Preface

This textbook represents more than a decade of work. During this time, some material became obsolete and had to be deleted. In my days as a telecommunication engineer and a university professor, much has changed in the fields of data communications and computer networks. Nonetheless, this text covers both the foundations and the latest advanced topics of computer networking.

The Internet is a revolutionary communication vehicle by which we all conveniently communicate every day and do business with one another. Because of its complexities at both hardware and software levels, the Internet is a challenge to those who want to study this field. The growing number and variety of communication services offer obvious challenges for computer network experts in designing cost-effective networks to meet the requirements of emerging communication systems. This book fills the gaps in current available texts.

Objectives

This textbook offers a mix of theory, architecture, and applications. The lack of computer communications books presenting moderate analysis with detailed drawing figures covering both wireline and wireless communication technologies led me to write this book. The main objective of this book is to help readers learn the fundamentals and certain advance concepts of computer and communication networks, using a unified set of symbols throughout a single textbook. The preparation of this book responds to the explosive demand for learning computer communication science and engineering.

This book targets two groups of people. For people in academia, at both the undergraduate and graduate levels, the book provides a thorough design and performance

evaluation of communication networks. The book can also give researchers the ability to analyze and simulate complex communication networks. For engineers who want to work in the communication and networking industry and need a reference covering various angles of computer networks, this book provides a variety of learning techniques: exercises, case studies, and computer simulation projects. The book makes it easy and fun for an engineer to review and learn from a reliable networking reference covering all the necessary concepts and performance models.

Organization of This Book

It would be impossible to cover all networking subjects in one textbook. The range of topics presented in this text, however, allows instructors to choose the topics best suited for their classes. Besides the explanations provided for each chapter, readers will learn how to model a communication network and how to mathematically analyze them. Readers of this text will benefit from the combination of theory and applications presented in each chapter, with the more theoretical portions of each chapter challenging those readers who are more ambitious. This book is organized into 20 chapters in two main parts as follows:

The ten chapters of Part I cover the fundamental topics in computer networking, with each chapter serving as a base for the following chapter. Part I of the book begins with an overview of networking, focusing on TCP/IP schemes, describing wireless networking, and ending with a discussion of the World Wide Web (WWW) and network security. Part I is most appropriate for readers with no experience in computer communications. The ten chapters in Part II cover detailed analytical aspects and a closer perspective of advanced networking protocols: switches, routers, multiplexers, delay and congestion analysis, multimedia networking, multicasting, data compression, voice over IP, optical networks, and sensor networks.

Chapter 1, Packet-Switched Networks, introduces computer networks, touching on the need for networks, explaining relevant packet-switched networks, and giving an overview of today's Internet. Fundamental concepts, such as *messages*, *packets*, and *frames* and *packet switching* versus *circuit switching*, are defined. Various types of packet-switched networks are defined, and how a message can be handled by either *connection-oriented networks* or *connectionless networks* is explained. Finally, this chapter presents a detailed analysis of packet size and optimizations.

Chapter 2, Foundation of Networking Protocols, presents the basics of the five-layer Internet Protocol reference model, as well as other protocols: the seven-layer OSI model and the *equal-size packet protocol* model.

Chapter 3, Networking Devices, introduces the overall architectures of networking devices, such as multiplexers, modems, and switching devices. *Multiplexers* are used in all layers of network. Networking modems are used for access to the Internet from remote and residential areas. Finally, switching devices, such as hubs, bridges, switches, and routers, are used to switch packets from one path to another.

Chapter 4, Data Links and Transmission, focuses on the links and transmission interfaces, the two basic components that networking starts with. This chapter presents both wired and wireless links and describes their characteristics, advantages, and channel access methods. This chapter also presents various *error-detection and correction* techniques at the link level and discusses the integrity of transmitted data. The chapter ends by presenting link-layer *stop-and-wait* and *sliding-window* flow control.

Chapter 5, Local Area Networks and Networks of LANs, explores the implementation of small networks, using the functional aspects of the fundamental knowledge gained in Chapters 2, 3, and Chapter 4 on basic protocols, devices, and links, respectively. The chapter provides some pointers for constructing a network with those devices and making connections, gives several examples of local area networks (LANs), and explains how such LANs are internetworked.

Chapter 6, Wireless Networks and Mobile IP, presents the basics of wireless networking. The chapter discusses challenges in designing a wireless network: *management of mobility*, *network reliability*, and *frequency reuse*. Next, the chapter presents an overview of wireless communication systems at all levels, from satellite to local-area networks and discusses wireless LANs and such standards as IEEE 802.11. The chapter then shifts to cellular networks, one of the main backbones of our wireless networking infrastructure. *Mobile IP* and *Wireless mesh networks* (WMNs), including WiFi and WiMAX technologies, are introduced at the end of this chapter.

Chapter 7, Routing and Internetworking, focuses on routing in wide area networks (WANs) and introduces related routing algorithms and protocols. Our networking infrastructure is clearly classified into those networks that use optimal routes and those that use nonoptimal routes. These two classes of algorithms are described in detail. Routing protocols are also classified as those that are applied within a domain and those that are applied beyond a domain. This chapter also presents congestion-control algorithms: *network-congestion control* and *link-flow control*. The chapter also looks at *random early detection* for congestion control and describes a useful technique to estimate the link-blocking probability.

Chapter 8, Transport and End-to-End Protocols, first looks at the basics of the *transport layer* and demonstrates how a simple file is transferred. This layer handles the details of data transmission. Several techniques for transmission control and protocol

(TCP) congestion control are discussed. Next, *congestion-avoidance* methods, which are methods of using precautionary algorithms to avoid a possible congestion in a TCP session, are presented. The chapter ends with a discussion of methods of ATM congestion control.

Chapter 9, Applications and Network Management, presents the fundamentals of the *application layer*, which determines how a specific user application should use a network. Among the applications are the *Domain Name System* (DNS); *e-mail protocols*, such as SMTP, and the *World Wide Web* (WWW).

Chapter 10, Network Security, focuses on security aspects of networks. After introducing network threats, hackers, and attacks, this chapter discusses encryption techniques: public- and private-key protocols, encryption standards, key-exchange algorithms, authentication methods, digital signature and secure connections, firewalls, IPsec, and security methods for virtual private networks.

Chapter 11, Packet Queues and Delay Analysis, begins Part II, discussing Little's theorem, Markov chain theorem, and birth and death processes. Queueing-node models are presented with several scenarios: finite versus infinite queueing capacity, one server versus several servers, and Markovian versus non-Markovian systems. Non-Markovian models are essential for many network applications, as multimedia traffic cannot be modeled by Markovian patterns. In addition, delay analysis, based on networks of queues, is discussed. *Burke's theorem* is applied in both serial and parallel queueing nodes. *Jackson's theorem* is presented for situations in which a packet visits a particular queue more than once, resulting in *loops* or *feedback*.

Chapter 12, Quality of Service and Resource Allocation, covers quality-of-service issues in networking. The two broad categories of QoS discussed are the *integrated services approach*, for providing service quality to networks that require maintaining certain features in switching nodes; and the *differentiated services approach* (DiffServ), which is based on providing quality-of-service support to a broad class of applications. These two categories include a number of QoS protocols and architectures, such as *traffic shaping*, *admission control*, *packet scheduling*, *reservation methods*, the Resource *Reservation Protocol* (RSVP), and traffic conditioner and bandwidth broker methods. This chapter also explains fundamentals of resource allocation in data networks.

Chapter 13, Networks in Switch Fabrics, looks inside switch fabrics of such Internet devices as routers. The chapter begins by classifying characteristics of switching networks and presenting features and basic definitions of switch fabrics. As the building blocks of switching fabrics, *crossbar switches* are emphasized. In particular, a case study at the end of chapter combines a number of buffered crosspoints to form a buffered crossbar. A number of other switch architectures—both blocking and nonblocking, as

well as, shared-memory, *concentration-based*, and *expansion-based* switching networks are presented.

Chapter 14, Optical Networks and WDM Systems, presents principles of fiber-optic communications and networking. The optical communication technology uses principles of light emission in the glass medium, which can carry more information over longer distances than electrical signals can carry in a copper or coaxial medium. The discussion on optical networks starts with basic optical devices, such as *optical filters*, *wavelength-division multiplexers* (WDMs), *optical switches*, and *optical buffers* and *optical delay lines*. After detailing optical networks using routing devices, the chapter discusses *wavelength re-use and allocation* as a link in all-optical networks. The chapter ends with a case study on an optical switching network, presenting a new topology: the *spherical switching network* (SSN).

Chapter 15, Multicasting Techniques and Protocols, covers the multicast extension of routing protocols in the Internet. First, the chapter defines basic terms and algorithms: multicast group, multicast addresses, and multicast tree algorithms, which form the next set of foundations for understanding packet multicast in the Internet. Two main classes of protocols are discussed: *intradomain* multicast routing protocols, by which packets are multicast within a domain; and *interdomain* routing protocol, by which packet multicast among domains is managed. In addition, techniques and algorithms used within the hardware of routers are introduced.

Chapter 16, VPNs, Tunneling, and Overlay Networks, introduces some useful Internet applications. The chapter explains how networks can be *overlaid* or *tunneled* and describes *virtual private networks* (VPNs), by which a private-sector entity tunnels over the public networking infrastructure, maintaining private connections. Other, related topics in this chapter are *multiprotocol label switching* (MPLS) networks and *overlay networks*.

Chapter 17, Compression of Digital Voice and Video, focuses on data-compression techniques for voice and video to prepare digital voice and video for multimedia networking. The chapter starts with the analysis of information-source fundamentals, source coding, and limits of data compression and explains all the steps of the conversion from raw voice to compressed binary form, such as sampling, quantization, and encoding. The chapter also summarizes the limits of compression and explains typical processes of still-image and video-compression techniques, such as JPEG, MPEG, and MP3. An end-of-chapter case study covers most of the chapter content, looking at FAX compression.

Chapter 18, VoIP and Multimedia Networking, presents the transportation of real-time signals along with the signaling protocols used in voice over IP (VoIP)

telephony and multimedia networking. The chapter presents protocols designed to provide real-time service requirements to the Internet. After discussing the *Session Initiation Protocol* (SIP) and the *H.323 series of protocols*, which are responsible for session signaling and numbering, real-time transport protocols, such as *Real-Time Transport protocol* (RTP) and the *Real-Time Control Protocol* (RTCP) are presented. The next topic is streaming video in a single server, using *content distribution networks* (CDNs). Also discussed is the *Stream Control Transmission Protocol* (SCTP), which provides a general-purpose transport protocol for transporting stream traffic. The chapter ends with detailed streaming source modeling and analysis.

Chapter 19, Mobile Ad-Hoc Networks, presents a special type of wireless networks, known as the *mobile ad-hoc network* (MANET). Ad-hoc networks do not need any fixed infrastructure to operate and support dynamic topology scenarios where no wired infrastructure exists. The chapter explains how a mobile user can act as a routing node and how a packet is routed from a source to its destination without having any static router in the network. The chapter also discusses *table-driven routing protocols* such as DSDV, CGSR, and WRP, and also *source-initiated routing protocols*, as well as DSR, ABR, TORA, and AODV. At the end of the chapter, we will disscuss the security of ad-hoc networks.

Chapter 20, Wireless Sensor Networks, presents an overview of such sensor networks and describes intelligent sensor nodes, as well as an overview of a protocol stack for sensor networks. The chapter explains how the "power" factor distinguishes the routing protocols of sensor networks from those of computer networks and describes *clustering protocols* in sensor networks. These protocols specify the topology of the hierarchical network partitioned into nonoverlapping *clusters* of sensor nodes. The chapter also presents a typical routing protocol for sensor networks, leading to a detailed numerical case study on the implementation of a clustering protocol. This chapter ends with *ZigBee technology*, based on IEEE standard 802.15.4. This technology uses low-power nodes and is a well-known low-power standard.

Exercises and Computer Simulation Projects

A number of exercises are given at the end of each chapter. The exercises normally challenge readers to find the directions to solutions in that chapter. The answers to the exercises are not always simple and may be more elusive, but this is typical of real and applied problems in networking. These problems encourage the reader to go back through the text and pick out what the instructor believes is significant. Besides typical exercises problems, there are numerous occasions for those who wish to incorporate

projects into their courses. The computer simulation projects are normally meant to be a programming miniproject. Projects listed in the exercises range from simulations to partial hardware design.

Throughout the text are case studies that show how and where computer communication integration is used with the materials studied in the associated chapter. A case study is basically a practical example for better understanding the essence of the corresponding chapter.

Appendixes

The book's appendixes make it essentially self-sufficient. **Appendix A, Glossary of Acronyms**, defines acronyms. **Appendix B, RFCs**, encourages readers to delve more deeply into each and every protocol presented in the book by consulting the many references provided. **Appendix C, Probabilities and Stochastic Processes**, reviews probabilities, random variables, and random processes.

Instructions and Instructor Supplements

The text can be used in a variety of ways. An instructor can use Part I of the book for the first graduate or a senior undergraduate course. In this case, one or two additional chapters from Part II, such as Chapters 13, 15, or 16, can also be included.

Part II of the text is aimed at the second graduate course in computer communication networks. An instructor can also choose the desired chapters, depending on the need and the content of her/his course. For example, if the graduate course is geared more toward architectures, Chapters 12, 13, 14, and 15 would be appropriate. Nevertheless, an instructor can include a couple of chapters from Part I, such as Chapters 3 and 6, in her/his graduate course.

An *instructor's solutions manual* is available to qualified instructors. Other instruction material, such as Power Point presentations will also be provided to instructors. Please contact the publisher for more information.

Acknowledgments

Writing a text is rarely an individual effort. Many experts from industry and academia -iously provided help. I would like to thank them all very warmly for their support. Many of them have given me invaluable ideas and support during this project. I should

acknowledge all those scientists, mathematicians, professors, engineers, authors, and publishers who helped me in this project.

I am honored to publish this book with Prentice Hall. I wish to express my deep gratitude to everyone who made a tremendous effort to make this project succeed. In particular, I would like to thank acquisitions editor Catherine Nolan for all her help. In the middle of the last phase of the review process, Catherine voluntarily set up a meeting while I was attending a conference in Boston and provided me with invaluable information on my remaining tasks in this project. I would also like to thank Managing Editor John Fuller, Marketing Manager Suzette Ciancio, Project Editor Lara Wysong, Development Editor Jinnifer Blackwell, Copy Editor Evelyn Pyle, and San Francisco Bay Sales Representative Ellen Wynn, who enthusiastically initiated my manuscript to the publisher.

I am grateful to all reviewers of this book for making constructive suggestions that helped me reshape the book to its present form. I would especially like to recognize the following people, who provided invaluable feedback during the writing phase of the manuscript. I took all their comments seriously and incorporated them into the manuscript. I greatly appreciate their time spent on this project.

Professor Nirwan Ansari (New Jersey Institute of Technology)

Professor Mohammed Atiquzzaman (University of Oklahoma)

Dr. Radu Balan (Siemens Corporate Research)

R. Bradley (About.Com, CompNetworking Guide)

Deepak Biala (On-Fiber Communications)

Dr. Robert Cane (VPP, United Kingdom)

M. Kevin Choy (Atmel, Colorado)

Professor Rod Fatohi (San Jose State University)

Professor Zongming Fei (University of Kentucky)

Dr. Carlos Ferari (JTN-Network Solutions)

Professor Jim Griffioen (University of Kentucky)

Dr. Jac Grolan (Alcatell)

Ajay Kalamber

Aurna Ketaraju (Intel)

Dr. Hardeep Maldia (Sermons Communications)

Will Morse (Texas Safe-Computing)

Professor Sarhan Musa (P. V. Texas A&M University)
Professor Achille Pattavina (Politecnico di Milano TNG)
Dr. Robert J. Paul (NsIM Communications)
Bala Peddireddi (Intel)
Christopher H. Pham (Cisco Systems)
Jasmin Sahara (University of Southern California)
Dipti Sathe (Altera Corporation)
Dr. Simon Sazeman (Siera Communications and Networks)
Professor Mukesh Singhal (University of Kentucky)
Professor Kazem Sohraby (University of Arkansas)
Dr. Richard Stevensson (BoRo Comm)
Kavitha Venkatesan (Cisco Systems)
Dr. Steve Willmard (SIM Technology)
Dr. Hemeret Zokhil (JPLab)

I am truly indebted to my graduate students, who helped me throughout the long phase of preparing the manuscript of this textbook. Over the years, more than 75 of my master's and PhD students read various portions of this book and made constructive comments. I wish them the best for their honest support and verbal comments on early versions of this book used in my class lectures. I am especially thankful to those who reviewed some sections of the book while taking my networking courses: Nasima Akhtar, Robert Bergman, Anushya Dharmarajan, Sudha Immaneni Howard Chan, Rekha Modipalle, Rinku Pal, Vishal Parikh, Preeti Pujni, Rashi Sharma, Maryam Tabesh, and Sitthapon Pumpichet.

Special thanks to Marzieh Veyseh, University of California–Santa Cruz, whose work on sensor networks resulted in our mutually published journal articles and for making all the information about sensor networks available for Chapter 20. Many thanks to Dr. Belle W. Wei, Dean of Engineering, SJSU, whose constructive comments resulted in great improvements in those journal articles and impacted Chapter 20.

I am also thankful to Dr. Eftekhari for a great technical help and support, and Professor Michael O'Flynn (not only a colleague but also a mentor), who graciously read through some sections of this book and made invaluable comments. Last but not least, thanks to Parviz Karandish, my best friend and a mentor who has helped me in many ways to make this project happen.

How to Contact the Author

Please feel free to send me any feedback at the Department of Electrical Engineering, San Jose State University, San Jose, CA 95192, U.S.A, or via e-mail at nmir@sjsu.edu. I would love to hear from you, especially if you have suggestions for improving this book. I will carefully read all review comments. You can find out more about me at www.engr.sjsu.edu/nmir. I hope that you enjoy the text and that you receive from it a little of my enthusiasm for computer communications.

Nader F. Mir
San Jose, California

About the Author

Nader F. Mir received the B.Sc. degree (with honors) in electrical and computer engineering in 1985 and the M.Sc. and PhD degrees, both in electrical engineering, from Washington University in St. Louis, Missouri, in 1990 and 1994, respectively.

He is currently a full professor and associate chairman of the Electrical Engineering Department at San Jose State University, California. He is also the director of MSE Program in Optical Sensors for Lockheed Martin Space Systems. Previously, he was an associate professor at this school and assistant professor at the University of Kentucky in Lexington. From 1994 to 1996, he was a research scientist at the Advanced Telecommunications Institute, Stevens Institute of Technology, in New Jersey, working on the design of advanced telecommunication networks. From 1990 to 1994, he was with the Computer and Communications Research Center at Washington University in St. Louis and worked as a research assistant on design and analysis of high-speed switching-systems projects.

His research interests are analysis of computer communication networks, design and analysis of switching systems, network design for wireless ad-hoc and sensor systems, and applications of digital integrated circuits in computer communications.

He is a senior member of the IEEE and has served as a member of the Technical Program Committee and the Steering Committee of a number of major IEEE networking conferences, such as WCNC, GLOBECOM, and ICC. Dr. Mir has published numerous technical journal and conference papers, all in the field of communications and networking. He has published a book on video communication engineering and a textbook, *Computer and Communication Networks,* published by Prentice Hall.

Dr. Mir has received a number of prestigious national and university awards, including the university teaching recognition award and research excellence award. He

is also the recipient of the 2004 IASTED Outstanding Tutorial Presentation award. He holds a successful United States patent for his invention of a new switching system for computer networks.

Currently, he has several journal editorial positions: editorial board member of the *International Journal of Internet Technology and Secured Transactions,* editor of the *Journal of Computing and Information Technology*, and associate editor of *IEEE Communication Magazine.*

PART I

FUNDAMENTAL CONCEPTS

CHAPTER 1

Packet-Switched Networks

Computer and communication networks provide a wide range of services, from simple networks of computers to remote-file access, digital libraries, videoconferencing, and networking billions of users and devices. Before exploring the world of computer and communication networks, we need to study the fundamentals of *packet-switched networks* as the first step. Packet-switched networks are the backbone of the data communication infrastructure. Therefore, our focus in this chapter is on the big picture and the conceptual aspects of this backbone:

- *Basic definitions in data networks*
- *Types of packet-switched networks*
- *Packet size and optimizations*

We start with the basic definitions and fundamental concepts, such as *messages*, *packets*, and *frames*, and *packet switching* versus *circuit switching*. We learn what the Internet is and how Internet service providers (ISPs) are formed. We then proceed to types of packet-switched networks and how a message can be handled by either *connection-oriented networks* or *connectionless networks*. Because readers must get a good understanding of *packets* as data units, packet size and optimizations are also discussed.

1.1 Basic Definitions in Data Networks

Communication networks have become essential media for homes and businesses. The design of modern computer and communication networks must meet all the requirements for new communications applications. A ubiquitous *broadband network* is the goal of the networking industry. Communication services need to be available anywhere and anytime. The broadband network is required to support the exchange of multiple types of information, such as voice, video, and data, among multiple types of users, while satisfying the performance requirement of each individual application. Consequently, the expanding diversity of high-bandwidth communication applications calls for a unified, flexible, and efficient network. The design goal of modern communication networks is to meet all the networking demands and to integrate capabilities of networks in a broadband network.

Packet-switched networks are the building blocks of computer communication systems in which data units known as *packets* flow across networks. The goal of a broadband packet-switched network is to provide flexible communication in handling all kinds of connections for a wide range of applications, such as telephone calls, data transfer, teleconferencing, video broadcasting, and distributed data processing. One obvious example for the form of traffic is *multirate* connections, whereby traffic containing several different bit rates flows to a communication node. The form of information in packet-switched networks is always digital bits. This kind of communication infrastructure is a significant improvement over the traditional telephone networks known as *circuit-switched networks*.

1.1.1 Packet Switching versus Circuit Switching

Circuit-switched networks, as the basis of conventional telephone systems, were the only existing personal communication infrastructures prior to the invention of packet-switched networks. In the new communication structure, voice and computer data are treated the same, and both are handled in a unified network known as a packet-switched network, or simply an integrated data network. In conventional telephone networks, a circuit between two users must be established for a communication to occur. Circuit-switched networks require resources to be reserved for each pair of end users. This implies that no other users can use the already dedicated resources for the duration of network use. The reservation of network resources for each user results in an inefficient use of bandwidth for applications in which information transfer is bursty.

Packet-switched networks with a unified, integrated data network infrastructure known as the Internet can provide a variety of communication services requiring

different bandwidths. The advantage of having a unified, integrated data network is the flexibility to handle existing and future services with remarkably better performance and higher economical resource utilizations. An integrated data network can also derive the benefits of central network management, operation, and maintenance. Numerous requirements for integrated packed-switched networks are explored in later chapters:

- Having robust routing protocols capable of adapting to dynamic changes in network topology
- Maximizing the utilization of network resources for the integration of all types of services
- Providing quality of service to users by means of priority and scheduling
- Enforcing effective congestion-control mechanisms that can minimize dropping packets

Circuit-switched networking is preferred for real-time applications. However, the use of packet-switched networks, especially for the integration and transmission of voice and data, results in the far more efficient utilization of available bandwidth. Network resources can be shared among other eligible users. Packet-switched networks can span a large geographical area and comprise a web of switching *nodes* interconnected through transmission links. A network provides links among multiple users facilitating the transfer of information. To make efficient use of available resources, packet-switched networks dynamically allocate resources only when required.

1.1.2 Messages, Packets, and Frames

A packet-switched network is organized as a multilevel hierarchy. In such networks, digital messages are fragmented into one or more smaller units of messages, each appended with a *header* to specify control information, such as the source and the destination addresses. This new unit of formatted message is called a *packet*, as shown in Figure 1.1. Packets are forwarded to a data network to be delivered to their destinations. In some circumstances, packets are also required to be attached together or further fragmented, forming a new packet known as a *frame*. Sometimes, a frame may be required to have multiple headers to carry out multiple tasks in multiple layers of a network, as shown in the figure.

As shown in Figure 1.2, two packets, A and B, are being forwarded from one side of a network to the other side. Packet-switched networks can be viewed from either an external or an internal perspective. The external perspective focuses on the network

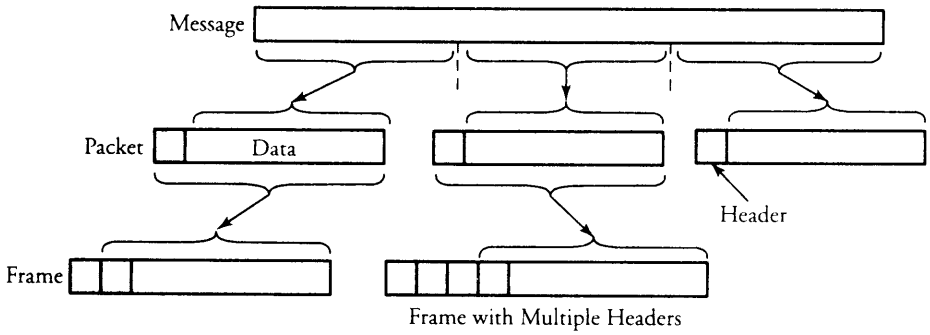


Figure 1.1 Creating packets and frames out of a raw digital message

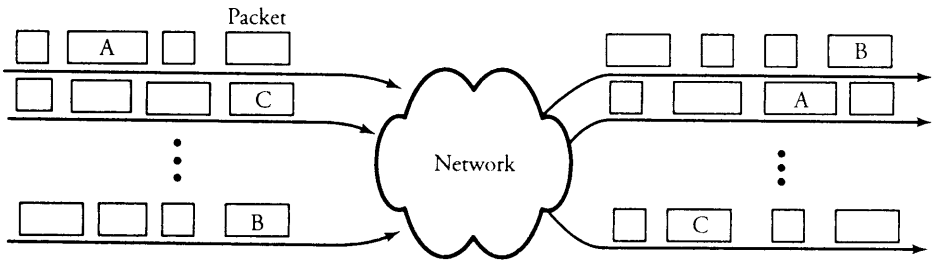


Figure 1.2 A packet-switched network receiving various-sized packets to route out

services provided to the upper layers; the internal perspective, on the fundamentals of *network topology*, structure of communication protocols, and addressing schemes.

A single packet, as the smallest unit of data for networking, may even be split into multiple packets before transmission. This well-known technique is called *packet fragmentation*. Apart from measuring the delay and ensuring that a packet is correctly sent to its destination, we also focus on delivering and receiving packets in a correct sequence when the data is fragmented. The primary function of a network is directing the flow of data among the users.

1.1.3 The Internet

The *Internet* is the collection of hardware and software components that make up our global communication network. The Internet is indeed a collaboration of interconnected communication vehicles that can network all connected communicating devices and equipment and provide services to all distributed applications. It is almost impossible to plot an exact representation of the Internet, since it is continuously being

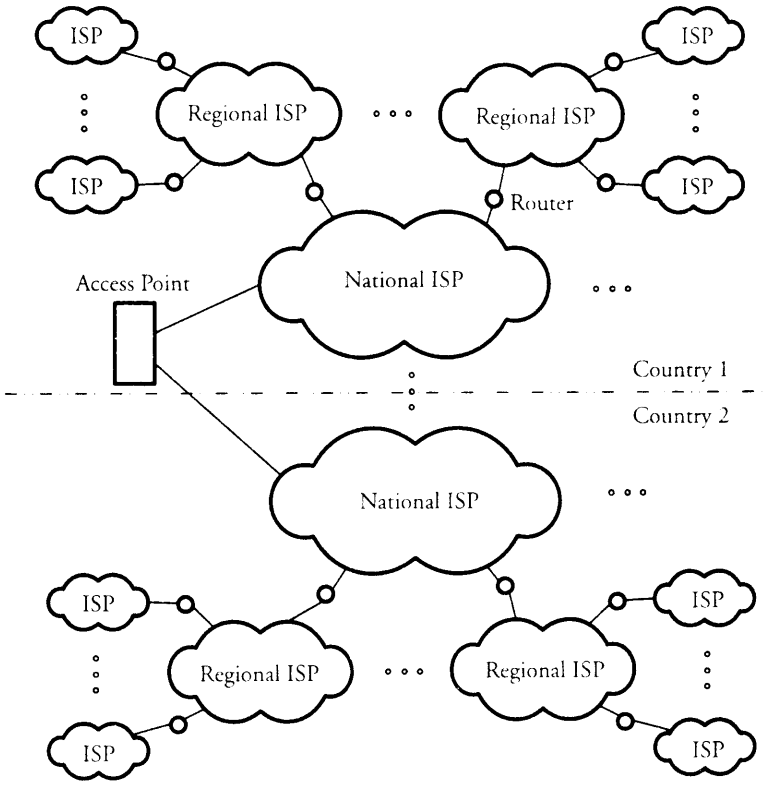


Figure 1.3 The Internet, a global interconnected network

expanded or altered. One way of imagining the Internet is shown in Figure 1.3, which illustrates a big-picture view of the worldwide computer network.

To connect to the Internet, users need the services of an *Internet service provider*. Each country has international or national service providers, regional service providers, and local service providers. At the top of the hierarchy, national Internet service providers connect nations or provinces together. The traffic between each two national ISPs is very heavy. Two such ISPs are connected together through complex switching stations called *network access points* (NAPs). Each NAP has its own system administrator.

In contrast, *regional Internet service providers* are smaller ISPs connected to a national ISP in a hierarchical chart. A *router* can operate as a device to connect to ISPs. Routers operate on the basis of one or more common *routing protocols*. In computer networks, the entities must agree on a protocol, a set of rules governing data communications and defining when and how two users can communicate with each other.

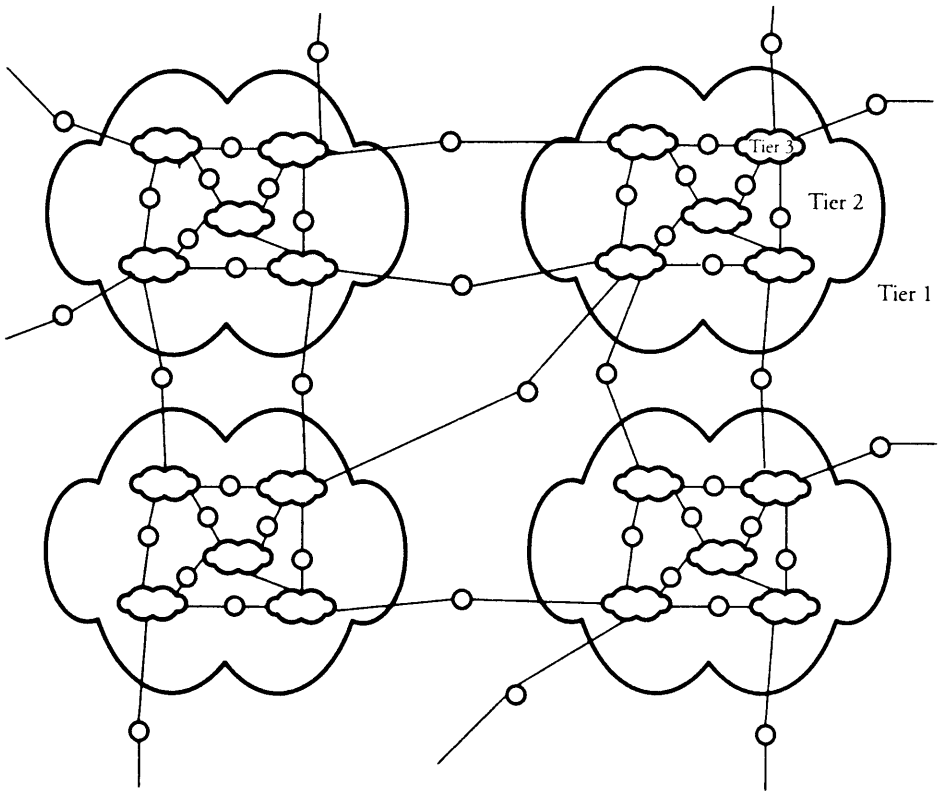


Figure 1.4 Hierarchy of networks

Each regional ISP can give services to part of a province or a city. The lowest networking entity of the Internet is a *local Internet service provider*. A local ISP is connected to a regional ISP or directly to a national service provider and provides a direct service to end users called *hosts*. End users and systems are connected together by communication *links*. An organization that supplies services to its own employees can also be a local ISP.

Figure 1.4 illustrates a different perspective of the global interconnected network. Imagine the global network in a hierarchical structure. Each ISP of a certain hierarchy or tier manages a number of other network domains at its lower hierarchy. The structure of such networks resembles the hierarchy of nature from the universe to atoms and molecules. Here, Tier 1, Tier 2, and Tier 3 represent, respectively, a national ISP, a regional ISP, and a local ISP.

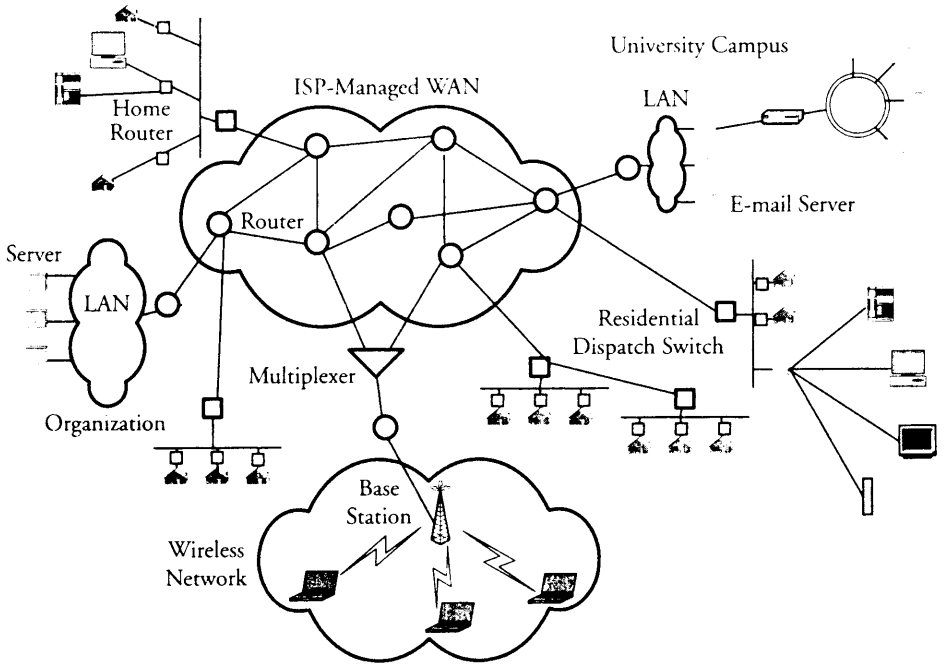


Figure 1.5 Overview of a regional Internet service provider (ISP)

1.1.4 ISPs and Internetwork Components

Figure 1.5 shows an Internet service provider. Networks can be classified into two main categories: *wide area networks* (WANs) and *local area networks* (LANs). A wide area network can be as large as the entire infrastructure of the data network access system known as the Internet. Figure 1.5 shows several networks, including LANs and WANs. End systems are indirectly connected to each other through intermediate *switching nodes* known as *packet routers*, or simply *routers*. Switching devices are key components that allow the flow of information to be switched over other links. Meanwhile, multiple local area networks are interconnected using *border routers* to form a wide area network. These routers contain information about the network routes, and their tasks are to route packets to requested destinations.

Multiple users accessing a single transmission medium at the same time are connected not only by switching nodes and interconnection links but also an access *multiplexer*. The multiplexer combines traffic from several nodes into a cumulative flow. This technique improves the bandwidth utilization efficiently. In Figure 1.5, we can

see that some aggregated links from a WAN are multiplexed into one link directed to a *wireless network*. In most cases, the aggregated packets are forwarded through a major network access node, as seen in Figure 1.5 for the wireless infrastructure. A separate network managed by a network administrator is known as a *domain*, or an *autonomous system*.

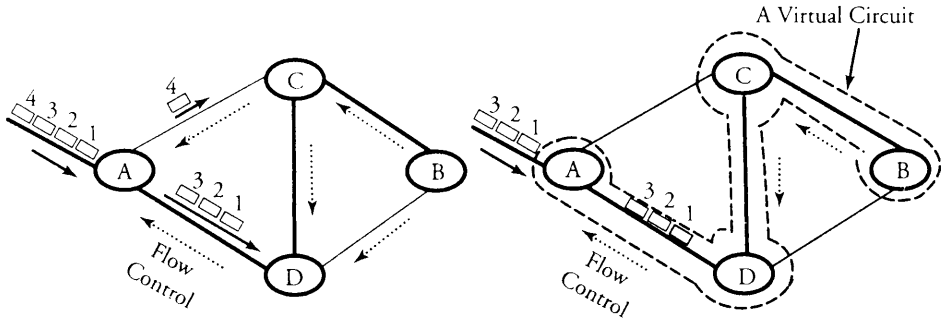
As an example of the local area network, a college campus network is connected to the Internet via a border router that connects the campus to an Internet service provider. ISP users are connected to access points of a WAN, as seen in Figure 1.5. Various ISPs can be interconnected through a high-speed router. Service providers have varying policies to overcome the problem of bandwidth allocations on routers. An ISP's *routing server* is conversant with the policies of all other service providers. Therefore, the routing server can direct the received routing information to an appropriate ISP.

When a link failure occurs, in packet-switched networks, the neighboring nodes share the fault information with other nodes, resulting in updating the routing tables. This way, packets may get routed through alternative paths bypassing the fault. Building the *routing table* in a router is one of the principal challenges of packet-switched networks. Designing the routing table for large networks requires maintaining data pertaining to traffic patterns and network topology information.

1.2 Types of Packet-Switched Networks

Packet-switched networks are classified into *datagram* or *connectionless* networks and *virtual-circuit* or *connection-oriented* networks, depending on the technique used for transferring information. The simplest form of a network service is based on the connectionless protocol. In this type of network, a user can transmit a packet anytime, without notifying the network layer. Packets are encapsulated into a certain "formatted" header, resulting in the basic Internet transmission unit of data, or *datagram*. A datagram is then sent over the network, with each router receiving the datagram forwarding it to the best router it knows, until the datagram reaches the destination. In this scheme, packets may be routed independently over different paths. However, the packets may arrive out of sequence. In this case, a certain network function (to be discussed later) takes care of the error control, flow control, and resequencing packets.

A related, though more complex, service is the connection-oriented protocol. Packets are transferred through an established virtual circuit between a source and a destination. When a connection is initially set up, network resources are reserved for the call duration. After the communication is finished, the connection is terminated, using a connection-termination procedure. During the call setup, the network can offer a



(a) Connectionless Network (b) Connection-Oriented Network

Figure 1.6 Two models of data networks: (a) a connectionless network and (b) a connection-oriented network

selection of options, such as best-effort service, reliable service, guaranteed delay service, and guaranteed bandwidth service, as explained in Chapters 8 and 12.

1.2.1 Connectionless Networks

Connectionless packet switching achieves high throughput at the cost of queuing delay. In this approach, large packets are normally fragmented into smaller packets. Packets from a source are routed independently of one another. The connectionless-switching approach does not require a call setup to transfer packets, but it has error-detection capability. The main advantage of this scheme is its capability to route packets through an alternative path in case a fault is present on the desired transmission link. On the flip side, this may result in the packet's arriving out of order and requiring sequencing at the destination.

Figure 1.6 (a) shows the routing of four packets in a connectionless network from point A to point B. The packets traverse the intermediate nodes in a *store-and-forward* fashion, whereby packets are received and stored at a node on a route; when the desired output of the node is free for that packet, the output is forwarded to its next node. In other words, on receipt of a packet at a node, the packet must wait in a queue for its turn to be transmitted. Nevertheless, packet loss may still occur if a node's buffer becomes full. The node determines the next hop read from the packet header. In this figure, the first three packets are moving along the path A, D, C, and B, whereas the fourth packet moves on a separate path, owing to congestion on path A–D.

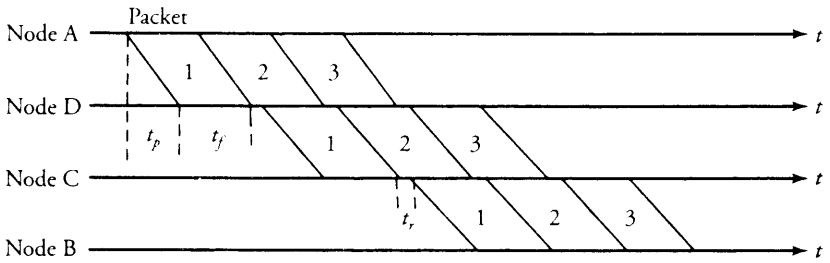


Figure 1.7 Signaling delay in a connectionless environment

The delay model of the first three packets discussed earlier is shown in Figure 1.7. The total transmission delay for a message three packets long traversing from the source node A to the destination node B can be approximately determined. Let t_p be the propagation delay between each two nodes, and let t_f be the packet-transfer time from one node to the next one. A packet is processed once it is received at a node with a processing time t_r. The total transmission delay, D_p for n_h nodes and n_p packets, in general is

$$D_p = [n_p + (n_h - 2)]t_f + (n_h - 1)t_p + n_h t_r. \tag{1.1}$$

In this formula, D_p gives only the transmission delay. As discussed in Chapter 11, another delay component, known as the *packet-queueing delay*, can be added to D_p. At this point, we focus only on the transmission delay and will discuss the queueing delay later.

Example. Figure 1.7 shows a timing diagram for the transmission of three packets on path A, D, C, B in Figure 1.6. Determine the total delay for transferring these three packets from node A to node B.

Solution. Assume that the first packet is transmitted from the source, node A, to the next hop, node D. The total delay for this transfer is $t_p + t_f + t_r$. Next, the packet is similarly transferred from node D to the next node to ultimately reach node B. The delay for each of these jumps is also $t_p + t_f + t_r$. However, when all three packets are released from node A, multiple and simultaneous transmissions of packets become possible. Thus, the total delay for all three packets to traverse the source and destination via two intermediate nodes is $D_p = 3t_p + 5t_f + 4t_r$.

Connectionless networks demonstrate the efficiency of transmitting a large message as a whole, especially in noisy environments, where the error rate is high. It is obvious

that the large message should be split into packets. Doing so also helps reduce the maximum delay imposed by a single packet on other packets. In fact, this realization resulted in the advent of connectionless packet switching.

1.2.2 Connection-Oriented Networks

In *connection-oriented*, or *virtual-circuit*, *networks*, a route set up between a source and a destination is required prior to data transfer, as in the case of conventional telephone networks. Figure 1.6 (b) shows a connection-oriented packet-switched network. The connection set-up procedure shown in this figure requires three packets to move along path A, D, C, and B with a prior connection establishment. During the connection set-up process, a virtual path is dedicated, and the forwarding routing tables are updated, at each node in the route.

Virtual-circuit packet switching typically reserves the network resources, such as the buffer capacity and the link bandwidth, to provide guaranteed quality of service and delay. The main disadvantage in connection-oriented packet-switched networks is that in case of a link or switch failure, the call set-up process has to be repeated for all the affected routes. Also, each switch needs to store information about all the flows routed through the switch.

The total delay in transmitting a packet in connection-oriented packet switching is the sum of the connection set-up time and the data-transfer time. The data-transfer time is the same as the delay obtained in connectionless packet switching. Figure 1.8 shows the overall delay for the three packets presented in the previous example. The transmission of the three packets starts with *connection request* packets and then *connection accept* packets. At this point, a circuit is established, and a partial path bandwidth is reserved for this connection. Then, the three packets are transmitted. At the end, a *connection release* packet clears and removes the established path.

The estimation of total delay time, D_t, to transmit n_p packets is similar to the one presented for connectionless networks. For connection-oriented networks, the

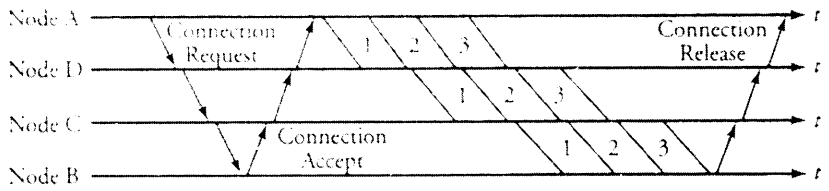


Figure 1.8 Signaling delay in a connection-oriented packet-switched environment

total time consists of two components: D_p, which represents the time to transmit packets, and D_c, which represents the time for the control packets. Control-packets' time includes the transmission delay for the connection-request packet, the connection-accept packet, and the connection-release packet:

$$D_t = D_p + D_c. \tag{1.2}$$

Another feature, called *cut-through switching*, can significantly reduce the delay. In this scheme, the packet is forwarded to the next hop as soon as the header is received and the destination is parsed. We see that the delay is reduced to the aggregate of the propagation times for each hop and the transfer time of one hop. This scheme is used in applications in which retransmissions are not necessary. Optical fiber transmission has a very low loss rate and hence uses cut-through switching to reduce the delay in transmitting a packet.

1.3 Packet Size and Optimizations

Packet size has a substantial impact on the performance of data transmission. Consider Figure 1.9, which compares the transmission of a 60-byte message using two different packet sizes with a packet header of 5 bytes. In the first scheme, the message is fragmented into three pieces of 20 bytes each, resulting in three packets of 25 bytes each. If these three packets are supposed to be transmitted over path A, D, C, and B in Figure 1.6, 75-byte units of delay are required. In contrast, if the message is fragmented into six messages of 10 bytes each, resulting in 15-byte packets, the total delay would be 60-byte units of delay. The reason for the time reduction is the parallel transmission of multiple packets at nodes D and C, since packets are smaller. This trend of delay reduction using smaller packets, however, is reversed at a certain point, owing to the dominance of packet overhead when a packet becomes very small.

To analyze packet size optimization, consider a link with a speed of s b/s or a bit rate of μ packets per second. Assume that packets of size $d + h$ are sent over this link at the rate λ packets per second, where d and h are the sizes of the packet data and the packet header, respectively, in bits. Clearly,

$$\mu = \frac{s}{d + h}. \tag{1.3}$$

We define *link utilization* to be $\rho = \lambda/\mu$. Then the percentage of link utilization used by data, ρ_d, is obtained by

$$\rho_d = \rho \left(\frac{d}{d + h} \right). \tag{1.4}$$

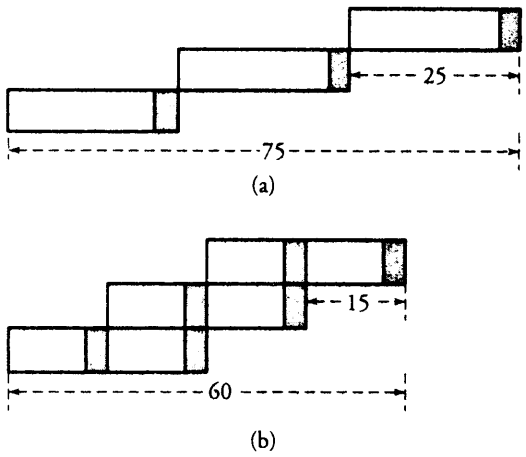


Figure 1.9 Comparison of two cases of transmitting data: (a) using three packets and (b) using six packets

The average delay per packet, D, can be calculated by using $\mu - \lambda$, where this term exhibits how close the offered load is to the link capacity:

$$D = \frac{1}{\mu - \lambda}. \tag{1.5}$$

Using equations (1.3) and (1.4), we can rewrite the average delay per packet as

$$D = \frac{1}{\mu(1-\rho)} = \frac{d+h}{s(1-\rho)} = \frac{d+h}{s\left[1 - \frac{\rho_d}{d}(d+h)\right]} \tag{1.6}$$

Apparently, the optimum size of a packet depends on several contributing factors. Here, we examine one of the factors by which the delay and the packet size become optimum. For optimality, consider d as one possible variable, where we want

$$\frac{\partial D}{\partial d} = 0. \tag{1.7}$$

This releases the two optimum values:

$$d_{opt} = h\left(\frac{\sqrt{\rho_d}}{1 - \sqrt{\rho_d}}\right) \tag{1.8}$$

and

$$D_{opt} = \frac{h}{s}\left(\frac{\sqrt{\rho_d}}{1-\sqrt{\rho_d}}\right)^2. \tag{1.9}$$

Note that here, d and D are optimized given only the mentioned variables. The optimality of d and D can also be derived by using a number of other factors that will result in a more accurate approach.

1.4 Summary

This chapter established a conceptual foundation for realizing all upcoming chapters. First, we clearly identified and defined all basic key terms in networking. We showed a big-picture view of computer networks in which from one side, connecting mainframe servers can be connected to a network backbone, and from the other side, home communication devices are connected to a backbone network over long-distance telephone lines. We illustrated how an Internet service provider (ISP) controls the functionality of networks. ISPs have become increasingly involved in supporting packet-switched networking services for carrying all sorts of data, not just voice, and the cable TV industry.

The transfer of data in packet-switched networks is organized as a multilevel hierarchy, with digital messages fragmented into units of formatted messages, or packets. In some circumstances, such as local area networks, packets must be modified further, forming a smaller or larger packet known as a frame. Two types of packet-switched networks are networks using connectionless protocol, in which no particular advanced connection is required, and networks using connection-oriented protocol, in which an advance dedication of a path is required.

A packet size can be optimized. Using the percentage of link utilization used by data, ρ_d, as a main variable, we showed that the optimized packet size and the optimized packet delay depend on ρ_d. The total delay of packet transfer in connectionless networks is significantly smaller than for connection-oriented networks, owing mainly to inclusion of signaling components in connection-oriented networks.

The next two chapters present an overview of the software and hardware foundations of our networking infrastructure. Chapter 2 provides some detail about communication protocols, and Chapter 3 introduces the devices used to construct a computer network.

1.5 Exercises

1. We transmit data directly between two servers 6,000 km apart through a geostationary satellite situated 10,000 km from Earth exactly between the two servers. The data enters this network at 100 Mb/s.
   (a) Find the propagation delay if data travels at the speed of light (2.3×10^8 m/s).
   (b) Find the number of bits in transit during the propagation delay.
   (c) Determine how long it takes to send 10 bytes of data and to receive 2.5 bytes of acknowledgment back.

2. Repeat exercise 1, but this time, suppose that the two servers are 60 meters apart on the same campus.

3. Stored on a CD-ROM is a 200 MB message to be transmitted by an e-mail from one mail server to another, passing three nodes of a *connectionless network*. This network forces packets to be of size 10 KB, including a packet header of 40 bytes. Nodes are 400 miles apart, and servers are 50 miles away from their corresponding nodes. All transmission links are of type 100 Mb/s. The processing time at each node is 0.2 seconds per packet.
   (a) Find the propagation delays per packet between a server and a node and between nodes.
   (b) Find the total time required to send this message.

4. Equation 1.2 gives the total delay time for connection-oriented networks. Let t_p be the packet-propagation delay between each two nodes, t_{f1} be the data packet transfer time to the next node, and t_{r1} be the data packet processing time. Also, let t_{f2} be the control-packet transfer time to the next node, and t_{r2} be the control-packet processing time. Give an expression for D in terms of all these variables.

5. Suppose that a 200 MB message stored on a CD-ROM is to be uploaded on a destination through a *virtual-circuit packet-switched network*. This network forces packets to be of size 10 KB, including a packet header of 40 bytes. Nodes are 400 miles apart, and servers are 50 miles away from their corresponding nodes. All transmission links are of type 100 Mb/s. The processing time at each node is 0.2 seconds per packet. For this purpose, the signaling packet is 500 bit long.
   (a) Find the total connection request/accept process time.
   (b) Find the total connection release process time.
   (c) Find the total time required to send this message.

6. We want to deliver a 12 KB message by uploading it in the destination's Web site through a 10-node path of a *virtual-circuit packet-switched network*. For this purpose, the signaling packet is 500 bits long. The network forces packets to be of size 10 KB including a packet header of 40 bytes. Nodes are 500 miles apart. All transmission links are of type 1 Gb/s. The processing time at each node is 100 ms per packet and the propagation speed is 2.3×10^8 m/s.
   (a) Find the total connection request/accept process time.
   (b) Find the total connection release process time.
   (c) Find the total time required to send this message.
7. To analyze the transmission of a 10,000-bit-long packet, we want the percentage of link utilization used by the data portion of a packet to be 72 percent. We also want the ratio of the packet header, h, to packet data, d, to be 0.04. The transmission link speed is $s = 100$ Mb/s.
   (a) Find the link utilization, ρ.
   (b) Find the link capacity rate, μ, in terms of packets per second.
   (c) Find the average delay per packet.
   (d) Find the optimum average delay per packet.
8. Consider a digital link with a maximum capacity of $s = 100$ Mb/s facing a situation resulting in 80 percent utilization. Equal-size packets arrive at 8,000 packets per second. The link utilization dedicated to headers of packets is 0.8 percent.
   (a) Find the total size of each packet.
   (b) Find the header and data sizes for each packet.
   (c) If the header size is not negotiable, what would the optimum size of packets be?
   (d) Find the delay for each optimal-sized packet.
9. Develop a signaling delay chart, similar to Figures 1.7 and 1.8, for circuit-switched networks. From required steps, get an idea that would result in the establishment of a telephone call over circuit-switched networks.
10. In practice, the optimum size of a packet estimated in Equation (1.7) depends on several other contributing factors.
    (a) Derive the optimization analysis, this time also including the header size, h. In this case, you have two variables: d and h.
    (b) What other factors might also contribute to the optimization of the packet size?

CHAPTER 2

Foundation of Networking Protocols

Users and networks are connected together by certain rules called *network communication protocols*. The Internet Protocol (IP), for example, is responsible for using prevailing rules to establish paths for packets. Communication protocols are the intelligence behind the driving force of packets and are tools by which a network designer can easily expand the capability of networks. One growth aspect of computer networking is clearly attributed to the ability to conveniently add new features to networks. New features can be added by connecting more hardware devices, thereby expanding networks. New features can also be added on top of existing hardware, allowing the network features to expand. The second method of network expansion requires modifications to communication protocols. This chapter focuses on the following:

- *Five-layer TCP/IP model*
- *Seven-layer OSI protocol model*
- *Internet protocols and addressing*
- *Equal-size packet protocol model*

The *five-layer TCP/IP model* is a widely accepted Internet backbone protocol structure. In this chapter, we give an overview of these five layers and leave any further details to be discussed in the remaining chapters. Among these five layers, the basics of *network layer* is designated a separate section in this chapter under Internet protocols. We make this

arrangement since basic definitions related to this layer are required in the following few chapters.

As there are numerous protocols formed together to be used for the movement of packets, the explanation of all other protocols will be spread over almost all upcoming chapters. In the meanwhile, the reader is cautiously reminded that getting a good grasp of the fundamental materials discussed in this chapter is essential for following the future details or extensions described in the remaining of the book. At the end of this chapter, the *equal-size packet protocol model* will also be introduced.

2.1 5-Layer TCP/IP Model

The basic structure of communication networks is represented by the *Transmission Control Protocol/Internet Protocol* (TCP/IP) model. This model is structured in five layers. An end system, an intermediate network node, or each communicating user or program is equipped with devices to run all or some portions of these layers, depending on where the system operates. These five layers, as shown in Figure 2.1, are as follows:

1. Physical layer
2. Link layer
3. Network layer
4. Transport layer
5. Application layer

Layer 1, the *physical layer*, defines electrical aspects of activating and maintaining physical links in networks. The physical layer represents the basic network hardware, such as switches and routers. The details of this layer are explained in later chapters, especially Chapters 3, 4, 6, 13, 15, 17, and 20.

Layer 2, the *link layer*, provides a reliable synchronization and transfer of information across the physical layer for accessing the transmission medium. Layer 2 specifies how packets access links and are attached to additional headers to form frames when entering a new networking environment, such as a LAN. Layer 2 also provides error detection and flow control. This layer is discussed further in Chapters 4, 5, 6, 19, and 20.

Layer 3, the *network layer* (IP) specifies the networking aspects. This layer handles the way that addresses are assigned to packets and the way that packets are supposed to be forwarded from one end point to another. Some related parts of this layer are described in Chapters 5, 6, 7, 10, 12, 14, 15, 16, 19, and 20.

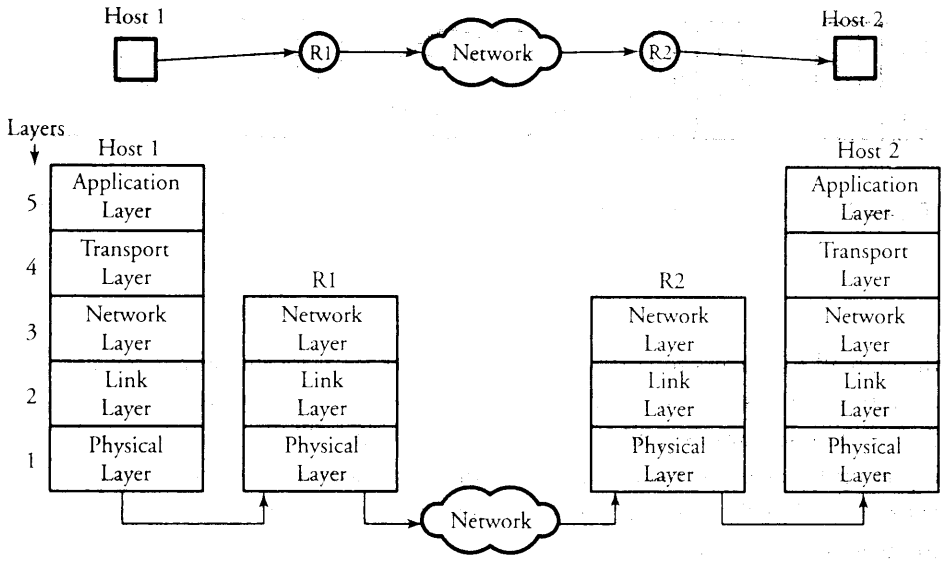


Figure 2.1 Hierarchy of 5-layer communication protocol model

Layer 4, the *transport layer*, lies just above the network layer and handles the details of data transmission. Layer 4 is implemented in the end-points but not in network routers and acts as an interface protocol between a communicating host and a network. Consequently, this layer provides logical communication between processes running on different hosts. The concepts of the transport layer are discussed further in Chapters 8, 9, 12, 16, 18, 19, and 20.

Layer 5, the *application layer*, determines how a specific user application should use a network. Among such applications are the *Simple Mail Transfer Protocol* (SMTP), *File Transfer Protocol* (FTP), and the *World Wide Web* (WWW). The details of layer 5 are described in Chapters 9, 10, 15, 16, 18, 19, and 20.

The transmission of a given message between two users is carried out by (1) flowing down the data through each and all layers of the transmitting end, (2) sending it to certain layers of protocols in the devices between two end points, and (3) when the message arrives at the other end, letting the data flow up through the layers of the receiving end until it reaches its destination. Figure 2.1 illustrates a scenario in which different layers of protocols are used to establish a connection. A message is transmitted from host 1 to host 2, and, as shown, all five layers of the protocol model participate in making this connection. The data being transmitted from host 1 is passed down

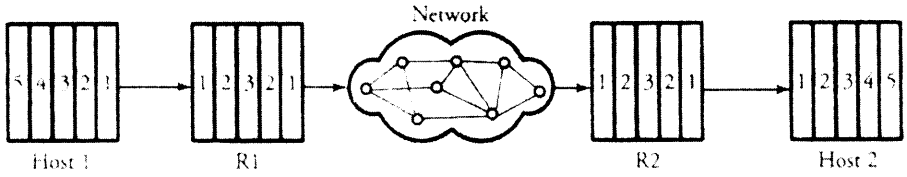


Figure 2.2 Structural view of protocol layers for two hosts communicating through two routers

through all five layers to reach router R1. Router R1 is located as a gateway to the operating regions of host 1 and therefore does not involve any tasks in layers 4 and 5. The same scenario is applied at the other end: router R2. Similarly, router R2, acting as a gateway to the operating regions of host 2, does not involve any tasks in layers 4 and 5. Finally at host 2, the data is transmitted upward from the physical layer to the application layer.

The main idea of the communication protocol stack is that the process of communication between two end points in a network can be partitioned into layers, with each layer adding its own set of special related functions. Figure 2.2 shows a different way of realizing protocol layers used for two hosts communicating through two routers. This figure illustrates a structural perspective of a communication set-up and identifies the order of fundamental protocol layers involved.

2.2 7-Layer OSI Model

The *open systems interconnection* (OSI) model was the original standard description for how messages should be transmitted between any two points. To the five TCP/IP layers, OSI adds the following two layers:

1. *Layer* 5, the *session layer*, which sets up and coordinates the applications at each end
2. *Layer* 6, the *presentation layer*, which is the operating system part that converts incoming and outgoing data from one presentation format to another

The tasks of these two additional layers are dissolved into the application and transport layers in the newer five-layer model. The OSI model is becoming less popular. TCP/IP is gaining more attention, owing to its stability and its ability to offer a better communication performance. Therefore, this book focuses on the five-layer model.

2.3 Internet Protocols and Addressing

The third layer of communication protocol hierarchy is the *network layer*, which specifies the networking aspects of a communication transaction. This *Internet Protocol* (IP) *layer* handles networking aspects and establishes routes for packets. The network layer, in fact, handles the method of assigning addresses to packets and determines how they should be forwarded from one end point to another.

The Internet Protocol produces a header for packets. An IP header contains the IP addresses of a source node and a destination node, respectively. An IP packet can be encapsulated in the layer 2 frames when the packet enters a LAN. The IP layer normally offers no QoS guarantees and provides a best-effort service. IP is inherently unreliable, relying on the higher layers, such as the transport protocol, to handle issues relating to system reliability.

IP provides seamless Internet connectivity and scalability. This layer is based on the *connectionless*, or so-called datagram switching, approach. The advantages of this kind of service are (1) flexibility to allow interconnection between diverse network topologies, and (2) robustness to node failure. Apart from the ability to connect diverse networks, the IP layer also fragments packets to the *maximum transmission unit* (MTU) and performs reassembly of packet fragments at destinations.

2.3.1 IP Packet

The packet format of IP version 4 (IPv4) is shown in Figure 2.3. Each packet comprises the header and data. The size of the header is variable, with 20 bytes of fixed-length header and an *options* field whose size is variable up to 40 bytes. A brief description of the fields follows.

- *Version* specifies the IP version.
- *Header length* (HL) specifies the length of the header.

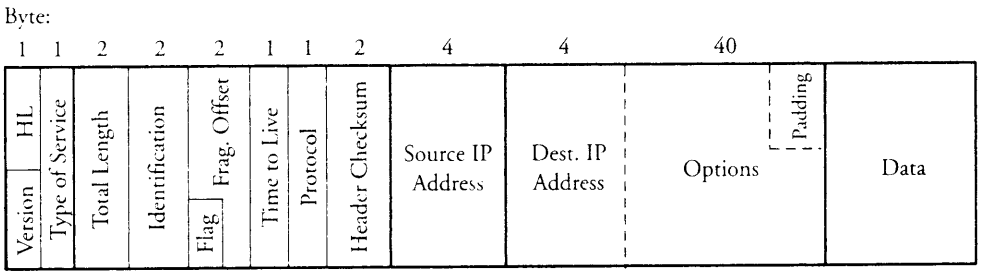


Figure 2.3 IP packet format

- *Type of service* specifies the quality-of-service (QoS) requirements of the packet, such as priority level, delay, reliability, throughput, and cost.
- *Total length* specifies the total length of the packet in bytes, including the header and data. A total of 16 bits are assigned to this field.
- *Identification, flags*, and *fragment offset* are used for packet fragmentation and reassembly.
- *Time to live* specifies the maximum number of hops after which a packet must be discarded.
- *Protocol* specifies the protocol used at the destination.
- *Header checksum* is a method of error detection and is described in Chapter 4.
- *Source address* and *destination address* are 32-bit fields specifying the source address and the destination address, respectively.
- *Options* is a rarely used variable-length field to specify security level, timestamp, and type of route.
- *Padding* is used to ensure that the header is a multiple of 32 bits.

Recall that the 16 bits in the *total length* field express the total length of a packet. Hence, the total length of the packet is limited to 2^{16} bytes. However, the maximum packet size of 2^{16} bytes is rarely used, since the packet size is limited by the physical network capacity. The real physical network capacity per packet is normally less than 10K and even gets smaller, to 1.5K when the packet reaches a LAN. To accomplish packet partitioning, the *identification, flags*, and *fragment offset* fields perform and keep track of the packet-fragmentation process when needed.

2.3.2 IP Addressing Scheme

The IP header has 32 bits assigned for addressing a desired device in the network. An IP address is a unique identifier used to locate a device on the IP network. To make the system scalable, the address structure is subdivided into the *network* ID and the *host* ID. The network ID identifies the network the device belongs to; the host ID identifies the device. This implies that all devices belonging to the same network have a single network ID. Based on the bit positioning assigned to the network ID and the host ID, the IP address is further subdivided into classes A, B, C, D (multicast), and E (reserved), as shown in Figure 2.4.

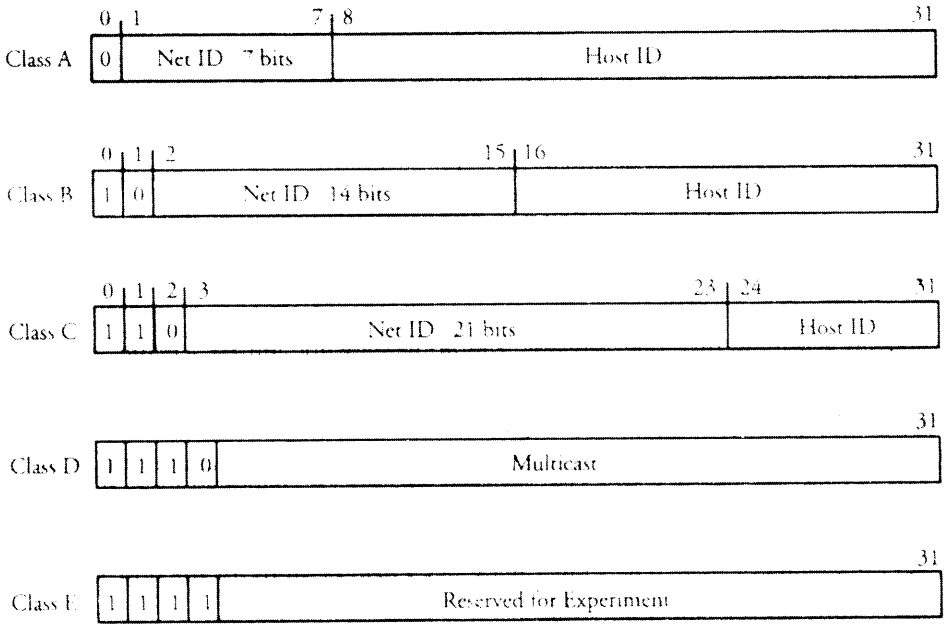


Figure 2.4 Classes of IP addresses

Consider the lengths of corresponding fields shown in this figure. Class A starts with a 0 and supports 126 networks and 16 million hosts per network. Class B addressing always starts with 10 and supports 16,382 networks and 65,535 hosts per network. Class C addressing starts with 110 and supports 2 million networks and 254 hosts per network. Class D addressing starts with 1110 and is specifically designed for multicasting and broadcasting. Class E always starts with 1111 reserved for network experiments. For ease of use, the IP address is represented in *dot-decimal* notation. The address is grouped into 4 dot-separated bytes.

Example. A host with an IP address of 10001000 11100101 11001001 00010000 belongs to class B, since it starts with 10, and its decimal equivalent is 136.229.201.16.

2.3.3 Subnet Addressing and Masking

The concept of subnetting was introduced to overcome the shortcomings of IP addressing. Managing the large number of hosts is an enormous task. For example, a company that uses a class B addressing scheme supports 65,535 hosts on one network. If the

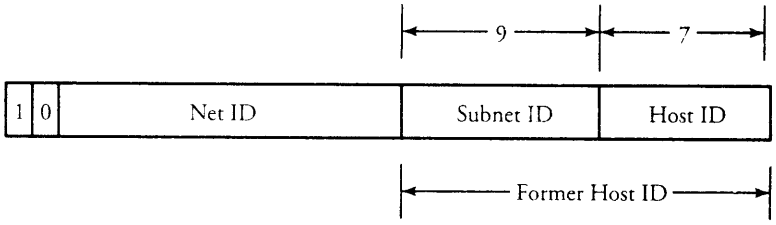


Figure 2.5 A subnet ID and host ID in class *B* addressing

company has more than one network, a multiple-network address scheme, or *subnet scheme*, is used. In this scheme, the host ID of the original IP address is subdivided into *subnet ID* and *host ID*, as shown in Figure 2.5.

Depending on the network size, different values of subnet ID and host ID can be chosen. Doing so would prevent the outside world from being burdened by a shortage of new network addresses. To determine the subnetting number, a subnet *mask*—logic AND function—is used. The subnet mask has a field of all 0s for the host ID and a field of all 1s for the remaining field.

Example. Given an IP address of 150.100.14.163 and a subnet mask of 255.255 .255.128, determine the maximum number of hosts per subnet.

Solution. Figure 2.6 shows the details of the solution. Masking 255.255.255.128 on the IP address results in 150.100.14.128. Clearly, the IP address 150.100.14.163 is a class B address. In a class B address, the lower 16 bits are assigned to the subnet and host fields. Applying the mask, we see that the maximum number of hosts is $2^7 = 128$.

Example. A router attached to a network receives a packet with the destination IP address 190.155.16.16. The network is assigned an address of 190.155.0.0. Assume that the network has two subnets with addresses 190.155.16.0 and 190.155.15.0 and that both subnet ID fields have 8 bits. Explain the details of routing the packet.

Solution. When it receives the packet, the router determines to which subnet the packet needs to be routed, as follows: The destination IP address is 190.155.16.16, the subnet mask used in the router is 255.255.255.0, and the result is 190.155.16.0. The router looks up its routing table for the next subnet corresponding to the subnet 190.155.16.0, which is subnet 2. When the packet arrives at subnet 2, the router determines that the destination is on its own subnet and routes the packet to its destination.

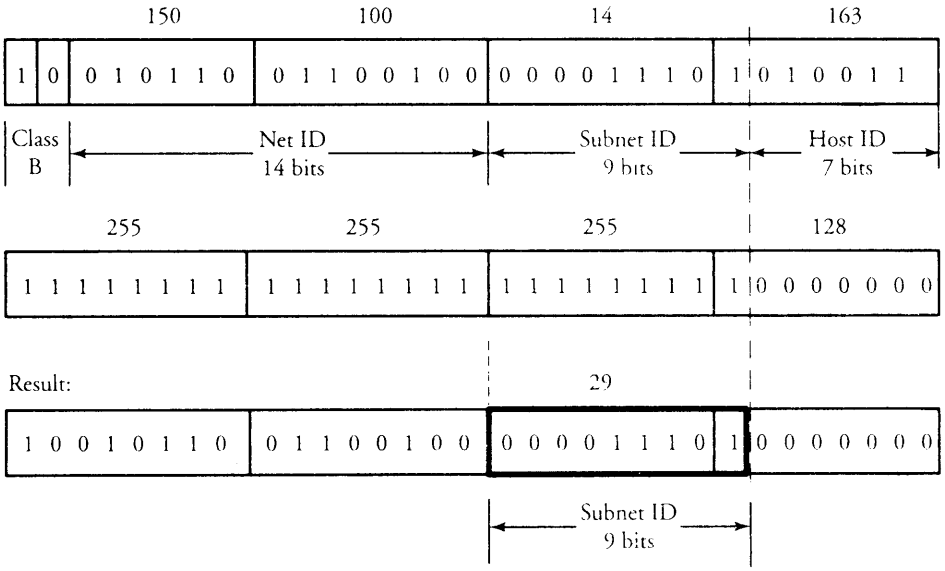


Figure 2.6 An example of subnet and masking

2.3.4 Classless Interdomain Routing (CIDR)

The preceding section described an addressing scheme requiring that the address space be subdivided into five classes. However, giving a certain class C address space to a certain university campus does not guarantee that all addresses within the space can be used and therefore might waste some addresses. This kind of situation is inflexible and would exhaust the IP address space. Thus, the classful addressing scheme consisting of classes A, B, C, D, and E results in an inefficient use of the address space.

A new scheme, with no restriction on the classes, emerged. *Classless interdomain routing* (CIDR) is extremely flexible, allowing a variable-length *prefix* to represent the network ID and the remaining bits of the 32-field address to represent the hosts within the network. For example, one organization may choose a 20-bit network ID, whereas another organization may choose a 21-bit network ID, with the first 20 bits of these two network IDs being identical. This means that the address space of one organization contains that of another one.

CIDR results in a significant increase in the speed of routers and has greatly reduced the size of routing tables. A routing table of a router using the CIDR address space has entries that include a pair of network IP addresses and the mask. *Supernetting* is a CIDR technique whereby a single routing entry is sufficient to represent a group of adjacent

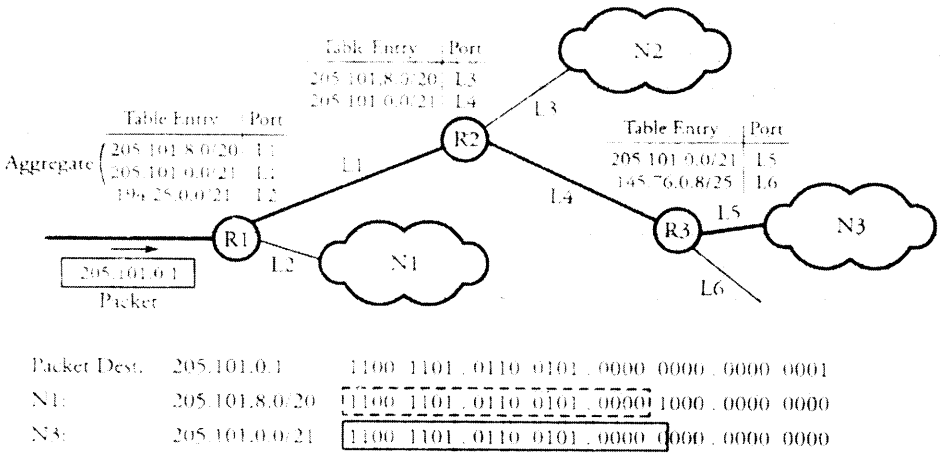


Figure 2.7 CIDR routing

addresses. Because of the use of a variable-length prefix, the routing table may have two entries with the same prefix. To route a packet that matches both of these entries, the router chooses between the two entries, using the longest-prefix-match technique.

Example. Assume that a packet with destination IP address 205.101.0.1 is received by router R1, as shown in Figure 2.7. In the entries of this router, two routes, L1 and L2, belonging to 205.101.8.0/20 and 205.101.0.0/21, respectively, are matched. CIDR dictates that the longer prefix be the eligible match. As indicated at the bottom of this figure, link L1, with its 21-bit prefix, is selected, owing to a longer match. This link eventually routes the packet to the destination network, N3.

CIDR allows us to reduce the number of entries in a router's table by using an *aggregate technique*, whereby all entries that have some common partial prefix can be combined into one entry. For example, in Figure 2.7, the two entries 205.101.8.0/20 and 205.101.0.0/21 can be combined into 205.101.0.0/20, saving one entry on the table. Combining entries in routing tables not only saves space but also enhances the speed of the routers, as each time, routers need to search among fewer addresses.

2.3.5 Packet Fragmentation and Reassembly

The physical capacity of networks enforces an upper bound on the size of packets. The *maximum transmission unit* (MTU) represents this restriction. For example, as a LAN

standard, Ethernet limits the size of flowing frames to be 1,500 bytes. The objective of inducing this method is that we need a mechanism that avoids requiring large buffers at intermediate routers to store the fragments. This restriction necessitates the Internet Protocol to break up large messages into fragments. The fragment sizes are limited to the MTU of the underlying physical network. The fragments could in turn be split into smaller fragments, depending on the physical network being used. Each fragment is routed independently through the network. Once all the fragments are received, they are reassembled at the final destination to form the original packet.

The identification, flag, and offset fields of the IP header help with the fragmentation and reassembly process. The identification field is used to distinguish between various fragments of different packets. The flag field has a more-fragment (MF) bit. When the MF bit is set, it implies that more fragments are on their way. The offset field indicates the position of a fragment in the sequence of fragments making up the packet. The lengths of all the fragments, with the exception of the last one, must be divisible by 8.

To be successfully reassembled, all fragments making up a packet must arrive at the destination. In the case of a missing fragment, the rest of the fragments have to be discarded, and thus the packet needs to be retransmitted. In such cases, the retransmission of packets results in an inefficient use of the network bandwidth.

Example. Suppose that a host application needs to transmit a packet of 3,500 bytes. The physical layer has an MTU of 1,500 bytes. The packet has an IP header of 20 bytes plus another attached header of 20 bytes. Fragment the packet, and specify the ID, MF, and offset fields of all fragments.

Solution. The allowable data length = 1,500 − 20 − 20 = 1,460 bytes. Because 1,460 is not divisible by 8, the allowable data length is limited to 1,456 bytes. Including the headers, the data to be transmitted is then 3,540 bytes to be split into fragments of 1,456, 1,456 and 628 bytes. Here, fragment 1 = total length 1,456, MF 1, offset 0; fragment 2 = total length 1,456, MF 1, offset 182; and fragment 3 = total length 628, MF 0, and offset 364.

2.3.6 Internet Control Message Protocol (ICMP)

In connectionless routing, routers operate autonomously. They forward and deliver packets without requiring any coordination with the source. In large communication networks, IP may not be able to deliver a packet to its destination, owing to possible failures in the connectivity of a destination. Besides the hardware failure, other factors

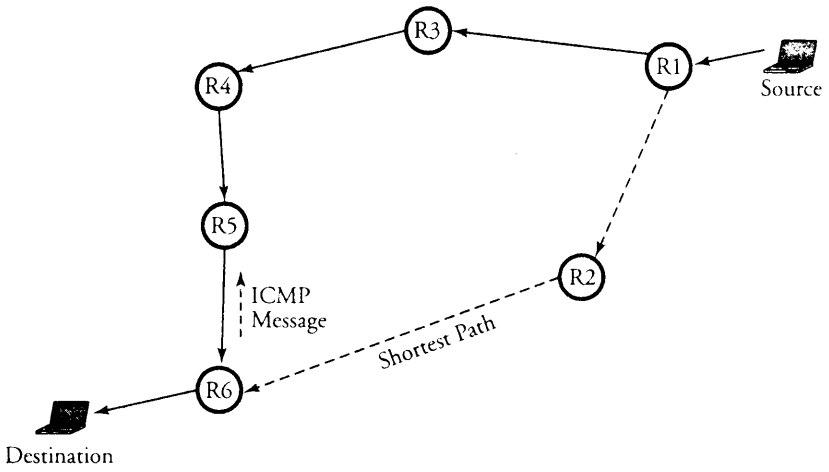


Figure 2.8 With ICMP, a redirect message cannot be sent to R1, since R6 does not know the address of R1.

may be present to create this problem. For example, as noted in Section 2.3.1, the *time-to-live* field in an IP packet specifies the maximum number of hops after which a packet must be discarded. If the counter of this field expires, packet delivery too can become impossible.

Another issue—related and equally important—is that a sender cannot know whether a delivery failure is a result of a local or a remote technical difficulty. With TCP/IP, routers in a network can report errors through the *Internet Control Message Protocol* (ICMP). An ICMP message is encapsulated in the data portion of an IP datagram (packet). When an error occurs, ICMP reports it to the originating source of the connection. This is compatible with the fact that an IP datagram header itself specifies only the original source and not any routers. The source must interpret the error.

One of the important ICMP messages is the *redirect* message. In Figure 2.8, a source tries to send a message to a destination. But R1 incorrectly sends the message to a wrong path (R1-R3-R4-R5-R6) instead of to the short one (R1-R2-R6). In this case, if in the middle of routing, R5 or R6 finds out about this error, it cannot issue an ICMP message to R1 to correct the routing, as they do not know the address of R1. Instead, they issue a redirect ICMP message to the source.

2.3.7 IP Version 6 (IPv6)

The use of IPv4 has resulted in the exhaustion of the 32-bit address space to the extent that IPv4 has run out of addressing spaces. Therefore, 128-bit address spacing was

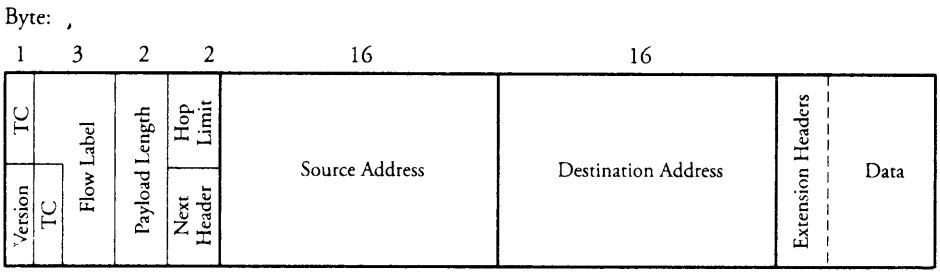


Figure 2.9 An IPv6 packet format

introduced with *Internet Protocol version 6* (IPv6). It enjoys tremendous popularity because of its simplicity and flexibility in adapting to diverse network technologies. Compatible with IPv4, IPv6 also supports real-time applications, including those that require guaranteed QoS. Figure 2.9 shows the IPv6 header. A brief description of the fields in the header follows.

- *Version* is the same as in IPv4, indicating the version number of the protocol.
- *Traffic class* specifies the priority level assigned to a packet.
- *Flow label* indicates the delay period within which application packets, such as real-time video, must be delivered.
- *Payload length* is the 16-bit specification of the length of the data, excluding the header.
- *Next header* specifies the type of extension header used. The functionality of the option field in IPv4 is specified in the extension header. In addition, the extension header is more flexible than the options field.
- *Hop limit* is the same as the time-to-live field in IPv4.
- *Source address* and *destination address* are each identified by a 128-bit field address.

The IPv4 and IPv6 header formats have some notable differences. First, IPv6 uses a 128-bit address field rather than the 32-bit field in IPv4. The 128-bit field can support a maximum of 3.4×10^{38} IP addresses. IPv6 has a simpler header format, eliminating the fragmentation, the checksum, and header length fields. The removal of the checksum field in IPv6 allows for faster processing at the routers without sacrificing functionality. In IPv6, *error detection* and *correction* are handled at the data link and the TCP layers. Note also that IPv6 can accommodate the QoS requirements for some applications. Besides all these significant advantages, IPv6 can provide built-in security features such as confidentiality and authentication. These features are discussed in Chapter 10.

IPv6 Addressing Format

With its large address spacing, IPv6 network addressing is very flexible. To efficiently represent the 128-bit address of IPv6 in a compact form, hexadecimal digits are used. A colon separates each of the four hexadecimal digits. For example, [2FB4 : 10AB : 4123 : CEBF : 54CD : 3912 : AE7B : 0932] can be a source address. In practice, IPv6 addresses contain a lot of bits that are zero. The address is commonly denoted in a more compact form. For example, an address denoted by [2FB4 : 0000 : 0000 : 0000 : 54CD : 3912 : 000B : 0932] can be compressed to [2FB4 : : : : 54CD : 3912 : B : 932].

The network address space is classified into various types, each of which is assigned a binary prefix. Currently, only a small portion of the address space has been assigned, with the remaining reserved for future use. One of the address types with a leading byte of 1s is assigned for multicast; the rest of the currently assigned types are used for unicast applications. Apart from the unicast and multicast addresses, IPv6 introduces *anycast* addresses. An anycast address is similar to a multicast address and identifies a group of network devices for making connections. However, unlike with multicast addressing, a packet needs to be forwarded to any one device in the group. Anycast addresses share the address space with unicast address types. IPv6 reserves some addresses for special purposes.

Extension Header

Extension headers are positioned between the header and the payload. If multiple extension headers are used, they are concatenated, as shown in Figure 2.10, making it mandatory for them to be processed in the sequence in which they are listed. Figure 2.10 specifies the sequence in which the extension headers are to be listed.

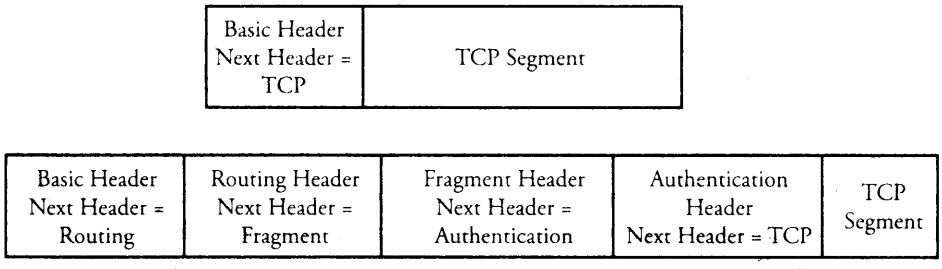


Figure 2.10 Concatenated IPv6 extension header

Packet Fragmentation

In IPv6, framentation is permitted only at the source. The result of this restriction is faster processing of packets at routers. Before transmitting a packet, a host performs a *maximum transmission unit* (MTU) discovery in the route of the packet. The minimum MTU obtained determines the packet size and thus requires the route from the host to the destination to remain steady. If this minimum value of the physical network is less than the packet size to be transmitted, the intermediate router discards the packet and sends an error message back to the source. In rare cases, the packet needs to be fragmented, and the extension header contains the fragmentation information.

2.4 Equal-Sized Packets Model: ATM

A networking model in which packets are of equal size can be constructed. Equal-sized packets, or *cells*, bring a tremendous amount of simplicity in the networking hardware, since buffering, multiplexing, and switching of cells become extremely simple. However, a disadvantage of this kind of networking is the typically high overall ratio of header to data. This issue normally arises when the message size is large and the standard size of packets is small. As discussed in Section 1.3, the dominance of headers in a network can cause delay and congestion. Here, we describe *Asynchronous Transfer Mode* technology as an example of this model.

The objective of *Asynchronous Transfer Mode* (ATM) technology is to provide a homogeneous backbone network in which all types of traffic are transported with the same small fixed-sized *cells*. One of the key advantages of ATM systems is flexible multiplexing to support multiple forms of data. ATM typically supports such *bursty* sources as FAX, coded video, and bulk data. Regardless of traffic types and the speed of sources, the traffic is converted into 53-byte ATM cells. Each cell has a 48-byte data payload and a 5-byte header. The header identifies the virtual channel to which the cell belongs.

Similar to a telephone network, ATM is a set of connection-oriented protocols, which means that a connection must be preestablished between two systems in a network before any data can be transmitted. ATM is capable of supporting and integrating data, voice, and video over one transmission medium with high bit data rate delivery services into a single network. ATM is bandwidth-on-demand networking and is scalable in bandwidth with the ability to support real multimedia applications.

The use of fixed-size cells can greatly reduce the overhead of processing ATM cells at the buffering and switching stages and hence increase the speed of routing,

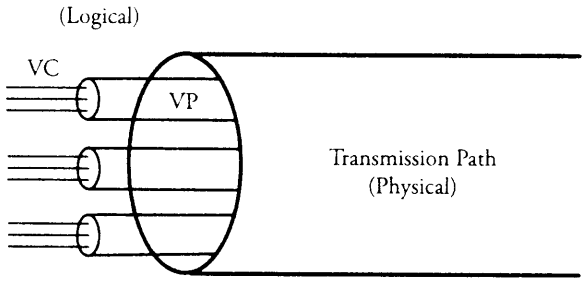


Figure 2.11 Overview of a typical ATM transmission medium

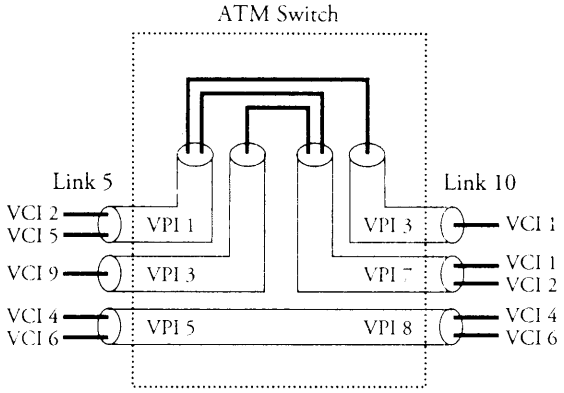
switching, and multiplexing functions. However, the great ratio of header to data makes the technology unsuitable for wide area networks and limits its applications in small networks. Like IPv6, ATM supports QoS mainly to reserve resources that guarantee specified maximum delay, minimum throughput, and maximum data loss. The QoS support allows ATM to concurrently handle all kinds of traffic.

ATM connections are identified by a *virtual channel identifier* (VCI) and a *virtual path identifier* (VPI). VCI and VPI are combined to be used in a switch to route a cell. As shown in Figure 2.11, the identity of a "physical" link is identified by two "logical" links: virtual channel (VC) and virtual path (VP). When a connection is set up, the values of these identifiers remain unchanged for the lifetime of the ATM connection.

Example. Figure 2.12 shows a routing table in an ATM switch, with routing information for all active connections passing through the switch. The routing information consists of the new VPI/VCI and new outgoing link for every incoming VC. Link 5, with VPIs 1, 3, and 5, can be switched on link 10 with VPIs 3, 7, and 8 through the ATM switch. A routing table provides the detail of the switching function. For example, a cell with VPI 3 and VCI 9 on link 5 is set to be forwarded with VPI 7 and VCI 2 on link 10.

2.4.1 ATM Protocol Structure

The ATM protocol structure is shown in Figure 2.13. The three-dimensional model includes four layers in the vertical dimension. The tightly linked layers consist of the *physical layer*, the *ATM layer*, the *ATM adaptation layer* (AAL), and *higher layers*. The physical layer includes two sublayers: the *physical* medium and *transmission convergence*. The physical medium sublayer defines the physical and electrical/optical interfaces with the transmission media on both the transmitter and the receiver. This layer also provides



Routing Table					
Incoming			Outgoing		
Link	VPI	VCI	Link	VPI	VCI
5	1	2	10	3	1
5	1	5	10	7	1
5	3	9	10	7	2
5	5	—	10	8	—

Figure 2.12 A routing table in an ATM switch

timing information and line coding. The transmission convergence sublayer provides frame adaptation and frame generation/recovery.

The ATM layer provides services, including cell multiplexing and demultiplexing, generic flow control, header cell check generation and extraction, and most important, remapping of VPIs and VCIs. The AAL layer maps higher-layer service data units, which are fragmented into fixed-size cells to be delivered over the ATM interface. In addition, this layer collects and reassembles ATM cells into service data units for transporting to higher layers. The four types of AALs support different classes of services.

1. AAL1 supports class A traffic, the required timing between a transmitter and a receiver, and the *constant bit rate* (CBR) traffic.
2. AAL2 supports class B traffic and time-sensitive—between source and sink—but *variable bit rate* (VBR) data traffic.
3. AAL3/4 supports class C or class D traffic and VBR data traffic.
4. AAL5 supports class D traffic in which VBR traffic can be transported and no timing relationship between source and sink is required.

The higher layers incorporate some of the functionality of layers 3 through 5 of the TCP/IP model. The control plane at the top of the cube shown in Figure 2.13 involves all kinds of network signaling and control. The *user plane* involves the transfer of user information, such as the flow-control and error-control mechanisms. The *management plane* provides management function and an information-exchange function between the *user plane* and the *control plane*. The management plane includes

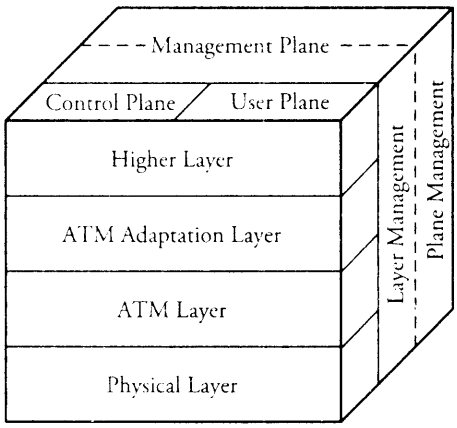


Figure 2.13 ATM protocol reference model

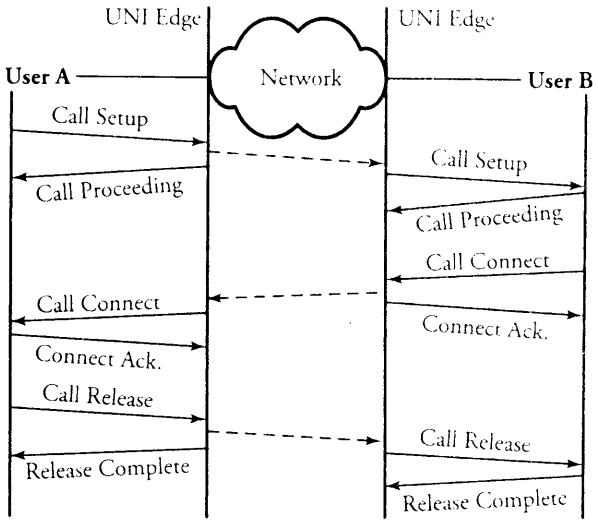
(1) plane management that performs management and coordination functions related to a system as a whole, and (2) the layer management that monitors bit error rates on a physical communications medium.

An ATM network can support both a *user-network interface* (UNI) and a *network-node interface* (NNI). A UNI is an interface connection between a terminal and an ATM switch, whereas an NNI connects two ATM switches. A summary of these two interfaces is shown in Figure 2.14. If privately owned switches are in the network, the interface between the public and private parts of the network is called the public NNI, and the connection between two private switches is known as the private NNI (P-NNI).

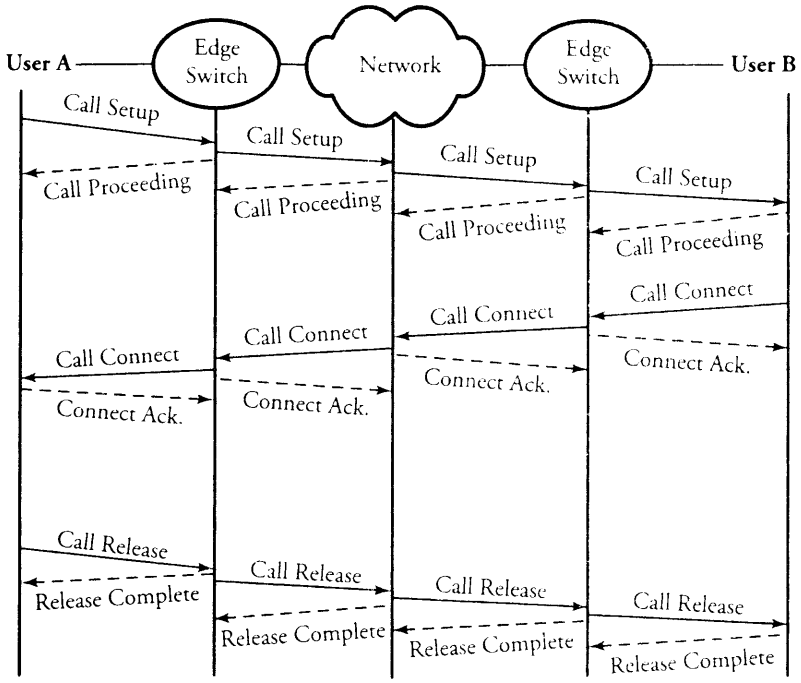
2.4.2 ATM Cell Structure

An ATM cell has two parts: a 48-byte payload and a 5-byte header, as shown in Figure 2.15. The choice of a 48-byte payload was a compromise among various design teams considering two important factors: packetization delay and transmission efficiency. (Refer back to Section 1.3 on packet size optimization.) The header consists of several fields. However, the ATM cell header has two different formats: UNI and NNI. The details of the UNI 5-byte header are as follows.

- The 4-bit *generic flow control* (GFC) field is used in UNI only for controlling local flow control. This field enables the participating equipment to regulate the flow of traffic for different grades of service. Two modes are defined for this field: the



(a)



(b)

Figure 2.14 Overview of signaling: (a) UNI format; (b) NNI format

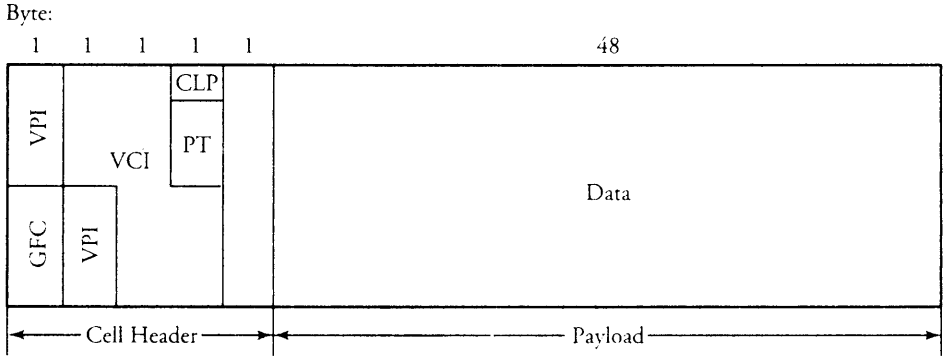


Figure 2.15 An ATM cell and its header structure

controlled GFC, to provide flow control between a user and a network, and the *uncontrolled* GFC, to indicate that the GFC function is not used.

- Together, the *virtual path identifier* and the *virtual channel identifier* represent an ATM address. A VPI identifies a group of virtual channels with the same end point. A VCI identifies a virtual channel within a virtual path.
- The 3-bit *payload type* field is used to indicate the type of data located in the payload portion of the cell: for example, 000, the current cell is a data cell and no congestion is reported; 010, this is a user data cell, and congestion is experienced. The payload could be congestion information, network management message, signaling information, or other forms of data.
- The 1-bit *cell-loss priority* (CLP) field is used to prioritize cells. When congestion occurs, cells with CLP set to 1 (considered low priority) are discarded first. If the bit is set to 0, the cell gets higher priority and should be discarded only if it could not be delivered.
- The 8-bit *header error control* (HEC) field is used for error checking. HEC functions include correcting single-bit errors and detecting multiple-bit errors.

The main difference between NNI and UNI formats is that the 4 bits used for the GFC field in the UNI cell header are added to the VPI field in the NNI cell header. Thus, for NNI, VPI is 12 bits, allowing for more VPs to be supported within the network. VCI is 16 bits for both cases of UNI and NNI. The values of VPI and VCI have local significance only with a transmission link. Each switching node maps an incoming VPI/VCI to an outgoing VPI/VCI, based on the connection setup or routing table, as shown in Figure 2.12.

HEC works like other checking methods. First, the transmitter calculates the HEC field value, and the receiver side runs an algorithm consisting of two modes of operation. At initialization step, this field starts with an error-correction mode. If a single-bit error is detected in the header, the error-correction algorithm identifies the error bit and then corrects it. If a multibit error is detected, the mode moves to detection mode; errors are discarded but not corrected. Error-detection mode remains whenever cells are received in error, moving back to correction mode only when cells are received without error.

2.5 Summary

This chapter covered a tremendous amount of fundamental networking protocol material. We presented the basic structure of the Internet network protocols and an overview of the TCP/IP layered architecture. This architectural model provides a communication service for peers running on different machines and exchanging messages.

We also covered the basics of protocol layers: the *network layer* and the structure of IPv4 and IPv6. IP addressing is further subdivided as either *classful* or *classless*. Classless addressing is more practical for managing routing tables. Finally, we compared the equal-sized packet networking environment to IP networks. Although packet multiplexing is easy, the traffic management is quite challenging.

The next chapter presents another fundamental discussion in networking. Chapter 3 focuses on the fundamental operations of networking devices.

2.6 Exercises

1. Specify the class of address and the subnet ID for the following cases:
   (a) A packet with IP address 127.156.28.31 using mask pattern 255.255.255.0
   (b) A packet with IP address 150.156.23.14 using mask pattern 255.255.255.128
   (c) A packet with IP address 150.18.23.101 using mask pattern 255.255.255.128
2. Specify the class of address and the subnet ID for the following cases:
   (a) A packet with IP address 173.168.28.45 using mask pattern 255.255.255.0
   (b) A packet with IP address 188.145.23.1 using mask pattern 255.255.255.128
   (c) A packet with IP address 139.189.91.190 using mask pattern 255.255.255.128
3. Apply CIDR aggregation on the following IP addresses: 150.97.28.0/24, 150.97.29.0/24, and 150.97.30.0/24.

4. Apply CIDR aggregation on the following IP addresses: 141.33.11.0/22, 141.33 .12.0/22, and 141.33.13.0/22.
5. Use the subnet mask 255.255.254.0 on the following IP addresses, and then convert them to CIDR forms:
   (a) 191.168.6.0
   (b) 173.168.28.45
   (c) 139.189.91.190
6. A packet with the destination IP address 180.19.18.3 arrives at a router. The router uses CIDR protocols, and its table contains three entries referring to the following connected networks: 180.19.0.0/18, 180.19.3.0/22, and 180.19.16.0 /20, respectively.
   (a) From the information in the table, identify the exact network ID of each network in binary form.
   (b) Find the right entry that is a match with the packet.
7. Part of a networking infrastructure consists of three routers R1, R2, and R3 and six networks N1 through N6, as shown in Figure 2.16. All address entries of each router are also given as seen in the figure. A packet with the destination IP address 195.25.17.3 arrives at router R1:
   (a) Find the exact network ID field of each network in binary form.
   (b) Find the destination network for packet (proof needed).
   (c) Specify how many hosts can be addressed in network N1.

R1 Table Entry	Port
195.25.0.0/21	L11
195.25.16.0/20	L12
195.25.8.0/22	L13
135.11.2.0/22	L14

R2 Table Entry	Port
195.25.24.0/19	L21
195.25.16.0/20	L22
195.25.8.0/22	L23

R3 Table Entry	Port
111.5.0.0/21	L31
Else	L32
195.25.16.0/20	L33

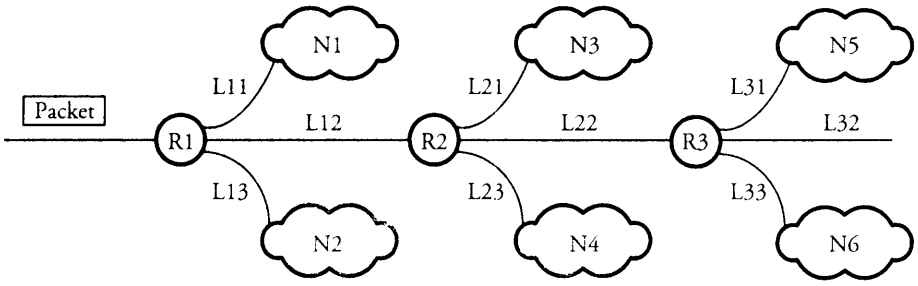


Figure 2.16 Exercise 7 network example

8. Consider an estimated population of 620 million people.
   (a) What is the maximum number of IP addresses that can be assigned per person using IPv4?
   (b) Design an appropriate CIDR to deliver the addressing in part (a).
   (c) What is the maximum number of IP addresses that can be assigned per person using IPv6?
9. For each of the following IPv6 addresses, give an abbreviated form and then convert the result to a binary form:
   (a) 1111:2A52:A123:0111:73C2:A123:56F4:1B3C
   (b) 2532:0000:0000:0000:FB58:909A:ABCD:0010
   (c) 2222:3333:AB01:1010:CD78:290B:0000:1111
10. Research why IPv6 allows fragmentation only at the source.
11. Suppose that virtual paths are set up between every pair of nodes in an ATM network. Explain why connection setup can be greatly simplified in this case.
12. Suppose that the ATM network concept is generalized so that packets can be variable in length. What features of ATM networking are retained? What features are lost?

CHAPTER 3

Networking Devices

This chapter focuses on networking devices. Familiarity with networking hardware devices is essential for understanding how a local area or a wide area network operates. This chapter covers the following aspects of network component functionality.

- *Multiplexers*
- *Modems and Internet access devices*
- *Switching and routing devices*
- *Router structure*

The three main categories of networking devices are *multiplexers*, *modems*, and *switching devices*. We start with the architecture of *multiplexers*, introducing types of multiplexers and some useful analytical methods of multiplexers. Next, we consider networking modems for accessing the Internet from remote and residential areas. Then, we analyze switching devices, such as *repeaters*, *bridges*, and *routers*, by which packets are switched from one path to another.

3.1 Multiplexers

Multiplexers are used in a network for maximum transmission capacity of a high-bandwidth line. Regardless of the type of multiplexer, multiplexing is a technique that allows many communication sources to transmit data over a single physical line.

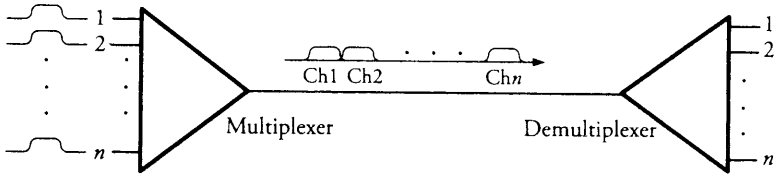


Figure 3.1 A frequency-division multiplexer (FDM) with n inputs

Multiplexing schemes can be divided into three basic categories: *frequency-division multiplexing*, *wavelength-division multiplexing*, and *time-division multiplexing*.

3.1.1 Frequency-Division Multiplexing (FDM)

In *frequency-division multiplexing* (FDM), the frequency spectrum is divided into frequency bands, or *channels*, in which each user can be assigned a band. Figure 3.1 shows how n frequency channels are multiplexed using FDM. When many channels are multiplexed together, a certain guard band is allocated to keep the channels well separated.

To implement a multiplexer, the original frequencies at any of n inputs of the multiplexer are raised, each by a different constant amount. Then, the n new frequency bands are combined to let no two channels occupy the same portion of the spectrum. Despite the guard bands between the channels, any two adjacent channels have some overlap because channel spectra do not have sharp edges. This overlap normally creates spike noise at the edge of each channel. FDM is normally used over copper wires or microwave channels and is suitable for analog circuitry.

3.1.2 Wavelength-Division Multiplexing (WDM)

Wavelength-division multiplexing (WDM) is fundamentally the same as FDM, as depicted in Figure 3.2. WDM was invented as a variation of frequency-division multiplexing and is basically a multiplexing method of different wavelengths instead of frequencies. In the figure, n optical fibers come together at an optical multiplexer, each with its energy present at a different wavelength. The n optic lines are combined onto a single shared link for transmission to a distant destination. At the demultiplexer, each frame, including n channels, is split up over as many optical fibers as there were on the input side. At each output of the demultiplexer, a tuned filter refines the desired signal at the tuned wavelength, and thus all other wavelengths are bypassed.

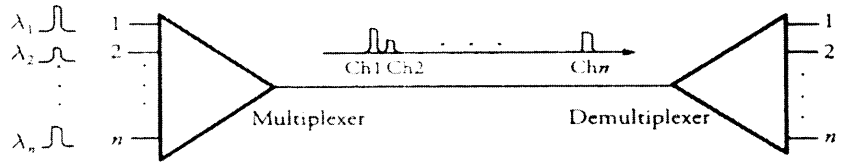


Figure 3.2 A wavelength-division multiplexer (WDM) with n inputs

The main issue of WDM compared with FDM is that an optical system using a diffraction grating is completely passive and thus highly reliable. With higher-speed variations of WDM, the number of channels is very large, and the wavelengths are as close as 0.1 nm. Such systems are referred to as DWDM (dense WDM). Chapter 14 presents much more detail about WDM and its applications.

Example. Consider a practical multiplexing system with 100 channels, each at rate 10 Gb/s. Compute the number of full-length movies per second that can be transferred with this WDM system.

Solution. The total bit rate of this WDM is 100 × 10, or 1,000 Gb/s. Since a movie (MPEG-2 technology) requires 32 Gb/s bandwidth, the system can carry approximately 31 full-length movies per second.

3.1.3 Time-Division Multiplexing

With a *time-division multiplexing* (TDM), users take turns in a predefined fashion, each one periodically getting the entire bandwidth for a portion of the total scanning time. Given n inputs, time is divided into frames, and each frame is further subdivided into time slots, or channels. Each channel is allocated to one input. (See Figure 3.3.) This type of multiplexing can be used only for digital data. Packets arrive on n lines, and the multiplexer scans them, forming a frame with n channels on its outgoing link. In practice, packet size is variable. Thus, to multiplex variable-sized packets, additional hardware is needed for efficient scanning and synchronization. TDM can be either *synchronous* or *statistical*.

Synchronous TDFM

In *synchronous* TDM, the multiplexer scans all lines without exception. The scanning time for each line is preallocated; as long as this time for a particular line is not altered by the system control, the scanner should stay on that line, whether or not there is

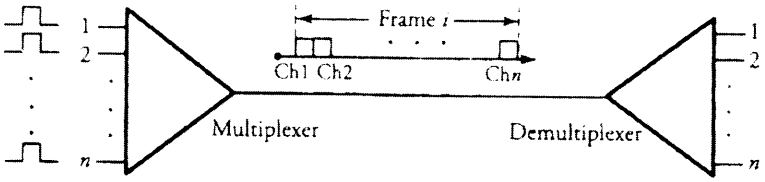


Figure 3.3 A time-division multiplexer with n inputs

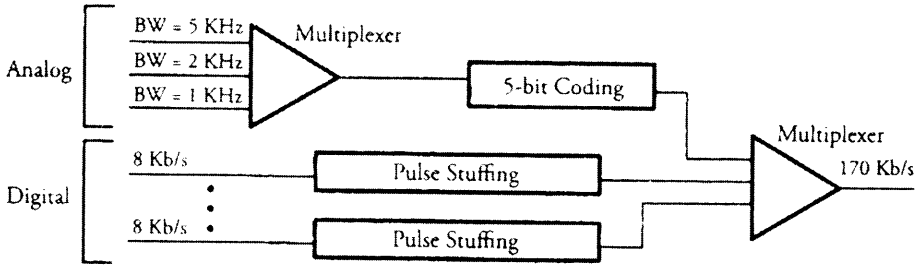


Figure 3.4 Integrated multiplexing on analog and digital signals

data for scanning within that time slot. Therefore, a synchronous multiplexer does not operate efficiently, though its complexity stays low.

Once a synchronous multiplexer is programmed to produce same-sized frames, the lack of data in any channel potentially creates changes to average bit rate on the ongoing link. In addition to this issues, the bit rates of analog and digital data being combined in a multiplexer sometimes need to be synchronized. In such cases, dummy pulses can be added to an input line with bit rate shortcoming. This technique of bit-rate synchronization is called *pulse stuffing*.

Example. Consider the integration of three analog sources and four identical digital sources through a time-division multiplexer, as shown in Figure 3.4. We would like to use the entire 170 Kb/s maximum capacity of this multiplexer. The analog lines with bandwidths 5 KHz, 2 KHz, and 1 KHz, respectively, are sampled, multiplexed, quantized, and 5-bit encoded. The digital lines are multiplexed, and each carries 8 Kb/s. Find the pulse-stuffing rate.

Solution. Each analog input is sampled by a frequency two times greater than its corresponding bandwidth according to the Nyquist sampling rule. Therefore, we have $5 \times 2 + 2 \times 2 + 1 \times 2 = 16$ K samples per second. Once it is encoded, we get a

total 16,000 × 5 = 80 Kb/s on analog lines. The total share of digital lines is then is 170 – 80 = 90 Kb/s. Since each digital line must generate 22.5 Kb/s while the actual rate of 8 Kb/s exists, each digital line must add a difference of 14.5 Kb/s pulse stuffing in order to balance the ultimate bit rate of multiplexer bit rate.

Consider a multiplexer with n available channels. If the number of requesting input sources, m, is greater than n channels, the multiplexer typically reacts by *blocking* where unassigned sources are not transmitted and therefore remain inactive. Let t_a and t_d be the two mean times during which a given input becomes active and idle, respectively. Assume that the transmission line has n channels available, where $m > n$.

If more than n inputs are active, we can choose only n out of m active sources and permanently block others. If one of the n chosen channels goes to idle, we can give service to one of the other requests. Typically, blocking is used when channels must be held for long time periods. The traditional telephone system is one example; channels are assigned at the start of a call, and other callers are blocked if no channel is available. Assuming that values of t_a and t_d are random and are exponentially distributed, the probability that a source is active, ρ, can be obtained by

$$\rho = \frac{t_a}{t_d + t_a}. \tag{3.1}$$

Let p_j be the probability that j out of m inputs are active; for $1 \le j \le m$,

$$p_j = \binom{m}{j} \rho^j (1-\rho)^{m-j} \tag{3.2}$$

Thus, the probability that j out of n *channels* on the transmission line are in use, P_j, can be expressed by normalizing P_j over n *inputs* as $P_j = p_j / \sum_{i=1}^{n} p_i$, or

$$P_j = \frac{\binom{m}{j} \rho^j (1-\rho)^{m-j}}{\sum_{i=0}^{n} \binom{m}{i} \rho^i (1-\rho)^{m-i}} \quad \text{for } 1 \le j \le n$$

$$= \frac{\binom{m}{j} \left(\frac{\rho}{1-\rho}\right)^j}{\sum_{i=0}^{n} \binom{m}{i} \left(\frac{\rho}{1-\rho}\right)^i} \text{ for } 1 \le j \le n. \tag{3.3}$$

The reason behind the normalization is that $\sum_{i=1}^{n} p_i$ can never be equal to 1, since $n \leq m$; according to the rule of total probability, we can say only $\sum_{i=1}^{m} p_i = 1$. The blocking probability of such multiplexers, P_n, can be calculated by simply letting j be n in Equation (3.3), denoting that all n channels are occupied. If we also substitute $\rho = t_d/(t_d + t_a)$, the blocking probability can be obtained when $j = n$:

$$P_n = \frac{\binom{m}{n}\left(\frac{t_a}{t_d}\right)^n}{\sum_{i=0}^{n}\binom{m}{i}\left(\frac{t_a}{t_d}\right)^i}. \tag{3.4}$$

The preceding discussion can be concluded by using basic probability theory to find the average number of busy channels or the expected number of busy channels as

$$E[C] = \sum_{j=0}^{n} jP_j. \tag{3.5}$$

Example. A 12-input TDM becomes an average of 2 μs active and 1 μs inactive on each input line. Frames can contain only five channels. Find the probability that a source is active, ρ, and probability that five channels are in use.

Solution. We know that $m = 12$ and $n = 5$; since $t_a = 2$ μs, and $t_d = 1$ μs:

$$\rho = \frac{t_a}{t_d + t_a} = 0.66,$$

and the probability that all five channels are in use is

$$P_{n=5} = \frac{\binom{12}{5}\left(\frac{2}{1}\right)^5}{\sum_{i=0}^{5}\binom{12}{i}\left(\frac{2}{1}\right)^i} = 0.72$$

Statistical TDM

In *statistical* TDM, a frame's time slots are dynamically allocated, based on demand. This method removes all the empty slots on a frame and makes the multiplexer operate more efficiently. Meanwhile, the trade-off of such a technique is the requirement that additional overhead be attached to each outgoing channel. This additional data is needed because each channel must carry information about which line it belonged to. As a result, in the statistical multiplexer, the frame length is variable not only because of different channel sizes but also because of the possible absence of some channels.

Consider a multiplexer with n available channels. If the number of requesting input sources, m, is greater than n channels, the multiplexer typically reacts by *clipping*, whereby unassigned sources are partially transmitted, or clipped. Designing multiplexers with $m > n$ stems from the fact that all m sources may not be practically active simultaneously. In some multiplexing systems, *clipping* is exploited especially for applications in which sources do not transmit continuously, and the best usage of the multiplex output line is targeted. By assigning active sources to channels dynamically, the multiplex output can achieve more efficient channel usage. For example, some satellite or microwave systems detect information energy and assign a user to a channel only when a user is active.

Consider the same definitions as presented in the blocking case for t_a, t_d, m, n ($m > n$), and ρ. Similarly, assume that values of t_a and t_d are random and are exponentially distributed. Assume also that the outgoing transmission line has n channels available but the multiplexer has m inputs, where $m > n$. If more than n inputs are active, we can dynamically choose n out of m active sources and temporarily block other sources. With temporary blocking, the source is forced to lose, or clip, data for a short period of time, but the source may return to a scanning scenario if a channel becomes free. This method maximizes the use of the common transmission line and offers a method of using the multiplexer bandwidth in silence mode. Note that the amount of data lost from each source depends on t_a, t_d, m, and n. Similar to the blocking case, ρ can be defined by $\rho = t_a/(t_d + t_a)$, and the probability that exactly k out of m sources are active is

$$P_k = \binom{m}{k} (\rho)^k (1 - \rho)^{m-k}. \tag{3.6}$$

Therefore, P_C, the clipping probability, or the probability that an idle source finds at least n sources busy at the time it becomes active, can be obtained by considering all m sources beyond n active sources minus 1 (the examining source):

$$P_C = \sum_{i=n}^{m-1} \binom{(m-1)}{i} \rho^i (1 - \rho)^{m-1-i}. \tag{3.7}$$

Clearly, the average number of used channels is $\sum_{i=1}^{n} i P_i$. The average number of busy channels can then be derived by

$$E[C] = \sum_{i=1}^{n} i P_i + n \sum_{i=n+1}^{m} P_i. \tag{3.8}$$

Example. For a multiplexer with ten inputs, six channels, and $\rho = 0.67$, find the clipping probability.

Solution. Since $m = 10$, $n = 6$, $P_c \approx 0.65$

3.2 Modems and Internet Access Devices

Users access the Internet from residential areas primarily through *modems*. A modem (*modulation/demodulation unit*) is a device that converts the digital data to a modulated form of signal that takes less bandwidth. A modem is required to create an appropriate digital signal to access the Internet. A user can access the Internet by using either the existing telephone link infrastructure or the existing cable TV infrastructure. As a result, an Internet user has several choices of modems. Two commonly used ones are the digital subscriber line (DSL) *modem* and the *cable modem* (Figure 3.5). A DSL company uses the existing twisted-pair copper lines to provide Internet access; a cable TV company uses optical fibers and coaxial cables to provide that access.

3.2.1 Line Coding Methods

Before processing a raw signal for modulation, a *line coding process* is performed on binary signals for digital transmission. With line coding, a binary information sequence is converted into a digital code. This process is required to maximize bit rate in digital transmission. Encoding process is essential to also recover the bit timing information from the digital signal so that the receiving sample clock can synchronize with the

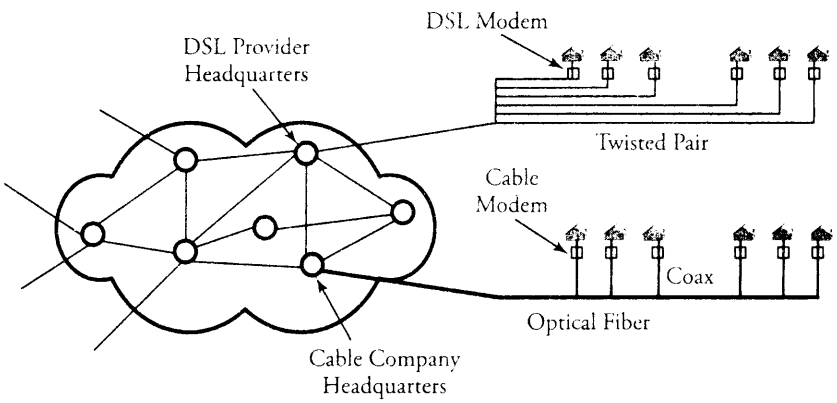


Figure 3.5 The choice of DSL modem or cable modem in residential areas for Internet access

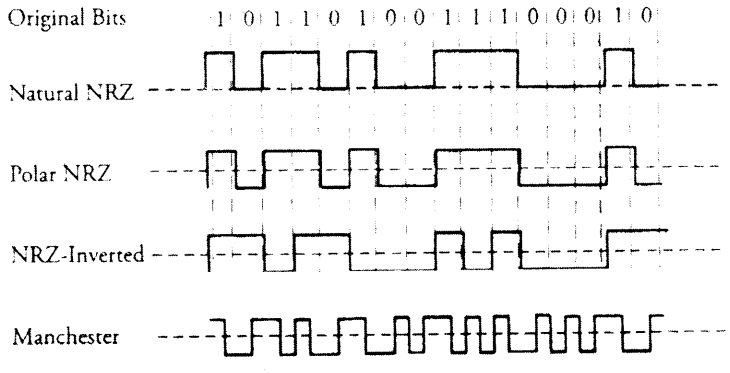


Figure 3.6 Typical line coding techniques for computer communications.

transmitting clock. The timing synchronization is especially crucial in the performance of LANs. Other reasons for line coding are reduction in transmitted power, and removal of DC voltage from transmission lines.

Typically, the cost and complexity of a line encoder is the main factor in the selection of encoder for a given application. Figure 3.6 shows several practical line coding techniques for computer communications. Encoded signals are produced by the line codes for the binary sequence 1011 0100 1110 0010. The simplest form of line coding is the *natural nonreturn-to-zero* (NRZ) where a binary 1 is represented by a $+V$ voltage level, and a 0 is represented by a 0 voltage. The average transmitted power in this method is $(1/2)V^2 + (1/2)0^2 = V^2/2$. This method creates an average of $V/2$ DC voltage on the transmission lines which is not popular for LAN systems.

A more power-efficient line coding method is known as *polar NRZ*. In this method, a binary 1 is mapped to $+V/2$ and a binary 0 is represented by $-V/2$. The average power is then given by $(1/2)(+V/2)^2 + (1/2)(-V/2)^2 = V^2/4$. On average, this method has no DC component and is suitable in most networking applications.

A problem with natural and polar NRZ coding methods is that a polarity error can cause all 1s to be delivered as a weak 1 or even a 0. As a solution to this problem the *NRZ–inverted* method is introduced. With the NRZ-inverted coding, the binary information is mapped into transitions at the beginning of each interval so that a binary 1 is converted to a transition at the beginning of a bit time and a 0 having no transition, and the signal remains constant during the actual bit time. Notice that, errors in this method of encoding occur in pairs. This means that any error in one bit time generates a wrong basis for the next time leading to a new error in the next bit.

From the frequency stand-point, both the natural NRZ and NRZ-inverted methods produce spectrum starting from very low frequencies close to zero due to existence of either DC components or less frequent transitions of 0s to 1s or vice versa. Although, the bipolar NRZ has a better spectrum distribution in this sense, the immunity to noise can still be an issue for all three types of NRZ coding. Low frequencies can also be a bottleneck in some communication systems, as telephone transmission systems, do not pass the frequencies below 200 Hz. To overcome this issue, the *Manchester encoding method* is introduced.

With the Manchester encoding method, a binary 1 is represented by a 1 plus a transition to 0 and then a 0; and a binary 0 is represented by a 0 plus a transition to 1 and then a 1. A great feature of the Manchester encoding is that it is self-clocking. The fact that the each binary bit contains a transition at the middle of its timing makes the timing recovery very easy. From the frequency standpoint, the bit rate is doubled compared with NRZ methods that significantly enhances the shape of its spectrum where lower frequencies are shifted up. This method of line encoding is suitable for LANs especially for Gigabit Ethernets to be discussed in Chapter 5.

3.2.2 Digital Modulation Techniques

In order to reduce the bandwidth of digital signals, *digital modulation technique* is required before any transmission. In a modulation process, the amplitude, phase, or frequency of a carrier signal varies with the variations of a digital information signal. Depending on the nature and objective of a modem, the following schemes can be used:

- *Amplitude shift keying* (ASK)
- *Frequency shift keying* (FSK)
- *Phase shift keying* (PSK)
- *Quadrature amplitude modulation* (QAM)

Knowing the types of modulation techniques, we now study the two types of modems for accessing the Internet: digital subscriber line (DSL) modems and cable modems.

Figure 3.7 shows the implementation of ASK, FSK, PSK, and QAM systems. In an ASK system, incoming data, $D(t)$, containing binary 0s and 1s, is modulated over a constant-frequency and constant amplitude sinusoidal carrier signal, $\cos 2\pi f t$, where f is the frequency of the carrier. The resulting modulated signal is represented by a cosine with the same frequency f where a binary 1 is present, and no signal when a binary 0 is present. In other words, if we multiply our binary data by a constant cosine,

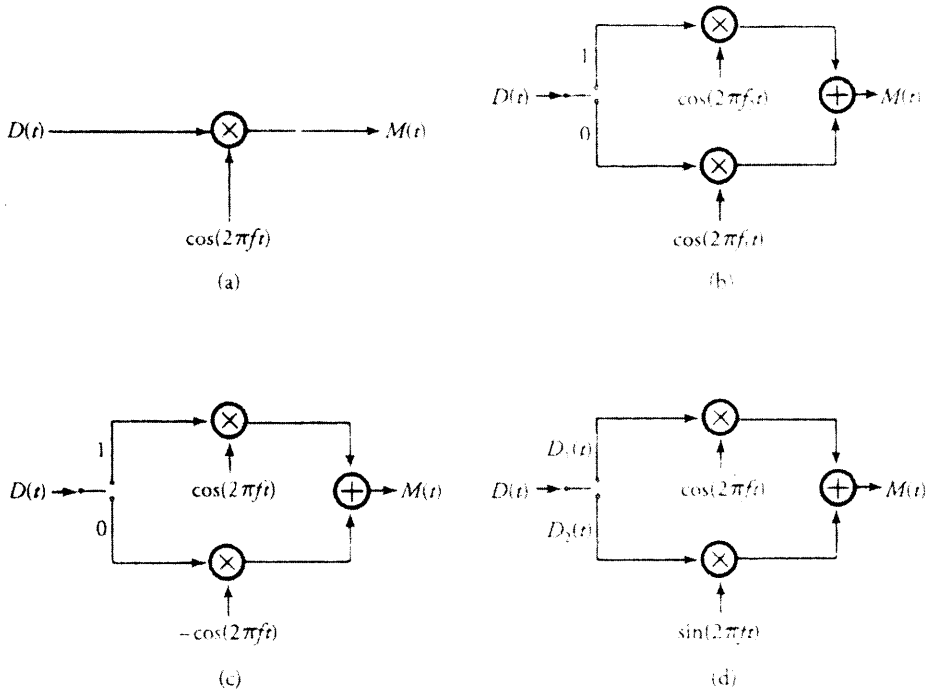


Figure 3.7 The four modulation techniques: (a) ASK, (b) FSK, (c) PSK, and (d) QAM

we obtain the ASK-modulated version of our data, $M(t) = D(t)\cos(2\pi f t)$. At the receiver, the ASK demodulator only needs to determine the presence or absence of the sinusoid in a given time interval in order to detect the original data. In practice, $D(t)$ can be extracted at the receiver if $M(t)$ is multiplied by the same carrier signal but with doubled amplitude, $2\cos(2\pi f t)$, as follows:

$$[D(t)\cos(2\pi f t)][2\cos(2\pi f t)] = 2D(t)\cos^2(2\pi f t) = D(t)[1 + \cos(4\pi f t)] \qquad (3.9)$$

Note that $\cos(4\pi f t)$ component can be easily filtered out. The FSK scheme is a bit more complicated compared to the ASK scheme. In a FSK modulator, there are two different sinusoidal carriers: f_1 to represent a binary 1 and f_2 to represent binary 0. Therefore, FSK and ASK are similar in the sense that we multiply binary data by a constant sinusoid to obtain the modulated version of data, and they are different in the sense that there is a sinusoid when 0 appears. Clearly, in the FSK system, the frequency

of the carrier varies according to the information such that we have $\cos(2\pi f_1 t)$ instead of a binary 1 and $\cos(2\pi f_2 t)$ instead of a binary 0.

In a PSK system, the phase of the sinusoidal carrier signal changes according to the information sequence as shown in Figure 3.7 (c). A binary 1 is represented by $\cos(2\pi f t)$ and a binary 0 by $\cos(2\pi f t + \pi)$. Similarly, we can rephrase the definition of a PSK system: in a PSK modulator we multiply the sinusoidal signal by +1 when the information is a 1, and by -1 when a 0 is present.

QAM is another modulator in which we split the original information stream into two equal sequences, $D_1(t)$ and $D_2(t)$, consisting of the odd and even symbols, respectively. Each sequence has a rate of s symbols/second and the number of bits per symbol is typically constant. As shown in Figure 3.7 (d), we take $D_1(t)$ and produce a modulated signal by multiplying it by $\cos(2\pi f t)$ for a T-second interval. Similarly, we take $D_2(t)$ and produce a modulated signal by multiplying it by $\sin(2\pi f t)$ for a T-second interval. The first component $D_1(t)$ is known as *in-phase component*, and the second component $D_2(t)$ is known as *quadrature-phase component*. Therefore, at the output of the modulator we have:

$$M(t) = D_1(t)\cos(2\pi f t) + D_2(t)\sin(2\pi f t) \tag{3.10}$$

The QAM scheme can be realized as the simultaneous modulation of the amplitude and the phase of a carrier signal as Equation (3.10) can be rearranged as:

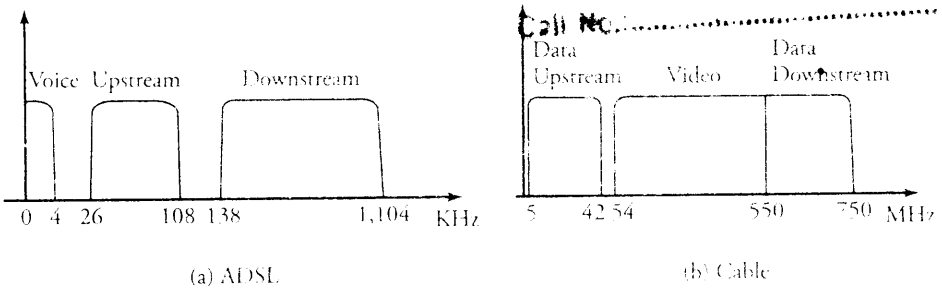
$$M(t) = \sqrt{D_1^2(t) + D_2^2(t)}\cos\left(2\pi f t + \tan^{-1}\frac{D_2(t)}{D_1(t)}\right) \tag{3.11}$$

Similar to what was explained for the ASK system, the original data, $D(t)$, can be extracted at the receiver if $M(t)$ is multiplied by the same carrier signal but with doubled amplitude. However, since $M(t)$ has two terms in this case, $D_1(t)\cos(2\pi f t)$ must be multiplied by $2\cos(2\pi f t)$, and $D_2(t)\sin(2\pi f t)$ must be multiplied by $2\sin(2\pi f t)$.

3.2.3 Digital Subscriber Line (DSL) Modems

Digital subscriber line (DSL) technology is a convenient option for home users to access the Internet. This technology offers various versions of DSL technology: ADSL, VDSL, HDSL, and SDSL, or, in general, xDSL.

Among the xDSL types, *asymmetric* DSL (ADSL) is popular and is designed for residential users. A modem is designed to be connected to telephone links. These links are capable of handling bandwidths up to 1.1 MHz. Out of this bandwidth, only 4 KHz are used for a phone conversation. Consequently, the remaining bandwidth can become available to be allocated to data communications. However, other factors, such



(a) ADSL (b) Cable

Figure 3.8 The frequency bands of (a) ADSL modems and (b) cable modems

as the distance between a user and a switching office, and the condition and size of the link might restrict this remaining bandwidth from being completely available.

The details of spectrum division for an ADSL modem are shown in Figure 3.8 (a). The standard modulation technique for ADSL is QAM. The available bandwidth of 1.1 MHz is divided into 256 channels, each using a bandwidth of approximately 4.312 KHz. Voice communication uses channel 0. Channels 1–5 remain idle and together act as a guard band between voice and data communication. Because data communication bandwidth is split into two bandwidths—*upstream* for communications from the user to the Internet and *downstream* for communications from the Internet to the user—the technique is said to be asymmetric. Channels 6–30 are allocated to the upstream bandwidth, with 1 channel dedicated to control and 24 channels assigned to data transfer. Thus, 24 channels with QAM offer 24 × 4 KHz × 15 = 1.44 Mb/s bandwidth in the upstream direction, as QAM requires 15-bit encoding. Channels 31–255 are assigned to the downstream bandwidth, with 1 channel for control and the remaining 224 channels for data. With QAM, up to 224 × 4 KHz × 15 = 13.4 Mb/s is achieved for the downstream bandwidth.

Another type of DSL technique is *symmetric digital subscriber line* (SDSL). This technique divides the available bandwidth equally between downstream and upstream data transfer. *High-bit-rate digital subscriber line* (HDSL), as another option, was designed to compete with T-1 lines (1.544 Mb/s). A T-1 line is susceptible to attenuation at high frequencies and thus limits the length of line to 1 km. HDSL uses two twisted-pair wires and 2B1Q, an encoding technique less susceptible to attenuation. As a result, HDSL can achieve a data rate of 2 Mb/s without needing repeaters for up to 3.6 km. *Very high bit-rate digital subscriber line* (VDSL) is similar to ADSL but uses coaxial or fiber-optic cable for a bit rate of 50 Mb/s to 55 Mb/s downstream and 1.5 Mb/s to 2.5 Mb/s upstream data transfer.

3.2.4 Cable Modems

As mentioned in the previous section, the DSL modem uses the existing twisted-pair telephone cables for providing residential users with access to the Internet, as shown in Figure 3.5. This type of cable clearly has limited bandwidth and is susceptible to errors. An alternative is to use the cable TV network for Internet access. A cable company lays out very high-speed backbone optical fiber cables all the way to the residential buildings, each of which can then be connected to the optical infrastructure for TV, radio, and the Internet through either a coaxial (coax) cable or optical fiber, depending on its demand and budget. This network is called *hybrid fiber-coaxial* (HFC). Video signals are transmitted downstream from headquarters to users. Communication in an HFC cable TV network is bidirectional. The cable company divides the bandwidth into video/radio, downstream data, and upstream data. Coaxial cables can carry signals up to 750 MHz.

Details of the spectrum division for a cable modem are shown in Figure 3.8 (b). About 500 MHz of this bandwidth is assigned to TV channels. As the bandwidth of each TV channel is 6 MHz, the assigned bandwidth can accommodate more than 80 channels. Some technical methods allow this number to increase to 180 TV channels. About 200 MHz of the coax bandwidth, from 550 MHz to 750 MHz, is allocated to the downstream data transfer: from the Internet side to a user. This bandwidth is also divided to about 33 channels, each with 6 MHz bandwidth. The cable modem uses the 64-QAM or 256-QAM modulation technique for the downstream data transfer. These modulation techniques use 5-bit encoding, so the bandwidth of the downstream data channel can be $5\ \text{b/Hz} \times 6 = 30\ \text{Mb/s}$.

The upstream data premises communicate to the Internet and occupy 37 MHz, from 5 MHz to 42 MHz, including 6 MHz-wide channels. The upstream data is modulated using the QPSK (Quadrature PSK) technique, which is less susceptible to noise in the lower frequency range. Using 2 bits per Hz offers the downstream data rate at $2\ \text{b/Hz} \times 6\ \text{MHz} = 12\ \text{Mb/s}$. The protocol for upstream communication is summarized as follows.

Begin Upstream Communication Protocol

1. The cable modem checks the downstream channels to determine whether any packet periodically sent by the cable company seeks any new modems attached to the cable.
2. The cable company sends a packet to the cable modem, specifying the modem allocated downstream.

3. The cable modem sends a packet to the cable company, requesting the Internet address.
4. The modem and the cable company go through a handshaking process and exchange some packets.
5. The cable company delivers an identifier to the modem.
6. Internet access in the allocated upstream channel bandwidth can then be granted to the modem in the allocated upstream channel. ■

Note that a user needs to use one 6-MHz-wide channel assigned by the cable company. To better use the optical fiber bandwidth, all users of the same neighborhood need to timeshare an FDM channel. Therefore, a user must contend for a channel with others and must wait if the channel is unavailable.

3.3 Switching and Routing Devices

Switching devices are categorized by their complexity, as follows:

- *Layers 1 and 2 switches* are typically simple. For example, *repeaters* and *hubs* are known as layer 1 switches; *bridges*, as layer 2 switches.
- *Layer 3 or higher switches* are complex. Routers, for example, are layer 3 switches.

Figure 3.9 depicts interconnections and switching functions at layer 1, layer 2, layer 3, and upper layers of the five layer protocol stack.

3.3.1 Repeaters, Hubs, and Bridges

Repeaters and *hubs*, the simplest switching devices, are designed primarily to interconnect very small LAN units without any involvement in the complex routing processes. Chapter 5 provides several examples of repeaters and hubs.

Repeaters are used to connect two segments of a LAN. A repeater's essential function, *signal regeneration*, differentiates it from a piece of cable. Signal regeneration is needed when the LAN length is extended. When a LAN is extended, bits can be corrupted and decayed. As shown in Figure 3.9, two LANs are interconnected using a repeater at the physical layer (layer 1). The repeater assumes that the connecting LANs have the same protocol and simply accepts bits from one LAN and transmits them on to the other LANs.

A *hub* is another simple device and is used to provide connections among multiple users in layer 1 of a protocol stack. A hub is similar to the repeater but connects several pieces of a LAN. In fact, a hub is a multipoint repeater. To perform this type of

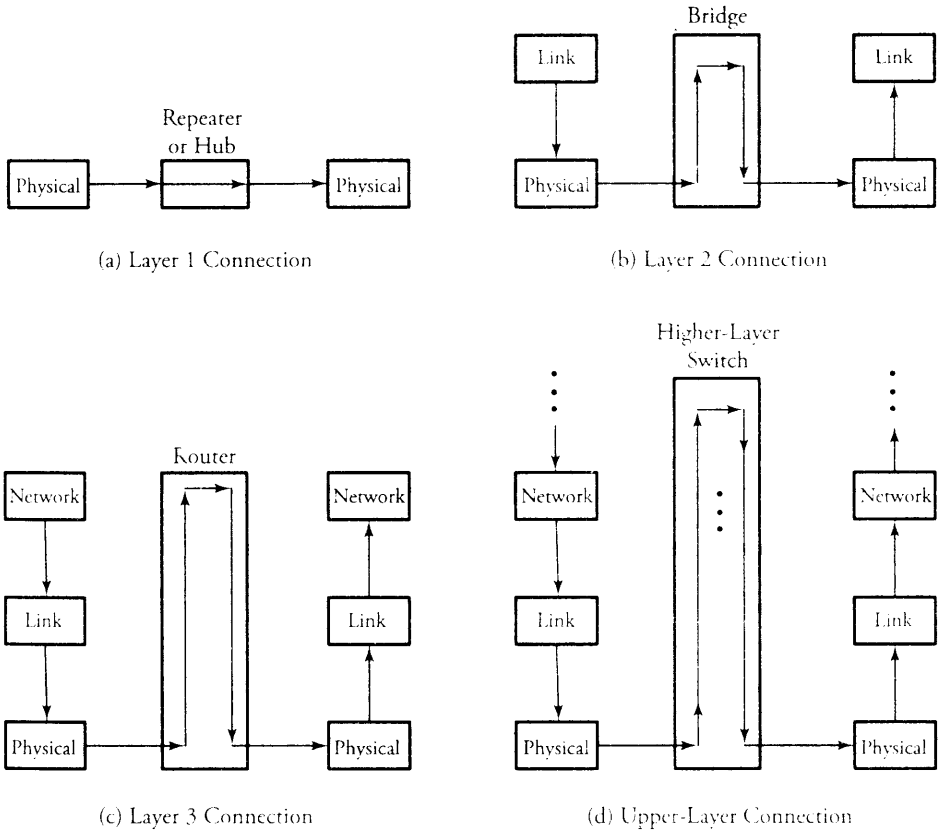


Figure 3.9 Connections in different layers of protocols

connection, a copying element is installed to copy and forward a packet or a frame to all connected users. As a typical repeater, a hub has regeneration capability to strengthen incoming data to prevent any decay.

A *bridge* is a switch that connects two pieces of a LAN or two LANs, especially those operating in layer 2 of the protocol stack. However, a bridge can also be used in layer 1 for signal regeneration. As it operates at layer 2, a bridge does more than simply extend the range of the network. A bridge checks the physical address of any destination user and enhances the efficiency of networks by facilitating simultaneous transmissions within multiple LANs. Unlike repeaters, which transmit a frame to all associated users within a LAN, a bridge does not forward a frame to all LAN users and

thus can isolate traffic between two LANs. Bridges can make decisions about where to forward frames. Bridges perform data link functions, such as forwarding, formatting, and error detection.

Bridges can forward only one frame at a time in a store-and-forward fashion. However, sophisticated layer 2 switches can forward multiple frames simultaneously through multiple parallel data paths. Layer 2 switches can also operate as cut-through devices, significantly enhancing communication performance. These advantages have made switches more popular than typical bridges. LAN bridges or other intelligent layer 2 switches are devices that use layer 2 data to forward or filter traffic.

3.3.2 Routers and Higher-Layer Switches

A *router* is a layer 3 switch that connects other routing nodes, providing a virtual or nonvirtual circuit. A router is dependent on protocols and establishes physical circuits for individual node-pair connection. In packet-switched networks, several pairs of communicating end points can share a circuit virtually, through their own dedicated channels, instead of occupying the entire physical circuit. Bandwidth in a packet-switched network is then dynamically released and shared as needed. A router has a routing look-up table for routing packets. If a communication is connectionless, packets of a message are sent individually but not in order. Owing to the presentation of more sophisticated input and output processors in routers, a router is not concerned about the order of the packets. Layer 3 switches are of two types:

- *Packet-by-packet switches*, by which packet forwarding is handled based on each individual packet.
- *Flow-based switches*, by which a number of packets having the same source and destination are identified and forwarded together. This scheme speeds up the forwarding process.

Layer 2 switches are developed to replace routers at a local area network. Typically, the general strategy of the network management is to use the technology of layer 2 switching at layer 3 routing to improve the quality of forwarding packets of routers and to make routers provide sufficient network guarantee. A layer 4 switch uses the information from higher levels for routing decisions. Basically, layer 4 switches are the application switches for both layer 2 and layer 3. In layer 4, packets are normally forwarded on a connectionless system.

3.4 Router Structure

Routers are the building blocks of wide area networks. Figure 3.10 shows an abstract model of a router as a layer 3 switch. Packets arrive at n input ports and are routed out from n output ports. The system consists of four main parts: *input port processors*, *output port processors*, *switch fabric* (switching network), and *switch controller*.

3.4.1 Input Port Processor (IPP)

Input and *output port processors*, as interfaces to switch fabric, are commercially implemented together in *router line cards*, which contain some of the task of the physical and data link layers. The functionality of the data link layer is implemented as a separate chip in IPP, which also provides a buffer to match the speed between the input and the switch fabric. Switch performance is limited by processing capability, storage elements, and bus bandwidth. The processing capability dictates the maximum rate of the switch. Owing to the speed mismatch between the rate at which a packet arrives on the switch and the processing speed of the switch fabric, input packet rate dictates the amount of required buffering storage. The bus bandwidth determines the time taken for a packet to be transferred between the input and output ports.

An *input port processor* (IPP) typically consists of several main modules, as shown in Figure 3.11. These modules are *packet fragmentation*, *main buffer*, *multicast process*, *routing table*, *packet encapsulator*, and a comprehensive *QoS*.

Packet Fragmentation

The *packet fragmentation unit*, converts packets to smaller sizes. Large packets cause different issues at the network and link layers. One obvious application of packet

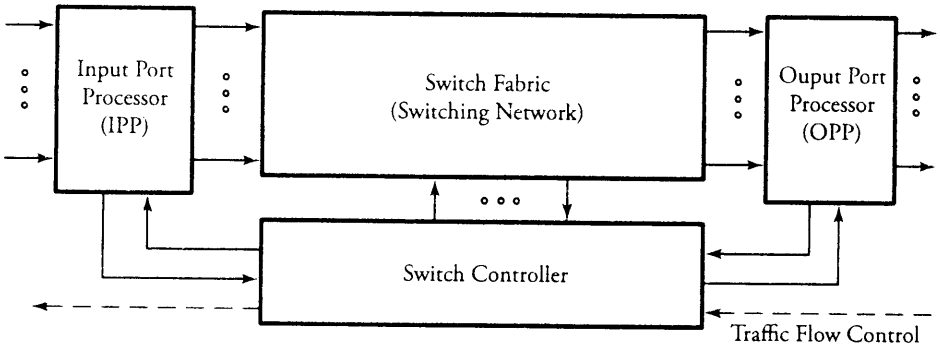


Figure 3.10 Overview of a typical router

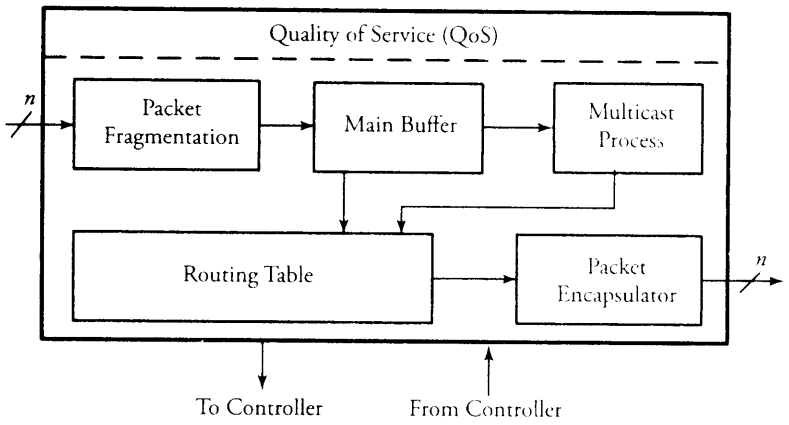


Figure 3.11 Overview of a typical IPP in routers

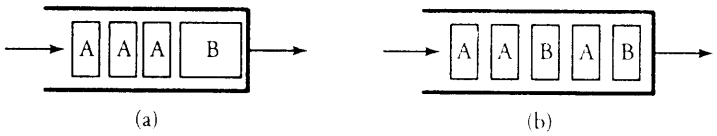


Figure 3.12 Packet fragmentation: (a) without fragmentation; (b) with fragmentation

fragmentation occurs in typical LANs, in which large packets must be fragmented into smaller frames. Another example occurs when large packets must be buffered at the input port interface of a router, as buffer slots are usually only 512 bytes long. One solution to this problem is to partition packets into smaller fragments and then reassemble them at the output port processor (OPP) after processing them in the switching system. Figure 3.12 shows simple packet fragmentation at the input buffer side of a switch. It is always desirable to find the optimum packet size that minimizes the delay.

Routing Table

The *routing table* is a look-up table containing all available destination addresses and the corresponding switch output port. An external algorithm fills this routing look-up table. Thus, the purpose of the routing table is to look up an entry corresponding to the destination address of the incoming packet and to provide the output network port. As soon as a routing decision is made, all the information should be saved on the routing table. When a packet enters an IPP, the destination port of the switch should

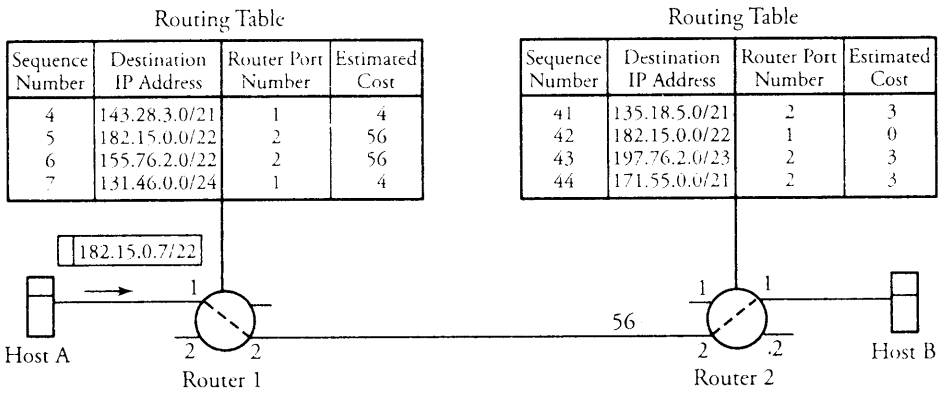


Figure 3.13 Routing tables at routers

be chosen, based on the destination address of the incoming packet. This destination port needs to be appended to the incoming packet as part of the switch header.

The look-up table management strategy takes advantage of first in, first out (FIFO) queues' speed and memory robustness. To increase memory performance, queue sizes are fixed to reduce control logic. Since network packets can be of various lengths, a memory device is needed to store packet payloads while a fixed-length header travels through the system. Since packets can arrive and leave the network in different order, a memory monitor is necessary to keep track of which locations in memory are free for use. Borrowing a concept from operating systems principles, a free-memory list serves as a memory manager implemented by a stack of pointers. When a packet carrying a destination address arrives from a given link i, its destination address is used to identify the corresponding output port j.

Figure 3.13 shows an example of routing tables at routers between hosts A and B. Assume that host B's address is requested by a packet with destination address 182.15.0.0/22 arriving at router 1. The routing table of this router stores the best-possible path for each destination. Assume that for a given time, this destination is found in entry row 5. The routing table then indicates that port 2 of the router is the right output to go. The table makes the routing decision, based on the estimated cost of the link, which is also stated in the corresponding entry. The cost of each link, as described in Chapter 7, is a measure of the load on each link. When the packet arrives at router 2, this switch performs the same procedure.

Multicast Process

A *multicast process* is necessary for copying packets when multiple copies of a packet are expected to be made on a switching node. Using a memory module for storage, copying is done efficiently. The copying function can easily be achieved by appending a counter field to memory locations to signify the needed number of copies of that location. The memory module is used to store packets and then duplicate multicast packets by holding memory until all instances of the multicast packet have exited IPP. Writing to memory takes two passes for a multicast packet and only one pass for a unicast packet. In order to keep track of how many copies a multicast packet needs, the packet counter in the memory module must be augmented after the multicast packet has been written to memory. Each entry in the memory module consists of a valid bit, a counter value, and memory data. The multicast techniques and protocols are described in a greater detail in Chapter 15.

Packet Encapsulation

Packet encapsulation instantiates the routing table module, performs the routing table lookups, and inserts the switch output port number into the network header. The *serial-to-parallel multiplexing* unit converts an incoming serial byte stream into a fully parallel data stream. This unit also processes the incoming IP header to determine whether the packet is unicast or multicast and extracts the type-of-service field. Once the full packet is received, it is stored into memory. The packet encapsulation unit formats the incoming packet with a header before forwarding the packet to the crossbar.

Congestion Controller

The *congestion controller* module shields the switching node from any disorders in the traffic flow. Congestion can be controlled in several ways. Sending a reverse-warning packet to the upstream node to avoid exceeding traffic is one common technology installed in the structure of advanced switching systems. Realistically, spacing between incoming packets is irregular. This irregularity may cause congestion in many cases. Congestion control is explained in Chapters 7, 8, and 12.

3.4.2 Switch Fabric

In the switch fabric of a router, packets are routed from input ports to the desired output ports. A packet can also be multicast to more than one output. Finally, in the output port processors, packets are buffered and resequenced in order to avoid packet

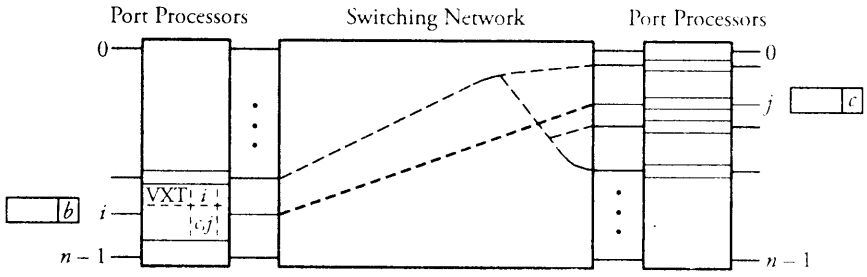


Figure 3.14 Interaction between an IPP and its switch fabric in a virtual-circuit switching router

misordering. In addition, a number of other important processes and functions taken place in each of the mentioned blocks.

Figure 3.14 shows an abstract model of a virtual-circuit switching router, another example of switching systems. This model can work for ATM technology: Cells (packets) arrive at n input ports and are routed out from n output ports. When a cell carrying VCI b arrives from a given link i, the cell's VCI is used to index a *virtual-circuit translation table* (VXT) in the corresponding input port processor to identify the output link address j and a new VCI c. In the switching network, cells are routed to the desired outputs. As shown in Figure 3.14, a cell can also be multicast to more than one output. Finally, in output port processors, cells are buffered; in some switch architectures, cells are resequenced in order to avoid misordering.

3.4.3 Switch Controller

The *controller* part of a switching system makes decisions leading to the transmission of packets to the requested output(s). The details of the controller are illustrated in Figure 3.15. The controller receives packets from an IPP, but only the headers of packets are processed in the controller. In the controller, the *header decoder* first converts the control information of an arriving packet into an initial requested output vector. This bit vector carries the information pertaining to the replication of a packet so that any bit of 1 represents a request for one of the corresponding switch outputs.

The initial request vector ultimately gets routed to the *buffer control* unit, which generates a priority value for each packet to enable it for arbitration. This information, along with the request vector, enters an array of *arbitration elements* in the *contention resolution unit*. Each packet in one column of an arbitration array contends with other packets on a shared bus to access the switch output associated with that column. After a packet wins the contention, its identity (buffer index number) is transmitted out to

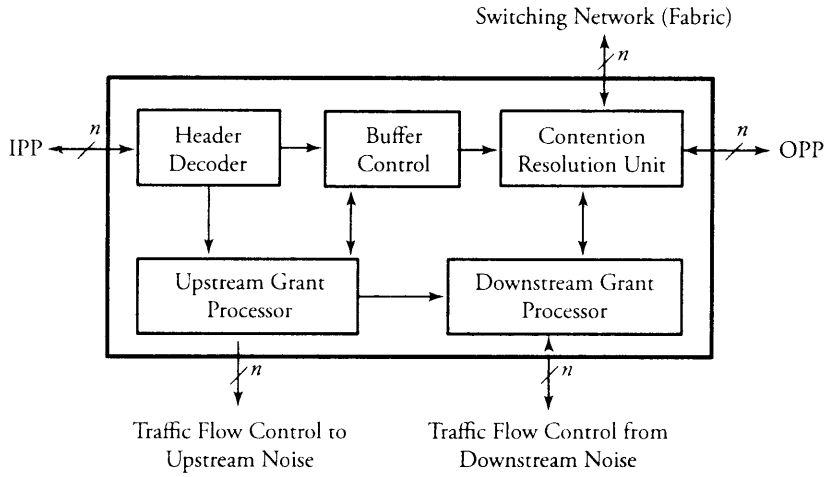


Figure 3.15 Overview of a switching system controller

an OPP. This identity and the buffer-control bit explained earlier are also transferred to the switching fabric (network), signaling them to release the packet. This mechanism ensures that a losing packet in the competition remains in the buffer. The buffer-control unit then raises the priority of the losing packet by 1 so that it can contribute in the next round of contention with a higher chance of winning. This process is repeated until eventually, the packet wins.

The identities of winning packets are transmitted to the switch fabric if traffic flow control signals from downstream neighboring nodes are active. The *upstream grant processor* in turn generates a corresponding set of traffic flow control signals, which are sent to the upstream neighboring nodes. This signal is an indication that the switch is prepared to receive a packet on the upstream node. This way, network congestion comes under control.

3.4.4 Output Port Processors (OPP)

Implementing *output port processors* in switches includes parallel-to-serial multiplexing, main buffer, local packet resequencer, global packet resequencer, error checker, and packet reassembler, as shown in Figure 3.16. Similar to IPP, OPP also contributes to congestion control. *Parallel-to-serial multiplexing* converts the parallel-packet format into serial packet format.

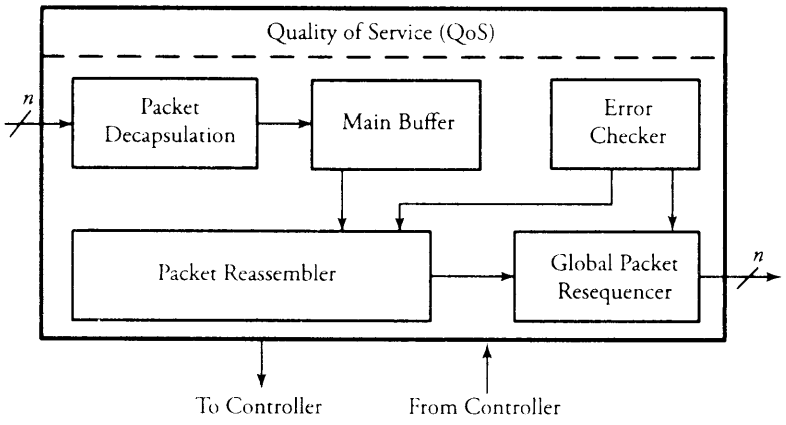


Figure 3.16 Overview of a typical OPP in routers

Main Buffer

The *buffer* unit serves as the OPP central shift register. The purpose of this buffer is to control the rate of the outgoing packets, which impacts the quality of service. After collecting signals serially from the switch fabric, the buffer forwards packets to resequencers. The queue runs on a clock driven by the link interface between the switch and an external link. This buffer must have features that support real-time and non-real-time data.

Reassembler and Resequencer

The output port processor receives a stream of packet fragments and has to identify and sort out all the related ones. The OPP reassembles them into a single packet, based on the information obtained from the fragment field of headers. For this process, the OPP must be able to handle the arrival of individual fragments at any time and in any order. Fragments may arrive out of order for many reasons. Misordered packets can occur because individual fragments, composed of a fairly large number of interconnections with different delay times, are independently routed through the switch fabric.

A *packet reassembler* buffer is used to combine fragments of IP packets. This unit resequences receiving packet fragments before transmitting them to external circuits, updates the total-length field of the IP header, and decapsulates all the local headers. The resequencer's internal buffer stores misordered fragments until a complete sequence is obtained. The in-sequence fragments are reassembled and transmitted to the external circuit. A *global packet resequencer* uses this same procedure to enforce another reordering, this time on sequences, not fragments, of packets that belong to a single user.

Error Checker and CRC

When a user sends a packet or a frame, a *cyclic redundancy check* (CRC) field is appended to the packet. The CRC is generated from an algorithm and is based on the data being carried in the packet. The CRC algorithms divide the message by another fixed-binary number in a polynomial form, producing a *checksum* as the remainder. The message receiver can perform the same division and compare the remainder with the received checksum. The *error checker* applies a series of error-checking processes on packets to ensure that no errors are on the packets and creates a stream of bits of a given length, called frame. A frame produces a *checksum bit*, called frame check sequence, which is attached to the data when transmitted.

3.5 Summary

This chapter introduced the hardware building blocks of computer networks. These building blocks, or nodes, are directly connected by physical links and allow message exchange in communication networks.

A *multiplexer* is a device that provides cost-effective connectivity by collecting data from multiple links and carrying that data on one link. Multiplexers are of various types and are subject to some useful analytical methods. Networking *modems* are used to access the Internet from remote and residential areas. A *DSL modem* uses twisted-pair telephone lines to access the Internet, whereas a *cable modem* a uses optical fiber cables to provide Internet access.

Switching devices are organized according to the layer at which they are connected in networks. *Repeaters* are layer 1 switching devices that primarily extend the geographic distance spanned by a LAN protocol. A *hub* or a *bridge* supports more ports than a repeater does. The most important component of the Internet, a *router*, is treated as a layer 3 switching device. A router is made up of *input port processors*, *output port processors*, *switch fabric*, and a *switch controller*.

In the next chapter, we look at issues related to data links. Data links can be evaluated both at the physical layer and the data link layer.

3.6 Exercises

1. Human voice frequency ranges from 16 Hz to about 20 KHz. Telephone companies use the most significant 4,000 Hz portion of this spectrum to deliver voice conversation between two users. This downgrade in the quality of conversation allows the transmission links to save bandwidth remarkably. Using a three-level hierarchy of multiplexing (12:1, 5:1, and 10:1):

(a) How many voice conversation can this system carry?
(b) What would be the final transmission capacity of this system?

2. Consider the integration of three analog sources and four identical digital sources through a time-division multiplexer that uses its entire 160 Kb/s maximum capacity. The analog lines—with bandwidths of 5 KHz, 2 KHz, and 1 KHz, respectively—are sampled, multiplexed, quantized, and 5-bit encoded. The digital lines are multiplexed, and each carries 8 Kb/s.
   (a) For the analog sources, find the total bit rate.
   (b) For the digital sources, find the pulse-stuffing rate.
   (c) Find the frame rate if a frame carries eight sampled analog channels, four digital data channels, a control channel, and a guard bit.

3. Assume that two 600 b/s terminals, five 300 b/s terminals, and a number of 150 b/s terminals are to be time-multiplexed in a character-interleaved format over a 4,800-b/s digital line. The terminals send 10 bits/character, and one synchronization character is inserted for every 99 data characters. All the terminals are asynchronous, and 3 percent of the line capacity is allocated for pulse stuffing to accommodate variations in the terminal clock rate.
   (a) Determine the number of 150 b/s terminals that can be accommodated.
   (b) Sketch a possible framing pattern for the multiplexer, assuming three characters per 150 b/s terminals.

4. Consider a time-division multiplexer having a frame time of 26 μs. Each user channel has 6 bits, and each frame has 10 bits of overhead information. Assume that the transmission line carries 2 Mb/s of information.
   (a) How many user channels can be accommodated on the line?
   (b) Consider ten sources in this system, and assume a probability of 0.9 that a source is busy. What is the clipping probability?

5. Consider a synchronous TDM with eight inputs, each becoming an average of 2 μs active and 6 μs inactive. Frames can receive four channels only.
   (a) Find the probability that a source is active.
   (b) Find the probability that three channels of the frame are in use.
   (c) Find the blocking probability for this multiplexer.
   (d) Find the average number of used channels in the frame.

6. For a four-input statistical TDM, consider two different frame sizes of 2 and 3. Sketch one set of three plots, each showing the clipping probability versus ρ = 0.2, 0.4, 0.6, 0.8.

7. Consider a statistical TDM in which 11 sources and 10 channels are present. Find the clipping probability, assuming a probability of 0.9 that a given source is busy.
8. Find the *average clipping time* for each burst of information in statistical TDMs.
9. A string of 110011101 arrives at the line coder of a modem. Give the output form if the line coder is designed by:
   (a) Natural NRZ
   (b) Polar NRZ
   (c) Manchester
10. A string of 100101011 arrives at the modulation unit of a modem. Give the output signal if the modulator is designed by :
   (a) ASK
   (b) FSK
   (c) PSK
11. We want to design the input port processor of a high-speed router and analyze the delay. Incoming packets to IPP are fragmented into smaller segments, with each segment consisting of d bits data plus 50 bits of header. The switch fabric requires segments to be transmitted at r b/s. To achieve the highest possible speed:
   (a) Is there any way to optimize the transmission delay D of each segment in terms of d and r? How?
   (b) Is propagation delay in the switch fabric significant compared to D? Why?
12. *Computer simulation project.* To simulate the routing table of routers, write a computer program to construct a look-up routing table with ten entries. Each entry must contain a sequence number, time, destination address, cost of a destination node (given a certain network), and router port number. Your program should demonstrate the updating mechanism on a frequent basis.

CHAPTER 4

Data Links and Transmission

So far, we have discussed basic networking protocols and devices. This chapter focuses on data *links*, especially on methods of data transmission, both wired and wireless. After introducing general properties of transmission media, we investigate issues in the link layer of the overall protocol stack. The highlights of this chapter are

- *Data links*
- *Wired links and transmission*
- *Wireless links and transmission*
- *Methods of channel access on links*
- *Error detection and correction*
- *Flow control at the link layer*

We begin by discussing wired and wireless transmission media. We examine most guided and unguided link alternatives, briefly discuss their applications, and summarize key transmission characteristics. Our next major topic is methods of focusing on channel access, the physical and link layers. We also discuss *error detection* and *error correction* on data links. Finally, we explain the *stop-and-wait* and *sliding-window*. protocols, which guarantee the control of link flows.

4.1 Data Links

A *data link* is the physical path between a data transmitter and data receiver. Figure 4.1 shows the range of electromagnetic-spectrum frequencies for various applications in data communications.

- The *low-frequency* subspectrum covers all the frequencies in the range 0 to approximately 15,000 Hz, which is the range of human-voice frequencies generally used in *telephone systems*.
- The *radio* frequency (RF) subspectrum covers frequencies from several KHz to several GHz. RF applications in telecommunications includes *radio systems*, *television systems*, *Bluetooth communications*, and *cell phones*.
- The *microwave* frequency subspectrum ranges from several GHz up to more than 10^{11} Hz and is used for such applications as *microwave systems*, *radar*, and *satellite communications*.
- The *infrared* frequency subspectrum ranges from more than 10^{11} Hz to less than 10^{14} Hz. The infrared signal can be used for *remote controls*, *lasers*, and *guided missiles*.
- The *light* frequency subspectrum covers all the visible-light components and is used mainly for *fiber-optic communications*.

Data transmission may use either *wired links* or *wireless links*, depending on the application and the available bandwidth of links. Transmission links can also be classified as *guided*, or directional, and *unguided*. A wired link is normally considered a guided medium for the propagation of data. A wireless medium can be designed to propagate signals in more than one direction, causing the medium to become unguided. Signals travel on the link in the form of electromagnetic waves. Let c be the speed of electro-

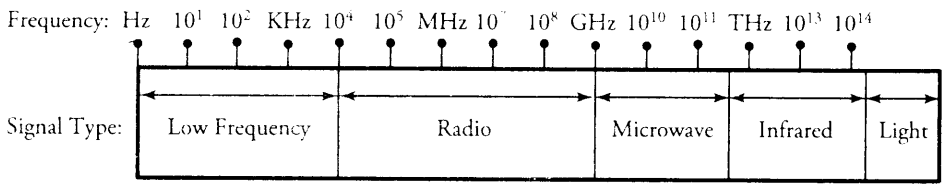


Figure 4.1 Frequency range for various data communications applications

magnetic waves, f be the frequency of the traveling signal, and λ be the wavelength of the signal. Then:

$$\lambda = \frac{c}{f}. \tag{4.1}$$

A guided link, whether wired or wireless, can be *full-duplex*, whereby two bitstreams in opposite directions can be transmitted, or *half-duplex*, whereby only one bitstream in one direction can be carried at any time. The following section explores the most commonly used data links.

4.2 Wired Links and Transmission

Wired links provide a physical path for signals to propagate. Three types of wired links are *twisted pair*, *coaxial cable*, and *optical fiber*.

4.2.1 Twisted-Pair Links

A *twisted-pair link* is the simplest form of guided medium used for data transmission. A twisted pair is normally manufactured using copper and consists of two insulated wires. The twisting action on wires reduces the cross-talk interferences generated between each two pairs of transmission links. To increase the capacity of transmission cables, especially for long-distance applications, several pairs of such links are bundled together and wrapped in a protective sheath. One of the most common applications of the twisted-pair link is for telephone network transmission links. The frequency range of twisted-pair cables is approximately 0 to 1 MHz.

4.2.2 Coaxial Cable

A higher data rate for longer-distance applications can be achieved with *coaxial cable*, a hollow outer cylindrical conductor surrounding an inner wire. The outer conductor is spaced tightly with inner wire by a solid dielectric material. The outer conductor is also shielded from outside. This concentric construction makes coaxial cables susceptible to interference. Coaxial cables have a wide variety of applications, such as cable television distribution, long-distance telephone transmission, and local area networks. The frequency range that coaxial cables can carry is 0 to 750 MHz.

4.2.3 Optical Fiber

Remarkably higher-bandwidth communication links can be achieved using optical fibers. An optical fiber is a thin glass or plastic wire that can guide an optical ray. One

of the best substances used to make optical fibers is *ultrapure fused silica*. These fibers are, however, more expensive than regular glass fibers. Plastic fibers are normally used for short-distance links where higher losses are tolerable.

Similar to coaxial cables, optical fiber cables have a cylindrical layout. The three concentrics are the *core*, the *cladding*, and the *jacket*. The core consists of several very thin fibers, each of which is surrounded by its own cladding. Each cladding also has a glass or plastic coating but different from those of the core. This difference is the key mechanism that confines the light in the cable. Basically, the boundary between the core and cladding reflects the light into the core and runs it through the cable.

The combined core and cladding is surrounded by a jacket. Jackets are made of materials that can protect the cable against interference and damage. Optical fibers are superior to coaxial cables mainly because of their higher bandwidths, lighter weights, lower signal attenuation, and lower impact by external interferences. Optical fiber links are used in all types of data communication LAN and WAN applications. The frequency range of fiber optics is approximately 180 THz to 330 THz.

4.3 Wireless Links and Transmission

Computer networks can take advantage of the wireless infrastructure where physical wires cannot be laid out. An obvious example is mobile data communication, whereby mobile users attempt to connect and stay connected to the Internet. Wireless class education is another example; an instructor teaches class through wireless media, and students can follow the lecture with their portable computers from any location within a defined vicinity.

One of the key challenges in wireless networking is the efficient utilization of the available transmission spectrum. Because the frequency spectrum available for wireless communication is normally limited, frequencies must be reused within the same geographic area. The spectrum used for wireless communications typically ranges up to several GHz. Security is also a concern in wireless networks. The open-air interface makes it difficult to prevent snooping.

The link-level design techniques involve making trade-offs among the various parameters relevant to the link layer. The optimum design would involve the use of minimum bandwidth and transmit power while maintaining a high data rate, low latency, and low bit error rates (BER). These design challenges must be achieved in the presence of channel imperfections, such as flat fading, multipath effects, shadowing, and interference.

Wireless links, both guided and unguided, are used for data communications. Wireless links use devices as an antenna for transmitting signals through *vacuum, space, air*, or *substances*. Electromagnetic waves can be propagated through the first three, as well as through water and wood. The frequency range depends on the type of substance. The two key challenges faced in overall design of efficient wireless links and transmission systems are the *choice of antenna* and *wireless channels*.

4.3.1 Choice of Antenna

A good antenna in a wireless system can potentially enhance the signal-to-noise ratio. Antennas are classified in several ways. The two main types of antennas used in wireless systems are *isotropic antennas* and *directional antennas*.

Isotropic Antennas

Isotropic antennas transmit signals equally in all directions. Consider an isotropic transmitter that radiates P_t watts equally in all directions, forming a sphere of flux with radius d. Given that the surface area of the sphere is $4\pi d^2$, power-flux density measured on the surface of the sphere used by a receiver located at distance d is

$$\phi_r = \frac{P_t}{4\pi d^2}. \tag{4.2}$$

At the other end of communication systems, P_r, as the captured power, depends on the size and orientation of the antenna with respect to the transmitter. If we let a be the effective area of the receiving antenna, P_t and P_r are related by

$$P_r = \phi_r a = \left(\frac{P_t}{4\pi d^2}\right) a. \tag{4.3}$$

According to electromagnetic theory, the effective area of an isotropic antenna is obtained by $a = \frac{\lambda^2}{4\pi}$. Thus, Equation (4.3) can be rewritten as

$$P_r = \frac{P_t}{\left(\frac{4\pi d}{\lambda}\right)^2}, \tag{4.4}$$

where λ is the wavelength of the signal obtained from Equation (4.1). In most propagation media other than free space, the received signal power varies inversely with d^3 or d^4, compared to d^2 for free space.

Directional Antennas

Directional antennas are used to mitigate unwanted effects. Directional antennas amplify the signal in a small angular range but attenuate the signal at all other angles. This helps reduce the power in the various multipath components at the receiver. Directional antennas can be used to reduce the effects of interference from other users. The antenna must be accurately pointed to the correct user and must follow the user's path. This fact should lead to the development of smart antennas that can be used to track mobile users accurately by antenna steering.

4.3.2 Wireless Channels

Wireless communication is characterized by several channel impediments. A wireless channel is a portion of transmission bandwidth through which communication can be established. Channels are susceptible to interference and noise. The characteristics of a wireless channel vary with time and user movement. Most commercial wireless systems use radio waves in the ultrahigh frequency (UHF) band for communication. The UHF band ranges from 0.3 GHz to about 3 GHz. Satellite communication typically uses super-high frequency (SHF) band ranging from 3 GHz to 30 GHz. The transmitted signal reaches a receiver via three different paths: *direct*, *scattering*, and *reflection*. Signals arriving at a receiver through scattering and reflection are normally shifted in amplitude and phase. Wireless channels are characterized by four main characteristics: *path loss*, *shadowing*, *multipath fading*, and *interference*.

Path Loss

Path loss is a measure of degradation in the received signal power. The path loss depends on the transmitted power and the propagation distance. An important measure of the strength of the received signal is the signal-to-noise ratio (*SNR*). If the average noise power at the receiver is P_n, the signal-to-noise ratio is given by

$$SNR = \frac{P_r}{P_n}, \tag{4.5}$$

where we defined P_r to be the captured power in Equation (4.4). The received signal power decreases for higher frequencies, since these signals carry more power. Thus, the path loss increases with higher frequencies. The error rate in the channel is reduced

when the signal-to-noise ratio is maintained at a high level. The path loss, L_p, is obtained from Equation (4.6):

$$L_p = \left(\frac{4\pi d}{\lambda}\right)^2. \tag{4.6}$$

Note that the path loss is the ratio of transmitted power to received power.

Example. Consider a commercial wireless mobile telephone system. Assume a transmitter operating at a frequency of 850 MHz and with a power of 100 miliwatts communicates with a mobile receiver with received power of 10^{-6} microwatts. Find the distance between the transmitter and the receiver.

Solution. We can first find the path loss by

$$L_p = \frac{P_t}{P_r} = \frac{100 \times 10^{-3}}{10^{-6} \times 10^{-6}} = 10^{11}. \tag{4.7}$$

Knowing that $\lambda = c/f$, where f is the operating frequency of 850 MHz and c is the speed of light in free space estimated at 3×10^8 m/s, we can use Equation (4.6) to obtain d. The answer is $d = 8.8$ km.

Shadow Fading

As it propagates through the wireless medium, a signal encounters various obstructions, such as buildings, walls, and other objects. Physical obstructions make the transmitted signal face signal attenuation. The variation of the received signal power due to these obstructions is called *shadow fading*. In typical cases, the variation in the received signal power because of shadow fading follows a Gaussian distribution. From Equation (4.3), the received signal power seems to be the same at equal distance from the transmitter. However, even when d is the same in Equation (4.3), the received signal power varies, since some locations face greater shadow fading than do others. Normally, the transmit power P_t should be increased to compensate for the shadow-fading effect. Figure 4.2 shows the shadow-fading effect of a received signal.

Flat and Deep Fading

At each wireless receiver, the received signal power fluctuates rapidly with time and slightly with distances, a phenomenon called *flat fading*. Figure 4.2 shows a received signal power that varies with distance. The figure shows that the received signal power

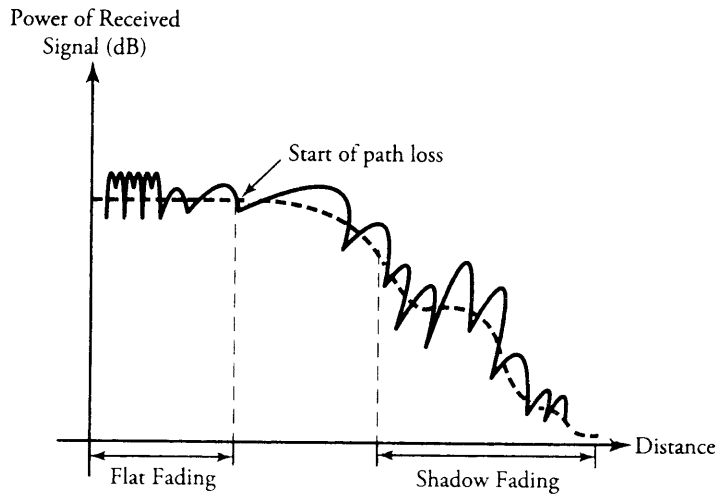


Figure 4.2 Flat fading, path loss, and shadow fading

falls from its average value. This phenomenon, whereby the received signal power falls below the value required to meet link-performance constraints, is called *deep fading*.

Doppler Frequency Shift

Let v_r be the relative velocity between a transmitter and a receiver. The shift in the frequency of the transmitted signal caused by the relative velocity is called *Doppler shift* and is expressed by f_D:

$$f_D = \frac{v_r}{\lambda}, \tag{4.8}$$

where λ is the wavelength of the transmitted signal. As the relative velocity changes with time, the Doppler shift also varies. In frequency modulations, this shift could result in an increase in the bandwidth of the signal. In most scenarios, the Doppler shift can be of the order of several Hz, whereas the signal bandwidth is of the order of several KHz. Thus, the effect of Doppler shift is negligible in these cases.

Interference

The limited frequency spectrum in wireless channels leads to frequency reuse at spatially separated locations. Frequency reuse can lead to *interference*. The interference can be reduced by using more complex systems, such as dynamic channel allocation, multiuser detection, and directional antennas. Interference can also result from adjacent