL with a splitter). For more information about ADSL, see Starr (2003).

Fiber To The Home

Deployed copper local loops limit the performance of ADSL and telephone modems. To let them provide faster and better network services, telephone companies are upgrading local loops at every opportunity by installing optical fiber all the way to houses and offices. The result is called **FttH** (**Fiber To The Home**). While FttH technology has been available for some time, deployments only began to take off in 2005 with growth in the demand for high-speed Internet from customers used to DSL and cable who wanted to download movies. Around 4% of U.S. houses are now connected to FttH with Internet access speeds of up to 100 Mbps.

Several variations of the form "FttX" (where *X* stands for the basement, curb, or neighborhood) exist. They are used to note that the fiber deployment may reach close to the house. In this case, copper (twisted pair or coaxial cable) provides fast enough speeds over the last short distance. The choice of how far to lay the fiber is an economic one, balancing cost with expected revenue. In any case, the point is that optical fiber has crossed the traditional barrier of the "last mile." We will focus on FttH in our discussion.

Like the copper wires before it, the fiber local loop is passive. This means no powered equipment is required to amplify or otherwise process signals. The fiber simply carries signals between the home and the end office. This in turn reduces cost and improves reliability.

Usually, the fibers from the houses are joined together so that only a single fiber reaches the end office per group of up to 100 houses. In the downstream direction, optical splitters divide the signal from the end office so that it reaches all the houses. Encryption is needed for security if only one house should be able to decode the signal. In the upstream direction, optical combiners merge the signals from the houses into a single signal that is received at the end office.

This architecture is called a **PON** (**Passive Optical Network**), and it is shown in Fig. 2-36. It is common to use one wavelength shared between all the houses for downstream transmission, and another wavelength for upstream transmission.

Even with the splitting, the tremendous bandwidth and low attenuation of fiber mean that PONs can provide high rates to users over distances of up to 20 km. The actual data rates and other details depend on the type of PON. Two kinds are common. **GPONs** (**Gigabit-capable PONs**) come from the world of telecommunications, so they are defined by an ITU standard. **EPONs** (**Ethernet PONs**)

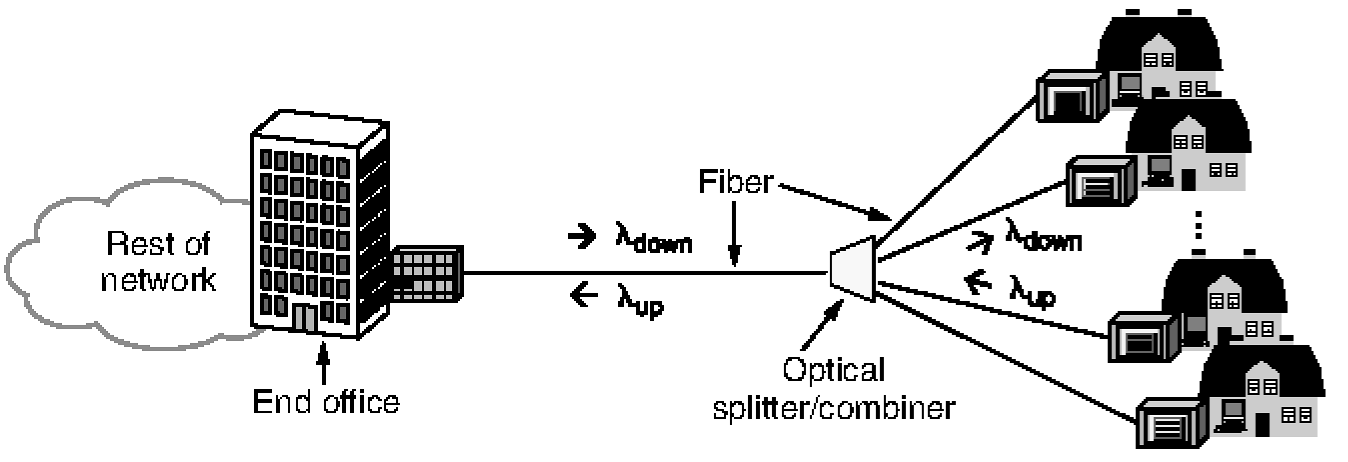


Figure 2-36. Passive optical network for Fiber To The Home.

are more in tune with the world of networking, so they are defined by an IEEE standard. Both run at around a gigabit and can carry traffic for different services, including Internet, video, and voice. For example, GPONs provide 2.4 Gbps downstream and 1.2 or 2.4 Gbps upstream.

Some protocol is needed to share the capacity of the single fiber at the end office between the different houses. The downstream direction is easy. The end office can send messages to each different house in whatever order it likes. In the upstream direction, however, messages from different houses cannot be sent at the same time, or different signals would collide. The houses also cannot hear each other's transmissions so they cannot listen before transmitting. The solution is that equipment at the houses requests and is granted time slots to use by equipment in the end office. For this to work, there is a ranging process to adjust the transmission times from the houses so that all the signals received at the end office are synchronized. The design is similar to cable modems, which we cover later in this chapter. For more information on the future of PONs, see Grobe and Elbers (2008).

2.6.4 Trunks and Multiplexing

Trunks in the telephone network are not only much faster than the local loops, they are different in two other respects. The core of the telephone network carries digital information, not analog information; that is, bits not voice. This necessitates a conversion at the end office to digital form for transmission over the long-haul trunks. The trunks carry thousands, even millions, of calls simultaneously. This sharing is important for achieving economies of scale, since it costs essentially the same amount of money to install and maintain a high-bandwidth trunk as a low-bandwidth trunk between two switching offices. It is accomplished with versions of TDM and FDM multiplexing.

Below we will briefly examine how voice signals are digitized so that they can be transported by the telephone network. After that, we will see how TDM is used to carry bits on trunks, including the TDM system used for fiber optics

(SONET). Then we will turn to FDM as it is applied to fiber optics, which is called wavelength division multiplexing.

Digitizing Voice Signals

Early in the development of the telephone network, the core handled voice calls as analog information. FDM techniques were used for many years to multiplex 4000-Hz voice channels (comprised of 3100 Hz plus guard bands) into larger and larger units. For example, 12 calls in the 60 kHz–to–108 kHz band is known as a **group** and five groups (a total of 60 calls) are known as a **supergroup**, and so on. These FDM methods are still used over some copper wires and microwave channels. However, FDM requires analog circuitry and is not amenable to being done by a computer. In contrast, TDM can be handled entirely by digital electronics, so it has become far more widespread in recent years. Since TDM can only be used for digital data and the local loops produce analog signals, a conversion is needed from analog to digital in the end office, where all the individual local loops come together to be combined onto outgoing trunks.

The analog signals are digitized in the end office by a device called a **codec** (short for "*co*der-*dec*oder"). The codec makes 8000 samples per second (125 μsec/sample) because the Nyquist theorem says that this is sufficient to capture all the information from the 4-kHz telephone channel bandwidth. At a lower sampling rate, information would be lost; at a higher one, no extra information would be gained. Each sample of the amplitude of the signal is quantized to an 8-bit number.

This technique is called **PCM** (**Pulse Code Modulation**). It forms the heart of the modern telephone system. As a consequence, virtually all time intervals within the telephone system are multiples of 125 μsec. The standard uncompressed data rate for a voice-grade telephone call is thus 8 bits every 125 μsec, or 64 kbps.

At the other end of the call, an analog signal is recreated from the quantized samples by playing them out (and smoothing them) over time. It will not be exactly the same as the original analog signal, even though we sampled at the Nyquist rate, because the samples were quantized. To reduce the error due to quantization, the quantization levels are unevenly spaced. A logarithmic scale is used that gives relatively more bits to smaller signal amplitudes and relatively fewer bits to large signal amplitudes. In this way the error is proportional to the signal amplitude.

Two versions of quantization are widely used: **μ-law**, used in North America and Japan, and **A-law**, used in Europe and the rest of the world. Both versions are specified in standard ITU G.711. An equivalent way to think about this process is to imagine that the dynamic range of the signal (or the ratio between the largest and smallest possible values) is compressed before it is (evenly) quantized, and then expanded when the analog signal is recreated. For this reason it is called

companding. It is also possible to compress the samples after they are digitized so that they require much less than 64 kbps. However, we will leave this topic for when we explore audio applications such as voice over IP.

Time Division Multiplexing

TDM based on PCM is used to carry multiple voice calls over trunks by sending a sample from each call every 125 μsec. When digital transmission began emerging as a feasible technology, ITU (then called CCITT) was unable to reach agreement on an international standard for PCM. Consequently, a variety of incompatible schemes are now in use in different countries around the world.

The method used in North America and Japan is the **T1** carrier, depicted in Fig. 2-37. (Technically speaking, the format is called DS1 and the carrier is called T1, but following widespread industry tradition, we will not make that subtle distinction here.) The T1 carrier consists of 24 voice channels multiplexed together. Each of the 24 channels, in turn, gets to insert 8 bits into the output stream.

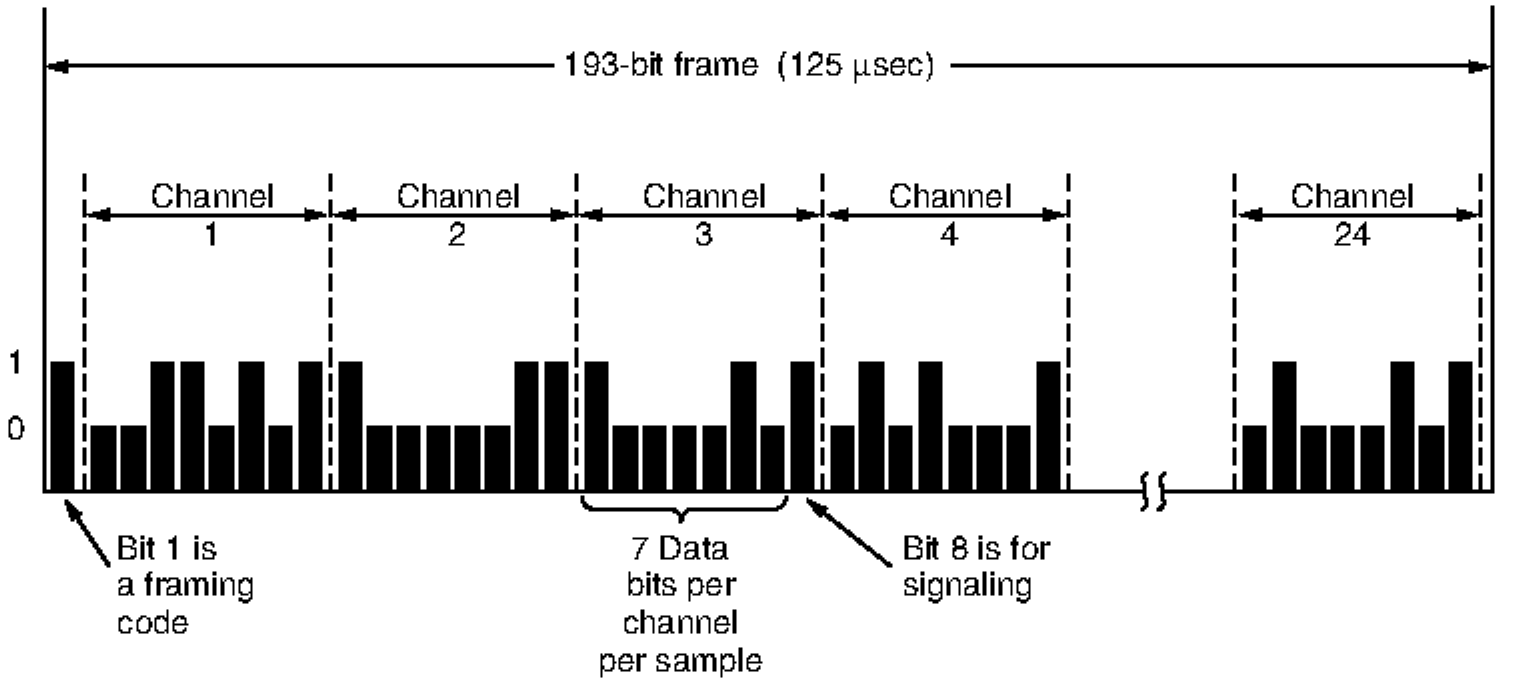


Figure 2-37. The T1 carrier (1.544 Mbps).

A frame consists of 24 × 8 = 192 bits plus one extra bit for control purposes, yielding 193 bits every 125 μsec. This gives a gross data rate of 1.544 Mbps, of which 8 kbps is for signaling. The 193rd bit is used for frame synchronization and signaling. In one variation, the 193rd bit is used across a group of 24 frames called an **extended superframe**. Six of the bits, in the 4th, 8th, 12th, 16th, 20th, and 24th positions, take on the alternating pattern 001011 . . . . Normally, the receiver keeps checking for this pattern to make sure that it has not lost synchronization. Six more bits are used to send an error check code to help the receiver confirm that it is synchronized. If it does get out of sync, the receiver can scan for the pattern and validate the error check code to get resynchronized. The remaining 12 bits are used for control information for operating and maintaining the network, such as performance reporting from the remote end.

The T1 format has several variations. The earlier versions sent signaling information **in-band**, meaning in the same channel as the data, by using some of the data bits. This design is one form of **channel-associated signaling**, because each channel has its own private signaling subchannel. In one arrangement, the least significant bit out of an 8-bit sample on each channel is used in every sixth frame. It has the colorful name of **robbed-bit signaling**. The idea is that a few stolen bits will not matter for voice calls. No one will hear the difference.

For data, however, it is another story. Delivering the wrong bits is unhelpful, to say the least. If older versions of T1 are used to carry data, only 7 of 8 bits, or 56 kbps can be used in each of the 24 channels. Instead, newer versions of T1 provide clear channels in which all of the bits may be used to send data. Clear channels are what businesses who lease a T1 line want when they send data across the telephone network in place of voice samples. Signaling for any voice calls is then handled **out-of-band**, meaning in a separate channel from the data. Often, the signaling is done with **common-channel signaling** in which there is a shared signaling channel. One of the 24 channels may be used for this purpose.

Outside North America and Japan, the 2.048-Mbps **E1** carrier is used instead of T1. This carrier has 32 8-bit data samples packed into the basic 125-μsec frame. Thirty of the channels are used for information and up to two are used for signaling. Each group of four frames provides 64 signaling bits, half of which are used for signaling (whether channel-associated or common-channel) and half of which are used for frame synchronization or are reserved for each country to use as it wishes.

Time division multiplexing allows multiple T1 carriers to be multiplexed into higher-order carriers. Figure 2-38 shows how this can be done. At the left we see four T1 channels being multiplexed into one T2 channel. The multiplexing at T2 and above is done bit for bit, rather than byte for byte with the 24 voice channels that make up a T1 frame. Four T1 streams at 1.544 Mbps should generate 6.176 Mbps, but T2 is actually 6.312 Mbps. The extra bits are used for framing and recovery in case the carrier slips. T1 and T3 are widely used by customers, whereas T2 and T4 are only used within the telephone system itself, so they are not well known.

At the next level, seven T2 streams are combined bitwise to form a T3 stream. Then six T3 streams are joined to form a T4 stream. At each step a small amount of overhead is added for framing and recovery in case the synchronization between sender and receiver is lost.

Just as there is little agreement on the basic carrier between the United States and the rest of the world, there is equally little agreement on how it is to be multiplexed into higher-bandwidth carriers. The U.S. scheme of stepping up by 4, 7, and 6 did not strike everyone else as the way to go, so the ITU standard calls for multiplexing four streams into one stream at each level. Also, the framing and

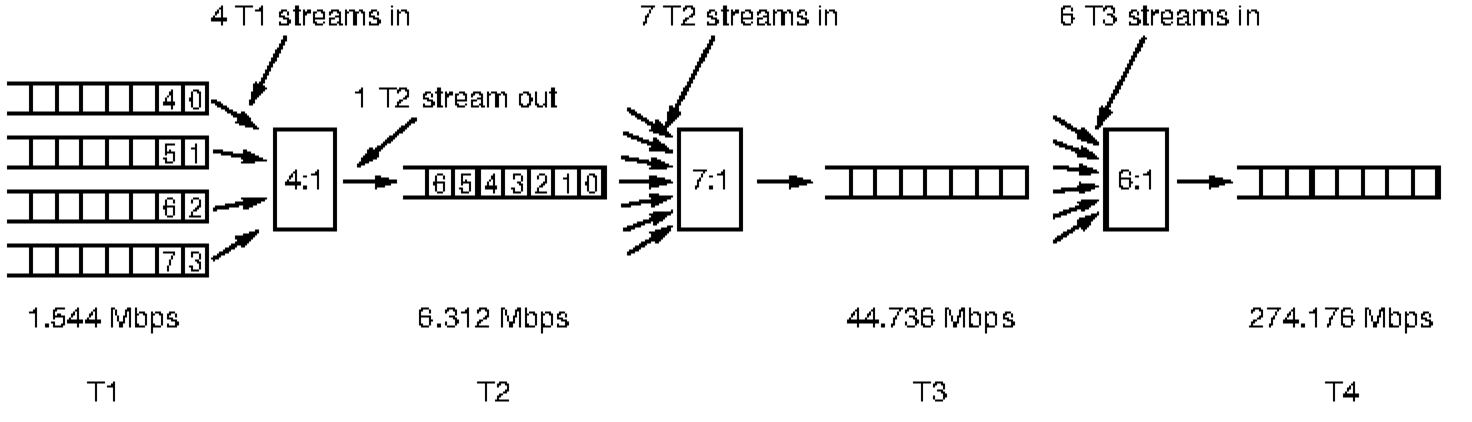


Figure 2-38. Multiplexing T1 streams into higher carriers.

recovery data are different in the U.S. and ITU standards. The ITU hierarchy for 32, 128, 512, 2048, and 8192 channels runs at speeds of 2.048, 8.848, 34.304, 139.264, and 565.148 Mbps.

SONET/SDH

In the early days of fiber optics, every telephone company had its own proprietary optical TDM system. After AT&T was broken up in 1984, local telephone companies had to connect to multiple long-distance carriers, all with different optical TDM systems, so the need for standardization became obvious. In 1985, Bellcore, the RBOC's research arm, began working on a standard, called **SONET** (**Synchronous Optical NETwork**).

Later, ITU joined the effort, which resulted in a SONET standard and a set of parallel ITU recommendations (G.707, G.708, and G.709) in 1989. The ITU recommendations are called **SDH** (**Synchronous Digital Hierarchy**) but differ from SONET only in minor ways. Virtually all the long-distance telephone traffic in the United States, and much of it elsewhere, now uses trunks running SONET in the physical layer. For additional information about SONET, see Bellamy (2000), Goralski (2002), and Shepard (2001).

The SONET design had four major goals. First and foremost, SONET had to make it possible for different carriers to interwork. Achieving this goal required defining a common signaling standard with respect to wavelength, timing, framing structure, and other issues.

Second, some means was needed to unify the U.S., European, and Japanese digital systems, all of which were based on 64-kbps PCM channels but combined them in different (and incompatible) ways.

Third, SONET had to provide a way to multiplex multiple digital channels. At the time SONET was devised, the highest-speed digital carrier actually used widely in the United States was T3, at 44.736 Mbps. T4 was defined, but not used much, and nothing was even defined above T4 speed. Part of SONET's mission was to continue the hierarchy to gigabits/sec and beyond. A standard way to multiplex slower channels into one SONET channel was also needed.

Fourth, SONET had to provide support for operations, administration, and maintenance (OAM), which are needed to manage the network. Previous systems did not do this very well.

An early decision was to make SONET a traditional TDM system, with the entire bandwidth of the fiber devoted to one channel containing time slots for the various subchannels. As such, SONET is a synchronous system. Each sender and receiver is tied to a common clock. The master clock that controls the system has an accuracy of about 1 part in 10^9. Bits on a SONET line are sent out at extremely precise intervals, controlled by the master clock.

The basic SONET frame is a block of 810 bytes put out every 125 μsec. Since SONET is synchronous, frames are emitted whether or not there are any useful data to send. Having 8000 frames/sec exactly matches the sampling rate of the PCM channels used in all digital telephony systems.

The 810-byte SONET frames are best described as a rectangle of bytes, 90 columns wide by 9 rows high. Thus, $8 \times 810 = 6480$ bits are transmitted 8000 times per second, for a gross data rate of 51.84 Mbps. This layout is the basic SONET channel, called **STS-1** (**Synchronous Transport Signal-1**). All SONET trunks are multiples of STS-1.

The first three columns of each frame are reserved for system management information, as illustrated in Fig. 2-39. In this block, the first three rows contain the section overhead; the next six contain the line overhead. The section overhead is generated and checked at the start and end of each section, whereas the line overhead is generated and checked at the start and end of each line.

A SONET transmitter sends back-to-back 810-byte frames, without gaps between them, even when there are no data (in which case it sends dummy data). From the receiver's point of view, all it sees is a continuous bit stream, so how does it know where each frame begins? The answer is that the first 2 bytes of each frame contain a fixed pattern that the receiver searches for. If it finds this pattern in the same place in a large number of consecutive frames, it assumes that it is in sync with the sender. In theory, a user could insert this pattern into the payload in a regular way, but in practice it cannot be done due to the multiplexing of multiple users into the same frame and other reasons.

The remaining 87 columns of each frame hold $87 \times 9 \times 8 \times 8000 = 50.112$ Mbps of user data. This user data could be voice samples, T1 and other carriers swallowed whole, or packets. SONET is simply a convenient container for transporting bits. The **SPE** (**Synchronous Payload Envelope**), which carries the user data does not always begin in row 1, column 4. The SPE can begin anywhere within the frame. A pointer to the first byte is contained in the first row of the line overhead. The first column of the SPE is the path overhead (i.e., the header for the end-to-end path sublayer protocol).

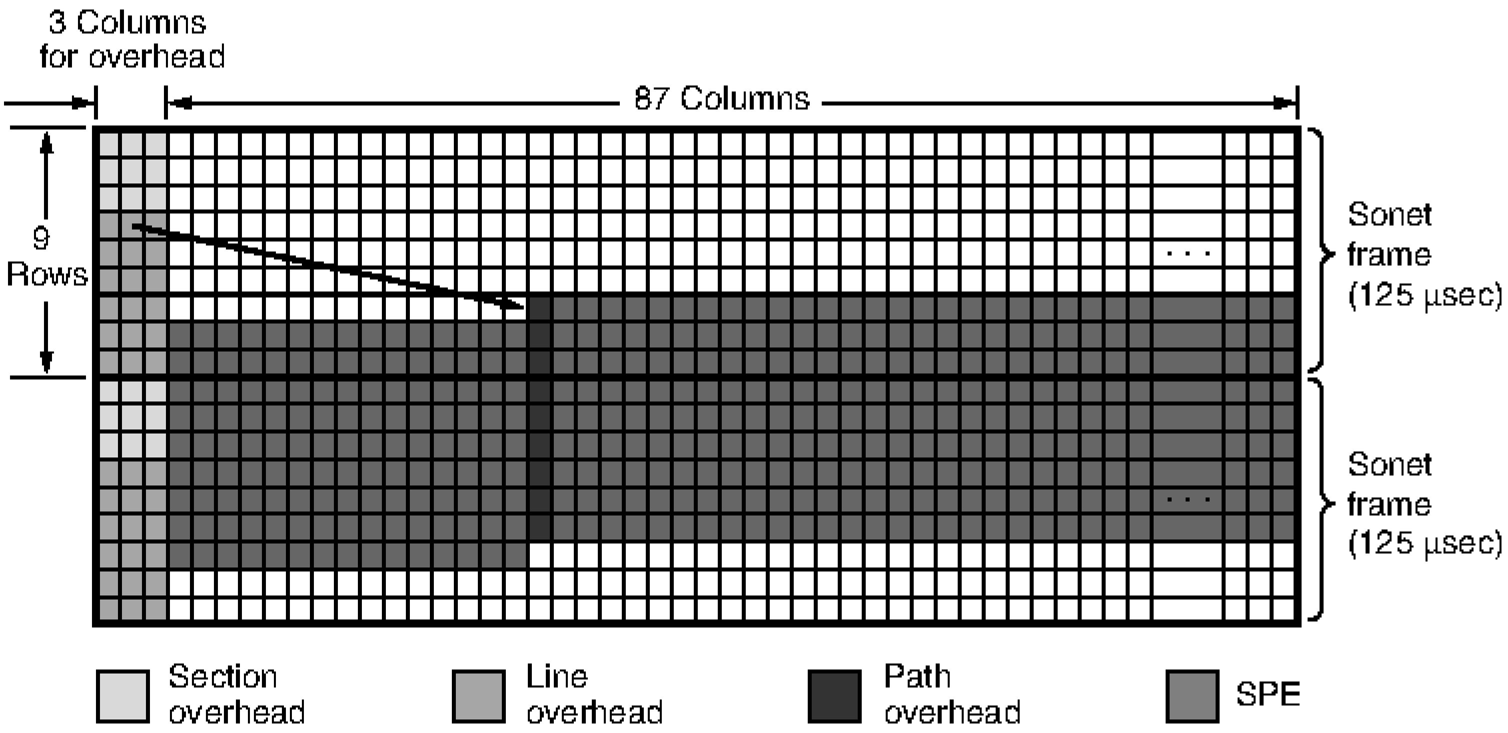


Figure 2-39. Two back-to-back SONET frames.

The ability to allow the SPE to begin anywhere within the SONET frame and even to span two frames, as shown in Fig. 2-39, gives added flexibility to the system. For example, if a payload arrives at the source while a dummy SONET frame is being constructed, it can be inserted into the current frame instead of being held until the start of the next one.

The SONET/SDH multiplexing hierarchy is shown in Fig. 2-40. Rates from STS-1 to STS-768 have been defined, ranging from roughly a T3 line to 40 Gbps. Even higher rates will surely be defined over time, with OC-3072 at 160 Gbps being the next in line if and when it becomes technologically feasible. The optical carrier corresponding to STS-*n* is called OC-*n* but is bit for bit the same except for a certain bit reordering needed for synchronization. The SDH names are different, and they start at OC-3 because ITU-based systems do not have a rate near 51.84 Mbps. We have shown the common rates, which proceed from OC-3 in multiples of four. The gross data rate includes all the overhead. The SPE data rate excludes the line and section overhead. The user data rate excludes all overhead and counts only the 87 payload columns.

As an aside, when a carrier, such as OC-3, is not multiplexed, but carries the data from only a single source, the letter *c* (for concatenated) is appended to the designation, so OC-3 indicates a 155.52-Mbps carrier consisting of three separate OC-1 carriers, but OC-3c indicates a data stream from a single source at 155.52 Mbps. The three OC-1 streams within an OC-3c stream are interleaved by column—first column 1 from stream 1, then column 1 from stream 2, then column 1 from stream 3, followed by column 2 from stream 1, and so on—leading to a frame 270 columns wide and 9 rows deep.

SONET		SDH	Data rate (Mbps)		
Electrical	Optical	Optical	Gross	SPE	User
STS-1	OC-1		51.84	50.112	49.536
STS-3	OC-3	STM-1	155.52	150.336	148.608
STS-12	OC-12	STM-4	622.08	601.344	594.432
STS-48	OC-48	STM-16	2488.32	2405.376	2377.728
STS-192	OC-192	STM-64	9953.28	9621.504	9510.912
STS-768	OC-768	STM-256	39813.12	38486.016	38043.648

Figure 2-40. SONET and SDH multiplex rates.

Wavelength Division Multiplexing

A form of frequency division multiplexing is used as well as TDM to harness the tremendous bandwidth of fiber optic channels. It is called **WDM** (**Wavelength Division Multiplexing**). The basic principle of WDM on fibers is depicted in Fig. 2-41. Here four fibers come together at an optical combiner, each with its energy present at a different wavelength. The four beams are combined onto a single shared fiber for transmission to a distant destination. At the far end, the beam is split up over as many fibers as there were on the input side. Each output fiber contains a short, specially constructed core that filters out all but one wavelength. The resulting signals can be routed to their destination or recombined in different ways for additional multiplexed transport.

There is really nothing new here. This way of operating is just frequency division multiplexing at very high frequencies, with the term WDM owing to the description of fiber optic channels by their wavelength or "color" rather than frequency. As long as each channel has its own frequency (i.e., wavelength) range and all the ranges are disjoint, they can be multiplexed together on the long-haul fiber. The only difference with electrical FDM is that an optical system using a diffraction grating is completely passive and thus highly reliable.

The reason WDM is popular is that the energy on a single channel is typically only a few gigahertz wide because that is the current limit of how fast we can convert between electrical and optical signals. By running many channels in parallel on different wavelengths, the aggregate bandwidth is increased linearly with the number of channels. Since the bandwidth of a single fiber band is about 25,000 GHz (see Fig. 2-7), there is theoretically room for 2500 10-Gbps channels even at 1 bit/Hz (and higher rates are also possible).

WDM technology has been progressing at a rate that puts computer technology to shame. WDM was invented around 1990. The first commercial systems had eight channels of 2.5 Gbps per channel. By 1998, systems with 40 channels

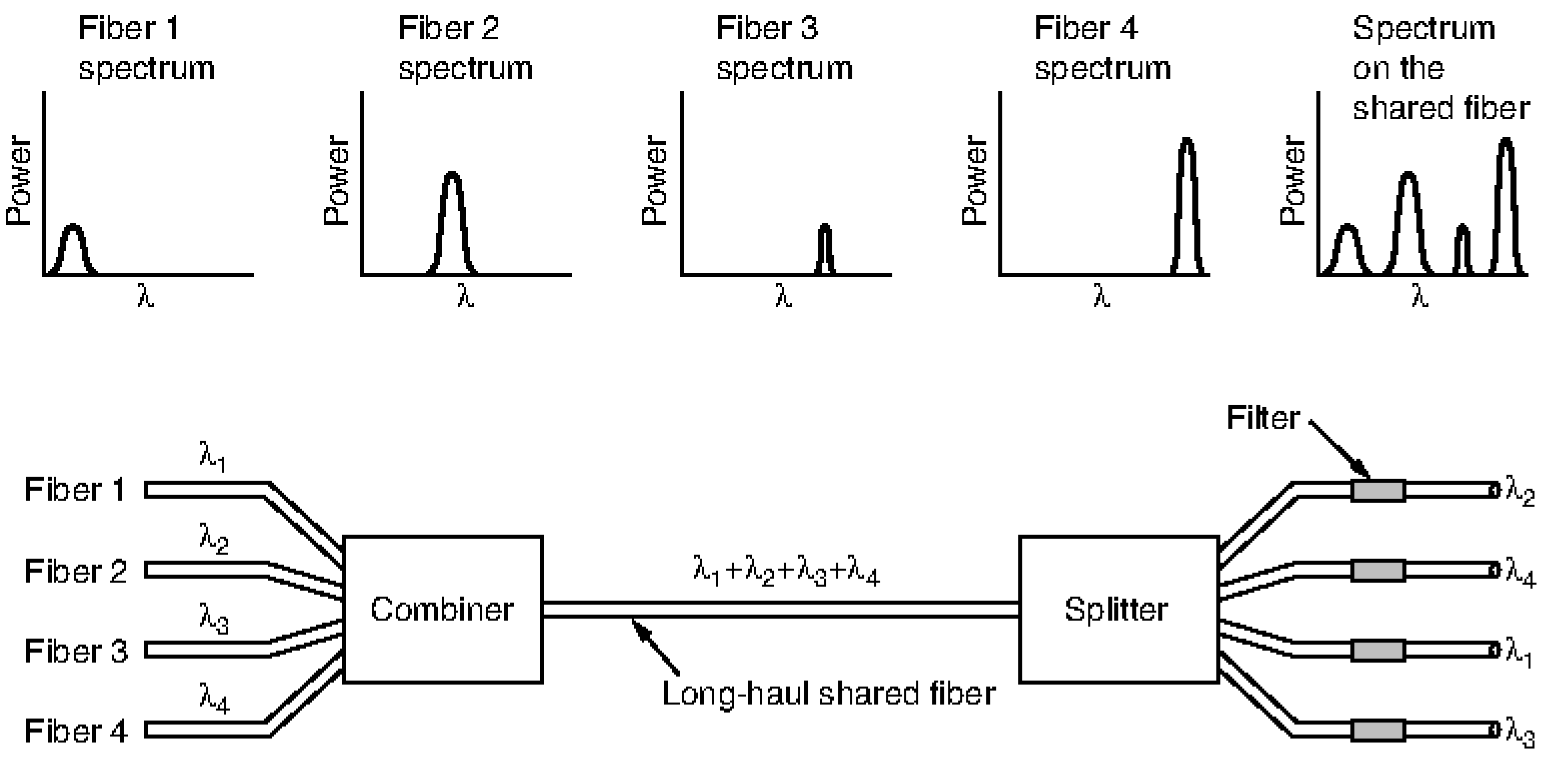


Figure 2-41. Wavelength division multiplexing.

of 2.5 Gbps were on the market. By 2006, there were products with 192 channels of 10 Gbps and 64 channels of 40 Gbps, capable of moving up to 2.56 Tbps. This bandwidth is enough to transmit 80 full-length DVD movies per second. The channels are also packed tightly on the fiber, with 200, 100, or as little as 50 GHz of separation. Technology demonstrations by companies after bragging rights have shown 10 times this capacity in the lab, but going from the lab to the field usually takes at least a few years. When the number of channels is very large and the wavelengths are spaced close together, the system is referred to as **DWDM (Dense WDM)**.

One of the drivers of WDM technology is the development of all-optical components. Previously, every 100 km it was necessary to split up all the channels and convert each one to an electrical signal for amplification separately before reconverting them to optical signals and combining them. Nowadays, all-optical amplifiers can regenerate the entire signal once every 1000 km without the need for multiple opto-electrical conversions.

In the example of Fig. 2-41, we have a fixed-wavelength system. Bits from input fiber 1 go to output fiber 3, bits from input fiber 2 go to output fiber 1, etc. However, it is also possible to build WDM systems that are switched in the optical domain. In such a device, the output filters are tunable using Fabry-Perot or Mach-Zehnder interferometers. These devices allow the selected frequencies to be changed dynamically by a control computer. This ability provides a large amount of flexibility to provision many different wavelength paths through the telephone network from a fixed set of fibers. For more information about optical networks and WDM, see Ramaswami et al. (2009).

2.6.5 Switching

From the point of view of the average telephone engineer, the phone system is divided into two principal parts: outside plant (the local loops and trunks, since they are physically outside the switching offices) and inside plant (the switches, which are inside the switching offices). We have just looked at the outside plant. Now it is time to examine the inside plant.

Two different switching techniques are used by the network nowadays: circuit switching and packet switching. The traditional telephone system is based on circuit switching, but packet switching is beginning to make inroads with the rise of voice over IP technology. We will go into circuit switching in some detail and contrast it with packet switching. Both kinds of switching are important enough that we will come back to them when we get to the network layer.

Circuit Switching

Conceptually, when you or your computer places a telephone call, the switching equipment within the telephone system seeks out a physical path all the way from your telephone to the receiver's telephone. This technique is called **circuit switching**. It is shown schematically in Fig. 2-42(a). Each of the six rectangles represents a carrier switching office (end office, toll office, etc.). In this example, each office has three incoming lines and three outgoing lines. When a call passes through a switching office, a physical connection is (conceptually) established between the line on which the call came in and one of the output lines, as shown by the dotted lines.

In the early days of the telephone, the connection was made by the operator plugging a jumper cable into the input and output sockets. In fact, a surprising little story is associated with the invention of automatic circuit switching equipment. It was invented by a 19th-century Missouri undertaker named Almon B. Strowger. Shortly after the telephone was invented, when someone died, one of the survivors would call the town operator and say "Please connect me to an undertaker." Unfortunately for Mr. Strowger, there were two undertakers in his town, and the other one's wife was the town telephone operator. He quickly saw that either he was going to have to invent automatic telephone switching equipment or he was going to go out of business. He chose the first option. For nearly 100 years, the circuit-switching equipment used worldwide was known as **Strowger gear**. (History does not record whether the now-unemployed switchboard operator got a job as an information operator, answering questions such as "What is the phone number of an undertaker?")

The model shown in Fig. 2-42(a) is highly simplified, of course, because parts of the physical path between the two telephones may, in fact, be microwave or fiber links onto which thousands of calls are multiplexed. Nevertheless, the basic idea is valid: once a call has been set up, a dedicated path between both ends exists and will continue to exist until the call is finished.

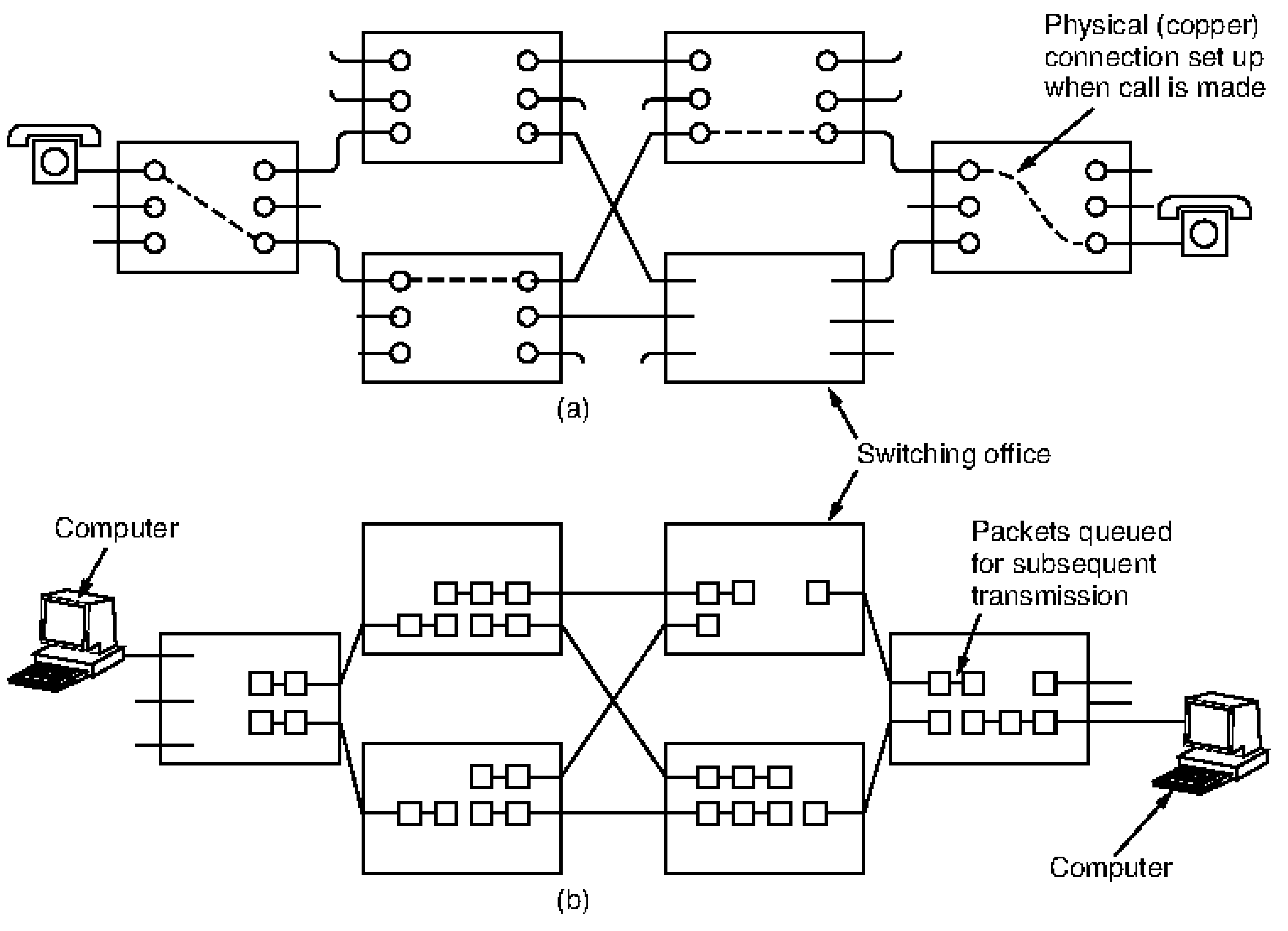


Figure 2-42. (a) Circuit switching. (b) Packet switching.

An important property of circuit switching is the need to set up an end-to-end path *before* any data can be sent. The elapsed time between the end of dialing and the start of ringing can easily be 10 sec, more on long-distance or international calls. During this time interval, the telephone system is hunting for a path, as shown in Fig. 2-43(a). Note that before data transmission can even begin, the call request signal must propagate all the way to the destination and be acknowledged. For many computer applications (e.g., point-of-sale credit verification), long setup times are undesirable.

As a consequence of the reserved path between the calling parties, once the setup has been completed, the only delay for data is the propagation time for the electromagnetic signal, about 5 msec per 1000 km. Also as a consequence of the established path, there is no danger of congestion—that is, once the call has been put through, you never get busy signals. Of course, you might get one before the connection has been established due to lack of switching or trunk capacity.

Packet Switching

The alternative to circuit switching is **packet switching**, shown in Fig. 2-42(b) and described in Chap. 1. With this technology, packets are sent as soon as they are available. There is no need to set up a dedicated path in advance, unlike

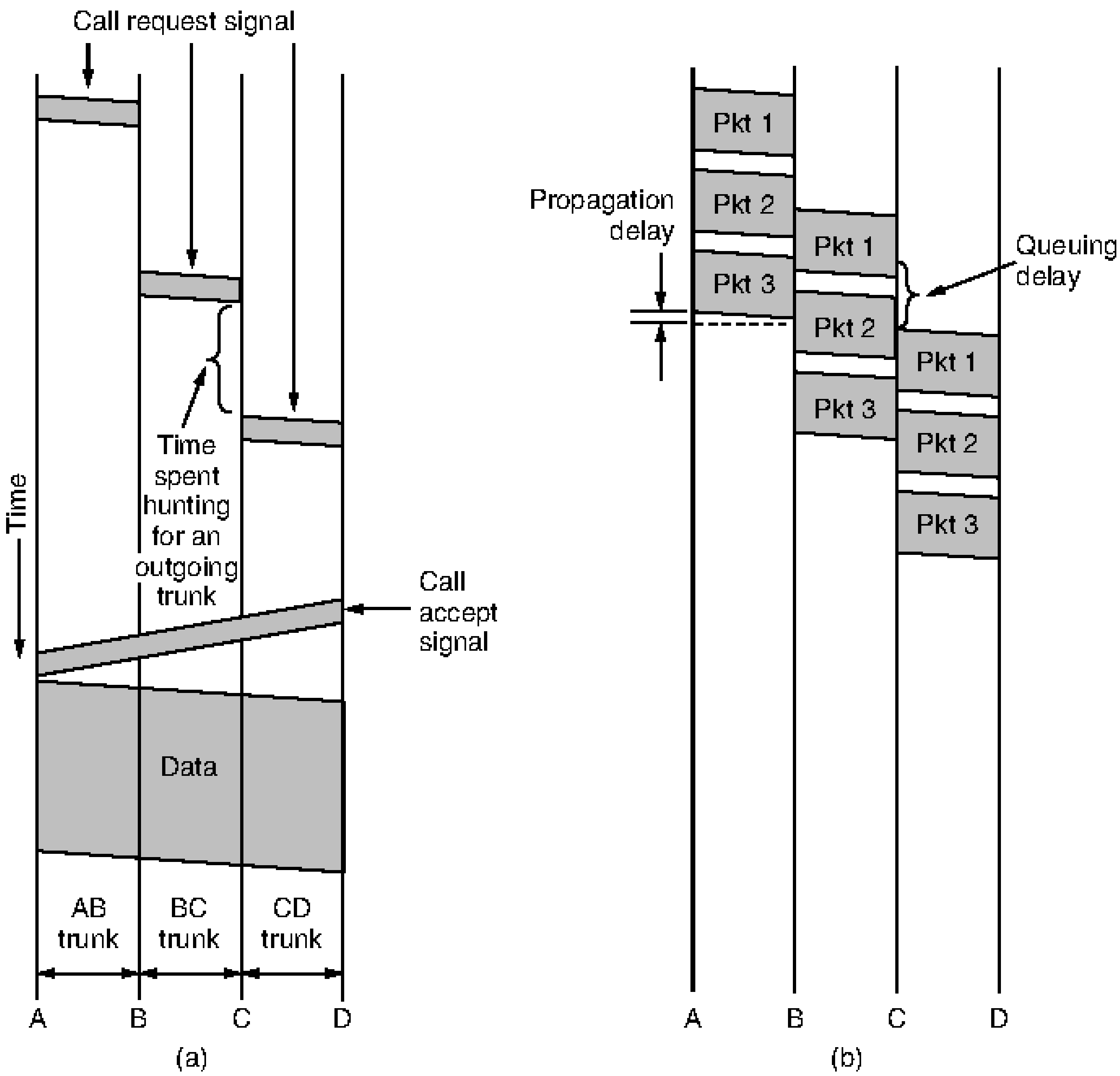


Figure 2-43. Timing of events in (a) circuit switching, (b) packet switching.

with circuit switching. It is up to routers to use store-and-forward transmission to send each packet on its way to the destination on its own. This procedure is unlike circuit switching, in which the result of the connection setup is the reservation of bandwidth all the way from the sender to the receiver. All data on the circuit follows this path. Among other properties, having all the data follow the same path means that it cannot arrive out of order. With packet switching there is no fixed path, so different packets can follow different paths, depending on network conditions at the time they are sent, and they may arrive out of order.

Packet-switching networks place a tight upper limit on the size of packets. This ensures that no user can monopolize any transmission line for very long (e.g., many milliseconds), so that packet-switched networks can handle interactive traffic. It also reduces delay since the first packet of a long message can be forwarded before the second one has fully arrived. However, the store-and-forward delay of accumulating a packet in the router's memory before it is sent on to the next router exceeds that of circuit switching. With circuit switching, the bits just flow through the wire continuously.

Packet and circuit switching also differ in other ways. Because no bandwidth is reserved with packet switching, packets may have to wait to be forwarded. This introduces **queuing delay** and congestion if many packets are sent at the same time. On the other hand, there is no danger of getting a busy signal and being unable to use the network. Thus, congestion occurs at different times with circuit switching (at setup time) and packet switching (when packets are sent).

If a circuit has been reserved for a particular user and there is no traffic, its bandwidth is wasted. It cannot be used for other traffic. Packet switching does not waste bandwidth and thus is more efficient from a system perspective. Understanding this trade-off is crucial for comprehending the difference between circuit switching and packet switching. The trade-off is between guaranteed service and wasting resources versus not guaranteeing service and not wasting resources.

Packet switching is more fault tolerant than circuit switching. In fact, that is why it was invented. If a switch goes down, all of the circuits using it are terminated and no more traffic can be sent on any of them. With packet switching, packets can be routed around dead switches.

A final difference between circuit and packet switching is the charging algorithm. With circuit switching, charging has historically been based on distance and time. For mobile phones, distance usually does not play a role, except for international calls, and time plays only a coarse role (e.g., a calling plan with 2000 free minutes costs more than one with 1000 free minutes and sometimes nights or weekends are cheap). With packet switching, connect time is not an issue, but the volume of traffic is. For home users, ISPs usually charge a flat monthly rate because it is less work for them and their customers can understand this model, but backbone carriers charge regional networks based on the volume of their traffic.

The differences are summarized in Fig. 2-44. Traditionally, telephone networks have used circuit switching to provide high-quality telephone calls, and computer networks have used packet switching for simplicity and efficiency. However, there are notable exceptions. Some older computer networks have been circuit switched under the covers (e.g., X.25) and some newer telephone networks use packet switching with voice over IP technology. This looks just like a standard telephone call on the outside to users, but inside the network packets of voice data are switched. This approach has let upstarts market cheap international calls via calling cards, though perhaps with lower call quality than the incumbents.

2.7 THE MOBILE TELEPHONE SYSTEM

The traditional telephone system, even if it someday gets multigigabit end-to-end fiber, will still not be able to satisfy a growing group of users: people on the go. People now expect to make phone calls and to use their phones to check

Item	Circuit switched	Packet switched
Call setup	Required	Not needed
Dedicated physical path	Yes	No
Each packet follows the same route	Yes	No
Packets arrive in order	Yes	No
Is a switch crash fatal	Yes	No
Bandwidth available	Fixed	Dynamic
Time of possible congestion	At setup time	On every packet
Potentially wasted bandwidth	Yes	No
Store-and-forward transmission	No	Yes
Charging	Per minute	Per packet

Figure 2-44. A comparison of circuit-switched and packet-switched networks.

email and surf the Web from airplanes, cars, swimming pools, and while jogging in the park. Consequently, there is a tremendous amount of interest in wireless telephony. In the following sections we will study this topic in some detail.

The mobile phone system is used for wide area voice and data communication. **Mobile phones** (sometimes called **cell phones**) have gone through three distinct generations, widely called **1G**, **2G**, and **3G**. The generations are:

1. Analog voice.
2. Digital voice.
3. Digital voice and data (Internet, email, etc.).

(Mobile phones should not be confused with **cordless phones** that consist of a base station and a handset sold as a set for use within the home. These are never used for networking, so we will not examine them further.)

Although most of our discussion will be about the technology of these systems, it is interesting to note how political and tiny marketing decisions can have a huge impact. The first mobile system was devised in the U.S. by AT&T and mandated for the whole country by the FCC. As a result, the entire U.S. had a single (analog) system and a mobile phone purchased in California also worked in New York. In contrast, when mobile phones came to Europe, every country devised its own system, which resulted in a fiasco.

Europe learned from its mistake and when digital came around, the government-run PTTs got together and standardized on a single system (GSM), so any European mobile phone will work anywhere in Europe. By then, the U.S. had decided that government should not be in the standardization business, so it left digital to the marketplace. This decision resulted in different equipment manufacturers producing different kinds of mobile phones. As a consequence, in the U.S.

two major—and completely incompatible—digital mobile phone systems were deployed, as well as other minor systems.

Despite an initial lead by the U.S., mobile phone ownership and usage in Europe is now far greater than in the U.S. Having a single system that works anywhere in Europe and with any provider is part of the reason, but there is more. A second area where the U.S. and Europe differed is in the humble matter of phone numbers. In the U.S., mobile phones are mixed in with regular (fixed) telephones. Thus, there is no way for a caller to see if, say, (212) 234-5678 is a fixed telephone (cheap or free call) or a mobile phone (expensive call). To keep people from getting nervous about placing calls, the telephone companies decided to make the mobile phone owner pay for incoming calls. As a consequence, many people hesitated buying a mobile phone for fear of running up a big bill by just receiving calls. In Europe, mobile phone numbers have a special area code (analogous to 800 and 900 numbers) so they are instantly recognizable. Consequently, the usual rule of "caller pays" also applies to mobile phones in Europe (except for international calls, where costs are split).

A third issue that has had a large impact on adoption is the widespread use of prepaid mobile phones in Europe (up to 75% in some areas). These can be purchased in many stores with no more formality than buying a digital camera. You pay and you go. They are preloaded with a balance of, for example, 20 or 50 euros and can be recharged (using a secret PIN code) when the balance drops to zero. As a consequence, practically every teenager and many small children in Europe have (usually prepaid) mobile phones so their parents can locate them, without the danger of the child running up a huge bill. If the mobile phone is used only occasionally, its use is essentially free since there is no monthly charge or charge for incoming calls.

2.7.1 First-Generation (1G) Mobile Phones: Analog Voice

Enough about the politics and marketing aspects of mobile phones. Now let us look at the technology, starting with the earliest system. Mobile radiotelephones were used sporadically for maritime and military communication during the early decades of the 20th century. In 1946, the first system for car-based telephones was set up in St. Louis. This system used a single large transmitter on top of a tall building and had a single channel, used for both sending and receiving. To talk, the user had to push a button that enabled the transmitter and disabled the receiver. Such systems, known as **push-to-talk systems**, were installed in several cities beginning in the late 1950s. CB radio, taxis, and police cars often use this technology.

In the 1960s, **IMTS** (**Improved Mobile Telephone System**) was installed. It, too, used a high-powered (200-watt) transmitter on top of a hill but it had two frequencies, one for sending and one for receiving, so the push-to-talk button was no longer needed. Since all communication from the mobile telephones went inbound on a different channel than the outbound signals, the mobile users could not hear each other (unlike the push-to-talk system used in taxis).

IMTS supported 23 channels spread out from 150 MHz to 450 MHz. Due to the small number of channels, users often had to wait a long time before getting a dial tone. Also, due to the large power of the hilltop transmitters, adjacent systems had to be several hundred kilometers apart to avoid interference. All in all, the limited capacity made the system impractical.

Advanced Mobile Phone System

All that changed with **AMPS** (**Advanced Mobile Phone System**), invented by Bell Labs and first installed in the United States in 1982. It was also used in England, where it was called TACS, and in Japan, where it was called MCS-L1. AMPS was formally retired in 2008, but we will look at it to understand the context for the 2G and 3G systems that improved on it.

In all mobile phone systems, a geographic region is divided up into **cells**, which is why the devices are sometimes called cell phones. In AMPS, the cells are typically 10 to 20 km across; in digital systems, the cells are smaller. Each cell uses some set of frequencies not used by any of its neighbors. The key idea that gives cellular systems far more capacity than previous systems is the use of relatively small cells and the reuse of transmission frequencies in nearby (but not adjacent) cells. Whereas an IMTS system 100 km across can have only one call on each frequency, an AMPS system might have 100 10-km cells in the same area and be able to have 10 to 15 calls on each frequency, in widely separated cells. Thus, the cellular design increases the system capacity by at least an order of magnitude, more as the cells get smaller. Furthermore, smaller cells mean that less power is needed, which leads to smaller and cheaper transmitters and handsets.

The idea of frequency reuse is illustrated in Fig. 2-45(a). The cells are normally roughly circular, but they are easier to model as hexagons. In Fig. 2-45(a), the cells are all the same size. They are grouped in units of seven cells. Each letter indicates a group of frequencies. Notice that for each frequency set, there is a buffer about two cells wide where that frequency is not reused, providing for good separation and low interference.

Finding locations high in the air to place base station antennas is a major issue. This problem has led some telecommunication carriers to forge alliances with the Roman Catholic Church, since the latter owns a substantial number of exalted potential antenna sites worldwide, all conveniently under a single management.

In an area where the number of users has grown to the point that the system is overloaded, the power can be reduced and the overloaded cells split into smaller

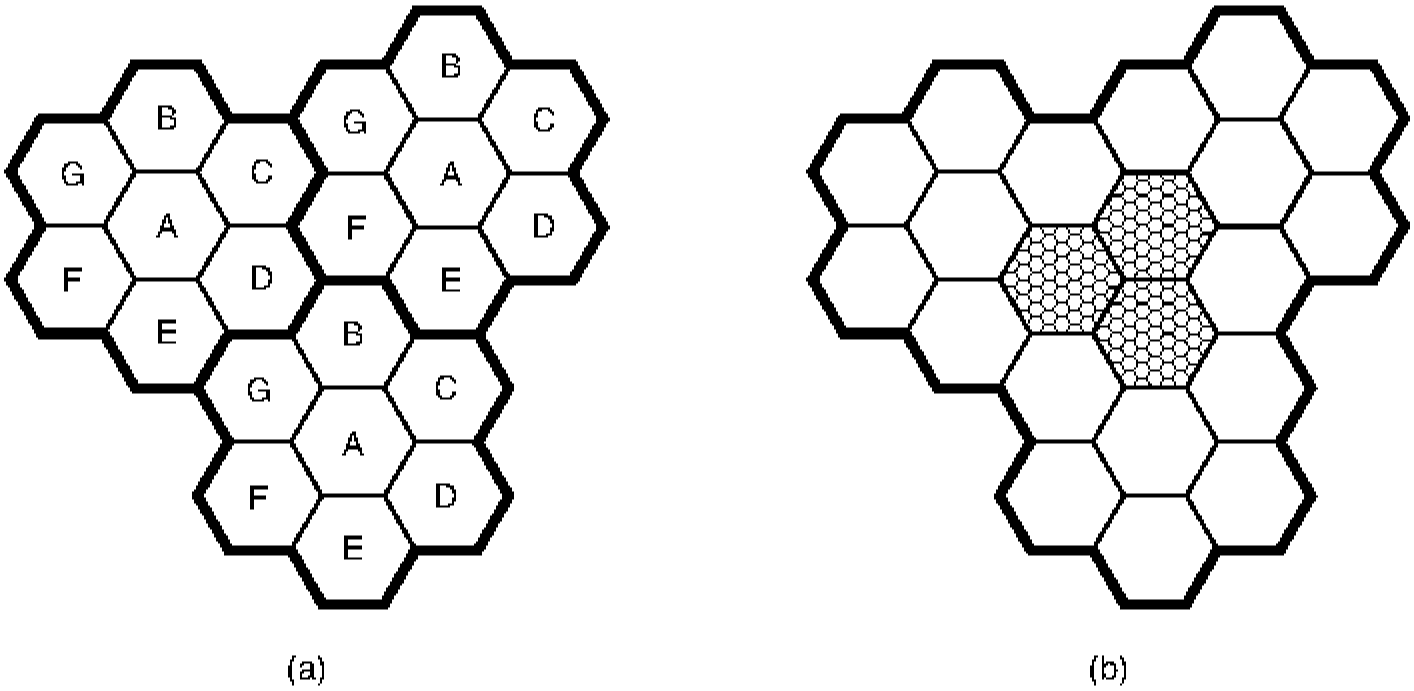


Figure 2-45. (a) Frequencies are not reused in adjacent cells. (b) To add more users, smaller cells can be used.

microcells to permit more frequency reuse, as shown in Fig. 2-45(b). Telephone companies sometimes create temporary microcells, using portable towers with satellite links at sporting events, rock concerts, and other places where large numbers of mobile users congregate for a few hours.

At the center of each cell is a base station to which all the telephones in the cell transmit. The base station consists of a computer and transmitter/receiver connected to an antenna. In a small system, all the base stations are connected to a single device called an **MSC** (**Mobile Switching Center**) or **MTSO** (**Mobile Telephone Switching Office**). In a larger one, several MSCs may be needed, all of which are connected to a second-level MSC, and so on. The MSCs are essentially end offices as in the telephone system, and are in fact connected to at least one telephone system end office. The MSCs communicate with the base stations, each other, and the PSTN using a packet-switching network.

At any instant, each mobile telephone is logically in one specific cell and under the control of that cell's base station. When a mobile telephone physically leaves a cell, its base station notices the telephone's signal fading away and asks all the surrounding base stations how much power they are getting from it. When the answers come back, the base station then transfers ownership to the cell getting the strongest signal; under most conditions that is the cell where the telephone is now located. The telephone is then informed of its new boss, and if a call is in progress, it is asked to switch to a new channel (because the old one is not reused in any of the adjacent cells). This process, called **handoff**, takes about 300 msec. Channel assignment is done by the MSC, the nerve center of the system. The base stations are really just dumb radio relays.

Channels

AMPS uses FDM to separate the channels. The system uses 832 full-duplex channels, each consisting of a pair of simplex channels. This arrangement is known as **FDD** (**Frequency Division Duplex**). The 832 simplex channels from 824 to 849 MHz are used for mobile to base station transmission, and 832 simplex channels from 869 to 894 MHz are used for base station to mobile transmission. Each of these simplex channels is 30 kHz wide.

The 832 channels are divided into four categories. Control channels (base to mobile) are used to manage the system. Paging channels (base to mobile) alert mobile users to calls for them. Access channels (bidirectional) are used for call setup and channel assignment. Finally, data channels (bidirectional) carry voice, fax, or data. Since the same frequencies cannot be reused in nearby cells and 21 channels are reserved in each cell for control, the actual number of voice channels available per cell is much smaller than 832, typically about 45.

Call Management

Each mobile telephone in AMPS has a 32-bit serial number and a 10-digit telephone number in its programmable read-only memory. The telephone number is represented as a 3-digit area code in 10 bits and a 7-digit subscriber number in 24 bits. When a phone is switched on, it scans a preprogrammed list of 21 control channels to find the most powerful signal. The phone then broadcasts its 32-bit serial number and 34-bit telephone number. Like all the control information in AMPS, this packet is sent in digital form, multiple times, and with an error-correcting code, even though the voice channels themselves are analog.

When the base station hears the announcement, it tells the MSC, which records the existence of its new customer and also informs the customer's home MSC of his current location. During normal operation, the mobile telephone reregisters about once every 15 minutes.

To make a call, a mobile user switches on the phone, enters the number to be called on the keypad, and hits the SEND button. The phone then transmits the number to be called and its own identity on the access channel. If a collision occurs there, it tries again later. When the base station gets the request, it informs the MSC. If the caller is a customer of the MSC's company (or one of its partners), the MSC looks for an idle channel for the call. If one is found, the channel number is sent back on the control channel. The mobile phone then automatically switches to the selected voice channel and waits until the called party picks up the phone.

Incoming calls work differently. To start with, all idle phones continuously listen to the paging channel to detect messages directed at them. When a call is placed to a mobile phone (either from a fixed phone or another mobile phone), a packet is sent to the callee's home MSC to find out where it is. A packet is then sent to the base station in its current cell, which sends a broadcast on the paging channel of the form "Unit 14, are you there?" The called phone responds with a "Yes" on the access channel. The base then says something like: "Unit 14, call for you on channel 3." At this point, the called phone switches to channel 3 and starts making ringing sounds (or playing some melody the owner was given as a birthday present).

2.7.2 Second-Generation (2G) Mobile Phones: Digital Voice

The first generation of mobile phones was analog; the second generation is digital. Switching to digital has several advantages. It provides capacity gains by allowing voice signals to be digitized and compressed. It improves security by allowing voice and control signals to be encrypted. This in turn deters fraud and eavesdropping, whether from intentional scanning or echoes of other calls due to RF propagation. Finally, it enables new services such as text messaging.

Just as there was no worldwide standardization during the first generation, there was also no worldwide standardization during the second, either. Several different systems were developed, and three have been widely deployed. **D-AMPS** (**Digital Advanced Mobile Phone System**) is a digital version of AMPS that coexists with AMPS and uses TDM to place multiple calls on the same frequency channel. It is described in International Standard IS-54 and its successor IS-136. **GSM** (**Global System for Mobile communications**) has emerged as the dominant system, and while it was slow to catch on in the U.S. it is now used virtually everywhere in the world. Like D-AMPS, GSM is based on a mix of FDM and TDM. **CDMA** (**Code Division Multiple Access**), described in **International Standard IS-95**, is a completely different kind of system and is based on neither FDM mor TDM. While CDMA has not become the dominant 2G system, its technology has become the basis for 3G systems.

Also, the name **PCS** (**Personal Communications Services**) is sometimes used in the marketing literature to indicate a second-generation (i.e., digital) system. Originally it meant a mobile phone using the 1900 MHz band, but that distinction is rarely made now.

We will now describe GSM, since it is the dominant 2G system. In the next section we will have more to say about CDMA when we describe 3G systems.

GSM—The Global System for Mobile Communications

GSM started life in the 1980s as an effort to produce a single European 2G standard. The task was assigned to a telecommunications group called (in French) Groupe Specialé Mobile. The first GSM systems were deployed starting in 1991 and were a quick success. It soon became clear that GSM was going to be more than a European success, with uptake stretching to countries as far away as Australia, so GSM was renamed to have a more worldwide appeal.

GSM and the other mobile phone systems we will study retain from 1G systems a design based on cells, frequency reuse across cells, and mobility with handoffs as subscribers move. It is the details that differ. Here, we will briefly discuss some of the main properties of GSM. However, the printed GSM standard is over 5000 [sic] pages long. A large fraction of this material relates to engineering aspects of the system, especially the design of receivers to handle multipath signal propagation, and synchronizing transmitters and receivers. None of this will be even mentioned here.

Fig. 2-46 shows that the GSM architecture is similar to the AMPS architecture, though the components have different names. The mobile itself is now divided into the handset and a removable chip with subscriber and account information called a **SIM card**, short for **Subscriber Identity Module**. It is the SIM card that activates the handset and contains secrets that let the mobile and the network identify each other and encrypt conversations. A SIM card can be removed and plugged into a different handset to turn that handset into your mobile as far as the network is concerned.

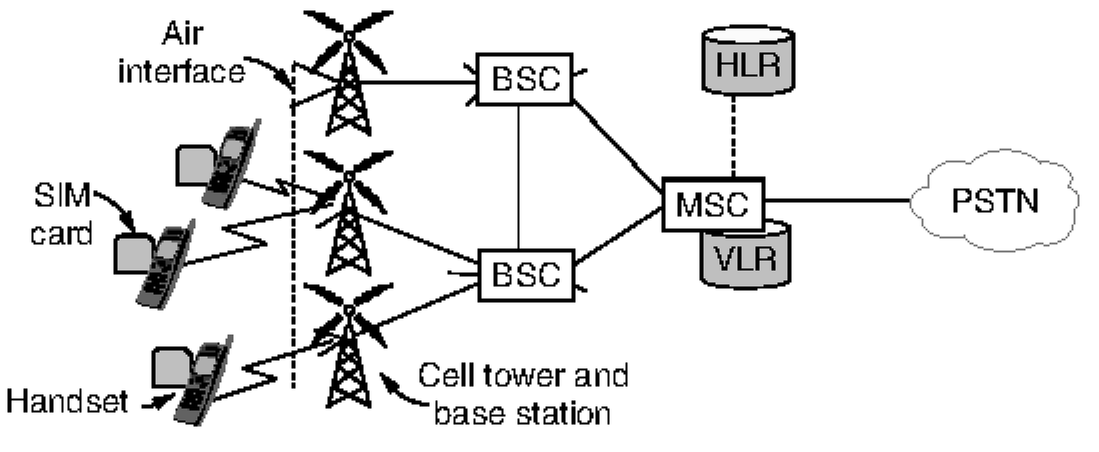


Figure 2-46. GSM mobile network architecture.

The mobile talks to cell base stations over an **air interface** that we will describe in a moment. The cell base stations are each connected to a **BSC** (**Base Station Controller**) that controls the radio resources of cells and handles handoff. The BSC in turn is connected to an MSC (as in AMPS) that routes calls and connects to the PSTN (Public Switched Telephone Network).

To be able to route calls, the MSC needs to know where mobiles can currently be found. It maintains a database of nearby mobiles that are associated with the cells it manages. This database is called the **VLR** (**Visitor Location Register**). There is also a database in the mobile network that gives the last known location of each mobile. It is called the **HLR** (**Home Location Register**). This database is used to route incoming calls to the right locations. Both databases must be kept up to date as mobiles move from cell to cell.

We will now describe the air interface in some detail. GSM runs on a range of frequencies worldwide, including 900, 1800, and 1900 MHz. More spectrum is allocated than for AMPS in order to support a much larger number of users. GSM is a frequency division duplex cellular system, like AMPS. That is, each mobile transmits on one frequency and receives on another, higher frequency (55 MHz higher for GSM versus 80 MHz higher for AMPS). However, unlike with AMPS, with GSM a single frequency pair is split by time-division multiplexing into time slots. In this way it is shared by multiple mobiles.

To handle multiple mobiles, GSM channels are much wider than the AMPS channels (200-kHz versus 30 kHz). One 200-kHz channel is shown in Fig. 2-47. A GSM system operating in the 900-MHz region has 124 pairs of simplex channels. Each simplex channel is 200 kHz wide and supports eight separate connections on it, using time division multiplexing. Each currently active station is assigned one time slot on one channel pair. Theoretically, 992 channels can be supported in each cell, but many of them are not available, to avoid frequency conflicts with neighboring cells. In Fig. 2-47, the eight shaded time slots all belong to the same connection, four of them in each direction. Transmitting and receiving does not happen in the same time slot because the GSM radios cannot transmit and receive at the same time and it takes time to switch from one to the other. If the mobile device assigned to 890.4/935.4 MHz and time slot 2 wanted to transmit to the base station, it would use the lower four shaded slots (and the ones following them in time), putting some data in each slot until all the data had been sent.

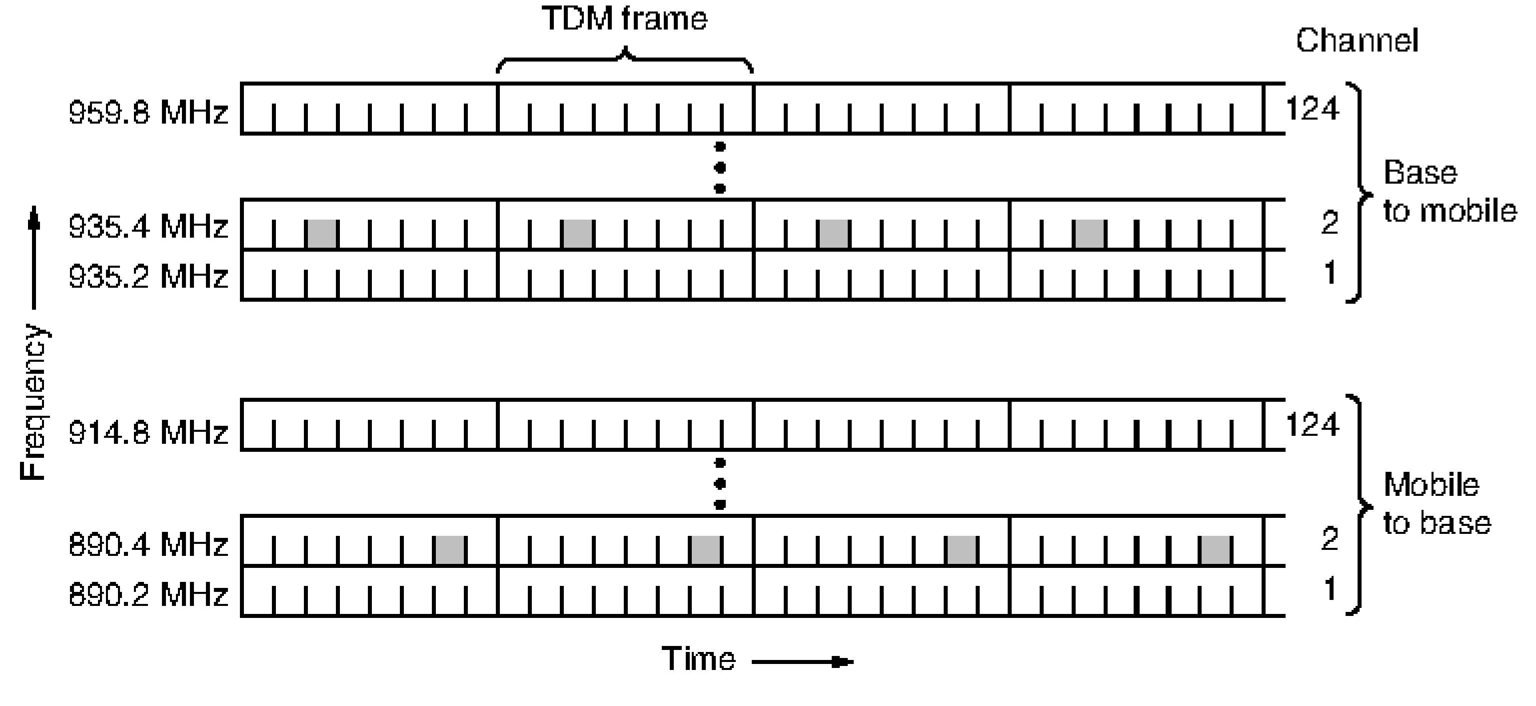


Figure 2-47. GSM uses 124 frequency channels, each of which uses an eight-slot TDM system.

The TDM slots shown in Fig. 2-47 are part of a complex framing hierarchy. Each TDM slot has a specific structure, and groups of TDM slots form multiframes, also with a specific structure. A simplified version of this hierarchy is shown in Fig. 2-48. Here we can see that each TDM slot consists of a 148-bit data frame that occupies the channel for 577 μsec (including a 30-μsec guard time after each slot). Each data frame starts and ends with three 0 bits, for frame delineation purposes. It also contains two 57-bit *Information* fields, each one having a control bit that indicates whether the following *Information* field is for voice or data. Between the *Information* fields is a 26-bit *Sync* (training) field that is used by the receiver to synchronize to the sender's frame boundaries.

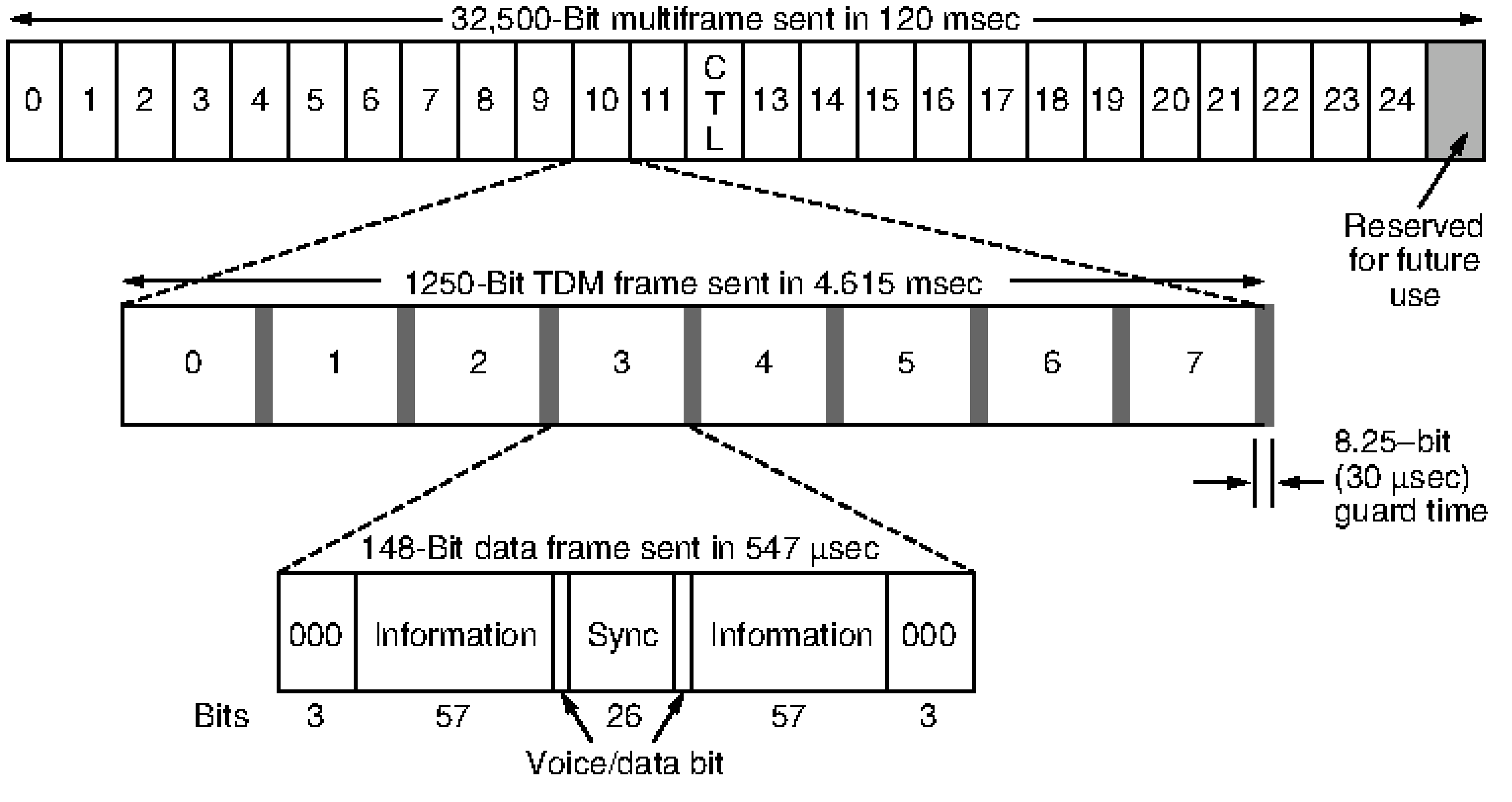


Figure 2-48. A portion of the GSM framing structure.

A data frame is transmitted in 547 μsec, but a transmitter is only allowed to send one data frame every 4.615 msec, since it is sharing the channel with seven other stations. The gross rate of each channel is 270,833 bps, divided among eight users. However, as with AMPS, the overhead eats up a large fraction of the bandwidth, ultimately leaving 24.7 kbps worth of payload per user before error correction. After error correction, 13 kbps is left for speech. While this is substantially less than 64 kbps PCM for uncompressed voice signals in the fixed telephone network, compression on the mobile device can reach these levels with little loss of quality.

As can be seen from Fig. 2-48, eight data frames make up a TDM frame and 26 TDM frames make up a 120-msec multiframe. Of the 26 TDM frames in a multiframe, slot 12 is used for control and slot 25 is reserved for future use, so only 24 are available for user traffic.

However, in addition to the 26-slot multiframe shown in Fig. 2-48, a 51-slot multiframe (not shown) is also used. Some of these slots are used to hold several control channels used to manage the system. The **broadcast control channel** is a continuous stream of output from the base station containing the base station's identity and the channel status. All mobile stations monitor their signal strength to see when they have moved into a new cell.

The **dedicated control channel** is used for location updating, registration, and call setup. In particular, each BSC maintains a database of mobile stations currently under its jurisdiction, the VLR. Information needed to maintain the VLR is sent on the dedicated control channel.

Finally, there is the **common control channel**, which is split up into three logical subchannels. The first of these subchannels is the **paging channel**, which the base station uses to announce incoming calls. Each mobile station monitors it continuously to watch for calls it should answer. The second is the **random access channel**, which allows users to request a slot on the dedicated control channel. If two requests collide, they are garbled and have to be retried later. Using the dedicated control channel slot, the station can set up a call. The assigned slot is announced on the third subchannel, the **access grant channel**.

Finally, GSM differs from AMPS in how handoff is handled. In AMPS, the MSC manages it completely without help from the mobile devices. With time slots in GSM, the mobile is neither sending nor receiving most of the time. The idle slots are an opportunity for the mobile to measure signal quality to other nearby base stations. It does so and sends this information to the BSC. The BSC can use it to determine when a mobile is leaving one cell and entering another so it can perform the handoff. This design is called **MAHO (Mobile Assisted HandOff)**.

2.7.3 Third-Generation (3G) Mobile Phones: Digital Voice and Data

The first generation of mobile phones was analog voice, and the second generation was digital voice. The third generation of mobile phones, or **3G** as it is called, is all about digital voice *and* data.

A number of factors are driving the industry. First, data traffic already exceeds voice traffic on the fixed network and is growing exponentially, whereas voice traffic is essentially flat. Many industry experts expect data traffic to dominate voice on mobile devices as well soon. Second, the telephone, entertainment, and computer industries have all gone digital and are rapidly converging. Many people are drooling over lightweight, portable devices that act as a telephone, music and video player, email terminal, Web interface, gaming machine, and more, all with worldwide wireless connectivity to the Internet at high bandwidth.

Apple's iPhone is a good example of this kind of 3G device. With it, people get hooked on wireless data services, and AT&T wireless data volumes are rising steeply with the popularity of iPhones. The trouble is, the iPhone uses a 2.5G network (an enhanced 2G network, but not a true 3G network) and there is not enough data capacity to keep users happy. 3G mobile telephony is all about providing enough wireless bandwidth to keep these future users happy.

ITU tried to get a bit more specific about this vision starting back around 1992. It issued a blueprint for getting there called **IMT-2000**, where IMT stood for **International Mobile Telecommunications**. The basic services that the IMT-2000 network was supposed to provide to its users are:

1. High-quality voice transmission.
2. Messaging (replacing email, fax, SMS, chat, etc.).
3. Multimedia (playing music, viewing videos, films, television, etc.).
4. Internet access (Web surfing, including pages with audio and video).

Additional services might be video conferencing, telepresence, group game playing, and m-commerce (waving your telephone at the cashier to pay in a store). Furthermore, all these services are supposed to be available worldwide (with automatic connection via a satellite when no terrestrial network can be located), instantly (always on), and with quality of service guarantees.

ITU envisioned a single worldwide technology for IMT-2000, so manufacturers could build a single device that could be sold and used anywhere in the world (like CD players and computers and unlike mobile phones and televisions). Having a single technology would also make life much simpler for network operators and would encourage more people to use the services. Format wars, such as the Betamax versus VHS battle with videorecorders, are not good for business.

As it turned out, this was a bit optimistic. The number 2000 stood for three things: (1) the year it was supposed to go into service, (2) the frequency it was supposed to operate at (in MHz), and (3) the bandwidth the service should have (in kbps). It did not make it on any of the three counts. Nothing was implemented by 2000. ITU recommended that all governments reserve spectrum at 2 GHz so devices could roam seamlessly from country to country. China reserved the required bandwidth but nobody else did. Finally, it was recognized that 2 Mbps is not currently feasible for users who are *too* mobile (due to the difficulty of performing handoffs quickly enough). More realistic is 2 Mbps for stationary indoor users (which will compete head-on with ADSL), 384 kbps for people walking, and 144 kbps for connections in cars.

Despite these initial setbacks, much has been accomplished since then. Several IMT proposals were made and, after some winnowing, it came down to two main ones. The first one, **WCDMA** (**Wideband CDMA**), was proposed by Ericsson and was pushed by the European Union, which called it **UMTS** (**Universal Mobile Telecommunications System**). The other contender was **CDMA2000**, proposed by Qualcomm.

Both of these systems are more similar than different in that they are based on broadband CDMA; WCDMA uses 5-MHz channels and CDMA2000 uses 1.25-MHz channels. If the Ericsson and Qualcomm engineers were put in a room and told to come to a common design, they probably could find one fairly quickly. The trouble is that the real problem is not engineering, but politics (as usual). Europe wanted a system that interworked with GSM, whereas the U.S. wanted a system that was compatible with one already widely deployed in the U.S. (IS-95). Each side also supported its local company (Ericsson is based in Sweden; Qualcomm is in California). Finally, Ericsson and Qualcomm were involved in numerous lawsuits over their respective CDMA patents.

Worldwide, 10–15% of mobile subscribers already use 3G technologies. In North America and Europe, around a third of mobile subscribers are 3G. Japan was an early adopter and now nearly all mobile phones in Japan are 3G. These figures include the deployment of both UMTS and CDMA2000, and 3G continues to be one great cauldron of activity as the market shakes out. To add to the confusion, UMTS became a single 3G standard with multiple incompatible options, including CDMA2000. This change was an effort to unify the various camps, but it just papers over the technical differences and obscures the focus of ongoing efforts. We will use UMTS to mean WCDMA, as distinct from CDMA2000.

We will focus our discussion on the use of CDMA in cellular networks, as it is the distinguishing feature of both systems. CDMA is neither FDM nor TDM but a kind of mix in which each user sends on the same frequency band at the same time. When it was first proposed for cellular systems, the industry gave it approximately the same reaction that Columbus first got from Queen Isabella when he proposed reaching India by sailing in the wrong direction. However, through the persistence of a single company, Qualcomm, CDMA succeeded as a 2G system (IS-95) and matured to the point that it became the technical basis for 3G.

To make CDMA work in the mobile phone setting requires more than the basic CDMA technique that we described in the previous section. Specifically, we described synchronous CDMA, in which the chip sequences are exactly orthogonal. This design works when all users are synchronized on the start time of their chip sequences, as in the case of the base station transmitting to mobiles. The base station can transmit the chip sequences starting at the same time so that the signals will be orthogonal and able to be separated. However, it is difficult to synchronize the transmissions of independent mobile phones. Without care, their transmissions would arrive at the base station at different times, with no guarantee of orthogonality. To let mobiles send to the base station without synchronization, we want code sequences that are orthogonal to each other at all possible offsets, not simply when they are aligned at the start.

While it is not possible to find sequences that are exactly orthogonal for this general case, long pseudorandom sequences come close enough. They have the property that, with high probability, they have a low **cross-correlation** with each other at all offsets. This means that when one sequence is multiplied by another sequence and summed up to compute the inner product, the result will be small; it would be zero if they were orthogonal. (Intuitively, random sequences should always look different from each other. Multiplying them together should then produce a random signal, which will sum to a small result.) This lets a receiver filter unwanted transmissions out of the received signal. Also, the **auto-correlation** of pseudorandom sequences is also small, with high probability, except at a zero offset. This means that when one sequence is multiplied by a delayed copy of itself and summed, the result will be small, except when the delay is zero. (Intuitively, a delayed random sequence looks like a different random sequence, and we are back to the cross-correlation case.) This lets a receiver lock onto the beginning of the wanted transmission in the received signal.

The use of pseudorandom sequences lets the base station receive CDMA messages from unsynchronized mobiles. However, an implicit assumption in our discussion of CDMA is that the power levels of all mobiles are the same at the receiver. If they are not, a small cross-correlation with a powerful signal might overwhelm a large auto-correlation with a weak signal. Thus, the transmit power on mobiles must be controlled to minimize interference between competing signals. It is this interference that limits the capacity of CDMA systems.

The power levels received at a base station depend on how far away the transmitters are as well as how much power they transmit. There may be many mobile stations at varying distances from the base station. A good heuristic to equalize the received power is for each mobile station to transmit to the base station at the inverse of the power level it receives from the base station. In other words, a mobile station receiving a weak signal from the base station will use more power than one getting a strong signal. For more accuracy, the base station also gives each mobile feedback to increase, decrease, or hold steady its transmit power. The feedback is frequent (1500 times per second) because good power control is important to minimize interference.

Another improvement over the basic CDMA scheme we described earlier is to allow different users to send data at different rates. This trick is accomplished naturally in CDMA by fixing the rate at which chips are transmitted and assigning users chip sequences of different lengths. For example, in WCDMA, the chip rate is 3.84 Mchips/sec and the spreading codes vary from 4 to 256 chips. With a 256-chip code, around 12 kbps is left after error correction, and this capacity is sufficient for a voice call. With a 4-chip code, the user data rate is close to 1 Mbps. Intermediate-length codes give intermediate rates; to get to multiple Mbps, the mobile must use more than one 5-MHz channel at once.

Now let us describe the advantages of CDMA, given that we have dealt with the problems of getting it to work. It has three main advantages. First, CDMA can improve capacity by taking advantage of small periods when some transmitters are silent. In polite voice calls, one party is silent while the other talks. On average, the line is busy only 40% of the time. However, the pauses may be small and are difficult to predict. With TDM or FDM systems, it is not possible to reassign time slots or frequency channels quickly enough to benefit from these small silences. However, in CDMA, by simply not transmitting one user lowers the interference for other users, and it is likely that some fraction of users will not be transmitting in a busy cell at any given time. Thus CDMA takes advantage of expected silences to allow a larger number of simultaneous calls.

Second, with CDMA each cell uses the same frequencies. Unlike GSM and AMPS, FDM is not needed to separate the transmissions of different users. This eliminates complicated frequency planning tasks and improves capacity. It also makes it easy for a base station to use multiple directional antennas, or **sectored antennas**, instead of an omnidirectional antenna. Directional antennas concentrate a signal in the intended direction and reduce the signal, and hence interference, in other directions. This in turn increases capacity. Three sector designs are common. The base station must track the mobile as it moves from sector to sector. This tracking is easy with CDMA because all frequencies are used in all sectors.

Third, CDMA facilitates **soft handoff**, in which the mobile is acquired by the new base station before the previous one signs off. In this way there is no loss of continuity. Soft handoff is shown in Fig. 2-49. It is easy with CDMA because all frequencies are used in each cell. The alternative is a **hard handoff**, in which the old base station drops the call before the new one acquires it. If the new one is unable to acquire it (e.g., because there is no available frequency), the call is disconnected abruptly. Users tend to notice this, but it is inevitable occasionally with the current design. Hard handoff is the norm with FDM designs to avoid the cost of having the mobile transmit or receive on two frequencies simultaneously.

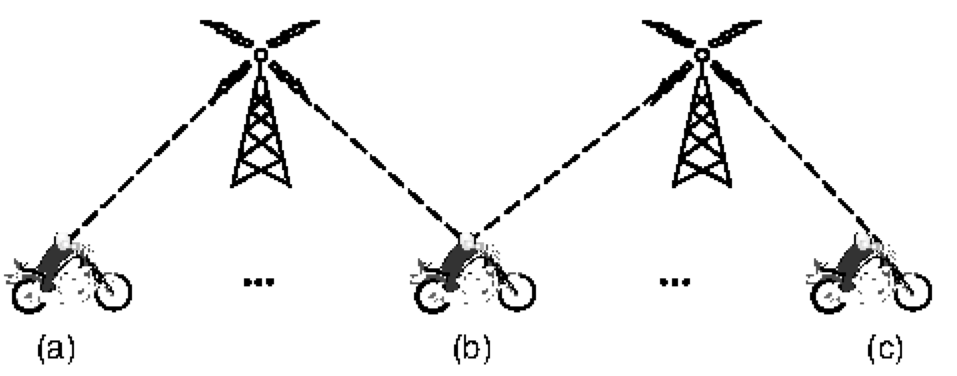


Figure 2-49. Soft handoff (a) before, (b) during, and (c) after.

Much has been written about 3G, most of it praising it as the greatest thing since sliced bread. Meanwhile, many operators have taken cautious steps in the direction of 3G by going to what is sometimes called **2.5G**, although 2.1G might be more accurate. One such system is **EDGE** (**Enhanced Data rates for GSM Evolution**), which is just GSM with more bits per symbol. The trouble is, more bits per symbol also means more errors per symbol, so EDGE has nine different schemes for modulation and error correction, differing in terms of how much of the bandwidth is devoted to fixing the errors introduced by the higher speed. EDGE is one step along an evolutionary path that is defined from GSM to WCDMA. Similarly, there is an evolutionary path defined for operators to upgrade from IS-95 to CDMA2000 networks.

Even though 3G networks are not fully deployed yet, some researchers regard 3G as a done deal. These people are already working on 4G systems under the name of **LTE** (**Long Term Evolution**). Some of the proposed features of 4G include: high bandwidth; ubiquity (connectivity everywhere); seamless integration with other wired and wireless IP networks, including 802.11 access points; adaptive resource and spectrum management; and high quality of service for multimedia. For more information see Astely et al. (2009) and Larmo et al. (2009).

Meanwhile, wireless networks with 4G levels of performance are already available. The main example is **802.16**, also known as **WiMAX**. For an overview of mobile WiMAX see Ahmadi (2009). To say the industry is in a state of flux is a huge understatement. Check back in a few years to see what has happened.

2.8 CABLE TELEVISION

We have now studied both the fixed and wireless telephone systems in a fair amount of detail. Both will clearly play a major role in future networks. But there is another major player that has emerged over the past decade for Internet access: cable television networks. Many people nowadays get their telephone and Internet service over cable. In the following sections we will look at cable television as a network in more detail and contrast it with the telephone systems we have just studied. Some relevant references for more information are Donaldson and Jones (2001), Dutta-Roy (2001), and Fellows and Jones (2001).

2.8.1 Community Antenna Television

Cable television was conceived in the late 1940s as a way to provide better reception to people living in rural or mountainous areas. The system initially consisted of a big antenna on top of a hill to pluck the television signal out of the air, an amplifier, called the **headend**, to strengthen it, and a coaxial cable to deliver it to people's houses, as illustrated in Fig. 2-50.

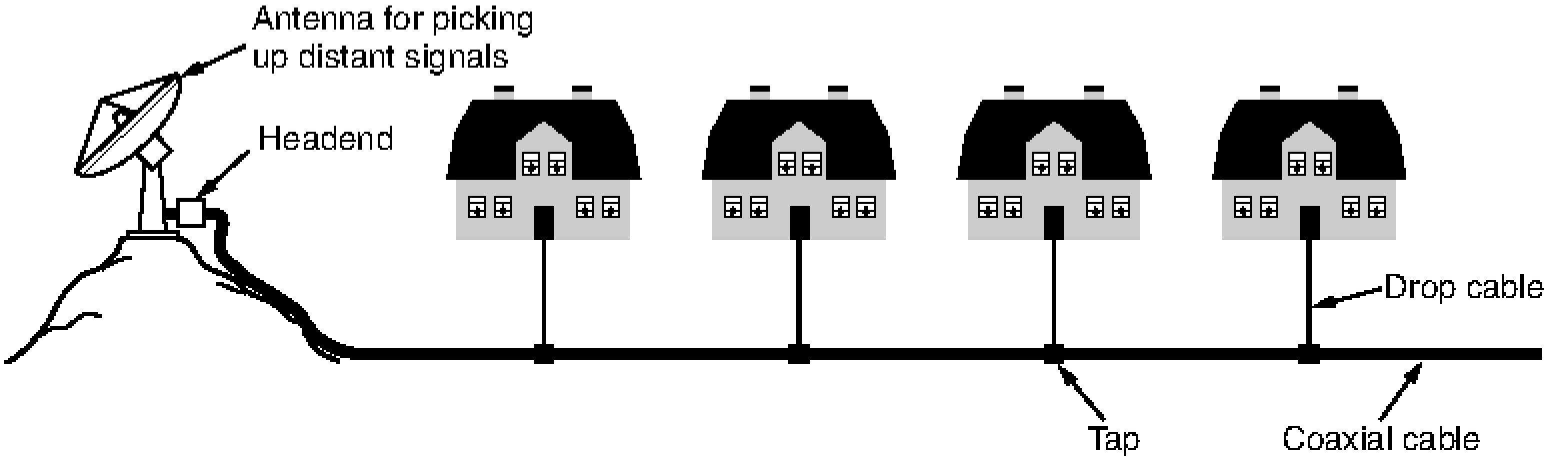


Figure 2-50. An early cable television system.

In the early years, cable television was called **Community Antenna Television**. It was very much a mom-and-pop operation; anyone handy with electronics could set up a service for his town, and the users would chip in to pay the costs. As the number of subscribers grew, additional cables were spliced onto the original cable and amplifiers were added as needed. Transmission was one way, from the headend to the users. By 1970, thousands of independent systems existed.

In 1974, Time Inc. started a new channel, Home Box Office, with new content (movies) distributed only on cable. Other cable-only channels followed, focusing on news, sports, cooking, and many other topics. This development gave rise to two changes in the industry. First, large corporations began buying up existing cable systems and laying new cable to acquire new subscribers. Second, there was now a need to connect multiple systems, often in distant cities, in order to distribute the new cable channels. The cable companies began to lay cable between the cities to connect them all into a single system. This pattern was analogous to what happened in the telephone industry 80 years earlier with the connection of previously isolated end offices to make long-distance calling possible.

2.8.2 Internet over Cable

Over the course of the years the cable system grew and the cables between the various cities were replaced by high-bandwidth fiber, similar to what happened in the telephone system. A system with fiber for the long-haul runs and coaxial cable to the houses is called an **HFC** (**Hybrid Fiber Coax**) system. The electro-optical converters that interface between the optical and electrical parts of the system are called **fiber nodes**. Because the bandwidth of fiber is so much greater than that of coax, a fiber node can feed multiple coaxial cables. Part of a modern HFC system is shown in Fig. 2-51(a).

Over the past decade, many cable operators decided to get into the Internet access business, and often the telephony business as well. Technical differences between the cable plant and telephone plant had an effect on what had to be done to achieve these goals. For one thing, all the one-way amplifiers in the system had to be replaced by two-way amplifiers to support upstream as well as downstream transmissions. While this was happening, early Internet over cable systems used the cable television network for downstream transmissions and a dial-up connection via the telephone network for upstream transmissions. It was a clever workaround, but not much of a network compared to what it could be.

However, there is another difference between the HFC system of Fig. 2-51(a) and the telephone system of Fig. 2-51(b) that is much harder to remove. Down in the neighborhoods, a single cable is shared by many houses, whereas in the telephone system, every house has its own private local loop. When used for television broadcasting, this sharing is a natural fit. All the programs are broadcast on the cable and it does not matter whether there are 10 viewers or 10,000 viewers. When the same cable is used for Internet access, however, it matters a lot if there are 10 users or 10,000. If one user decides to download a very large file, that bandwidth is potentially being taken away from other users. The more users there

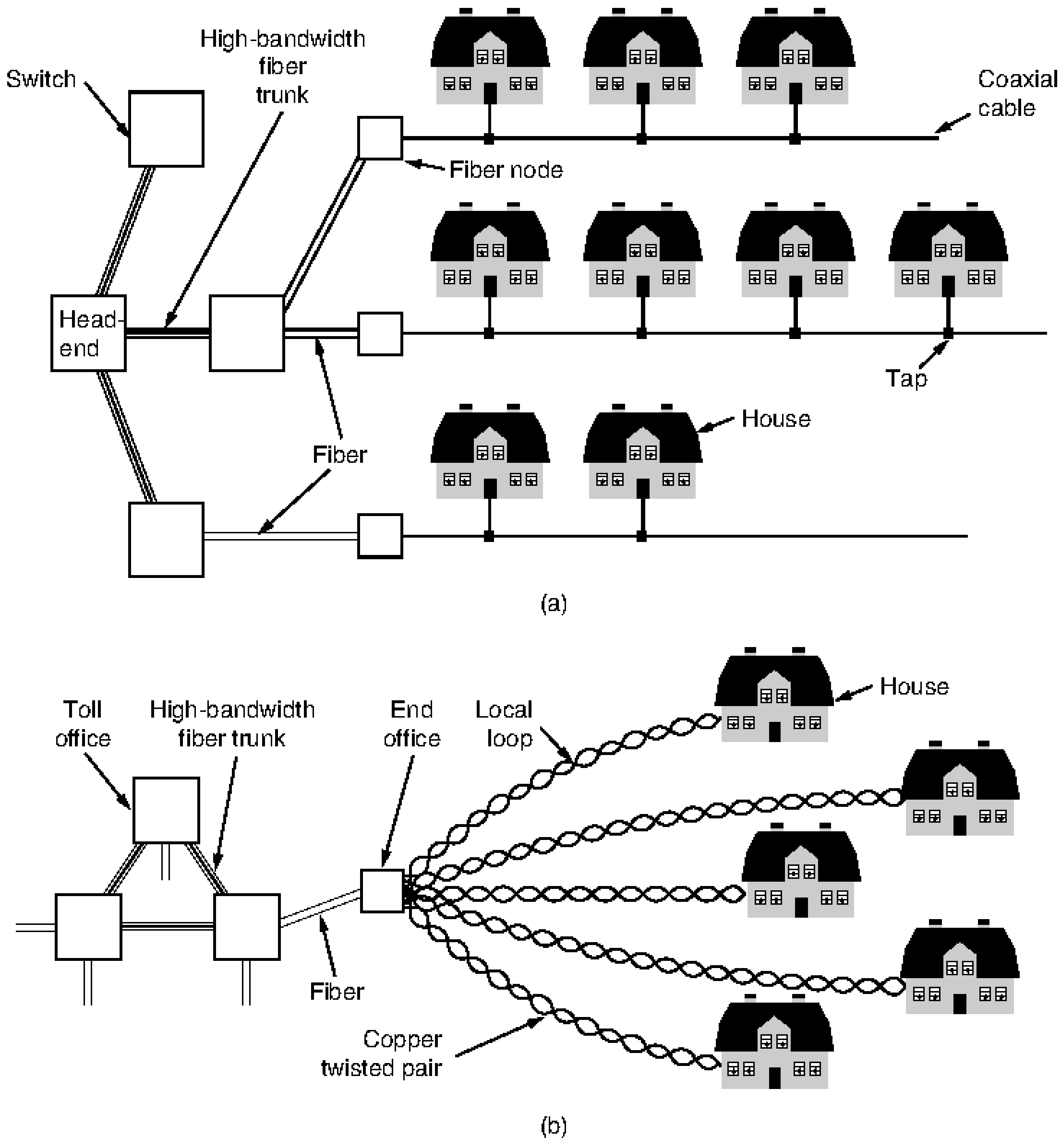


Figure 2-51. (a) Cable television. (b) The fixed telephone system.

are, the more competition there is for bandwidth. The telephone system does not have this particular property: downloading a large file over an ADSL line does not reduce your neighbor's bandwidth. On the other hand, the bandwidth of coax is much higher than that of twisted pairs, so you can get lucky if your neighbors do not use the Internet much.

The way the cable industry has tackled this problem is to split up long cables and connect each one directly to a fiber node. The bandwidth from the headend to each fiber node is effectively infinite, so as long as there are not too many subscribers on each cable segment, the amount of traffic is manageable. Typical cables nowadays have 500–2000 houses, but as more and more people subscribe to Internet over cable, the load may become too great, requiring more splitting and more fiber nodes.

2.8.3 Spectrum Allocation

Throwing off all the TV channels and using the cable infrastructure strictly for Internet access would probably generate a fair number of irate customers, so cable companies are hesitant to do this. Furthermore, most cities heavily regulate what is on the cable, so the cable operators would not be allowed to do this even if they really wanted to. As a consequence, they needed to find a way to have television and Internet peacefully coexist on the same cable.

The solution is to build on frequency division multiplexing. Cable television channels in North America occupy the 54–550 MHz region (except for FM radio, from 88 to 108 MHz). These channels are 6-MHz wide, including guard bands, and can carry one traditional analog television channel or several digital television channels. In Europe the low end is usually 65 MHz and the channels are 6–8 MHz wide for the higher resolution required by PAL and SECAM, but otherwise the allocation scheme is similar. The low part of the band is not used. Modern cables can also operate well above 550 MHz, often at up to 750 MHz or more. The solution chosen was to introduce upstream channels in the 5–42 MHz band (slightly higher in Europe) and use the frequencies at the high end for the downstream signals. The cable spectrum is illustrated in Fig. 2-52.

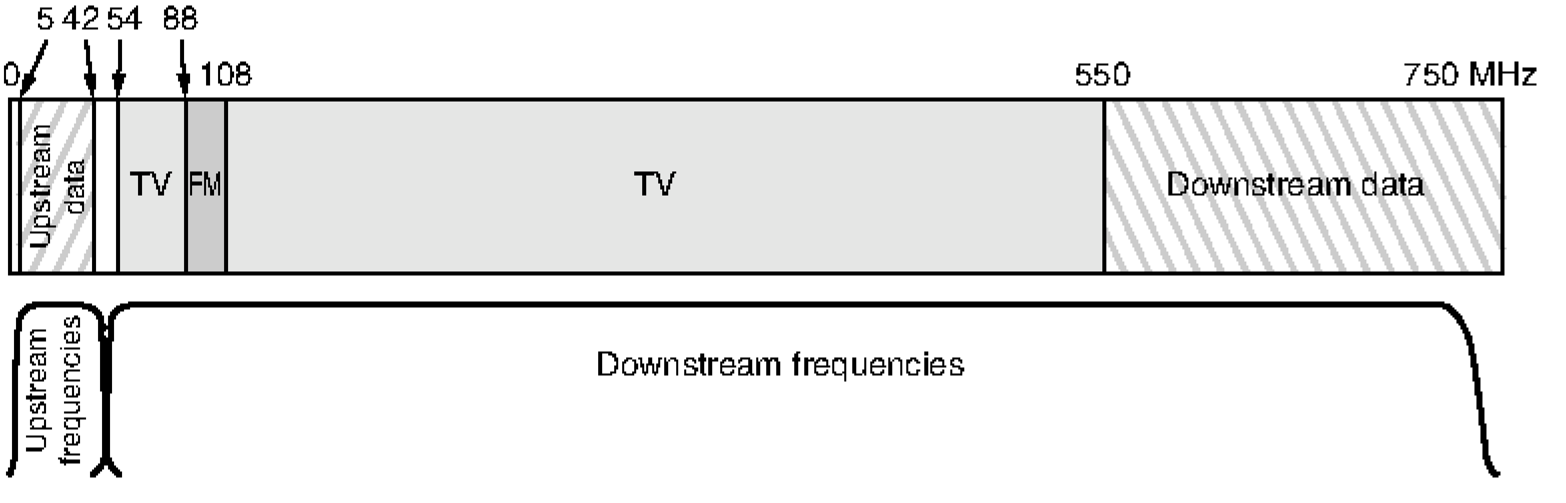


Figure 2-52. Frequency allocation in a typical cable TV system used for Internet access.

Note that since the television signals are all downstream, it is possible to use upstream amplifiers that work only in the 5–42 MHz region and downstream amplifiers that work only at 54 MHz and up, as shown in the figure. Thus, we get an asymmetry in the upstream and downstream bandwidths because more spectrum is available above television than below it. On the other hand, most users want more downstream traffic, so cable operators are not unhappy with this fact of life. As we saw earlier, telephone companies usually offer an asymmetric DSL service, even though they have no technical reason for doing so.

In addition to upgrading the amplifiers, the operator has to upgrade the headend, too, from a dumb amplifier to an intelligent digital computer system with a high-bandwidth fiber interface to an ISP. Often the name gets upgraded as well, from "headend" to **CMTS** (**Cable Modem Termination System**). In the following text, we will refrain from doing a name upgrade and stick with the traditional "headend."

2.8.4 Cable Modems

Internet access requires a cable modem, a device that has two interfaces on it: one to the computer and one to the cable network. In the early years of cable Internet, each operator had a proprietary cable modem, which was installed by a cable company technician. However, it soon became apparent that an open standard would create a competitive cable modem market and drive down prices, thus encouraging use of the service. Furthermore, having the customers buy cable modems in stores and install them themselves (as they do with wireless access points) would eliminate the dreaded truck rolls.

Consequently, the larger cable operators teamed up with a company called CableLabs to produce a cable modem standard and to test products for compliance. This standard, called **DOCSIS** (**Data Over Cable Service Interface Specification**), has mostly replaced proprietary modems. DOCSIS version 1.0 came out in 1997, and was soon followed by DOCSIS 2.0 in 2001. It increased upstream rates to better support symmetric services such as IP telephony. The most recent version of the standard is DOCSIS 3.0, which came out in 2006. It uses more bandwidth to increase rates in both directions. The European version of these standards is called **EuroDOCSIS**. Not all cable operators like the idea of a standard, however, since many of them were making good money leasing their modems to their captive customers. An open standard with dozens of manufacturers selling cable modems in stores ends this lucrative practice.

The modem-to-computer interface is straightforward. It is normally Ethernet, or occasionally USB. The other end is more complicated as it uses all of FDM, TDM, and CDMA to share the bandwidth of the cable between subscribers.

When a cable modem is plugged in and powered up, it scans the downstream channels looking for a special packet periodically put out by the headend to provide system parameters to modems that have just come online. Upon finding this packet, the new modem announces its presence on one of the upstream channels. The headend responds by assigning the modem to its upstream and downstream channels. These assignments can be changed later if the headend deems it necessary to balance the load.

The use of 6-MHz or 8-MHz channels is the FDM part. Each cable modem sends data on one upstream and one downstream channel, or multiple channels under DOCSIS 3.0. The usual scheme is to take each 6 (or 8) MHz downstream channel and modulate it with QAM-64 or, if the cable quality is exceptionally good, QAM-256. With a 6-MHz channel and QAM-64, we get about 36 Mbps. When the overhead is subtracted, the net payload is about 27 Mbps. With QAM-256, the net payload is about 39 Mbps. The European values are 1/3 larger.

For upstream, there is more RF noise because the system was not originally designed for data, and noise from multiple subscribers is funneled to the headend, so a more conservative scheme is used. This ranges from QPSK to QAM-128, where some of the symbols are used for error protection with Trellis Coded Modulation. With fewer bits per symbol on the upstream, the asymmetry between upstream and downstream rates is much more than suggested by Fig. 2-52.

TDM is then used to share bandwidth on the upstream across multiple subscribers. Otherwise their transmissions would collide at the headend. Time is divided into **minislots** and different subscribers send in different minislots. To make this work, the modem determines its distance from the headend by sending it a special packet and seeing how long it takes to get the response. This process is called **ranging**. It is important for the modem to know its distance to get the timing right. Each upstream packet must fit in one or more consecutive minislots at the headend when it is received. The headend announces the start of a new round of minislots periodically, but the starting gun is not heard at all modems simultaneously due to the propagation time down the cable. By knowing how far it is from the headend, each modem can compute how long ago the first minislot really started. Minislot length is network dependent. A typical payload is 8 bytes.

During initialization, the headend assigns each modem to a minislot to use for requesting upstream bandwidth. When a computer wants to send a packet, it transfers the packet to the modem, which then requests the necessary number of minislots for it. If the request is accepted, the headend puts an acknowledgement on the downstream channel telling the modem which minislots have been reserved for its packet. The packet is then sent, starting in the minislot allocated to it. Additional packets can be requested using a field in the header.

As a rule, multiple modems will be assigned the same minislot, which leads to contention. Two different possibilities exist for dealing with it. The first is that CDMA is used to share the minislot between subscribers. This solves the contention problem because all subscribers with a CDMA code sequence can send at the same time, albeit at a reduced rate. The second option is that CDMA is not used, in which case there may be no acknowledgement to the request because of a collision. In this case, the modem just waits a random time and tries again. After each successive failure, the randomization time is doubled. (For readers already somewhat familiar with networking, this algorithm is just slotted ALOHA with binary exponential backoff. Ethernet cannot be used on cable because stations cannot sense the medium. We will come back to these issues in Chap. 4.)

The downstream channels are managed differently from the upstream channels. For starters, there is only one sender (the headend), so there is no contention and no need for minislots, which is actually just statistical time division multiplexing. For another, the amount of traffic downstream is usually much larger than upstream, so a fixed packet size of 204 bytes is used. Part of that is a Reed-Solomon error-correcting code and some other overhead, leaving a user payload of 184 bytes. These numbers were chosen for compatibility with digital television using MPEG-2, so the TV and downstream data channels are formatted the same way. Logically, the connections are as depicted in Fig. 2-53.

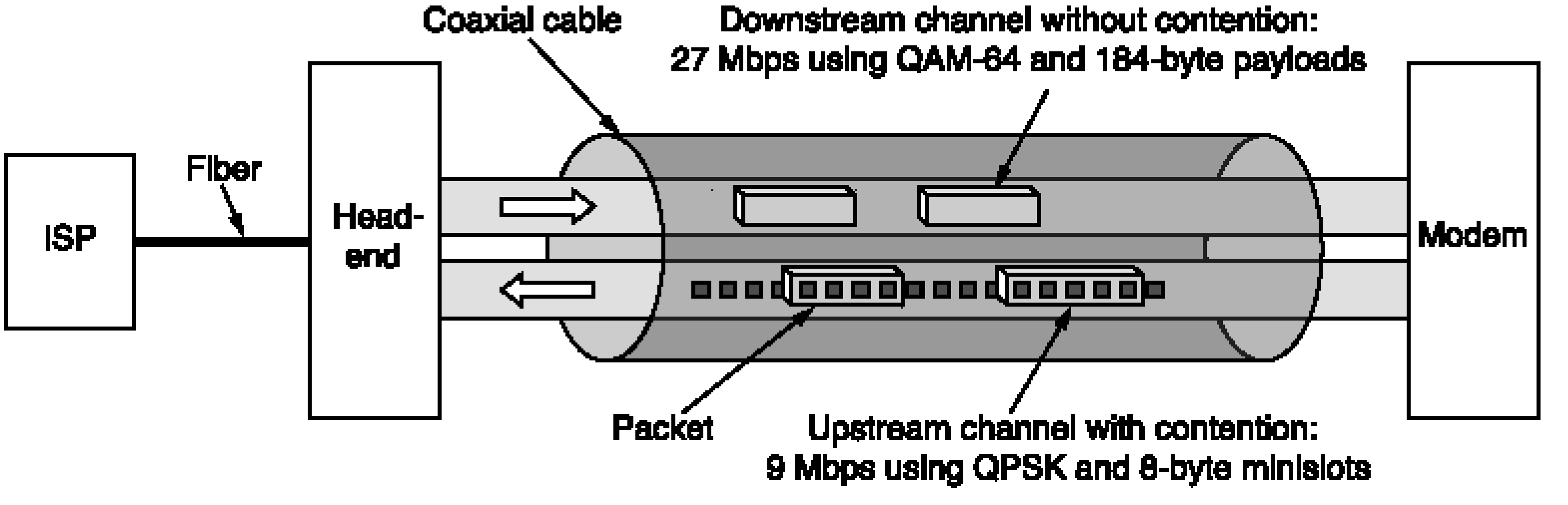


Figure 2-53. Typical details of the upstream and downstream channels in North America.

2.8.5 ADSL Versus Cable

Which is better, ADSL or cable? That is like asking which operating system is better. Or which language is better. Or which religion. Which answer you get depends on whom you ask. Let us compare ADSL and cable on a few points. Both use fiber in the backbone, but they differ on the edge. Cable uses coax; ADSL uses twisted pair. The theoretical carrying capacity of coax is hundreds of times more than twisted pair. However, the full capacity of the cable is not available for data users because much of the cable's bandwidth is wasted on useless stuff such as television programs.

In practice, it is hard to generalize about effective capacity. ADSL providers give specific statements about the bandwidth (e.g., 1 Mbps downstream, 256 kbps upstream) and generally achieve about 80% of it consistently. Cable providers may artificially cap the bandwidth to each user to help them make performance predictions, but they cannot really give guarantees because the effective capacity depends on how many people are currently active on the user's cable segment. Sometimes it may be better than ADSL and sometimes it may be worse. What can be annoying, though, is the unpredictability. Having great service one minute does not guarantee great service the next minute since the biggest bandwidth hog in town may have just turned on his computer.

As an ADSL system acquires more users, their increasing numbers have little effect on existing users, since each user has a dedicated connection. With cable, as more subscribers sign up for Internet service, performance for existing users will drop. The only cure is for the cable operator to split busy cables and connect each one to a fiber node directly. Doing so costs time and money, so there are business pressures to avoid it.

As an aside, we have already studied another system with a shared channel like cable: the mobile telephone system. Here, too, a group of users—we could call them cellmates—share a fixed amount of bandwidth. For voice traffic, which is fairly smooth, the bandwidth is rigidly divided in fixed chunks among the active users using FDM and TDM. But for data traffic, this rigid division is very inefficient because data users are frequently idle, in which case their reserved bandwidth is wasted. As with cable, a more dynamic means is used to allocate the shared bandwidth.

Availability is an issue on which ADSL and cable differ. Everyone has a telephone, but not all users are close enough to their end offices to get ADSL. On the other hand, not everyone has cable, but if you do have cable and the company provides Internet access, you can get it. Distance to the fiber node or headend is not an issue. It is also worth noting that since cable started out as a television distribution medium, few businesses have it.

Being a point-to-point medium, ADSL is inherently more secure than cable. Any cable user can easily read all the packets going down the cable. For this reason, any decent cable provider will encrypt all traffic in both directions. Nevertheless, having your neighbor get your encrypted messages is still less secure than having him not get anything at all.

The telephone system is generally more reliable than cable. For example, it has backup power and continues to work normally even during a power outage. With cable, if the power to any amplifier along the chain fails, all downstream users are cut off instantly.

Finally, most ADSL providers offer a choice of ISPs. Sometimes they are even required to do so by law. Such is not always the case with cable operators.

The conclusion is that ADSL and cable are much more alike than they are different. They offer comparable service and, as competition between them heats up, probably comparable prices.

2.9 SUMMARY

The physical layer is the basis of all networks. Nature imposes two fundamental limits on all channels, and these determine their bandwidth. These limits are the Nyquist limit, which deals with noiseless channels, and the Shannon limit, which deals with noisy channels.

Transmission media can be guided or unguided. The principal guided media are twisted pair, coaxial cable, and fiber optics. Unguided media include terrestrial radio, microwaves, infrared, lasers through the air, and satellites.

Digital modulation methods send bits over guided and unguided media as analog signals. Line codes operate at baseband, and signals can be placed in a passband by modulating the amplitude, frequency, and phase of a carrier. Channels can be shared between users with time, frequency and code division multiplexing.

A key element in most wide area networks is the telephone system. Its main components are the local loops, trunks, and switches. ADSL offers speeds up to 40 Mbps over the local loop by dividing it into many subcarriers that run in parallel. This far exceeds the rates of telephone modems. PONs bring fiber to the home for even greater access rates than ADSL.

Trunks carry digital information. They are multiplexed with WDM to provision many high capacity links over individual fibers, as well as with TDM to share each high rate link between users. Both circuit switching and packet switching are important.

For mobile applications, the fixed telephone system is not suitable. Mobile phones are currently in widespread use for voice, and increasingly for data. They have gone through three generations. The first generation, 1G, was analog and dominated by AMPS. 2G was digital, with GSM presently the most widely deployed mobile phone system in the world. 3G is digital and based on broadband CDMA, with WCDMA and also CDMA2000 now being deployed.

An alternative system for network access is the cable television system. It has gradually evolved from coaxial cable to hybrid fiber coax, and from television to television and Internet. Potentially, it offers very high bandwidth, but the bandwidth in practice depends heavily on the other users because it is shared.

PROBLEMS

1. Compute the Fourier coefficients for the function $f(t) = t \quad (0 \le t \le 1)$.
2. A noiseless 4-kHz channel is sampled every 1 msec. What is the maximum data rate? How does the maximum data rate change if the channel is noisy, with a signal-to-noise ratio of 30 dB?
3. Television channels are 6 MHz wide. How many bits/sec can be sent if four-level digital signals are used? Assume a noiseless channel.
4. If a binary signal is sent over a 3-kHz channel whose signal-to-noise ratio is 20 dB, what is the maximum achievable data rate?
5. What signal-to-noise ratio is needed to put a T1 carrier on a 50-kHz line?
6. What are the advantages of fiber optics over copper as a transmission medium? Is there any downside of using fiber optics over copper?

7. How much bandwidth is there in 0.1 microns of spectrum at a wavelength of 1 micron?

8. It is desired to send a sequence of computer screen images over an optical fiber. The screen is 2560×1600 pixels, each pixel being 24 bits. There are 60 screen images per second. How much bandwidth is needed, and how many microns of wavelength are needed for this band at 1.30 microns?

9. Is the Nyquist theorem true for high-quality single-mode optical fiber or only for copper wire?

10. Radio antennas often work best when the diameter of the antenna is equal to the wavelength of the radio wave. Reasonable antennas range from 1 cm to 5 meters in diameter. What frequency range does this cover?

11. A laser beam 1 mm wide is aimed at a detector 1 mm wide 100 m away on the roof of a building. How much of an angular diversion (in degrees) does the laser have to have before it misses the detector?

12. The 66 low-orbit satellites in the Iridium project are divided into six necklaces around the earth. At the altitude they are using, the period is 90 minutes. What is the average interval for handoffs for a stationary transmitter?

13. Calculate the end-to-end transit time for a packet for both GEO (altitude: 35,800 km), MEO (altitude: 18,000 km) and LEO (altitude: 750 km) satellites.

14. What is the latency of a call originating at the North Pole to reach the South Pole if the call is routed via Iridium satellites? Assume that the switching time at the satellites is 10 microseconds and earth's radius is 6371 km.

15. What is the minimum bandwidth needed to achieve a data rate of B bits/sec if the signal is transmitted using NRZ, MLT-3, and Manchester encoding? Explain your answer.

16. Prove that in 4B/5B encoding, a signal transition will occur at least every four bit times.

17. How many end office codes were there pre-1984, when each end office was named by its three-digit area code and the first three digits of the local number? Area codes started with a digit in the range 2–9, had a 0 or 1 as the second digit, and ended with any digit. The first two digits of a local number were always in the range 2–9. The third digit could be any digit.

18. A simple telephone system consists of two end offices and a single toll office to which each end office is connected by a 1-MHz full-duplex trunk. The average telephone is used to make four calls per 8-hour workday. The mean call duration is 6 min. Ten percent of the calls are long distance (i.e., pass through the toll office). What is the maximum number of telephones an end office can support? (Assume 4 kHz per circuit.) Explain why a telephone company may decide to support a lesser number of telephones than this maximum number at the end office.

19. A regional telephone company has 10 million subscribers. Each of their telephones is connected to a central office by a copper twisted pair. The average length of these twisted pairs is 10 km. How much is the copper in the local loops worth? Assume

that the cross section of each strand is a circle 1 mm in diameter, the density of copper is 9.0 grams/cm^3, and that copper sells for \$6 per kilogram.

20. Is an oil pipeline a simplex system, a half-duplex system, a full-duplex system, or none of the above? What about a river or a walkie-talkie-style communication?

21. The cost of a fast microprocessor has dropped to the point where it is now possible to put one in each modem. How does that affect the handling of telephone line errors? Does it negate the need for error checking/correction in layer 2?

22. A modem constellation diagram similar to Fig. 2-23 has data points at the following coordinates: (1, 1), (1, −1), (−1, 1), and (−1, −1). How many bps can a modem with these parameters achieve at 1200 symbols/second?

23. What is the maximum bit rate achievable in a V.32 standard modem if the baud rate is 1200 and no error correction is used?

24. How many frequencies does a full-duplex QAM-64 modem use?

25. Ten signals, each requiring 4000 Hz, are multiplexed onto a single channel using FDM. What is the minimum bandwidth required for the multiplexed channel? Assume that the guard bands are 400 Hz wide.

26. Why has the PCM sampling time been set at 125 μsec?

27. What is the percent overhead on a T1 carrier? That is, what percent of the 1.544 Mbps are not delivered to the end user? How does it relate to the percent overhead in OC-1 or OC-768 lines?

28. Compare the maximum data rate of a noiseless 4-kHz channel using
(a) Analog encoding (e.g., QPSK) with 2 bits per sample.
(b) The T1 PCM system.

29. If a T1 carrier system slips and loses track of where it is, it tries to resynchronize using the first bit in each frame. How many frames will have to be inspected on average to resynchronize with a probability of 0.001 of being wrong?

30. What is the difference, if any, between the demodulator part of a modem and the coder part of a codec? (After all, both convert analog signals to digital ones.)

31. SONET clocks have a drift rate of about 1 part in 10^9. How long does it take for the drift to equal the width of 1 bit? Do you see any practical implications of this calculation? If so, what?

32. How long will it take to transmit a 1-GB file from one VSAT to another using a hub as shown in Figure 2-17? Assume that the uplink is 1 Mbps, the downlink is 7 Mbps, and circuit switching is used with 1.2 sec circuit setup time.

33. Calculate the transmit time in the previous problem if packet switching is used instead. Assume that the packet size is 64 KB, the switching delay in the satellite and hub is 10 microseconds, and the packet header size is 32 bytes.

34. In Fig. 2-40, the user data rate for OC-3 is stated to be 148.608 Mbps. Show how this number can be derived from the SONET OC-3 parameters. What will be the gross, SPE, and user data rates of an OC-3072 line?

35. To accommodate lower data rates than STS-1, SONET has a system of virtual tributaries (VTs). A VT is a partial payload that can be inserted into an STS-1 frame and combined with other partial payloads to fill the data frame. VT1.5 uses 3 columns, VT2 uses 4 columns, VT3 uses 6 columns, and VT6 uses 12 columns of an STS-1 frame. Which VT can accommodate

(a) A DS-1 service (1.544 Mbps)?
(b) European CEPT-1 service (2.048 Mbps)?
(c) A DS-2 service (6.312 Mbps)?

36. What is the available user bandwidth in an OC-12c connection?

37. Three packet-switching networks each contain n nodes. The first network has a star topology with a central switch, the second is a (bidirectional) ring, and the third is fully interconnected, with a wire from every node to every other node. What are the best-, average-, and worst-case transmission paths in hops?

38. Compare the delay in sending an x-bit message over a k-hop path in a circuit-switched network and in a (lightly loaded) packet-switched network. The circuit setup time is s sec, the propagation delay is d sec per hop, the packet size is p bits, and the data rate is b bps. Under what conditions does the packet network have a lower delay? Also, explain the conditions under which a packet-switched network is preferable to a circuit-switched network.

39. Suppose that x bits of user data are to be transmitted over a k-hop path in a packet-switched network as a series of packets, each containing p data bits and h header bits, with $x \gg p + h$. The bit rate of the lines is b bps and the propagation delay is negligible. What value of p minimizes the total delay?

40. In a typical mobile phone system with hexagonal cells, it is forbidden to reuse a frequency band in an adjacent cell. If 840 frequencies are available, how many can be used in a given cell?

41. The actual layout of cells is seldom as regular that as shown in Fig. 2-45. Even the shapes of individual cells are typically irregular. Give a possible reason why this might be. How do these irregular shapes affect frequency assignment to each cell?

42. Make a rough estimate of the number of PCS microcells 100 m in diameter it would take to cover San Francisco (120 square km).

43. Sometimes when a mobile user crosses the boundary from one cell to another, the current call is abruptly terminated, even though all transmitters and receivers are functioning perfectly. Why?

44. Suppose that A, B, and C are simultaneously transmitting 0 bits, using a CDMA system with the chip sequences of Fig. 2-28(a). What is the resulting chip sequence?

45. Consider a different way of looking at the orthogonality property of CDMA chip sequences. Each bit in a pair of sequences can match or not match. Express the orthogonality property in terms of matches and mismatches.

46. A CDMA receiver gets the following chips: $(-1\ +1\ -3\ +1\ -1\ -3\ +1\ +1)$. Assuming the chip sequences defined in Fig. 2-28(a), which stations transmitted, and which bits did each one send?

47. In Figure 2-28, there are four stations that can transmit. Suppose four more stations are added. Provide the chip sequences of these stations.

48. At the low end, the telephone system is star shaped, with all the local loops in a neighborhood converging on an end office. In contrast, cable television consists of a single long cable snaking its way past all the houses in the same neighborhood. Suppose that a future TV cable were 10-Gbps fiber instead of copper. Could it be used to simulate the telephone model of everybody having their own private line to the end office? If so, how many one-telephone houses could be hooked up to a single fiber?

49. A cable company decides to provide Internet access over cable in a neighborhood consisting of 5000 houses. The company uses a coaxial cable and spectrum allocation allowing 100 Mbps downstream bandwidth per cable. To attract customers, the company decides to guarantee at least 2 Mbps downstream bandwidth to each house at any time. Describe what the cable company needs to do to provide this guarantee.

50. Using the spectral allocation shown in Fig. 2-52 and the information given in the text, how many Mbps does a cable system allocate to upstream and how many to downstream?

51. How fast can a cable user receive data if the network is otherwise idle? Assume that the user interface is
(a) 10-Mbps Ethernet
(b) 100-Mbps Ethernet
(c) 54-Mbps Wireless.

52. Multiplexing STS-1 multiple data streams, called tributaries, plays an important role in SONET. A 3:1 multiplexer multiplexes three input STS-1 tributaries onto one output STS-3 stream. This multiplexing is done byte for byte. That is, the first three output bytes are the first bytes of tributaries 1, 2, and 3, respectively. the next three output bytes are the second bytes of tributaries 1, 2, and 3, respectively, and so on. Write a program that simulates this 3:1 multiplexer. Your program should consist of five processes. The main process creates four processes, one each for the three STS-1 tributaries and one for the multiplexer. Each tributary process reads in an STS-1 frame from an input file as a sequence of 810 bytes. They send their frames (byte by byte) to the multiplexer process. The multiplexer process receives these bytes and outputs an STS-3 frame (byte by byte) by writing it to standard output. Use pipes for communication among processes.

53. Write a program to implement CDMA. Assume that the length of a chip sequence is eight and the number of stations transmitting is four. Your program consists of three sets of processes: four transmitter processes (t0, t1, t2, and t3), one joiner process, and four receiver processes (r0, r1, r2, and r3). The main program, which also acts as the joiner process first reads four chip sequences (bipolar notation) from the standard input and a sequence of 4 bits (1 bit per transmitter process to be transmitted), and forks off four pairs of transmitter and receiver processes. Each pair of transmitter/receiver processes (t0,r0; t1,r1; t2,r2; t3,r3) is assigned one chip sequence and each transmitter process is assigned 1 bit (first bit to t0, second bit to t1, and so on). Next, each transmitter process computes the signal to be transmitted (a sequence of 8 bits) and sends it to the joiner process. After receiving signals from all four transmitter processes, the joiner process combines the signals and sends the combined signal to

the four receiver processes. Each receiver process then computes the bit it has received and prints it to standard output. Use pipes for communication between processes.

3

THE DATA LINK LAYER

In this chapter we will study the design principles for the second layer in our model, the data link layer. This study deals with algorithms for achieving reliable, efficient communication of whole units of information called frames (rather than individual bits, as in the physical layer) between two adjacent machines. By adjacent, we mean that the two machines are connected by a communication channel that acts conceptually like a wire (e.g., a coaxial cable, telephone line, or wireless channel). The essential property of a channel that makes it "wire-like" is that the bits are delivered in exactly the same order in which they are sent.

At first you might think this problem is so trivial that there is nothing to study—machine *A* just puts the bits on the wire, and machine *B* just takes them off. Unfortunately, communication channels make errors occasionally. Furthermore, they have only a finite data rate, and there is a nonzero propagation delay between the time a bit is sent and the time it is received. These limitations have important implications for the efficiency of the data transfer. The protocols used for communications must take all these factors into consideration. These protocols are the subject of this chapter.

After an introduction to the key design issues present in the data link layer, we will start our study of its protocols by looking at the nature of errors and how they can be detected and corrected. Then we will study a series of increasingly complex protocols, each one solving more and more of the problems present in this layer. Finally, we will conclude with some examples of data link protocols.

3.1 DATA LINK LAYER DESIGN ISSUES

The data link layer uses the services of the physical layer to send and receive bits over communication channels. It has a number of functions, including:

1. Providing a well-defined service interface to the network layer.
2. Dealing with transmission errors.
3. Regulating the flow of data so that slow receivers are not swamped by fast senders.

To accomplish these goals, the data link layer takes the packets it gets from the network layer and encapsulates them into **frames** for transmission. Each frame contains a frame header, a payload field for holding the packet, and a frame trailer, as illustrated in Fig. 3-1. Frame management forms the heart of what the data link layer does. In the following sections we will examine all the above-mentioned issues in detail.

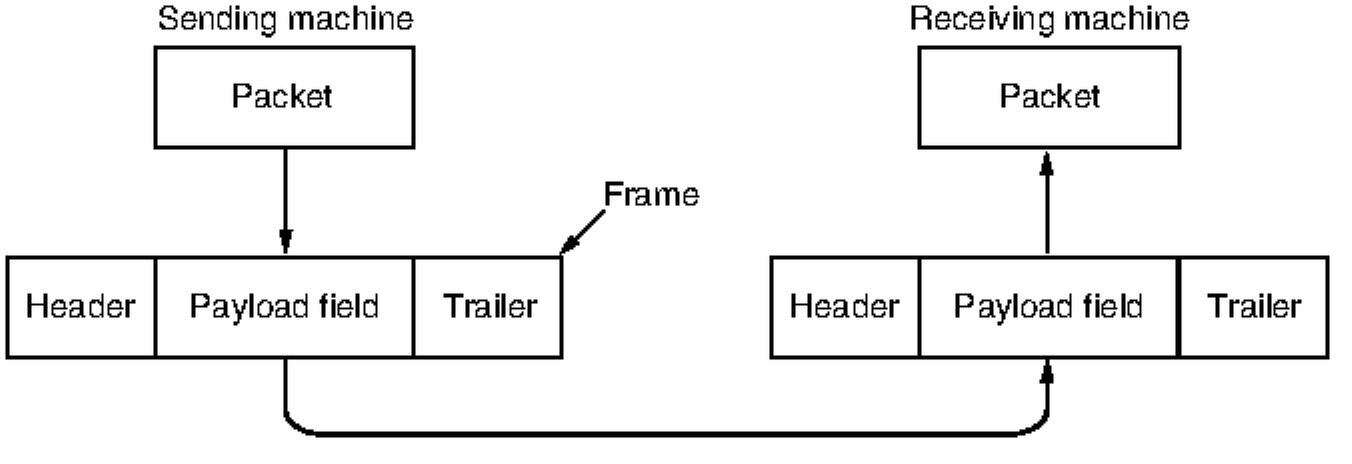


Figure 3-1. Relationship between packets and frames.

Although this chapter is explicitly about the data link layer and its protocols, many of the principles we will study here, such as error control and flow control, are found in transport and other protocols as well. That is because reliability is an overall goal, and it is achieved when all the layers work together. In fact, in many networks, these functions are found mostly in the upper layers, with the data link layer doing the minimal job that is "good enough." However, no matter where they are found, the principles are pretty much the same. They often show up in their simplest and purest forms in the data link layer, making this a good place to examine them in detail.

3.1.1 Services Provided to the Network Layer

The function of the data link layer is to provide services to the network layer. The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine. On the source machine is an entity, call it a process, in the network layer that hands some bits to the data link layer for transmission to the destination. The job of the data link layer is to transmit the bits to the destination machine so they can be handed over to the network layer there, as shown in Fig. 3-2(a). The actual transmission follows the path of Fig. 3-2(b), but it is easier to think in terms of two data link layer processes communicating using a data link protocol. For this reason, we will implicitly use the model of Fig. 3-2(a) throughout this chapter.

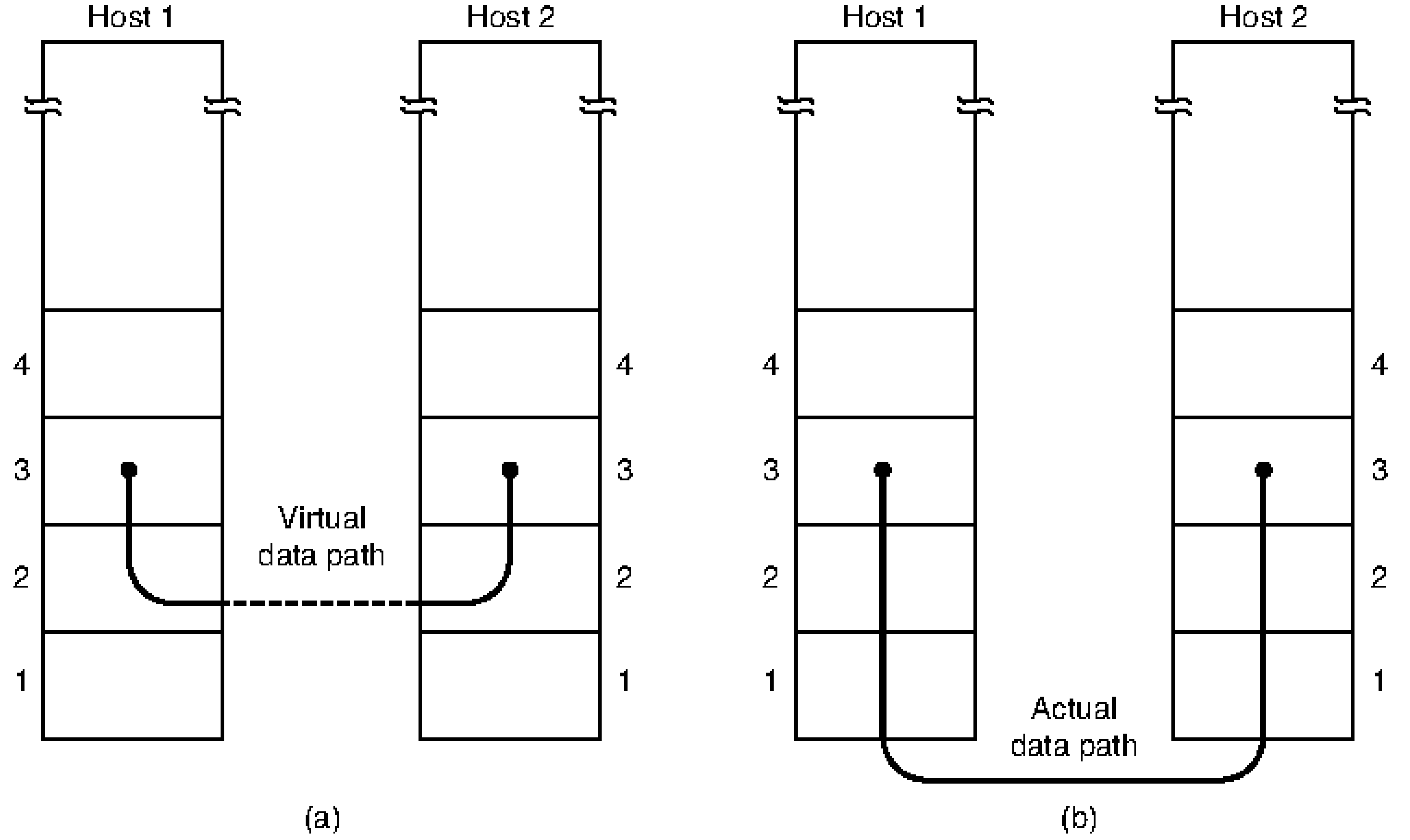


Figure 3-2. (a) Virtual communication. (b) Actual communication.

The data link layer can be designed to offer various services. The actual services that are offered vary from protocol to protocol. Three reasonable possibilities that we will consider in turn are:

1. Unacknowledged connectionless service.
2. Acknowledged connectionless service.
3. Acknowledged connection-oriented service.

Unacknowledged connectionless service consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them. Ethernet is a good example of a data link layer that provides this class of service. No logical connection is established beforehand or released afterward. If a frame is lost due to noise on the line, no attempt is made to detect the loss or recover from it in the data link layer. This class of service is appropriate when the error rate is very low, so recovery is left to higher layers. It is also appropriate for real-time traffic, such as voice, in which late data are worse than bad data.

The next step up in terms of reliability is acknowledged connectionless service. When this service is offered, there are still no logical connections used, but each frame sent is individually acknowledged. In this way, the sender knows whether a frame has arrived correctly or been lost. If it has not arrived within a specified time interval, it can be sent again. This service is useful over unreliable channels, such as wireless systems. 802.11 (WiFi) is a good example of this class of service.

It is perhaps worth emphasizing that providing acknowledgements in the data link layer is just an optimization, never a requirement. The network layer can always send a packet and wait for it to be acknowledged by its peer on the remote machine. If the acknowledgement is not forthcoming before the timer expires, the sender can just send the entire message again. The trouble with this strategy is that it can be inefficient. Links usually have a strict maximum frame length imposed by the hardware, and known propagation delays. The network layer does not know these parameters. It might send a large packet that is broken up into, say, 10 frames, of which 2 are lost on average. It would then take a very long time for the packet to get through. Instead, if individual frames are acknowledged and retransmitted, then errors can be corrected more directly and more quickly. On reliable channels, such as fiber, the overhead of a heavyweight data link protocol may be unnecessary, but on (inherently unreliable) wireless channels it is well worth the cost.

Getting back to our services, the most sophisticated service the data link layer can provide to the network layer is connection-oriented service. With this service, the source and destination machines establish a connection before any data are transferred. Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received. Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order. Connection-oriented service thus provides the network layer processes with the equivalent of a reliable bit stream. It is appropriate over long, unreliable links such as a satellite channel or a long-distance telephone circuit. If acknowledged connectionless service were used, it is conceivable that lost acknowledgements could cause a frame to be sent and received several times, wasting bandwidth.

When connection-oriented service is used, transfers go through three distinct phases. In the first phase, the connection is established by having both sides initialize variables and counters needed to keep track of which frames have been received and which ones have not. In the second phase, one or more frames are actually transmitted. In the third and final phase, the connection is released, freeing up the variables, buffers, and other resources used to maintain the connection.

3.1.2 Framing

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. If the channel is noisy, as it is for most wireless and some wired links, the physical layer will add some redundancy to its signals to reduce the bit error rate to a tolerable level. However, the bit stream received by the data link layer is not guaranteed to be error free. Some bits may have different values and the number of bits received may be less than, equal to, or more than the number of bits transmitted. It is up to the data link layer to detect and, if necessary, correct errors.

The usual approach is for the data link layer to break up the bit stream into discrete frames, compute a short token called a checksum for each frame, and include the checksum in the frame when it is transmitted. (Checksum algorithms will be discussed later in this chapter.) When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report).

Breaking up the bit stream into frames is more difficult than it at first appears. A good design must make it easy for a receiver to find the start of new frames while using little of the channel bandwidth. We will look at four methods:

1. Byte count.
2. Flag bytes with byte stuffing.
3. Flag bits with bit stuffing.
4. Physical layer coding violations.

The first framing method uses a field in the header to specify the number of bytes in the frame. When the data link layer at the destination sees the byte count, it knows how many bytes follow and hence where the end of the frame is. This technique is shown in Fig. 3-3(a) for four small example frames of sizes 5, 5, 8, and 8 bytes, respectively.

The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the byte count of 5 in the second frame of Fig. 3-3(b) becomes a 7 due to a single bit flip, the destination will get out of synchronization. It will then be unable to locate the correct start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many bytes to skip over to get to the start of the retransmission. For this reason, the byte count method is rarely used by itself.

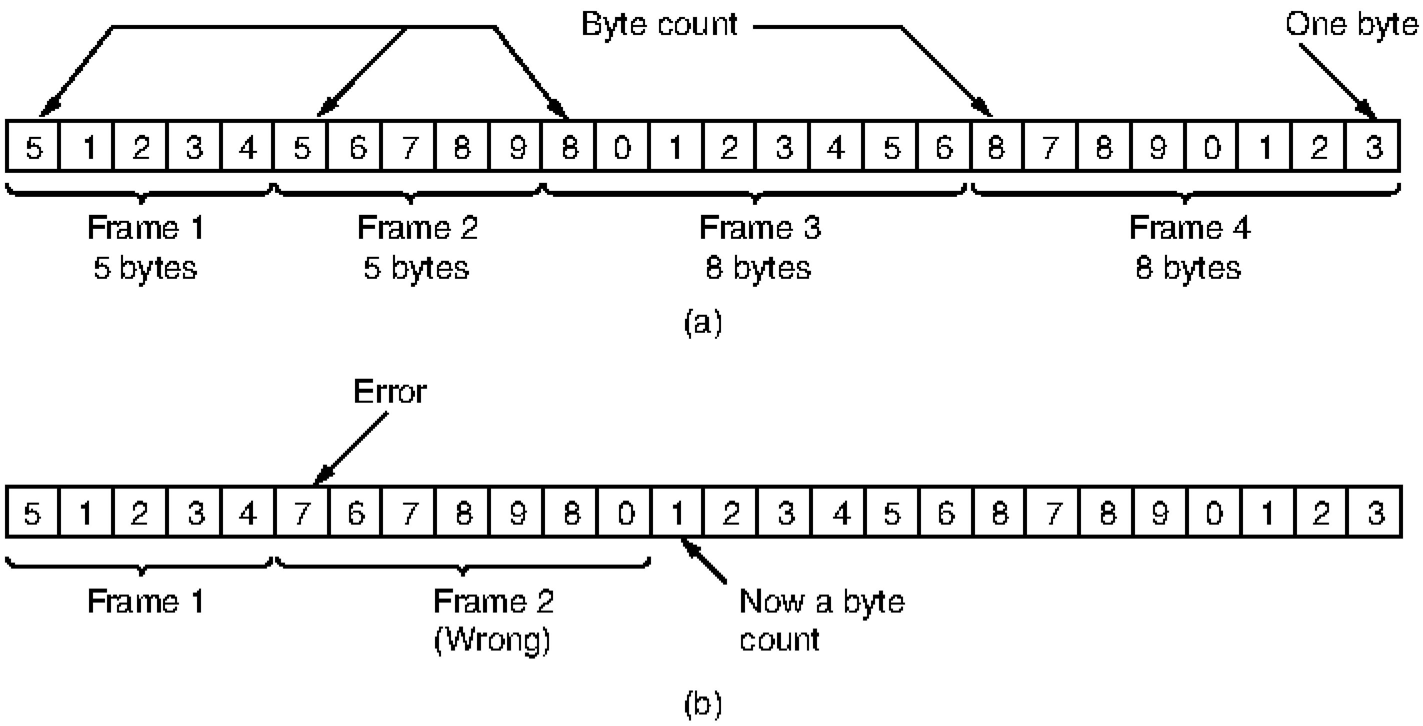


Figure 3-3. A byte stream. (a) Without errors. (b) With one error.

The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. Often the same byte, called a **flag byte**, is used as both the starting and ending delimiter. This byte is shown in Fig. 3-4(a) as FLAG. Two consecutive flag bytes indicate the end of one frame and the start of the next. Thus, if the receiver ever loses synchronization it can just search for two flag bytes to find the end of the current frame and the start of the next frame.

However, there is a still a problem we have to solve. It may happen that the flag byte occurs in the data, especially when binary data such as photographs or songs are being transmitted. This situation would interfere with the framing. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data. Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it. The data link layer on the receiving end removes the escape bytes before giving the data to the network layer. This technique is called **byte stuffing**.

Of course, the next question is: what happens if an escape byte occurs in the middle of the data? The answer is that it, too, is stuffed with an escape byte. At the receiver, the first escape byte is removed, leaving the data byte that follows it (which might be another escape byte or the flag byte). Some examples are shown in Fig. 3-4(b). In all cases, the byte sequence delivered after destuffing is exactly the same as the original byte sequence. We can still search for a frame boundary by looking for two flag bytes in a row, without bothering to undo escapes.

The byte-stuffing scheme depicted in Fig. 3-4 is a slight simplification of the one used in **PPP** (**Point-to-Point Protocol**), which is used to carry packets over communications links. We will discuss PPP near the end of this chapter.

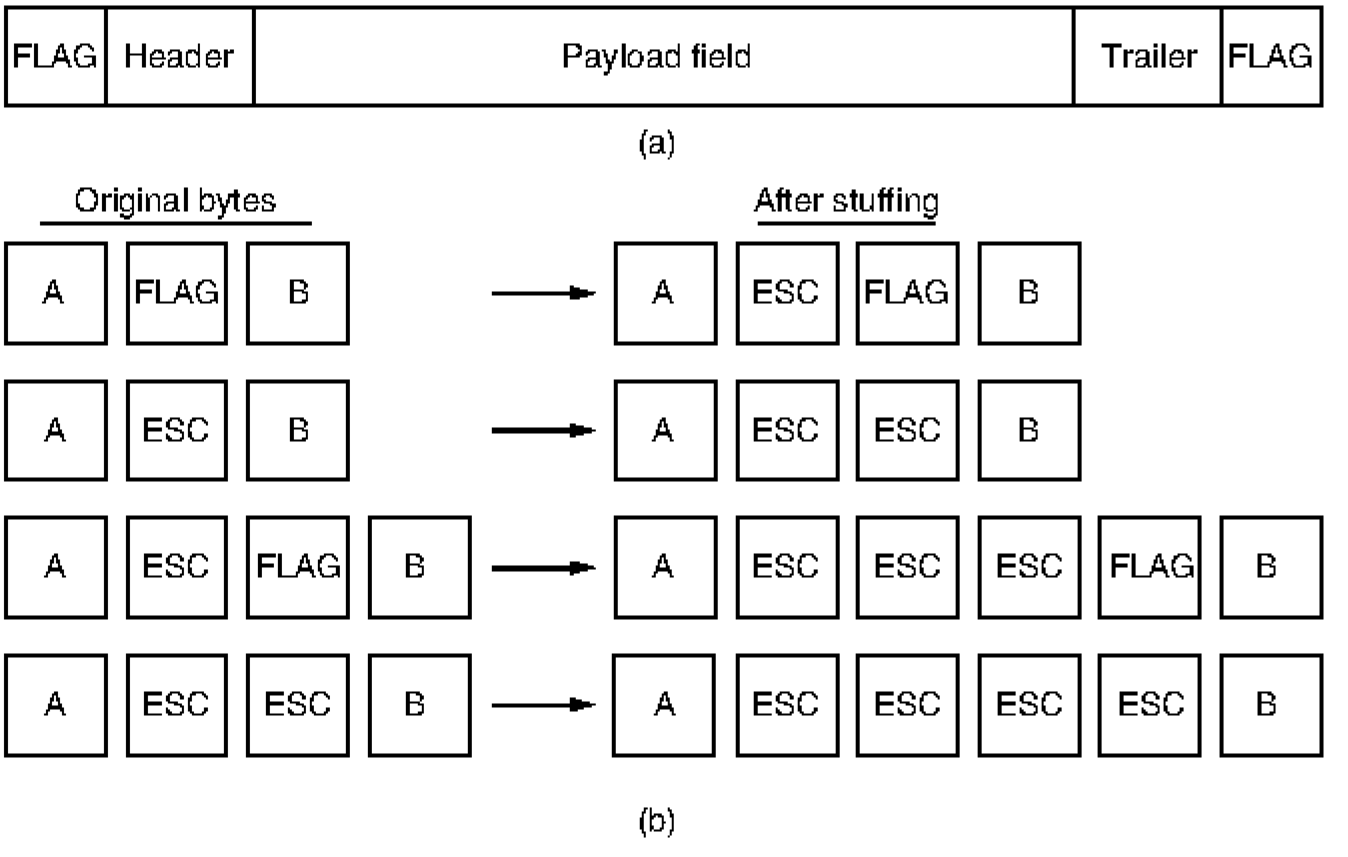


Figure 3-4. (a) A frame delimited by flag bytes. (b) Four examples of byte sequences before and after byte stuffing.

The third method of delimiting the bit stream gets around a disadvantage of byte stuffing, which is that it is tied to the use of 8-bit bytes. Framing can be also be done at the bit level, so frames can contain an arbitrary number of bits made up of units of any size. It was developed for the once very popular **HDLC** (**High-level Data Link Control**) protocol. Each frame begins and ends with a special bit pattern, 01111110 or 0x7E in hexadecimal. This pattern is a flag byte. Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This **bit stuffing** is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data. It also ensures a minimum density of transitions that help the physical layer maintain synchronization. USB (Universal Serial Bus) uses bit stuffing for this reason.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 3-5 gives an example of bit stuffing.

With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Figure 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

With both bit and byte stuffing, a side effect is that the length of a frame now depends on the contents of the data it carries. For instance, if there are no flag bytes in the data, 100 bytes might be carried in a frame of roughly 100 bytes. If, however, the data consists solely of flag bytes, each flag byte will be escaped and the frame will become roughly 200 bytes long. With bit stuffing, the increase would be roughly 12.5% as 1 bit is added to every byte.

The last method of framing is to use a shortcut from the physical layer. We saw in Chap. 2 that the encoding of bits as signals often includes redundancy to help the receiver. This redundancy means that some signals will not occur in regular data. For example, in the 4B/5B line code 4 data bits are mapped to 5 signal bits to ensure sufficient bit transitions. This means that 16 out of the 32 signal possibilities are not used. We can use some reserved signals to indicate the start and end of frames. In effect, we are using "coding violations" to delimit frames. The beauty of this scheme is that, because they are reserved signals, it is easy to find the start and end of frames and there is no need to stuff the data.

Many data link protocols use a combination of these methods for safety. A common pattern used for Ethernet and 802.11 is to have a frame begin with a well-defined pattern called a **preamble**. This pattern might be quite long (72 bits is typical for 802.11) to allow the receiver to prepare for an incoming packet. The preamble is then followed by a length (i.e., count) field in the header that is used to locate the end of the frame.

3.1.3 Error Control

Having solved the problem of marking the start and end of each frame, we come to the next problem: how to make sure all frames are eventually delivered to the network layer at the destination and in the proper order. Assume for the moment that the receiver can tell whether a frame that it receives contains correct or faulty information (we will look at the codes that are used to detect and correct transmission errors in Sec. 3.2). For unacknowledged connectionless service it might be fine if the sender just kept outputting frames without regard to whether they were arriving properly. But for reliable, connection-oriented service it would not be fine at all.

The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line. Typically, the protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames. If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely. On the other hand, a negative acknowledgement means that something has gone wrong and the frame must be transmitted again.

An additional complication comes from the possibility that hardware troubles may cause a frame to vanish completely (e.g., in a noise burst). In this case, the receiver will not react at all, since it has no reason to react. Similarly, if the acknowledgement frame is lost, the sender will not know how to proceed. It should be clear that a protocol in which the sender transmits a frame and then waits for an acknowledgement, positive or negative, will hang forever if a frame is ever lost due to, for example, malfunctioning hardware or a faulty communication channel.

This possibility is dealt with by introducing timers into the data link layer. When the sender transmits a frame, it generally also starts a timer. The timer is set to expire after an interval long enough for the frame to reach the destination, be processed there, and have the acknowledgement propagate back to the sender. Normally, the frame will be correctly received and the acknowledgement will get back before the timer runs out, in which case the timer will be canceled.

However, if either the frame or the acknowledgement is lost, the timer will go off, alerting the sender to a potential problem. The obvious solution is to just transmit the frame again. However, when frames may be transmitted multiple times there is a danger that the receiver will accept the same frame two or more times and pass it to the network layer more than once. To prevent this from happening, it is generally necessary to assign sequence numbers to outgoing frames, so that the receiver can distinguish retransmissions from originals.

The whole issue of managing the timers and sequence numbers so as to ensure that each frame is ultimately passed to the network layer at the destination exactly once, no more and no less, is an important part of the duties of the data link layer (and higher layers). Later in this chapter, we will look at a series of increasingly sophisticated examples to see how this management is done.

3.1.4 Flow Control

Another important design issue that occurs in the data link layer (and higher layers as well) is what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them. This situation can occur when the sender is running on a fast, powerful computer and the receiver is running on a slow, low-end machine. A common situation is when a smart phone requests a Web page from a far more powerful server, which then turns on the fire hose and blasts the data at the poor helpless phone until it is completely swamped. Even if the transmission is error free, the receiver may be unable to handle the frames as fast as they arrive and will lose some.

Clearly, something has to be done to prevent this situation. Two approaches are commonly used. In the first one, **feedback-based flow control**, the receiver sends back information to the sender giving it permission to send more data, or at least telling the sender how the receiver is doing. In the second one, **rate-based flow control**, the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver.

In this chapter we will study feedback-based flow control schemes, primarily because rate-based schemes are only seen as part of the transport layer (Chap. 5). Feedback-based schemes are seen at both the link layer and higher layers. The latter is more common these days, in which case the link layer hardware is designed to run fast enough that it does not cause loss. For example, hardware implementations of the link layer as **NICs** (**Network Interface Cards**) are sometimes said to run at "wire speed," meaning that they can handle frames as fast as they can arrive on the link. Any overruns are then not a link problem, so they are handled by higher layers.

Various feedback-based flow control schemes are known, but most of them use the same basic principle. The protocol contains well-defined rules about when a sender may transmit the next frame. These rules often prohibit frames from being sent until the receiver has granted permission, either implicitly or explicitly. For example, when a connection is set up the receiver might say: "You may send me *n* frames now, but after they have been sent, do not send any more until I have told you to continue." We will examine the details shortly.

3.2 ERROR DETECTION AND CORRECTION

We saw in Chap. 2 that communication channels have a range of characteristics. Some channels, like optical fiber in telecommunications networks, have tiny error rates so that transmission errors are a rare occurrence. But other channels, especially wireless links and aging local loops, have error rates that are orders of magnitude larger. For these links, transmission errors are the norm. They cannot be avoided at a reasonable expense or cost in terms of performance. The conclusion is that transmission errors are here to stay. We have to learn how to deal with them.

Network designers have developed two basic strategies for dealing with errors. Both add redundant information to the data that is sent. One strategy is to include enough redundant information to enable the receiver to deduce what the transmitted data must have been. The other is to include only enough redundancy to allow the receiver to deduce that an error has occurred (but not which error) and have it request a retransmission. The former strategy uses **error-correcting codes** and the latter uses **error-detecting codes**. The use of error-correcting codes is often referred to as **FEC** (**Forward Error Correction**).

Each of these techniques occupies a different ecological niche. On channels that are highly reliable, such as fiber, it is cheaper to use an error-detecting code and just retransmit the occasional block found to be faulty. However, on channels such as wireless links that make many errors, it is better to add redundancy to each block so that the receiver is able to figure out what the originally transmitted block was. FEC is used on noisy channels because retransmissions are just as likely to be in error as the first transmission.

A key consideration for these codes is the type of errors that are likely to occur. Neither error-correcting codes nor error-detecting codes can handle all possible errors since the redundant bits that offer protection are as likely to be received in error as the data bits (which can compromise their protection). It would be nice if the channel treated redundant bits differently than data bits, but it does not. They are all just bits to the channel. This means that to avoid undetected errors the code must be strong enough to handle the expected errors.

One model is that errors are caused by extreme values of thermal noise that overwhelm the signal briefly and occasionally, giving rise to isolated single-bit errors. Another model is that errors tend to come in bursts rather than singly. This model follows from the physical processes that generate them—such as a deep fade on a wireless channel or transient electrical interference on a wired channel/

Both models matter in practice, and they have different trade-offs. Having the errors come in bursts has both advantages and disadvantages over isolated single-bit errors. On the advantage side, computer data are always sent in blocks of bits. Suppose that the block size was 1000 bits and the error rate was 0.001 per bit. If errors were independent, most blocks would contain an error. If the errors came in bursts of 100, however, only one block in 100 would be affected, on average. The disadvantage of burst errors is that when they do occur they are much harder to correct than isolated errors.

Other types of errors also exist. Sometimes, the location of an error will be known, perhaps because the physical layer received an analog signal that was far from the expected value for a 0 or 1 and declared the bit to be lost. This situation is called an **erasure channel**. It is easier to correct errors in erasure channels than in channels that flip bits because even if the value of the bit has been lost, at least we know which bit is in error. However, we often do not have the benefit of erasures.

We will examine both error-correcting codes and error-detecting codes next. Please keep two points in mind, though. First, we cover these codes in the link layer because this is the first place that we have run up against the problem of reliably transmitting groups of bits. However, the codes are widely used because reliability is an overall concern. Error-correcting codes are also seen in the physical layer, particularly for noisy channels, and in higher layers, particularly for real-time media and content distribution. Error-detecting codes are commonly used in link, network, and transport layers.

The second point to bear in mind is that error codes are applied mathematics. Unless you are particularly adept at Galois fields or the properties of sparse matrices, you should get codes with good properties from a reliable source rather than making up your own. In fact, this is what many protocol standards do, with the same codes coming up again and again. In the material below, we will study a simple code in detail and then briefly describe advanced codes. In this way, we can understand the trade-offs from the simple code and talk about the codes that are used in practice via the advanced codes.

3.2.1 Error-Correcting Codes

We will examine four different error-correcting codes:

1. Hamming codes.
2. Binary convolutional codes.
3. Reed-Solomon codes.
4. Low-Density Parity Check codes.

All of these codes add redundancy to the information that is sent. A frame consists of m data (i.e., message) bits and r redundant (i.e. check) bits. In a **block code**, the r check bits are computed solely as a function of the m data bits with which they are associated, as though the m bits were looked up in a large table to find their corresponding r check bits. In a **systematic code**, the m data bits are sent directly, along with the check bits, rather than being encoded themselves before they are sent. In a **linear code**, the r check bits are computed as a linear function of the m data bits. Exclusive OR (XOR) or modulo 2 addition is a popular choice. This means that encoding can be done with operations such as matrix multiplications or simple logic circuits. The codes we will look at in this section are linear, systematic block codes unless otherwise noted.

Let the total length of a block be n (i.e., $n = m + r$). We will describe this as an (n,m) code. An n-bit unit containing data and check bits is referred to as an n-bit **codeword**. The **code rate**, or simply rate, is the fraction of the codeword that carries information that is not redundant, or m/n. The rates used in practice vary widely. They might be 1/2 for a noisy channel, in which case half of the received information is redundant, or close to 1 for a high-quality channel, with only a small number of check bits added to a large message.

To understand how errors can be handled, it is necessary to first look closely at what an error really is. Given any two codewords that may be transmitted or received—say, 10001001 and 10110001—it is possible to determine how many corresponding bits differ. In this case, 3 bits differ. To determine how many bits differ, just XOR the two codewords and count the number of 1 bits in the result. For example:

$$\begin{array}{l} 10001001 \\ \underline{10110001} \\ 00111000 \end{array}$$

The number of bit positions in which two codewords differ is called the **Hamming distance** (Hamming, 1950). Its significance is that if two codewords are a Hamming distance d apart, it will require d single-bit errors to convert one into the other.

Given the algorithm for computing the check bits, it is possible to construct a complete list of the legal codewords, and from this list to find the two codewords with the smallest Hamming distance. This distance is the Hamming distance of the complete code.

In most data transmission applications, all 2^m possible data messages are legal, but due to the way the check bits are computed, not all of the 2^n possible codewords are used. In fact, when there are r check bits, only the small fraction of $2^m/2^n$ or $1/2^r$ of the possible messages will be legal codewords. It is the sparseness with which the message is embedded in the space of codewords that allows the receiver to detect and correct errors.

The error-detecting and error-correcting properties of a block code depend on its Hamming distance. To reliably detect d errors, you need a distance $d + 1$ code because with such a code there is no way that d single-bit errors can change a valid codeword into another valid codeword. When the receiver sees an illegal codeword, it can tell that a transmission error has occurred. Similarly, to correct d errors, you need a distance $2d + 1$ code because that way the legal codewords are so far apart that even with d changes the original codeword is still closer than any other codeword. This means the original codeword can be uniquely determined based on the assumption that a larger number of errors are less likely.

As a simple example of an error-correcting code, consider a code with only four valid codewords:

0000000000, 0000011111, 1111100000, and 1111111111

This code has a distance of 5, which means that it can correct double errors or detect quadruple errors. If the codeword 0000000111 arrives and we expect only single- or double-bit errors, the receiver will know that the original must have been 0000011111. If, however, a triple error changes 0000000000 into 0000000111, the error will not be corrected properly. Alternatively, if we expect all of these errors, we can detect them. None of the received codewords are legal codewords so an error must have occurred. It should be apparent that in this example we cannot both correct double errors and detect quadruple errors because this would require us to interpret a received codeword in two different ways.

In our example, the task of decoding by finding the legal codeword that is closest to the received codeword can be done by inspection. Unfortunately, in the most general case where all codewords need to be evaluated as candidates, this task can be a time-consuming search. Instead, practical codes are designed so that they admit shortcuts to find what was likely the original codeword.

Imagine that we want to design a code with m message bits and r check bits that will allow all single errors to be corrected. Each of the 2^m legal messages has n illegal codewords at a distance of 1 from it. These are formed by systematically inverting each of the n bits in the n-bit codeword formed from it. Thus, each of the 2^m legal messages requires $n + 1$ bit patterns dedicated to it. Since the total number of bit patterns is 2^n, we must have $(n + 1)2^m \leq 2^n$. Using $n = m + r$, this requirement becomes

$$(m + r + 1) \leq 2^r \tag{3-1}$$

Given m, this puts a lower limit on the number of check bits needed to correct single errors.

This theoretical lower limit can, in fact, be achieved using a method due to Hamming (1950). In **Hamming codes** the bits of the codeword are numbered consecutively, starting with bit 1 at the left end, bit 2 to its immediate right, and so on. The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits. The rest (3, 5, 6, 7, 9, etc.) are filled up with the m data bits. This pattern is shown for an (11,7) Hamming code with 7 data bits and 4 check bits in Fig. 3-6. Each check bit forces the modulo 2 sum, or parity, of some collection of bits, including itself, to be even (or odd). A bit may be included in several check bit computations. To see which check bits the data bit in position k contributes to, rewrite k as a sum of powers of 2. For example, 11 = 1 + 2 + 8 and 29 = 1 + 4 + 8 + 16. A bit is checked by just those check bits occurring in its expansion (e.g., bit 11 is checked by bits 1, 2, and 8). In the example, the check bits are computed for even parity sums for a message that is the ASCII letter "A."

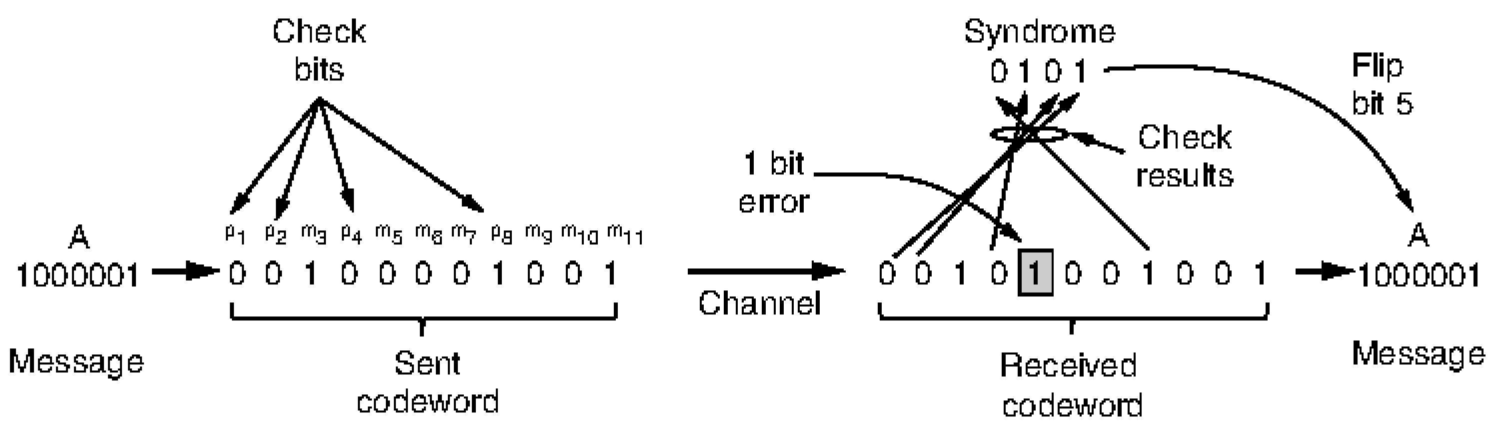


Figure 3-6. Example of an (11, 7) Hamming code correcting a single-bit error.

This construction gives a code with a Hamming distance of 3, which means that it can correct single errors (or detect double errors). The reason for the very careful numbering of message and check bits becomes apparent in the decoding process. When a codeword arrives, the receiver redoes the check bit computations including the values of the received check bits. We call these the check results. If the check bits are correct then, for even parity sums, each check result should be zero. In this case the codeword is accepted as valid.

If the check results are not all zero, however, an error has been detected. The set of check results forms the **error syndrome** that is used to pinpoint and correct the error. In Fig. 3-6, a single-bit error occurred on the channel so the check results are 0, 1, 0, and 1 for k = 8, 4, 2, and 1, respectively. This gives a syndrome of 0101 or 4 + 1=5. By the design of the scheme, this means that the fifth bit is in error. Flipping the incorrect bit (which might be a check bit or a data bit) and discarding the check bits gives the correct message of an ASCII "A."

Hamming distances are valuable for understanding block codes, and Hamming codes are used in error-correcting memory. However, most networks use stronger codes. The second code we will look at is a **convolutional code**. This code is the only one we will cover that is not a block code. In a convolutional code, an encoder processes a sequence of input bits and generates a sequence of output bits. There is no natural message size or encoding boundary as in a block code. The output depends on the current and previous input bits. That is, the encoder has memory. The number of previous bits on which the output depends is called the **constraint length** of the code. Convolutional codes are specified in terms of their rate and constraint length.

Convolutional codes are widely used in deployed networks, for example, as part of the GSM mobile phone system, in satellite communications, and in 802.11. As an example, a popular convolutional code is shown in Fig. 3-7. This code is known as the NASA convolutional code of r = 1/2 and k = 7, since it was first used for the Voyager space missions starting in 1977. Since then it has been liberally reused, for example, as part of 802.11.

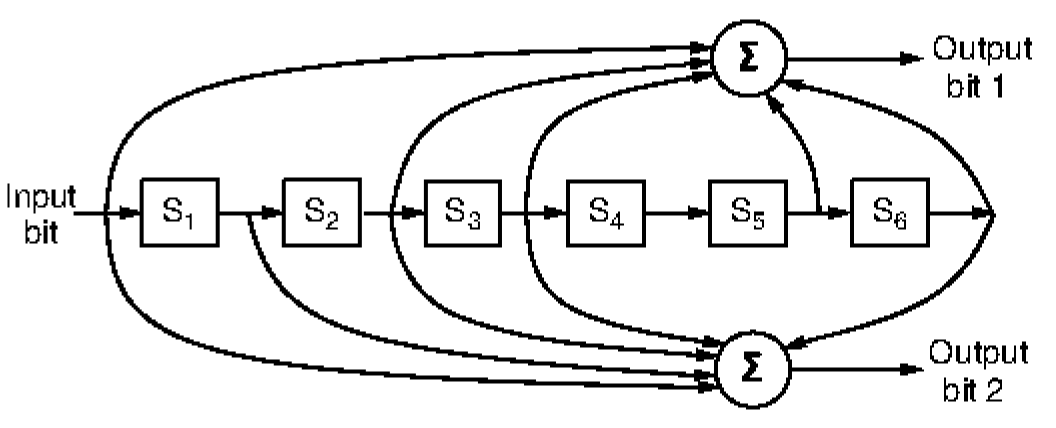


Figure 3-7. The NASA binary convolutional code used in 802.11.

In Fig. 3-7, each input bit on the left-hand side produces two output bits on the right-hand side that are XOR sums of the input and internal state. Since it deals with bits and performs linear operations, this is a binary, linear convolutional code. Since 1 input bit produces 2 output bits, the code rate is 1/2. It is not systematic since none of the output bits is simply the input bit.

The internal state is kept in six memory registers. Each time another bit is input the values in the registers are shifted to the right. For example, if 111 is input and the initial state is all zeros, the internal state, written left to right, will become 100000, 110000, and 111000 after the first, second, and third bits have been input. The output bits will be 11, followed by 10, and then 01. It takes seven shifts to flush an input completely so that it does not affect the output. The constraint length of this code is thus $k = 7$.

A convolutional code is decoded by finding the sequence of input bits that is most likely to have produced the observed sequence of output bits (which includes any errors). For small values of k, this is done with a widely used algorithm developed by Viterbi (Forney, 1973). The algorithm walks the observed sequence, keeping for each step and for each possible internal state the input sequence that would have produced the observed sequence with the fewest errors. The input sequence requiring the fewest errors at the end is the most likely message.

Convolutional codes have been popular in practice because it is easy to factor the uncertainty of a bit being a 0 or a 1 into the decoding. For example, suppose −1V is the logical 0 level and +1V is the logical 1 level, we might receive 0.9V and −0.1V for 2 bits. Instead of mapping these signals to 1 and 0 right away, we would like to treat 0.9V as "very likely a 1" and −0.1V as "maybe a 0" and correct the sequence as a whole. Extensions of the Viterbi algorithm can work with these uncertainties to provide stronger error correction. This approach of working with the uncertainty of a bit is called **soft-decision decoding**. Conversely, deciding whether each bit is a 0 or a 1 before subsequent error correction is called **hard-decision decoding**.

The third kind of error-correcting code we will describe is the **Reed-Solomon code**. Like Hamming codes, Reed-Solomon codes are linear block codes, and they are often systematic too. Unlike Hamming codes, which operate on individual bits, Reed-Solomon codes operate on m bit symbols. Naturally, the mathematics are more involved, so we will describe their operation by analogy.

Reed-Solomon codes are based on the fact that every n degree polynomial is uniquely determined by $n + 1$ points. For example, a line having the form $ax + b$ is determined by two points. Extra points on the same line are redundant, which is helpful for error correction. Imagine that we have two data points that represent a line and we send those two data points plus two check points chosen to lie on the same line. If one of the points is received in error, we can still recover the data points by fitting a line to the received points. Three of the points will lie on the line, and one point, the one in error, will not. By finding the line we have corrected the error.

Reed-Solomon codes are actually defined as polynomials that operate over finite fields, but they work in a similar manner. For m bit symbols, the codewords are $2^m - 1$ symbols long. A popular choice is to make $m = 8$ so that symbols are bytes. A codeword is then 255 bytes long. The (255, 233) code is widely used; it adds 32 redundant symbols to 233 data symbols. Decoding with error correction is done with an algorithm developed by Berlekamp and Massey that can efficiently perform the fitting task for moderate-length codes (Massey, 1969).

Reed-Solomon codes are widely used in practice because of their strong error-correction properties, particularly for burst errors. They are used for DSL, data over cable, satellite communications, and perhaps most ubiquitously on CDs, DVDs, and Blu-ray discs. Because they are based on m bit symbols, a single-bit error and an m-bit burst error are both treated simply as one symbol error. When $2t$ redundant symbols are added, a Reed-Solomon code is able to correct up to t errors in any of the transmitted symbols. This means, for example, that the (255, 233) code, which has 32 redundant symbols, can correct up to 16 symbol errors. Since the symbols may be consecutive and they are each 8 bits, an error burst of up to 128 bits can be corrected. The situation is even better if the error model is one of erasures (e.g., a scratch on a CD that obliterates some symbols). In this case, up to $2t$ errors can be corrected.

Reed-Solomon codes are often used in combination with other codes such as a convolutional code. The thinking is as follows. Convolutional codes are effective at handling isolated bit errors, but they will fail, likely with a burst of errors, if there are too many errors in the received bit stream. By adding a Reed-Solomon code within the convolutional code, the Reed-Solomon decoding can mop up the error bursts, a task at which it is very good. The overall code then provides good protection against both single and burst errors.

The final error-correcting code we will cover is the **LDPC** (**Low-Density Parity Check**) code. LDPC codes are linear block codes that were invented by Robert Gallagher in his doctoral thesis (Gallagher, 1962). Like most theses, they were promptly forgotten, only to be reinvented in 1995 when advances in computing power had made them practical.

In an LDPC code, each output bit is formed from only a fraction of the input bits. This leads to a matrix representation of the code that has a low density of 1s, hence the name for the code. The received codewords are decoded with an approximation algorithm that iteratively improves on a best fit of the received data to a legal codeword. This corrects errors.

LDPC codes are practical for large block sizes and have excellent error-correction abilities that outperform many other codes (including the ones we have looked at) in practice. For this reason they are rapidly being included in new protocols. They are part of the standard for digital video broadcasting, 10 Gbps Ethernet, power-line networks, and the latest version of 802.11. Expect to see more of them in future networks.

3.2.2 Error-Detecting Codes

Error-correcting codes are widely used on wireless links, which are notoriously noisy and error prone when compared to optical fibers. Without error-correcting codes, it would be hard to get anything through. However, over fiber or high-quality copper, the error rate is much lower, so error detection and retransmission is usually more efficient there for dealing with the occasional error.

We will examine three different error-detecting codes. They are all linear, systematic block codes:

1. Parity.
2. Checksums.
3. Cyclic Redundancy Checks (CRCs).

To see how they can be more efficient than error-correcting codes, consider the first error-detecting code, in which a single **parity bit** is appended to the data. The parity bit is chosen so that the number of 1 bits in the codeword is even (or odd). Doing this is equivalent to computing the (even) parity bit as the modulo 2 sum or XOR of the data bits. For example, when 1011010 is sent in even parity, a bit is added to the end to make it 10110100. With odd parity 1011010 becomes 10110101. A code with a single parity bit has a distance of 2, since any single-bit error produces a codeword with the wrong parity. This means that it can detect single-bit errors.

Consider a channel on which errors are isolated and the error rate is 10^{-6} per bit. This may seem a tiny error rate, but it is at best a fair rate for a long wired cable that is challenging for error detection. Typical LAN links provide bit error rates of $10^{-10.}$ Let the block size be 1000 bits. To provide error correction for 1000-bit blocks, we know from Eq. (3-1) that 10 check bits are needed. Thus, a megabit of data would require 10,000 check bits. To merely detect a block with a single 1-bit error, one parity bit per block will suffice. Once every 1000 blocks, a block will be found to be in error and an extra block (1001 bits) will have to be transmitted to repair the error. The total overhead for the error detection and retransmission method is only 2001 bits per megabit of data, versus 10,000 bits for a Hamming code.

One difficulty with this scheme is that a single parity bit can only reliably detect a single-bit error in the block. If the block is badly garbled by a long burst error, the probability that the error will be detected is only 0.5, which is hardly acceptable. The odds can be improved considerably if each block to be sent is regarded as a rectangular matrix n bits wide and k bits high. Now, if we compute and send one parity bit for each row, up to k bit errors will be reliably detected as long as there is at most one error per row.

However, there is something else we can do that provides better protection against burst errors: we can compute the parity bits over the data in a different order than the order in which the data bits are transmitted. Doing so is called **interleaving**. In this case, we will compute a parity bit for each of the n columns and send all the data bits as k rows, sending the rows from top to bottom and the bits in each row from left to right in the usual manner. At the last row, we send the n parity bits. This transmission order is shown in Fig. 3-8 for $n = 7$ and $k = 7$.

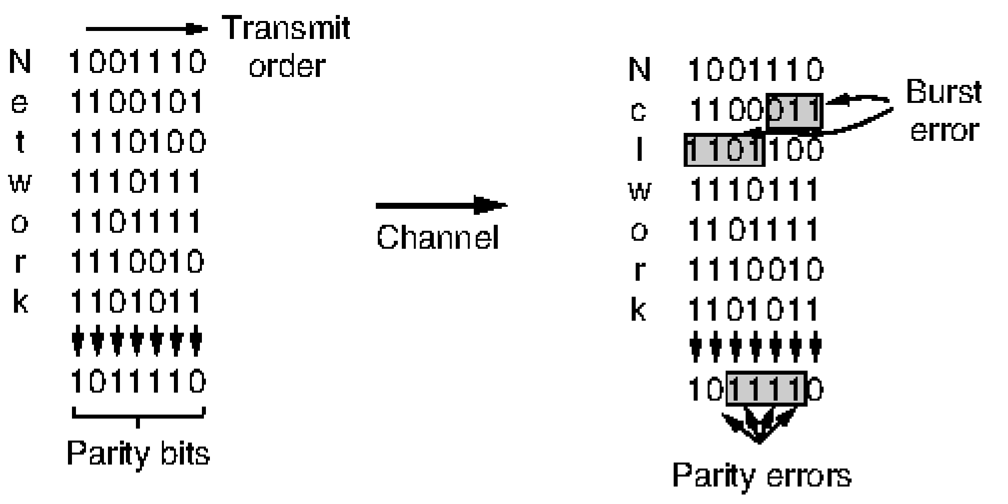


Figure 3-8. Interleaving of parity bits to detect a burst error.

Interleaving is a general technique to convert a code that detects (or corrects) isolated errors into a code that detects (or corrects) burst errors. In Fig. 3-8, when a burst error of length $n = 7$ occurs, the bits that are in error are spread across different columns. (A burst error does not imply that all the bits are wrong; it just implies that at least the first and last are wrong. In Fig. 3-8, 4 bits were flipped over a range of 7 bits.) At most 1 bit in each of the n columns will be affected, so the parity bits on those columns will detect the error. This method uses n parity bits on blocks of kn data bits to detect a single burst error of length n or less.

A burst of length $n + 1$ will pass undetected, however, if the first bit is inverted, the last bit is inverted, and all the other bits are correct. If the block is badly garbled by a long burst or by multiple shorter bursts, the probability that any of the n columns will have the correct parity by accident is 0.5, so the probability of a bad block being accepted when it should not be is 2^{-n}.

The second kind of error-detecting code, the **checksum**, is closely related to groups of parity bits. The word "checksum" is often used to mean a group of check bits associated with a message, regardless of how are calculated. A group of parity bits is one example of a checksum. However, there are other, stronger checksums based on a running sum of the data bits of the message. The checksum is usually placed at the end of the message, as the complement of the sum function. This way, errors may be detected by summing the entire received codeword, both data bits and checksum. If the result comes out to be zero, no error has been detected.

One example of a checksum is the 16-bit Internet checksum used on all Internet packets as part of the IP protocol (Braden et al., 1988). This checksum is a sum of the message bits divided into 16-bit words. Because this method operates on words rather than on bits, as in parity, errors that leave the parity unchanged can still alter the sum and be detected. For example, if the lowest order bit in two different words is flipped from a 0 to a 1, a parity check across these bits would fail to detect an error. However, two 1s will be added to the 16-bit checksum to produce a different result. The error can then be detected.

The Internet checksum is computed in one's complement arithmetic instead of as the modulo 2^{16} sum. In one's complement arithmetic, a negative number is the bitwise complement of its positive counterpart. Modern computers run two's complement arithmetic, in which a negative number is the one's complement plus one. On a two's complement computer, the one's complement sum is equivalent to taking the sum modulo 2^{16} and adding any overflow of the high order bits back into the low-order bits. This algorithm gives a more uniform coverage of the data by the checksum bits. Otherwise, two high-order bits can be added, overflow, and be lost without changing the sum. There is another benefit, too. One's complement has two representations of zero, all 0s and all 1s. This allows one value (e.g., all 0s) to indicate that there is no checksum, without the need for another field.

For decades, it has always been assumed that frames to be checksummed contain random bits. All analyses of checksum algorithms have been made under this assumption. Inspection of real data by Partridge et al. (1995) has shown this assumption to be quite wrong. As a consequence, undetected errors are in some cases much more common than had been previously thought.

The Internet checksum in particular is efficient and simple but provides weak protection in some cases precisely because it is a simple sum. It does not detect the deletion or addition of zero data, nor swapping parts of the message, and it provides weak protection against message splices in which parts of two packets are put together. These errors may seem very unlikely to occur by random processes, but they are just the sort of errors that can occur with buggy hardware.

A better choice is **Fletcher's checksum** (Fletcher, 1982). It includes a positional component, adding the product of the data and its position to the running sum. This provides stronger detection of changes in the position of data.

Although the two preceding schemes may sometimes be adequate at higher layers, in practice, a third and stronger kind of error-detecting code is in widespread use at the link layer: the **CRC** (**Cyclic Redundancy Check**), also known as a **polynomial code**. Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only. A k-bit frame is regarded as the coefficient list for a polynomial with k terms, ranging from x^{k-1} to x^0. Such a polynomial is said to be of degree $k - 1$. The high-order (leftmost) bit is the coefficient of x^{k-1}, the next bit is the coefficient of x^{k-2}, and so on. For example, 110001 has 6 bits and thus represents a six-term polynomial with coefficients 1, 1, 0, 0, 0, and 1: $1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0$.

Polynomial arithmetic is done modulo 2, according to the rules of algebraic field theory. It does not have carries for addition or borrows for subtraction. Both addition and subtraction are identical to exclusive OR. For example:

$$
\begin{array}{r}
10011011 \\
+\ 11001010 \\
\hline
01010001
\end{array}
\qquad
\begin{array}{r}
00110011 \\
+\ 11001101 \\
\hline
11111110
\end{array}
\qquad
\begin{array}{r}
11110000 \\
-\ 10100110 \\
\hline
01010110
\end{array}
\qquad
\begin{array}{r}
01010101 \\
-\ 10101111 \\
\hline
11111010
\end{array}
$$

Long division is carried out in exactly the same way as it is in binary except that the subtraction is again done modulo 2. A divisor is said "to go into" a dividend if the dividend has as many bits as the divisor.

When the polynomial code method is employed, the sender and receiver must agree upon a **generator polynomial**, $G(x)$, in advance. Both the high- and low-order bits of the generator must be 1. To compute the CRC for some frame with m bits corresponding to the polynomial $M(x)$, the frame must be longer than the generator polynomial. The idea is to append a CRC to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by $G(x)$. When the receiver gets the checksummed frame, it tries dividing it by $G(x)$. If there is a remainder, there has been a transmission error.

The algorithm for computing the CRC is as follows:

1. Let r be the degree of $G(x)$. Append r zero bits to the low-order end of the frame so it now contains $m + r$ bits and corresponds to the polynomial $x^r M(x)$.
2. Divide the bit string corresponding to $G(x)$ into the bit string corresponding to $x^r M(x)$, using modulo 2 division.
3. Subtract the remainder (which is always r or fewer bits) from the bit string corresponding to $x^r M(x)$ using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial $T(x)$.

Figure 3-9 illustrates the calculation for a frame 1101011111 using the generator $G(x) = x^4 + x + 1$.

It should be clear that $T(x)$ is divisible (modulo 2) by $G(x)$. In any division problem, if you diminish the dividend by the remainder, what is left over is divisible by the divisor. For example, in base 10, if you divide 210,278 by 10,941, the remainder is 2399. If you then subtract 2399 from 210,278, what is left over (207,879) is divisible by 10,941.

Now let us analyze the power of this method. What kinds of errors will be detected? Imagine that a transmission error occurs, so that instead of the bit string for $T(x)$ arriving, $T(x) + E(x)$ arrives. Each 1 bit in $E(x)$ corresponds to a bit that has been inverted. If there are k 1 bits in $E(x)$, k single-bit errors have occurred. A single burst error is characterized by an initial 1, a mixture of 0s and 1s, and a final 1, with all other bits being 0.

Upon receiving the checksummed frame, the receiver divides it by $G(x)$; that is, it computes $[T(x) + E(x)]/G(x)$. $T(x)/G(x)$ is 0, so the result of the computation is simply $E(x)/G(x)$. Those errors that happen to correspond to polynomials containing $G(x)$ as a factor will slip by; all other errors will be caught.

If there has been a single-bit error, $E(x) = x^i$, where i determines which bit is in error. If $G(x)$ contains two or more terms, it will never divide into $E(x)$, so all single-bit errors will be detected.

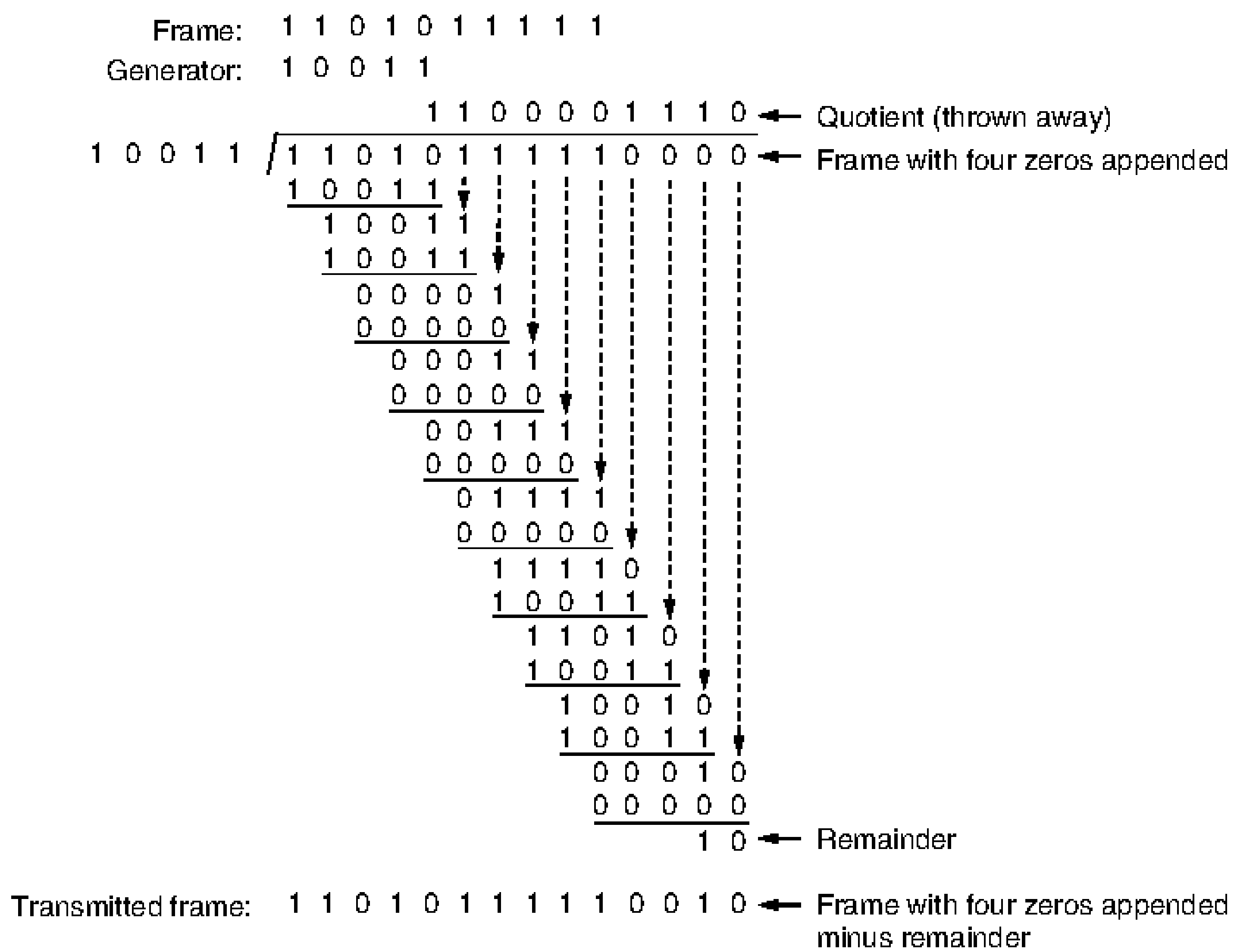


Figure 3-9. Example calculation of the CRC.

If there have been two isolated single-bit errors, $E(x) = x^i + x^j$, where $i > j$. Alternatively, this can be written as $E(x) = x^j(x^{i-j} + 1)$. If we assume that $G(x)$ is not divisible by x, a sufficient condition for all double errors to be detected is that $G(x)$ does not divide $x^k + 1$ for any k up to the maximum value of $i - j$ (i.e., up to the maximum frame length). Simple, low-degree polynomials that give protection to long frames are known. For example, $x^{15} + x^{14} + 1$ will not divide $x^k + 1$ for any value of k below 32,768.

If there are an odd number of bits in error, $E(X)$ contains an odd number of terms (e.g., $x^5 + x^2 + 1$, but not $x^2 + 1$). Interestingly, no polynomial with an odd number of terms has $x + 1$ as a factor in the modulo 2 system. By making $x + 1$ a factor of $G(x)$, we can catch all errors with an odd number of inverted bits.

Finally, and importantly, a polynomial code with r check bits will detect all burst errors of length $\leq r$. A burst error of length k can be represented by $x^i(x^{k-1} + \ldots + 1)$, where i determines how far from the right-hand end of the received frame the burst is located. If $G(x)$ contains an x^0 term, it will not have x^i as a factor, so if the degree of the parenthesized expression is less than the degree of $G(x)$, the remainder can never be zero.

If the burst length is $r + 1$, the remainder of the division by $G(x)$ will be zero if and only if the burst is identical to $G(x)$. By definition of a burst, the first and last bits must be 1, so whether it matches depends on the $r - 1$ intermediate bits. If all combinations are regarded as equally likely, the probability of such an incorrect frame being accepted as valid is $½^{r-1}$.

It can also be shown that when an error burst longer than $r + 1$ bits occurs or when several shorter bursts occur, the probability of a bad frame getting through unnoticed is $½^r$, assuming that all bit patterns are equally likely.

Certain polynomials have become international standards. The one used in IEEE 802 followed the example of Ethernet and is

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

Among other desirable properties, it has the property that it detects all bursts of length 32 or less and all bursts affecting an odd number of bits. It has been used widely since the 1980s. However, this does not mean it is the best choice. Using an exhaustive computational search, Castagnoli et al. (1993) and Koopman (2002) found the best CRCs. These CRCs have a Hamming distance of 6 for typical message sizes, while the IEEE standard CRC-32 has a Hamming distance of only 4.

Although the calculation required to compute the CRC may seem complicated, it is easy to compute and verify CRCs in hardware with simple shift register circuits (Peterson and Brown, 1961). In practice, this hardware is nearly always used. Dozens of networking standards include various CRCs, including virtually all LANs (e.g., Ethernet, 802.11) and point-to-point links (e.g., packets over SONET).

3.3 ELEMENTARY DATA LINK PROTOCOLS

To introduce the subject of protocols, we will begin by looking at three protocols of increasing complexity. For interested readers, a simulator for these and subsequent protocols is available via the Web (see the preface). Before we look at the protocols, it is useful to make explicit some of the assumptions underlying the model of communication.

To start with, we assume that the physical layer, data link layer, and network layer are independent processes that communicate by passing messages back and forth. A common implementation is shown in Fig. 3-10. The physical layer process and some of the data link layer process run on dedicate hardware called a **NIC** (**Network Interface Card**). The rest of the link layer process and the network layer process run on the main CPU as part of the operating system, with the software for the link layer process often taking the form of a **device driver**. However, other implementations are also possible (e.g., three processes offloaded to dedicated hardware called a **network accelerator**, or three processes running on the main CPU on a software-defined ratio). Actually, the preferred implementation changes from decade to decade with technology trade-offs. In any event, treating the three layers as separate processes makes the discussion conceptually cleaner and also serves to emphasize the independence of the layers.

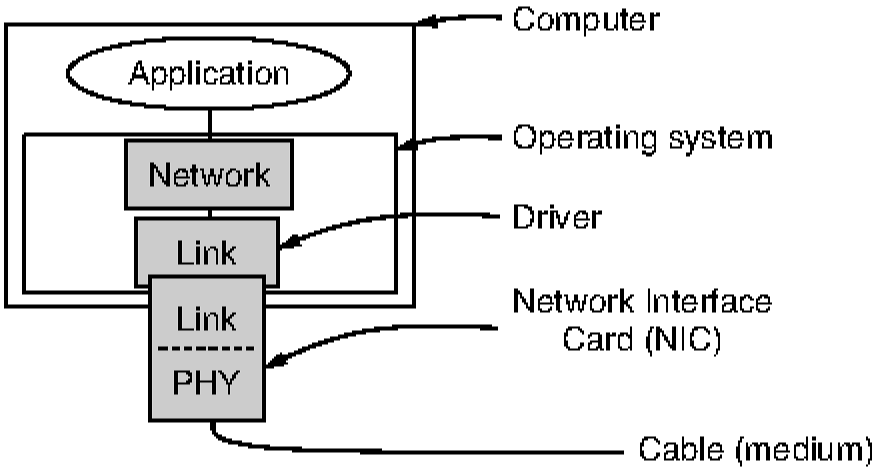


Figure 3-10. Implementation of the physical, data link, and network layers.

Another key assumption is that machine *A* wants to send a long stream of data to machine *B*, using a reliable, connection-oriented service. Later, we will consider the case where *B* also wants to send data to *A* simultaneously. *A* is assumed to have an infinite supply of data ready to send and never has to wait for data to be produced. Instead, when *A*'s data link layer asks for data, the network layer is always able to comply immediately. (This restriction, too, will be dropped later.)

We also assume that machines do not crash. That is, these protocols deal with communication errors, but not the problems caused by computers crashing and rebooting.

As far as the data link layer is concerned, the packet passed across the interface to it from the network layer is pure data, whose every bit is to be delivered to the destination's network layer. The fact that the destination's network layer may interpret part of the packet as a header is of no concern to the data link layer.

When the data link layer accepts a packet, it encapsulates the packet in a frame by adding a data link header and trailer to it (see Fig. 3-1). Thus, a frame consists of an embedded packet, some control information (in the header), and a checksum (in the trailer). The frame is then transmitted to the data link layer on the other machine. We will assume that there exist suitable library procedures *to_physical_layer* to send a frame and *from_physical_layer* to receive a frame. These procedures compute and append or check the checksum (which is usually done in hardware) so that we do not need to worry about it as part of the protocols we develop in this section. They might use the CRC algorithm discussed in the previous section, for example.

Initially, the receiver has nothing to do. It just sits around waiting for something to happen. In the example protocols throughout this chapter we will indicate that the data link layer is waiting for something to happen by the procedure call