PART I

Overview

In the first part of the book, we discuss some general ideas related to both data communications and networking. This part lays the plan for the rest of the book. The part is made of two chapters that prepare the reader for the long journey ahead.

CHAPTER 1

Introduction

Data communications and networking have changed the way we do business and the way we live. Business decisions have to be made ever more quickly, and the decision makers require immediate access to accurate information. Why wait a week for that report from Europe to arrive by mail when it could appear almost instantaneously through computer networks? Businesses today rely on computer networks and internetworks.

Data communication and networking have found their way not only through business and personal communication, they have found many applications in political and social issues. People have found how to communicate with other people in the world to express their social and political opinions and problems. Communities in the world are not isolated anymore.

But before we ask how quickly we can get hooked up, we need to know how networks operate, what types of technologies are available, and which design best fills which set of needs.

This chapter paves the way for the rest of the book. It is divided into five sections.

- The first section introduces data communications and defines their components and the types of data exchanged. It also shows how different types of data are represented and how data is flowed through the network.
- The second section introduces networks and defines their criteria and structures. It introduces four different network topologies that are encountered throughout the book.
- The third section discusses different types of networks: LANs, WANs, and internetworks (internets). It also introduces the Internet, the largest internet in the world. The concept of switching is also introduced in this section to show how small networks can be combined to create larger ones.
- The fourth section covers a brief history of the Internet. The section is divided into three eras: early history, the birth of the Internet, and the issues related to the Internet today. This section can be skipped if the reader is familiar with this history.
- The fifth section covers standards and standards organizations. The section covers Internet standards and Internet administration. We refer to these standards and organizations throughout the book.

1.1 DATA COMMUNICATIONS

When we communicate, we are sharing information. This sharing can be local or remote. Between individuals, local communication usually occurs face to face, while remote communication takes place over distance. The term ***telecommunication,*** which includes telephony, telegraphy, and television, means communication at a distance (*tele* is Greek for "far"). The word ***data*** refers to information presented in whatever form is agreed upon by the parties creating and using the data.

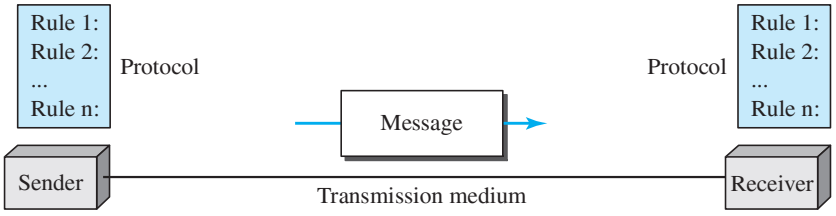
Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

1. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.
2. **Accuracy.** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.
3. **Timeliness.** The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called *real-time* transmission.
4. **Jitter.** Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

1.1.1 Components

A data communications system has five components (see Figure 1.1).

Figure 1.1 *Five components of data communication*



1. **Message.** The **message** is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
2. **Sender.** The **sender** is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

3. **Receiver.** The **receiver** is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
4. **Transmission medium.** The **transmission medium** is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.
5. **Protocol.** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

1.1.2 Data Representation

Information today comes in different forms such as text, numbers, images, audio, and video.

Text

In data communications, text is represented as a bit pattern, a sequence of bits (0s or 1s). Different sets of bit patterns have been designed to represent text symbols. Each set is called a **code,** and the process of representing symbols is called coding. Today, the prevalent coding system is called **Unicode,** which uses 32 bits to represent a symbol or character used in any language in the world. The **American Standard Code for Information Interchange (ASCII),** developed some decades ago in the United States, now constitutes the first 127 characters in Unicode and is also referred to as **Basic Latin.** Appendix A includes part of the Unicode.

Numbers

Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations. Appendix B discusses several different numbering systems.

Images

Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the *resolution.* For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image (better resolution), but more memory is needed to store the image.

After an image is divided into pixels, each pixel is assigned a bit pattern. The size and the value of the pattern depend on the image. For an image made of only black-and-white dots (e.g., a chessboard), a 1-bit pattern is enough to represent a pixel.

If an image is not made of pure white and pure black pixels, we can increase the size of the bit pattern to include gray scale. For example, to show four levels of gray scale, we can use 2-bit patterns. A black pixel can be represented by 00, a dark gray pixel by 01, a light gray pixel by 10, and a white pixel by 11.

There are several methods to represent color images. One method is called **RGB,** so called because each color is made of a combination of three primary colors: *r*ed, *g*reen, and *b*lue. The intensity of each color is measured, and a bit pattern is assigned to

it. Another method is called **YCM,** in which a color is made of a combination of three other primary colors: *y*ellow, *c*yan, and *m*agenta.

Audio

Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal. We will learn more about audio in Chapter 26.

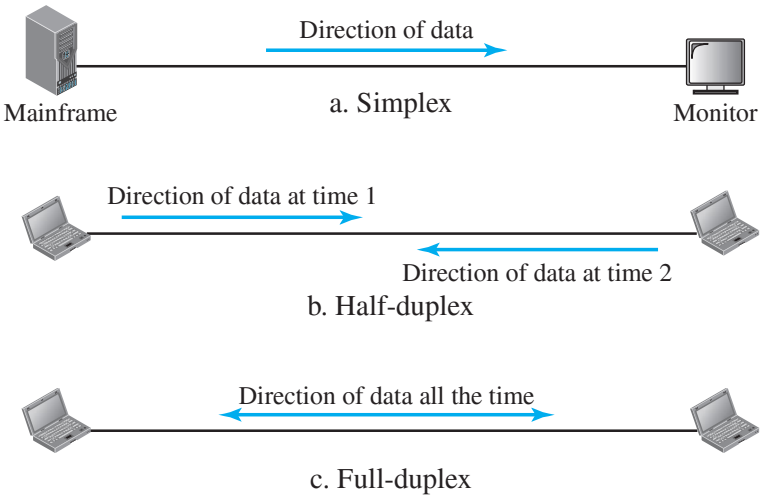
Video

Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion. We will learn more about video in Chapter 26.

1.1.3 Data Flow

Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure 1.2.

Figure 1.2 *Data flow (simplex, half-duplex, and full-duplex)*



Simplex

In **simplex mode,** the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (see Figure 1.2a).

Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output. The simplex mode can use the entire capacity of the channel to send data in one direction.

Half-Duplex

In **half-duplex mode,** each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa (see Figure 1.2b).

The half-duplex mode is like a one-lane road with traffic allowed in both directions. When cars are traveling in one direction, cars going the other way must wait. In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

Full-Duplex

In **full-duplex mode** (also called *duplex*), both stations can transmit and receive simultaneously (see Figure 1.2c).

The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in one direction share the capacity of the link with signals going in the other direction. This sharing can occur in two ways: Either the link must contain two physically separate transmission paths, one for sending and the other for receiving; or the capacity of the channel is divided between signals traveling in both directions.

One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time.

The full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the two directions.

1.2 NETWORKS

A **network** is the interconnection of a set of devices capable of communication. In this definition, a device can be a **host** (or an *end system* as it is sometimes called) such as a large computer, desktop, laptop, workstation, cellular phone, or security system. A device in this definition can also be a **connecting device** such as a router, which connects the network to other networks, a switch, which connects devices together, a modem (modulator-demodulator), which changes the form of data, and so on. These devices in a network are connected using wired or wireless transmission media such as cable or air. When we connect two computers at home using a plug-and-play router, we have created a network, although very small.

1.2.1 Network Criteria

A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

Performance

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software.

Performance is often evaluated by two networking metrics: **throughput** and **delay.** We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability

In addition to accuracy of delivery, network **reliability** is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security

Network **security** issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

1.2.2 Physical Structures

Before discussing networks, we need to define some network attributes.

Type of Connection

A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. For visualization purposes, it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time. There are two possible types of connections: point-to-point and multipoint.

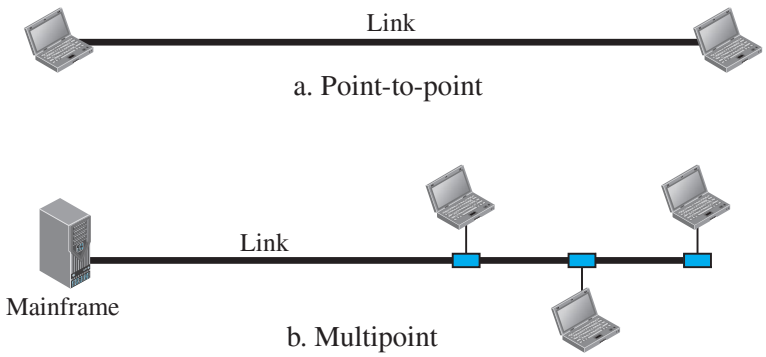
Point-to-Point

A **point-to-point connection** provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible (see Figure 1.3a). When we change television channels by infrared remote control, we are establishing a point-to-point connection between the remote control and the television's control system.

Multipoint

A **multipoint** (also called **multidrop**) **connection** is one in which more than two specific devices share a single link (see Figure 1.3b).

Figure 1.3 *Types of connections: point-to-point and multipoint*



In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.

Physical Topology

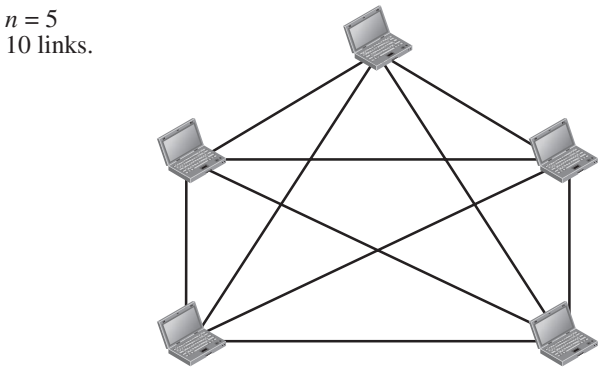
The term ***physical topology*** refers to the way in which a network is laid out physically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called ***nodes***) to one another. There are four basic topologies possible: mesh, star, bus, and ring.

Mesh Topology

In a **mesh topology,** every device has a dedicated point-to-point link to every other device. The term *dedicated* means that the link carries traffic only between the two devices it connects. To find the number of physical links in a fully connected mesh network with n nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to $n - 1$ nodes, node 2 must be connected to $n - 1$ nodes, and finally node n must be connected to $n - 1$ nodes. We need $n\ (n - 1)$ physical links. However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need $\boldsymbol{n\ (n - 1) / 2}$ duplex-mode links. To accommodate that many links, every device on the network must have $n - 1$ input/output (I/O) ports (see Figure 1.4) to be connected to the other $n - 1$ stations.

A mesh offers several advantages over other network topologies. First, the use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices. Second, a mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system. Third, there is the advantage of privacy or security. When every message travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages. Finally, point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems. This facility enables the network manager to discover the precise location of the fault and aids in finding its cause and solution.

Figure 1.4 *A fully connected mesh topology (five devices)*



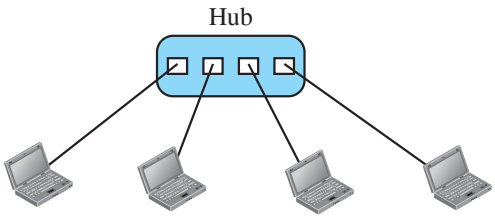
The main disadvantages of a mesh are related to the amount of cabling and the number of I/O ports required. First, because every device must be connected to every other device, installation and reconnection are difficult. Second, the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate. Finally, the hardware required to connect each link (I/O ports and cable) can be prohibitively expensive. For these reasons a mesh topology is usually implemented in a limited fashion, for example, as a backbone connecting the main computers of a hybrid network that can include several other topologies.

One practical example of a mesh topology is the connection of telephone regional offices in which each regional office needs to be connected to every other regional office.

Star Topology

In a **star topology,** each device has a dedicated point-to-point link only to a central controller, usually called a ***hub.*** The devices are not directly linked to one another. Unlike a mesh topology, a star topology does not allow direct traffic between devices. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device (see Figure 1.5) .

Figure 1.5 *A star topology connecting four stations*



A star topology is less expensive than a mesh topology. In a star, each device needs only one link and one I/O port to connect it to any number of others. This factor also makes it easy to install and reconfigure. Far less cabling needs to be housed, and

additions, moves, and deletions involve only one connection: between that device and the hub.

Other advantages include robustness. If one link fails, only that link is affected. All other links remain active. This factor also lends itself to easy fault identification and fault isolation. As long as the hub is working, it can be used to monitor link problems and bypass defective links.

One big disadvantage of a star topology is the dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead.

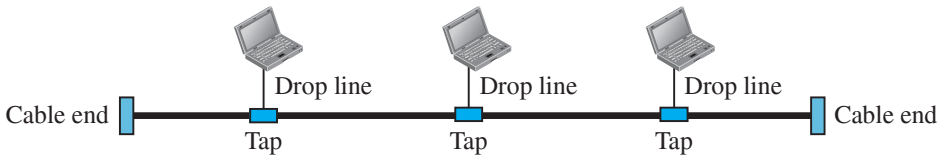
Although a star requires far less cable than a mesh, each node must be linked to a central hub. For this reason, often more cabling is required in a star than in some other topologies (such as ring or bus).

The star topology is used in local-area networks (LANs), as we will see in Chapter 13. High-speed LANs often use a star topology with a central hub.

Bus Topology

The preceding examples all describe point-to-point connections. A **bus topology,** on the other hand, is multipoint. One long cable acts as a **backbone** to link all the devices in a network (see Figure 1.6).

Figure 1.6 *A bus topology connecting three stations*



Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core. As a signal travels along the backbone, some of its energy is transformed into heat. Therefore, it becomes weaker and weaker as it travels farther and farther. For this reason there is a limit on the number of taps a bus can support and on the distance between those taps.

Advantages of a bus topology include ease of installation. Backbone cable can be laid along the most efficient path, then connected to the nodes by drop lines of various lengths. In this way, a bus uses less cabling than mesh or star topologies. In a star, for example, four network devices in the same room require four lengths of cable reaching all the way to the hub. In a bus, this redundancy is eliminated. Only the backbone cable stretches through the entire facility. Each drop line has to reach only as far as the nearest point on the backbone.

Disadvantages include difficult reconnection and fault isolation. A bus is usually designed to be optimally efficient at installation. It can therefore be difficult to add new devices. Signal reflection at the taps can cause degradation in quality. This degradation can be controlled by limiting the number and spacing of devices connected to a given

length of cable. Adding new devices may therefore require modification or replacement of the backbone.

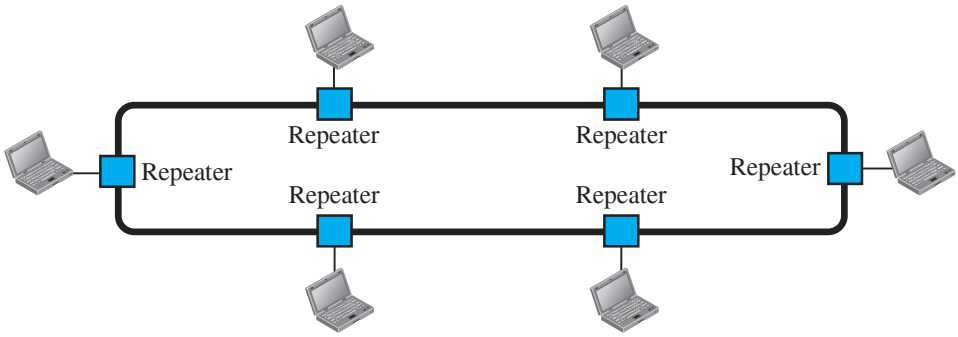
In addition, a fault or break in the bus cable stops all transmission, even between devices on the same side of the problem. The damaged area reflects signals back in the direction of origin, creating noise in both directions.

Bus topology was the one of the first topologies used in the design of early local-area networks. Traditional Ethernet LANs can use a bus topology, but they are less popular now for reasons we will discuss in Chapter 13.

Ring Topology

In a **ring topology,** each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along (see Figure 1.7).

Figure 1.7 *A ring topology connecting six stations*



A ring is relatively easy to install and reconfigure. Each device is linked to only its immediate neighbors (either physically or logically). To add or delete a device requires changing only two connections. The only constraints are media and traffic considerations (maximum ring length and number of devices). In addition, fault isolation is simplified. Generally, in a ring a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location.

However, unidirectional traffic can be a disadvantage. In a simple ring, a break in the ring (such as a disabled station) can disable the entire network. This weakness can be solved by using a dual ring or a switch capable of closing off the break.

Ring topology was prevalent when IBM introduced its local-area network, Token Ring. Today, the need for higher-speed LANs has made this topology less popular.

1.3 NETWORK TYPES

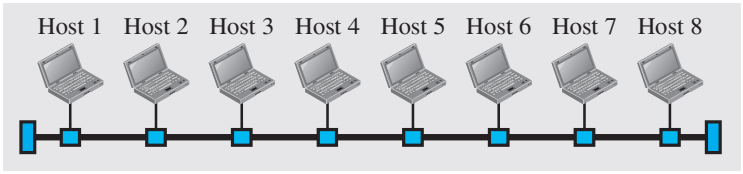
After defining networks in the previous section and discussing their physical structures, we need to discuss different types of networks we encounter in the world today. The criteria of distinguishing one type of network from another is difficult and sometimes confusing. We use a few criteria such as size, geographical coverage, and ownership to make this distinction. After discussing two types of networks, LANs and WANs, we define switching, which is used to connect networks to form an internetwork (a network of networks).

1.3.1 Local Area Network

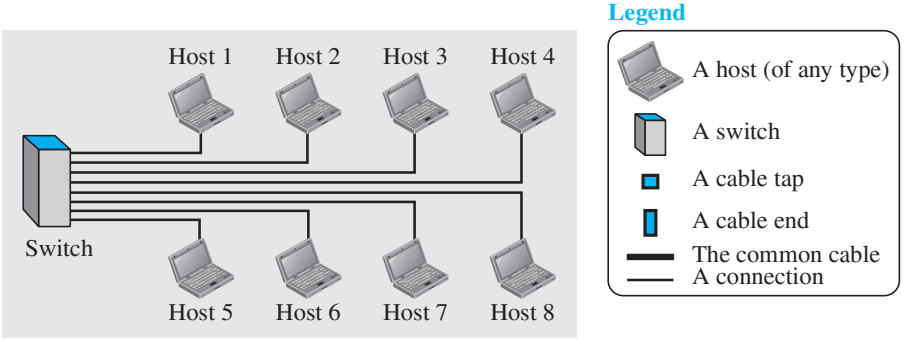
A **local area network** (**LAN**) is usually privately owned and connects some hosts in a single office, building, or campus. Depending on the needs of an organization, a LAN can be as simple as two PCs and a printer in someone's home office, or it can extend throughout a company and include audio and video devices. Each host in a LAN has an identifier, an address, that uniquely defines the host in the LAN. A packet sent by a host to another host carries both the source host's and the destination host's addresses.

In the past, all hosts in a network were connected through a common cable, which meant that a packet sent from one host to another was received by all hosts. The intended recipient kept the packet; the others dropped the packet. Today, most LANs use a smart connecting switch, which is able to recognize the destination address of the packet and guide the packet to its destination without sending it to all other hosts. The switch alleviates the traffic in the LAN and allows more than one pair to communicate with each other at the same time if there is no common source and destination among them. Note that the above definition of a LAN does not define the minimum or maximum number of hosts in a LAN. Figure 1.8 shows a LAN using either a common cable or a switch.

Figure 1.8 *An isolated LAN in the past and today*



a. LAN with a common cable (past)



b. LAN with a switch (today)

LANs are discussed in more detail in Part III of the book.

When LANs were used in isolation (which is rare today), they were designed to allow resources to be shared between the hosts. As we will see shortly, LANs today are connected to each other and to WANs (discussed next) to create communication at a wider level.

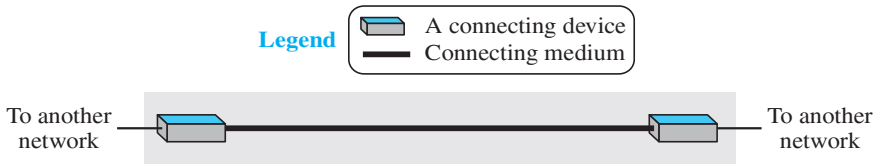
1.3.2 Wide Area Network

A **wide area network (WAN)** is also an interconnection of devices capable of communication. However, there are some differences between a LAN and a WAN. A LAN is normally limited in size, spanning an office, a building, or a campus; a WAN has a wider geographical span, spanning a town, a state, a country, or even the world. A LAN interconnects hosts; a WAN interconnects connecting devices such as switches, routers, or modems. A LAN is normally privately owned by the organization that uses it; a WAN is normally created and run by communication companies and leased by an organization that uses it. We see two distinct examples of WANs today: point-to-point WANs and switched WANs.

Point-to-Point WAN

A point-to-point WAN is a network that connects two communicating devices through a transmission media (cable or air). We will see examples of these WANs when we discuss how to connect the networks to one another. Figure 1.9 shows an example of a point-to-point WAN.

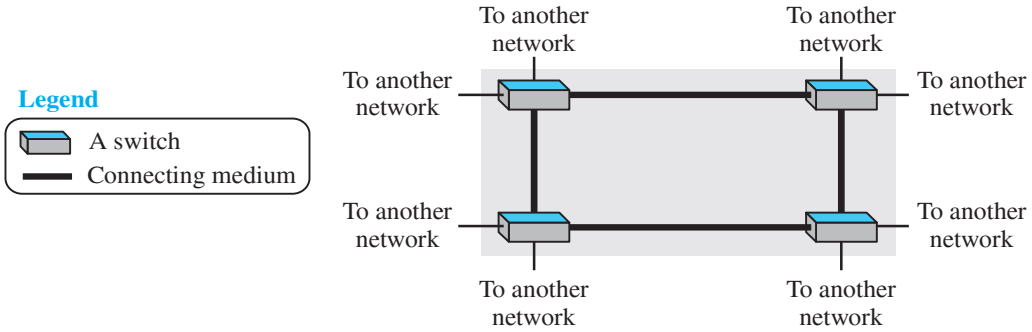
Figure 1.9 *A point-to-point WAN*



Switched WAN

A switched WAN is a network with more than two ends. A switched WAN, as we will see shortly, is used in the backbone of global communication today. We can say that a switched WAN is a combination of several point-to-point WANs that are connected by switches. Figure 1.10 shows an example of a switched WAN.

Figure 1.10 *A switched WAN*

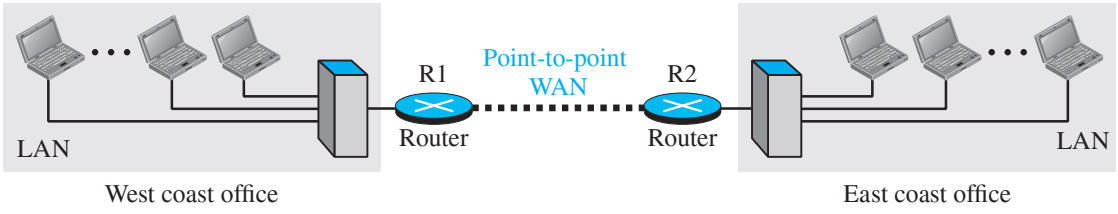


WANs are discussed in more detail in Part II of the book.

Internetwork

Today, it is very rare to see a LAN or a WAN in isolation; they are connected to one another. When two or more networks are connected, they make an **internetwork,** or **internet.** As an example, assume that an organization has two offices, one on the east coast and the other on the west coast. Each office has a LAN that allows all employees in the office to communicate with each other. To make the communication between employees at different offices possible, the management leases a point-to-point dedicated WAN from a service provider, such as a telephone company, and connects the two LANs. Now the company has an internetwork, or a private internet (with lowercase *i*). Communication between offices is now possible. Figure 1.11 shows this internet.

Figure 1.11 *An internetwork made of two LANs and one point-to-point WAN*



When a host in the west coast office sends a message to another host in the same office, the router blocks the message, but the switch directs the message to the destination. On the other hand, when a host on the west coast sends a message to a host on the east coast, router R1 routes the packet to router R2, and the packet reaches the destination.

Figure 1.12 (see next page) shows another internet with several LANs and WANs connected. One of the WANs is a switched WAN with four switches.

1.3.3 Switching

An internet is a **switched network** in which a switch connects at least two links together. A switch needs to forward data from a network to another network when required. The two most common types of switched networks are circuit-switched and packet-switched networks. We discuss both next.

Circuit-Switched Network

In a **circuit-switched network,** a dedicated connection, called a circuit, is always available between the two end systems; the switch can only make it active or inactive. Figure 1.13 shows a very simple switched network that connects four telephones to each end. We have used telephone sets instead of computers as an end system because circuit switching was very common in telephone networks in the past, although part of the telephone network today is a packet-switched network.

In Figure 1.13, the four telephones at each side are connected to a switch. The switch connects a telephone set at one side to a telephone set at the other side. The thick

Figure 1.12 *A heterogeneous network made of four WANs and three LANs*

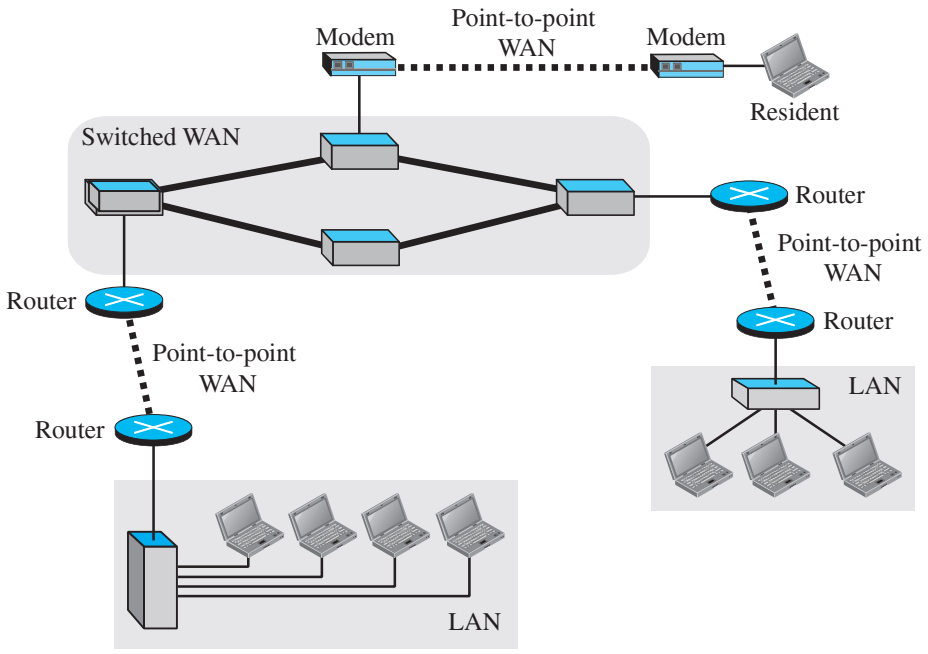
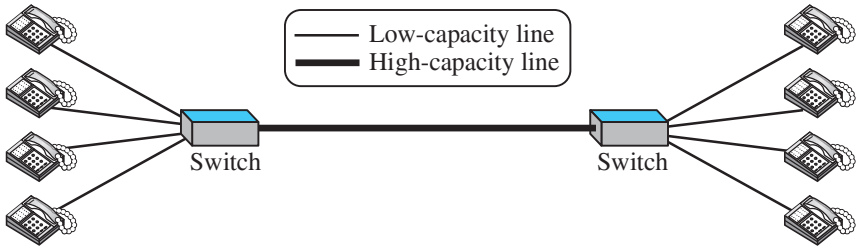


Figure 1.13 *A circuit-switched network*



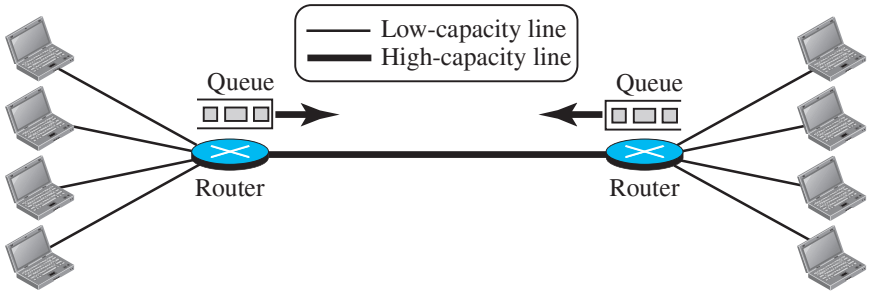
line connecting two switches is a high-capacity communication line that can handle four voice communications at the same time; the capacity can be shared between all pairs of telephone sets. The switches used in this example have forwarding tasks but no storing capability.

Let us look at two cases. In the first case, all telephone sets are busy; four people at one site are talking with four people at the other site; the capacity of the thick line is fully used. In the second case, only one telephone set at one side is connected to a telephone set at the other side; only one-fourth of the capacity of the thick line is used. This means that a circuit-switched network is efficient only when it is working at its full capacity; most of the time, it is inefficient because it is working at partial capacity. The reason that we need to make the capacity of the thick line four times the capacity of each voice line is that we do not want communication to fail when all telephone sets at one side want to be connected with all telephone sets at the other side.

Packet-Switched Network

In a computer network, the communication between the two ends is done in blocks of data called **packets.** In other words, instead of the continuous communication we see between two telephone sets when they are being used, we see the exchange of individual data packets between the two computers. This allows us to make the switches function for both storing and forwarding because a packet is an independent entity that can be stored and sent later. Figure 1.14 shows a small packet-switched network that connects four computers at one site to four computers at the other site.

Figure 1.14 *A packet-switched network*



A router in a packet-switched network has a queue that can store and forward the packet. Now assume that the capacity of the thick line is only twice the capacity of the data line connecting the computers to the routers. If only two computers (one at each site) need to communicate with each other, there is no waiting for the packets. However, if packets arrive at one router when the thick line is already working at its full capacity, the packets should be stored and forwarded in the order they arrived. The two simple examples show that a packet-switched network is more efficient than a circuit-switched network, but the packets may encounter some delays.

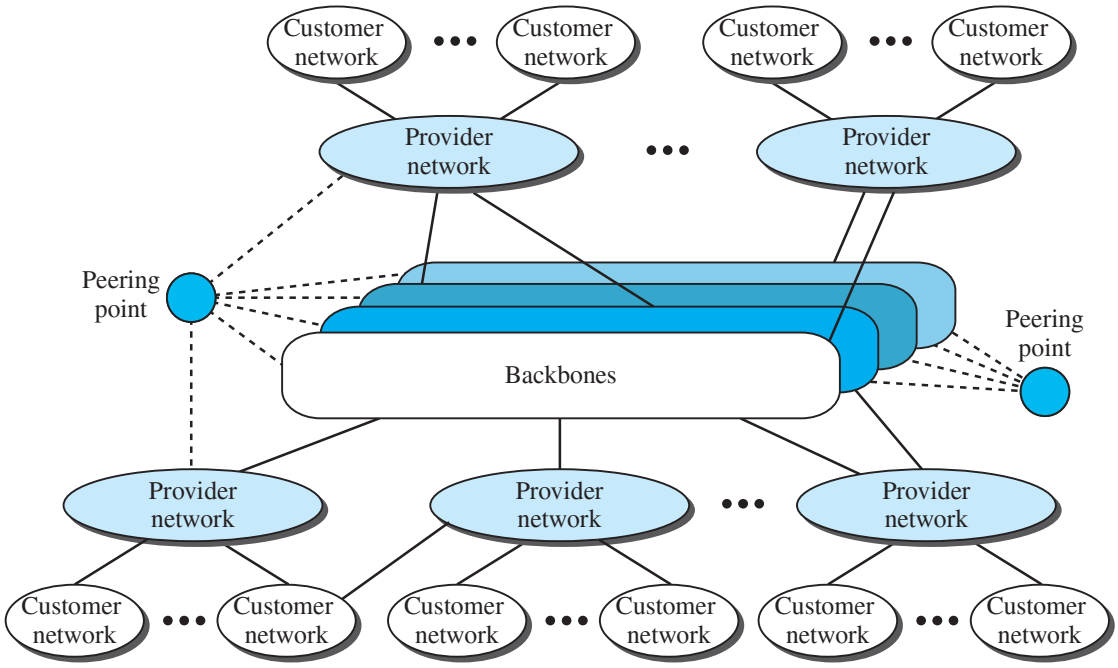
In this book, we mostly discuss packet-switched networks. In Chapter 18, we discuss packet-switched networks in more detail and discuss the performance of these networks.

1.3.4 The Internet

As we discussed before, an internet (note the lowercase *i*) is two or more networks that can communicate with each other. The most notable internet is called the **Internet** (uppercase *I*), and is composed of thousands of interconnected networks. Figure 1.15 shows a conceptual (not geographical) view of the Internet.

The figure shows the Internet as several backbones, provider networks, and customer networks. At the top level, the *backbones* are large networks owned by some communication companies such as Sprint, Verizon (MCI), AT&T, and NTT. The backbone networks are connected through some complex switching systems, called *peering points*. At the second level, there are smaller networks, called *provider networks*, that use the services of the backbones for a fee. The provider networks are connected to backbones and sometimes to other provider networks. The *customer networks* are

Figure 1.15 *The Internet today*



networks at the edge of the Internet that actually use the services provided by the Internet. They pay fees to provider networks for receiving services.

Backbones and provider networks are also called **Internet Service Providers (ISPs).** The backbones are often referred to as *international ISPs;* the provider networks are often referred to as *national* or *regional ISPs*.

1.3.5 Accessing the Internet

The Internet today is an internetwork that allows any user to become part of it. The user, however, needs to be physically connected to an ISP. The physical connection is normally done through a point-to-point WAN. In this section, we briefly describe how this can happen, but we postpone the technical details of the connection until Chapters 14 and 16.

Using Telephone Networks

Today most residences and small businesses have telephone service, which means they are connected to a telephone network. Since most telephone networks have already connected themselves to the Internet, one option for residences and small businesses to connect to the Internet is to change the voice line between the residence or business and the telephone center to a point-to-point WAN. This can be done in two ways.

- ***Dial-up service.*** The first solution is to add to the telephone line a modem that converts data to voice. The software installed on the computer dials the ISP and imitates making a telephone connection. Unfortunately, the dial-up service is

very slow, and when the line is used for Internet connection, it cannot be used for telephone (voice) connection. It is only useful for small residences. We discuss dial-up service in Chapter 14.

- ***DSL Service.*** Since the advent of the Internet, some telephone companies have upgraded their telephone lines to provide higher speed Internet services to residences or small businesses. The DSL service also allows the line to be used simultaneously for voice and data communication. We discuss DSL in Chapter 14.

Using Cable Networks

More and more residents over the last two decades have begun using cable TV services instead of antennas to receive TV broadcasting. The cable companies have been upgrading their cable networks and connecting to the Internet. A residence or a small business can be connected to the Internet by using this service. It provides a higher speed connection, but the speed varies depending on the number of neighbors that use the same cable. We discuss the cable networks in Chapter 14.

Using Wireless Networks

Wireless connectivity has recently become increasingly popular. A household or a small business can use a combination of wireless and wired connections to access the Internet. With the growing wireless WAN access, a household or a small business can be connected to the Internet through a wireless WAN. We discuss wireless access in Chapter 16.

Direct Connection to the Internet

A large organization or a large corporation can itself become a local ISP and be connected to the Internet. This can be done if the organization or the corporation leases a high-speed WAN from a carrier provider and connects itself to a regional ISP. For example, a large university with several campuses can create an internetwork and then connect the internetwork to the Internet.

1.4 INTERNET HISTORY

Now that we have given an overview of the Internet, let us give a brief history of the Internet. This brief history makes it clear how the Internet has evolved from a private network to a global one in less than 40 years.

1.4.1 Early History

There were some communication networks, such as telegraph and telephone networks, before 1960. These networks were suitable for constant-rate communication at that time, which means that after a connection was made between two users, the encoded message (telegraphy) or voice (telephony) could be exchanged. A computer network, on the other hand, should be able to handle *bursty* data, which means data received at variable rates at different times. The world needed to wait for the packet-switched network to be invented.

Birth of Packet-Switched Networks

The theory of packet switching for bursty traffic was first presented by Leonard Kleinrock in 1961 at MIT. At the same time, two other researchers, Paul Baran at Rand Institute and Donald Davies at National Physical Laboratory in England, published some papers about packet-switched networks.

ARPANET

In the mid-1960s, mainframe computers in research organizations were stand-alone devices. Computers from different manufacturers were unable to communicate with one another. The **Advanced Research Projects Agency (ARPA)** in the Department of Defense (DOD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for the **Advanced Research Projects Agency Network (ARPANET),** a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *interface message processor* (IMP). The IMPs, in turn, would be connected to each other. Each IMP had to be able to communicate with other IMPs as well as with its own attached host.

By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.

1.4.2 Birth of the Internet

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the *Internetting Project*. They wanted to link dissimilar networks so that a host on one network could communicate with a host on another. There were many problems to overcome: diverse packet sizes, diverse interfaces, and diverse transmission rates, as well as differing reliability requirements. Cerf and Kahn devised the idea of a device called a *gateway* to serve as the intermediary hardware to transfer data from one network to another.

TCP/IP

Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of data. This was a new version of NCP. This paper on transmission control protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. A radical idea was the transfer of responsibility for error correction from the IMP to the host machine. This ARPA Internet now became the focus of the communication effort. Around this time, responsibility for the ARPANET was handed over to the Defense Communication Agency (DCA).

In October 1977, an internet consisting of three different networks (ARPANET, packet radio, and packet satellite) was successfully demonstrated. Communication between networks was now possible.

Shortly thereafter, authorities made a decision to split TCP into two protocols: **Transmission Control Protocol (TCP)** and **Internet Protocol (IP).** IP would handle datagram routing while TCP would be responsible for higher level functions such as segmentation, reassembly, and error detection. The new combination became known as TCP/IP.

In 1981, under a Defence Department contract, UC Berkeley modified the UNIX operating system to include TCP/IP. This inclusion of network software along with a popular operating system did much for the popularity of internetworking. The open (non-manufacturer-specific) implementation of the Berkeley UNIX gave every manufacturer a working code base on which they could build their products.

In 1983, authorities abolished the original ARPANET protocols, and TCP/IP became the official protocol for the ARPANET. Those who wanted to use the Internet to access a computer on a different network had to be running TCP/IP.

MILNET

In 1983, ARPANET split into two networks: **Military Network (MILNET)** for military users and ARPANET for nonmilitary users.

CSNET

Another milestone in Internet history was the creation of CSNET in 1981. **Computer Science Network (CSNET)** was a network sponsored by the National Science Foundation (NSF). The network was conceived by universities that were ineligible to join ARPANET due to an absence of ties to the Department of Defense. CSNET was a less expensive network; there were no redundant links and the transmission rate was slower.

By the mid-1980s, most U.S. universities with computer science departments were part of CSNET. Other institutions and companies were also forming their own networks and using TCP/IP to interconnect. The term *Internet,* originally associated with government-funded connected networks, now referred to the connected networks using TCP/IP protocols.

NSFNET

With the success of CSNET, the NSF in 1986 sponsored the **National Science Foundation Network (NSFNET),** a backbone that connected five supercomputer centers located throughout the United States. Community networks were allowed access to this backbone, a T-1 line (see Chapter 6) with a 1.544-Mbps data rate, thus providing connectivity throughout the United States. In 1990, ARPANET was officially retired and replaced by NSFNET. In 1995, NSFNET reverted back to its original concept of a research network.

ANSNET

In 1991, the U.S. government decided that NSFNET was not capable of supporting the rapidly increasing Internet traffic. Three companies, IBM, Merit, and Verizon, filled the void by forming a nonprofit organization called Advanced Network & Services (ANS) to build a new, high-speed Internet backbone called **Advanced Network Services Network (ANSNET).**

1.4.3 Internet Today

Today, we witness a rapid growth both in the infrastructure and new applications. The Internet today is a set of pier networks that provide services to the whole world. What has made the Internet so popular is the invention of new applications.

World Wide Web

The 1990s saw the explosion of Internet applications due to the emergence of the World Wide Web (WWW). The Web was invented at CERN by Tim Berners-Lee. This invention has added the commercial applications to the Internet.

Multimedia

Recent developments in the multimedia applications such as voice over IP (telephony), video over IP (Skype), view sharing (YouTube), and television over IP (PPLive) has increased the number of users and the amount of time each user spends on the network. We discuss multimedia in Chapter 28.

Peer-to-Peer Applications

Peer-to-peer networking is also a new area of communication with a lot of potential. We introduce some peer-to-peer applications in Chapter 29.

1.5 STANDARDS AND ADMINISTRATION

In the discussion of the Internet and its protocol, we often see a reference to a standard or an administration entity. In this section, we introduce these standards and administration entities for those readers that are not familiar with them; the section can be skipped if the reader is familiar with them.

1.5.1 Internet Standards

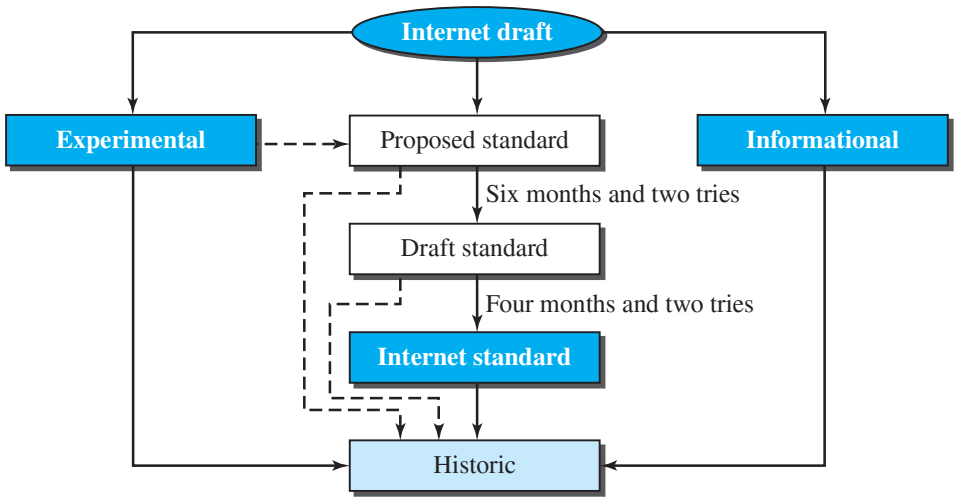
An **Internet standard** is a thoroughly tested specification that is useful to and adhered to by those who work with the Internet. It is a formalized regulation that must be followed. There is a strict procedure by which a specification attains Internet standard status. A specification begins as an Internet draft. An **Internet draft** is a working document (a work in progress) with no official status and a six-month lifetime. Upon recommendation from the Internet authorities, a draft may be published as a **Request for Comment (RFC).** Each RFC is edited, assigned a number, and made available to all interested parties. RFCs go through maturity levels and are categorized according to their requirement level.

Maturity Levels

An RFC, during its lifetime, falls into one of six *maturity levels:* proposed standard, draft standard, Internet standard, historic, experimental, and informational (see Figure 1.16).

- ***Proposed Standard.*** A proposed standard is a specification that is stable, well understood, and of sufficient interest to the Internet community. At this level, the specification is usually tested and implemented by several different groups.

Figure 1.16 *Maturity levels of an RFC*



- ***Draft Standard.*** A proposed standard is elevated to draft standard status after at least two successful independent and interoperable implementations. Barring difficulties, a draft standard, with modifications if specific problems are encountered, normally becomes an Internet standard.
- ***Internet Standard.*** A draft standard reaches Internet standard status after demonstrations of successful implementation.
- ***Historic.*** The historic RFCs are significant from a historical perspective. They either have been superseded by later specifications or have never passed the necessary maturity levels to become an Internet standard.
- ***Experimental.*** An RFC classified as experimental describes work related to an experimental situation that does not affect the operation of the Internet. Such an RFC should not be implemented in any functional Internet service.
- ***Informational.*** An RFC classified as informational contains general, historical, or tutorial information related to the Internet. It is usually written by someone in a non-Internet organization, such as a vendor.

Requirement Levels

RFCs are classified into five *requirement levels:* required, recommended, elective, limited use, and not recommended.

- ***Required.*** An RFC is labeled *required* if it must be implemented by all Internet systems to achieve minimum conformance. For example, IP and ICMP (Chapter 19) are required protocols.
- ***Recommended.*** An RFC labeled recommended is not required for minimum conformance; it is recommended because of its usefulness. For example, FTP (Chapter 26) and TELNET (Chapter 26) are recommended protocols.
- ***Elective.*** An RFC labeled elective is not required and not recommended. However, a system can use it for its own benefit.

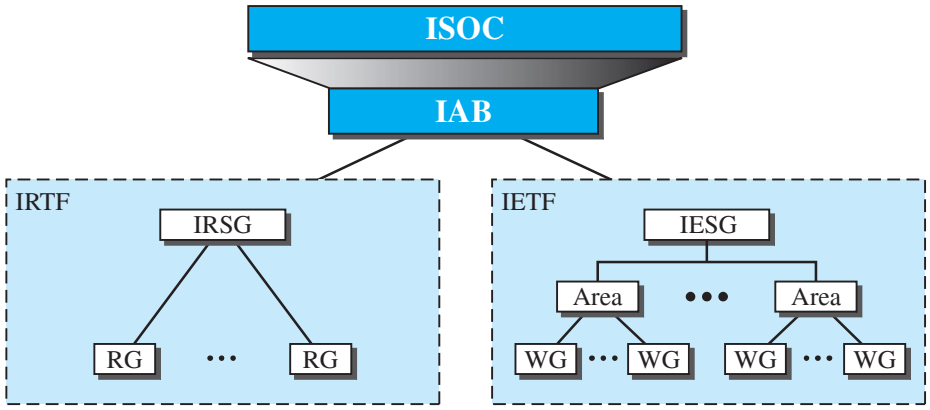
- ***Limited Use.*** An RFC labeled limited use should be used only in limited situations. Most of the experimental RFCs fall under this category.
- ***Not Recommended.*** An RFC labeled not recommended is inappropriate for general use. Normally a historic (deprecated) RFC may fall under this category.

RFCs can be found at http://www.rfc-editor.org.

1.5.2 Internet Administration

The Internet, with its roots primarily in the research domain, has evolved and gained a broader user base with significant commercial activity. Various groups that coordinate Internet issues have guided this growth and development. Appendix G gives the addresses, e-mail addresses, and telephone numbers for some of these groups. Figure 1.17 shows the general organization of Internet administration.

Figure 1.17 *Internet administration*



ISOC

The **Internet Society (ISOC)** is an international, nonprofit organization formed in 1992 to provide support for the Internet standards process. ISOC accomplishes this through maintaining and supporting other Internet administrative bodies such as IAB, IETF, IRTF, and IANA (see the following sections). ISOC also promotes research and other scholarly activities relating to the Internet.

IAB

The **Internet Architecture Board (IAB)** is the technical advisor to the ISOC. The main purposes of the IAB are to oversee the continuing development of the TCP/IP Protocol Suite and to serve in a technical advisory capacity to research members of the Internet community. IAB accomplishes this through its two primary components, the Internet Engineering Task Force (IETF) and the Internet Research Task Force (IRTF). Another responsibility of the IAB is the editorial management of the RFCs, described

earlier. IAB is also the external liaison between the Internet and other standards organizations and forums.

IETF

The **Internet Engineering Task Force (IETF)** is a forum of working groups managed by the Internet Engineering Steering Group (IESG). IETF is responsible for identifying operational problems and proposing solutions to these problems. IETF also develops and reviews specifications intended as Internet standards. The working groups are collected into areas, and each area concentrates on a specific topic. Currently nine areas have been defined. The areas include applications, protocols, routing, network management next generation (IPng), and security.

IRTF

The **Internet Research Task Force (IRTF)** is a forum of working groups managed by the Internet Research Steering Group (IRSG). IRTF focuses on long-term research topics related to Internet protocols, applications, architecture, and technology.

1.6 END-CHAPTER MATERIALS

1.6.1 Recommended Reading

For more details about subjects discussed in this chapter, we recommend the following books. The items enclosed in brackets [. . .] refer to the reference list at the end of the book.

Books

The introductory materials covered in this chapter can be found in [Sta04] and [PD03]. [Tan03] also discusses standardization.

1.6.2 Key Terms

Advanced Network Services Network (ANSNET)
Advanced Research Projects Agency (ARPA)
Advanced Research Projects Agency Network (ARPANET)
American Standard Code for Information Interchange (ASCII)
audio
backbone
Basic Latin
bus topology
circuit-switched network
code
Computer Science Network (CSNET)
data
data communications
delay
full-duplex mode
half-duplex mode
hub
image
internet
Internet
Internet Architecture Board (IAB)
Internet draft
Internet Engineering Task Force (IETF)
Internet Research Task Force (IRTF)
Internet Service Provider (ISP)
Internet Society (ISOC)
Internet standard
internetwork
local area network (LAN)
mesh topology
message

Military Network (MILNET)
multipoint or multidrop connection
National Science Foundation Network (NSFNET)
network
node
packet
packet-switched network
performance
physical topology
point-to-point connection
protocol
Request for Comment (RFC)
RGB
ring topology
simplex mode
star topology
switched network
TCP/IP protocol suite
telecommunication
throughput
Transmission Control Protocol/ Internet Protocol (TCP/IP)
transmission medium
Unicode
video
wide area network (WAN)
YCM

1.6.3 Summary

Data communications are the transfer of data from one device to another via some form of transmission medium. A data communications system must transmit data to the correct destination in an accurate and timely manner. The five components that make up a data communications system are the message, sender, receiver, medium, and protocol. Text, numbers, images, audio, and video are different forms of information. Data flow between two devices can occur in one of three ways: simplex, half-duplex, or full-duplex.

A network is a set of communication devices connected by media links. In a point-to-point connection, two and only two devices are connected by a dedicated link. In a multipoint connection, three or more devices share a link. Topology refers to the physical or logical arrangement of a network. Devices may be arranged in a mesh, star, bus, or ring topology.

A network can be categorized as a local area network or a wide area network. A LAN is a data communication system within a building, plant, or campus, or between nearby buildings. A WAN is a data communication system spanning states, countries, or the whole world. An internet is a network of networks. The Internet is a collection of many separate networks.

The Internet history started with the theory of packet switching for bursty traffic. The history continued when The ARPA was interested in finding a way to connect computers so that the researchers they funded could share their findings, resulting in the creation of ARPANET. The Internet was born when Cerf and Kahn devised the idea of a device called a *gateway* to serve as the intermediary hardware to transfer data from one network to another. The TCP/IP protocol suite paved the way for creation of today's Internet. The invention of WWW, the use of multimedia, and peer-to-peer communication helps the growth of the Internet.

An Internet standard is a thoroughly tested specification. An Internet draft is a working document with no official status and a six-month lifetime. A draft may be published as a Request for Comment (RFC). RFCs go through maturity levels and are categorized according to their requirement level. The Internet administration has

evolved with the Internet. ISOC promotes research and activities. IAB is the technical advisor to the ISOC. IETF is a forum of working groups responsible for operational problems. IRTF is a forum of working groups focusing on long-term research topics.

1.7 PRACTICE SET

1.7.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

1.7.2 Questions

Q1-1. Identify the five components of a data communications system.

Q1-2. What are the three criteria necessary for an effective and efficient network?

Q1-3. What are the advantages of a multipoint connection over a point-to-point one?

Q1-4. What are the two types of line configuration?

Q1-5. Categorize the four basic topologies in terms of line configuration.

Q1-6. What is the difference between half-duplex and full-duplex transmission modes?

Q1-7. Name the four basic network topologies, and cite an advantage of each type.

Q1-8. For n devices in a network, what is the number of cable links required for a mesh, ring, bus, and star topology?

Q1-9. What are some of the factors that determine whether a communication system is a LAN or WAN?

Q1-10. What is an internet? What is the Internet?

Q1-11. Why are protocols needed?

Q1-12. In a LAN with a link-layer switch (Figure 1.8b), Host 1 wants to send a message to Host 3. Since communication is through the link-layer switch, does the switch need to have an address? Explain.

Q1-13. How many point-to-point WANs are needed to connect n LANs if each LAN should be able to directly communicate with any other LAN?

Q1-14. When we use local telephones to talk to a friend, are we using a circuit-switched network or a packet-switched network?

Q1-15. When a resident uses a dial-up or DLS service to connect to the Internet, what is the role of the telephone company?

Q1-16. What is the first principle we discussed in this chapter for protocol layering that needs to be followed to make the communication bidirectional?

Q1-17. Explain the difference between an Internet draft and a proposed standard.

Q1-18. Explain the difference between a required RFC and a recommended RFC.

Q1-19. Explain the difference between the duties of the IETF and IRTF.

1.7.3 Problems

P1-1. What is the maximum number of characters or symbols that can be represented by Unicode?

P1-2. A color image uses 16 bits to represent a pixel. What is the maximum number of different colors that can be represented?

P1-3. Assume six devices are arranged in a mesh topology. How many cables are needed? How many ports are needed for each device?

P1-4. For each of the following four networks, discuss the consequences if a connection fails.

- **a.** Five devices arranged in a mesh topology
- **b.** Five devices arranged in a star topology (not counting the hub)
- **c.** Five devices arranged in a bus topology
- **d.** Five devices arranged in a ring topology

P1-5. We have two computers connected by an Ethernet hub at home. Is this a LAN or a WAN? Explain the reason.

P1-6. In the ring topology in Figure 1.7, what happens if one of the stations is unplugged?

P1-7. In the bus topology in Figure 1.6, what happens if one of the stations is unplugged?

P1-8. Performance is inversely related to delay. When we use the Internet, which of the following applications are more sensitive to delay?

- **a.** Sending an e-mail
- **b.** Copying a file
- **c.** Surfing the Internet

P1-9. When a party makes a local telephone call to another party, is this a point-to-point or multipoint connection? Explain the answer.

P1-10. Compare the telephone network and the Internet. What are the similarities? What are the differences?

1.8 SIMULATION EXPERIMENTS

1.8.1 Applets

One of the ways to show the network protocols in action or visually see the solution to some examples is through the use of interactive animation. We have created some Java applets to show some of the main concepts discussed in this chapter. It is strongly recommended that the students activate these applets on the book website and carefully examine the protocols in action. However, note that applets have been created only for some chapters, not all (see the book website).

1.8.2 Lab Assignments

Experiments with networks and network equipment can be done using at least two methods. In the first method, we can create an isolated networking laboratory and use

networking hardware and software to simulate the topics discussed in each chapter. We can create an internet and send and receive packets from any host to another. The flow of packets can be observed and the performance can be measured. Although the first method is more effective and more instructional, it is expensive to implement and not all institutions are ready to invest in such an exclusive laboratory.

In the second method, we can use the Internet, the largest network in the world, as our virtual laboratory. We can send and receive packets using the Internet. The existence of some free-downloadable software allows us to capture and examine the packets exchanged. We can analyze the packets to see how theoretical aspects of networking are put into action. Although the second method may not be as effective as the first method, in that we cannot control and change the packet routes to see how the Internet behaves, the method is much cheaper to implement. It does not need a physical networking lab; it can be implemented using our desktop or laptop. The required software is also free to download.

There are many programs and utilities available for Windows and UNIX operating systems that allow us to sniff, capture, trace, and analyze packets that are exchanged between our computer and the Internet. Some of these, such as *Wireshark* and *PingPlotter,* have graphical user interface (GUI); others, such as *traceroute, nslookup*, *dig*, *ipconfig,* and *ifconfig,* are network administration command-line utilities. Any of these programs and utilities can be a valuable debugging tool for network administrators and educational tool for computer network students.

In this book, we mostly use Wireshark for lab assignments, although we occasionally use other tools. It captures live packet data from a network interface and displays them with detailed protocol information. Wireshark, however, is a passive analyzer. It only "measures" things from the network without manipulating them; it doesn't send packets on the network or perform other active operations. Wireshark is not an intrusion detection tool either. It does not give warning about any network intrusion. It, nevertheless, can help network administrators or network security engineers to figure out what is going on inside a network and to troubleshoot network problems. In addition to being an indispensable tool for network administrators and security engineers, Wireshark is a valuable tool for protocol developers, who may use it to debug protocol implementations, and a great educational tool for computer networking students who can use it to see details of protocol operations in real time. However, note that we can use lab assignments only with a few chapters.

Lab1-1. In this lab assignment we learn how to download and install Wireshark. The instructions for downloading and installing the software are posted on the book website in the lab section for Chapter 1. In this document, we also discuss the general idea behind the software, the format of its window, and how to use it. The full study of this lab prepares the student to use Wireshark in the lab assignments for other chapters.

CHAPTER 2

Network Models

The second chapter is a preparation for the rest of the book. The next five parts of the book is devoted to one of the layers in the TCP/IP protocol suite. In this chapter, we first discuss the idea of network models in general and the TCP/IP protocol suite in particular.

Two models have been devised to define computer network operations: the TCP/IP protocol suite and the OSI model. In this chapter, we first discuss a general subject, protocol layering, which is used in both models. We then concentrate on the TCP/IP protocol suite, on which the book is based. The OSI model is briefly discuss for comparison with the TCP/IP protocol suite.

- The first section introduces the concept of protocol layering using two scenarios. The section also discusses the two principles upon which the protocol layering is based. The first principle dictates that each layer needs to have two opposite tasks. The second principle dictates that the corresponding layers should be identical. The section ends with a brief discussion of logical connection between two identical layers in protocol layering. Throughout the book, we need to distinguish between logical and physical connections.
- The second section discusses the five layers of the TCP/IP protocol suite. We show how packets in each of the five layers (physical, data-link, network, transport, and application) are named. We also mention the addressing mechanism used in each layer. Each layer of the TCP/IP protocol suite is a subject of a part of the book. In other words, each layer is discussed in several chapters; this section is just an introduction and preparation.
- The third section gives a brief discussion of the OSI model. This model was never implemented in practice, but a brief discussion of the model and its comparison with the TCP/IP protocol suite may be useful to better understand the TCP/IP protocol suite. In this section we also give a brief reason for the OSI model's lack of success.

2.1 PROTOCOL LAYERING

We defined the term *protocol* in Chapter 1. In data communication and networking, a protocol defines the rules that both the sender and receiver and all intermediate devices need to follow to be able to communicate effectively. When communication is simple, we may need only one simple protocol; when the communication is complex, we may need to divide the task between different layers, in which case we need a protocol at each layer, or **protocol layering.**

2.1.1 Scenarios

Let us develop two simple scenarios to better understand the need for protocol layering.

First Scenario

In the first scenario, communication is so simple that it can occur in only one layer. Assume Maria and Ann are neighbors with a lot of common ideas. Communication between Maria and Ann takes place in one layer, face to face, in the same language, as shown in Figure 2.1.

Figure 2.1 *A single-layer protocol*

Even in this simple scenario, we can see that a set of rules needs to be followed. First, Maria and Ann know that they should greet each other when they meet. Second, they know that they should confine their vocabulary to the level of their friendship. Third, each party knows that she should refrain from speaking when the other party is speaking. Fourth, each party knows that the conversation should be a dialog, not a monolog: both should have the opportunity to talk about the issue. Fifth, they should exchange some nice words when they leave.

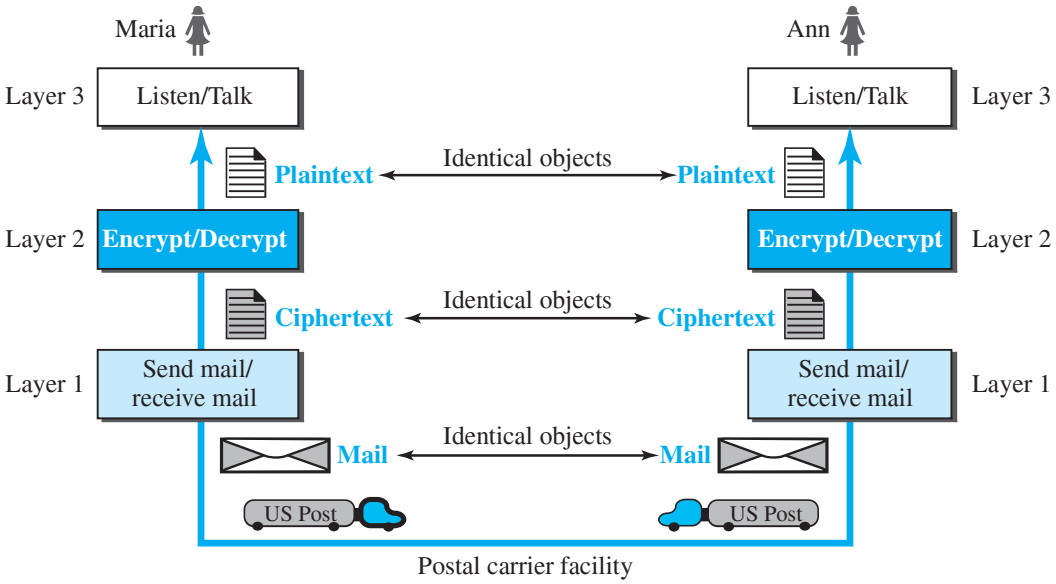
We can see that the protocol used by Maria and Ann is different from the communication between a professor and the students in a lecture hall. The communication in the second case is mostly monolog; the professor talks most of the time unless a student has a question, a situation in which the protocol dictates that she should raise her hand and wait for permission to speak. In this case, the communication is normally very formal and limited to the subject being taught.

Second Scenario

In the second scenario, we assume that Ann is offered a higher-level position in her company, but needs to move to another branch located in a city very far from Maria. The two friends still want to continue their communication and exchange ideas because

they have come up with an innovative project to start a new business when they both retire. They decide to continue their conversation using regular mail through the post office. However, they do not want their ideas to be revealed by other people if the letters are intercepted. They agree on an encryption/decryption technique. The sender of the letter encrypts it to make it unreadable by an intruder; the receiver of the letter decrypts it to get the original letter. We discuss the encryption/decryption methods in Chapter 31, but for the moment we assume that Maria and Ann use one technique that makes it hard to decrypt the letter if one does not have the key for doing so. Now we can say that the communication between Maria and Ann takes place in three layers, as shown in Figure 2.2. We assume that Ann and Maria each have three machines (or robots) that can perform the task at each layer.

Figure 2.2 *A three-layer protocol*



Let us assume that Maria sends the first letter to Ann. Maria talks to the machine at the third layer as though the machine is Ann and is listening to her. The third layer machine listens to what Maria says and creates the plaintext (a letter in English), which is passed to the second layer machine. The second layer machine takes the plaintext, encrypts it, and creates the ciphertext, which is passed to the first layer machine. The first layer machine, presumably a robot, takes the ciphertext, puts it in an envelope, adds the sender and receiver addresses, and mails it.

At Ann's side, the first layer machine picks up the letter from Ann's mail box, recognizing the letter from Maria by the sender address. The machine takes out the ciphertext from the envelope and delivers it to the second layer machine. The second layer machine decrypts the message, creates the plaintext, and passes the plaintext to the third-layer machine. The third layer machine takes the plaintext and reads it as though Maria is speaking.

Protocol layering enables us to divide a complex task into several smaller and simpler tasks. For example, in Figure 2.2, we could have used only one machine to do the job of all three machines. However, if Maria and Ann decide that the encryption/decryption done by the machine is not enough to protect their secrecy, they would have to change the whole machine. In the present situation, they need to change only the second layer machine; the other two can remain the same. This is referred to as *modularity*. Modularity in this case means independent layers. A layer (module) can be defined as a black box with inputs and outputs, without concern about how inputs are changed to outputs. If two machines provide the same outputs when given the same inputs, they can replace each other. For example, Ann and Maria can buy the second layer machine from two different manufacturers. As long as the two machines create the same ciphertext from the same plaintext and vice versa, they do the job.

One of the advantages of protocol layering is that it allows us to separate the services from the implementation. A layer needs to be able to receive a set of services from the lower layer and to give the services to the upper layer; we don't care about how the layer is implemented. For example, Maria may decide not to buy the machine (robot) for the first layer; she can do the job herself. As long as Maria can do the tasks provided by the first layer, in both directions, the communication system works.

Another advantage of protocol layering, which cannot be seen in our simple examples but reveals itself when we discuss protocol layering in the Internet, is that communication does not always use only two end systems; there are intermediate systems that need only some layers, but not all layers. If we did not use protocol layering, we would have to make each intermediate system as complex as the end systems, which makes the whole system more expensive.

Is there any disadvantage to protocol layering? One can argue that having a single layer makes the job easier. There is no need for each layer to provide a service to the upper layer and give service to the lower layer. For example, Ann and Maria could find or build one machine that could do all three tasks. However, as mentioned above, if one day they found that their code was broken, each would have to replace the whole machine with a new one instead of just changing the machine in the second layer.

2.1.2 Principles of Protocol Layering

Let us discuss two principles of protocol layering.

First Principle

The first principle dictates that if we want bidirectional communication, we need to make each layer so that it is able to perform two opposite tasks, one in each direction. For example, the third layer task is to listen (in one direction) and *talk* (in the other direction). The second layer needs to be able to encrypt and decrypt. The first layer needs to send and receive mail.

Second Principle

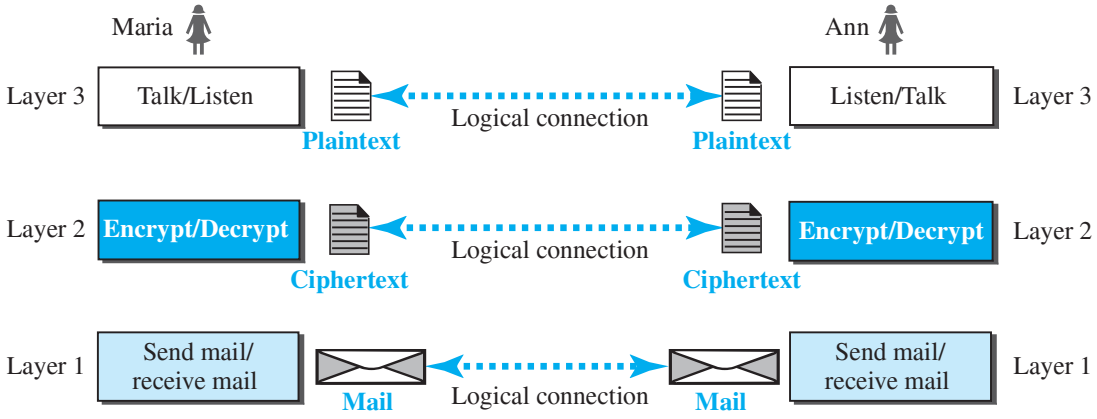
The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical. For example, the object under layer 3 at both sites should be a plaintext letter. The object under layer 2 at

both sites should be a ciphertext letter. The object under layer 1 at both sites should be a piece of mail.

2.1.3 Logical Connections

After following the above two principles, we can think about logical connection between each layer as shown in Figure 2.3. This means that we have layer-to-layer communication. Maria and Ann can think that there is a logical (imaginary) connection at each layer through which they can send the object created from that layer. We will see that the concept of logical connection will help us better understand the task of layering we encounter in data communication and networking.

Figure 2.3 *Logical connection between peer layers*



2.2 TCP/IP PROTOCOL SUITE

Now that we know about the concept of protocol layering and the logical communication between layers in our second scenario, we can introduce the TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP is a protocol suite (a set of protocols organized in different layers) used in the Internet today. It is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality. The term *hierarchical* means that each upper level protocol is supported by the services provided by one or more lower level protocols. The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model. Figure 2.4 shows both configurations.

2.2.1 Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch. We also assume that the links are connected by one router, as shown in Figure 2.5.

Figure 2.4 *Layers in the TCP/IP protocol suite*

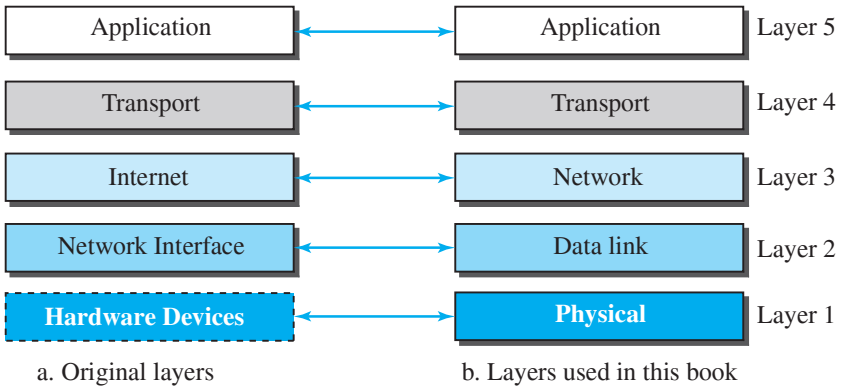
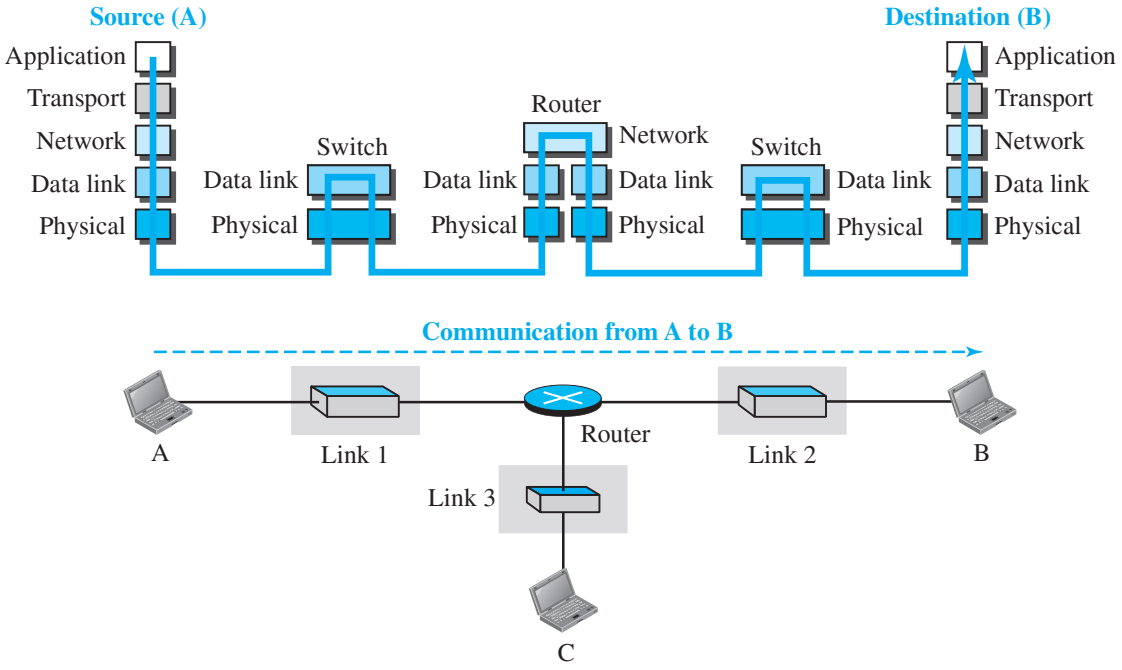


Figure 2.5 *Communication through an internet*



Let us assume that computer A communicates with computer B. As the figure shows, we have five communicating devices in this communication: source host (computer A), the link-layer switch in link 1, the router, the link-layer switch in link 2, and the destination host (computer B). Each device is involved with a set of layers depending on the role of the device in the internet. The two hosts are involved in all five layers; the source host needs to create a message in the application layer and send it down the layers so that it is physically sent to the destination host. The destination host needs to receive the communication at the physical layer and then deliver it through the other layers to the application layer.

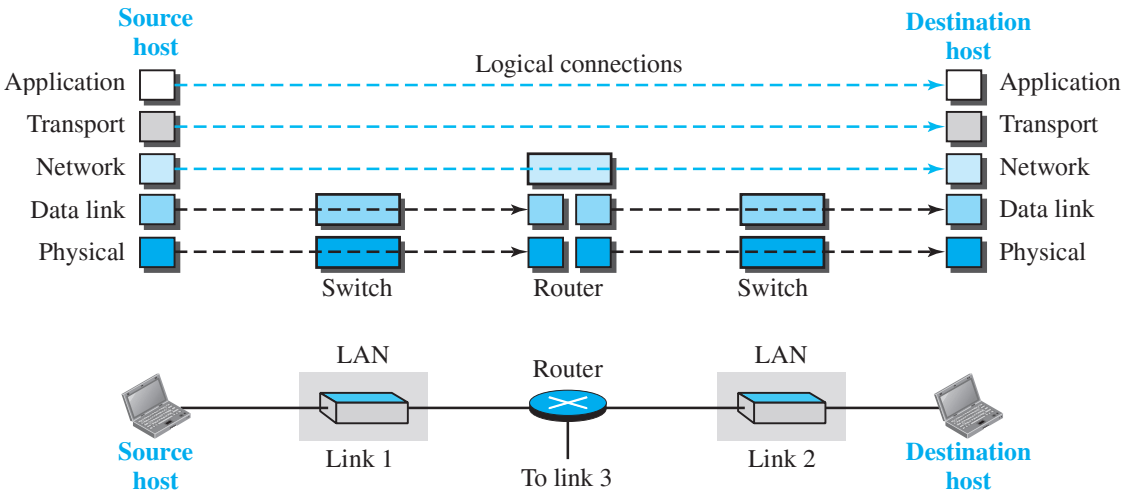
The router is involved in only three layers; there is no transport or application layer in a router as long as the router is used only for routing. Although a router is always involved in one network layer, it is involved in *n* combinations of link and physical layers in which *n* is the number of links the router is connected to. The reason is that each link may use its own data-link or physical protocol. For example, in the above figure, the router is involved in three links, but the message sent from source A to destination B is involved in two links. Each link may be using different link-layer and physical-layer protocols; the router needs to receive a packet from link 1 based on one pair of protocols and deliver it to link 2 based on another pair of protocols.

A link-layer switch in a link, however, is involved only in two layers, data-link and physical. Although each switch in the above figure has two different connections, the connections are in the same link, which uses only one set of protocols. This means that, unlike a router, a link-layer switch is involved only in one data-link and one physical layer.

2.2.2 Layers in the TCP/IP Protocol Suite

After the above introduction, we briefly discuss the functions and duties of layers in the TCP/IP protocol suite. Each layer is discussed in detail in the next five parts of the book. To better understand the duties of each layer, we need to think about the logical connections between layers. Figure 2.6 shows logical connections in our simple internet.

Figure 2.6 *Logical connections between layers of the TCP/IP protocol suite*



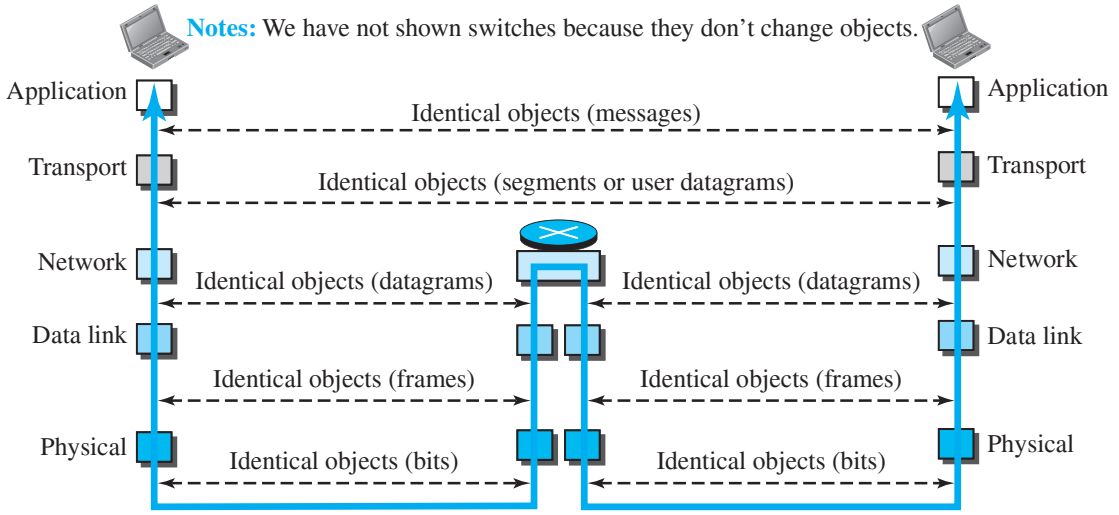
Using logical connections makes it easier for us to think about the duty of each layer. As the figure shows, the duty of the application, transport, and network layers is end-to-end. However, the duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router. In other words, the domain of duty of the top three layers is the internet, and the domain of duty of the two lower layers is the link.

Another way of thinking of the logical connections is to think about the data unit created from each layer. In the top three layers, the data unit (packets) should not be

changed by any router or link-layer switch. In the bottom two layers, the packet created by the host is changed only by the routers, not by the link-layer switches.

Figure 2.7 shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device.

Figure 2.7 *Identical objects in the TCP/IP protocol suite*



Note that, although the logical connection at the network layer is between the two hosts, we can only say that identical objects exist between two hops in this case because a router may fragment the packet at the network layer and send more packets than received (see fragmentation in Chapter 19). Note that the link between two hops does not change the object.

2.2.3 Description of Each Layer

After understanding the concept of logical communication, we are ready to briefly discuss the duty of each layer. Our discussion in this chapter will be very brief, but we come back to the duty of each layer in next five parts of the book.

Physical Layer

We can say that the physical layer is responsible for carrying individual bits in a frame across the link. Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a logical communication because there is another, hidden layer, the transmission media, under the physical layer. Two devices are connected by a transmission medium (cable or air). We need to know that the transmission medium does not carry bits; it carries electrical or optical signals. So the bits received in a frame from the data-link layer are transformed and sent through the transmission media, but we can think that the logical unit between two physical layers in two devices is a *bit*. There are several protocols that transform a bit to a signal. We discuss them in Part II when we discuss the physical layer and the transmission media.

Data-link Layer

We have seen that an internet is made up of several links (LANs and WANs) connected by routers. There may be several overlapping sets of links that a datagram can travel from the host to the destination. The routers are responsible for choosing the *best* links. However, when the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link. The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN. We can also have different protocols used with any link type. In each case, the data-link layer is responsible for moving the packet through the link.

TCP/IP does not define any specific protocol for the data-link layer. It supports all the standard and proprietary protocols. Any protocol that can take the datagram and carry it through the link suffices for the network layer. The data-link layer takes a datagram and encapsulates it in a packet called a *frame*.

Each link-layer protocol may provide a different service. Some link-layer protocols provide complete error detection and correction, some provide only error correction. We discuss wired links in Chapters 13 and 14 and wireless links in Chapters 15 and 16.

Network Layer

The network layer is responsible for creating a connection between the source computer and the destination computer. The communication at the network layer is host-to-host. However, since there can be several routers from the source to the destination, the routers in the path are responsible for choosing the best route for each packet. We can say that the network layer is responsible for host-to-host communication and routing the packet through possible routes. Again, we may ask ourselves why we need the network layer. We could have added the routing duty to the transport layer and dropped this layer. One reason, as we said before, is the separation of different tasks between different layers. The second reason is that the routers do not need the application and transport layers. Separating the tasks allows us to use fewer protocols on the routers.

The network layer in the Internet includes the main protocol, Internet Protocol (IP), that defines the format of the packet, called a datagram at the network layer. IP also defines the format and the structure of addresses used in this layer. IP is also responsible for routing a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path.

IP is a connectionless protocol that provides no flow control, no error control, and no congestion control services. This means that if any of theses services is required for an application, the application should rely only on the transport-layer protocol. The network layer also includes unicast (one-to-one) and multicast (one-to-many) routing protocols. A routing protocol does not take part in routing (it is the responsibility of IP), but it creates forwarding tables for routers to help them in the routing process.

The network layer also has some auxiliary protocols that help IP in its delivery and routing tasks. The Internet Control Message Protocol (ICMP) helps IP to report some problems when routing a packet. The Internet Group Management Protocol (IGMP) is another protocol that helps IP in multitasking. The Dynamic Host Configuration Protocol (DHCP) helps IP to get the network-layer address for a host. The Address Resolution Protocol (ARP) is a protocol that helps IP to find the link-layer address of a host or

a router when its network-layer address is given. ARP is discussed in Chapter 9, ICMP in Chapter 19, and IGMP in Chapter 21.

Transport Layer

The logical connection at the transport layer is also end-to-end. The transport layer at the source host gets the message from the application layer, encapsulates it in a transport-layer packet (called a *segment* or a *user datagram* in different protocols) and sends it, through the logical (imaginary) connection, to the transport layer at the destination host. In other words, the transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host. We may ask why we need an end-to-end transport layer when we already have an end-to-end application layer. The reason is the separation of tasks and duties, which we discussed earlier. The transport layer should be independent of the application layer. In addition, we will see that we have more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement.

As we said, there are a few transport-layer protocols in the Internet, each designed for some specific task. The main protocol, Transmission Control Protocol (TCP), is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data. It creates a logical pipe between two TCPs for transferring a stream of bytes. TCP provides flow control (matching the sending data rate of the source host with the receiving data rate of the destination host to prevent overwhelming the destination), error control (to guarantee that the segments arrive at the destination without error and resending the corrupted ones), and congestion control to reduce the loss of segments due to congestion in the network. The other common protocol, User Datagram Protocol (UDP), is a connectionless protocol that transmits user datagrams without first creating a logical connection. In UDP, each user datagram is an independent entity without being related to the previous or the next one (the meaning of the term *connectionless*). UDP is a simple protocol that does not provide flow, error, or congestion control. Its simplicity, which means small overhead, is attractive to an application program that needs to send short messages and cannot afford the retransmission of the packets involved in TCP, when a packet is corrupted or lost. A new protocol, Stream Control Transmission Protocol (SCTP) is designed to respond to new applications that are emerging in the multimedia. We will discuss UDP, TCP, and SCTP in Chapter 24.

Application Layer

As Figure 2.6 shows, the logical connection between the two application layers is end-to-end. The two application layers exchange *messages* between each other as though there were a bridge between the two layers. However, we should know that the communication is done through all the layers.

Communication at the application layer is between two *processes* (two programs running at this layer). To communicate, a process sends a request to the other process and receives a response. Process-to-process communication is the duty of the application layer. The application layer in the Internet includes many predefined protocols, but

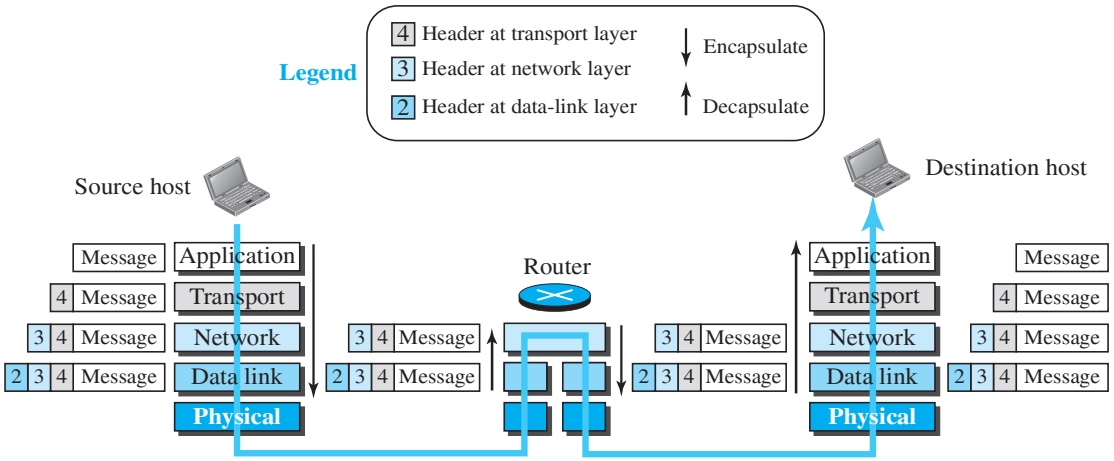
a user can also create a pair of processes to be run at the two hosts. In Chapter 25, we explore this situation.

The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW). The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e-mail) service. The File Transfer Protocol (FTP) is used for transferring files from one host to another. The Terminal Network (TELNET) and Secure Shell (SSH) are used for accessing a site remotely. The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels. The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer. The Internet Group Management Protocol (IGMP) is used to collect membership in a group. We discuss most of these protocols in Chapter 26 and some in other chapters.

2.2.4 Encapsulation and Decapsulation

One of the important concepts in protocol layering in the Internet is encapsulation/decapsulation. Figure 2.8 shows this concept for the small internet in Figure 2.5.

Figure 2.8 *Encapsulation/Decapsulation*



We have not shown the layers for the link-layer switches because no encapsulation/decapsulation occurs in this device. In Figure 2.8, we show the encapsulation in the source host, decapsulation in the destination host, and encapsulation and decapsulation in the router.

Encapsulation at the Source Host

At the source, we have only encapsulation.

1. At the application layer, the data to be exchanged is referred to as a *message*. A message normally does not contain any header or trailer, but if it does, we refer to the whole as the message. The message is passed to the transport layer.
2. The transport layer takes the message as the payload, the load that the transport layer should take care of. It adds the transport layer header to the payload, which contains the identifiers of the source and destination application programs that

want to communicate plus some more information that is needed for the end-to-end delivery of the message, such as information needed for flow, error control, or congestion control. The result is the transport-layer packet, which is called the *segment* (in TCP) and the *user datagram* (in UDP). The transport layer then passes the packet to the network layer.

3. The network layer takes the transport-layer packet as data or payload and adds its own header to the payload. The header contains the addresses of the source and destination hosts and some more information used for error checking of the header, fragmentation information, and so on. The result is the network-layer packet, called a *datagram*. The network layer then passes the packet to the data-link layer.
4. The data-link layer takes the network-layer packet as data or payload and adds its own header, which contains the link-layer addresses of the host or the next hop (the router). The result is the link-layer packet, which is called a *frame*. The frame is passed to the physical layer for transmission.

Decapsulation and Encapsulation at the Router

At the router, we have both decapsulation and encapsulation because the router is connected to two or more links.

1. After the set of bits are delivered to the data-link layer, this layer decapsulates the datagram from the frame and passes it to the network layer.
2. The network layer only inspects the source and destination addresses in the datagram header and consults its forwarding table to find the next hop to which the datagram is to be delivered. The contents of the datagram should not be changed by the network layer in the router unless there is a need to fragment the datagram if it is too big to be passed through the next link. The datagram is then passed to the data-link layer of the next link.
3. The data-link layer of the next link encapsulates the datagram in a frame and passes it to the physical layer for transmission.

Decapsulation at the Destination Host

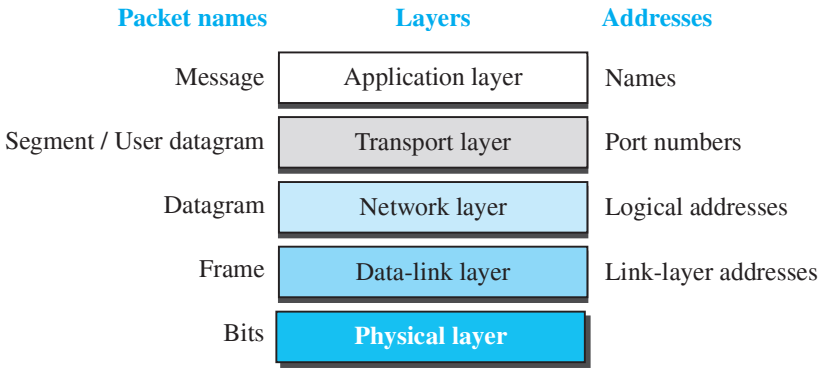
At the destination host, each layer only decapsulates the packet received, removes the payload, and delivers the payload to the next-higher layer protocol until the message reaches the application layer. It is necessary to say that decapsulation in the host involves error checking.

2.2.5 Addressing

It is worth mentioning another concept related to protocol layering in the Internet, *addressing*. As we discussed before, we have logical communication between pairs of layers in this model. Any communication that involves two parties needs two addresses: source address and destination address. Although it looks as if we need five pairs of addresses, one pair per layer, we normally have only four because the physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address. Figure 2.9 shows the addressing at each layer.

As the figure shows, there is a relationship between the layer, the address used in that layer, and the packet name at that layer. At the application layer, we normally use names to define the site that provides services, such as *someorg.com*, or the e-mail

Figure 2.9 *Addressing in the TCP/IP protocol suite*

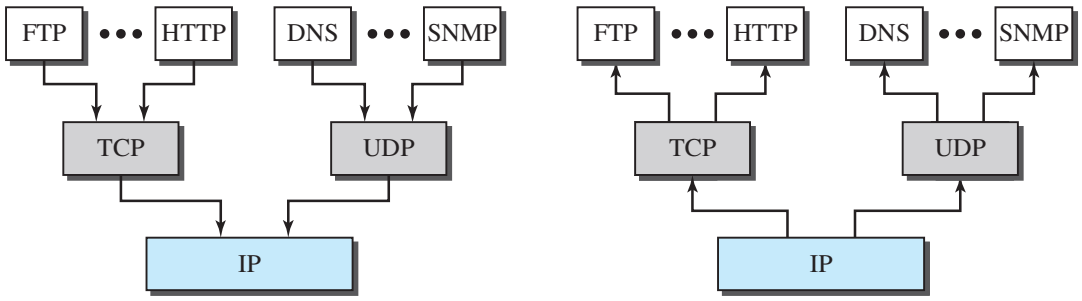


address, such as *somebody@coldmail.com.* At the transport layer, addresses are called port numbers, and these define the application-layer programs at the source and destination. Port numbers are local addresses that distinguish between several programs running at the same time. At the network-layer, the addresses are global, with the whole Internet as the scope. A network-layer address uniquely defines the connection of a device to the Internet. The link-layer addresses, sometimes called MAC addresses, are locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN). We will come back to these addresses in future chapters.

2.2.6 Multiplexing and Demultiplexing

Since the TCP/IP protocol suite uses several protocols at some layers, we can say that we have multiplexing at the source and demultiplexing at the destination. Multiplexing in this case means that a protocol at a layer can encapsulate a packet from several next-higher layer protocols (one at a time); demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time). Figure 2.10 shows the concept of multiplexing and demultiplexing at the three upper layers.

Figure 2.10 *Multiplexing and demultiplexing*



a. Multiplexing at source

b. Demultiplexing at destination

To be able to multiplex and demultiplex, a protocol needs to have a field in its header to identify to which protocol the encapsulated packets belong. At the transport

layer, either UDP or TCP can accept a message from several application-layer protocols. At the network layer, IP can accept a segment from TCP or a user datagram from UDP. IP can also accept a packet from other protocols such as ICMP, IGMP, and so on. At the data-link layer, a frame may carry the payload coming from IP or other protocols such as ARP (see Chapter 9).

2.3 THE OSI MODEL

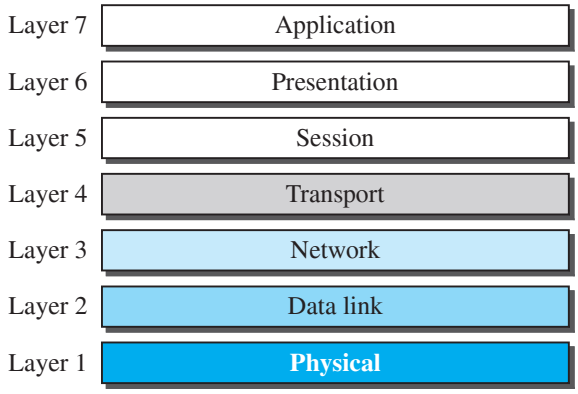
Although, when speaking of the Internet, everyone talks about the TCP/IP protocol suite, this suite is not the only suite of protocols defined. Established in 1947, the **International Organization for Standardization (ISO)** is a multinational body dedicated to worldwide agreement on international standards. Almost three-fourths of the countries in the world are represented in the ISO. An ISO standard that covers all aspects of network communications is the **Open Systems Interconnection (OSI) model.** It was first introduced in the late 1970s.

ISO is the organization; OSI is the model.

An *open system* is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable. The OSI model was intended to be the basis for the creation of the protocols in the OSI stack.

The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network (see Figure 2.11).

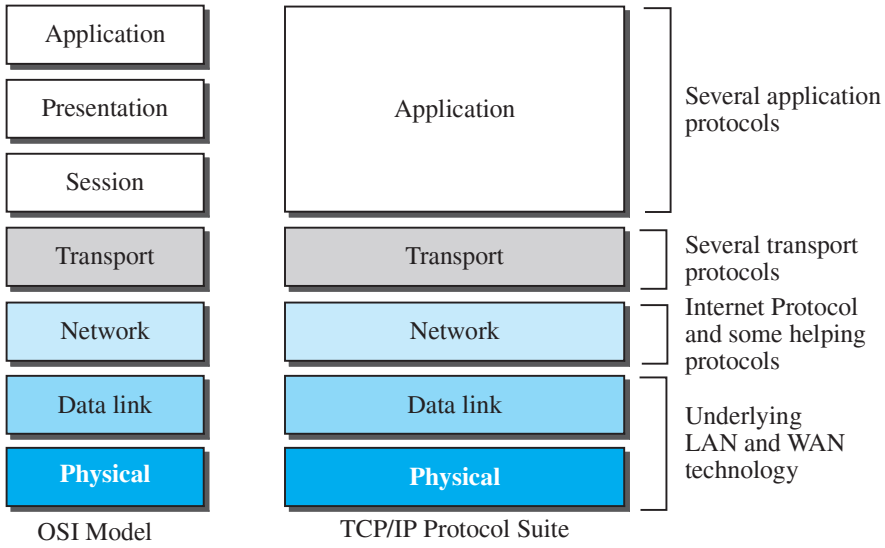
Figure 2.11 *The OSI model*



2.3.1 OSI versus TCP/IP

When we compare the two models, we find that two layers, session and presentation, are missing from the TCP/IP protocol suite. These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model. The application layer in the suite is usually considered to be the combination of three layers in the OSI model, as shown in Figure 2.12.

Figure 2.12 *TCP/IP and OSI model*



Two reasons were mentioned for this decision. First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport-layer protocols. Second, the application layer is not only one piece of software. Many applications can be developed at this layer. If some of the functionalities mentioned in the session and presentation layers are needed for a particular application, they can be included in the development of that piece of software.

2.3.2 Lack of OSI Model's Success

The OSI model appeared after the TCP/IP protocol suite. Most experts were at first excited and thought that the TCP/IP protocol would be fully replaced by the OSI model. This did not happen for several reasons, but we describe only three, which are agreed upon by all experts in the field. First, OSI was completed when TCP/IP was fully in place and a lot of time and money had been spent on the suite; changing it would cost a lot. Second, some layers in the OSI model were never fully defined. For example, although the services provided by the presentation and the session layers were listed in the document, actual protocols for these two layers were not fully defined, nor were they fully described, and the corresponding software was not fully

developed. Third, when OSI was implemented by an organization in a different application, it did not show a high enough level of performance to entice the Internet authority to switch from the TCP/IP protocol suite to the OSI model.

2.4 END-CHAPTER MATERIALS

2.4.1 Recommended Reading

For more details about subjects discussed in this chapter, we recommend the following books, and RFCs. The items enclosed in brackets refer to the reference list at the end of the book.

Books and Papers

Several books and papers give a thorough coverage about the materials discussed in this chapter: [Seg 98], [Lei et al. 98], [Kle 04], [Cer 89], and [Jen et al. 86].

RFCs

Two RFCs in particular discuss the TCP/IP suite: RFC 791 (IP) and RFC 817 (TCP). In future chapters we list different RFCs related to each protocol in each layer.

2.4.2 Key Terms

International Organization for Standardization (ISO)
Open Systems Interconnection (OSI) model
protocol layering

2.4.3 Summary

A protocol is a set of rules that governs communication. In protocol layering, we need to follow two principles to provide bidirectional communication. First, each layer needs to perform two opposite tasks. Second, two objects under each layer at both sides should be identical. In a protocol layering, we need to distinguish between a logical connection and a physical connection. Two protocols at the same layer can have a logical connection; a physical connection is only possible through the physical layers.

TCP/IP is a hierarchical protocol suite made of five layers: physical, data link, network, transport, and application. The physical layer coordinates the functions required to transmit a bit stream over a physical medium. The data-link layer is responsible for delivering data units from one station to the next without errors. The network layer is responsible for the source-to-destination delivery of a packet across multiple network links. The transport layer is responsible for the process-to-process delivery of the entire message. The application layer enables the users to access the network.

Four levels of addresses are used in an internet following the TCP/IP protocols: physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses. The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. The IP address uniquely defines a host on the Internet. The port address identifies a process on a host. A specific address is a user-friendly address.

Another model that defines protocol layering is the Open Systems Interconnection (OSI) model. Two layers in the OSI model, session and presentation, are missing from the TCP/IP protocol suite. These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model. The application layer in the suite is usually considered to be the combination of three layers in the OSI model. The OSI model did not replace the TCP/IP protocol suite because it was completed when TCP/IP was fully in place and because some layers in the OSI model were never fully defined.

2.5 PRACTICE SET

2.5.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

2.5.2 Questions

Q2-1. What is the first principle we discussed in this chapter for protocol layering that needs to be followed to make the communication bidirectional?

Q2-2. Which layers of the TCP/IP protocol suite are involved in a link-layer switch?

Q2-3. A router connects three links (networks). How many of each of the following layers can the router be involved with?

a. physical layer **b.** data-link layer **c.** network layer

Q2-4. In the TCP/IP protocol suite, what are the identical objects at the sender and the receiver sites when we think about the logical connection at the application layer?

Q2-5. A host communicates with another host using the TCP/IP protocol suite. What is the unit of data sent or received at each of the following layers?

a. application layer **b.** network layer **c.** data-link layer

Q2-6. Which of the following data units is encapsulated in a frame?

a. a user datagram **b.** a datagram **c.** a segment

Q2-7. Which of the following data units is decapsulated from a user datagram?

a. a datagram **b.** a segment **c.** a message

Q2-8. Which of the following data units has an application-layer message plus the header from layer 4?

a. a frame **b.** a user datagram **c.** a bit

Q2-9. List some application-layer protocols mentioned in this chapter.

Q2-10. If a port number is 16 bits (2 bytes), what is the minimum header size at the transport layer of the TCP/IP protocol suite?

Q2-11. What are the types of addresses (identifiers) used in each of the following layers?

a. application layer **b.** network layer **c.** data-link layer

Q2-12. When we say that the transport layer multiplexes and demultiplexes application-layer messages, do we mean that a transport-layer protocol can combine several messages from the application layer in one packet? Explain.

Q2-13. Can you explain why we did not mention multiplexing/demultiplexing services for the application layer?

Q2-14. Assume we want to connect two isolated hosts together to let each host communicate with the other. Do we need a link-layer switch between the two? Explain.

Q2-15. If there is a single path between the source host and the destination host, do we need a router between the two hosts?

2.5.3 Problems

P2-1. Answer the following questions about Figure 2.2 when the communication is from Maria to Ann:

a. What is the service provided by layer 1 to layer 2 at Maria's site?

b. What is the service provided by layer 1 to layer 2 at Ann's site?

P2-2. Answer the following questions about Figure 2.2 when the communication is from Maria to Ann:

a. What is the service provided by layer 2 to layer 3 at Maria's site?

b. What is the service provided by layer 2 to layer 3 at Ann's site?

P2-3. Assume that the number of hosts connected to the Internet at year 2010 is five hundred million. If the number of hosts increases only 20 percent per year, what is the number of hosts in year 2020?

P2-4. Assume a system uses five protocol layers. If the application program creates a message of 100 bytes and each layer (including the fifth and the first) adds a header of 10 bytes to the data unit, what is the efficiency (the ratio of application-layer bytes to the number of bytes transmitted) of the system?

P2-5. Assume we have created a packet-switched internet. Using the TCP/IP protocol suite, we need to transfer a huge file. What are the advantage and disadvantage of sending large packets?

P2-6. Match the following to one or more layers of the TCP/IP protocol suite:

a. route determination

b. connection to transmission media

c. providing services for the end user

P2-7. Match the following to one or more layers of the TCP/IP protocol suite:

a. creating user datagrams

b. responsibility for handling frames between adjacent nodes

c. transforming bits to electromagnetic signals

P2-8. In Figure 2.10, when the IP protocol decapsulates the transport-layer packet, how does it know to which upper-layer protocol (UDP or TCP) the packet should be delivered?

P2-9. Assume a private internet uses three different protocols at the data-link layer (L1, L2, and L3). Redraw Figure 2.10 with this assumption. Can we say that,

in the data-link layer, we have demultiplexing at the source node and multiplexing at the destination node?

P2-10. Assume that a private internet requires that the messages at the application layer be encrypted and decrypted for security purposes. If we need to add some information about the encryption/decryption process (such as the algorithms used in the process), does it mean that we are adding one layer to the TCP/IP protocol suite? Redraw the TCP/IP layers (Figure 2.4 part b) if you think so.

P2-11. Protocol layering can be found in many aspects of our lives such as air travelling. Imagine you make a round-trip to spend some time on vacation at a resort. You need to go through some processes at your city airport before flying. You also need to go through some processes when you arrive at the resort airport. Show the protocol layering for the round trip using some layers such as baggage checking/claiming, boarding/unboarding, takeoff/landing.

P2-12. The presentation of data is becoming more and more important in today's Internet. Some people argue that the TCP/IP protocol suite needs to add a new layer to take care of the presentation of data. If this new layer is added in the future, where should its position be in the suite? Redraw Figure 2.4 to include this layer.

P2-13. In an internet, we change the LAN technology to a new one. Which layers in the TCP/IP protocol suite need to be changed?

P2-14. Assume that an application-layer protocol is written to use the services of UDP. Can the application-layer protocol uses the services of TCP without change?

P2-15. Using the internet in Figure 1.11 (Chapter 1) in the text, show the layers of the TCP/IP protocol suite and the flow of data when two hosts, one on the west coast and the other on the east coast, exchange messages.

PART II

Physical Layer

In the second part of the book, we discuss the physical layer, including the transmission media that is connected to the physical layer. The part is made of six chapters. The first introduces the entities involved in the physical layer. The next two chapters cover transmission. The following chapter discusses how to use the available bandwidth. The transmission media alone occupy all of the next chapter. Finally, the last chapter discusses switching, which can occur in any layer, but we introduce the topic in this part of the book.

CHAPTER 3

Introduction to Physical Layer

One of the major functions of the physical layer is to move data in the form of electromagnetic signals across a transmission medium. Whether you are collecting numerical statistics from another computer, sending animated pictures from a design workstation, or causing a bell to ring at a distant control center, you are working with the transmission of **data** across network connections.

Generally, the data usable to a person or application are not in a form that can be transmitted over a network. For example, a photograph must first be changed to a form that transmission media can accept. Transmission media work by conducting energy along a physical path. For transmission, data needs to be changed to **signals.**

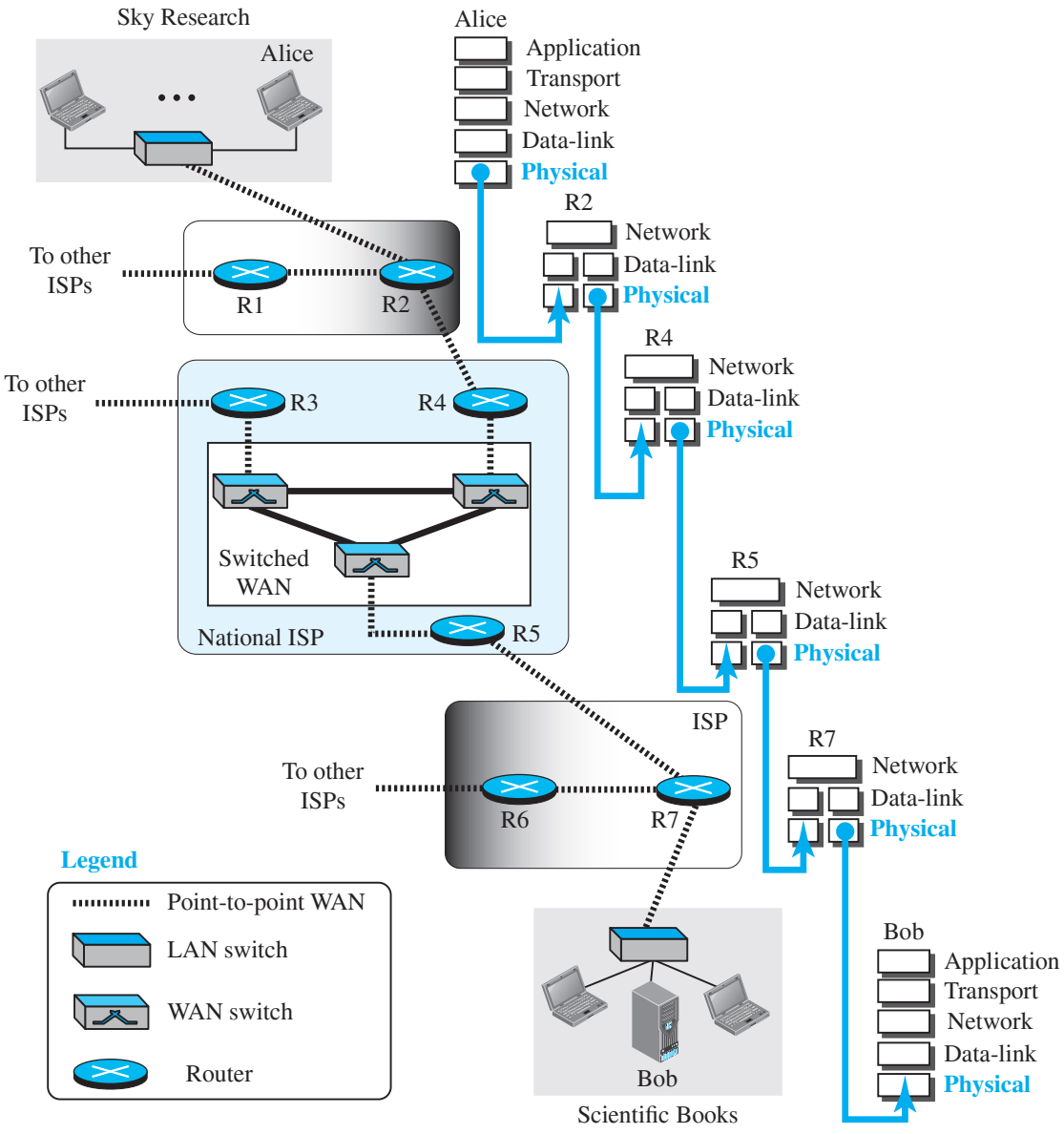
This chapter is divided into six sections:

- The first section shows how data and signals can be either analog or digital. Analog refers to an entity that is continuous; digital refers to an entity that is discrete.
- The second section shows that only periodic analog signals can be used in data communication. The section discusses simple and composite signals. The attributes of analog signals such as period, frequency, and phase are also explained.
- The third section shows that only nonperiodic digital signals can be used in data communication. The attributes of a digital signal such as bit rate and bit length are discussed. We also show how digital data can be sent using analog signals. Baseband and broadband transmission are also discussed in this section.
- The fourth section is devoted to transmission impairment. The section shows how attenuation, distortion, and noise can impair a signal.
- The fifth section discusses the data rate limit: how many bits per second we can send with the available channel. The data rates of noiseless and noisy channels are examined and compared.
- The sixth section discusses the performance of data transmission. Several channel measurements are examined including bandwidth, throughput, latency, and jitter. Performance is an issue that is revisited in several future chapters.

3.1 DATA AND SIGNALS

Figure 3.1 shows a scenario in which a scientist working in a research company, Sky Research, needs to order a book related to her research from an online bookseller, Scientific Books.

Figure 3.1 *Communication at the physical layer*



We can think of five different levels of communication between Alice, the computer on which our scientist is working, and Bob, the computer that provides online service. Communication at application, transport, network, or data-link is *logical*; communication at the physical layer is *physical*. For simplicity, we have shown only

host-to-router, router-to-router, and router-to-host, but the switches are also involved in the physical communication.

Although Alice and Bob need to exchange *data*, communication at the physical layer means exchanging *signals*. Data need to be transmitted and received, but the media have to change data to signals. Both data and the signals that represent them can be either **analog** or **digital** in form.

3.1.1 Analog and Digital Data

Data can be analog or digital. The term **analog data** refers to information that is continuous; **digital data** refers to information that has discrete states. For example, an analog clock that has hour, minute, and second hands gives information in a continuous form; the movements of the hands are continuous. On the other hand, a digital clock that reports the hours and the minutes will change suddenly from 8:05 to 8:06.

Analog data, such as the sounds made by a human voice, take on continuous values. When someone speaks, an analog wave is created in the air. This can be captured by a microphone and converted to an analog signal or sampled and converted to a digital signal.

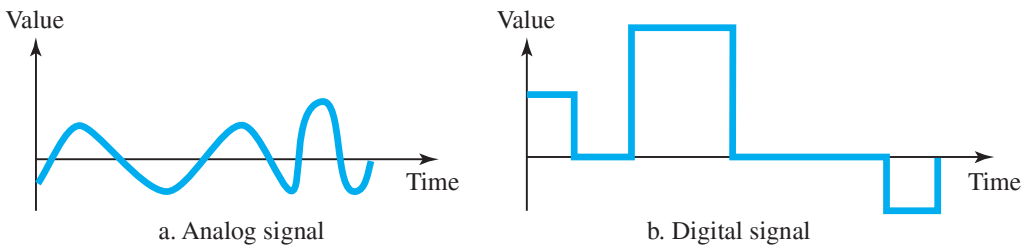
Digital data take on discrete values. For example, data are stored in computer memory in the form of 0s and 1s. They can be converted to a digital signal or modulated into an analog signal for transmission across a medium.

3.1.2 Analog and Digital Signals

Like the data they represent, **signals** can be either analog or digital. An **analog signal** has infinitely many levels of intensity over a period of time. As the wave moves from value *A* to value *B*, it passes through and includes an infinite number of values along its path. A **digital signal,** on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and 0.

The simplest way to show signals is by plotting them on a pair of perpendicular axes. The vertical axis represents the value or strength of a signal. The horizontal axis represents time. Figure 3.2 illustrates an analog signal and a digital signal. The curve representing the analog signal passes through an infinite number of points. The vertical lines of the digital signal, however, demonstrate the sudden jump that the signal makes from value to value.

Figure 3.2 *Comparison of analog and digital signals*



a. Analog signal b. Digital signal

3.1.3 Periodic and Nonperiodic

Both analog and digital signals can take one of two forms: *periodic* or *nonperiodic* (sometimes referred to as *aperiodic;* the prefix *a* in Greek means "non").

A **periodic signal** completes a pattern within a measurable time frame, called a **period,** and repeats that pattern over subsequent identical periods. The completion of one full pattern is called a **cycle.** A **nonperiodic signal** changes without exhibiting a pattern or cycle that repeats over time.

Both analog and digital signals can be periodic or nonperiodic. In data communications, we commonly use periodic analog signals and nonperiodic digital signals, as we will see in future chapters.

In data communications, we commonly use periodic analog signals and nonperiodic digital signals.

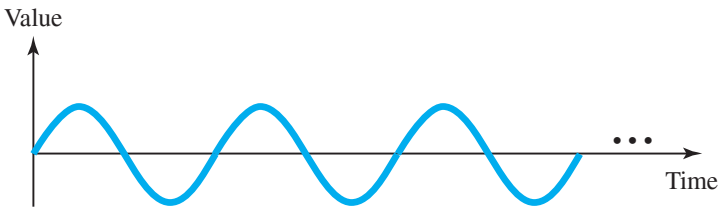
3.2 PERIODIC ANALOG SIGNALS

Periodic analog signals can be classified as simple or composite. A simple periodic analog signal, a **sine wave,** cannot be decomposed into simpler signals. A composite periodic analog signal is composed of multiple sine waves.

3.2.1 Sine Wave

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow. Figure 3.3 shows a sine wave. Each cycle consists of a single arc above the time axis followed by a single arc below it.

Figure 3.3 *A sine wave*



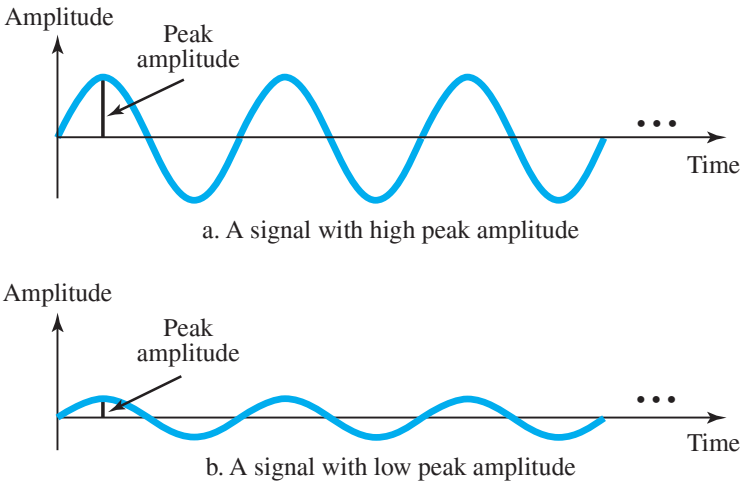
We discuss a mathematical approach to sine waves in Appendix E.

A sine wave can be represented by three parameters: the *peak amplitude,* the *frequency,* and the *phase*. These three parameters fully describe a sine wave.

Peak Amplitude

The **peak amplitude** of a signal is the absolute value of its highest intensity, proportional to the energy it carries. For electric signals, peak amplitude is normally measured in *volts*. Figure 3.4 shows two signals and their peak amplitudes.

Figure 3.4 *Two signals with the same phase and frequency, but different amplitudes*



Example 3.1

The power in your house can be represented by a sine wave with a peak amplitude of 155 to 170 V. However, it is common knowledge that the voltage of the power in U.S. homes is 110 to 120 V. This discrepancy is due to the fact that these are root mean square (rms) values. The signal is squared and then the average amplitude is calculated. The peak value is equal to $2^{1/2} \times$ rms value.

Example 3.2

The voltage of a battery is a constant; this constant value can be considered a sine wave, as we will see later. For example, the peak value of an AA battery is normally 1.5 V.

Period and Frequency

Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle. **Frequency** refers to the number of periods in 1 s. Note that period and frequency are just one characteristic defined in two ways. Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show.

$$f = \frac{1}{T} \quad \text{and} \quad T = \frac{1}{f}$$

Frequency and period are the inverse of each other.

Figure 3.5 shows two signals and their frequencies. Period is formally expressed in seconds. Frequency is formally expressed in **Hertz (Hz),** which is cycle per second. Units of period and frequency are shown in Table 3.1.

Figure 3.5 *Two signals with the same amplitude and phase, but different frequencies*

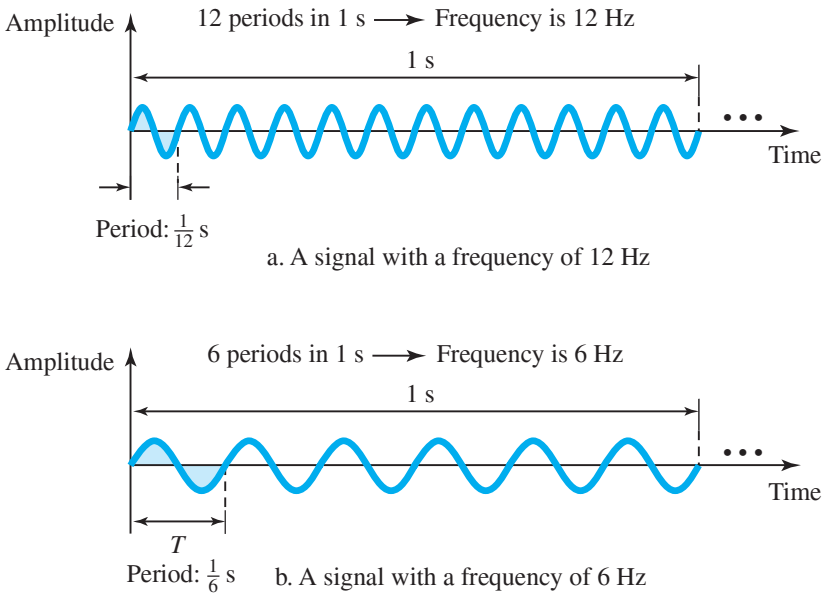


Table 3.1 *Units of period and frequency*

Period		*Frequency*	
Unit	*Equivalent*	*Unit*	*Equivalent*
Seconds (s)	1 s	Hertz (Hz)	1 Hz
Milliseconds (ms)	10^{-3} s	Kilohertz (kHz)	10^{3} Hz
Microseconds (μs)	10^{-6} s	Megahertz (MHz)	10^{6} Hz
Nanoseconds (ns)	10^{-9} s	Gigahertz (GHz)	10^{9} Hz
Picoseconds (ps)	10^{-12} s	Terahertz (THz)	10^{12} Hz

Example 3.3

The power we use at home has a frequency of 60 Hz (50 Hz in Europe). The period of this sine wave can be determined as follows:

$$T = \frac{1}{f} = \frac{1}{60} = 0.0166 \text{ s} = 0.0166 \times 10^3 \text{ ms} = 16.6 \text{ ms}$$

This means that the period of the power for our lights at home is 0.0116 s, or 16.6 ms. Our eyes are not sensitive enough to distinguish these rapid changes in amplitude.

Example 3.4

Express a period of 100 ms in microseconds.

Solution

From Table 3.1 we find the equivalents of 1 ms (1 ms is 10^{-3} s) and 1 s (1 s is 10^{6} μs). We make the following substitutions:

$$100 \text{ ms} = 100 \times 10^{-3} \text{ s} = 100 \times 10^{-3} \times 10^{6} \text{ μs} = 10^{2} \times 10^{-3} \times 10^{6} \text{ μs} = 10^{5} \text{ μs}$$

Example 3.5

The period of a signal is 100 ms. What is its frequency in kilohertz?

Solution

First we change 100 ms to seconds, and then we calculate the frequency from the period (1 Hz = 10^{-3} kHz).

$$100 \text{ ms} = 100 \times 10^{-3} \text{ s} = 10^{-1} \text{ s}$$

$$f = \frac{1}{T} = \frac{1}{10^{-1}} \text{Hz} = 10 \text{ Hz} = 10 \times 10^{-3} \text{ kHz} = 10^{-2} \text{ kHz}$$

More About Frequency

We already know that frequency is the relationship of a signal to time and that the frequency of a wave is the number of cycles it completes in 1 s. But another way to look at frequency is as a measurement of the rate of change. Electromagnetic signals are oscillating waveforms; that is, they fluctuate continuously and predictably above and below a mean energy level. A 40-Hz signal has one-half the frequency of an 80-Hz signal; it completes 1 cycle in twice the time of the 80-Hz signal, so each cycle also takes twice as long to change from its lowest to its highest voltage levels. Frequency, therefore, though described in cycles per second (hertz), is a general measurement of the rate of change of a signal with respect to time.

Frequency is the rate of change with respect to time. Change in a short span of time means high frequency. Change over a long span of time means low frequency.

If the value of a signal changes over a very short span of time, its frequency is high. If it changes over a long span of time, its frequency is low.

Two Extremes

What if a signal does not change at all? What if it maintains a constant voltage level for the entire time it is active? In such a case, its frequency is zero. Conceptually, this idea is a simple one. If a signal does not change at all, it never completes a cycle, so its frequency is 0 Hz.

But what if a signal changes instantaneously? What if it jumps from one level to another in no time? Then its frequency is infinite. In other words, when a signal changes instantaneously, its period is zero; since frequency is the inverse of period, in this case, the frequency is 1/0, or infinite (unbounded).

If a signal does not change at all, its frequency is zero.
If a signal changes instantaneously, its frequency is infinite.

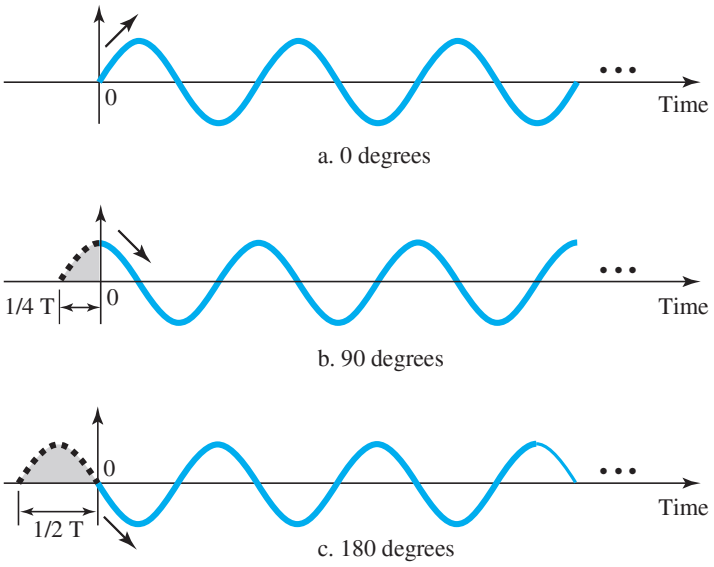
3.2.2 Phase

The term **phase,** or phase shift, describes the position of the waveform relative to time 0. If we think of the wave as something that can be shifted backward or forward along the time axis, phase describes the amount of that shift. It indicates the status of the first cycle.

Phase describes the position of the waveform relative to time 0.

Phase is measured in degrees or radians [360º is 2π rad; 1º is 2π/360 rad, and 1 rad is 360/(2π)]. A phase shift of 360º corresponds to a shift of a complete period; a phase shift of 180° corresponds to a shift of one-half of a period; and a phase shift of 90º corresponds to a shift of one-quarter of a period (see Figure 3.6).

Figure 3.6 *Three sine waves with the same amplitude and frequency, but different phases*



Looking at Figure 3.6, we can say that

- **a.** A sine wave with a phase of 0° starts at time 0 with a zero amplitude. The amplitude is increasing.
- **b.** A sine wave with a phase of 90° starts at time 0 with a peak amplitude. The amplitude is decreasing.
- **c.** A sine wave with a phase of 180° starts at time 0 with a zero amplitude. The amplitude is decreasing.

Another way to look at the phase is in terms of shift or offset. We can say that

- **a.** A sine wave with a phase of 0° is not shifted.
- **b.** A sine wave with a phase of 90° is shifted to the left by $\frac{1}{4}$ cycle. However, note that the signal does not really exist before time 0.
- **c.** A sine wave with a phase of 180° is shifted to the left by $\frac{1}{2}$ cycle. However, note that the signal does not really exist before time 0.

Example 3.6

A sine wave is offset $\frac{1}{6}$ cycle with respect to time 0. What is its phase in degrees and radians?

Solution

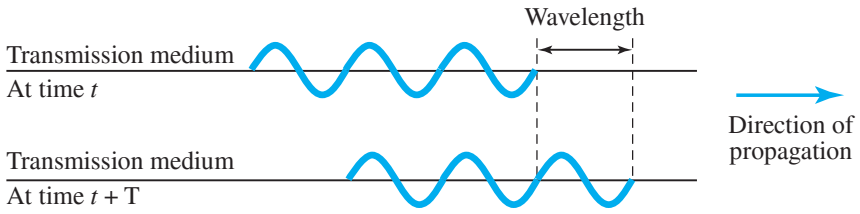
We know that 1 complete cycle is 360°. Therefore, $\frac{1}{6}$ cycle is

$$\frac{1}{6} \times 360 = 60° = 60 \times \frac{2\pi}{360} \text{ rad} = \frac{\pi}{3} \text{ rad} = 1.046 \text{ rad}$$

3.2.3 Wavelength

Wavelength is another characteristic of a signal traveling through a transmission medium. Wavelength binds the period or the frequency of a simple sine wave to the **propagation speed** of the medium (see Figure 3.7).

Figure 3.7 *Wavelength and period*



While the frequency of a signal is independent of the medium, the wavelength depends on both the frequency and the medium. Wavelength is a property of any type of signal. In data communications, we often use wavelength to describe the transmission of light in an optical fiber. The wavelength is the distance a simple signal can travel in one period.

Wavelength can be calculated if one is given the propagation speed (the speed of light) and the period of the signal. However, since period and frequency are related to each other, if we represent wavelength by λ, propagation speed by c (speed of light), and frequency by f, we get

$$\textbf{Wavelength} = \textbf{(propagation speed)} \times \textbf{period} = \frac{\textbf{propagation speed}}{\textbf{frequency}}$$

$$\lambda = \frac{c}{f}$$

The propagation speed of electromagnetic signals depends on the medium and on the frequency of the signal. For example, in a vacuum, light is propagated with a speed of 3×10^8 m/s. That speed is lower in air and even lower in cable.

The wavelength is normally measured in micrometers (microns) instead of meters. For example, the wavelength of red light (frequency = 4×10^{14}) in air is

$$\lambda = \frac{c}{f} = \frac{3 \times 10^8}{4 \times 10^{14}} = 0.75 \times 10^{-6}\ \text{m} = 0.75\ \mu\text{m}$$

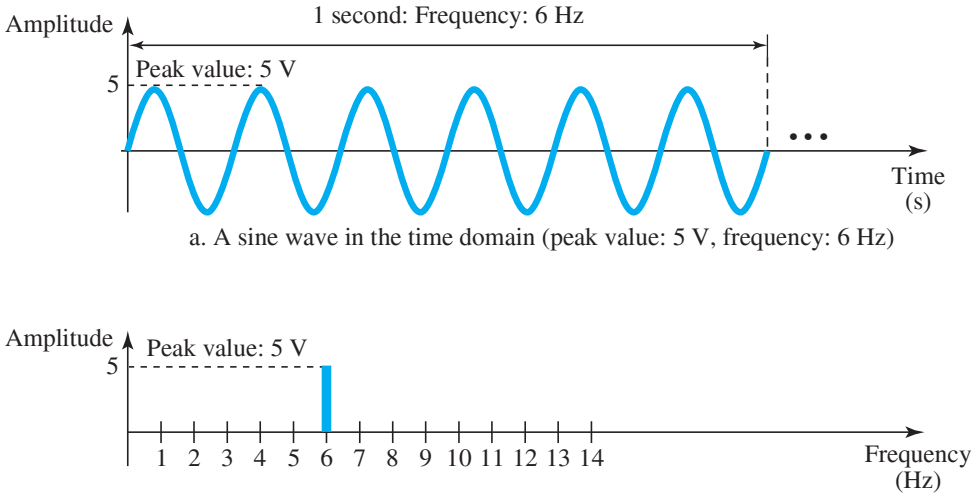
In a coaxial or fiber-optic cable, however, the wavelength is shorter (0.5 μm) because the propagation speed in the cable is decreased.

3.2.4 Time and Frequency Domains

A sine wave is comprehensively defined by its amplitude, frequency, and phase. We have been showing a sine wave by using what is called a **time-domain** plot. The time-domain plot shows changes in signal amplitude with respect to time (it is an amplitude-versus-time plot). Phase is not explicitly shown on a time-domain plot.

To show the relationship between amplitude and frequency, we can use what is called a **frequency-domain** plot. A frequency-domain plot is concerned with only the peak value and the frequency. Changes of amplitude during one period are not shown. Figure 3.8 shows a signal in both the time and frequency domains.

Figure 3.8 *The time-domain and frequency-domain plots of a sine wave*



a. A sine wave in the time domain (peak value: 5 V, frequency: 6 Hz)

b. The same sine wave in the frequency domain (peak value: 5 V, frequency: 6 Hz)

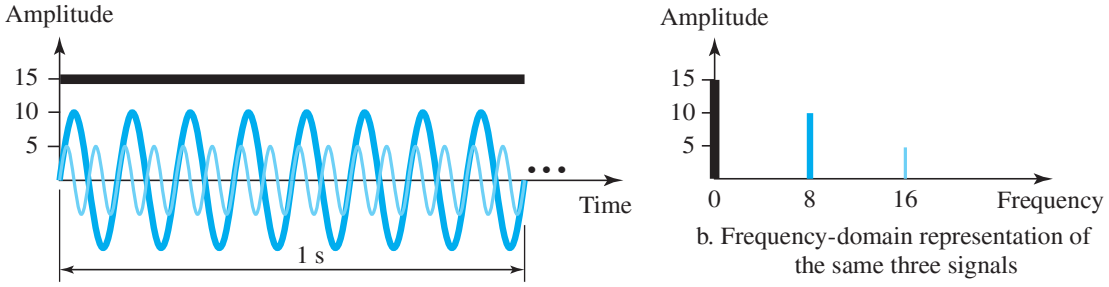
It is obvious that the frequency domain is easy to plot and conveys the information that one can find in a time domain plot. The advantage of the frequency domain is that we can immediately see the values of the frequency and peak amplitude. A complete sine wave is represented by one spike. The position of the spike shows the frequency; its height shows the peak amplitude.

A complete sine wave in the time domain can be represented by one single spike in the frequency domain.

Example 3.7

The frequency domain is more compact and useful when we are dealing with more than one sine wave. For example, Figure 3.9 shows three sine waves, each with different amplitude and frequency. All can be represented by three spikes in the frequency domain.

Figure 3.9 *The time domain and frequency domain of three sine waves*



a. Time-domain representation of three sine waves with frequencies 0, 8, and 16

b. Frequency-domain representation of the same three signals

3.2.5 Composite Signals

So far, we have focused on simple sine waves. Simple sine waves have many applications in daily life. We can send a single sine wave to carry electric energy from one place to another. For example, the power company sends a single sine wave with a frequency of 60 Hz to distribute electric energy to houses and businesses. As another example, we can use a single sine wave to send an alarm to a security center when a burglar opens a door or window in the house. In the first case, the sine wave is carrying energy; in the second, the sine wave is a signal of danger.

If we had only one single sine wave to convey a conversation over the phone, it would make no sense and carry no information. We would just hear a buzz. As we will see in Chapters 4 and 5, we need to send a composite signal to communicate data. A **composite signal** is made of many simple sine waves.

A single-frequency sine wave is not useful in data communications; we need to send a composite signal, a signal made of many simple sine waves.

In the early 1900s, the French mathematician Jean-Baptiste Fourier showed that any composite signal is actually a combination of simple sine waves with different frequencies, amplitudes, and phases. **Fourier analysis** is discussed in Appendix E; for our purposes, we just present the concept.

According to Fourier analysis, any composite signal is a combination of simple sine waves with different frequencies, amplitudes, and phases. Fourier analysis is discussed in Appendix E.

A composite signal can be periodic or nonperiodic. A periodic composite signal can be decomposed into a series of simple sine waves with discrete frequencies—frequencies that have integer values (1, 2, 3, and so on). A nonperiodic composite signal can be decomposed into a combination of an infinite number of simple sine waves with continuous frequencies, frequencies that have real values.

If the composite signal is periodic, the decomposition gives a series of signals with discrete frequencies; if the composite signal is nonperiodic, the decomposition gives a combination of sine waves with continuous frequencies.

Example 3.8

Figure 3.10 shows a periodic composite signal with frequency f. This type of signal is not typical of those found in data communications.We can consider it to be three alarm systems, each with a different frequency. The analysis of this signal can give us a good understanding of how to decompose signals.

It is very difficult to manually decompose this signal into a series of simple sine waves. However, there are tools, both hardware and software, that can help us do the job. We are not concerned about how it is done; we are only interested in the result. Figure 3.11 shows the result of decomposing the above signal in both the time and frequency domains.

The amplitude of the sine wave with frequency f is almost the same as the peak amplitude of the composite signal. The amplitude of the sine wave with frequency $3f$ is one-third of that of

Figure 3.10 *A composite periodic signal*

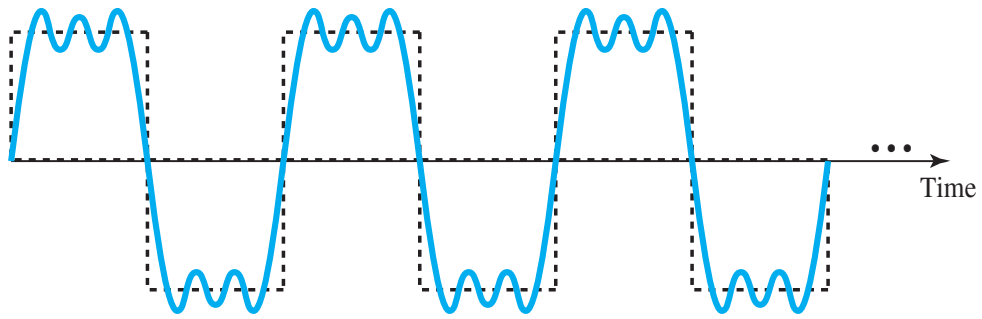
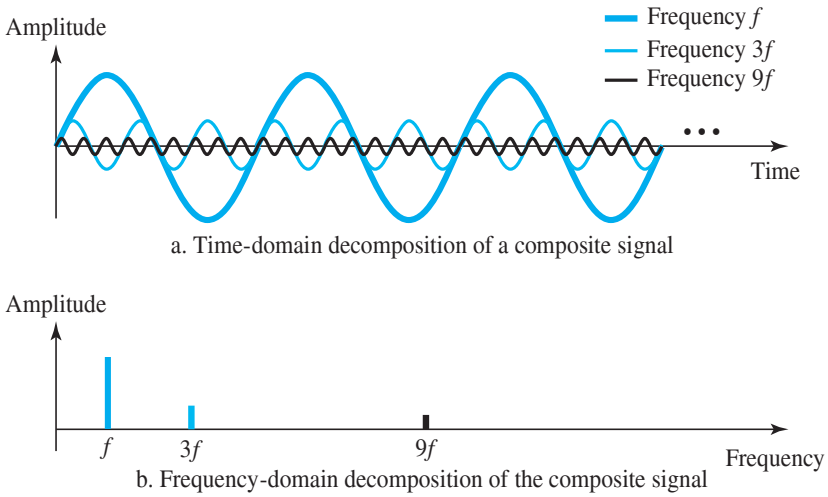


Figure 3.11 *Decomposition of a composite periodic signal in the time and frequency domains*



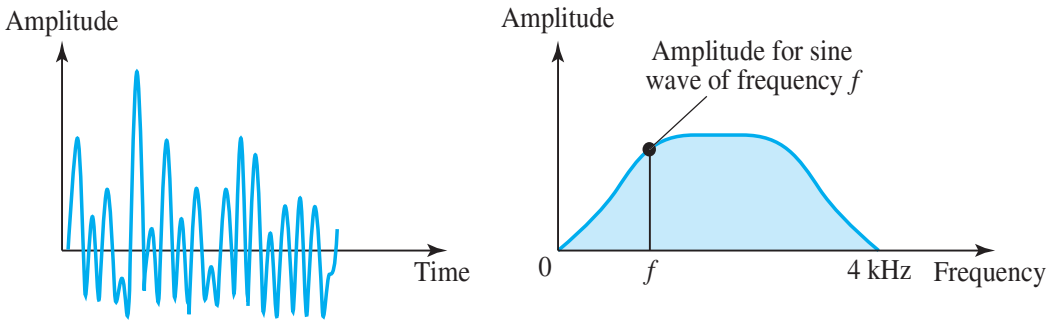
the first, and the amplitude of the sine wave with frequency $9f$ is one-ninth of the first. The frequency of the sine wave with frequency f is the same as the frequency of the composite signal; it is called the **fundamental frequency,** or first **harmonic.** The sine wave with frequency $3f$ has a frequency of 3 times the fundamental frequency; it is called the third harmonic. The third sine wave with frequency $9f$ has a frequency of 9 times the fundamental frequency; it is called the ninth harmonic.

Note that the frequency decomposition of the signal is discrete; it has frequencies f, $3f$, and $9f$. Because f is an integral number, $3f$ and $9f$ are also integral numbers. There are no frequencies such as $1.2f$ or $2.6f$. The frequency domain of a periodic composite signal is always made of discrete spikes.

Example 3.9

Figure 3.12 shows a nonperiodic composite signal. It can be the signal created by a microphone or a telephone set when a word or two is pronounced. In this case, the composite signal cannot be periodic, because that implies that we are repeating the same word or words with exactly the same tone.

Figure 3.12 *The time and frequency domains of a nonperiodic signal*



In a time-domain representation of this composite signal, there are an infinite number of simple sine frequencies. Although the number of frequencies in a human voice is infinite, the range is limited. A normal human being can create a continuous range of frequencies between 0 and 4 kHz.

Note that the frequency decomposition of the signal yields a continuous curve. There are an infinite number of frequencies between 0.0 and 4000.0 (real values). To find the amplitude related to frequency f, we draw a vertical line at f to intersect the envelope curve. The height of the vertical line is the amplitude of the corresponding frequency.

3.2.6 Bandwidth

The range of frequencies contained in a composite signal is its **bandwidth.** The bandwidth is normally a difference between two numbers. For example, if a composite signal contains frequencies between 1000 and 5000, its bandwidth is 5000 – 1000, or 4000.

The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.

Figure 3.13 shows the concept of bandwidth. The figure depicts two composite signals, one periodic and the other nonperiodic. The bandwidth of the periodic signal contains all integer frequencies between 1000 and 5000 (1000, 1001, 1002, . . .). The bandwidth of the nonperiodic signals has the same range, but the frequencies are continuous.

Example 3.10

If a periodic signal is decomposed into five sine waves with frequencies of 100, 300, 500, 700, and 900 Hz, what is its bandwidth? Draw the spectrum, assuming all components have a maximum amplitude of 10 V.

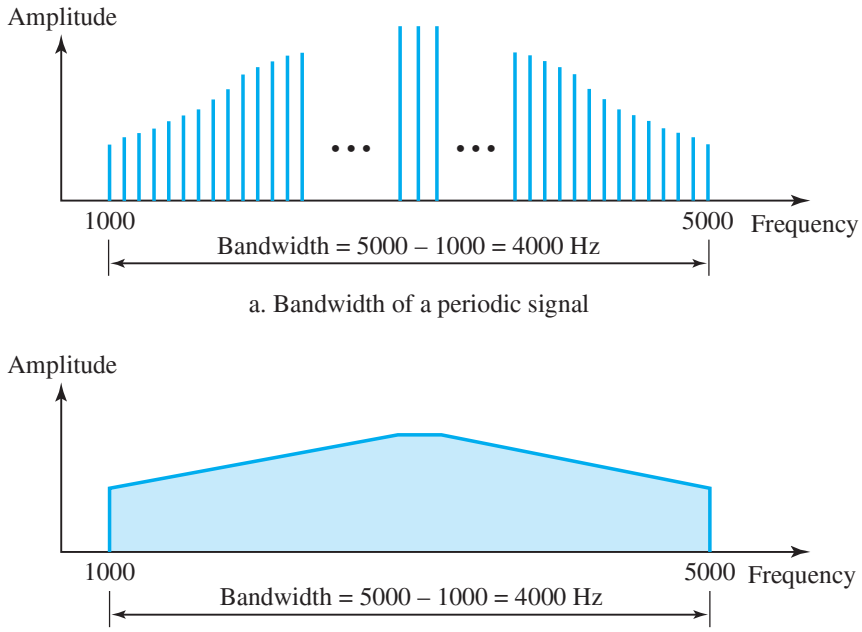
Solution

Let f_h be the highest frequency, f_l the lowest frequency, and B the bandwidth. Then

$$B = f_h - f_l = 900 - 100 = 800 \text{ Hz}$$

The spectrum has only five spikes, at 100, 300, 500, 700, and 900 Hz (see Figure 3.14).

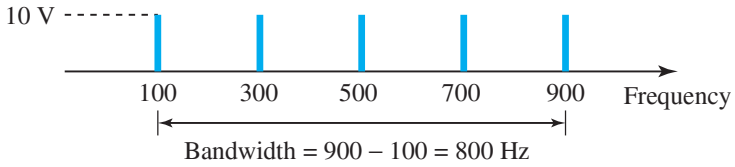
Figure 3.13 *The bandwidth of periodic and nonperiodic composite signals*



a. Bandwidth of a periodic signal

b. Bandwidth of a nonperiodic signal

Figure 3.14 *The bandwidth for Example 3.10*



Example 3.11

A periodic signal has a bandwidth of 20 Hz. The highest frequency is 60 Hz. What is the lowest frequency? Draw the spectrum if the signal contains all frequencies of the same amplitude.

Solution

Let f_h be the highest frequency, f_l the lowest frequency, and B the bandwidth. Then

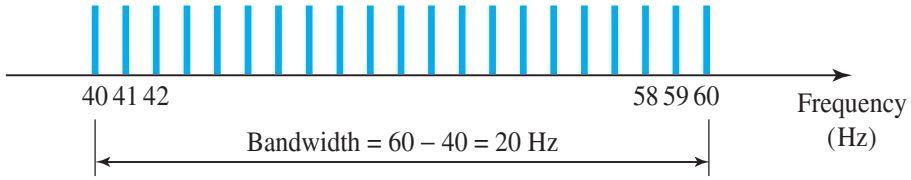
$$B = f_h - f_l \longrightarrow 20 = 60 - f_l \longrightarrow f_l = 60 - 20 = 40 \text{ Hz}$$

The spectrum contains all integer frequencies. We show this by a series of spikes (see Figure 3.15).

Example 3.12

A nonperiodic composite signal has a bandwidth of 200 kHz, with a middle frequency of 140 kHz and peak amplitude of 20 V. The two extreme frequencies have an amplitude of 0. Draw the frequency domain of the signal.

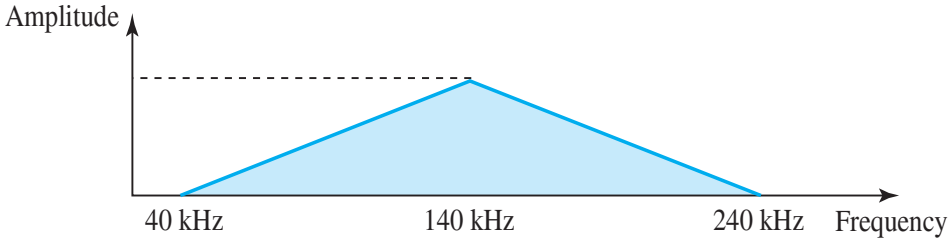
Figure 3.15 *The bandwidth for Example 3.11*



Solution

The lowest frequency must be at 40 kHz and the highest at 240 kHz. Figure 3.16 shows the frequency domain and the bandwidth.

Figure 3.16 *The bandwidth for Example 3.12*



Example 3.13

An example of a nonperiodic composite signal is the signal propagated by an AM radio station. In the United States, each AM radio station is assigned a 10-kHz bandwidth. The total bandwidth dedicated to AM radio ranges from 530 to 1700 kHz. We will show the rationale behind this 10-kHz bandwidth in Chapter 5.

Example 3.14

Another example of a nonperiodic composite signal is the signal propagated by an FM radio station. In the United States, each FM radio station is assigned a 200-kHz bandwidth. The total bandwidth dedicated to FM radio ranges from 88 to 108 MHz. We will show the rationale behind this 200-kHz bandwidth in Chapter 5.

Example 3.15

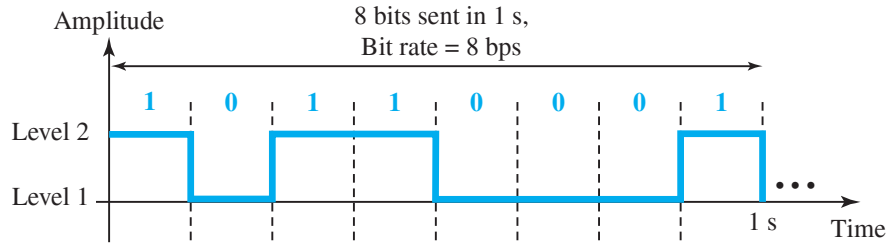
Another example of a nonperiodic composite signal is the signal received by an old-fashioned analog black-and-white TV. A TV screen is made up of pixels (picture elements) with each pixel being either white or black. The screen is scanned 30 times per second. (Scanning is actually 60 times per second, but odd lines are scanned in one round and even lines in the next and then interleaved.) If we assume a resolution of 525 × 700 (525 vertical lines and 700 horizontal lines), which is a ratio of 3:4, we have 367,500 pixels per screen. If we scan the screen 30 times per second, this is 367,500 × 30 = 11,025,000 pixels per second. The worst-case scenario is alternating black and white pixels. In this case, we need to represent one color by the minimum amplitude and the other color by the maximum amplitude. We can send 2 pixels per cycle. Therefore, we need 11,025,000 / 2 = 5,512,500 cycles per second, or Hz. The bandwidth needed is 5.5124 MHz.

This worst-case scenario has such a low probability of occurrence that the assumption is that we need only 70 percent of this bandwidth, which is 3.85 MHz. Since audio and synchronization signals are also needed, a 4-MHz bandwidth has been set aside for each black and white TV channel. An analog color TV channel has a 6-MHz bandwidth.

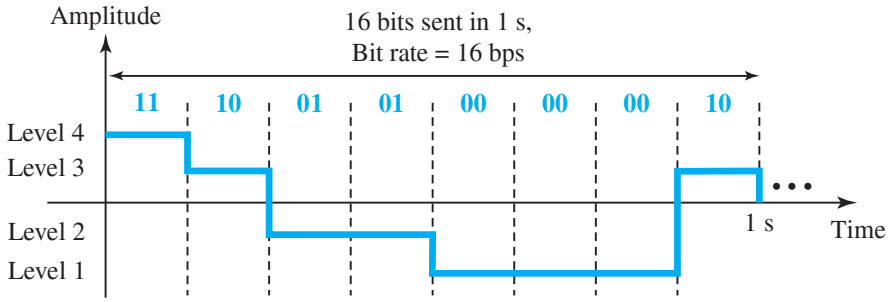
3.3 DIGITAL SIGNALS

In addition to being represented by an analog signal, information can also be represented by a digital signal. For example, a 1 can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels. In this case, we can send more than 1 bit for each level. Figure 3.17 shows two signals, one with two levels and the other with four. We send 1 bit per level in part a of the figure and 2 bits per level in part b of the figure. In general, if a signal has L levels, each level needs $\log_2 L$ bits. For this reason, we can send $\log_2 4 = 2$ bits in part b.

Figure 3.17 *Two digital signals: one with two signal levels and the other with four signal levels*



a. A digital signal with two levels



b. A digital signal with four levels

Example 3.16

A digital signal has eight levels. How many bits are needed per level? We calculate the number of bits from the following formula. Each signal level is represented by 3 bits.

$$\text{Number of bits per level} = \log_2 8 = 3$$

Example 3.17

A digital signal has nine levels. How many bits are needed per level? We calculate the number of bits by using the formula. Each signal level is represented by 3.17 bits. However, this answer is

not realistic. The number of bits sent per level needs to be an integer as well as a power of 2. For this example, 4 bits can represent one level.

3.3.1 Bit Rate

Most digital signals are nonperiodic, and thus period and frequency are not appropriate characteristics. Another term—*bit rate* (instead of *frequency*)—is used to describe digital signals. The **bit rate** is the number of bits sent in 1s, expressed in **bits per second (bps).** Figure 3.17 shows the bit rate for two signals.

Example 3.18

Assume we need to download text documents at the rate of 100 pages per second. What is the required bit rate of the channel?

Solution

A page is an average of 24 lines with 80 characters in each line. If we assume that one character requires 8 bits, the bit rate is

$$100 \times 24 \times 80 \times 8 = 1{,}536{,}000 \text{ bps} = 1.536 \text{ Mbps}$$

Example 3.19

A digitized voice channel, as we will see in Chapter 4, is made by digitizing a 4-kHz bandwidth analog voice signal. We need to sample the signal at twice the highest frequency (two samples per hertz). We assume that each sample requires 8 bits. What is the required bit rate?

Solution

The bit rate can be calculated as

$$2 \times 4000 \times 8 = 64{,}000 \text{ bps} = 64 \text{ kbps}$$

Example 3.20

What is the bit rate for high-definition TV (HDTV)?

Solution

HDTV uses digital signals to broadcast high quality video signals. The HDTV screen is normally a ratio of 16 : 9 (in contrast to 4 : 3 for regular TV), which means the screen is wider. There are 1920 by 1080 pixels per screen, and the screen is renewed 30 times per second. Twenty-four bits represents one color pixel. We can calculate the bit rate as

$$1920 \times 1080 \times 30 \times 24 = 1{,}492{,}992{,}000 \approx 1.5 \text{ Gbps}$$

The TV stations reduce this rate to 20 to 40 Mbps through compression.

3.3.2 Bit Length

We discussed the concept of the wavelength for an analog signal: the distance one cycle occupies on the transmission medium. We can define something similar for a digital signal: the bit length. The **bit length** is the distance one bit occupies on the transmission medium.

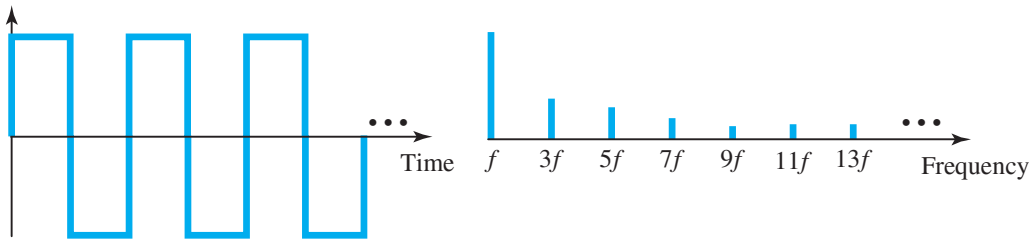
$$\text{Bit length} = \text{propagation speed} \times \text{bit duration}$$

3.3.3 Digital Signal as a Composite Analog Signal

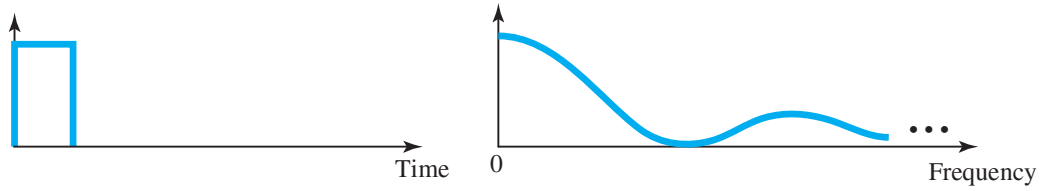
Based on Fourier analysis (See Appendix E), a digital signal is a composite analog signal. The bandwidth is infinite, as you may have guessed. We can intuitively come up with this concept when we consider a digital signal. A digital signal, in the time domain, comprises connected vertical and horizontal line segments. A vertical line in the time domain means a frequency of infinity (sudden change in time); a horizontal line in the time domain means a frequency of zero (no change in time). Going from a frequency of zero to a frequency of infinity (and vice versa) implies all frequencies in between are part of the domain.

Fourier analysis can be used to decompose a digital signal. If the digital signal is periodic, which is rare in data communications, the decomposed signal has a frequency-domain representation with an infinite bandwidth and discrete frequencies. If the digital signal is nonperiodic, the decomposed signal still has an infinite bandwidth, but the frequencies are continuous. Figure 3.18 shows a periodic and a nonperiodic digital signal and their bandwidths.

Figure 3.18 *The time and frequency domains of periodic and nonperiodic digital signals*



a. Time and frequency domains of periodic digital signal



b. Time and frequency domains of nonperiodic digital signal

Note that both bandwidths are infinite, but the periodic signal has discrete frequencies while the nonperiodic signal has continuous frequencies.

3.3.4 Transmission of Digital Signals

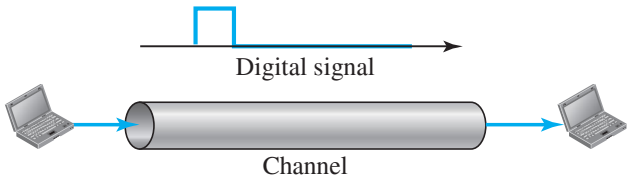
The previous discussion asserts that a digital signal, periodic or nonperiodic, is a composite analog signal with frequencies between zero and infinity. For the remainder of the discussion, let us consider the case of a nonperiodic digital signal, similar to the ones we encounter in data communications. The fundamental question is, How can we send a digital signal from point *A* to point *B*? We can transmit a digital signal by using one of two different approaches: baseband transmission or broadband transmission (using modulation).

A digital signal is a composite analog signal with an infinite bandwidth.

Baseband Transmission

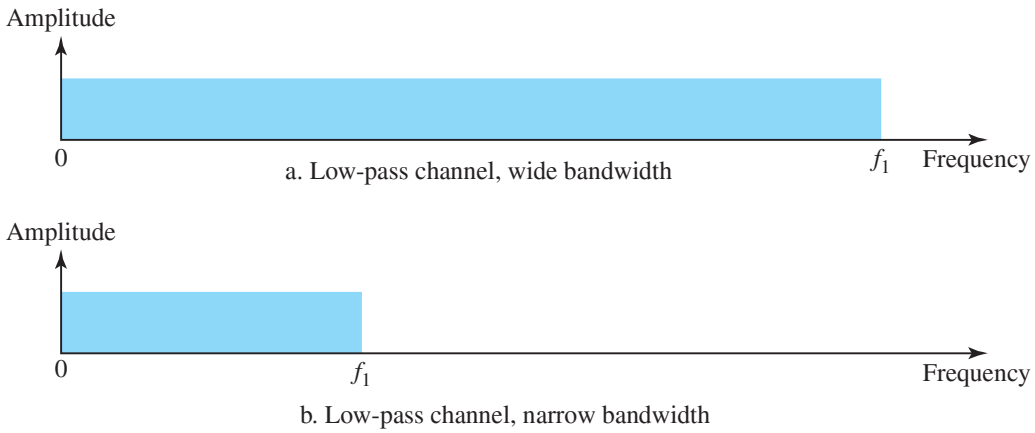
Baseband transmission means sending a digital signal over a channel without changing the digital signal to an analog signal. Figure 3.19 shows **baseband** transmission.

Figure 3.19 *Baseband transmission*



Baseband transmission requires that we have a **low-pass channel,** a channel with a bandwidth that starts from zero. This is the case if we have a dedicated medium with a bandwidth constituting only one channel. For example, the entire bandwidth of a cable connecting two computers is one single channel. As another example, we may connect several computers to a bus, but not allow more than two stations to communicate at a time. Again we have a low-pass channel, and we can use it for baseband communication. Figure 3.20 shows two low-pass channels: one with a narrow bandwidth and the other with a wide bandwidth. We need to remember that a low-pass channel with infinite bandwidth is ideal, but we cannot have such a channel in real life. However, we can get close.

Figure 3.20 *Bandwidths of two low-pass channels*

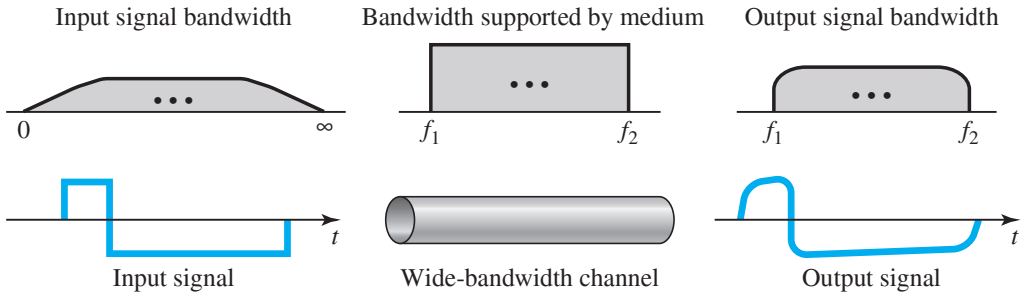


Let us study two cases of a baseband communication: a low-pass channel with a wide bandwidth and one with a limited bandwidth.

Case 1: Low-Pass Channel with Wide Bandwidth

If we want to preserve the exact form of a nonperiodic digital signal with vertical segments vertical and horizontal segments horizontal, we need to send the entire spectrum, the continuous range of frequencies between zero and infinity. This is possible if we have a dedicated medium with an infinite bandwidth between the sender and receiver that preserves the exact amplitude of each component of the composite signal. Although this may be possible inside a computer (e.g., between CPU and memory), it is not possible between two devices. Fortunately, the amplitudes of the frequencies at the border of the bandwidth are so small that they can be ignored. This means that if we have a medium, such as a coaxial or fiber optic cable, with a very wide bandwidth, two stations can communicate by using digital signals with very good accuracy, as shown in Figure 3.21. Note that f_1 is close to zero, and f_2 is very high.

Figure 3.21 *Baseband transmission using a dedicated medium*



Although the output signal is not an exact replica of the original signal, the data can still be deduced from the received signal. Note that although some of the frequencies are blocked by the medium, they are not critical.

Baseband transmission of a digital signal that preserves the shape of the digital signal is possible only if we have a low-pass channel with an infinite or very wide bandwidth.

Example 3.21

An example of a dedicated channel where the entire bandwidth of the medium is used as one single channel is a LAN. Almost every wired LAN today uses a dedicated channel for two stations communicating with each other. In a bus topology LAN with multipoint connections, only two stations can communicate with each other at each moment in time (timesharing); the other stations need to refrain from sending data. In a star topology LAN, the entire channel between each station and the hub is used for communication between these two entities. We study LANs in Chapter 13.

Case 2: Low-Pass Channel with Limited Bandwidth

In a low-pass channel with limited bandwidth, we approximate the digital signal with an analog signal. The level of approximation depends on the bandwidth available.

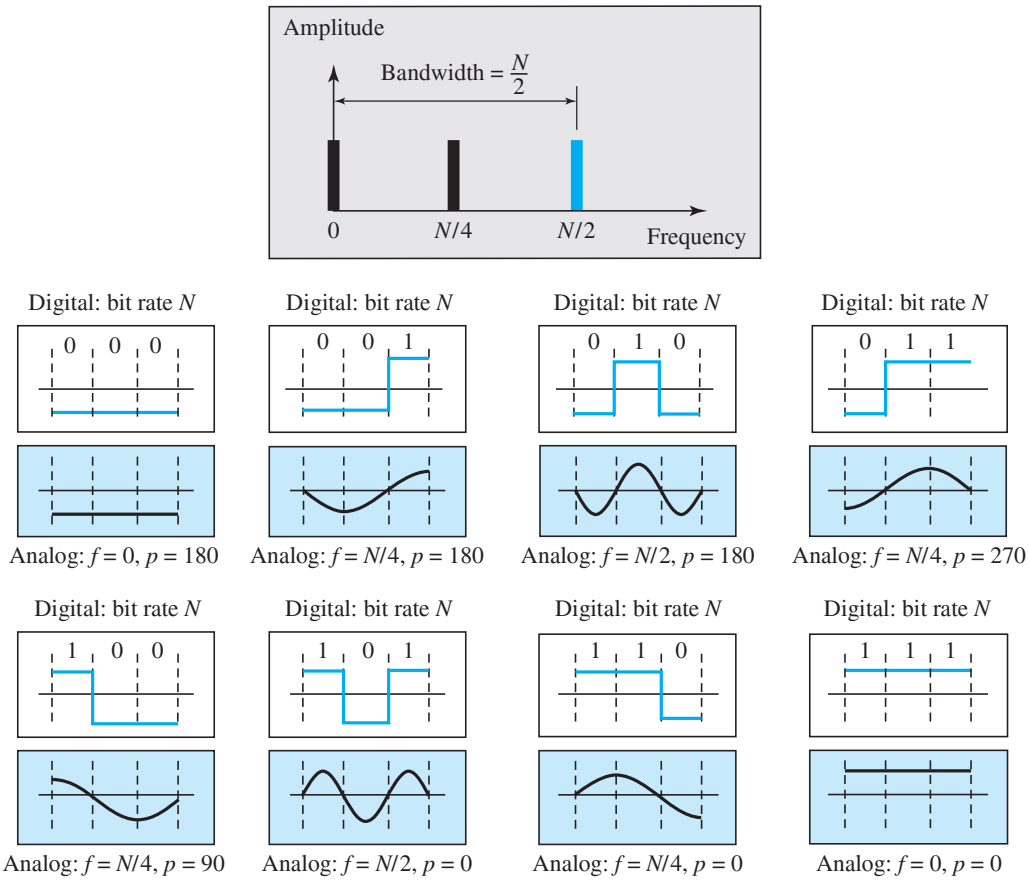
Rough Approximation

Let us assume that we have a digital signal of bit rate N. If we want to send analog signals to roughly simulate this signal, we need to consider the worst case, a maximum number of changes in the digital signal. This happens when the signal carries the

sequence 01010101 . . . or the sequence 10101010. . . . To simulate these two cases, we need an analog signal of frequency $f = N/2$. Let 1 be the positive peak value and 0 be the negative peak value. We send 2 bits in each cycle; the frequency of the analog signal is one-half of the bit rate, or $N/2$. However, just this one frequency cannot make all patterns; we need more components. The maximum frequency is $N/2$. As an example of this concept, let us see how a digital signal with a 3-bit pattern can be simulated by using analog signals. Figure 3.22 shows the idea. The two similar cases (000 and 111) are simulated with a signal with frequency $f = 0$ and a phase of 180° for 000 and a phase of 0° for 111. The two worst cases (010 and 101) are simulated with an analog signal with frequency $f = N/2$ and phases of 180° and 0°. The other four cases can only be simulated with an analog signal with $f = N/4$ and phases of 180°, 270°, 90°, and 0°. In other words, we need a channel that can handle frequencies 0, $N/4$, and $N/2$. This rough approximation is referred to as using the first harmonic ($N/2$) frequency. The required bandwidth is

$$\text{Bandwidth} = \frac{N}{2} - 0 = \frac{N}{2}$$

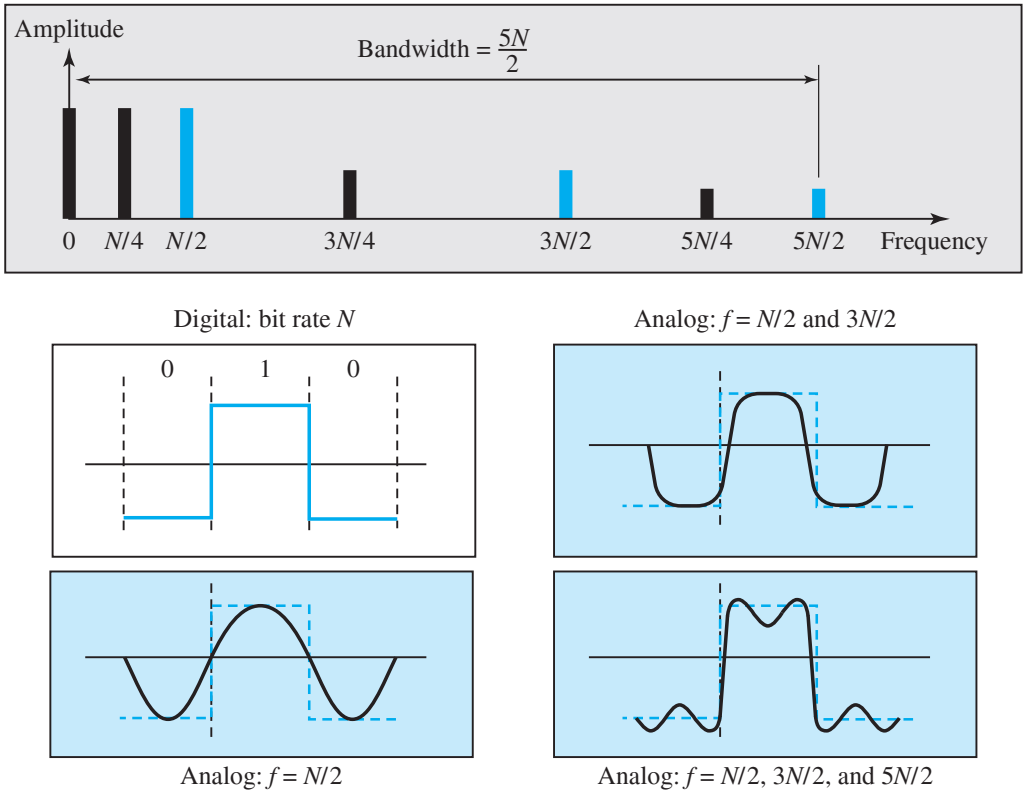
Figure 3.22 *Rough approximation of a digital signal using the first harmonic for worst case*



Better Approximation

To make the shape of the analog signal look more like that of a digital signal, we need to add more harmonics of the frequencies. We need to increase the bandwidth. We can increase the bandwidth to $3N/2$, $5N/2$, $7N/2$, and so on. Figure 3.23 shows the effect of

Figure 3.23 *Simulating a digital signal with first three harmonics*



this increase for one of the worst cases, the pattern 010. Note that we have shown only the highest frequency for each harmonic. We use the first, third, and fifth harmonics. The required bandwidth is now $5N/2$, the difference between the lowest frequency 0 and the highest frequency $5N/2$. As we emphasized before, we need to remember that the required bandwidth is proportional to the bit rate.

In baseband transmission, the required bandwidth is proportional to the bit rate; if we need to send bits faster, we need more bandwidth.

By using this method, Table 3.2 shows how much bandwidth we need to send data at different rates.

Table 3.2 *Bandwidth requirements*

Bit Rate	*Harmonic 1*	*Harmonics 1, 3*	*Harmonics 1, 3, 5*
n = 1 kbps	B = 500 Hz	B = 1.5 kHz	B = 2.5 kHz
n = 10 kbps	B = 5 kHz	B = 15 kHz	B = 25 kHz
n = 100 kbps	B = 50 kHz	B = 150 kHz	B = 250 kHz

Example 3.22

What is the required bandwidth of a low-pass channel if we need to send 1 Mbps by using baseband transmission?

Solution

The answer depends on the accuracy desired.

- **a.** The minimum bandwidth, a rough approximation, is B = bit rate /2, or 500 kHz. We need a low-pass channel with frequencies between 0 and 500 kHz.
- **b.** A better result can be achieved by using the first and the third harmonics with the required bandwidth $B = 3 \times 500$ kHz = 1.5 MHz.
- **c.** A still better result can be achieved by using the first, third, and fifth harmonics with $B = 5 \times 500$ kHz = 2.5 MHz.

Example 3.23

We have a low-pass channel with bandwidth 100 kHz. What is the maximum bit rate of this channel?

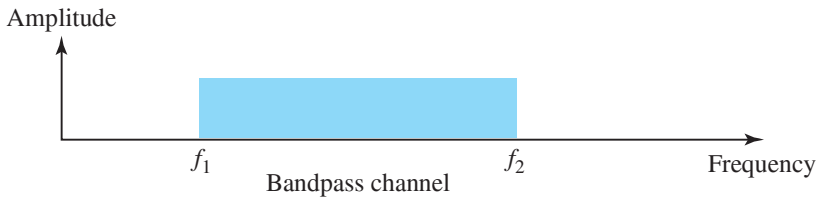
Solution

The maximum bit rate can be achieved if we use the first harmonic. The bit rate is 2 times the available bandwidth, or 200 kbps.

Broadband Transmission (Using Modulation)

Broadband transmission or modulation means changing the digital signal to an analog signal for transmission. Modulation allows us to use a **bandpass channel**—a channel with a bandwidth that does not start from zero. This type of channel is more available than a low-pass channel. Figure 3.24 shows a bandpass channel.

Figure 3.24 *Bandwidth of a bandpass channel*

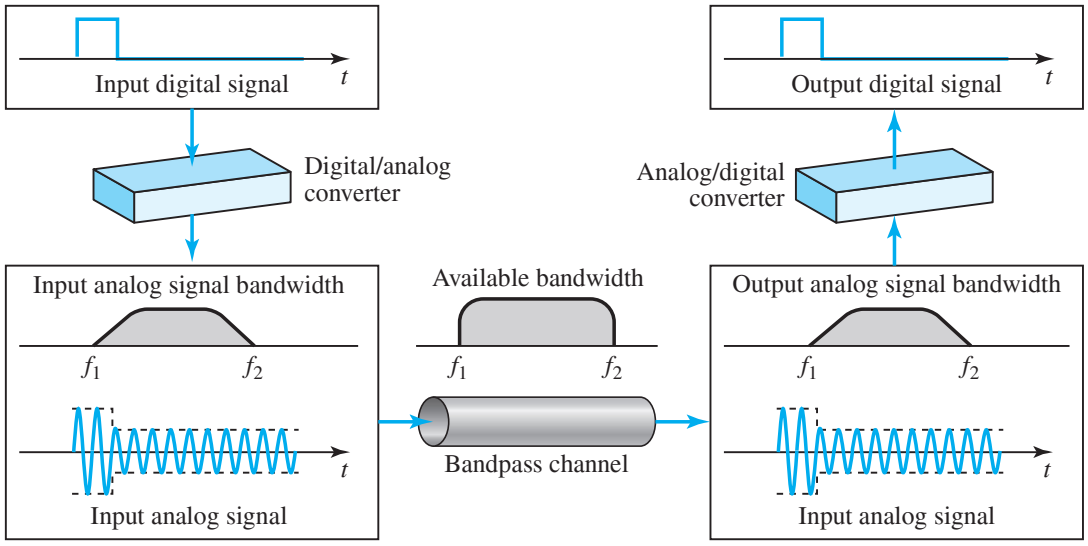


Note that a low-pass channel can be considered a bandpass channel with the lower frequency starting at zero.

Figure 3.25 shows the modulation of a digital signal. In the figure, a digital signal is converted to a composite analog signal. We have used a single-frequency analog signal (called a carrier); the amplitude of the carrier has been changed to look like the digital signal. The result, however, is not a single-frequency signal; it is a composite signal, as we will see in Chapter 5. At the receiver, the received analog signal is converted to digital, and the result is a replica of what has been sent.

If the available channel is a bandpass channel, we cannot send the digital signal directly to the channel; we need to convert the digital signal to an analog signal before transmission.

Figure 3.25 *Modulation of a digital signal for transmission on a bandpass channel*



Example 3.24

An example of broadband transmission using modulation is the sending of computer data through a telephone subscriber line, the line connecting a resident to the central telephone office. These lines, installed many years ago, are designed to carry voice (analog signal) with a limited bandwidth (frequencies between 0 and 4 kHz). Although this channel can be used as a low-pass channel, it is normally considered a bandpass channel. One reason is that the bandwidth is so narrow (4 kHz) that if we treat the channel as low-pass and use it for baseband transmission, the maximum bit rate can be only 8 kbps. The solution is to consider the channel a bandpass channel, convert the digital signal from the computer to an analog signal, and send the analog signal. We can install two converters to change the digital signal to analog and vice versa at the receiving end. The converter, in this case, is called a *modem* (*mo*dulator/*dem*odulator), which we discuss in detail in Chapter 5.

Example 3.25

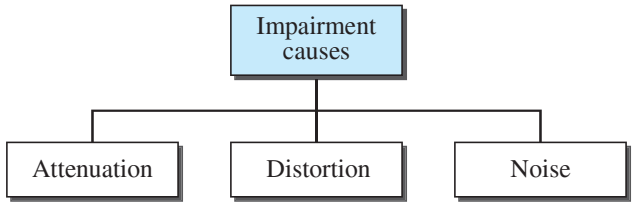
A second example is the digital cellular telephone. For better reception, digital cellular phones convert the analog voice signal to a digital signal (see Chapter 16). Although the bandwidth allocated to a company providing digital cellular phone service is very wide, we still cannot send the digital signal without conversion. The reason is that we only have a bandpass channel available between caller and callee. For example, if the available bandwidth is W and we allow 1000 couples to talk simultaneously, this means the available channel is W/1000, just part of the entire bandwidth. We need to convert the digitized voice to a composite analog signal before sending. The digital cellular phones convert the analog audio signal to digital and then convert it again to analog for transmission over a bandpass channel.

3.4 TRANSMISSION IMPAIRMENT

Signals travel through transmission media, which are not perfect. The imperfection causes signal impairment. This means that the signal at the beginning of the medium is not the

same as the signal at the end of the medium. What is sent is not what is received. Three causes of impairment are attenuation, distortion, and noise (see Figure 3.26).

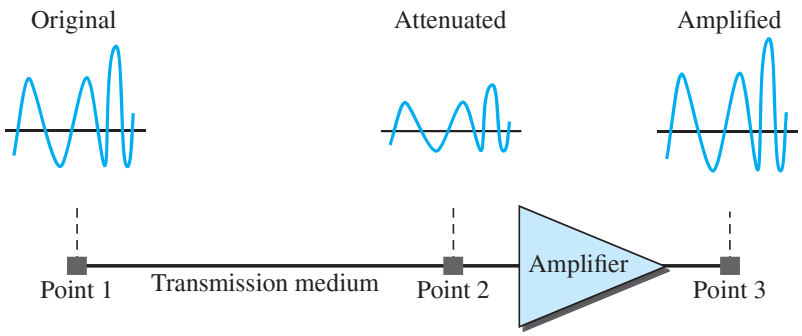
Figure 3.26 *Causes of impairment*



3.4.1 Attenuation

Attenuation means a loss of energy. When a signal, simple or composite, travels through a medium, it loses some of its energy in overcoming the resistance of the medium. That is why a wire carrying electric signals gets warm, if not hot, after a while. Some of the electrical energy in the signal is converted to heat. To compensate for this loss, amplifiers are used to amplify the signal. Figure 3.27 shows the effect of attenuation and amplification.

Figure 3.27 *Attenuation*



Decibel

To show that a signal has lost or gained strength, engineers use the unit of the decibel. The **decibel (dB)** measures the relative strengths of two signals or one signal at two different points. Note that the decibel is negative if a signal is attenuated and positive if a signal is amplified.

$$\mathbf{dB = 10 \log_{10} \frac{P_2}{P_1}}$$

Variables P_1 and P_2 are the powers of a signal at points 1 and 2, respectively. Note that some engineering books define the decibel in terms of voltage instead of power. In this case, because power is proportional to the square of the voltage, the formula is $dB = 20 \log_{10} (V_2/V_1)$. In this text, we express dB in terms of power.

Example 3.26

Suppose a signal travels through a transmission medium and its power is reduced to one-half. This means that $P_2 = \frac{1}{2} P_1$. In this case, the attenuation (loss of power) can be calculated as

$$10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{0.5 P_1}{P_1} = 10 \log_{10} 0.5 = 10(-0.3) = -3 \text{ dB}$$

A loss of 3 dB (−3 dB) is equivalent to losing one-half the power.

Example 3.27

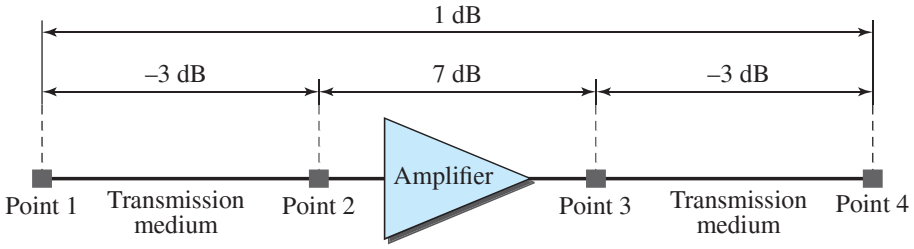
A signal travels through an amplifier, and its power is increased 10 times. This means that $P_2 = 10P_1$. In this case, the amplification (gain of power) can be calculated as

$$10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{10 P_1}{P_1} = 10 \log_{10} 10 = 10(1) = 10 \text{ dB}$$

Example 3.28

One reason that engineers use the decibel to measure the changes in the strength of a signal is that decibel numbers can be added (or subtracted) when we are measuring several points (cascading) instead of just two. In Figure 3.28 a signal travels from point 1 to point 4. The signal is attenuated by the time it reaches point 2. Between points 2 and 3, the signal is amplified. Again, between points 3 and 4, the signal is attenuated. We can find the resultant decibel value for the signal just by adding the decibel measurements between each set of points.

Figure 3.28 *Decibels for Example 3.28*



In this case, the decibel value can be calculated as

$$\text{dB} = -3 + 7 - 3 = +1$$

The signal has gained in power.

Example 3.29

Sometimes the decibel is used to measure signal power in milliwatts. In this case, it is referred to as dB_m and is calculated as $\text{dB}_\text{m} = 10 \log_{10} P_m$, where P_m is the power in milliwatts. Calculate the power of a signal if its $\text{dB}_\text{m} = -30$.

Solution

We can calculate the power in the signal as

$$\text{dB}_\text{m} = 10 \log_{10} \longrightarrow dB_m = -30 \longrightarrow \log_{10} P_m = -3 \longrightarrow P_m = 10^{-3} \text{ mW}$$

Example 3.30

The loss in a cable is usually defined in decibels per kilometer (dB/km). If the signal at the beginning of a cable with −0.3 dB/km has a power of 2 mW, what is the power of the signal at 5 km?

Solution

The loss in the cable in decibels is $5 \times (-0.3) = -1.5$ dB. We can calculate the power as

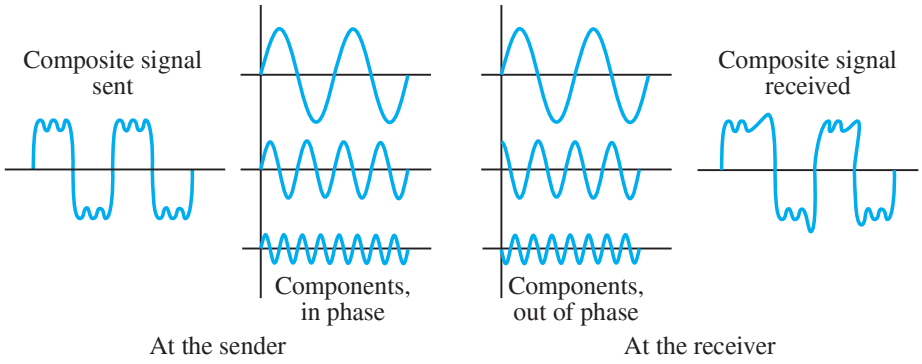
$$\text{dB} = 10 \log_{10} (P_2 / P_1) = -1.5 \quad \longrightarrow \quad (P_2 / P_1) = 10^{-0.15} = 0.71$$

$$P_2 = 0.71 P_1 = 0.7 \times 2 \text{ mW} = 1.4 \text{ mW}$$

3.4.2 Distortion

Distortion means that the signal changes its form or shape. Distortion can occur in a composite signal made of different frequencies. Each signal component has its own propagation speed (see the next section) through a medium and, therefore, its own delay in arriving at the final destination. Differences in delay may create a difference in phase if the delay is not exactly the same as the period duration. In other words, signal components at the receiver have phases different from what they had at the sender. The shape of the composite signal is therefore not the same. Figure 3.29 shows the effect of distortion on a composite signal.

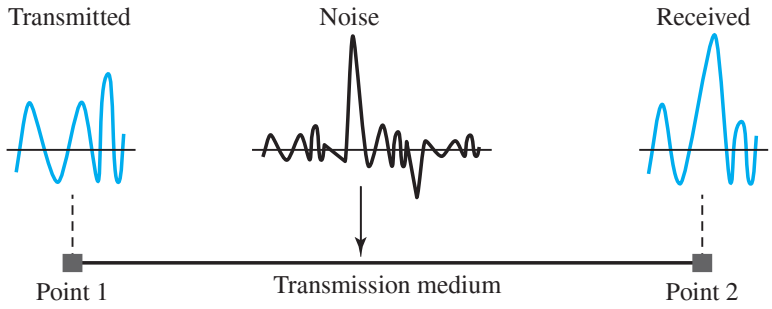
Figure 3.29 *Distortion*



3.4.3 Noise

Noise is another cause of impairment. Several types of noise, such as thermal noise, induced noise, crosstalk, and impulse noise, may corrupt the signal. Thermal noise is the random motion of electrons in a wire, which creates an extra signal not originally sent by the transmitter. Induced noise comes from sources such as motors and appliances. These devices act as a sending antenna, and the transmission medium acts as the receiving antenna. Crosstalk is the effect of one wire on the other. One wire acts as a sending antenna and the other as the receiving antenna. Impulse noise is a spike (a signal with high energy in a very short time) that comes from power lines, lightning, and so on. Figure 3.30 shows the effect of noise on a signal. We discuss error in Chapter 10.

Figure 3.30 *Noise*



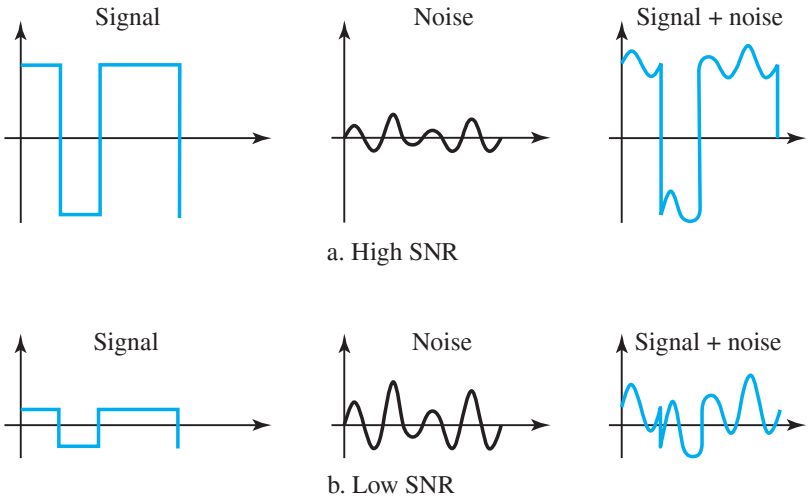
Signal-to-Noise Ratio (SNR)

As we will see later, to find the theoretical bit rate limit, we need to know the ratio of the signal power to the noise power. The **signal-to-noise ratio** is defined as

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

We need to consider the average signal power and the average noise power because these may change with time. Figure 3.31 shows the idea of SNR.

Figure 3.31 *Two cases of SNR: a high SNR and a low SNR*



SNR is actually the ratio of what is wanted (signal) to what is not wanted (noise). A high SNR means the signal is less corrupted by noise; a low SNR means the signal is more corrupted by noise.

Because SNR is the ratio of two powers, it is often described in decibel units, SNR_{dB}, defined as

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \text{SNR}$$

Example 3.31

The power of a signal is 10 mW and the power of the noise is 1 μW; what are the values of SNR and SNR_{dB}?

Solution

The values of SNR and SNR_{dB} can be calculated as follows:

$$\text{SNR} = (10{,}000\ \mu\text{w}) / (1\ \mu\text{w}) = 10{,}000 \qquad \text{SNR}_{\text{dB}} = 10 \log_{10} 10{,}000 = 10 \log_{10} 10^4 = 40$$

Example 3.32

The values of SNR and SNR_{dB} for a noiseless channel are

$$\text{SNR} = (\text{signal power}) / 0 = \infty \longrightarrow \text{SNR}_{\text{dB}} = 10 \log_{10} \infty = \infty$$

We can never achieve this ratio in real life; it is an ideal.

3.5 DATA RATE LIMITS

A very important consideration in data communications is how fast we can send data, in bits per second, over a channel. Data rate depends on three factors:

1. The bandwidth available
2. The level of the signals we use
3. The quality of the channel (the level of noise)

Two theoretical formulas were developed to calculate the data rate: one by Nyquist for a noiseless channel, another by Shannon for a noisy channel.

3.5.1 Noiseless Channel: Nyquist Bit Rate

For a noiseless channel, the **Nyquist bit rate** formula defines the theoretical maximum bit rate

$$\text{BitRate} = 2 \times \text{bandwidth} \times \log_2 L$$

In this formula, bandwidth is the bandwidth of the channel, L is the number of signal levels used to represent data, and BitRate is the bit rate in bits per second.

According to the formula, we might think that, given a specific bandwidth, we can have any bit rate we want by increasing the number of signal levels. Although the idea is theoretically correct, practically there is a limit. When we increase the number of signal levels, we impose a burden on the receiver. If the number of levels in a signal is just 2, the receiver can easily distinguish between a 0 and a 1. If the level of a signal is 64, the receiver must be very sophisticated to distinguish between 64 different levels. In other words, increasing the levels of a signal reduces the reliability of the system.

Increasing the levels of a signal may reduce the reliability of the system.

Example 3.33

Does the Nyquist theorem bit rate agree with the intuitive bit rate described in baseband transmission?

Solution

They match when we have only two levels. We said, in baseband transmission, the bit rate is 2 times the bandwidth if we use only the first harmonic in the worst case. However, the Nyquist formula is more general than what we derived intuitively; it can be applied to baseband transmission and modulation. Also, it can be applied when we have two or more levels of signals.

Example 3.34

Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. The maximum bit rate can be calculated as

$$\textbf{BitRate} = 2 \times 3000 \times \log_2 2 = 6000 \text{ bps}$$

Example 3.35

Consider the same noiseless channel transmitting a signal with four signal levels (for each level, we send 2 bits). The maximum bit rate can be calculated as

$$\textbf{BitRate} = 2 \times 3000 \times \log_2 4 = 12{,}000 \text{ bps}$$

Example 3.36

We need to send 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signal levels do we need?

Solution

We can use the Nyquist formula as shown:

$$265{,}000 = 2 \times 20{,}000 \times \log_2 L \longrightarrow \log_2 L = 6.625 \longrightarrow L = 2^{6.625} = 98.7 \text{ levels}$$

Since this result is not a power of 2, we need to either increase the number of levels or reduce the bit rate. If we have 128 levels, the bit rate is 280 kbps. If we have 64 levels, the bit rate is 240 kbps.

3.5.2 Noisy Channel: Shannon Capacity

In reality, we cannot have a noiseless channel; the channel is always noisy. In 1944, Claude Shannon introduced a formula, called the **Shannon capacity,** to determine the theoretical highest data rate for a noisy channel:

$$\textbf{Capacity} = \textbf{bandwidth} \times \log_2(1 + \text{SNR})$$

In this formula, bandwidth is the bandwidth of the channel, SNR is the signal-to-noise ratio, and capacity is the capacity of the channel in bits per second. Note that in the Shannon formula there is no indication of the signal level, which means that no matter how many levels we have, we cannot achieve a data rate higher than the capacity of the channel. In other words, the formula defines a characteristic of the channel, not the method of transmission.

Example 3.37

Consider an extremely noisy channel in which the value of the signal-to-noise ratio is almost zero. In other words, the noise is so strong that the signal is faint. For this channel the capacity C is calculated as

$$C = B \log_2 (1 + \text{SNR}) = B \log_2(1 + 0) = B \log_2 1 = B \times 0 = 0$$

This means that the capacity of this channel is zero regardless of the bandwidth. In other words, we cannot receive any data through this channel.

Example 3.38

We can calculate the theoretical highest bit rate of a regular telephone line. A telephone line normally has a bandwidth of 3000 Hz (300 to 3300 Hz) assigned for data communications. The signal-to-noise ratio is usually 3162. For this channel the capacity is calculated as

$$C = B \log_2 (1 + \text{SNR}) = 3000 \log_2(1 + 3162) = 3000 \times 11.62 = 34{,}860 \text{ bps}$$

This means that the highest bit rate for a telephone line is 34.860 kbps. If we want to send data faster than this, we can either increase the bandwidth of the line or improve the signal-to-noise ratio.

Example 3.39

The signal-to-noise ratio is often given in decibels. Assume that $\text{SNR}_{\text{dB}} = 36$ and the channel bandwidth is 2 MHz. The theoretical channel capacity can be calculated as

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \text{SNR} \longrightarrow \text{SNR} = 10^{\text{SNR}_{\text{dB}}/10} \longrightarrow \text{SNR} = 10^{3.6} = 3981$$

$$C = B \log_2(1 + \text{SNR}) = 2 \times 10^6 \times \log_2 3982 = 24 \text{ Mbps}$$

Example 3.40

When the SNR is very high, we can assume that SNR + 1 is almost the same as SNR. In these cases, the theoretical channel capacity can be simplified to $C = B \times \text{SNR}_{\text{dB}}$. For example, we can calculate the theoretical capacity of the previous example as

$$C = 2 \text{ MHz} \times (36 / 3) = 24 \text{ Mbps}$$

3.5.3 Using Both Limits

In practice, we need to use both methods to find the limits and signal levels. Let us show this with an example.

Example 3.41

We have a channel with a 1-MHz bandwidth. The SNR for this channel is 63. What are the appropriate bit rate and signal level?

Solution

First, we use the Shannon formula to find the upper limit.

$$C = B \log_2(1 + \text{SNR}) = 10^6 \log_2(1 + 63) = 10^6 \log_2 64 = 6 \text{ Mbps}$$

The Shannon formula gives us 6 Mbps, the upper limit. For better performance we choose something lower, 4 Mbps, for example. Then we use the Nyquist formula to find the number of signal levels.

$$4 \text{ Mbps} = 2 \times 1 \text{ MHz} \times \log_2 L \longrightarrow L = 4$$

**The Shannon capacity gives us the upper limit;
the Nyquist formula tells us how many signal levels we need.**

3.6 PERFORMANCE

Up to now, we have discussed the tools of transmitting data (signals) over a network and how the data behave. One important issue in networking is the performance of the network—how good is it? We discuss quality of service, an overall measurement of network performance, in greater detail in Chapter 30. In this section, we introduce terms that we need for future chapters.

3.6.1 Bandwidth

One characteristic that measures network performance is bandwidth. However, the term can be used in two different contexts with two different measuring values: bandwidth in hertz and bandwidth in bits per second.

Bandwidth in Hertz

We have discussed this concept. Bandwidth in hertz is the range of frequencies contained in a composite signal or the range of frequencies a channel can pass. For example, we can say the bandwidth of a subscriber telephone line is 4 kHz.

Bandwidth in Bits per Seconds

The term *bandwidth* can also refer to the number of bits per second that a channel, a link, or even a network can transmit. For example, one can say the bandwidth of a Fast Ethernet network (or the links in this network) is a maximum of 100 Mbps. This means that this network can send 100 Mbps.

Relationship

There is an explicit relationship between the bandwidth in hertz and bandwidth in bits per second. Basically, an increase in bandwidth in hertz means an increase in bandwidth in bits per second. The relationship depends on whether we have baseband transmission or transmission with modulation. We discuss this relationship in Chapters 4 and 5.

In networking, we use the term *bandwidth* in two contexts.

- **The first, *bandwidth in hertz,* refers to the range of frequencies in a composite signal or the range of frequencies that a channel can pass.**
- **The second, *bandwidth in bits per second,* refers to the speed of bit transmission in a channel or link.**

Example 3.42

The bandwidth of a subscriber line is 4 kHz for voice or data. The bandwidth of this line for data transmission can be up to 56,000 bps using a sophisticated modem to change the digital signal to analog.

Example 3.43

If the telephone company improves the quality of the line and increases the bandwidth to 8 kHz, we can send 112,000 bps by using the same technology as mentioned in Example 3.42.

3.6.2 Throughput

The **throughput** is a measure of how fast we can actually send data through a network. Although, at first glance, bandwidth in bits per second and throughput seem the same, they are different. A link may have a bandwidth of *B* bps, but we can only send *T* bps through this link with *T* always less than *B*. In other words, the bandwidth is a potential measurement of a link; the throughput is an actual measurement of how fast we can send data. For example, we may have a link with a bandwidth of 1 Mbps, but the devices connected to the end of the link may handle only 200 kbps. This means that we cannot send more than 200 kbps through this link.

Imagine a highway designed to transmit 1000 cars per minute from one point to another. However, if there is congestion on the road, this figure may be reduced to 100 cars per minute. The bandwidth is 1000 cars per minute; the throughput is 100 cars per minute.

Example 3.44

A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?

Solution

We can calculate the throughput as

$$\textbf{Throughput} = (12{,}000 \times 10{,}000) / 60 = 2 \text{ Mbps}$$

The throughput is almost one-fifth of the bandwidth in this case.

3.6.3 Latency (Delay)

The **latency** or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source. We can say that latency is made of four components: propagation time, transmission time, queuing time and processing delay.

$$\textbf{Latency} = \textbf{propagation time} + \textbf{transmission time} + \textbf{queuing time} + \textbf{processing delay}$$

Propagation Time

Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.

$$\textbf{Propagation time} = \textbf{Distance} / (\textbf{Propagation Speed})$$

The propagation speed of electromagnetic signals depends on the medium and on the frequency of the signal. For example, in a vacuum, light is propagated with a speed of 3×10^8 m/s. It is lower in air; it is much lower in cable.

Example 3.45

What is the propagation time if the distance between the two points is 12,000 km? Assume the propagation speed to be 2.4×10^8 m/s in cable.

Solution

We can calculate the propagation time as

$$\textbf{Propagation time} = (12{,}000 \times 10{,}000) / (2.4 \times 2^8) = 50 \text{ ms}$$

The example shows that a bit can go over the Atlantic Ocean in only 50 ms if there is a direct cable between the source and the destination.

Transmission Time

In data communications we don't send just 1 bit, we send a message. The first bit may take a time equal to the propagation time to reach its destination; the last bit also may take the same amount of time. However, there is a time between the first bit leaving the sender and the last bit arriving at the receiver. The first bit leaves earlier and arrives earlier; the last bit leaves later and arrives later. The **transmission time** of a message depends on the size of the message and the bandwidth of the channel.

$$\textbf{Transmission time} = \textbf{(Message size) / Bandwidth}$$

Example 3.46

What are the propagation time and the transmission time for a 2.5-KB (kilobyte) message (an e-mail) if the bandwidth of the network is 1 Gbps? Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4×10^8 m/s.

Solution

We can calculate the propagation and transmission time as

$$\textbf{Propagation time} = (12{,}000 \times 1000) / (2.4 \times 10^8) = 50 \text{ ms}$$

$$\textbf{Transmission time} = (2500 \times 8) / 10^9 = 0.020 \text{ ms}$$

Note that in this case, because the message is short and the bandwidth is high, the dominant factor is the propagation time, not the transmission time. The transmission time can be ignored.

Example 3.47

What are the propagation time and the transmission time for a 5-MB (megabyte) message (an image) if the bandwidth of the network is 1 Mbps? Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4×10^8 m/s.

Solution

We can calculate the propagation and transmission times as

$$\textbf{Propagation time} = (12{,}000 \times 1000) / (2.4 \times 10^8) = 50 \text{ ms}$$

$$\textbf{Transmission time} = (5{,}000{,}000 \times 8) / 10^6 = 40 \text{ s}$$

Note that in this case, because the message is very long and the bandwidth is not very high, the dominant factor is the transmission time, not the propagation time. The propagation time can be ignored.

Queuing Time

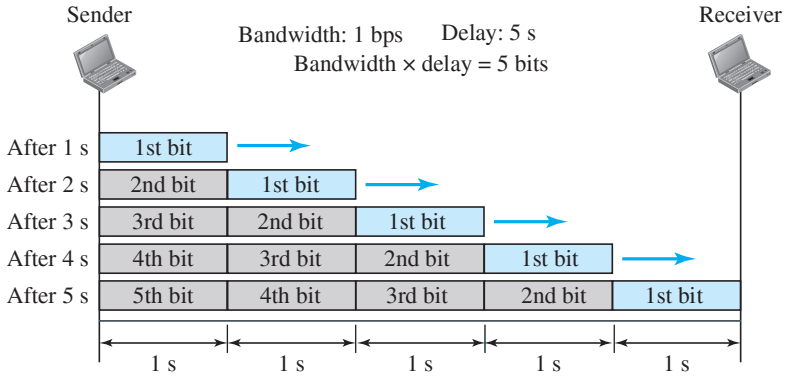
The third component in latency is the **queuing time**, the time needed for each intermediate or end device to hold the message before it can be processed. The queuing time is not a fixed factor; it changes with the load imposed on the network. When there is heavy traffic on the network, the queuing time increases. An intermediate device, such as a router, queues the arrived messages and processes them one by one. If there are many messages, each message will have to wait.

3.6.4 Bandwidth-Delay Product

Bandwidth and delay are two performance metrics of a link. However, as we will see in this chapter and future chapters, what is very important in data communications is the product of the two, the bandwidth-delay product. Let us elaborate on this issue, using two hypothetical cases as examples.

- **Case 1.** Figure 3.32 shows case 1.

Figure 3.32 *Filling the link with bits for case 1*

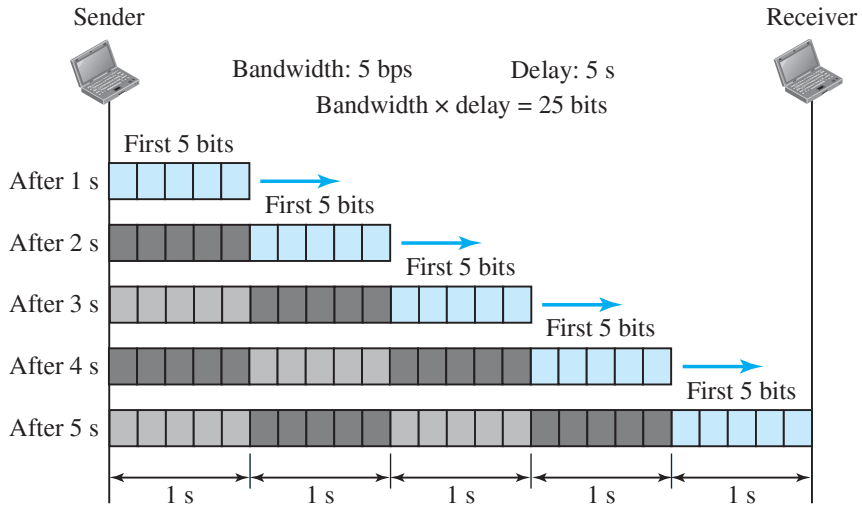


Let us assume that we have a link with a bandwidth of 1 bps (unrealistic, but good for demonstration purposes). We also assume that the delay of the link is 5 s (also unrealistic). We want to see what the bandwidth-delay product means in this case. Looking at the figure, we can say that this product 1×5 is the maximum number of bits that can fill the link. There can be no more than 5 bits at any time on the link.

- **Case 2.** Now assume we have a bandwidth of 5 bps. Figure 3.33 shows that there can be maximum $5 \times 5 = 25$ bits on the line. The reason is that, at each second, there are 5 bits on the line; the duration of each bit is 0.20 s.

The above two cases show that the product of bandwidth and delay is the number of bits that can fill the link. This measurement is important if we need to send data in bursts and wait for the acknowledgment of each burst before sending the next one. To use the maximum capability of the link, we need to make the size of our burst 2 times the product

Figure 3.33 *Filling the link with bits in case 2*



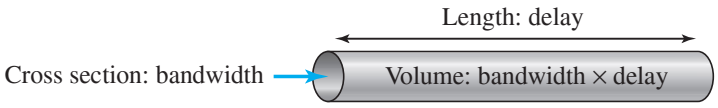
of bandwidth and delay; we need to fill up the full-duplex channel (two directions). The sender should send a burst of data of (2 × bandwidth × delay) bits. The sender then waits for receiver acknowledgment for part of the burst before sending another burst. The amount 2 × bandwidth × delay is the number of bits that can be in transition at any time.

The bandwidth-delay product defines the number of bits that can fill the link.

Example 3.48

We can think about the link between two points as a pipe. The cross section of the pipe represents the bandwidth, and the length of the pipe represents the delay. We can say the volume of the pipe defines the bandwidth-delay product, as shown in Figure 3.34.

Figure 3.34 *Concept of bandwidth-delay product*



3.6.5 Jitter

Another performance issue that is related to delay is **jitter.** We can roughly say that jitter is a problem if different packets of data encounter different delays and the application using the data at the receiver site is time-sensitive (audio and video data, for example). If the delay for the first packet is 20 ms, for the second is 45 ms, and for the third is 40 ms, then the real-time application that uses the packets endures jitter. We discuss jitter in greater detail in Chapter 28.

3.7 END-CHAPTER MATERIALS

3.7.1 Recommended Reading

For more details about subjects discussed in this chapter, we recommend the following books. The items in brackets [. . .] refer to the reference list at the end of the text.

Books

Data and signals are discussed in [Pea92]. [Cou01] gives excellent coverage of signals. More advanced materials can be found in [Ber96]. [Hsu03] gives a good mathematical approach to signaling. Complete coverage of Fourier Analysis can be found in [Spi74]. Data and signals are discussed in [Sta04] and [Tan03].

3.7.2 Key Terms

analog
analog data
analog signal
attenuation
bandpass channel
bandwidth
baseband transmission
bit length
bit rate
bits per second (bps)
broadband transmission
composite signal
cycle
data
decibel (dB)
digital
digital data
digital signal
distortion
Fourier analysis
frequency
frequency-domain
fundamental frequency
harmonic
Hertz (Hz)
jitter
latency
low-pass channel
noise
nonperiodic signal
Nyquist bit rate
peak amplitude
period
periodic signal
phase
processing delay
propagation speed
propagation time
queuing time
Shannon capacity
signal
signal-to-noise ratio (SNR)
sine wave
throughput
time-domain
transmission time
wavelength

3.7.3 Summary

Data must be transformed to electromagnetic signals to be transmitted. Data can be analog or digital. Analog data are continuous and take continuous values. Digital data have discrete states and take discrete values. Signals can be analog or digital. Analog signals can have an infinite number of values in a range; digital signals can have only a limited number of values.

In data communications, we commonly use periodic analog signals and nonperiodic digital signals. Frequency and period are the inverse of each other. Frequency is the rate of change with respect to time. Phase describes the position of the waveform relative to time 0. A complete sine wave in the time domain can be represented by one single spike in the frequency domain. A single-frequency sine wave is not useful in data communications; we need to send a composite signal, a signal made of many simple sine waves. According to Fourier analysis, any composite signal is a combination of simple sine waves with different frequencies, amplitudes, and phases. The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.

A digital signal is a composite analog signal with an infinite bandwidth. Baseband transmission of a digital signal that preserves the shape of the digital signal is possible only if we have a low-pass channel with an infinite or very wide bandwidth. If the available channel is a bandpass channel, we cannot send a digital signal directly to the channel; we need to convert the digital signal to an analog signal before transmission.

For a noiseless channel, the Nyquist bit rate formula defines the theoretical maximum bit rate. For a noisy channel, we need to use the Shannon capacity to find the maximum bit rate. Attenuation, distortion, and noise can impair a signal. Attenuation is the loss of a signal's energy due to the resistance of the medium. Distortion is the alteration of a signal due to the differing propagation speeds of each of the frequencies that make up a signal. Noise is the external energy that corrupts a signal. The bandwidth-delay product defines the number of bits that can fill the link.

3.8 PRACTICE SET

3.8.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

3.8.2 Questions

Q3-1. What is the relationship between period and frequency?

Q3-2. What does the amplitude of a signal measure? What does the frequency of a signal measure? What does the phase of a signal measure?

Q3-3. How can a composite signal be decomposed into its individual frequencies?

Q3-4. Name three types of transmission impairment.

Q3-5. Distinguish between baseband transmission and broadband transmission.

Q3-6. Distinguish between a low-pass channel and a band-pass channel.

Q3-7. What does the Nyquist theorem have to do with communications?

Q3-8. What does the Shannon capacity have to do with communications?

Q3-9. Why do optical signals used in fiber optic cables have a very short wave length?

Q3-10. Can we say whether a signal is periodic or nonperiodic by just looking at its frequency domain plot? How?

Q3-11. Is the frequency domain plot of a voice signal discrete or continuous?

Q3-12. Is the frequency domain plot of an alarm system discrete or continuous?

Q3-13. We send a voice signal from a microphone to a recorder. Is this baseband or broadband transmission?

Q3-14. We send a digital signal from one station on a LAN to another station. Is this baseband or broadband transmission?

Q3-15. We modulate several voice signals and send them through the air. Is this baseband or broadband transmission?

3.8.3 Problems

P3-1. Given the frequencies listed below, calculate the corresponding periods.

a. 24 Hz **b.** 8 MHz **c.** 140 KHz

P3-2. Given the following periods, calculate the corresponding frequencies.

a. 5 s **b.** 12 μs **c.** 220 ns

P3-3. What is the phase shift for the following?

a. A sine wave with the maximum amplitude at time zero

b. A sine wave with maximum amplitude after 1/4 cycle

c. A sine wave with zero amplitude after 3/4 cycle and increasing

P3-4. What is the bandwidth of a signal that can be decomposed into five sine waves with frequencies at 0, 20, 50, 100, and 200 Hz? All peak amplitudes are the same. Draw the bandwidth.

P3-5. A periodic composite signal with a bandwidth of 2000 Hz is composed of two sine waves. The first one has a frequency of 100 Hz with a maximum amplitude of 20 V; the second one has a maximum amplitude of 5 V. Draw the bandwidth.

P3-6. Which signal has a wider bandwidth, a sine wave with a frequency of 100 Hz or a sine wave with a frequency of 200 Hz?

P3-7. What is the bit rate for each of the following signals?

a. A signal in which 1 bit lasts 0.001 s

b. A signal in which 1 bit lasts 2 ms

c. A signal in which 10 bits last 20 μs

P3-8. A device is sending out data at the rate of 1000 bps.

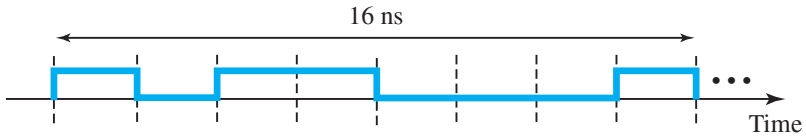
a. How long does it take to send out 10 bits?

b. How long does it take to send out a single character (8 bits)?

c. How long does it take to send a file of 100,000 characters?

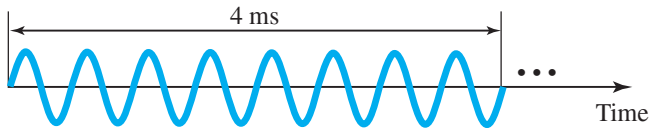
P3-9. What is the bit rate for the signal in Figure 3.35?

Figure 3.35 *Problem P3-9*



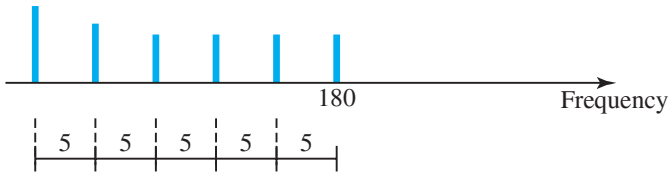
P3-10. What is the frequency of the signal in Figure 3.36?

Figure 3.36 *Problem P3-10*



P3-11. What is the bandwidth of the composite signal shown in Figure 3.37?

Figure 3.37 *Problem P3-11*



P3-12. A periodic composite signal contains frequencies from 10 to 30 KHz, each with an amplitude of 10 V. Draw the frequency spectrum.

P3-13. A nonperiodic composite signal contains frequencies from 10 to 30 KHz. The peak amplitude is 10 V for the lowest and the highest signals and is 30 V for the 20-KHz signal. Assuming that the amplitudes change gradually from the minimum to the maximum, draw the frequency spectrum.

P3-14. A TV channel has a bandwidth of 6 MHz. If we send a digital signal using one channel, what are the data rates if we use one harmonic, three harmonics, and five harmonics?

P3-15. A signal travels from point A to point B. At point A, the signal power is 100 W. At point B, the power is 90 W. What is the attenuation in decibels?

P3-16. The attenuation of a signal is −10 dB. What is the final signal power if it was originally 5 W?

P3-17. A signal has passed through three cascaded amplifiers, each with a 4 dB gain. What is the total gain? How much is the signal amplified?

P3-18. If the bandwidth of the channel is 5 Kbps, how long does it take to send a frame of 100,000 bits out of this device?

P3-19. The light of the sun takes approximately eight minutes to reach the earth. What is the distance between the sun and the earth?

P3-20. A signal has a wavelength of 1 μm in air. How far can the front of the wave travel during 1000 periods?

P3-21. A line has a signal-to-noise ratio of 1000 and a bandwidth of 4000 KHz. What is the maximum data rate supported by this line?

P3-22. We measure the performance of a telephone line (4 KHz of bandwidth). When the signal is 10 V, the noise is 5 mV. What is the maximum data rate supported by this telephone line?

P3-23. A file contains 2 million bytes. How long does it take to download this file using a 56-Kbps channel? 1-Mbps channel?

P3-24. A computer monitor has a resolution of 1200 by 1000 pixels. If each pixel uses 1024 colors, how many bits are needed to send the complete contents of a screen?

P3-25. A signal with 200 milliwatts power passes through 10 devices, each with an average noise of 2 microwatts. What is the SNR? What is the SNRdB?

P3-26. If the peak voltage value of a signal is 20 times the peak voltage value of the noise, what is the SNR? What is the SNR_{dB}?

P3-27. What is the theoretical capacity of a channel in each of the following cases?

a. Bandwidth: 20 KHz $SNR_{dB} = 40$

b. Bandwidth: 200 KHz $SNR_{dB} = 4$

c. Bandwidth: 1 MHz $SNR_{dB} = 20$

P3-28. We need to upgrade a channel to a higher bandwidth. Answer the following questions:

a. How is the rate improved if we double the bandwidth?

b. How is the rate improved if we double the SNR?

P3-29. We have a channel with 4 KHz bandwidth. If we want to send data at 100 Kbps, what is the minimum SNR_{dB}? What is the SNR?

P3-30. What is the transmission time of a packet sent by a station if the length of the packet is 1 million bytes and the bandwidth of the channel is 200 Kbps?

P3-31. What is the length of a bit in a channel with a propagation speed of 2×10^8 m/s if the channel bandwidth is

a. 1 Mbps? **b.** 10 Mbps? **c.** 100 Mbps?

P3-32. How many bits can fit on a link with a 2 ms delay if the bandwidth of the link is

a. 1 Mbps? **b.** 10 Mbps? **c.** 100 Mbps?

P3-33. What is the total delay (latency) for a frame of size 5 million bits that is being sent on a link with 10 routers each having a queuing time of 2 μs and a processing time of 1 μs. The length of the link is 2000 Km. The speed of light inside the link is 2×10^8 m/s. The link has a bandwidth of 5 Mbps. Which component of the total delay is dominant? Which one is negligible?

3.9 SIMULATION EXPERIMENTS

3.9.1 Applets

We have created some Java applets to show some of the main concepts discussed in this chapter. It is strongly recommended that the students activate these applets on the book website and carefully examine the protocols in action.

CHAPTER 4

Digital Transmission

A computer network is designed to send information from one point to another. This information needs to be converted to either a digital signal or an analog signal for transmission. In this chapter, we discuss the first choice, conversion to digital signals; in Chapter 5, we discuss the second choice, conversion to analog signals.

We discussed the advantages and disadvantages of digital transmission over analog transmission in Chapter 3. In this chapter, we show the schemes and techniques that we use to transmit data digitally. First, we discuss **digital-to-digital conversion** techniques, methods which convert digital data to digital signals. Second, we discuss **analog-to-digital conversion** techniques, methods which change an analog signal to a digital signal. Finally, we discuss **transmission modes.** We have divided this chapter into three sections:

- The first section discusses digital-to-digital conversion. Line coding is used to convert digital data to a digital signal. Several common schemes are discussed. The section also describes block coding, which is used to create redundancy in the digital data before they are encoded as a digital signal. Redundancy is used as an inherent error detecting tool. The last topic in this section discusses scrambling, a technique used for digital-to-digital conversion in long-distance transmission.
- The second section discusses analog-to-digital conversion. Pulse code modulation is described as the main method used to sample an analog signal. Delta modulation is used to improve the efficiency of the pulse code modulation.
- The third section discusses transmission modes. When we want to transmit data digitally, we need to think about parallel or serial transmission. In parallel transmission, we send multiple bits at a time; in serial transmission, we send one bit at a time.

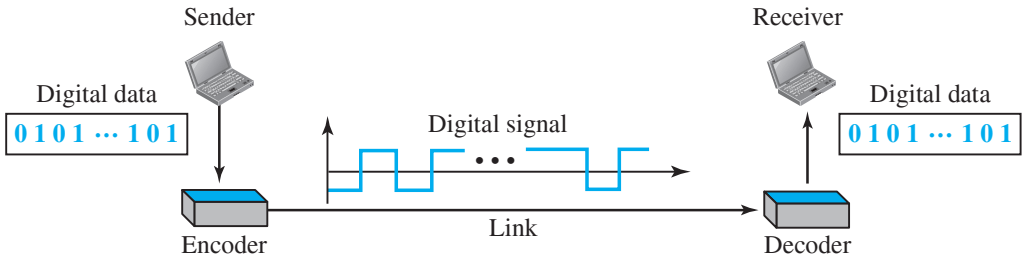
4.1 DIGITAL-TO-DIGITAL CONVERSION

In Chapter 3, we discussed data and signals. We said that data can be either digital or analog. We also said that signals that represent data can also be digital or analog. In this section, we see how we can represent digital data by using digital signals. The conversion involves three techniques: line coding, block coding, and scrambling. Line coding is always needed; block coding and scrambling may or may not be needed.

4.1.1 Line Coding

Line coding is the process of converting digital data to digital signals. We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits (see Chapter 1). Line coding converts a sequence of bits to a digital signal. At the sender, digital data are encoded into a digital signal; at the receiver, the digital data are recreated by decoding the digital signal. Figure 4.1 shows the process.

Figure 4.1 *Line coding and decoding*



Characteristics

Before discussing different line coding schemes, we address their common characteristics.

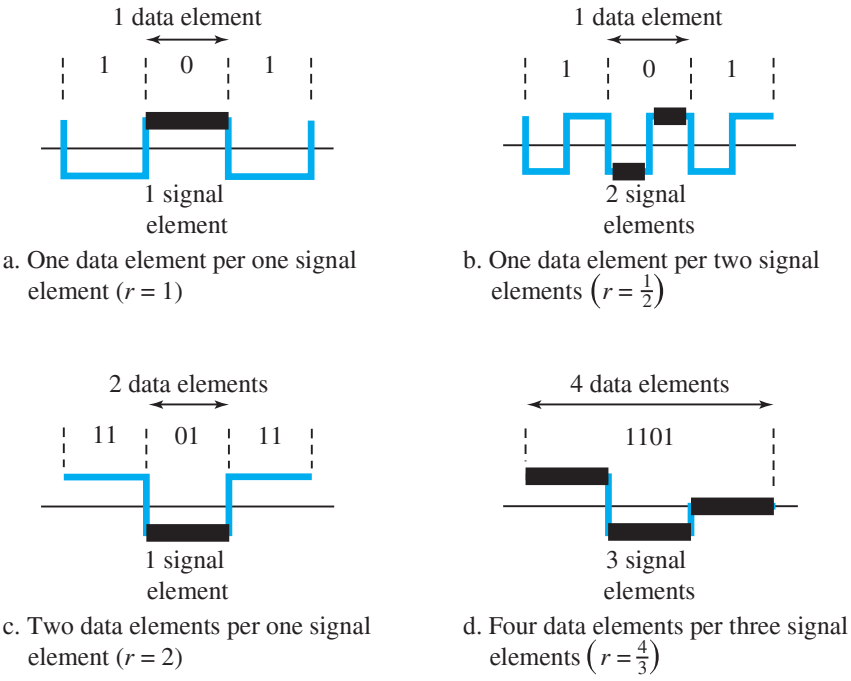
Signal Element Versus Data Element

Let us distinguish between a **data element** and a **signal element.** In data communications, our goal is to send data elements. A data element is the smallest entity that can represent a piece of information: this is the bit. In digital data communications, a signal element carries data elements. A signal element is the shortest unit (timewise) of a digital signal. In other words, data elements are what we need to send; signal elements are what we can send. Data elements are being carried; signal elements are the carriers.

We define a ratio r which is the number of data elements carried by each signal element. Figure 4.2 shows several situations with different values of r.

In part a of the figure, one data element is carried by one signal element ($r = 1$). In part b of the figure, we need two signal elements (two transitions) to carry each data element ($r = \frac{1}{2}$). We will see later that the extra signal element is needed to guarantee synchronization. In part c of the figure, a signal element carries two data elements ($r = 2$).

Figure 4.2 *Signal element versus data element*



Finally, in part d, a group of 4 bits is being carried by a group of three signal elements ($r = 4/3$). For every line coding scheme we discuss, we will give the value of r.

An analogy may help here. Suppose each data element is a person who needs to be carried from one place to another. We can think of a signal element as a vehicle that can carry people. When $r = 1$, it means each person is driving a vehicle. When $r > 1$, it means more than one person is travelling in a vehicle (a carpool, for example). We can also have the case where one person is driving a car and a trailer ($r = 1/2$).

Data Rate Versus Signal Rate

The **data rate** defines the number of data elements (bits) sent in 1s. The unit is bits per second (bps). The **signal rate** is the number of signal elements sent in 1s. The unit is the baud. There are several common terminologies used in the literature. The data rate is sometimes called the **bit rate;** the signal rate is sometimes called the **pulse rate,** the **modulation rate,** or the **baud rate.**

One goal in data communications is to increase the data rate while decreasing the signal rate. Increasing the data rate increases the speed of transmission; decreasing the signal rate decreases the bandwidth requirement. In our vehicle-people analogy, we need to carry more people in fewer vehicles to prevent traffic jams. We have a limited *bandwidth* in our transportation system.

We now need to consider the relationship between data rate (N) and signal rate (S)

$$S = N/r$$

in which r has been previously defined. This relationship, of course, depends on the value of r. It also depends on the data pattern. If we have a data pattern of all 1s or all 0s, the signal rate may be different from a data pattern of alternating 0s and 1s. To

derive a formula for the relationship, we need to define three cases: the worst, best, and average. The worst case is when we need the maximum signal rate; the best case is when we need the minimum. In data communications, we are usually interested in the average case. We can formulate the relationship between data rate and signal rate as

$$S_{ave} = c \times N \times (1/r) \quad \text{baud}$$

where N is the data rate (bps); c is the case factor, which varies for each case; S is the number of signal elements per second; and r is the previously defined factor.

Example 4.1

A signal is carrying data in which one data element is encoded as one signal element ($r = 1$). If the bit rate is 100 kbps, what is the average value of the baud rate if c is between 0 and 1?

Solution

We assume that the average value of c is 1/2. The baud rate is then

$$S = c \times N \times (1 / r) = 1/2 \times 100{,}000 \times (1/1) = 50{,}000 = 50 \text{ kbaud}$$

Bandwidth

We discussed in Chapter 3 that a digital signal that carries information is nonperiodic. We also showed that the bandwidth of a nonperiodic signal is continuous with an infinite range. However, most digital signals we encounter in real life have a bandwidth with finite values. In other words, the bandwidth is theoretically infinite, but many of the components have such a small amplitude that they can be ignored. The effective bandwidth is finite. From now on, when we talk about the bandwidth of a digital signal, we need to remember that we are talking about this effective bandwidth.

Although the actual bandwidth of a digital signal is infinite, the effective bandwidth is finite.

We can say that the baud rate, not the bit rate, determines the required bandwidth for a digital signal. If we use the transportation analogy, the number of vehicles, not the number of people being carried, affects the traffic. More changes in the signal mean injecting more frequencies into the signal. (Recall that frequency means change and change means frequency.) The bandwidth reflects the range of frequencies we need. There is a relationship between the baud rate (signal rate) and the bandwidth. Bandwidth is a complex idea. When we talk about the bandwidth, we normally define a range of frequencies. We need to know where this range is located as well as the values of the lowest and the highest frequencies. In addition, the amplitude (if not the phase) of each component is an important issue. In other words, we need more information about the bandwidth than just its value; we need a diagram of the bandwidth. We will show the bandwidth for most schemes we discuss in the chapter. For the moment, we can say that the bandwidth (range of frequencies) is proportional to the signal rate (baud rate). The minimum bandwidth can be given as

$$B_{min} = c \times N \times (1 / r)$$

We can solve for the maximum data rate if the bandwidth of the channel is given.

$$N_{max} = (1 / c) \times B \times r$$

Example 4.2

The maximum data rate of a channel (see Chapter 3) is $N_{max} = 2 \times B \times \log_2 L$ (defined by the Nyquist formula). Does this agree with the previous formula for N_{max}?

Solution

A signal with L levels actually can carry $\log_2 L$ bits per level. If each level corresponds to one signal element and we assume the average case ($c = 1/2$), then we have

$$N_{max} = (1/c) \times B \times r = 2 \times B \times \log_2 L$$

Baseline Wandering

In decoding a digital signal, the receiver calculates a running average of the received signal power. This average is called the ***baseline*.** The incoming signal power is evaluated against this baseline to determine the value of the data element. A long string of 0s or 1s can cause a drift in the baseline (**baseline wandering**) and make it difficult for the receiver to decode correctly. A good line coding scheme needs to prevent baseline wandering.

DC Components

When the voltage level in a digital signal is constant for a while, the spectrum creates very low frequencies (results of Fourier analysis). These frequencies around zero, called DC (direct-current) *components,* present problems for a system that cannot pass low frequencies or a system that uses electrical coupling (via a transformer). We can say that DC component means 0/1 parity that can cause base-line wondering. For example, a telephone line cannot pass frequencies below 200 Hz. Also a long-distance link may use one or more transformers to isolate different parts of the line electrically. For these systems, we need a scheme with no **DC component.**

Self-synchronization

To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals. If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might misinterpret the signals. Figure 4.3 shows a situation in which the receiver has a shorter bit duration. The sender sends 10110001, while the receiver receives 110111000011.

A **self-synchronizing** digital signal includes timing information in the data being transmitted. This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle, or end of the pulse. If the receiver's clock is out of synchronization, these points can reset the clock.

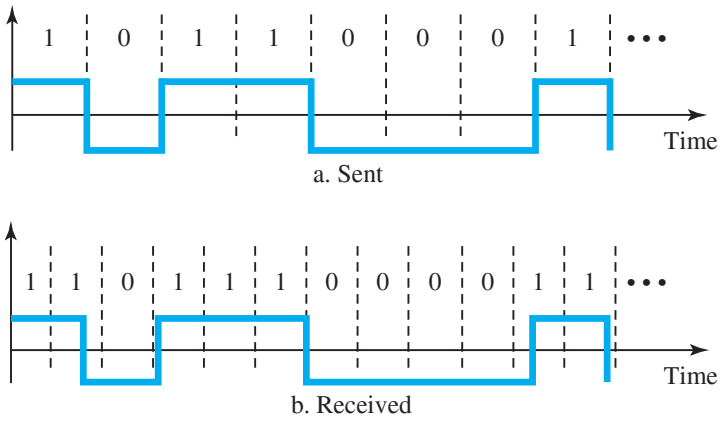
Example 4.3

In a digital transmission, the receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 kbps? How many if the data rate is 1 Mbps?

Solution

At 1 kbps, the receiver receives 1001 bps instead of 1000 bps.

Figure 4.3 *Effect of lack of synchronization*



1000 bits sent → 1001 bits received → 1 extra bps

At 1 Mbps, the receiver receives 1,001,000 bps instead of 1,000,000 bps.

1,000,000 bits sent → 1,001,000 bits received → 1000 extra bps

Built-in Error Detection

It is desirable to have a built-in error-detecting capability in the generated code to detect some or all of the errors that occurred during transmission. Some encoding schemes that we will discuss have this capability to some extent.

Immunity to Noise and Interference

Another desirable code characteristic is a code that is immune to noise and other interferences. Some encoding schemes that we will discuss have this capability.

Complexity

A complex scheme is more costly to implement than a simple one. For example, a scheme that uses four signal levels is more difficult to interpret than one that uses only two levels.

4.1.2 Line Coding Schemes

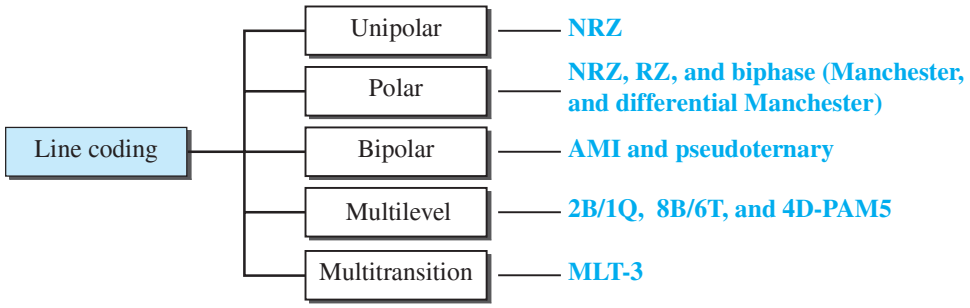
We can roughly divide line coding schemes into five broad categories, as shown in Figure 4.4.

There are several schemes in each category. We need to be familiar with all schemes discussed in this section to understand the rest of the book. This section can be used as a reference for schemes encountered later.

Unipolar Scheme

In a **unipolar** scheme, all the signal levels are on one side of the time axis, either above or below.

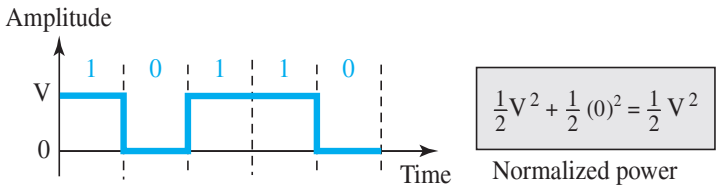
Figure 4.4 *Line coding schemes*



NRZ (Non-Return-to-Zero)

Traditionally, a unipolar scheme was designed as a **non-return-to-zero (NRZ)** scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. It is called NRZ because the signal does not return to zero at the middle of the bit. Figure 4.5 shows a unipolar NRZ scheme.

Figure 4.5 *Unipolar NRZ scheme*



Compared with its polar counterpart (see the next section), this scheme is very costly. As we will see shortly, the normalized power (the power needed to send 1 bit per unit line resistance) is double that for polar NRZ. For this reason, this scheme is normally not used in data communications today.

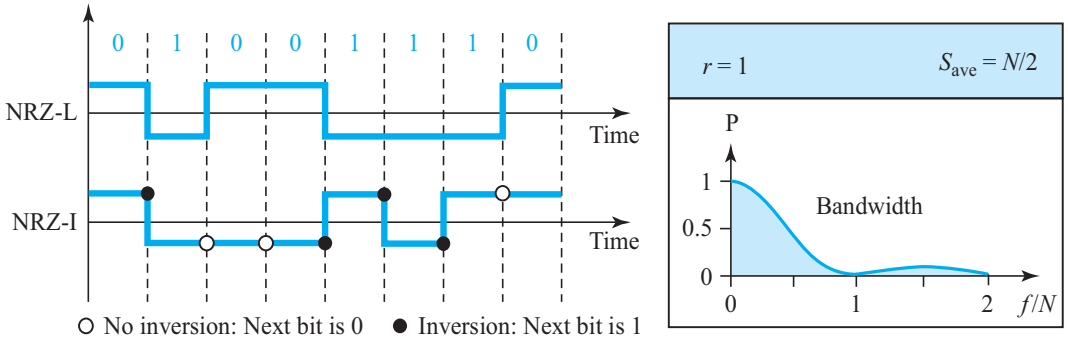
Polar Schemes

In **polar** schemes, the voltages are on both sides of the time axis. For example, the voltage level for 0 can be positive and the voltage level for 1 can be negative.

Non-Return-to-Zero (NRZ)

In **polar NRZ** encoding, we use two levels of voltage amplitude. We can have two versions of polar NRZ: NRZ-L and NRZ-I, as shown in Figure 4.6. The figure also shows the value of r, the average baud rate, and the bandwidth. In the first variation, NRZ-L (**NRZ-Level**), the level of the voltage determines the value of the bit. In the second variation, NRZ-I (**NRZ-Invert**), the change or lack of change in the level of the voltage determines the value of the bit. If there is no change, the bit is 0; if there is a change, the bit is 1.

Figure 4.6 *Polar NRZ-L and NRZ-I schemes*



In NRZ-L the level of the voltage determines the value of the bit. In NRZ-I the inversion or the lack of inversion determines the value of the bit.

Let us compare these two schemes based on the criteria we previously defined. Although baseline wandering is a problem for both variations, it is twice as severe in NRZ-L. If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes skewed. The receiver might have difficulty discerning the bit value. In NRZ-I this problem occurs only for a long sequence of 0s. If somehow we can eliminate the long sequence of 0s, we can avoid baseline wandering. We will see shortly how this can be done.

The synchronization problem (sender and receiver clocks are not synchronized) also exists in both schemes. Again, this problem is more serious in NRZ-L than in NRZ-I. While a long sequence of 0s can cause a problem in both schemes, a long sequence of 1s affects only NRZ-L.

Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. For example, if twisted-pair cable is the medium, a change in the polarity of the wire results in all 0s interpreted as 1s and all 1s interpreted as 0s. NRZ-I does not have this problem. Both schemes have an average signal rate of $N/2$ Bd.

NRZ-L and NRZ-I both have an average signal rate of $N/2$ Bd.

Let us discuss the bandwidth. Figure 4.6 also shows the normalized bandwidth for both variations. The vertical axis shows the power density (the power for each 1 Hz of bandwidth); the horizontal axis shows the frequency. The bandwidth reveals a very serious problem for this type of encoding. The value of the power density is very high around frequencies close to zero. This means that there are DC components that carry a high level of energy. As a matter of fact, most of the energy is concentrated in frequencies between 0 and $N/2$. This means that although the average of the signal rate is $N/2$, the energy is not distributed evenly between the two halves.

NRZ-L and NRZ-I both have a DC component problem.

Example 4.4

A system is using NRZ-I to transfer 10-Mbps data. What are the average signal rate and minimum bandwidth?

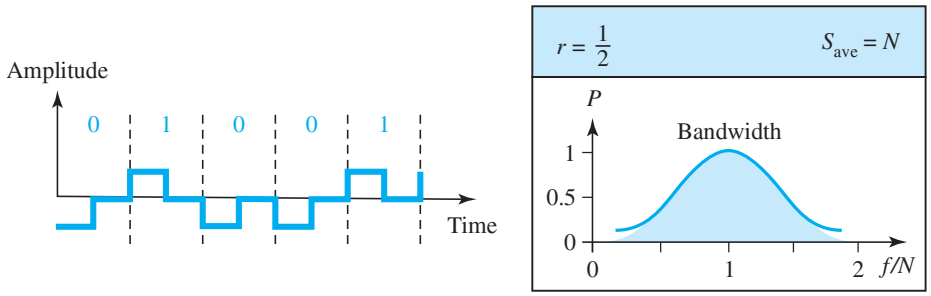
Solution

The average signal rate is $S = N/2 = 500$ kbaud. The minimum bandwidth for this average baud rate is $B_{min} = S = 500$ kHz.

Return-to-Zero (RZ)

The main problem with NRZ encoding occurs when the sender and receiver clocks are not synchronized. The receiver does not know when one bit has ended and the next bit is starting. One solution is the **return-to-zero (RZ)** scheme, which uses three values: positive, negative, and zero. In RZ, the signal changes not between bits but during the bit. In Figure 4.7 we see that the signal goes to 0 in the middle of each bit. It remains there until the beginning of the next bit. The main disadvantage of RZ encoding is that it requires two signal changes to encode a bit and therefore occupies greater bandwidth. The same problem we mentioned, a sudden change of polarity resulting in all 0s interpreted as 1s and all 1s interpreted as 0s, still exists here, but there is no DC component problem. Another problem is the complexity: RZ uses three levels of voltage, which is more complex to create and discern. As a result of all these deficiencies, the scheme is not used today. Instead, it has been replaced by the better-performing Manchester and differential Manchester schemes (discussed next).

Figure 4.7 *Polar RZ scheme*

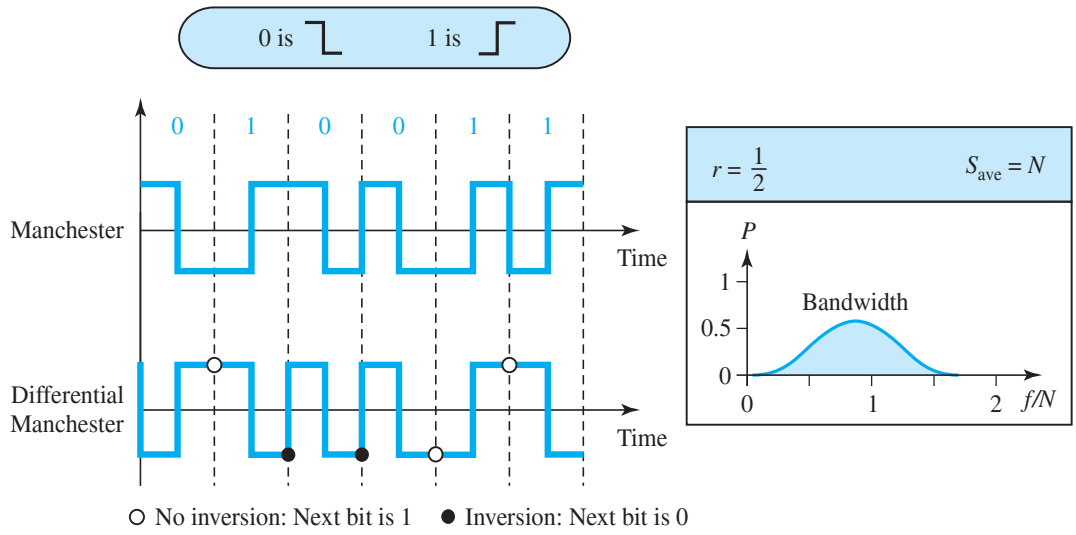


Biphase: Manchester and Differential Manchester

The idea of RZ (transition at the middle of the bit) and the idea of NRZ-L are combined into the **Manchester** scheme. In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization. **Differential Manchester,** on the other hand, combines the ideas of RZ and NRZ-I. There is always a transition at the middle of the bit, but the bit values are determined at the beginning of the bit. If the next bit is 0, there is a transition; if the next bit is 1, there is none. Figure 4.8 shows both Manchester and differential Manchester encoding.

The Manchester scheme overcomes several problems associated with NRZ-L, and differential Manchester overcomes several problems associated with NRZ-I. First, there

Figure 4.8 *Polar biphase: Manchester and differential Manchester schemes*



In Manchester and differential Manchester encoding, the transition at the middle of the bit is used for synchronization.

is no baseline wandering. There is no DC component because each bit has a positive and negative voltage contribution. The only drawback is the signal rate. The signal rate for Manchester and differential Manchester is double that for NRZ. The reason is that there is always one transition at the middle of the bit and maybe one transition at the end of each bit. Figure 4.8 shows both Manchester and differential Manchester encoding schemes. Note that Manchester and differential Manchester schemes are also called **biphase** schemes.

The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ.

Bipolar Schemes

In **bipolar** encoding (sometimes called ***multilevel binary***), there are three voltage levels: positive, negative, and zero. The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.

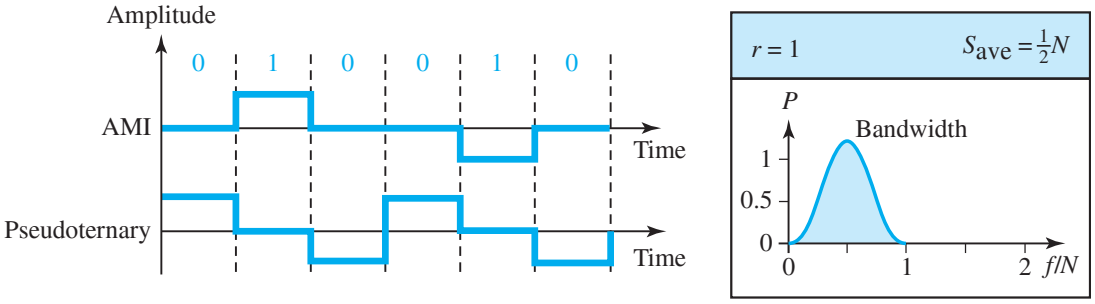
In bipolar encoding, we use three levels: positive, zero, and negative.

AMI and Pseudoternary

Figure 4.9 shows two variations of bipolar encoding: AMI and pseudoternary. A common bipolar encoding scheme is called bipolar **alternate mark inversion (AMI).** In the term *alternate mark inversion,* the word *mark* comes from telegraphy and means 1. So AMI means alternate 1 inversion. A neutral zero voltage represents binary 0. Binary

1s are represented by alternating positive and negative voltages. A variation of AMI encoding is called **pseudoternary** in which the 1 bit is encoded as a zero voltage and the 0 bit is encoded as alternating positive and negative voltages.

Figure 4.9 *Bipolar schemes: AMI and pseudoternary*



The bipolar scheme was developed as an alternative to NRZ. The bipolar scheme has the same signal rate as NRZ, but there is no DC component. The NRZ scheme has most of its energy concentrated near zero frequency, which makes it unsuitable for transmission over channels with poor performance around this frequency. The concentration of the energy in bipolar encoding is around frequency $N/2$. Figure 4.9 shows the typical energy concentration for a bipolar scheme.

One may ask why we do not have a DC component in bipolar encoding. We can answer this question by using the Fourier transform, but we can also think about it intuitively. If we have a long sequence of 1s, the voltage level alternates between positive and negative; it is not constant. Therefore, there is no DC component. For a long sequence of 0s, the voltage remains constant, but its amplitude is zero, which is the same as having no DC component. In other words, a sequence that creates a constant zero voltage does not have a DC component.

AMI is commonly used for long-distance communication, but it has a synchronization problem when a long sequence of 0s is present in the data. Later in the chapter, we will see how a scrambling technique can solve this problem.

Multilevel Schemes

The desire to increase the data rate or decrease the required bandwidth has resulted in the creation of many schemes. The goal is to increase the number of bits per baud by encoding a pattern of m data elements into a pattern of n signal elements. We only have two types of data elements (0s and 1s), which means that a group of m data elements can produce a combination of 2^m data patterns. We can have different types of signal elements by allowing different signal levels. If we have L different levels, then we can produce L^n combinations of signal patterns. If $2^m = L^n$, then each data pattern is encoded into one signal pattern. If $2^m < L^n$, data patterns occupy only a subset of signal patterns. The subset can be carefully designed to prevent baseline wandering, to provide synchronization, and to detect errors that occurred during data transmission. Data encoding is not possible if $2^m > L^n$ because some of the data patterns cannot be encoded.

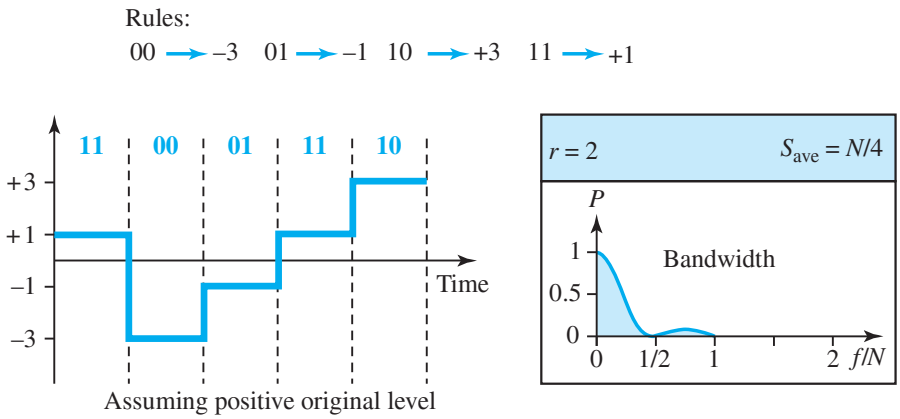
The code designers have classified these types of coding as *mBnL*, where *m* is the length of the binary pattern, *B* means binary data, *n* is the length of the signal pattern, and *L* is the number of levels in the signaling. A letter is often used in place of *L*: *B* (binary) for $L = 2$, *T* (ternary) for $L = 3$, and *Q* (quaternary) for $L = 4$. Note that the first two letters define the data pattern, and the second two define the signal pattern.

In *mBnL* schemes, a pattern of *m* data elements is encoded as a pattern of *n* signal elements in which $2^m \leq L^n$.

2B1Q

The first *mBnL* scheme we discuss, **two binary, one quaternary (2B1Q),** uses data patterns of size 2 and encodes the 2-bit patterns as one signal element belonging to a four-level signal. In this type of encoding $m = 2$, $n = 1$, and $L = 4$ (quaternary). Figure 4.10 shows an example of a 2B1Q signal.

Figure 4.10 *Multilevel: 2B1Q scheme*



The average signal rate of 2B1Q is $S = N/4$. This means that using 2B1Q, we can send data 2 times faster than by using NRZ-L. However, 2B1Q uses four different signal levels, which means the receiver has to discern four different thresholds. The reduced bandwidth comes with a price. There are no redundant signal patterns in this scheme because $2^2 = 4^1$.

The 2B1Q scheme is used in DSL (Digital Subscriber Line) technology to provide a high-speed connection to the Internet by using subscriber telephone lines (see Chapter 14).

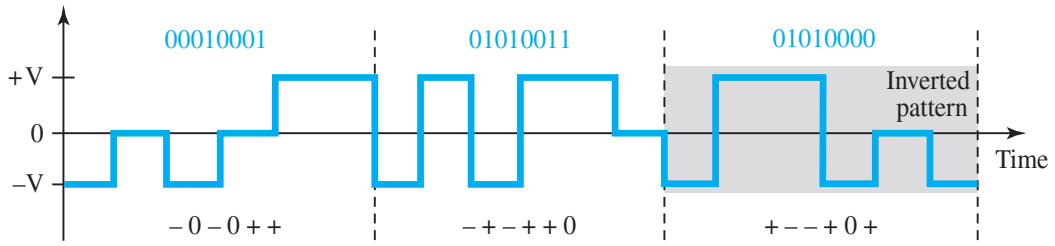
8B6T

A very interesting scheme is **eight binary, six ternary (8B6T).** This code is used with 100BASE-4T cable, as we will see in Chapter 13. The idea is to encode a pattern of 8 bits as a pattern of six signal elements, where the signal has three levels (ternary). In this type of scheme, we can have $2^8 = 256$ different data patterns and $3^6 = 729$ different signal patterns. The mapping table is shown in Appendix F. There are $729 - 256 = 473$ redundant signal elements that provide synchronization and error detection. Part of the

redundancy is also used to provide DC balance. Each signal pattern has a weight of 0 or +1 DC values. This means that there is no pattern with the weight −1. To make the whole stream DC-balanced, the sender keeps track of the weight. If two groups of weight 1 are encountered one after another, the first one is sent as is, while the next one is totally inverted to give a weight of −1.

Figure 4.11 shows an example of three data patterns encoded as three signal patterns. The three possible signal levels are represented as −, 0, and +. The first 8-bit pattern 00010001 is encoded as the signal pattern − 0 − 0 + + with weight 0; the second 8-bit pattern 01010011 is encoded as − + − + + 0 with weight +1. The third 8-bit pattern 01010000 should be encoded as + − − + 0 + with weight +1. To create DC balance, the sender inverts the actual signal. The receiver can easily recognize that this is an inverted pattern because the weight is −1. The pattern is inverted before decoding.

Figure 4.11 *Multilevel: 8B6T scheme*



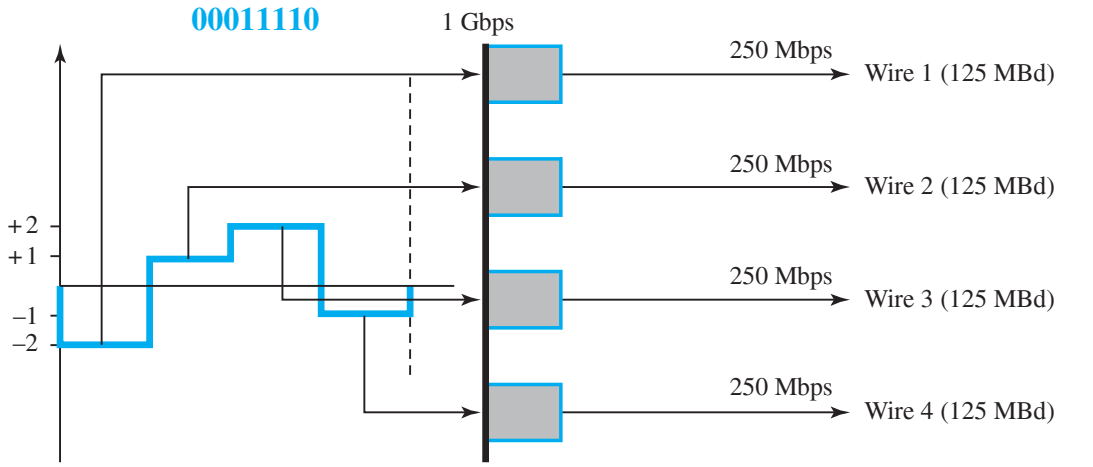
The average signal rate of the scheme is theoretically $S_{ave} = \frac{1}{2} \times N \times \frac{6}{8}$; in practice the minimum bandwidth is very close to 6*N*/8.

4D-PAM5

The last signaling scheme we discuss in this category is called **four-dimensional five-level pulse amplitude modulation (4D-PAM5).** The 4D means that data is sent over four wires at the same time. It uses five voltage levels, such as −2, −1, 0, 1, and 2. However, one level, level 0, is used only for forward error detection (discussed in Chapter 10). If we assume that the code is just one-dimensional, the four levels create something similar to 8B4Q. In other words, an 8-bit word is translated to a signal element of four different levels. The worst signal rate for this imaginary one-dimensional version is $N \times 4/8$, or $N/2$.

The technique is designed to send data over four channels (four wires). This means the signal rate can be reduced to *N*/8, a significant achievement. All 8 bits can be fed into a wire simultaneously and sent by using one signal element. The point here is that the four signal elements comprising one signal group are sent simultaneously in a four-dimensional setting. Figure 4.12 shows the imaginary one-dimensional and the actual four-dimensional implementation. Gigabit LANs (see Chapter 13) use this technique to send 1-Gbps data over four copper cables that can handle 125 Mbaud. This scheme has a lot of redundancy in the signal pattern because 2^8 data patterns are matched to $4^4 = 256$ signal patterns. The extra signal patterns can be used for other purposes such as error detection.

Figure 4.12 *Multilevel: 4D-PAM5 scheme*



Multitransition: MLT-3

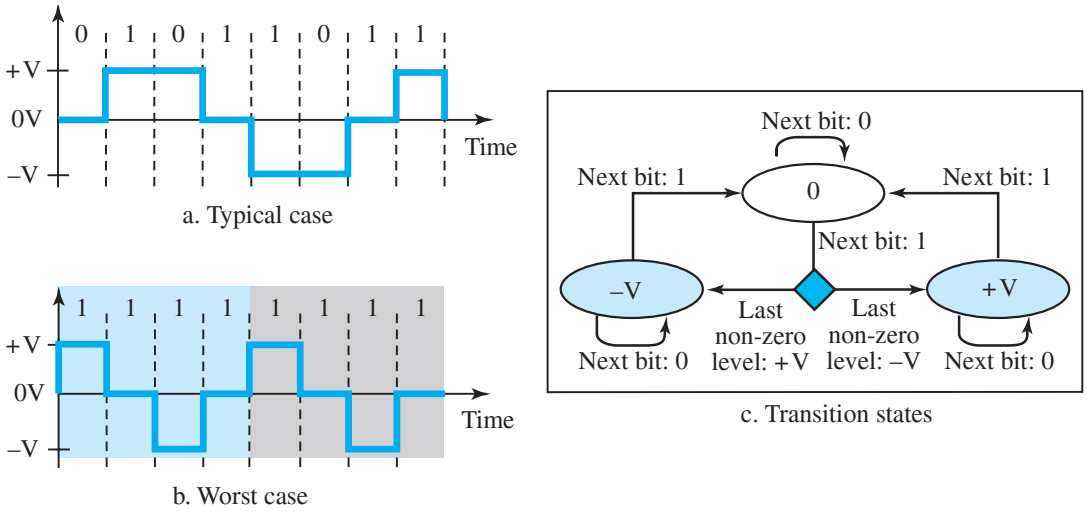
NRZ-I and differential Manchester are classified as differential encoding but use two transition rules to encode binary data (no inversion, inversion). If we have a signal with more than two levels, we can design a differential encoding scheme with more than two transition rules. MLT-3 is one of them. The **multiline transmission, three-level (MLT-3) scheme uses** three levels ($+V$, 0, and $-V$) and three transition rules to move between the levels.

1. If the next bit is 0, there is no transition.
2. If the next bit is 1 and the current level is not 0, the next level is 0.
3. If the next bit is 1 and the current level is 0, the next level is the opposite of the last nonzero level.

The behavior of MLT-3 can best be described by the state diagram shown in Figure 4.13. The three voltage levels ($-V$, 0, and $+V$) are shown by three states (ovals). The transition from one state (level) to another is shown by the connecting lines. Figure 4.13 also shows two examples of an MLT-3 signal.

One might wonder why we need to use MLT-3, a scheme that maps one bit to one signal element. The signal rate is the same as that for NRZ-I, but with greater complexity (three levels and complex transition rules). It turns out that the shape of the signal in this scheme helps to reduce the required bandwidth. Let us look at the worst-case scenario, a sequence of 1s. In this case, the signal element pattern $+V\,0\,-V\,0$ is repeated every 4 bits. A nonperiodic signal has changed to a periodic signal with the period equal to 4 times the bit duration. This worst-case situation can be simulated as an analog signal with a frequency one-fourth of the bit rate. In other words, the signal rate for MLT-3 is one-fourth the bit rate. This makes MLT-3 a suitable choice when we need to send 100 Mbps on a copper wire that cannot support more than 32 MHz (frequencies above this level create electromagnetic emissions). MLT-3 and LANs are discussed in Chapter 13.

Figure 4.13 *Multitransition: MLT-3 scheme*



Summary of Line Coding Schemes

We summarize in Table 4.1 the characteristics of the different schemes discussed.

Table 4.1 *Summary of line coding schemes*

Category	*Scheme*	*Bandwidth (average)*	*Characteristics*
Unipolar	NRZ	$B = N/2$	Costly, no self-synchronization if long 0s or 1s, DC
Polar	NRZ-L	$B = N/2$	No self-synchronization if long 0s or 1s, DC
	NRZ-I	$B = N/2$	No self-synchronization for long 0s, DC
	Biphase	$B = N$	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	$B = N/2$	No self-synchronization for long 0s, DC
Multilevel	2B1Q	$B = N/4$	No self-synchronization for long same double bits
	8B6T	$B = 3N/4$	Self-synchronization, no DC
	4D-PAM5	$B = N/8$	Self-synchronization, no DC
Multitransition	MLT-3	$B = N/3$	No self-synchronization for long 0s

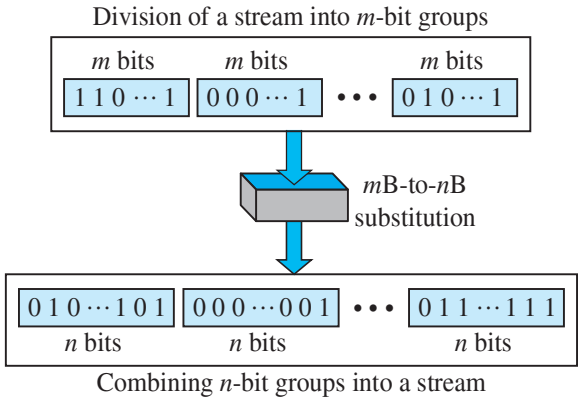
4.1.3 Block Coding

We need redundancy to ensure synchronization and to provide some kind of inherent error detecting. Block coding can give us this redundancy and improve the performance of line coding. In general, **block coding** changes a block of m bits into a block of n bits, where n is larger than m. Block coding is referred to as an mB/nB encoding technique.

Block coding is normally referred to as *mB/nB* coding; it replaces each *m*-bit group with an *n*-bit group.

The slash in block encoding (for example, 4B/5B) distinguishes block encoding from multilevel encoding (for example, 8B6T), which is written without a slash. Block coding normally involves three steps: division, substitution, and combination. In the division step, a sequence of bits is divided into groups of *m* bits. For example, in 4B/5B encoding, the original bit sequence is divided into 4-bit groups. The heart of block coding is the substitution step. In this step, we substitute an *m*-bit group with an *n*-bit group. For example, in 4B/5B encoding we substitute a 4-bit group with a 5-bit group. Finally, the *n*-bit groups are combined to form a stream. The new stream has more bits than the original bits. Figure 4.14 shows the procedure.

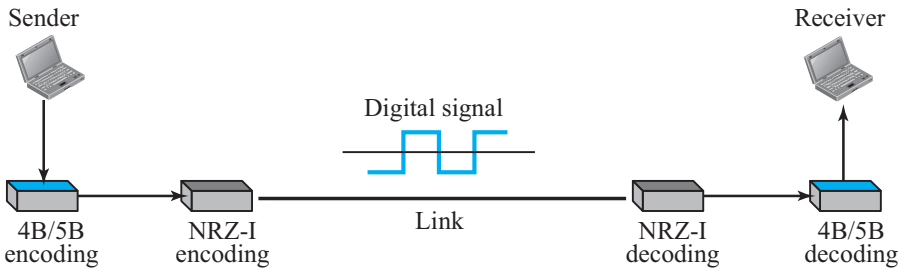
Figure 4.14 *Block coding concept*



4B/5B

The **four binary/five binary (4B/5B)** coding scheme was designed to be used in combination with NRZ-I. Recall that NRZ-I has a good signal rate, one-half that of the biphase, but it has a synchronization problem. A long sequence of 0s can make the receiver clock lose synchronization. One solution is to change the bit stream, prior to encoding with NRZ-I, so that it does not have a long stream of 0s. The 4B/5B scheme achieves this goal. The block-coded stream does not have more that three consecutive 0s, as we will see later. At the receiver, the NRZ-I encoded digital signal is first decoded into a stream of bits and then decoded to remove the redundancy. Figure 4.15 shows the idea.

Figure 4.15 *Using block coding 4B/5B with NRZ-I line coding scheme*



In 4B/5B, the 5-bit output that replaces the 4-bit input has no more than one leading zero (left bit) and no more than two trailing zeros (right bits). So when different groups are combined to make a new sequence, there are never more than three consecutive 0s. (Note that NRZ-I has no problem with sequences of 1s.) Table 4.2 shows the corresponding pairs used in 4B/5B encoding. Note that the first two columns pair a 4-bit group with a 5-bit group. A group of 4 bits can have only 16 different combinations while a group of 5 bits can have 32 different combinations. This means that there are 16 groups that are not used for 4B/5B encoding. Some of these unused groups are used for control purposes; the others are not used at all. The latter provide a kind of error detection. If a 5-bit group arrives that belongs to the unused portion of the table, the receiver knows that there is an error in the transmission.

Table 4.2 *4B/5B mapping codes*

Data Sequence	*Encoded Sequence*	*Control Sequence*	*Encoded Sequence*
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

Figure 4.16 shows an example of substitution in 4B/5B coding. 4B/5B encoding solves the problem of synchronization and overcomes one of the deficiencies of NRZ-I. However, we need to remember that it increases the signal rate of NRZ-I. The redundant bits add 20 percent more baud. Still, the result is less than the biphase scheme which has a signal rate of 2 times that of NRZ-I. However, 4B/5B block encoding does not solve the DC component problem of NRZ-I. If a DC component is unacceptable, we need to use biphase or bipolar encoding.

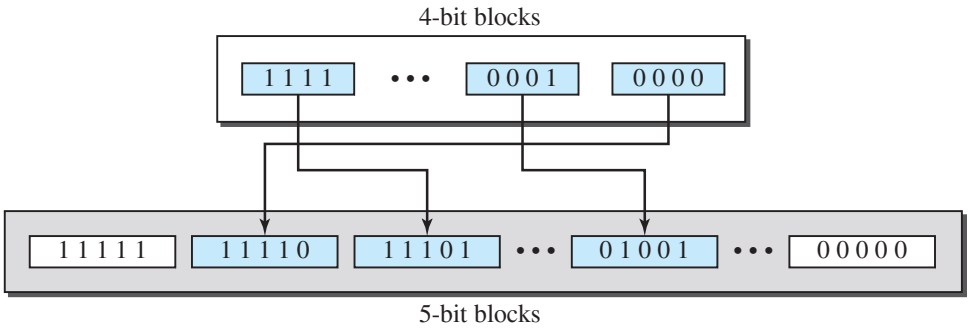
Example 4.5

We need to send data at a 1-Mbps rate. What is the minimum required bandwidth, using a combination of 4B/5B and NRZ-I or Manchester coding?

Solution

First 4B/5B block coding increases the bit rate to 1.25 Mbps. The minimum bandwidth using NRZ-I is $N/2$ or 625 kHz. The Manchester scheme needs a minimum bandwidth of 1 MHz. The

Figure 4.16 *Substitution in 4B/5B block coding*

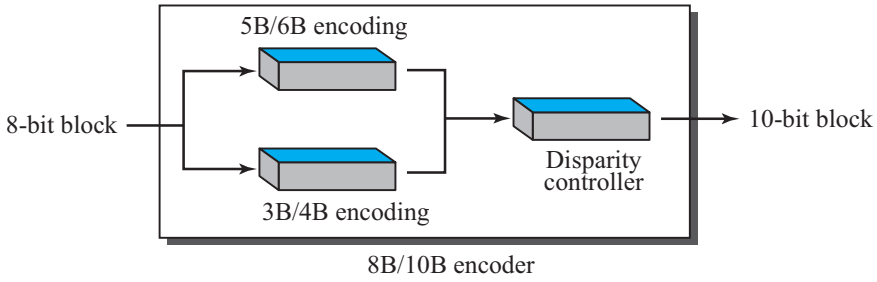


first choice needs a lower bandwidth, but has a DC component problem; the second choice needs a higher bandwidth, but does not have a DC component problem.

8B/10B

The **eight binary/ten binary (8B/10B)** encoding is similar to 4B/5B encoding except that a group of 8 bits of data is now substituted by a 10-bit code. It provides greater error detection capability than 4B/5B. The 8B/10B block coding is actually a combination of 5B/6B and 3B/4B encoding, as shown in Figure 4.17.

Figure 4.17 *8B/10B block encoding*

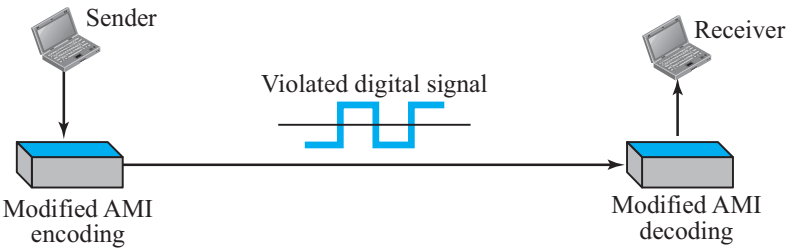


The five most significant bits of a 10-bit block are fed into the 5B/6B encoder; the three least significant bits are fed into a 3B/4B encoder. The split is done to simplify the mapping table. To prevent a long run of consecutive 0s or 1s, the code uses a disparity controller which keeps track of excess 0s over 1s (or 1s over 0s). If the bits in the current block create a disparity that contributes to the previous disparity (either direction), then each bit in the code is complemented (a 0 is changed to a 1 and a 1 is changed to a 0). The coding has $2^{10} - 2^8 = 768$ redundant groups that can be used for disparity checking and error detection. In general, the technique is superior to 4B/5B because of better built-in error-checking capability and better synchronization.

4.1.4 Scrambling

Biphase schemes that are suitable for dedicated links between stations in a LAN are not suitable for long-distance communication because of their wide bandwidth requirement. The combination of block coding and NRZ line coding is not suitable for long-distance encoding either, because of the DC component. Bipolar AMI encoding, on the other hand, has a narrow bandwidth and does not create a DC component. However, a long sequence of 0s upsets the synchronization. If we can find a way to avoid a long sequence of 0s in the original stream, we can use bipolar AMI for long distances. We are looking for a technique that does not increase the number of bits and does provide synchronization. We are looking for a solution that substitutes long zero-level pulses with a combination of other levels to provide synchronization. One solution is called **scrambling.** We modify part of the AMI rule to include scrambling, as shown in Figure 4.18. Note that scrambling, as opposed to block coding, is done at the same time as encoding. The system needs to insert the required pulses based on the defined scrambling rules. Two common scrambling techniques are B8ZS and HDB3.

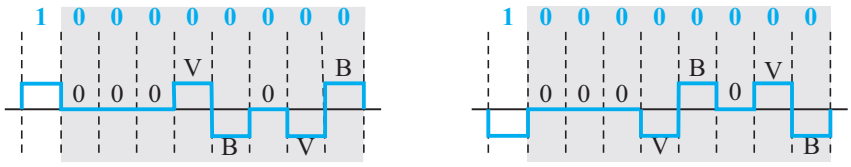
Figure 4.18 *AMI used with scrambling*



B8ZS

Bipolar with 8-zero substitution (B8ZS) is commonly used in North America. In this technique, eight consecutive zero-level voltages are replaced by the sequence **000VB0VB.** The V in the sequence denotes *violation;* this is a nonzero voltage that breaks an AMI rule of encoding (opposite polarity from the previous). The B in the sequence denotes *bipolar,* which means a nonzero level voltage in accordance with the AMI rule. There are two cases, as shown in Figure 4.19.

Figure 4.19 *Two cases of B8ZS scrambling technique*



Note that the scrambling in this case does not change the bit rate. Also, the technique balances the positive and negative voltage levels (two positives and two negatives), which means that the DC balance is maintained. Note that the substitution may change the polarity of a 1 because, after the substitution, AMI needs to follow its rules.

B8ZS substitutes eight consecutive zeros with 000VB0VB.

One more point is worth mentioning. The letter V (violation) or B (bipolar) here is relative. The V means the same polarity as the polarity of the previous nonzero pulse; B means the polarity opposite to the polarity of the previous nonzero pulse.

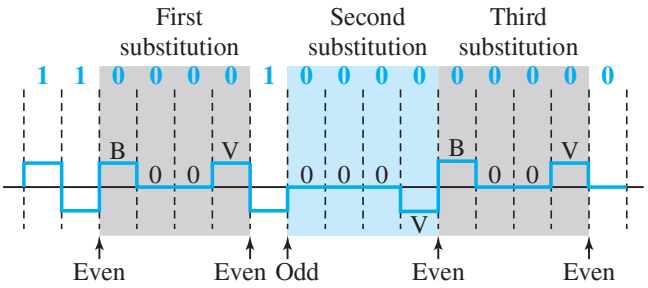
HDB3

High-density bipolar 3-zero (HDB3) is commonly used outside of North America. In this technique, which is more conservative than B8ZS, four consecutive zero-level voltages are replaced with a sequence of **000V** or **B00V.** The reason for two different substitutions is to maintain the even number of nonzero pulses after each substitution. The two rules can be stated as follows:

1. If the number of nonzero pulses after the last substitution is odd, the substitution pattern will be **000V**, which makes the total number of nonzero pulses even.
2. If the number of nonzero pulses after the last substitution is even, the substitution pattern will be **B00V**, which makes the total number of nonzero pulses even.

Figure 4.20 shows an example.

Figure 4.20 *Different situations in HDB3 scrambling technique*



There are several points we need to mention here. First, before the first substitution, the number of nonzero pulses is even, so the first substitution is B00V. After this substitution, the polarity of the 1 bit is changed because the AMI scheme, after each substitution, must follow its own rule. After this bit, we need another substitution, which is 000V because we have only one nonzero pulse (odd) after the last substitution. The third substitution is B00V because there are no nonzero pulses after the second substitution (even).

HDB3 substitutes four consecutive zeros with 000V or B00V depending on the number of nonzero pulses after the last substitution.

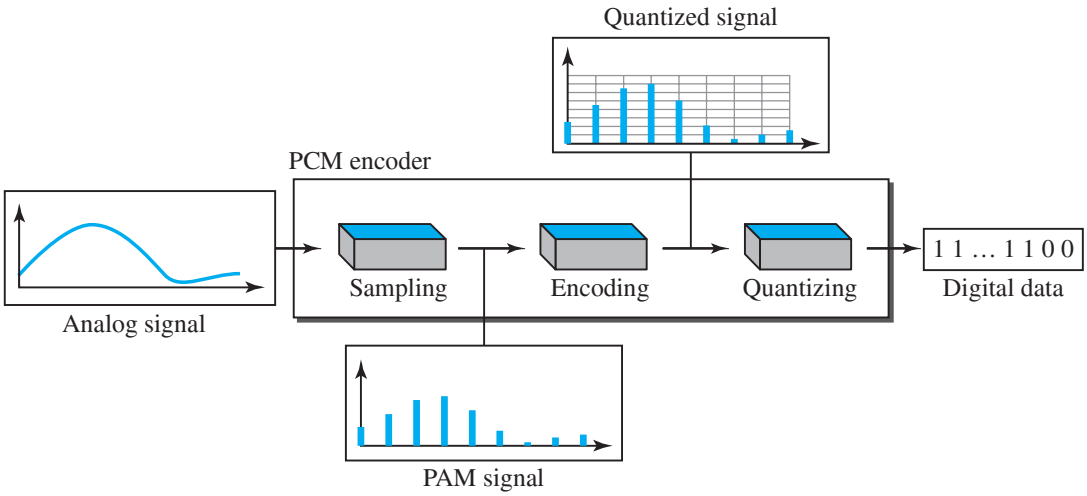
4.2 ANALOG-TO-DIGITAL CONVERSION

The techniques described in Section 4.1 convert digital data to digital signals. Sometimes, however, we have an analog signal such as one created by a microphone or camera. We have seen in Chapter 3 that a digital signal is superior to an analog signal. The tendency today is to change an analog signal to digital data. In this section we describe two techniques, pulse code modulation and delta modulation. After the digital data are created (digitization), we can use one of the techniques described in Section 4.1 to convert the digital data to a digital signal.

4.2.1 Pulse Code Modulation (PCM)

The most common technique to change an analog signal to digital data (**digitization**) is called **pulse code modulation (PCM).** A PCM encoder has three processes, as shown in Figure 4.21.

Figure 4.21 *Components of PCM encoder*



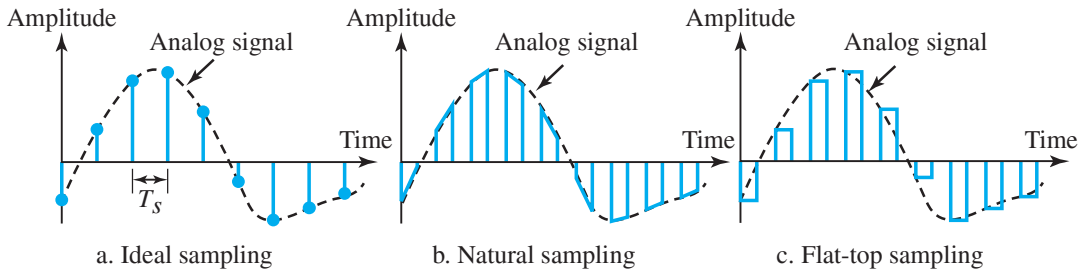
1. The analog signal is sampled.
2. The sampled signal is quantized.
3. The quantized values are encoded as streams of bits.

Sampling

The first step in PCM is **sampling.** The analog signal is sampled every T_s s, where T_s is the sample interval or period. The inverse of the sampling interval is called the ***sampling rate*** or ***sampling frequency*** and denoted by f_s, where $f_s = 1/T_s$. There are three sampling methods—ideal, natural, and flat-top—as shown in Figure 4.22.

In ideal sampling, pulses from the analog signal are sampled. This is an ideal sampling method and cannot be easily implemented. In natural sampling, a high-speed switch is turned on for only the small period of time when the sampling occurs. The result is a sequence of samples that retains the shape of the analog signal. The most

Figure 4.22 *Three different sampling methods for PCM*



common sampling method, called ***sample and hold,*** however, creates flat-top samples by using a circuit.

The sampling process is sometimes referred to as **pulse amplitude modulation (PAM).** We need to remember, however, that the result is still an analog signal with nonintegral values.

Sampling Rate

One important consideration is the sampling rate or frequency. What are the restrictions on T_s? This question was elegantly answered by Nyquist. According to the **Nyquist theorem,** to reproduce the original analog signal, one necessary condition is that the *sampling rate* be at least twice the highest frequency in the original signal.

According to the Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal.

We need to elaborate on the theorem at this point. First, we can sample a signal only if the signal is band-limited. In other words, a signal with an infinite bandwidth cannot be sampled. Second, the sampling rate must be at least 2 times the highest frequency, not the bandwidth. If the analog signal is low-pass, the bandwidth and the highest frequency are the same value. If the analog signal is bandpass, the bandwidth value is lower than the value of the maximum frequency. Figure 4.23 shows the value of the sampling rate for two types of signals.

Example 4.6

For an intuitive example of the Nyquist theorem, let us sample a simple sine wave at three sampling rates: $f_s = 4f$ (2 times the Nyquist rate), $f_s = 2f$ (Nyquist rate), and $f_s = f$ (one-half the Nyquist rate). Figure 4.24 shows the sampling and the subsequent recovery of the signal.

It can be seen that sampling at the Nyquist rate can create a good approximation of the original sine wave (part a). Oversampling in part b can also create the same approximation, but it is redundant and unnecessary. Sampling below the Nyquist rate (part c) does not produce a signal that looks like the original sine wave.

Example 4.7

As an interesting example, let us see what happens if we sample a periodic event such as the revolution of a hand of a clock. The second hand of a clock has a period of 60 s. According to the

Figure 4.23 *Nyquist sampling rate for low-pass and bandpass signals*

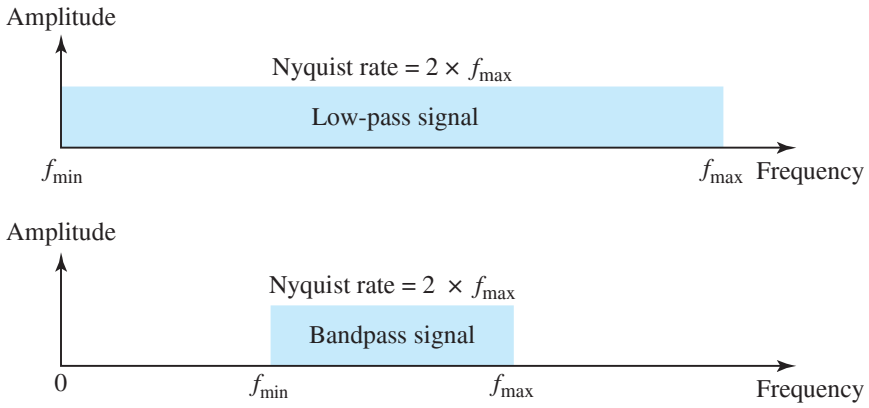
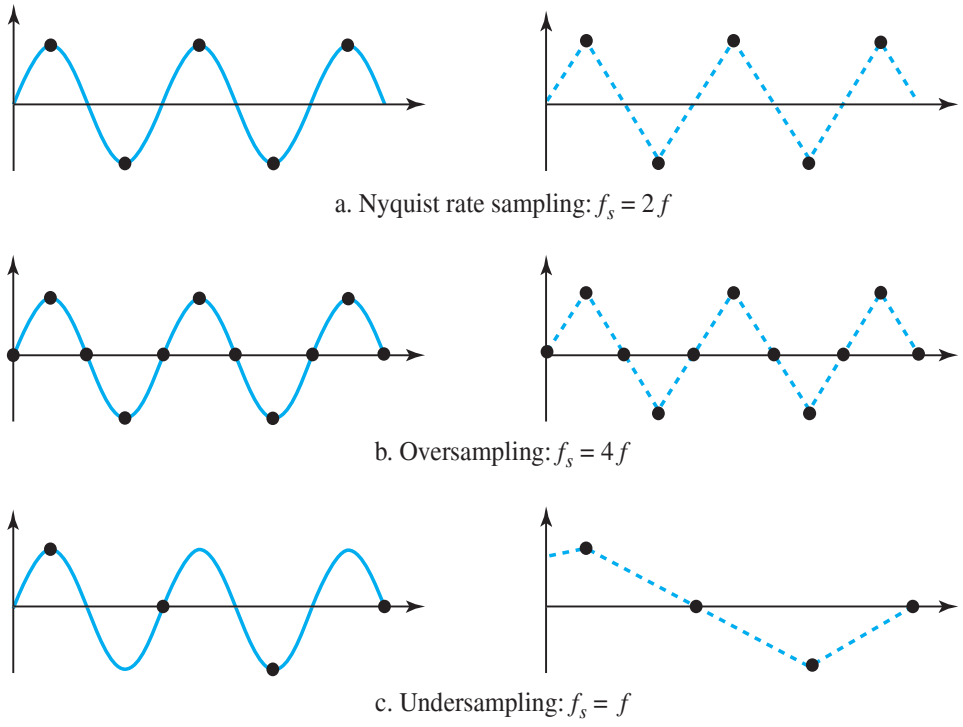
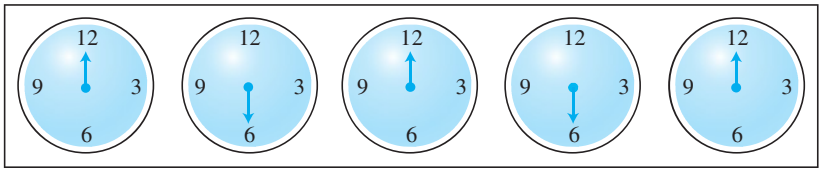


Figure 4.24 *Recovery of a sampled sine wave for different sampling rates*

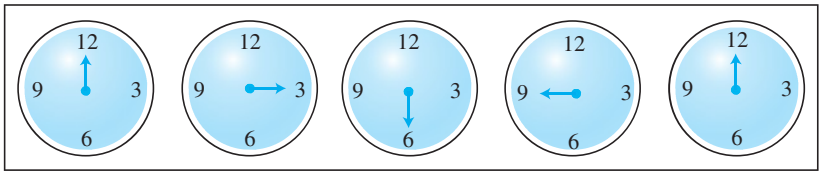


Nyquist theorem, we need to sample the hand (take and send a picture) every 30 s ($T_s = \frac{1}{2}T$ or $f_s = 2f$). In Figure 4.25a, the sample points, in order, are 12, 6, 12, 6, 12, and 6. The receiver of the samples cannot tell if the clock is moving forward or backward. In part b, we sample at double the Nyquist rate (every 15 s). The sample points, in order, are 12, 3, 6, 9, and 12. The clock is moving forward. In part c, we sample below the Nyquist rate ($T_s = \frac{3}{4}T$ or $f_s = \frac{4}{3}f$). The sample

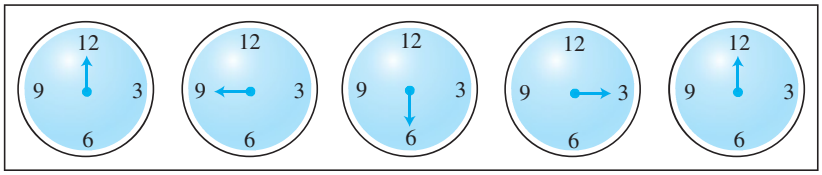
Figure 4.25 *Sampling of a clock with only one hand*



a. Sampling at Nyquist rate: $T_s = T\,\frac{1}{2}$



b. Oversampling (above Nyquist rate): $T_s = \quad T\,\frac{1}{4}$



c. Undersampling (below Nyquist rate): $T_s = T\,\frac{3}{4}$

points, in order, are 12, 9, 6, 3, and 12. Although the clock is moving forward, the receiver thinks that the clock is moving backward.

Example 4.8

An example related to Example 4.7 is the seemingly backward rotation of the wheels of a forward-moving car in a movie. This can be explained by undersampling. A movie is filmed at 24 frames per second. If a wheel is rotating more than 12 times per second, the undersampling creates the impression of a backward rotation.

Example 4.9

Telephone companies digitize voice by assuming a maximum frequency of 4000 Hz. The sampling rate therefore is 8000 samples per second.

Example 4.10

A complex low-pass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

The bandwidth of a low-pass signal is between 0 and f, where f is the maximum frequency in the signal. Therefore, we can sample this signal at 2 times the highest frequency (200 kHz). The sampling rate is therefore 400,000 samples per second.

Example 4.11

A complex bandpass signal has a bandwidth of 200 kHz. What is the minimum sampling rate for this signal?

Solution

We cannot find the minimum sampling rate in this case because we do not know where the bandwidth starts or ends. We do not know the maximum frequency in the signal.

Quantization

The result of sampling is a series of pulses with amplitude values between the maximum and minimum amplitudes of the signal. The set of amplitudes can be infinite with nonintegral values between the two limits. These values cannot be used in the encoding process. The following are the steps in quantization:

1. We assume that the original analog signal has instantaneous amplitudes between V_{min} and V_{max}.
2. We divide the range into L zones, each of height Δ (delta).

$$\Delta = \frac{V_{max} - V_{min}}{L}$$

3. We assign quantized values of 0 to $L - 1$ to the midpoint of each zone.
4. We approximate the value of the sample amplitude to the quantized values.

As a simple example, assume that we have a sampled signal and the sample amplitudes are between −20 and +20 V. We decide to have eight levels ($L = 8$). This means that $\Delta = 5$ V. Figure 4.26 shows this example.

We have shown only nine samples using ideal sampling (for simplicity). The value at the top of each sample in the graph shows the actual amplitude. In the chart, the first row is the normalized value for each sample (actual amplitude/Δ). The quantization process selects the quantization value from the middle of each zone. This means that the normalized quantized values (second row) are different from the normalized amplitudes. The difference is called the *normalized error* (third row). The fourth row is the quantization code for each sample based on the quantization levels at the left of the graph. The encoded words (fifth row) are the final products of the conversion.

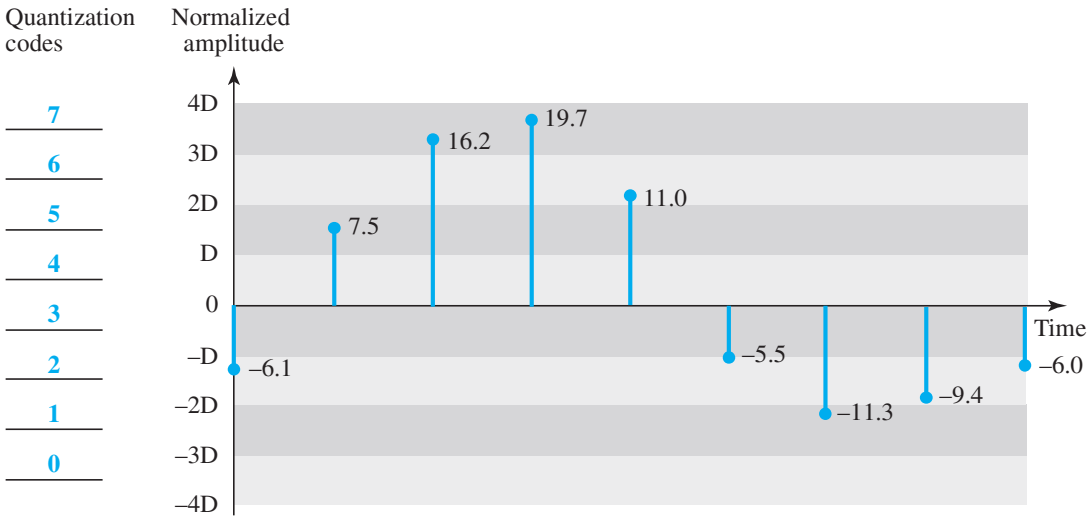
Quantization Levels

In the previous example, we showed eight quantization levels. The choice of L, the number of levels, depends on the range of the amplitudes of the analog signal and how accurately we need to recover the signal. If the amplitude of a signal fluctuates between two values only, we need only two levels; if the signal, like voice, has many amplitude values, we need more quantization levels. In audio digitizing, L is normally chosen to be 256; in video it is normally thousands. Choosing lower values of L increases the quantization error if there is a lot of fluctuation in the signal.

Quantization Error

One important issue is the error created in the quantization process. (Later, we will see how this affects high-speed modems.) Quantization is an approximation process. The

Figure 4.26 *Quantization and encoding of a sampled signal*



Normalized PAM values	–1.22	1.50	3.24	3.94	2.20	–1.10	–2.26	–1.88	–1.20
Normalized quantized values	–1.50	1.50	3.50	3.50	2.50	–1.50	–2.50	–1.50	–1.50
Normalized error	–0.38	0	+0.26	–0.44	+0.30	–0.40	–0.24	+0.38	–0.30
Quantization code	2	5	7	7	6	2	1	2	2
Encoded words	010	101	111	111	110	010	001	010	010

input values to the quantizer are the real values; the output values are the approximated values. The output values are chosen to be the middle value in the zone. If the input value is also at the middle of the zone, there is no quantization error; otherwise, there is an error. In the previous example, the normalized amplitude of the third sample is 3.24, but the normalized quantized value is 3.50. This means that there is an error of +0.26. The value of the error for any sample is less than $\Delta/2$. In other words, we have $-\Delta/2 \leq \text{error} \leq \Delta/2$.

The quantization error changes the signal-to-noise ratio of the signal, which in turn reduces the upper limit capacity according to Shannon.

It can be proven that the contribution of the **quantization error** to the SNR_{dB} of the signal depends on the number of quantization levels L, or the bits per sample n_b, as shown in the following formula:

$$\mathbf{SNR_{dB} = 6.02}n_b \mathbf{+ 1.76 \quad dB}$$

Example 4.12

What is the SNR_{dB} in the example of Figure 4.26?

Solution

We can use the formula to find the quantization. We have eight levels and 3 bits per sample, so $SNR_{dB} = 6.02(3) + 1.76 = 19.82$ dB. Increasing the number of levels increases the SNR.

Example 4.13

A telephone subscriber line must have an SNR_{dB} above 40. What is the minimum number of bits per sample?

Solution

We can calculate the number of bits as

$$\mathbf{SNR_{dB} = 6.02}n_b \mathbf{+ 1.76 = 40} \rightarrow n = 6.35$$

Telephone companies usually assign 7 or 8 bits per sample.

Uniform Versus Nonuniform Quantization

For many applications, the distribution of the instantaneous amplitudes in the analog signal is not uniform. Changes in amplitude often occur more frequently in the lower amplitudes than in the higher ones. For these types of applications it is better to use nonuniform zones. In other words, the height of Δ is not fixed; it is greater near the lower amplitudes and less near the higher amplitudes. Nonuniform quantization can also be achieved by using a process called **companding and expanding.** The signal is companded at the sender before conversion; it is expanded at the receiver after conversion. *Companding* means reducing the instantaneous voltage amplitude for large values; expanding is the opposite process. Companding gives greater weight to strong signals and less weight to weak ones. It has been proved that nonuniform quantization effectively reduces the SNR_{dB} of quantization.

Encoding

The last step in PCM is encoding. After each sample is quantized and the number of bits per sample is decided, each sample can be changed to an n_b-bit code word. In Figure 4.26 the encoded words are shown in the last row. A quantization code of 2 is encoded as 010; 5 is encoded as 101; and so on. Note that the number of bits for each sample is determined from the number of quantization levels. If the number of quantization levels is L, the number of bits is $n_b = \log_2 L$. In our example L is 8 and n_b is therefore 3. The bit rate can be found from the formula

$$\textbf{Bit rate} = \textbf{sampling rate} \times \textbf{number of bits per sample} = f_s \times n_b$$

Example 4.14

We want to digitize the human voice. What is the bit rate, assuming 8 bits per sample?

Solution

The human voice normally contains frequencies from 0 to 4000 Hz. So the sampling rate and bit rate are calculated as follows:

$$\textbf{Sampling rate} = 4000 \times 2 = \textbf{8000 samples/s}$$

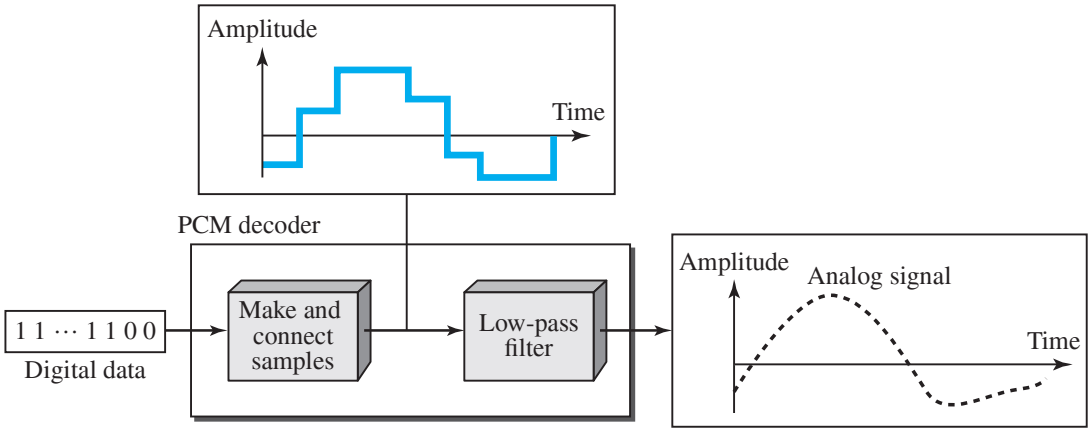
$$\textbf{Bit rate} = 8000 \times 8 = \textbf{64,000 bps} = \textbf{64 kbps}$$

Original Signal Recovery

The recovery of the original signal requires the PCM decoder. The decoder first uses circuitry to convert the code words into a pulse that holds the amplitude until the next

pulse. After the staircase signal is completed, it is passed through a low-pass filter to smooth the staircase signal into an analog signal. The filter has the same cutoff frequency as the original signal at the sender. If the signal has been sampled at (or greater than) the Nyquist sampling rate and if there are enough quantization levels, the original signal will be recreated. Note that the maximum and minimum values of the original signal can be achieved by using amplification. Figure 4.27 shows the simplified process.

Figure 4.27 *Components of a PCM decoder*



PCM Bandwidth

Suppose we are given the bandwidth of a low-pass analog signal. If we then digitize the signal, what is the new minimum bandwidth of the channel that can pass this digitized signal? We have said that the minimum bandwidth of a line-encoded signal is $B_{min} = c \times N \times (1/r)$. We substitute the value of N in this formula:

$$\boldsymbol{B_{\text{min}} = c \times N \times \frac{1}{r} = c \times n_b \times f_s \times \frac{1}{r} = c \times n_b \times 2 \times B_{\text{analog}} \times \frac{1}{r}}$$

When $1/r = 1$ (for a NRZ or bipolar signal) and $c = (1/2)$ (the average situation), the minimum bandwidth is

$$\boldsymbol{B_{\text{min}} = n_b \times B_{\text{analog}}}$$

This means the minimum bandwidth of the digital signal is n_b times greater than the bandwidth of the analog signal. This is the price we pay for digitization.

Example 4.15

We have a low-pass analog signal of 4 kHz. If we send the analog signal, we need a channel with a minimum bandwidth of 4 kHz. If we digitize the signal and send 8 bits per sample, we need a channel with a minimum bandwidth of 8×4 kHz = 32 kHz.

Maximum Data Rate of a Channel

In Chapter 3, we discussed the Nyquist theorem, which gives the data rate of a channel as $N_{max} = 2 \times B \times \log_2 L$. We can deduce this rate from the Nyquist sampling theorem by using the following arguments.

1. We assume that the available channel is low-pass with bandwidth B.
2. We assume that the digital signal we want to send has L levels, where each level is a signal element. This means $r = 1/\log_2 L$.
3. We first pass the digital signal through a low-pass filter to cut off the frequencies above B Hz.
4. We treat the resulting signal as an analog signal and sample it at $2 \times B$ samples per second and quantize it using L levels. Additional quantization levels are useless because the signal originally had L levels.
5. The resulting bit rate is $N = f_s \times n_b = 2 \times B \times \log_2 L$. This is the maximum bandwidth.

$$N_{max} = 2 \times B \times \log_2 L \quad \text{bps}$$

Minimum Required Bandwidth

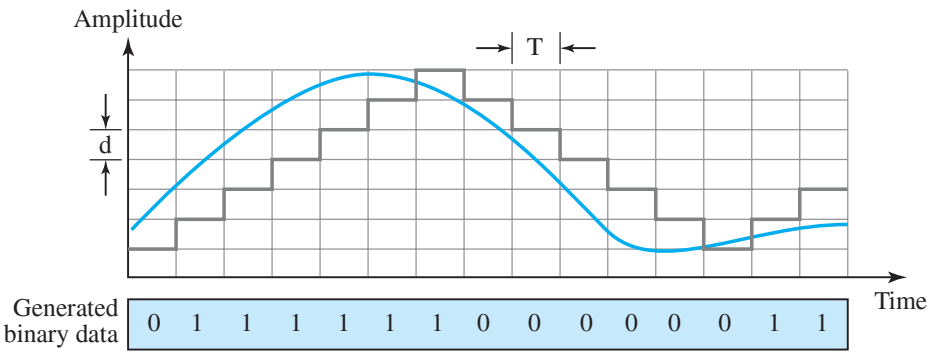
The previous argument can give us the minimum bandwidth if the data rate and the number of signal levels are fixed. We can say

$$B_{min} = \frac{N}{(2 \times \log_2) L} \quad \text{Hz}$$

4.2.2 Delta Modulation (DM)

PCM is a very complex technique. Other techniques have been developed to reduce the complexity of PCM. The simplest is **delta modulation.** PCM finds the value of the signal amplitude for each sample; DM finds the change from the previous sample. Figure 4.28 shows the process. Note that there are no code words here; bits are sent one after another.

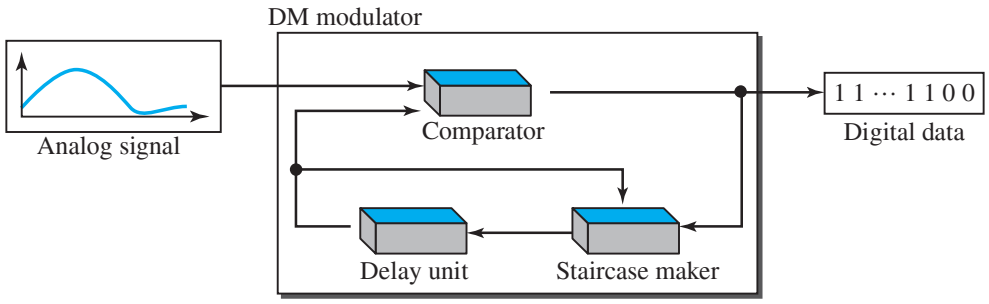
Figure 4.28 *The process of delta modulation*



Modulator

The modulator is used at the sender site to create a stream of bits from an analog signal. The process records the small positive or negative changes, called delta δ. If the delta is positive, the process records a 1; if it is negative, the process records a 0. However, the process needs a base against which the analog signal is compared. The modulator builds a second signal that resembles a staircase. Finding the change is then reduced to comparing the input signal with the gradually made staircase signal. Figure 4.29 shows a diagram of the process.

Figure 4.29 *Delta modulation components*



The modulator, at each sampling interval, compares the value of the analog signal with the last value of the staircase signal. If the amplitude of the analog signal is larger, the next bit in the digital data is 1; otherwise, it is 0. The output of the comparator, however, also makes the staircase itself. If the next bit is 1, the staircase maker moves the last point of the staircase signal δ up; if the next bit is 0, it moves it δ down. Note that we need a delay unit to hold the staircase function for a period between two comparisons.

Demodulator

The demodulator takes the digital data and, using the staircase maker and the delay unit, creates the analog signal. The created analog signal, however, needs to pass through a low-pass filter for smoothing. Figure 4.30 shows the schematic diagram.

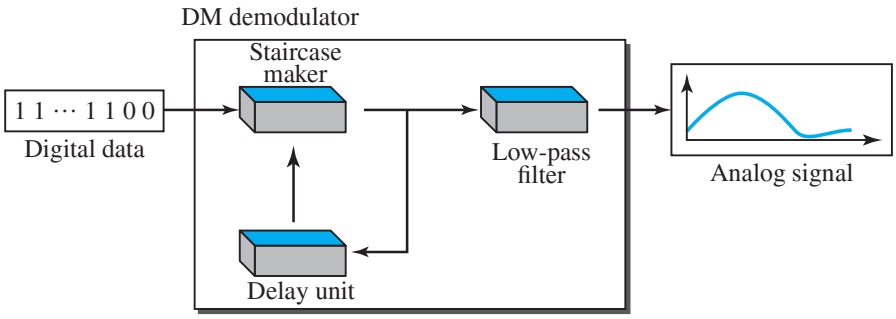
Adaptive DM

A better performance can be achieved if the value of δ is not fixed. In **adaptive delta modulation,** the value of δ changes according to the amplitude of the analog signal.

Quantization Error

It is obvious that DM is not perfect. Quantization error is always introduced in the process. The quantization error of DM, however, is much less than that for PCM.

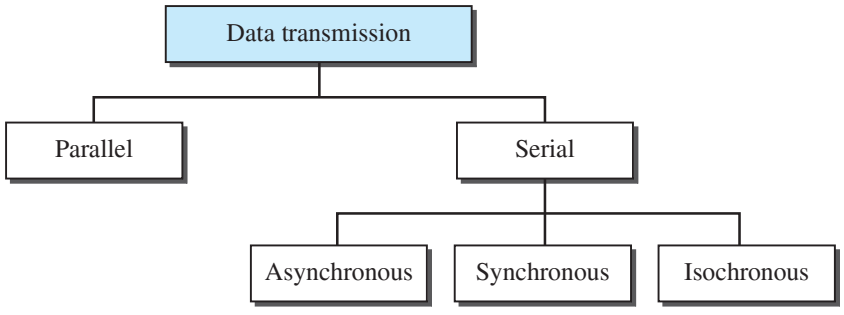
Figure 4.30 *Delta demodulation components*



4.3 TRANSMISSION MODES

Of primary concern when we are considering the transmission of data from one device to another is the wiring, and of primary concern when we are considering the wiring is the data stream. Do we send 1 bit at a time; or do we group bits into larger groups and, if so, how? The transmission of binary data across a link can be accomplished in either parallel or serial mode. In parallel mode, multiple bits are sent with each clock tick. In serial mode, 1 bit is sent with each clock tick. While there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous (see Figure 4.31).

Figure 4.31 *Data transmission and modes*



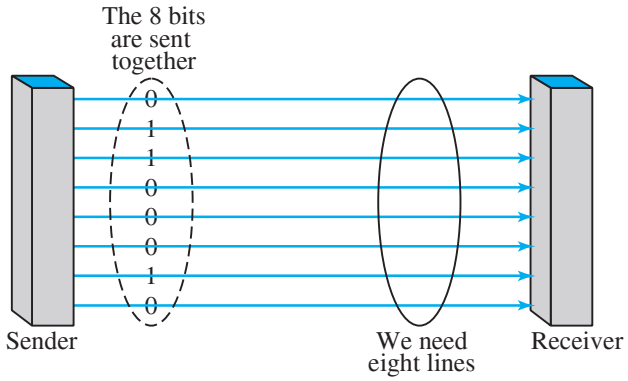
4.3.1 Parallel Transmission

Binary data, consisting of 1s and 0s, may be organized into groups of n bits each. Computers produce and consume data in groups of bits much as we conceive of and use spoken language in the form of words rather than letters. By grouping, we can send data n bits at a time instead of 1. This is called ***parallel transmission.***

The mechanism for parallel transmission is a conceptually simple one: Use n wires to send n bits at one time. That way each bit has its own wire, and all n bits of one

group can be transmitted with each clock tick from one device to another. Figure 4.32 shows how parallel transmission works for $n = 8$. Typically, the eight wires are bundled in a cable with a connector at each end.

Figure 4.32 *Parallel transmission*

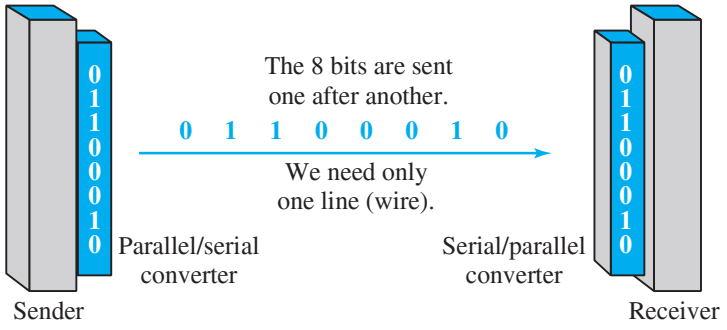


The advantage of parallel transmission is speed. All else being equal, parallel transmission can increase the transfer speed by a factor of n over serial transmission. But there is a significant disadvantage: cost. Parallel transmission requires n communication lines (wires in the example) just to transmit the data stream. Because this is expensive, parallel transmission is usually limited to short distances.

4.3.2 Serial Transmission

In **serial transmission** one bit follows another, so we need only one communication channel rather than n to transmit data between two communicating devices (see Figure 4.33).

Figure 4.33 *Serial transmission*



The advantage of serial over parallel transmission is that with only one communication channel, serial transmission reduces the cost of transmission over parallel by roughly a factor of n.

Since communication within devices is parallel, conversion devices are required at the interface between the sender and the line (parallel-to-serial) and between the line and the receiver (serial-to-parallel).

Serial transmission occurs in one of three ways: asynchronous, synchronous, and isochronous.

Asynchronous Transmission

Asynchronous transmission is so named because the timing of a signal is unimportant. Instead, information is received and translated by agreed upon patterns. As long as those patterns are followed, the receiving device can retrieve the information without regard to the rhythm in which it is sent. Patterns are based on grouping the bit stream into bytes. Each group, usually 8 bits, is sent along the link as a unit. The sending system handles each group independently, relaying it to the link whenever ready, without regard to a timer.

Without synchronization, the receiver cannot use timing to predict when the next group will arrive. To alert the receiver to the arrival of a new group, therefore, an extra bit is added to the beginning of each byte. This bit, usually a 0, is called the **start bit.** To let the receiver know that the byte is finished, 1 or more additional bits are appended to the end of the byte. These bits, usually 1s, are called **stop bits.** By this method, each byte is increased in size to at least 10 bits, of which 8 bits is information and 2 bits or more are signals to the receiver. In addition, the transmission of each byte may then be followed by a gap of varying duration. This gap can be represented either by an idle channel or by a stream of additional stop bits.

In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (1s) at the end of each byte. There may be a gap between bytes.

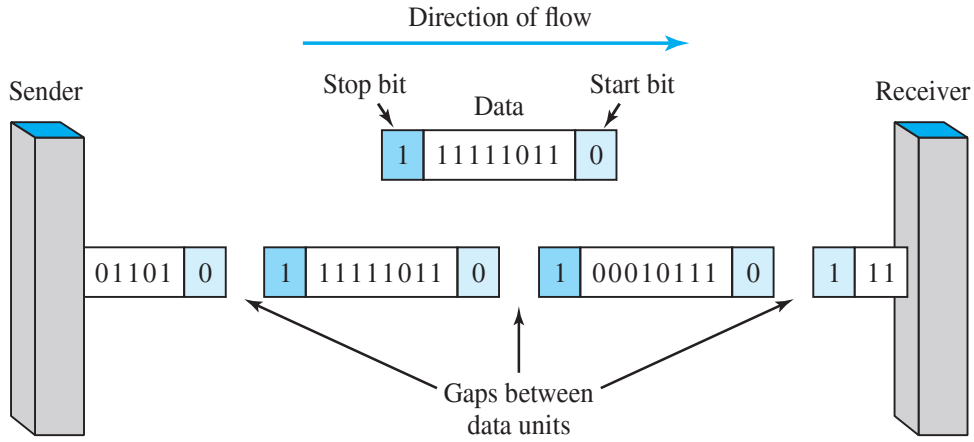
The start and stop bits and the gap alert the receiver to the beginning and end of each byte and allow it to synchronize with the data stream. This mechanism is called *asynchronous* because, at the byte level, the sender and receiver do not have to be synchronized. But within each byte, the receiver must still be synchronized with the incoming bit stream. That is, some synchronization is required, but only for the duration of a single byte. The receiving device resynchronizes at the onset of each new byte. When the receiver detects a start bit, it sets a timer and begins counting bits as they come in. After *n* bits, the receiver looks for a stop bit. As soon as it detects the stop bit, it waits until it detects the next start bit.

Asynchronous here means "asynchronous at the byte level," but the bits are still synchronized; their durations are the same.

Figure 4.34 is a schematic illustration of asynchronous transmission. In this example, the start bits are 0s, the stop bits are 1s, and the gap is represented by an idle line rather than by additional stop bits.

The addition of stop and start bits and the insertion of gaps into the bit stream make asynchronous transmission slower than forms of transmission that can operate

Figure 4.34 *Asynchronous transmission*



without the addition of control information. But it is cheap and effective, two advantages that make it an attractive choice for situations such as low-speed communication. For example, the connection of a keyboard to a computer is a natural application for asynchronous transmission. A user types only one character at a time, types extremely slowly in data processing terms, and leaves unpredictable gaps of time between characters.

Synchronous Transmission

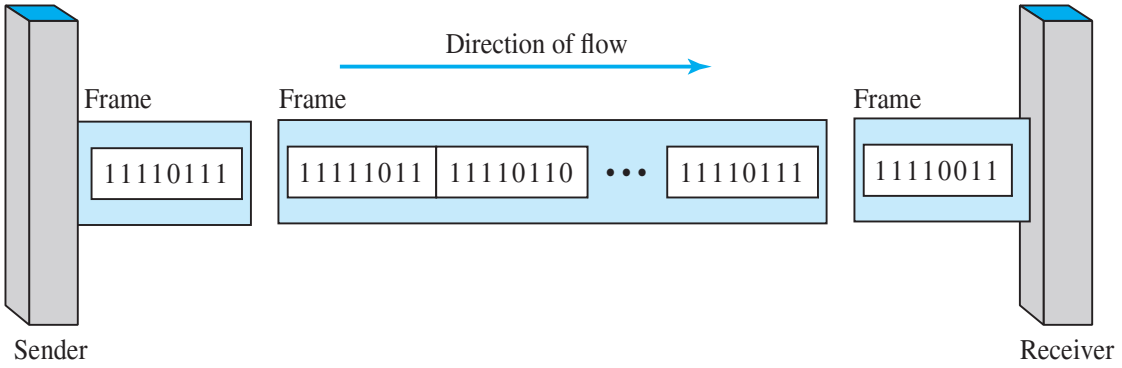
In **synchronous transmission,** the bit stream is combined into longer "frames," which may contain multiple bytes. Each byte, however, is introduced onto the transmission link without a gap between it and the next one. It is left to the receiver to separate the bit stream into bytes for decoding purposes. In other words, data are transmitted as an unbroken string of 1s and 0s, and the receiver separates that string into the bytes, or characters, it needs to reconstruct the information.

In synchronous transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.

Figure 4.35 gives a schematic illustration of synchronous transmission. We have drawn in the divisions between bytes. In reality, those divisions do not exist; the sender puts its data onto the line as one long string. If the sender wishes to send data in separate bursts, the gaps between bursts must be filled with a special sequence of 0s and 1s that means *idle*. The receiver counts the bits as they arrive and groups them in 8-bit units.

Without gaps and start and stop bits, there is no built-in mechanism to help the receiving device adjust its bit synchronization midstream. Timing becomes very important, therefore, because the accuracy of the received information is completely dependent on the ability of the receiving device to keep an accurate count of the bits as they come in.

Figure 4.35 *Synchronous transmission*



The advantage of synchronous transmission is speed. With no extra bits or gaps to introduce at the sending end and remove at the receiving end, and, by extension, with fewer bits to move across the link, synchronous transmission is faster than asynchronous transmission. For this reason, it is more useful for high-speed applications such as the transmission of data from one computer to another. Byte synchronization is accomplished in the data-link layer.

We need to emphasize one point here. Although there is no gap between characters in synchronous serial transmission, there may be uneven gaps between frames.

Isochronous

In real-time audio and video, in which uneven delays between frames are not acceptable, synchronous transmission fails. For example, TV images are broadcast at the rate of 30 images per second; they must be viewed at the same rate. If each image is sent by using one or more frames, there should be no delays between frames. For this type of application, synchronization between characters is not enough; the entire stream of bits must be synchronized. The **isochronous transmission** guarantees that the data arrive at a fixed rate.

4.4 END-CHAPTER MATERIALS

4.4.1 Recommended Reading

For more details about subjects discussed in this chapter, we recommend the following books. The items in brackets […] refer to the reference list at the end of the text.

Books

Digital to digital conversion is discussed in [Pea92], [Cou01], and [Sta04]. Sampling is discussed in [Pea92], [Cou01], and [Sta04]. [Hsu03] gives a good mathematical approach to modulation and sampling. More advanced materials can be found in [Ber96].

4.4.2 Key Terms

adaptive delta modulation
alternate mark inversion (AMI)
analog-to-digital conversion
asynchronous transmission
baseline
baseline wandering
baud rate
biphase
bipolar
bipolar with 8-zero substitution (B8ZS)
bit rate
block coding
companding and expanding
data element
data rate
DC component
delta modulation (DM)
differential Manchester
digital-to-digital conversion
digitization
eight binary/ten binary (8B/10B)
eight-binary, six-ternary (8B6T)
four binary/five binary (4B/5B)
four dimensional, five-level pulse amplitude modulation (4D-PAM5)
high-density bipolar 3-zero (HDB3)
isochronous transmission
line coding
Manchester
modulation rate
multilevel binary
multiline transmission, three-level (MLT-3)
non-return-to-zero (NRZ)
non-return-to-zero, invert (NRZ-I)
non-return-to-zero, level (NRZ-L)
Nyquist theorem
parallel transmission
polar
pseudoternary
pulse amplitude modulation (PAM)
pulse code modulation (PCM)
pulse rate
quantization
quantization error
return-to-zero (RZ)
sample and hold
sampling
sampling rate
scrambling
self-synchronizing
serial transmission
signal element
signal rate
start bit
stop bit
synchronous transmission
transmission mode
two-binary, one quaternary (2B1Q)
unipolar

4.4.3 Summary

Digital-to-digital conversion involves three techniques: line coding, block coding, and scrambling. Line coding is the process of converting digital data to a digital signal. We can roughly divide line coding schemes into five broad categories: unipolar, polar, bipolar, multilevel, and multitransition. Block coding provides redundancy to ensure synchronization and inherent error detection. Block coding is normally referred to as mB/nB coding; it replaces each m-bit group with an n-bit group. Scrambling provides synchronization without increasing the number of bits. Two common scrambling techniques are B8ZS and HDB3.

The most common technique to change an analog signal to digital data (digitization) is called pulse code modulation (PCM). The first step in PCM is sampling. The analog signal is sampled every T_s second, where T_s is the sample interval or period. The inverse of the sampling interval is called the *sampling rate* or *sampling frequency* and denoted by f_s, where $f_s = 1/T_s$. There are three sampling methods—ideal, natural, and flat-top. According to the *Nyquist theorem,* to reproduce the original analog signal, one necessary condition is that the *sampling rate* be at least twice the highest frequency in

the original signal. Other sampling techniques have been developed to reduce the complexity of PCM. The simplest is delta modulation. PCM finds the value of the signal amplitude for each sample; DM finds the change from the previous sample.

While there is only one way to send parallel data, there are three subclasses of serial transmission: asynchronous, synchronous, and isochronous. In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (1s) at the end of each byte. In synchronous transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits. The isochronous mode provides synchronization for the entire stream of bits. In other words, it guarantees that the data arrive at a fixed rate.

4.5 PRACTICE SET

4.5.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

4.5.2 Questions

Q4-1. List three techniques of digital-to-digital conversion.

Q4-2. Distinguish between a signal element and a data element.

Q4-3. Distinguish between data rate and signal rate.

Q4-4. Define baseline wandering and its effect on digital transmission.

Q4-5. Define a DC component and its effect on digital transmission.

Q4-6. Define the characteristics of a self-synchronizing signal.

Q4-7. List five line coding schemes discussed in this book.

Q4-8. Define block coding and give its purpose.

Q4-9. Define scrambling and give its purpose.

Q4-10. Compare and contrast PCM and DM.

Q4-11. What are the differences between parallel and serial transmission?

Q4-12. List three different techniques in serial transmission and explain the differences.

4.5.3 Problems

P4-1. Calculate the value of the signal rate for each case in Figure 4.2 if the data rate is 1 Mbps and c = 1/2.

P4-2. In a digital transmission, the sender clock is 0.2 percent faster than the receiver clock. How many extra bits per second does the sender send if the data rate is 1 Mbps?

P4-3. Draw the graph of the NRZ-L scheme using each of the following data streams, assuming that the last signal level has been positive. From the graphs, guess the bandwidth for this scheme using the average number of changes

in the signal level. Compare your guess with the corresponding entry in Table 4.1.

a. 00000000 **b.** 11111111 **c.** 01010101 **d.** 00110011

P4-4. Repeat Problem P4-3 for the NRZ-I scheme.

P4-5. Repeat Problem P4-3 for the Manchester scheme.

P4-6. Repeat Problem P4-3 for the differential Manchester scheme.

P4-7. Repeat Problem P4-3 for the 2B1Q scheme, but use the following data streams.

a. 0000000000000000

b. 1111111111111111

c. 0101010101010101

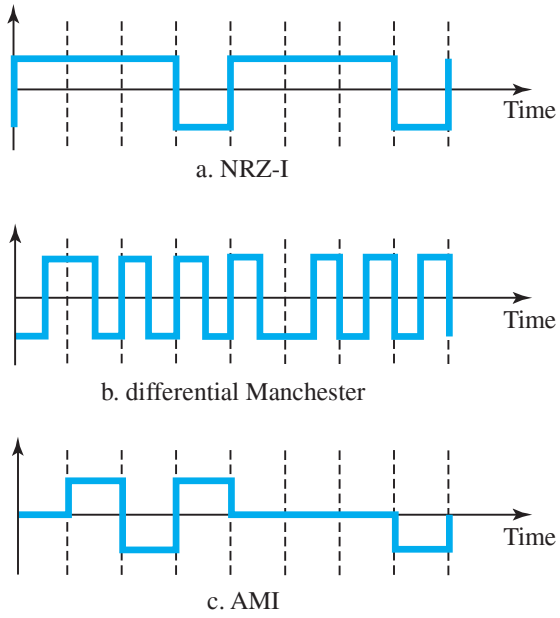
d. 0011001100110011

P4-8. Repeat Problem P4-3 for the MLT-3 scheme, but use the following data streams.

a. 00000000 **b.** 11111111 **c.** 01010101 **d.** 00011000

P4-9. Find the 8-bit data stream for each case depicted in Figure 4.36.

Figure 4.36 *Problem P4-9*



P4-10. An NRZ-I signal has a data rate of 100 Kbps. Using Figure 4.6, calculate the value of the normalized energy (P) for frequencies at 0 Hz, 50 KHz, and 100 KHz.

P4-11. A Manchester signal has a data rate of 100 Kbps. Using Figure 4.8, calculate the value of the normalized energy (P) for frequencies at 0 Hz, 50 KHz, 100 KHz.

P4-12. The input stream to a 4B/5B block encoder is

0100 0000 0000 0000 0000 0001

Answer the following questions:

a. What is the output stream?

b. What is the length of the longest consecutive sequence of 0s in the input?

c. What is the length of the longest consecutive sequence of 0s in the output?

P4-13. How many invalid (unused) code sequences can we have in 5B/6B encoding? How many in 3B/4B encoding?

P4-14. What is the result of scrambling the sequence 11100000000000 using each of the following scrambling techniques? Assume that the last non-zero signal level has been positive.

a. B8ZS

b. HDB3 (The number of nonzero pulses is odd after the last substitution.)

P4-15. What is the Nyquist sampling rate for each of the following signals?

a. A low-pass signal with bandwidth of 200 KHz?

b. A band-pass signal with bandwidth of 200 KHz if the lowest frequency is 100 KHz?

P4-16. We have sampled a low-pass signal with a bandwidth of 200 KHz using 1024 levels of quantization.

a. Calculate the bit rate of the digitized signal.

b. Calculate the SNR_{dB} for this signal.

c. Calculate the PCM bandwidth of this signal.

P4-17. What is the maximum data rate of a channel with a bandwidth of 200 KHz if we use four levels of digital signaling.

P4-18. An analog signal has a bandwidth of 20 KHz. If we sample this signal and send it through a 30 Kbps channel, what is the SNR_{dB}?

P4-19. We have a baseband channel with a 1-MHz bandwidth. What is the data rate for this channel if we use each of the following line coding schemes?

a. NRZ-L **b.** Manchester **c.** MLT-3 **d.** 2B1Q

P4-20. We want to transmit 1000 characters with each character encoded as 8 bits.

a. Find the number of transmitted bits for synchronous transmission.

b. Find the number of transmitted bits for asynchronous transmission.

c. Find the redundancy percent in each case.

4.6 SIMULATION EXPERIMENTS

4.6.1 Applets

We have created some Java applets to show some of the main concepts discussed in this chapter. It is strongly recommended that the students activate these applets on the book website and carefully examine the protocols in action.

CHAPTER 5

Analog Transmission

In Chapter 3, we discussed the advantages and disadvantages of digital and analog transmission. We saw that while digital transmission is very desirable, a low-pass channel is needed. We also saw that analog transmission is the only choice if we have a bandpass channel. Digital transmission was discussed in Chapter 4; we discuss analog transmission in this chapter.

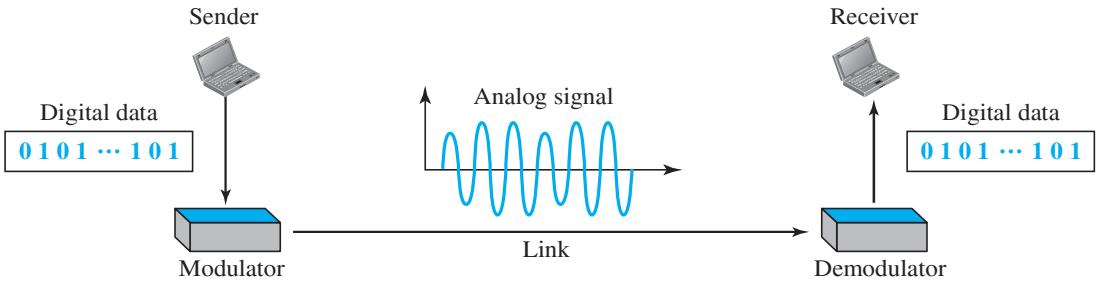
Converting digital data to a bandpass analog signal is traditionally called digital-to-analog conversion. Converting a low-pass analog signal to a bandpass analog signal is traditionally called analog-to-analog conversion. In this chapter, we discuss these two types of conversions in two sections:

- The first section discusses digital-to-analog conversion. The section shows how we can change digital data to an analog signal when a band-pass channel is available. The first method described is called amplitude shift keying (ASK), in which the amplitude of a carrier is changed using the digital data. The second method described is called frequency shift keying (FSK), in which the frequency of a carrier is changed using the digital data. The third method described is called phase shift keying (PSK), in which the phase of a carrier signal is changed to represent digital data. The fourth method described is called quadrature amplitude modulation (QAM), in which both amplitude and phase of a carrier signal are changed to represent digital data.
- The second section discusses analog-to-analog conversion. The section shows how we can change an analog signal to a new analog signal with a smaller bandwidth. The conversion is used when only a band-pass channel is available. The first method is called amplitude modulation (AM), in which the amplitude of a carrier is changed based on the changes in the original analog signal. The second method is called frequency modulation (FM), in which the phase of a carrier is changed based on the changes in the original analog signal. The third method is called phase modulation (PM), in which the phase of a carrier signal is changed to show the changes in the original signal.

5.1 DIGITAL-TO-ANALOG CONVERSION

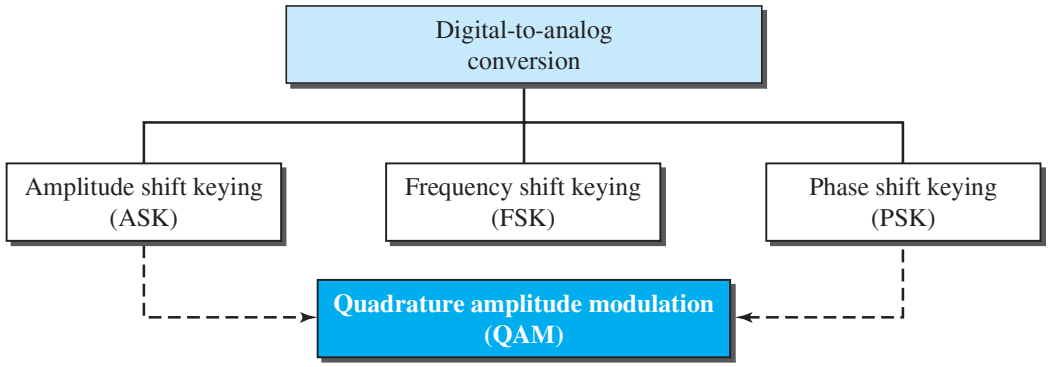
Digital-to-analog conversion is the process of changing one of the characteristics of an analog signal based on the information in digital data. Figure 5.1 shows the relationship between the digital information, the digital-to-analog modulating process, and the resultant analog signal.

Figure 5.1 *Digital-to-analog conversion*



As discussed in Chapter 3, a sine wave is defined by three characteristics: amplitude, frequency, and phase. When we vary any one of these characteristics, we create a different version of that wave. So, by changing one characteristic of a simple electric signal, we can use it to represent digital data. Any of the three characteristics can be altered in this way, giving us at least three mechanisms for modulating digital data into an analog signal: **amplitude shift keying (ASK), frequency shift keying (FSK),** and **phase shift keying (PSK).** In addition, there is a fourth (and better) mechanism that combines changing both the amplitude and phase, called **quadrature amplitude modulation (QAM).** QAM is the most efficient of these options and is the mechanism commonly used today (see Figure 5.2).

Figure 5.2 *Types of digital-to-analog conversion*



5.1.1 Aspects of Digital-to-Analog Conversion

Before we discuss specific methods of digital-to-analog modulation, two basic issues must be reviewed: bit and baud rates and the carrier signal.

Data Element Versus Signal Element

In Chapter 4, we discussed the concept of the data element versus the signal element. We defined a data element as the smallest piece of information to be exchanged, the bit. We also defined a signal element as the smallest unit of a signal that is constant. Although we continue to use the same terms in this chapter, we will see that the nature of the signal element is a little bit different in analog transmission.

Data Rate Versus Signal Rate

We can define the data rate (bit rate) and the signal rate (baud rate) as we did for digital transmission. The relationship between them is

$$S = N \times \frac{1}{r} \quad \text{baud}$$

where N is the data rate (bps) and r is the number of data elements carried in one signal element. The value of r in analog transmission is $r = \log_2 L$, where L is the number of different signal elements. The same nomenclature is used to simplify the comparisons.

Bit rate is the number of bits per second. Baud rate is the number of signal elements per second. In the analog transmission of digital data, the baud rate is less than or equal to the bit rate.

The same analogy we used in Chapter 4 for bit rate and baud rate applies here. In transportation, a baud is analogous to a vehicle, and a bit is analogous to a passenger. We need to maximize the number of people per car to reduce the traffic.

Example 5.1

An analog signal carries 4 bits per signal element. If 1000 signal elements are sent per second, find the bit rate.

Solution

In this case, $r = 4$, $S = 1000$, and N is unknown. We can find the value of N from

$$S = N \times (1/r) \quad \text{or} \quad N = S \times r = 1000 \times 4 = 4000 \text{ bps}$$

Example 5.2

An analog signal has a bit rate of 8000 bps and a baud rate of 1000 baud. How many data elements are carried by each signal element? How many signal elements do we need?

Solution

In this example, $S = 1000$, $N = 8000$, and r and L are unknown. We first find the value of r and then the value of L.

$$S = N \times 1/r \longrightarrow r = N / S = 8000 / 10{,}000 = 8 \text{ bits/baud}$$
$$r = \log_2 L \longrightarrow L = 2^r = 2^8 = 256$$

Bandwidth

The required bandwidth for analog transmission of digital data is proportional to the signal rate except for FSK, in which the difference between the carrier signals needs to be added. We discuss the bandwidth for each technique.

Carrier Signal

In analog transmission, the sending device produces a high-frequency signal that acts as a base for the information signal. This base signal is called the **carrier signal** or *carrier frequency*. The receiving device is tuned to the frequency of the carrier signal that it expects from the sender. Digital information then changes the carrier signal by modifying one or more of its characteristics (amplitude, frequency, or phase). This kind of modification is called modulation (shift keying).

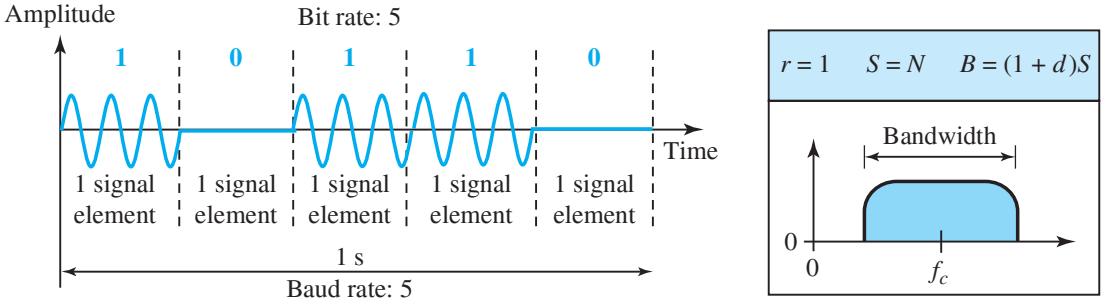
5.1.2 Amplitude Shift Keying

In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes.

Binary ASK (BASK)

Although we can have several levels (kinds) of signal elements, each with a different amplitude, ASK is normally implemented using only two levels. This is referred to as *binary amplitude shift keying* or *on-off keying* (OOK). The peak amplitude of one signal level is 0; the other is the same as the amplitude of the carrier frequency. Figure 5.3 gives a conceptual view of binary ASK.

Figure 5.3 *Binary amplitude shift keying*



Bandwidth for ASK

Figure 5.3 also shows the bandwidth for ASK. Although the carrier signal is only one simple sine wave, the process of modulation produces a nonperiodic composite signal. This signal, as was discussed in Chapter 3, has a continuous set of frequencies. As we expect, the bandwidth is proportional to the signal rate (baud rate). However, there is normally another factor involved, called d, which depends on the modulation and filtering process. The value of d is between 0 and 1. This means that the bandwidth can be expressed as shown, where S is the signal rate and the B is the bandwidth.

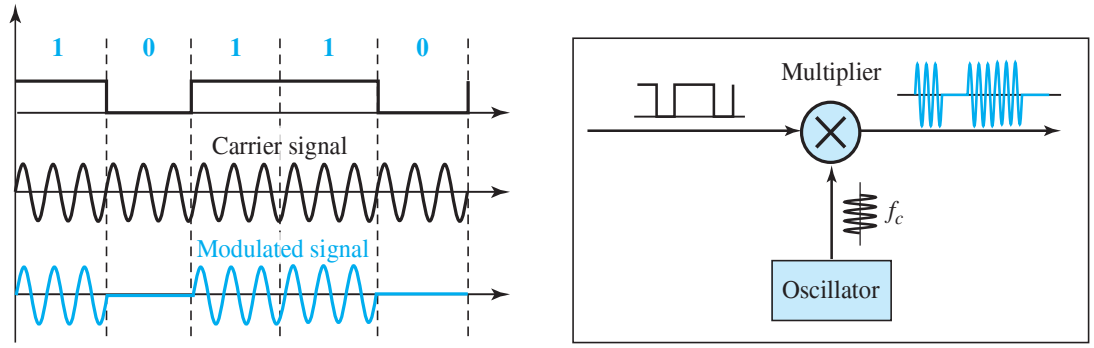
$$B = (1 + d) \times S$$

The formula shows that the required bandwidth has a minimum value of S and a maximum value of $2S$. The most important point here is the location of the bandwidth. The middle of the bandwidth is where f_c, the carrier frequency, is located. This means if we have a bandpass channel available, we can choose our f_c so that the modulated signal occupies that bandwidth. This is in fact the most important advantage of digital-to-analog conversion. We can shift the resulting bandwidth to match what is available.

Implementation

The complete discussion of ASK implementation is beyond the scope of this book. However, the simple ideas behind the implementation may help us to better understand the concept itself. Figure 5.4 shows how we can simply implement binary ASK.

Figure 5.4 *Implementation of binary ASK*



If digital data are presented as a unipolar NRZ (see Chapter 4) digital signal with a high voltage of 1 V and a low voltage of 0 V, the implementation can achieved by multiplying the NRZ digital signal by the carrier signal coming from an oscillator. When the amplitude of the NRZ signal is 1, the amplitude of the carrier frequency is held; when the amplitude of the NRZ signal is 0, the amplitude of the carrier frequency is zero.

Example 5.3

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What are the carrier frequency and the bit rate if we modulated our data by using ASK with $d = 1$?

Solution

The middle of the bandwidth is located at 250 kHz. This means that our carrier frequency can be at $f_c = 250$ kHz. We can use the formula for bandwidth to find the bit rate (with $d = 1$ and $r = 1$).

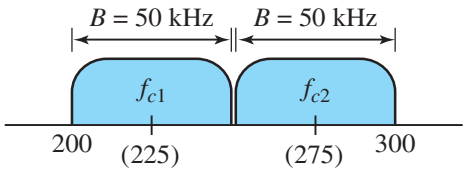
$$B = (1 + d) \times S = 2 \times N \times (1/r) = 2 \times N = 100 \text{ kHz} \quad \longrightarrow \quad N = 50 \text{ kbps}$$

Example 5.4

In data communications, we normally use full-duplex links with communication in both directions. We need to divide the bandwidth into two with two carrier frequencies, as shown in Figure 5.5. The figure shows the positions of two carrier frequencies and the bandwidths. The

available bandwidth for each direction is now 50 kHz, which leaves us with a data rate of 25 kbps in each direction.

Figure 5.5 *Bandwidth of full-duplex ASK used in Example 5.4*



Multilevel ASK

The above discussion uses only two amplitude levels. We can have multilevel ASK in which there are more than two levels. We can use 4, 8, 16, or more different amplitudes for the signal and modulate the data using 2, 3, 4, or more bits at a time. In these cases, $r = 2$, $r = 3$, $r = 4$, and so on. Although this is not implemented with pure ASK, it is implemented with QAM (as we will see later).

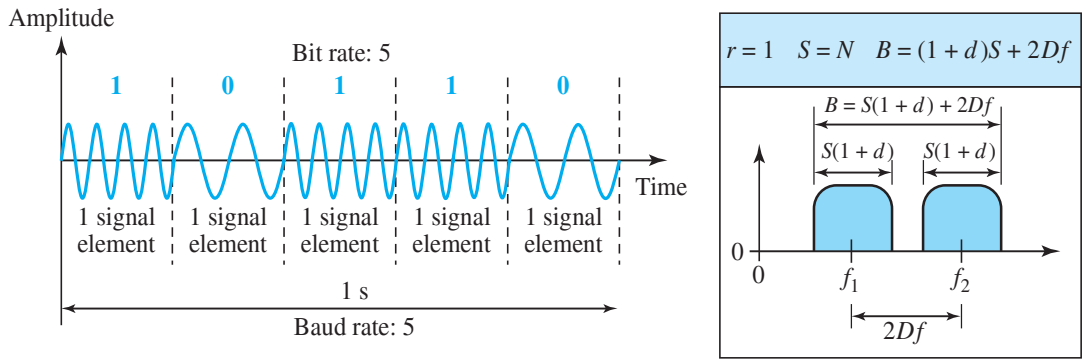
5.1.3 Frequency Shift Keying

In frequency shift keying, the frequency of the carrier signal is varied to represent data. The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes. Both peak amplitude and phase remain constant for all signal elements.

Binary FSK (BFSK)

One way to think about binary FSK (or BFSK) is to consider two carrier frequencies. In Figure 5.6, we have selected two carrier frequencies, f_1 and f_2. We use the first carrier if the data element is 0; we use the second if the data element is 1. However, note that this is an unrealistic example used only for demonstration purposes. Normally the carrier frequencies are very high, and the difference between them is very small.

Figure 5.6 *Binary frequency shift keying*



As Figure 5.6 shows, the middle of one bandwidth is f_1 and the middle of the other is f_2. Both f_1 and f_2 are Δ_f apart from the midpoint between the two bands. The difference between the two frequencies is $2\Delta_f$.

Bandwidth for BFSK

Figure 5.6 also shows the bandwidth of FSK. Again the carrier signals are only simple sine waves, but the modulation creates a nonperiodic composite signal with continuous frequencies. We can think of FSK as two ASK signals, each with its own carrier frequency (f_1 or f_2). If the difference between the two frequencies is $2\Delta_f$, then the required bandwidth is

$$B = (1 + d) \times S + 2\Delta\phi$$

What should be the minimum value of $2\Delta_f$? In Figure 5.6, we have chosen a value greater than $(1 + d)S$. It can be shown that the minimum value should be at least S for the proper operation of modulation and demodulation.

Example 5.5

We have an available bandwidth of 100 kHz which spans from 200 to 300 kHz. What should be the carrier frequency and the bit rate if we modulated our data by using FSK with $d = 1$?

Solution

This problem is similar to Example 5.3, but we are modulating by using FSK. The midpoint of the band is at 250 kHz. We choose $2\Delta_f$ to be 50 kHz; this means

$$B = (1 + d) \times S + 2\Delta_f = 100 \longrightarrow 2S = 50 \text{ kHz} \longrightarrow S = 25 \text{ kbaud} \longrightarrow N = 25 \text{ kbps}$$

Compared to Example 5.3, we can see the bit rate for ASK is 50 kbps while the bit rate for FSK is 25 kbps.

Implementation

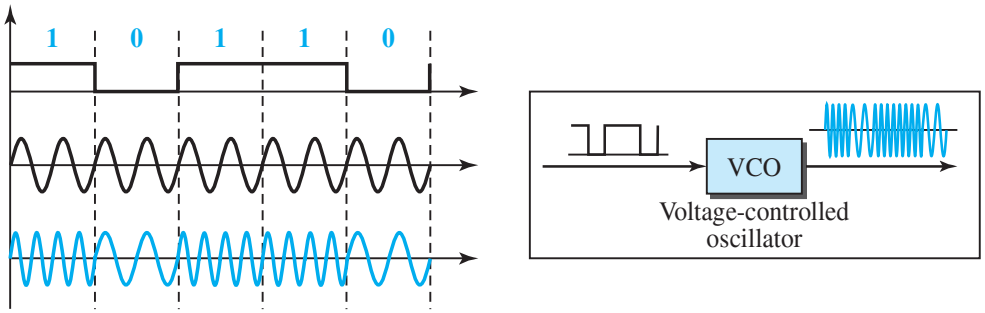
There are two implementations of BFSK: noncoherent and coherent. In noncoherent BFSK, there may be discontinuity in the phase when one signal element ends and the next begins. In coherent BFSK, the phase continues through the boundary of two signal elements. Noncoherent BFSK can be implemented by treating BFSK as two ASK modulations and using two carrier frequencies. Coherent BFSK can be implemented by using one *voltage-controlled oscillator* (VCO) that changes its frequency according to the input voltage. Figure 5.7 shows the simplified idea behind the second implementation. The input to the oscillator is the unipolar NRZ signal. When the amplitude of NRZ is zero, the oscillator keeps its regular frequency; when the amplitude is positive, the frequency is increased.

Multilevel FSK

Multilevel modulation (MFSK) is not uncommon with the FSK method. We can use more than two frequencies. For example, we can use four different frequencies f_1, f_2, f_3, and f_4 to send 2 bits at a time. To send 3 bits at a time, we can use eight frequencies. And so on. However, we need to remember that the frequencies need to be $2\Delta_f$ apart. For the proper operation of the modulator and demodulator, it can be shown that the minimum value of $2\Delta_f$ needs to be S. We can show that the bandwidth is

$$B = (1 + d) \times S + (L - 1)2\Delta_f \longrightarrow B = L \times S$$

Figure 5.7 *Implementation of BFSK*



Note that MFSK uses more bandwidth than the other techniques; it should be used when noise is a serious issue.

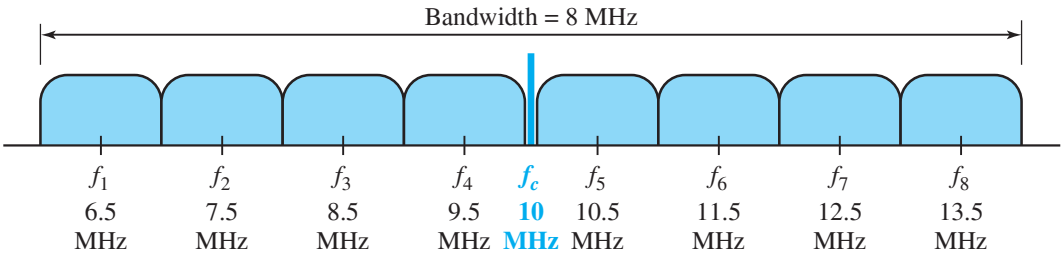
Example 5.6

We need to send data 3 bits at a time at a bit rate of 3 Mbps. The carrier frequency is 10 MHz. Calculate the number of levels (different frequencies), the baud rate, and the bandwidth.

Solution

We can have $L = 2^3 = 8$. The baud rate is $S = 3$ MHz/3 = 1 Mbaud. This means that the carrier frequencies must be 1 MHz apart ($2\Delta_f = 1$ MHz). The bandwidth is $B = 8 \times 1 = 8$ MHz. Figure 5.8 shows the allocation of frequencies and bandwidth.

Figure 5.8 *Bandwidth of MFSK used in Example 5.6*



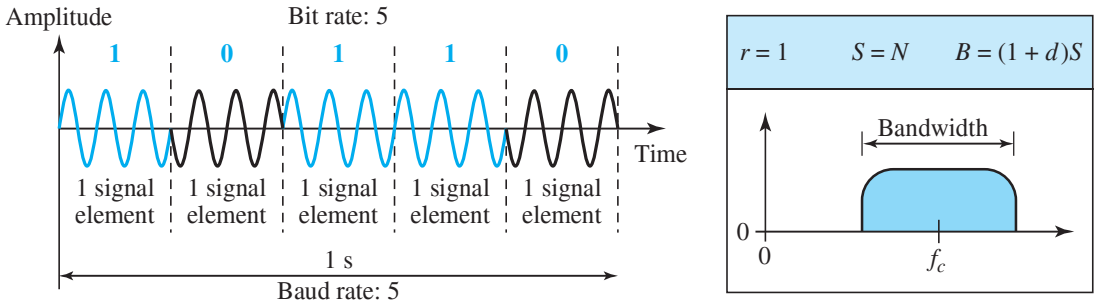
5.1.4 Phase Shift Keying

In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements. Both peak amplitude and frequency remain constant as the phase changes. Today, PSK is more common than ASK or FSK. However, we will see shortly that QAM, which combines ASK and PSK, is the dominant method of digital-to-analog modulation.

Binary PSK (BPSK)

The simplest PSK is binary PSK, in which we have only two signal elements, one with a phase of 0°, and the other with a phase of 180°. Figure 5.9 gives a conceptual view of PSK. Binary PSK is as simple as binary ASK with one big advantage—it is less susceptible to noise. In ASK, the criterion for bit detection is the amplitude of the

Figure 5.9 *Binary phase shift keying*



signal; in PSK, it is the phase. Noise can change the amplitude easier than it can change the phase. In other words, PSK is less susceptible to noise than ASK. PSK is superior to FSK because we do not need two carrier signals. However, PSK needs more sophisticated hardware to be able to distinguish between phases.

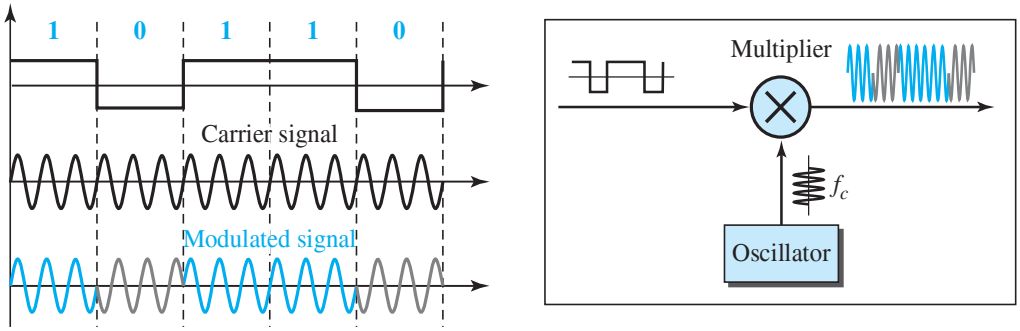
Bandwidth

Figure 5.9 also shows the bandwidth for BPSK. The bandwidth is the same as that for binary ASK, but less than that for BFSK. No bandwidth is wasted for separating two carrier signals.

Implementation

The implementation of BPSK is as simple as that for ASK. The reason is that the signal element with phase 180° can be seen as the complement of the signal element with phase 0°. This gives us a clue on how to implement BPSK. We use the same idea we used for ASK but with a polar NRZ signal instead of a unipolar NRZ signal, as shown in Figure 5.10. The polar NRZ signal is multiplied by the carrier frequency; the 1 bit (positive voltage) is represented by a phase starting at 0°; the 0 bit (negative voltage) is represented by a phase starting at 180°.

Figure 5.10 *Implementation of BASK*

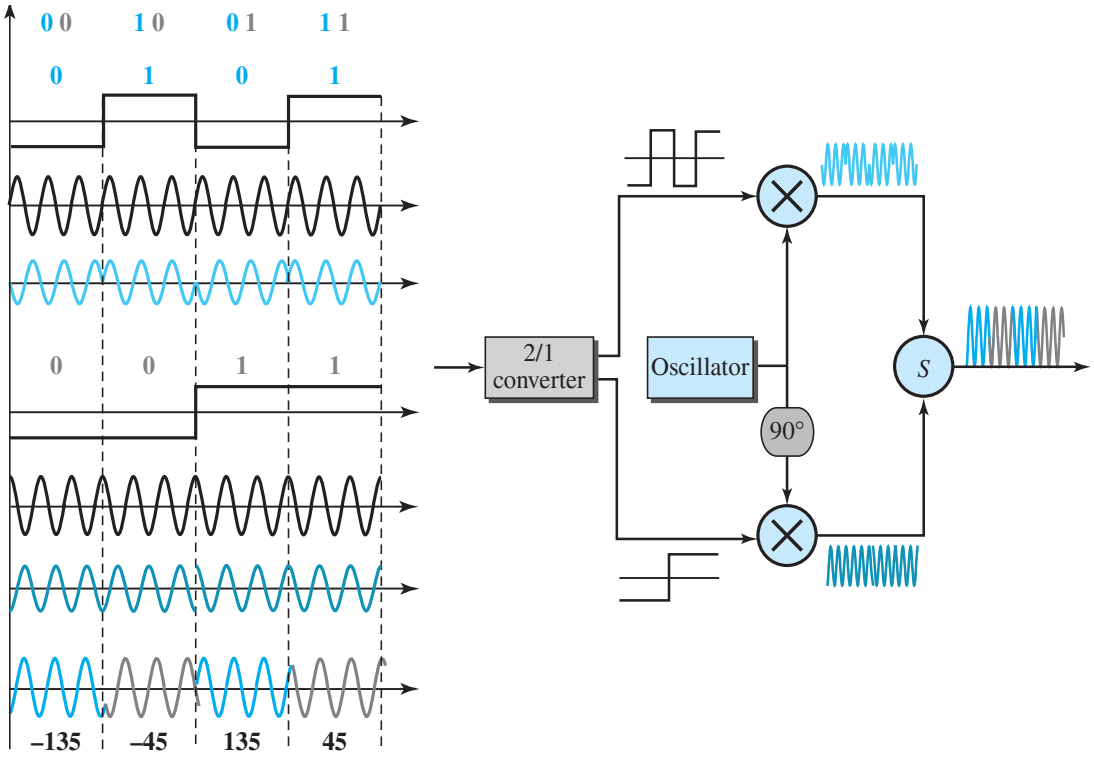


Quadrature PSK (QPSK)

The simplicity of BPSK enticed designers to use 2 bits at a time in each signal element, thereby decreasing the baud rate and eventually the required bandwidth. The scheme is

called *quadrature PSK* or *QPSK* because it uses two separate BPSK modulations; one is in-phase, the other quadrature (out-of-phase). The incoming bits are first passed through a serial-to-parallel conversion that sends one bit to one modulator and the next bit to the other modulator. If the duration of each bit in the incoming signal is T, the duration of each bit sent to the corresponding BPSK signal is $2T$. This means that the bit to each BPSK signal has one-half the frequency of the original signal. Figure 5.11 shows the idea.

Figure 5.11 *QPSK and its implementation*



The two composite signals created by each multiplier are sine waves with the same frequency, but different phases. When they are added, the result is another sine wave, with one of four possible phases: 45°, –45°, 135°, and –135°. There are four kinds of signal elements in the output signal ($L = 4$), so we can send 2 bits per signal element ($r = 2$).

Example 5.7

Find the bandwidth for a signal transmitting at 12 Mbps for QPSK. The value of $d = 0$.

Solution

For QPSK, 2 bits are carried by one signal element. This means that $r = 2$. So the signal rate (baud rate) is $S = N \times (1/r) = 6$ Mbaud. With a value of $d = 0$, we have $B = S = 6$ MHz.

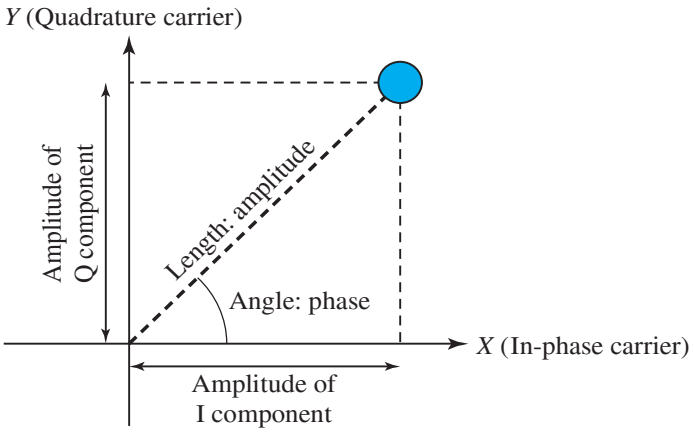
Constellation Diagram

A **constellation diagram** can help us define the amplitude and phase of a signal element, particularly when we are using two carriers (one in-phase and one quadrature). The

diagram is useful when we are dealing with multilevel ASK, PSK, or QAM (see next section). In a constellation diagram, a signal element type is represented as a dot. The bit or combination of bits it can carry is often written next to it.

The diagram has two axes. The horizontal *X* axis is related to the in-phase carrier; the vertical *Y* axis is related to the quadrature carrier. For each point on the diagram, four pieces of information can be deduced. The projection of the point on the *X* axis defines the peak amplitude of the in-phase component; the projection of the point on the *Y* axis defines the peak amplitude of the quadrature component. The length of the line (vector) that connects the point to the origin is the peak amplitude of the signal element (combination of the *X* and *Y* components); the angle the line makes with the *X* axis is the phase of the signal element. All the information we need can easily be found on a constellation diagram. Figure 5.12 shows a constellation diagram.

Figure 5.12 *Concept of a constellation diagram*



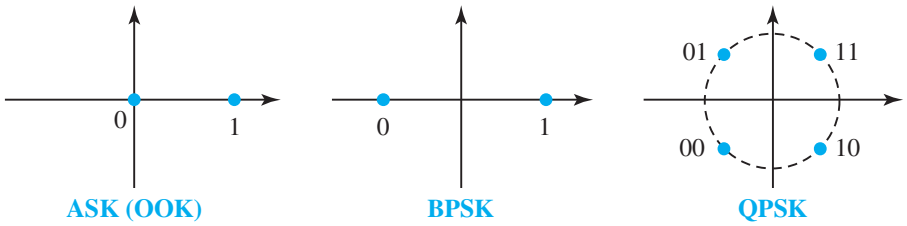
Example 5.8

Show the constellation diagrams for ASK (OOK), BPSK, and QPSK signals.

Solution

Figure 5.13 shows the three constellation diagrams. Let us analyze each case separately:

Figure 5.13 *Three constellation diagrams*



- For ASK, we are using only an in-phase carrier. Therefore, the two points should be on the *X* axis. Binary 0 has an amplitude of 0 V; binary 1 has an amplitude of 1 V (for example). The points are located at the origin and at 1 unit.

- BPSK also uses only an in-phase carrier. However, we use a polar NRZ signal for modulation. It creates two types of signal elements, one with amplitude 1 and the other with amplitude −1. This can be stated in other words: BPSK creates two different signal elements, one with amplitude 1 V and in phase and the other with amplitude 1 V and 180° out of phase.
- QPSK uses two carriers, one in-phase and the other quadrature. The point representing 11 is made of two combined signal elements, both with an amplitude of 1 V. One element is represented by an in-phase carrier, the other element by a quadrature carrier. The amplitude of the final signal element sent for this 2-bit data element is $2^{1/2}$, and the phase is 45°. The argument is similar for the other three points. All signal elements have an amplitude of $2^{1/2}$, but their phases are different (45°, 135°, −135°, and −45°). Of course, we could have chosen the amplitude of the carrier to be $1/(2^{1/2})$ to make the final amplitudes 1 V.

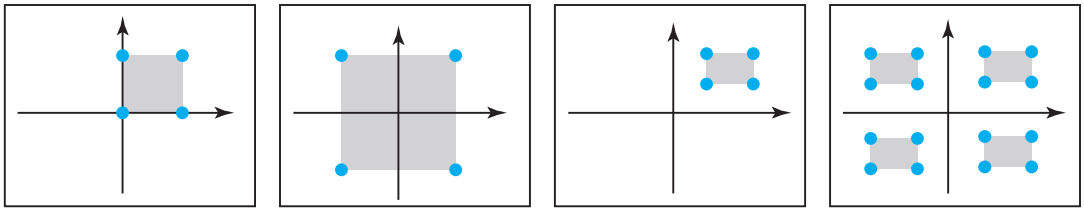
5.1.5 Quadrature Amplitude Modulation

PSK is limited by the ability of the equipment to distinguish small differences in phase. This factor limits its potential bit rate. So far, we have been altering only one of the three characteristics of a sine wave at a time; but what if we alter two? Why not combine ASK and PSK? The idea of using two carriers, one in-phase and the other quadrature, with different amplitude levels for each carrier is the concept behind **quadrature amplitude modulation (QAM).**

Quadrature amplitude modulation is a combination of ASK and PSK.

The possible variations of QAM are numerous. Figure 5.14 shows some of these schemes. Figure 5.14a shows the simplest 4-QAM scheme (four different signal element types) using a unipolar NRZ signal to modulate each carrier. This is the same mechanism we used for ASK (OOK). Part b shows another 4-QAM using polar NRZ, but this is exactly the same as QPSK. Part c shows another QAM-4 in which we used a signal with two positive levels to modulate each of the two carriers. Finally, Figure 5.14d shows a 16-QAM constellation of a signal with eight levels, four positive and four negative.

Figure 5.14 *Constellation diagrams for some QAMs*



a. 4-QAM b. 4-QAM c. 4-QAM d. 16-QAM

Bandwidth for QAM

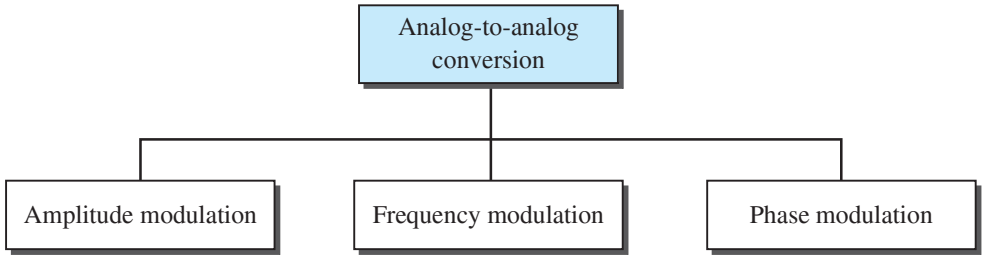
The minimum bandwidth required for QAM transmission is the same as that required for ASK and PSK transmission. QAM has the same advantages as PSK over ASK.

5.2 ANALOG-TO-ANALOG CONVERSION

Analog-to-analog conversion, or analog modulation, is the representation of analog information by an analog signal. One may ask why we need to modulate an analog signal; it is already analog. Modulation is needed if the medium is bandpass in nature or if only a bandpass channel is available to us. An example is radio. The government assigns a narrow bandwidth to each radio station. The analog signal produced by each station is a low-pass signal, all in the same range. To be able to listen to different stations, the low-pass signals need to be shifted, each to a different range.

Analog-to-analog conversion can be accomplished in three ways: **amplitude modulation (AM), frequency modulation (FM),** and **phase modulation (PM).** FM and PM are usually categorized together. See Figure 5.15.

Figure 5.15 *Types of analog-to-analog modulation*



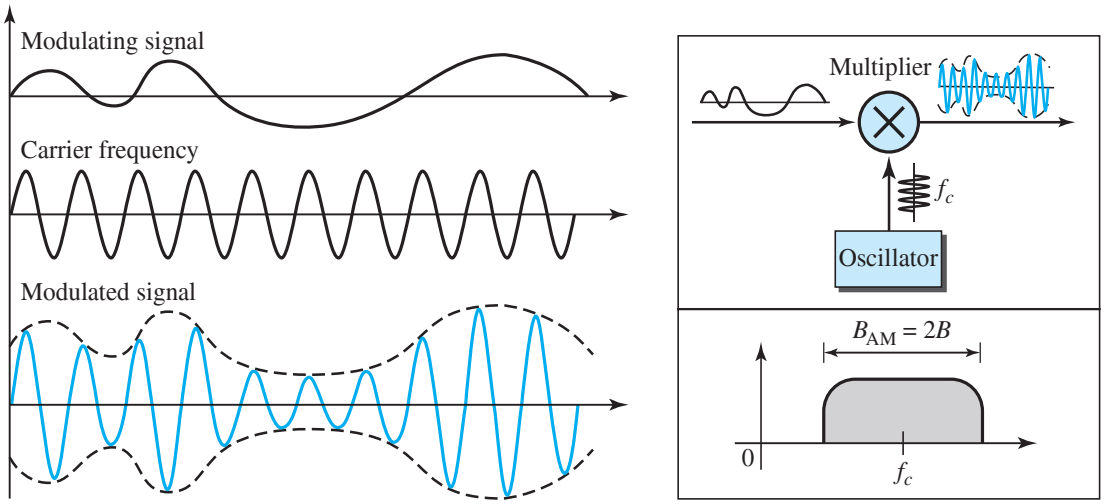
5.2.1 Amplitude Modulation (AM)

In AM transmission, the carrier signal is modulated so that its amplitude varies with the changing amplitudes of the modulating signal. The frequency and phase of the carrier remain the same; only the amplitude changes to follow variations in the information. Figure 5.16 shows how this concept works. The modulating signal is the envelope of the carrier. As Figure 5.16 shows, AM is normally implemented by using a simple multiplier because the amplitude of the carrier signal needs to be changed according to the amplitude of the modulating signal.

AM Bandwidth

Figure 5.16 also shows the bandwidth of an AM signal. The modulation creates a bandwidth that is twice the bandwidth of the modulating signal and covers a range centered on the carrier frequency. However, the signal components above and below the carrier frequency carry exactly the same information. For this reason, some implementations discard one-half of the signals and cut the bandwidth in half.

Figure 5.16 *Amplitude modulation*



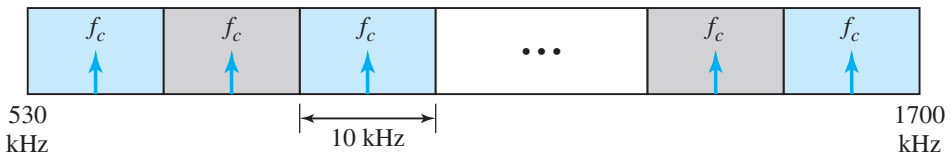
The total bandwidth required for AM can be determined from the bandwidth of the audio signal: $B_{AM} = 2B$.

Standard Bandwidth Allocation for AM Radio

The bandwidth of an audio signal (speech and music) is usually 5 kHz. Therefore, an AM radio station needs a bandwidth of 10 kHz. In fact, the Federal Communications Commission (FCC) allows 10 kHz for each AM station.

AM stations are allowed carrier frequencies anywhere between 530 and 1700 kHz (1.7 MHz). However, each station's carrier frequency must be separated from those on either side of it by at least 10 kHz (one AM bandwidth) to avoid interference. If one station uses a carrier frequency of 1100 kHz, the next station's carrier frequency cannot be lower than 1110 kHz (see Figure 5.17).

Figure 5.17 *AM band allocation*



5.2.2 Frequency Modulation (FM)

In FM transmission, the frequency of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and phase of the carrier signal remain constant, but as the amplitude of the information signal changes, the frequency of the carrier changes correspondingly. Figure 5.18 shows the relationships of the modulating signal, the carrier signal, and the resultant FM signal.

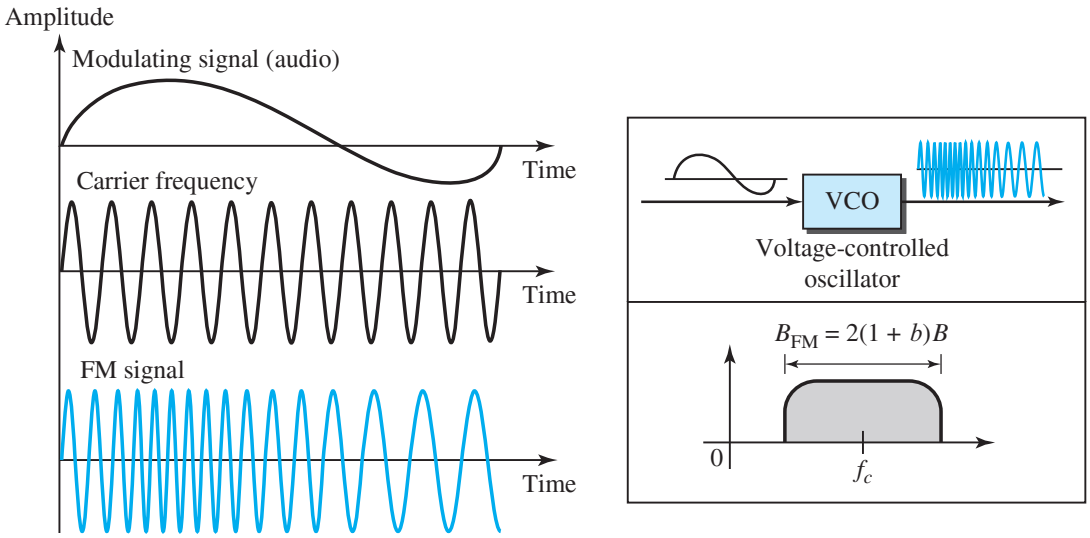
As Figure 5.18 shows, FM is normally implemented by using a voltage-controlled oscillator as with FSK. The frequency of the oscillator changes according to the input voltage which is the amplitude of the modulating signal.

FM Bandwidth

Figure 5.18 also shows the bandwidth of an FM signal. The actual bandwidth is difficult to determine exactly, but it can be shown empirically that it is several times that of the analog signal or $2(1 + \beta)B$ where β is a factor that depends on modulation technique with a common value of 4.

The total bandwidth required for FM can be determined from the bandwidth of the audio signal: $B_{FM} = 2(1 \times \beta)B$.

Figure 5.18 *Frequency modulation*



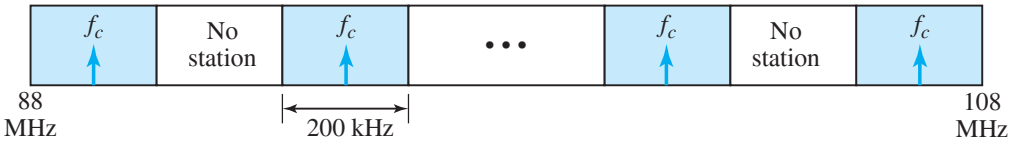
Standard Bandwidth Allocation for FM Radio

The bandwidth of an audio signal (speech and music) broadcast in stereo is almost 15 kHz. The FCC allows 200 kHz (0.2 MHz) for each station. This mean $\beta = 4$ with some extra guard band. FM stations are allowed carrier frequencies anywhere between 88 and 108 MHz. Stations must be separated by at least 200 kHz to keep their bandwidths from overlapping. To create even more privacy, the FCC requires that in a given area, only alternate bandwidth allocations may be used. The others remain unused to prevent any possibility of two stations interfering with each other. Given 88 to 108 MHz as a range, there are 100 potential FM bandwidths in an area, of which 50 can operate at any one time. Figure 5.19 illustrates this concept.

5.2.3 Phase Modulation (PM)

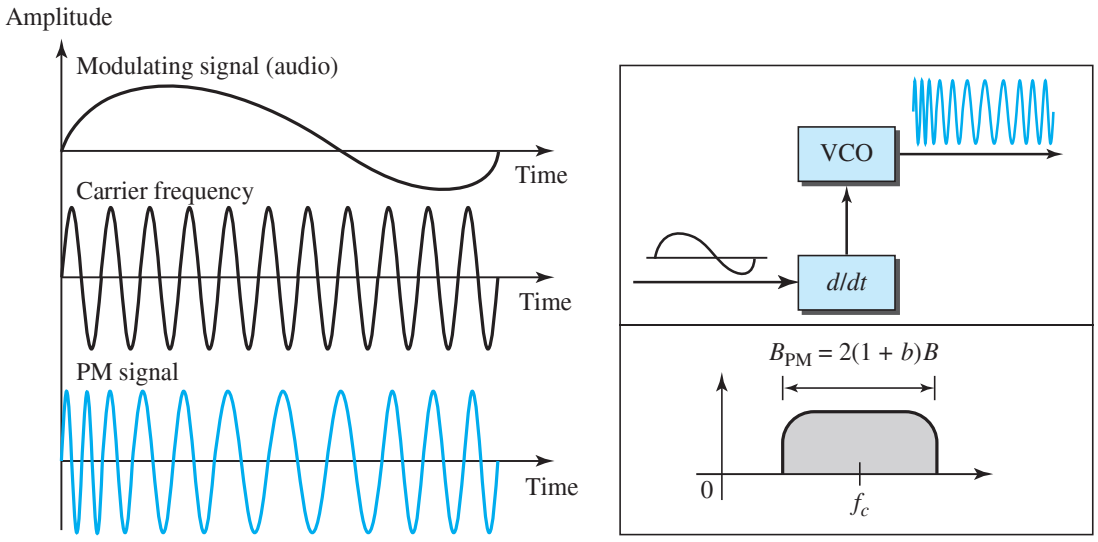
In PM transmission, the phase of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and frequency

Figure 5.19 *FM band allocation*



of the carrier signal remain constant, but as the amplitude of the information signal changes, the phase of the carrier changes correspondingly. It can be proved mathematically (see Appendix E) that PM is the same as FM with one difference. In FM, the instantaneous change in the carrier frequency is proportional to the amplitude of the modulating signal; in PM the instantaneous change in the carrier frequency is proportional to the derivative of the amplitude of the modulating signal. Figure 5.20 shows the relationships of the modulating signal, the carrier signal, and the resultant PM signal.

Figure 5.20 *Phase modulation*



As Figure 5.20 shows, PM is normally implemented by using a voltage-controlled oscillator along with a derivative. The frequency of the oscillator changes according to the derivative of the input voltage, which is the amplitude of the modulating signal.

PM Bandwidth

Figure 5.20 also shows the bandwidth of a PM signal. The actual bandwidth is difficult to determine exactly, but it can be shown empirically that it is several times that of the analog signal. Although the formula shows the same bandwidth for FM and PM, the value of β is lower in the case of PM (around 1 for narrowband and 3 for wideband).

The total bandwidth required for PM can be determined from the bandwidth and maximum amplitude of the modulating signal: $B_{PM} = 2(1 + \beta)B$.

5.3 END-CHAPTER MATERIALS

5.3.1 Recommended Reading

For more details about subjects discussed in this chapter, we recommend the following books. The items in brackets [. . .] refer to the reference list at the end of the text.

Books

Digital-to-analog conversion is discussed in [Pea92], [Cou01], and [Sta04]. Analog-to-analog conversion is discussed in [Pea92], Chapter 5 of [Cou01], [Sta04]. [Hsu03] gives a good mathematical approach to all materials discussed in this chapter. More advanced materials can be found in [Ber96].

5.3.2 Key Terms

amplitude modulation (AM)
amplitude shift keying (ASK)
analog-to-analog conversion
carrier signal
constellation diagram
digital-to-analog conversion
frequency modulation (FM)
frequency shift keying (FSK)
phase modulation (PM)
phase shift keying (PSK)
quadrature amplitude modulation (QAM)

5.3.3 Summary

Digital-to-analog conversion is the process of changing one of the characteristics of an analog signal based on the information in the digital data. Digital-to-analog conversion can be accomplished in several ways: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). Quadrature amplitude modulation (QAM) combines ASK and PSK. In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes. In frequency shift keying, the frequency of the carrier signal is varied to represent data. The frequency of the modulated signal is constant for the duration of one signal element, but changes for the next signal element if the data element changes. Both peak amplitude and phase remain constant for all signal elements. In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements. Both peak amplitude and frequency remain constant as the phase changes. A constellation diagram shows us the amplitude and phase of a signal element, particularly when we are using two carriers (one in-phase and one quadrature). Quadrature amplitude modulation (QAM) is a combination of ASK and PSK. QAM uses two carriers, one in-phase and the other quadrature, with different amplitude levels for each carrier. Analog-to-analog conversion is the representation of analog information by an analog signal. Conversion is needed if the medium is bandpass in nature or if only a bandpass bandwidth is available to us.

Analog-to-analog conversion can be accomplished in three ways: amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM). In AM transmission, the carrier signal is modulated so that its amplitude varies with the changing amplitudes of the modulating signal. The frequency and phase of the carrier remain the same; only the amplitude changes to follow variations in the information. In FM transmission, the frequency of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and phase of the carrier signal remain constant, but as the amplitude of the information signal changes, the frequency of the carrier changes correspondingly. In PM transmission, the phase of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal. The peak amplitude and frequency of the carrier signal remain constant, but as the amplitude of the information signal changes, the phase of the carrier changes correspondingly.

5.4 PRACTICE SET

5.4.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

5.4.2 Questions

Q5-1. Define *analog transmission*.

Q5-2. Define *carrier signal* and explain its role in analog transmission.

Q5-3. Define *digital-to-analog conversion*.

Q5-4. Which characteristics of an analog signal are changed to represent the digital signal in each of the following digital-to-analog conversions?

a. ASK **b.** FSK **c.** PSK **d.** QAM

Q5-5. Which of the four digital-to-analog conversion techniques (ASK, FSK, PSK or QAM) is the most susceptible to noise? Defend your answer.

Q5-6. Define *constellation diagram* and explain its role in analog transmission.

Q5-7. What are the two components of a signal when the signal is represented on a constellation diagram? Which component is shown on the horizontal axis? Which is shown on the vertical axis?

Q5-8. Define *analog-to-analog conversion*.

Q5-9. Which characteristics of an analog signal are changed to represent the lowpass analog signal in each of the following analog-to-analog conversions?

a. AM **b.** FM **c.** PM

Q5-10. Which of the three analog-to-analog conversion techniques (AM, FM, or PM) is the most susceptible to noise? Defend your answer.

5.4.3 Problems

P5-1. Calculate the baud rate for the given bit rate and type of modulation.

a. 2000 bps, FSK

b. 4000 bps, ASK

c. 6000 bps, QPSK

d. 36,000 bps, 64-QAM

P5-2. Calculate the bit rate for the given baud rate and type of modulation.

a. 1000 baud, FSK

b. 1000 baud, ASK

c. 1000 baud, BPSK

d. 1000 baud, 16-QAM

P5-3. What is the number of bits per baud for the following techniques?

a. ASK with four different amplitudes

b. FSK with eight different frequencies

c. PSK with four different phases

d. QAM with a constellation of 128 points

P5-4. Draw the constellation diagram for the following:

a. ASK, with peak amplitude values of 1 and 3

b. BPSK, with a peak amplitude value of 2

c. QPSK, with a peak amplitude value of 3

d. 8-QAM with two different peak amplitude values, 1 and 3, and four different phases

P5-5. Draw the constellation diagram for the following cases. Find the peak amplitude value for each case and define the type of modulation (ASK, FSK, PSK, or QAM).The numbers in parentheses define the values of I and Q respectively.

a. Two points at (2, 0) and (3, 0)

b. Two points at (3, 0) and (−3, 0)

c. Four points at (2, 2), (−2, 2), (−2, −2), and (2, −2)

d. Two points at (0, 2) and (0, −2)

P5-6. How many bits per baud can we send in each of the following cases if the signal constellation has one of the following number of points?

a. 2 **b.** 4 **c.** 16 **d.** 1024

P5-7. What is the required bandwidth for the following cases if we need to send 4000 bps? Let d = 1.

a. ASK

b. FSK with $2\Delta f = 4$ KHz

c. QPSK

d. 16-QAM

P5-8. The telephone line has 4 KHz bandwidth. What is the maximum number of bits we can send using each of the following techniques? Let d = 0.

a. ASK **b.** QPSK **c.** 16-QAM **d.** 64-QAM

P5-9. A corporation has a medium with a 1-MHz bandwidth (lowpass). The corporation needs to create 10 separate independent channels each capable of sending at least 10 Mbps. The company has decided to use QAM technology. What is the minimum number of bits per baud for each channel? What is the number of points in the constellation diagram for each channel? Let d = 0.

P5-10. A cable company uses one of the cable TV channels (with a bandwidth of 6 MHz) to provide digital communication for each resident. What is the available data rate for each resident if the company uses a 64-QAM technique?

P5-11. Find the bandwidth for the following situations if we need to modulate a 5-KHz voice.

a. AM **b.** FM ($\beta = 5$) **c.** PM ($\beta = 1$)

P5-12. Find the total number of channels in the corresponding band allocated by FCC.

a. AM **b.** FM

5.5 SIMULATION EXPERIMENTS

5.5.1 Applets

We have created some Java applets to show some of the main concepts discussed in this chapter. It is strongly recommended that the students activate these applets on the book website and carefully examine the protocols in action.