

Saved from: www.uotechnology.edu.iq/dep-cs



4th Class

2015-2016

Communications

الاتصالات

أستاذ المادة : م.د. إخلص خلف

Communications

Fourth class/Networks Management Branch

By:Dr. Ekhlas Khalaf

Introduction

Data communications and networking are changing 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 Germany to arrive by mail when it could appear almost instantaneously through computer networks? Businesses today rely on computer networks and internetworks 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.

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 (teleis 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.

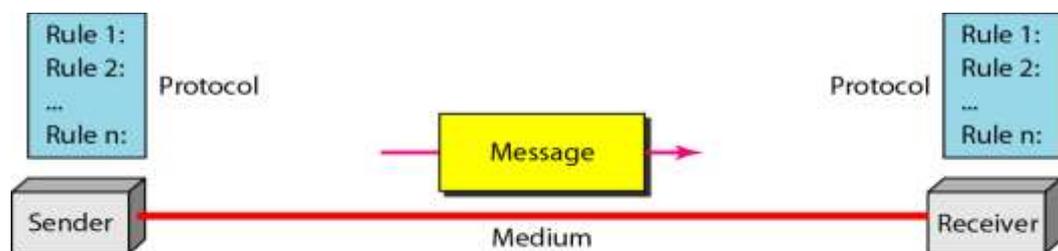
2. **Accuracy.** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.

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 3D ms. If some of the packets arrive with 3D-ms delay and others with 4D-ms delay, an uneven quality in the video is the result.

Components

Data Communication system has five Components (see figure 1)



FIGURE(!)

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.

NETWORKS

A network is a set of devices (often referred to as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

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 you change television channels by infrared remote control, you 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). 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.

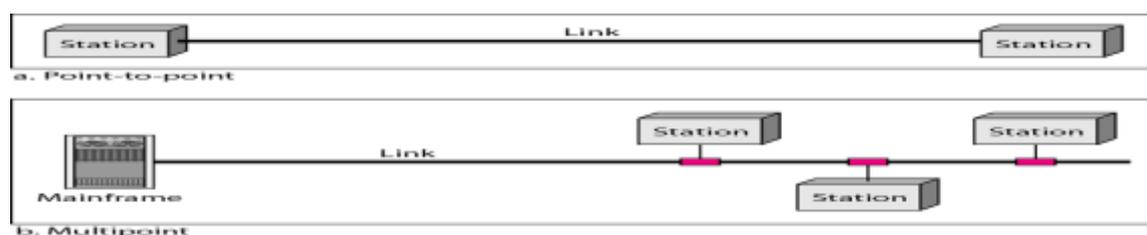


Figure (2) type of connection

Network Topology

Topology of a network refers to the configuration ترتيب of cables, computers, and other peripherals, topology should not be confused مشوش with logical topology which is the method used to pass information between workstations (computers). The main types of topologies are:

- a. Star Topology: Star topology consists of a central node to which all other nodes are connected.
- b. Bus Topology: Bus topology was the basis for most of the original LAN networks. Ideally suited for use with coaxial cable, the bus topology is a single length of transmission medium with nodes connected to it.
- c. Ring Topology: Ring topology uses lengths of transmission media to connect the nodes; each node is attached to its neighbor. The transmission signal moves around the ring in one direction and is repeated, instead of just passed, as it moves from node to node. When a station transmits a data message, the transmission is picked up by the next station on the ring, examined, and then retransmitted to the downstream neighbor. This process is continued until the transmitted signal is returned to the host that started the transmission, which then removes the data from the network.
- d. Tree Topology: A tree topology combines the characteristics of linear bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable. Tree topologies allow for the expansion of an existing network.

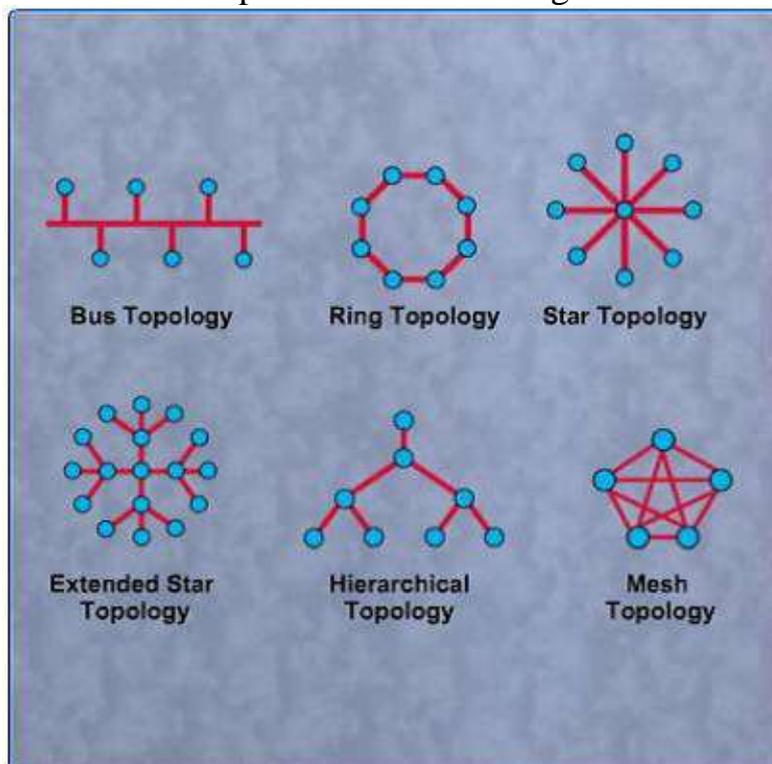


Figure (3) categories of Topology

Categories of Networks

Today when we speak of networks, we are generally referring to two primary categories: local-area networks and wide-area networks. The category into which a network falls is determined by its size. A LAN normally covers an area less than 2 mi; aWAN can be worldwide. Networks of a size in between are normally referred to as metropolitan area networks and span tens of miles.

Local Area Network

A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus. Depending on the needs of an organization and the type of technology used, 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 peripherals. Currently, LAN size is limited to a few kilometers.

LANs are designed to allow resources to be shared between personal computers or workstations. The resources to be shared can include hardware (e.g., a printer), software (e.g., an application program), or data. A common example of a LAN, found in many business environments, links a workgroup of task-related computers, for example, engineering workstations or accounting PCs. One of the computers may be given a large capacity disk drive and may become a server to clients. Software can be stored on this central server and used as needed by the whole group. In this example, the size of the LAN may be determined by licensing restrictions on the number of users per copy of software, or by restrictions on the number of users licensed to access the operating system.

In addition to size, LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium. The most common LAN topologies are bus, ring, and star.

Wide Area Network

A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world.. A WAN can be as complex as the backbones that connect the Internet or as simple as a dial-up line that connects a home computer to the Internet. We normally refer to the first as a switched WAN and to the second as a point-to-point WAN (Figure 1.11). The switched WAN connects the end systems, which usually comprise a router (internetworking connecting device) that connects to another LAN or WAN. The point-to-point WAN is normally a line leased from a telephone or cable TV provider that connects a home

computer or a small LAN to an Internet service provider (ISP). This type of WAN is often used to provide Internet access.

Metropolitan Area Networks

A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high-speed DSL line to the customer. Another example is the cable TV network that originally was designed for cable TV, but today can also be used for high-speed data connection to the internet.

Network Models

A network is a combination of hardware and software that sends data from one location to another. The hardware consists of the physical equipment that carries signals from one point of the network to another. The software consists of instruction sets that make possible the services that we expect from a network.

LAYERED TASKS

We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail the process of sending a letter to a friend would be complex if there were no services available from the post office. Figure 4 shows the steps in this task.

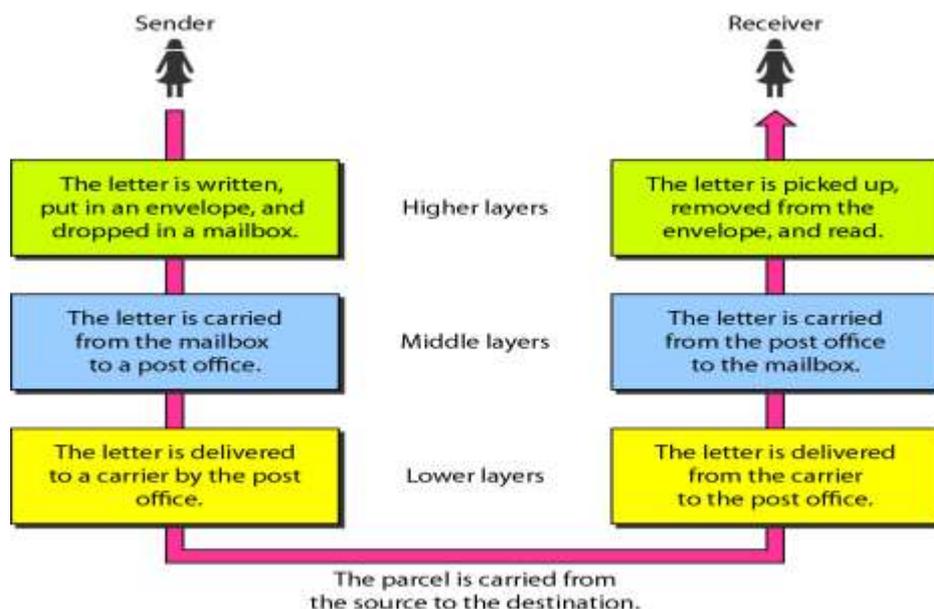


Figure (4)Tasks involved in sending a letter

Sender, Receiver, and Carrier

In Figure 4 we have a sender, a receiver, and a carrier that transports the letter. There is a hierarchy of tasks.

At the Sender Site

Let us first describe, in order, the activities that take place at the sender site.

- o **Higher layer.** The sender writes the letter, inserts the letter in an envelope, writes the sender and receiver addresses, and drops the letter in a mailbox.
- o **Middle layer.** The letter is picked up by a letter carrier and delivered to the post office.
- o **Lower layer.** The letter is sorted at the post office; a carrier transports the letter.

On the Way

The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the Receiver Site

- o **Lower layer.** The carrier transports the letter to the post office.
- o **Middle layer.** The letter is sorted and delivered to the recipient's mailbox.
- o **Higher layer.** The receiver picks up the letter, opens the envelope, and reads it.

Hierarchy

According to our analysis, there are three different activities at the sender site and another three activities at the receiver site. The task of

transporting the letter between the sender and the receiver is done by the carrier. Something that is not obvious immediately is that the tasks must be done in the order given in the hierarchy. At the sender site, the letter must be written and dropped in the mailbox before being picked

up by the letter carrier and delivered to the post office. At the receiver site, the letter must be dropped in the recipient mailbox before being picked up and read by the recipient.

Services

Each layer at the sending site uses the services of the layer immediately below it. The sender at the higher layer uses the services of the middle layer. The middle layer uses the services of the lower layer. The lower layer uses the services of the carrier. The layered model that dominated data communications and networking literature before 1990 was the Open Systems Interconnection (OSI) model. Everyone believed

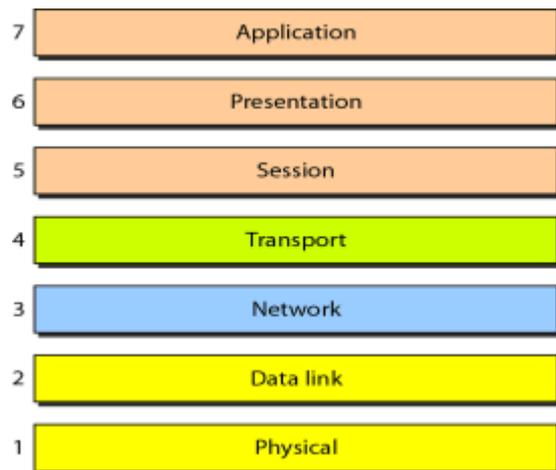
that the OSI model would become the ultimate standard for data communications, but this did not happen. The TCPIIP protocol suite became the dominant commercial architecture because it was used and tested extensively in the Internet; the OSI model was never fully implemented. In this chapter, first we briefly discuss the OSI model, and then we concentrate on TCPIIP as a protocol suite.

THE OSI MODEL

Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection model. It was first introduced in the late 1970s. 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.

ISO is the organization. OSI is the model.

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 2.2). An understanding of the fundamentals of the OSI model provides a solid basis for exploring data communications.



Figure(5) Seven layers of the OSI model

Layered Architecture

The OSI model is composed of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7). Figure 5 shows the layers involved when a message is sent from device A to device B. As the message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes usually involve only the first three layers of the OSI model.

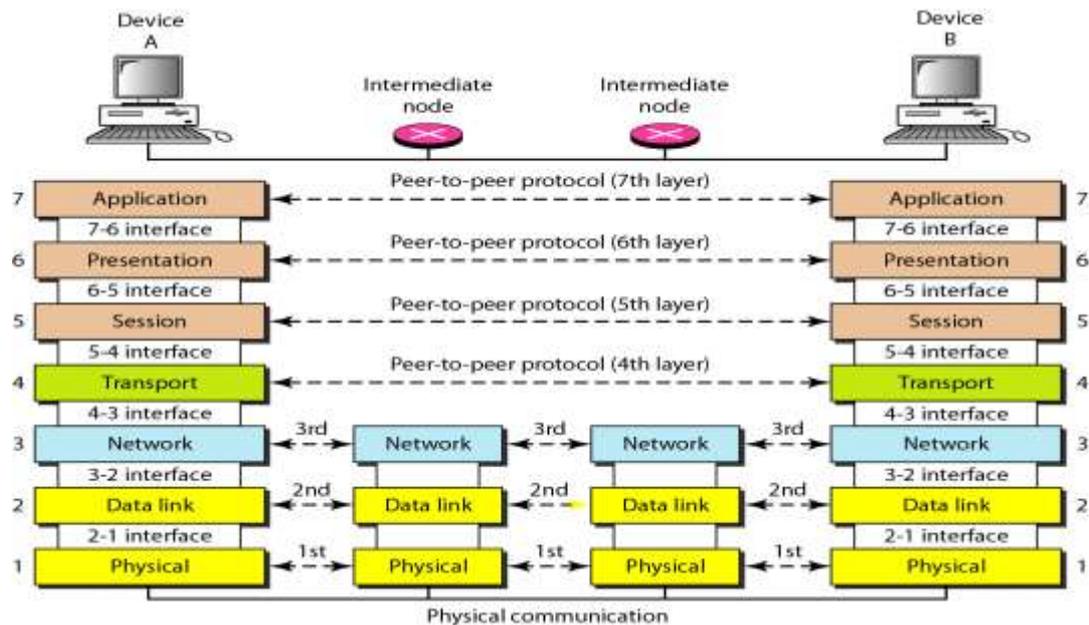


Figure (6) The interaction between layers in the OSI model

Interfaces Between Layers

The passing of the data and network information down through the layers of the sending device and back up through the layers of the receiving device is made possible by an interface between each pair of adjacent layers. Each interface defines the information and services a layer must provide for the layer above it. Well-defined interfaces and layer functions provide modularity to a network. As long as a layer provides the expected services to the layer above it, the specific implementation of its functions can be modified or replaced without requiring changes to the surrounding layers.

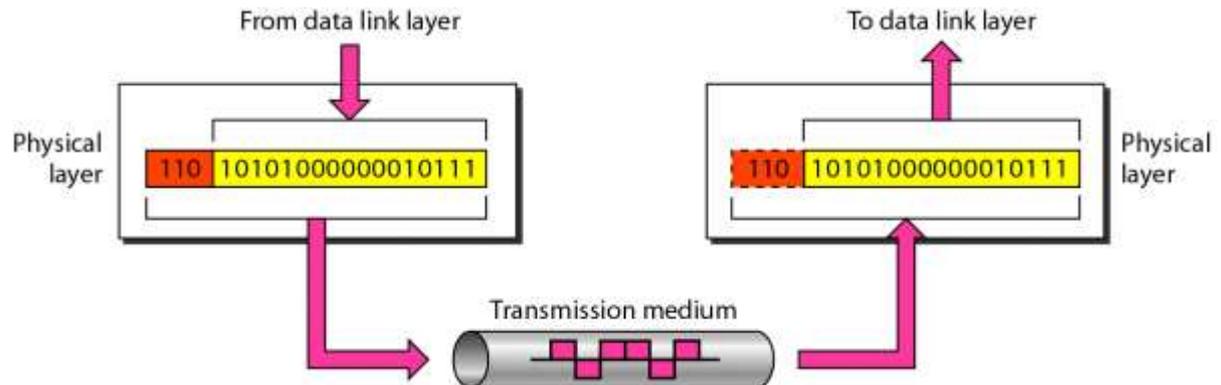
LAYERS IN THE OSI MODEL

In this section we briefly describe the functions of each layer in the OSI model.

1-Physical Layer

The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur. Figure 2.5 shows the position of the

physical layer with respect to the transmission medium and the data link layer.



The physical layer is also concerned with the following:

- ❖ Physical characteristics of interfaces and medium. The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
- ❖ Representation of bits. The physical layer data consists of a stream of bits (sequence of 0s or 1s) with no interpretation. To be transmitted, bits must be encoded into signals--electrical or optical. The physical layer defines the type of encoding (how 0s and 1s are changed to signals).
- ❖ Data rate. The transmission rate--the number of bits sent each second--is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.
- ❖ Synchronization of bits. The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.
- ❖ Line configuration. The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.
- ❖ Physical topology. The physical topology defines how devices are connected to make a network. Devices can be connected by using a

mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies)

- ❖ **Transmission mode.** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex mode, two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time.

2-Data Link Layer

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer).

Other responsibilities of the data link layer include the following:

- ❖ **Framing.** The data link layer divides the stream of bits received from the network layer into manageable data units called frames
- ❖ **Physical addressing.** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.
- ❖ **Flow control.** If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- ❖ **Error control.** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

- ❖ Access control. When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

3-Network Layer

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination. If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery.

Other responsibilities of the network layer include the following:

- ❖ Logical addressing. The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver. We discuss logical addresses later in this chapter.
- ❖ Routing. When independent networks or links are connected to create internetworks (network of networks) or a large network, the connecting devices (called routers or switches) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

4-Transport Layer

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged

to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.

Other responsibilities of the transport layer include the following:

- ❖ Service-point addressing. Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.
- ❖ Segmentation and reassembly. A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.
- ❖ Connection control. The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- ❖ Flow control. Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- ❖ Error control. Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to-process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission

5- Session Layer

The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network dialog controller. It establishes, maintains, and synchronizes the interaction among communicating systems.

Specific responsibilities of the session layer include the following:

- ❖ Dialog control. The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either halfduplex (one way at a time) or full-duplex (two ways at a time) mode.
- ❖ Synchronization. The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent.

6- Presentation Layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.

Specific responsibilities of the presentation layer include the following:

- ❖ Translation. The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation

layer at the receiving machine changes the common format into its receiver-dependent format.

- ❖ Encryption. To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- ❖ Compression. Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

7-Application Layer

XAOO (The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

Specific services provided by the application layer include the following:

- ❖ Network virtual terminal. A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.
- ❖ File transfer, access, and management. This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.
- ❖ Mail services. This application provides the basis for e-mail forwarding and storage.

- ❖ Directory services. This application provides distributed database sources and access for global information about various objects and services.

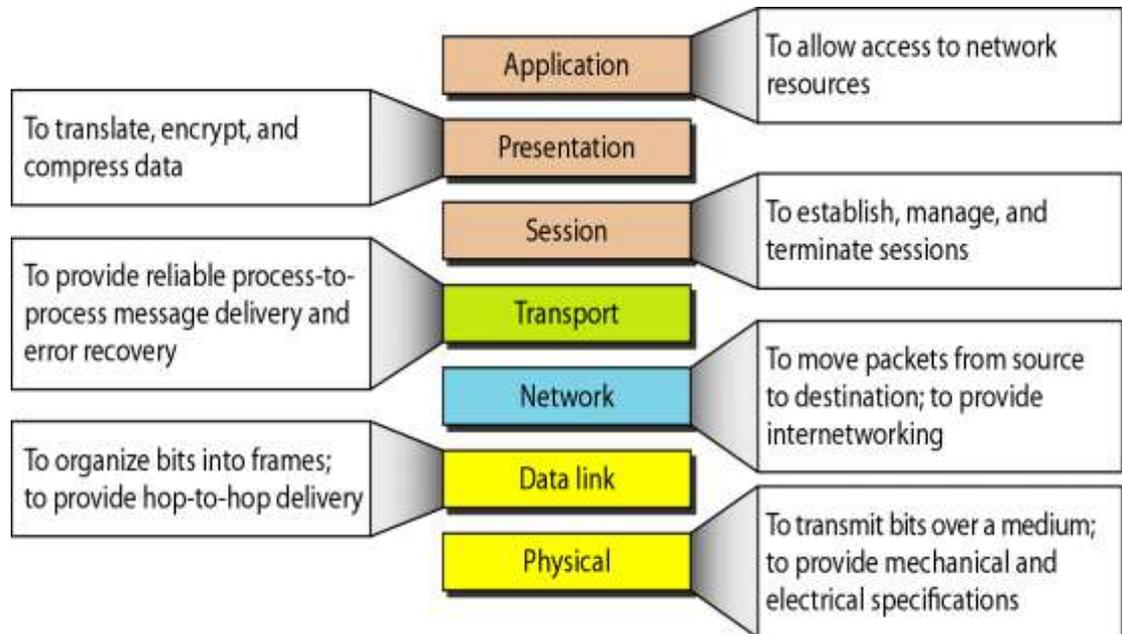


Figure 7 Summary of layers

TCP/IP PROTOCOL SUITE

The TCP/IP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application. However, when TCP/IP is compared to OSI, we can say that the host-to-network layer is equivalent to the combination of the physical and data link layers. The internet layer is equivalent to the network layer, and the application layer is roughly doing the job of the session, presentation, and application layers the transport layer in TCPIIP taking care of part of the duties of the session layer. So in this book, we assume that the TCPIIP protocol suite is made of five layers: physical, data link, network, transport, and application. The first four layers provide physical standards, network interfaces, internetworking, and transport functions that correspond to the first four layers of the OSI model. The three topmost layers in the OSI model, however, are represented in TCPIIP by a single layer called the application layer (see Figure 8)

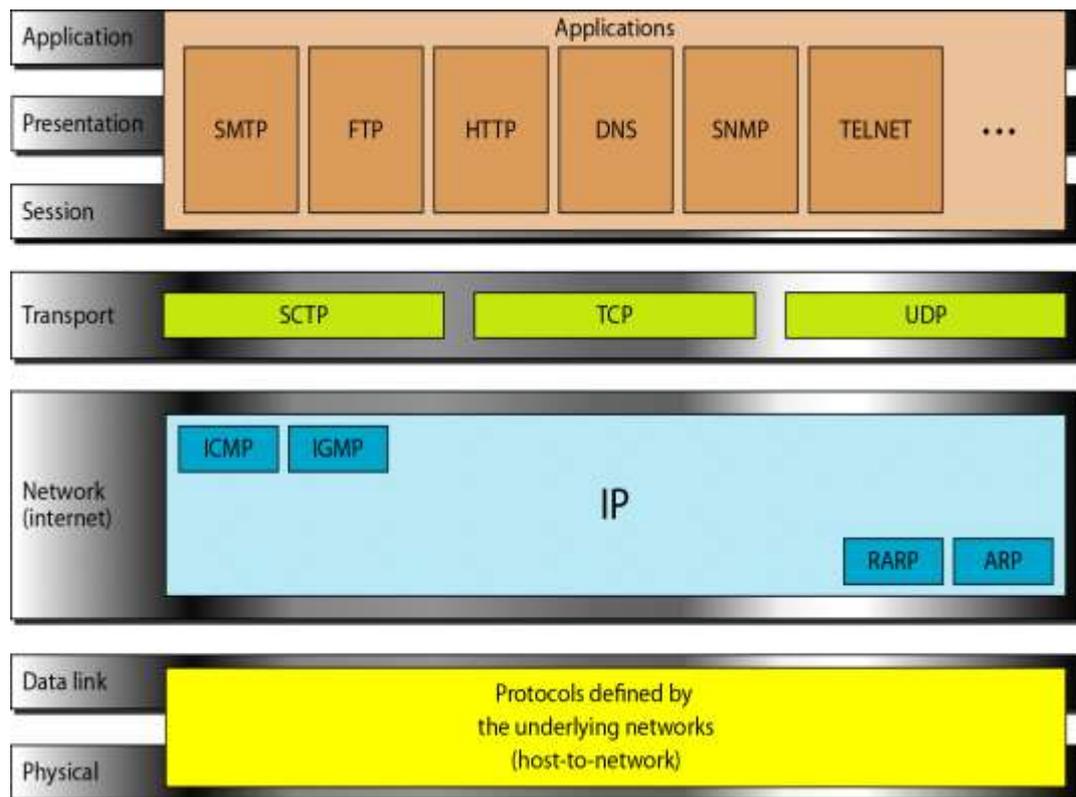


Figure (8) TCP/IP and OSI model

Data and Signals

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.

ANALOG AND DIGITAL

Both data and the signals that represent them can be either analog or digital in form.

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.

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 9 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.

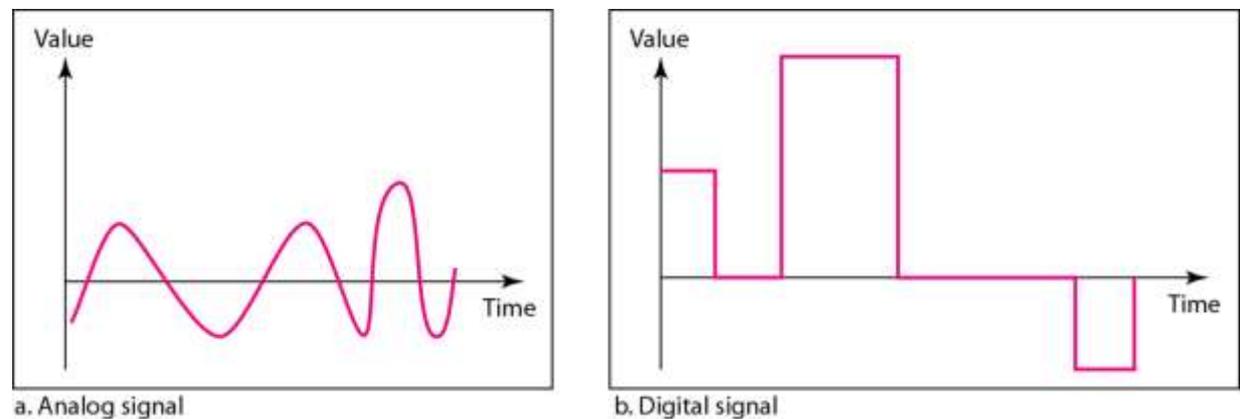


Figure (9) Comparison of analog and digital signals

Periodic and Nonperiodic Signals

Both analog and digital signals can take one of two forms: periodic or nonperiodic (sometimes refer to as aperiodic, because 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 متتابعه 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.

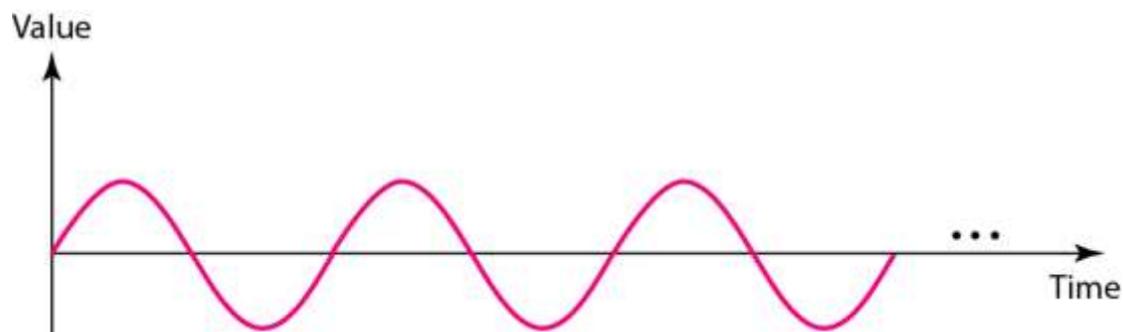
Both analog and digital signals can be periodic or nonperiodic. In data communications, we commonly use periodic analog signals (because they need less bandwidth,

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.

Sine Wave

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating متذبذب curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow. Figure 10 shows a sine wave. Each cycle consists of a single arc above the time axis followed by a single arc below it.

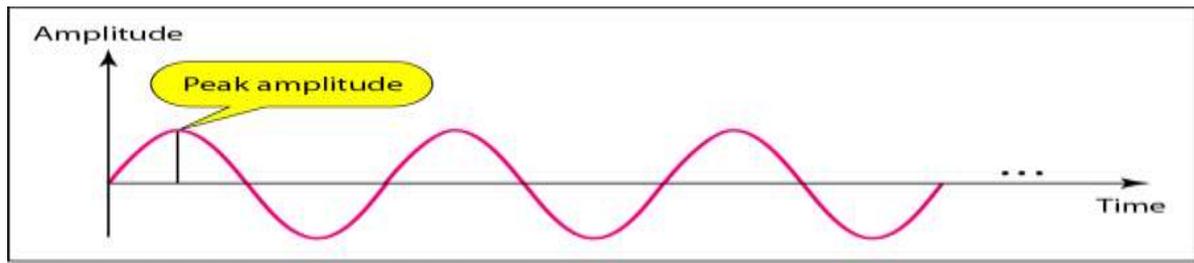


Figure(10) A sine wave

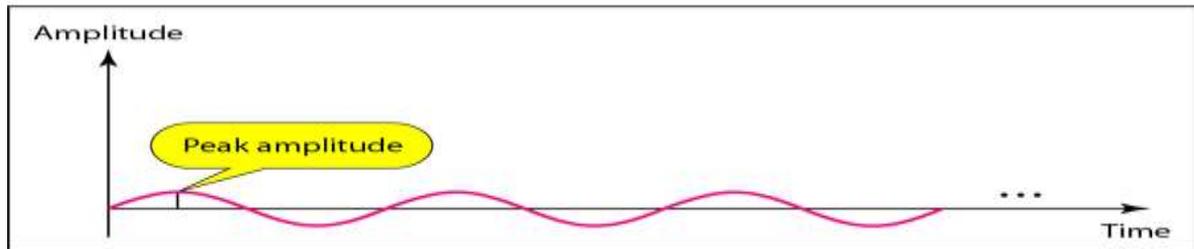
A sine wave can be represented by three parameters: the peak amplitude, the frequency and the phase. These three parameters fully describe a sine wave.

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 11 shows two signals and their peak amplitudes.



a. A signal with high peak amplitude



b. A signal with low peak amplitude

Figure(11) Two signals with the same phase and frequency, but different amplitudes

Example 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 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.

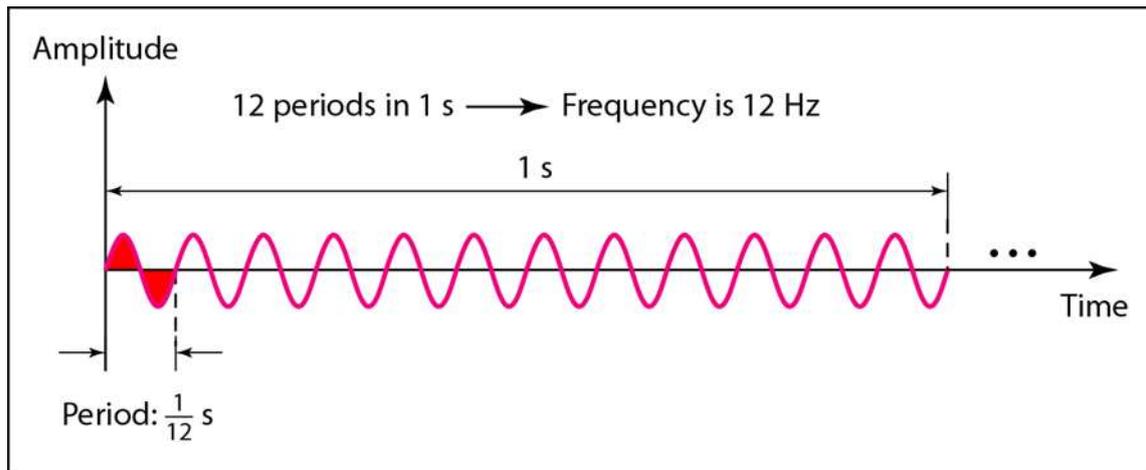
Period and Frequency

Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle.

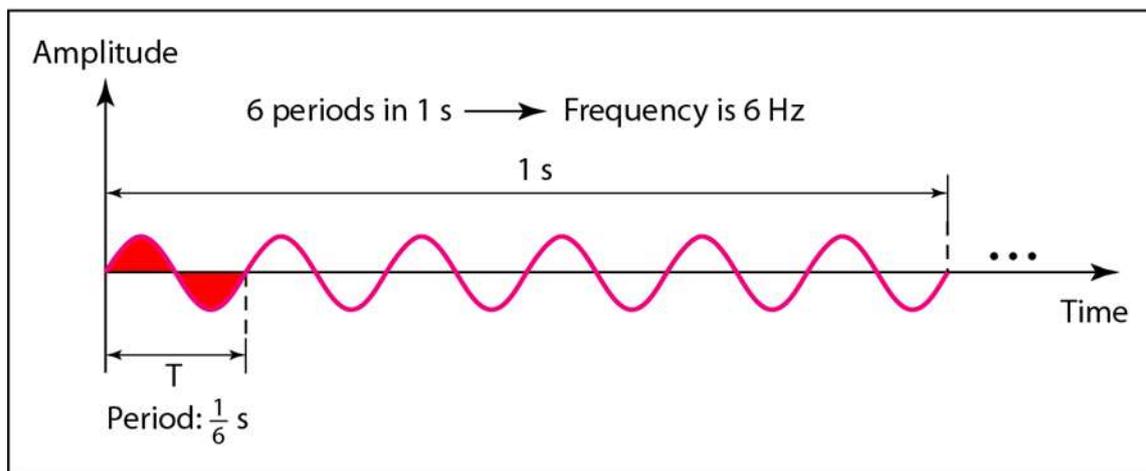
Frequency refers to the number of periods in 1 s. Note that period and frequency are just one characteristic defined in two ways. Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show.

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

Figure 12 shows two signals and their frequencies



a. A signal with a frequency of 12 Hz



b. A signal with a frequency of 6 Hz

Figure 13 Two signals with the same amplitude and phase, but different 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 1.

<i>Unit</i>	<i>Equivalent</i>	<i>Unit</i>	<i>Equivalent</i>
Seconds (s)	1 s	Hertz (Hz)	1 Hz
Milliseconds (ms)	10^{-3} s	Kilohertz (kHz)	10^3 Hz
Microseconds (μ s)	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

Table.1 Units of period and frequency

Example .3

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

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

Example 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 $10^6 \mu$ s). We make the following substitutions:.

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

Example 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 (الفوريه) that is, they fluctuate (تذبذب) continuously and predictably above and below a mean energy level. A 40-Hz signal has 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.

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.

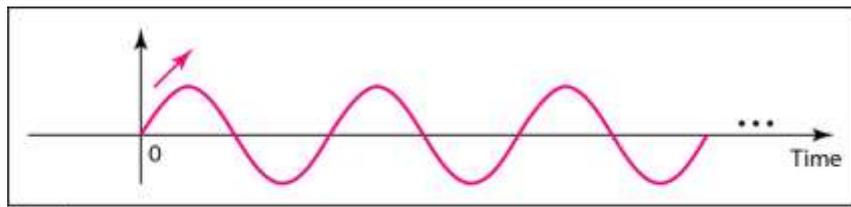
Two Extremes

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 aHz. 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).

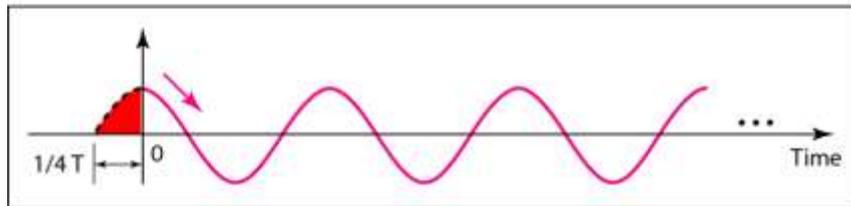
Phase

The term phase 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.

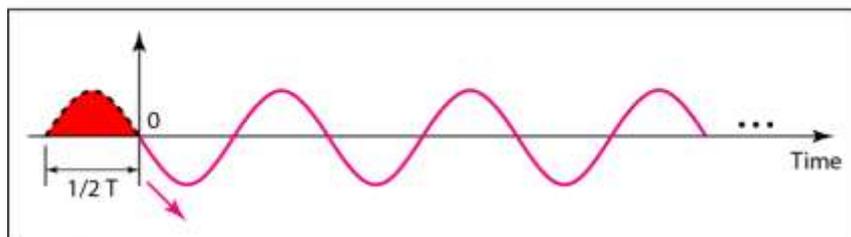
Phase is measured in degrees or radians [360° is 2π rad; 1° is $2\pi/360$ rad, and 1 rad is $360/(2\pi)$]. A phase shift of 360° corresponds to a shift of a complete period; a phase shift of 180° corresponds to a shift of one-half of a period; and a phase shift of 90° corresponds to a shift of one-quarter of a period (see Figure 14).



a. 0 degrees



b. 90 degrees



c. 180 degrees

Figure 14 Three sine waves with the same amplitude and frequency, but different phases

Looking at Figure 14, we can say that

1. A sine wave with a phase of 0° starts at time 0 with a zero amplitude. The amplitude is increasing.
2. A sine wave with a phase of 90° starts at time 0 with a peak amplitude. The amplitude is decreasing.
3. 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:

1. A sine wave with a phase of 0° is not shifted.
2. 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.
3. 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 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

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

Wavelength

Wavelength is another characteristic of a signal traveling through a transmission medium. Wavelength binds يربط the period or the frequency of a simple sine wave to the propagation speed of the medium (see Figure 15).

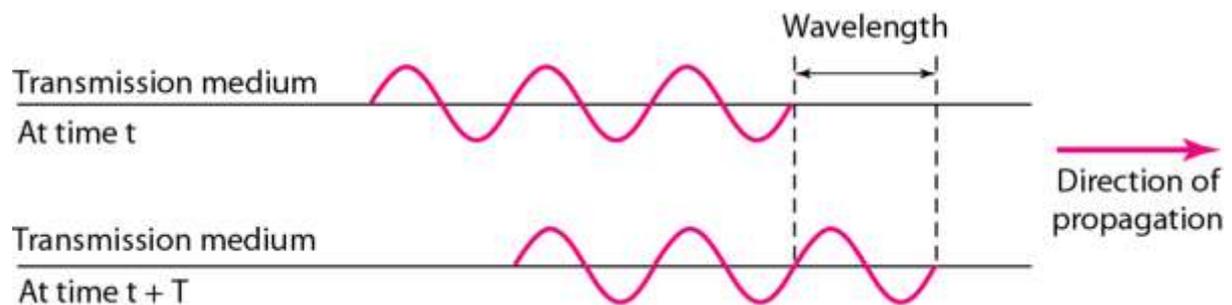


Figure 15 Wavelength and period

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

$$\text{Wavelength} = \text{propagation speed} \times \text{period} = \frac{\text{propagation speed}}{\text{Frequency}}$$

The propagation speed of electromagnetic signals depends on the medium and on 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

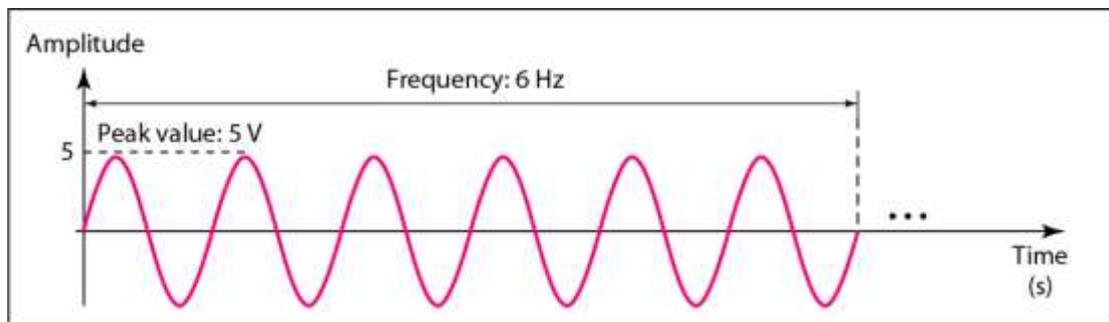
micrometers (microns) instead of meters. For example, the wavelength of red light (frequency = 4×10^{14}) in air is

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

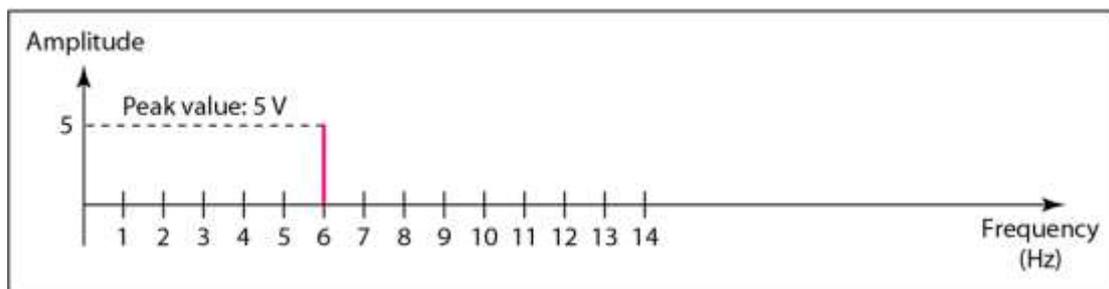
In a coaxial or fiber-optic cable, however, the wavelength is shorter ($0.5 \mu\text{m}$) because the propagation speed in the cable is decreased.

Time and Frequency Domains

A sine wave is comprehensively **شامله** 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 16 shows a signal in both the time and frequency domains.



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)

Figure 16 The time-domain and frequency-domain plots of a sine wave

It is obvious that the frequency domain is easy to plot and conveys **ينقل** 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.

Example 7

The frequency domain is more compact and useful when we are dealing with more than one sine wave. For example, Figure 17 shows three sine waves, each with different amplitude and frequency. All can be represented by three spikes in the frequency domain.

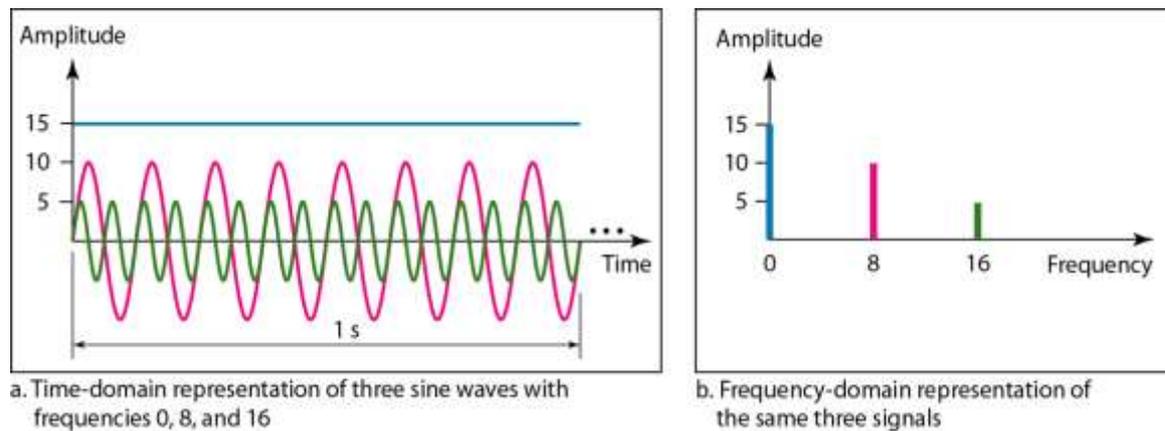


Figure 17 The time domain and frequency domain of three sine waves

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.

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 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.

Example 8

Figure 18 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.

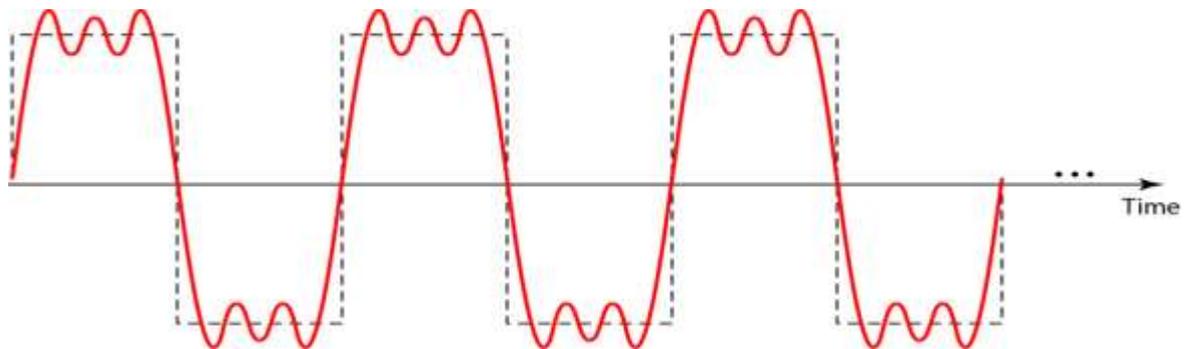
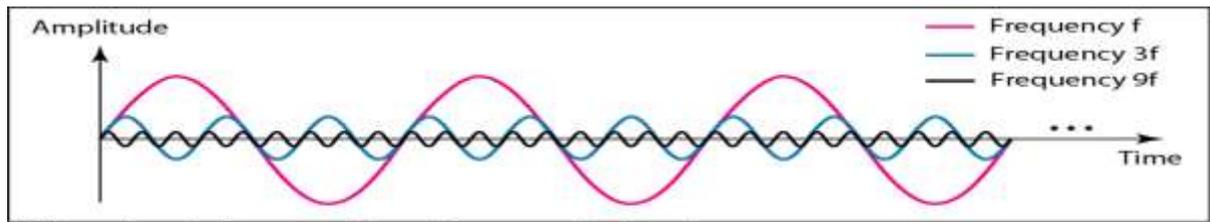
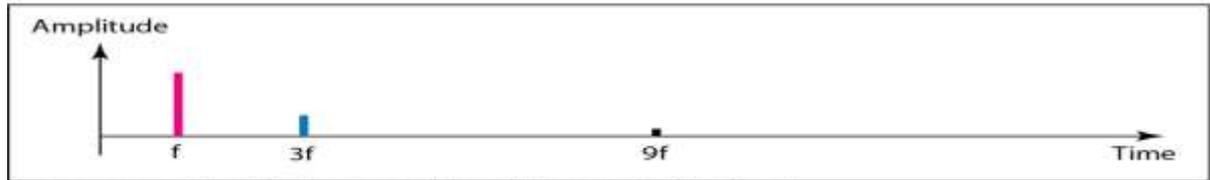


Figure 18 A composite periodic signal

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 19 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 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.



a. Time-domain decomposition of a composite signal

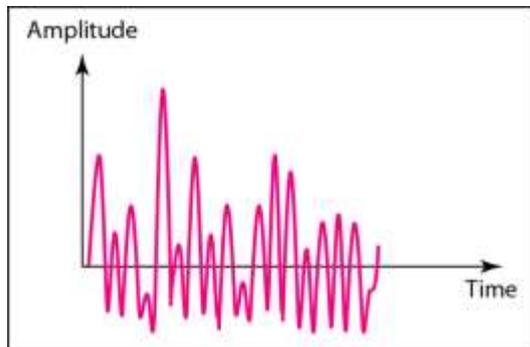


b. Frequency-domain decomposition of the composite signal

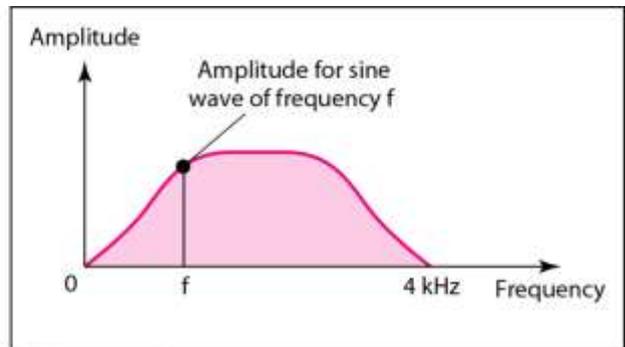
Figure 19 Decomposition of a composite periodic signal in the time and frequency domains

Example 9

Figure 20 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.



a. Time domain



b. Frequency domain

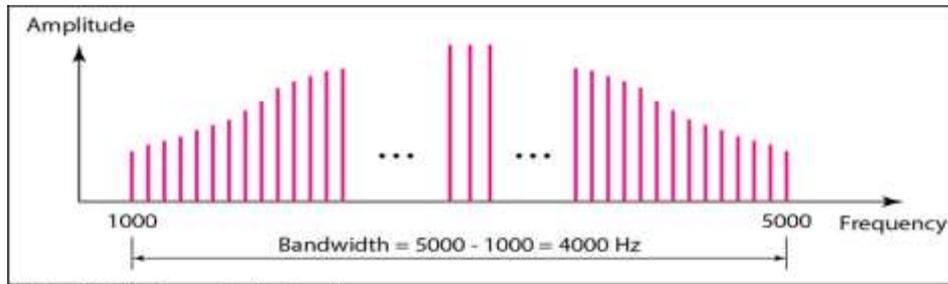
Figure 20 The time and frequency domains of a nonperiodic signal

Bandwidth

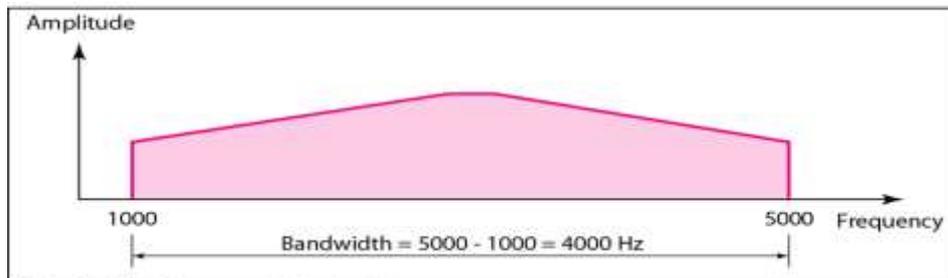
The range of frequencies contained in a composite signal is its bandwidth. The bandwidth is normally a difference between two numbers. For example, if a composite signal contains frequencies between 1000 and 5000, its bandwidth is $5000 - 1000$, or 4000.

Figure 21 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.



a. Bandwidth of a periodic signal



b. Bandwidth of a nonperiodic signal

Figure 21 The bandwidth of periodic and nonperiodic composite signals

Example 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 22).

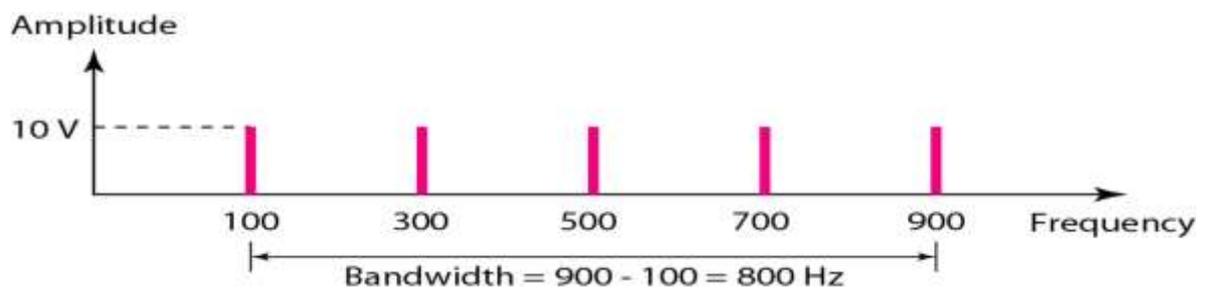
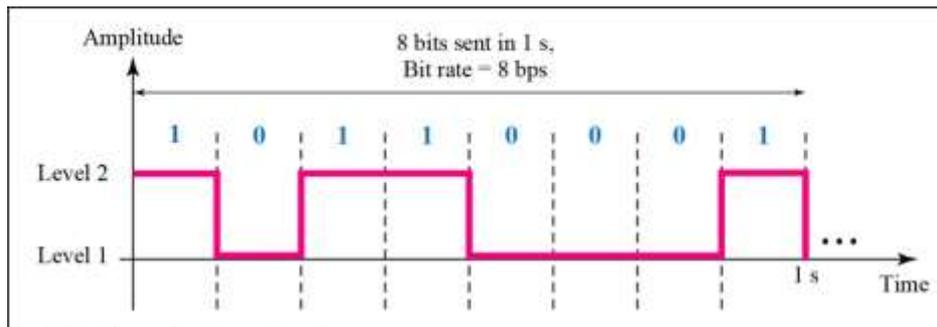


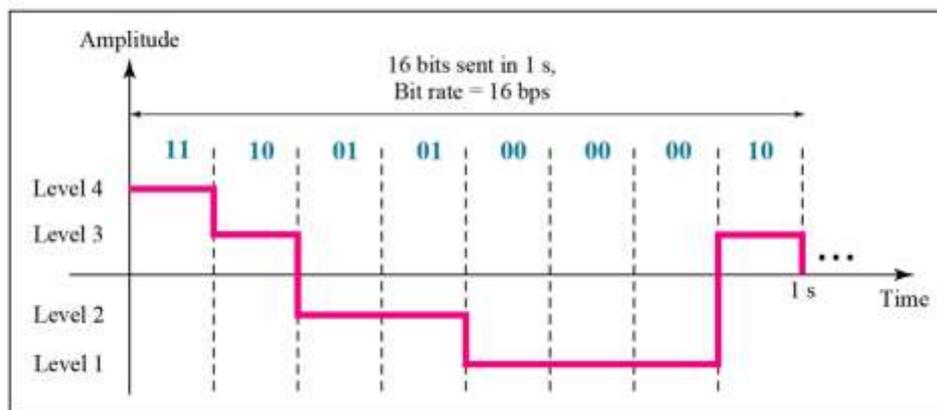
Figure 22 The bandwidth for Example 10

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 3.16 shows two signals, one with two levels and the other with four.



a. A digital signal with two levels



b. A digital signal with four levels

Figure 16 Two digital signals: one with two signal levels and the other with four signal levels

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.

Example 16

A digital signal has eight levels. How many bits are needed per level? We calculate the number of bits from the formula

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

Each signal level is represented by 3 bits.

Example 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.

Bit Rate

Most digital signals are nonperiodic, and thus period and frequency are not appropriate. 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 persecond (bps). Figure 16 shows the bit rate for two signals.

Example 18

Assume we need to download text documents at the rate of 100 pages per minute. 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

$$100 \times 24 \times 80 \times 8 = 1,636,000 \text{ bps} = 1.636 \text{ Mbps}$$

Example 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?

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

Example 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. There are 1920 by 1080 pixels per screen, and the screen is renewed 30 times per second. Twenty-four bits represents one color pixel.

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

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

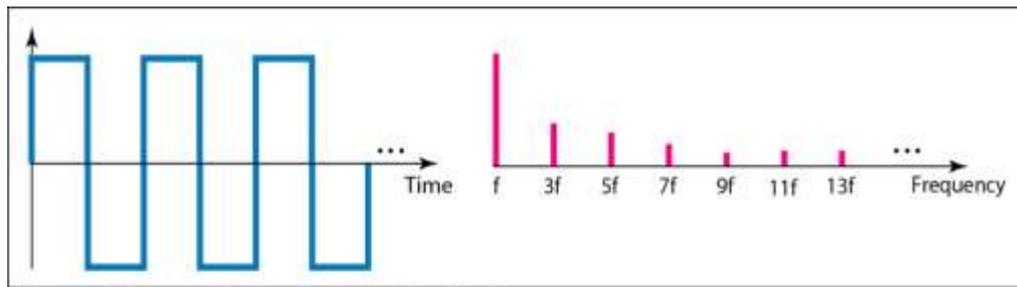
Bit Length

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.

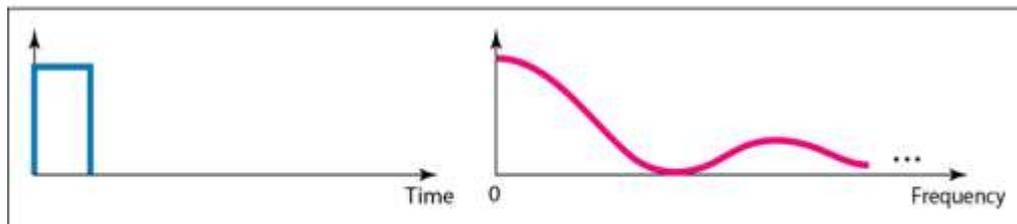
Bit length = propagation speed x bit duration

Digital Signal as a Composite Analog Signal

Based on Fourier analysis, 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.17 shows a periodic and a nonperiodic digital signal and their bandwidths.



a. Time and frequency domains of periodic digital signal



b. Time and frequency domains of nonperiodic digital signal

Figure 17 the time and frequency domains of periodic and nonperiodic digital signals

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

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).

Baseband Transmission

Baseband transmission means sending a digital signal over a channel without changing the digital signal to an analog signal. Figure.18 shows baseband transmission.

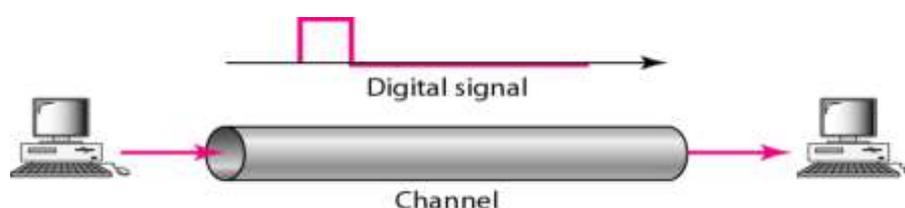


Figure 18 Baseband transmission

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 19 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.

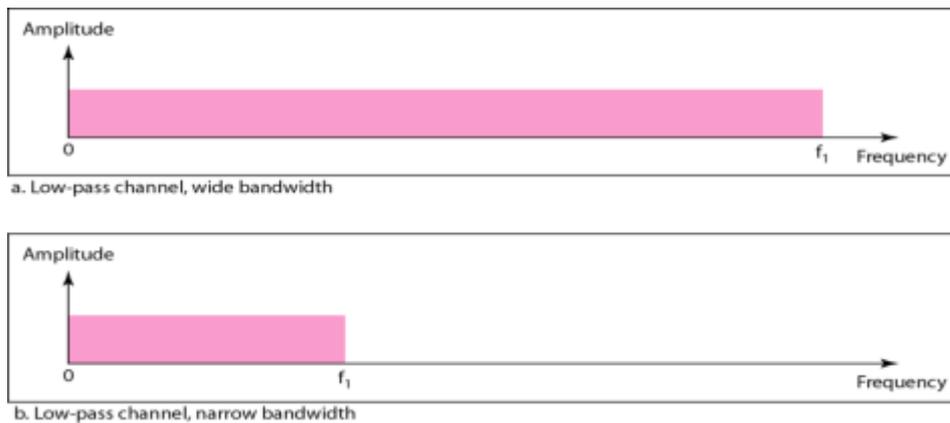


Figure 19 Bandwidths of two low-pass channels

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 cable or fiber optic, with a very wide bandwidth, two stations can communicate by using digital signals with very good accuracy, as shown in Figure 20. Note that f_1 is close to zero, and f_2 is very high.

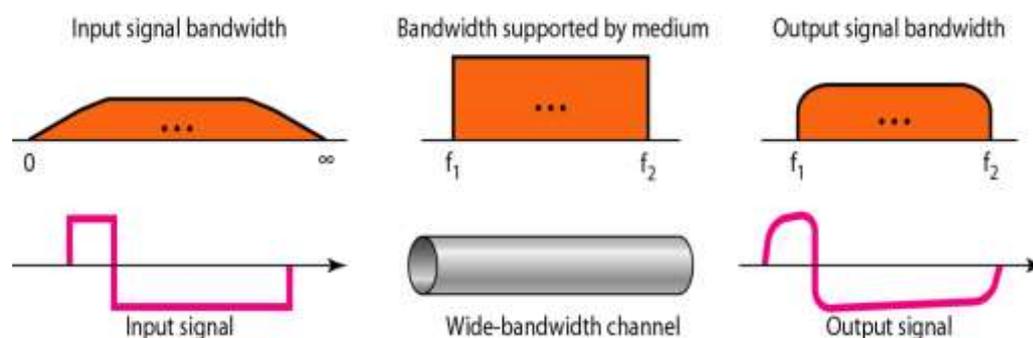


Figure 20 Baseband transmission using a dedicated medium

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.

Example 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.

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.21 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 $f = N/4$ 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**

$$\text{Bandwidth} = (N/2) - 0 = (N/2)$$

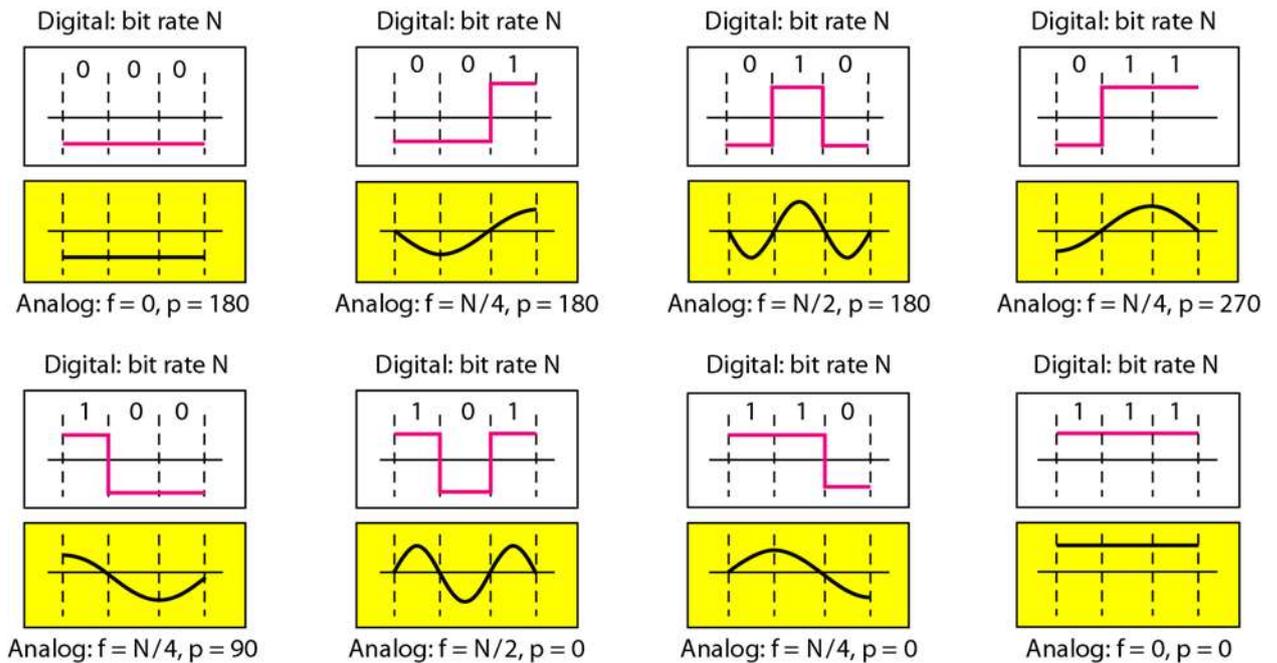
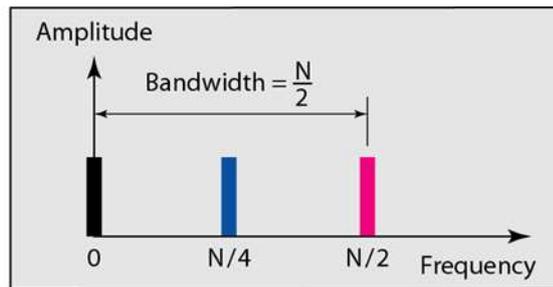


Figure 21 Rough approximation of a digital signal using the first harmonic for worst case

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 23 shows a bandpass channel.



Figure 3.23 Bandwidth of a bandpass channel

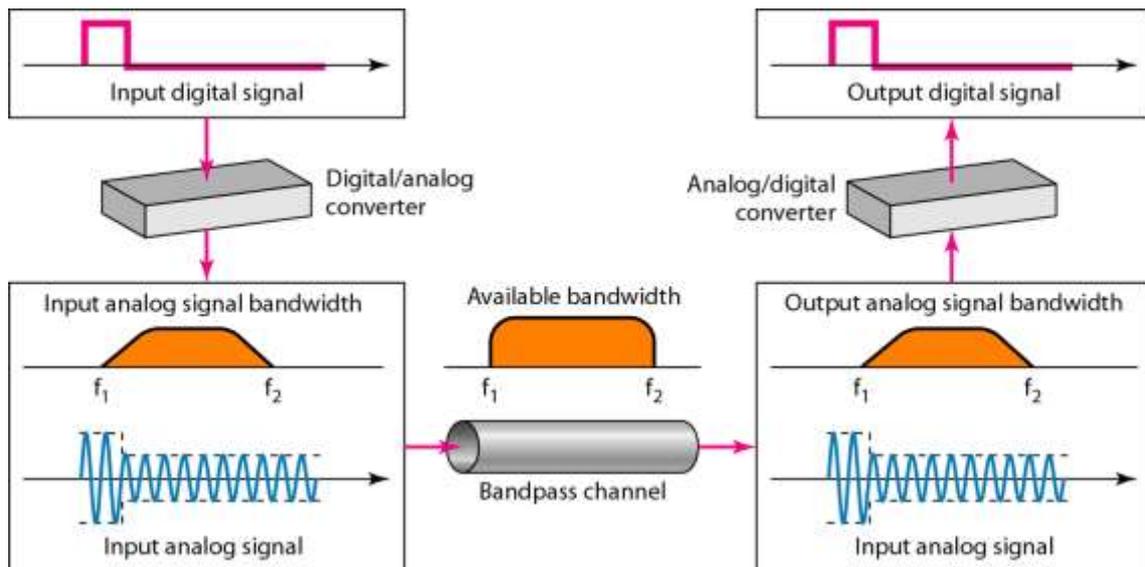


Figure 24 Modulation of a digital signal for transmission on a band pass channel

Note

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

Table 2 Bandwidth requirements

<i>Bit Rate</i>	<i>Harmonic 1</i>	<i>Harmonics 1, 3</i>	<i>Harmonics 1, 3, 5</i>
$n = 1 \text{ kbps}$	$B = 500 \text{ Hz}$	$B = 1.5 \text{ kHz}$	$B = 2.5 \text{ kHz}$
$n = 10 \text{ kbps}$	$B = 5 \text{ kHz}$	$B = 15 \text{ kHz}$	$B = 25 \text{ kHz}$
$n = 100 \text{ kbps}$	$B = 50 \text{ kHz}$	$B = 150 \text{ kHz}$	$B = 250 \text{ kHz}$

Example .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, is $B = \text{bit rate} / 2$, or 500 kHz.

- b. A better solution is to use the first and the third harmonics with $B = 3 \times 500 \text{ kHz} = 1.5 \text{ MHz}$.
- c. Still a better solution is to use the first, third, and fifth harmonics with $B = 5 \times 500 \text{ kHz} = 2.5 \text{ MHz}$.

Example 22

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.

Example 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 are designed to carry voice with a limited bandwidth. The channel is considered a bandpass channel. We 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.

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. We need to convert the digitized voice to a composite analog signal before sending.

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.

1. DIGITAL-TO-DIGITAL CONVERSION

In previous lectures, 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.

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. 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.

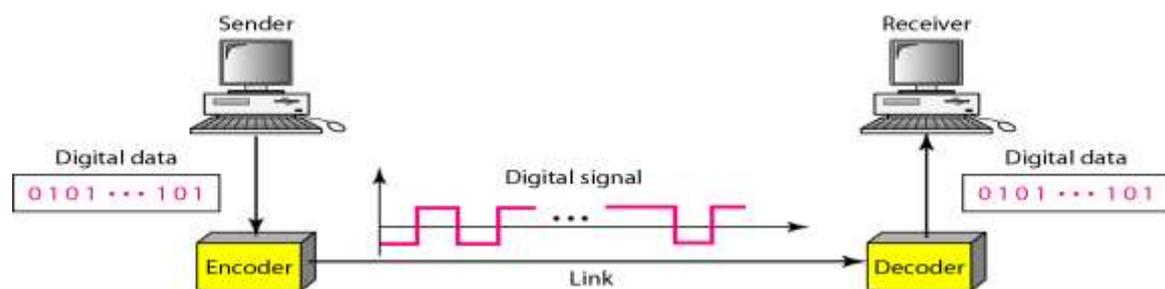


Figure .1 Line coding and decoding

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 (time wise) 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 .

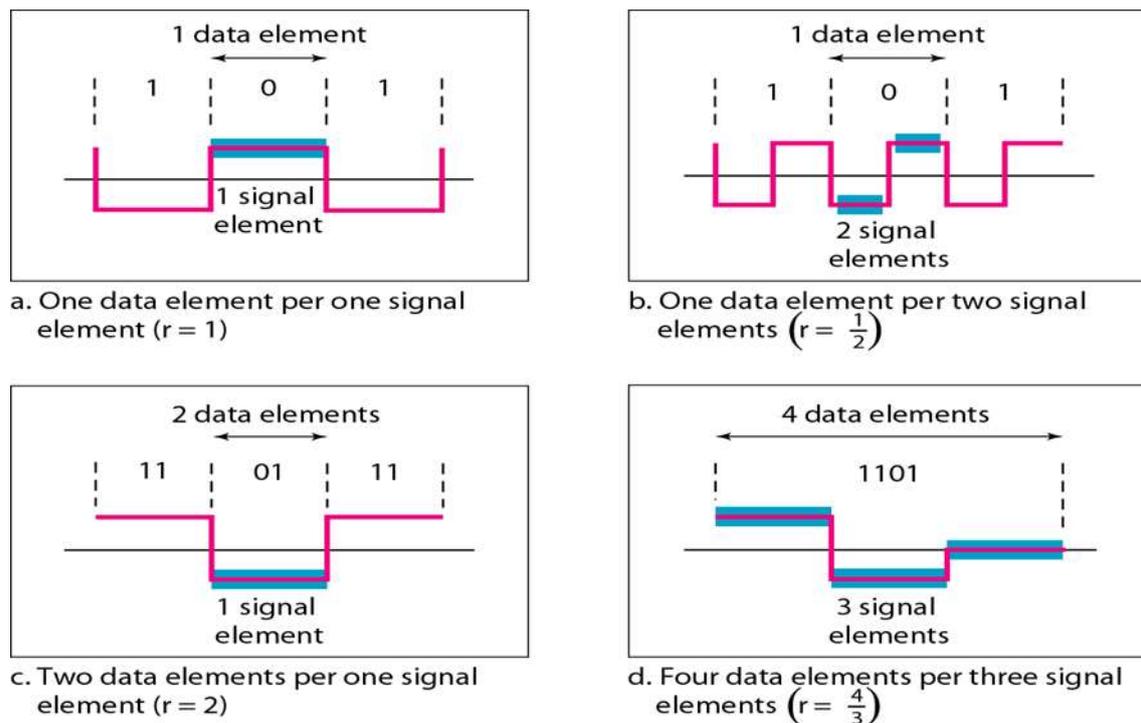


Figure 2 Signal elements versus data element

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 ($r = 2$). Finally, in part **d**, a group of 4 bits is being carried by a group of three signal elements ($r = \frac{4}{3}$). 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 ($r = \frac{1}{2}$).

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 and signal rate (bit rate and baud rate). 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

$$S = c \times N \times \frac{1}{r} \quad \text{baud}$$

Where N is the data rate (bps); c is the case factor, which varies for each case; S is the number of signal elements; 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

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

Bandwidth we discussed in previous lectures 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.

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 affects the traffic, not the number of people being carried. 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

$$B_{\min} = c \times N \times \frac{1}{r}$$

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

$$N_{\max} = \frac{1}{c} \times B \times r$$

Example 4.2

The maximum data rate of a channel 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

$$N_{\max} = \frac{1}{c} \times B \times r = 2 \times B \times \log_2 L$$

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. 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). 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 shorter bit duration. The sender sends 10110001, while the receiver receives 110111000011.

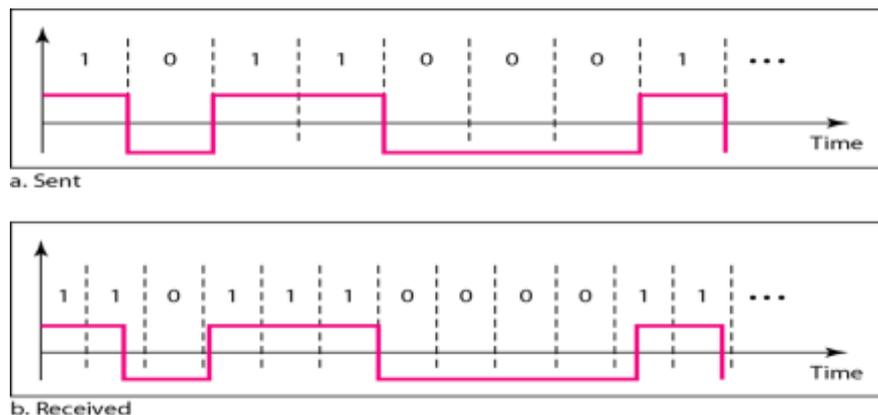


Figure 3 Effect of lack of synchronization

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.

Built-in Error Detection It is desirable to have a built-in error-detecting capability in the generated code to detect some of or all 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.

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.

2. ANALOG-TO-DIGITAL CONVERSION

Sometimes, we have an analog signal such as one created by a microphone or camera. We have seen previously 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**.

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.

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

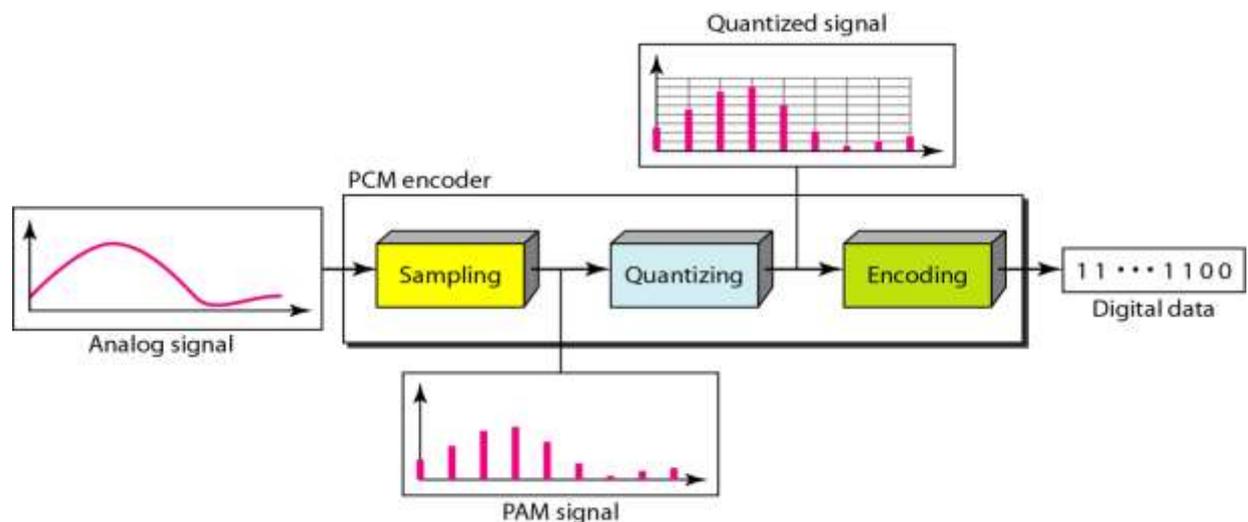
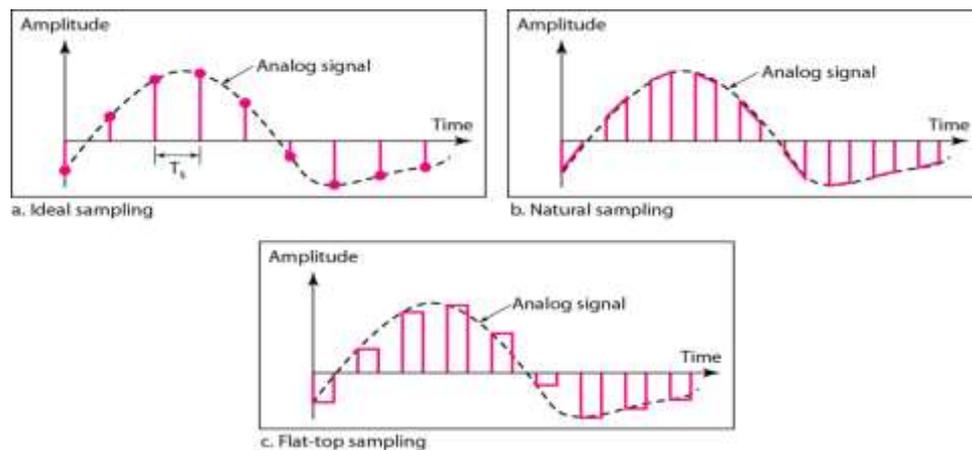


Figure 4 Components of PCM encoder

Sampling

The first step in PCM is sampling. The analog signal is sampled every T_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 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.



Three different sampling methods for PCM

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.

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 nb -bit code word. 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 $nb = \log_2 L$. In our example L is 8 and nb is therefore 3. The bit rate can be found from the formula

$$\text{Bit rate} = \text{sampling rate} \times \text{number of bits per sample} = f_s \times nb$$

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:

$$\text{Sampling rate} = 4000 \times 2 = 8000 \text{ samples/s}$$

$$\text{Bit rate} = 8000 \times 8 = 64,000 \text{ bps} = 64 \text{ kbps}$$

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.

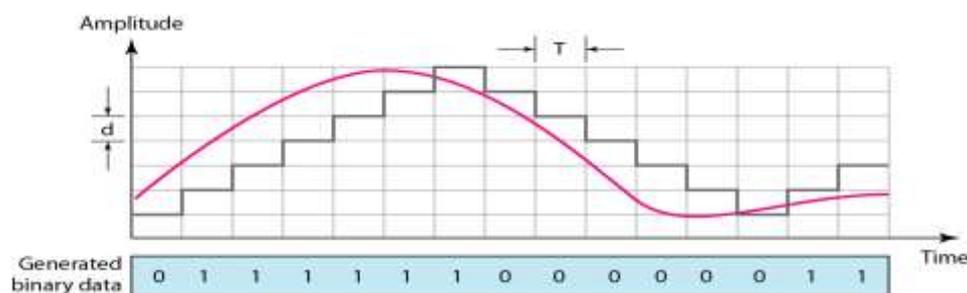


Figure 4.28 the process of delta modulation

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.

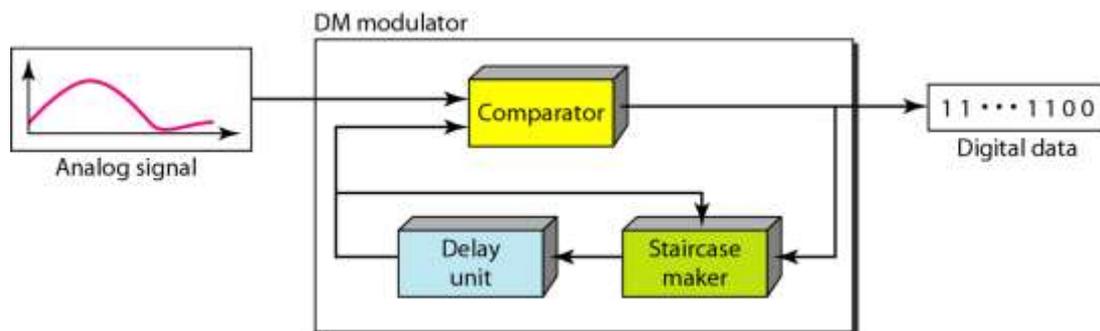


Figure4.29 /Delta modulation components

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 0 up; if the next bit is 0, it moves it 0 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.

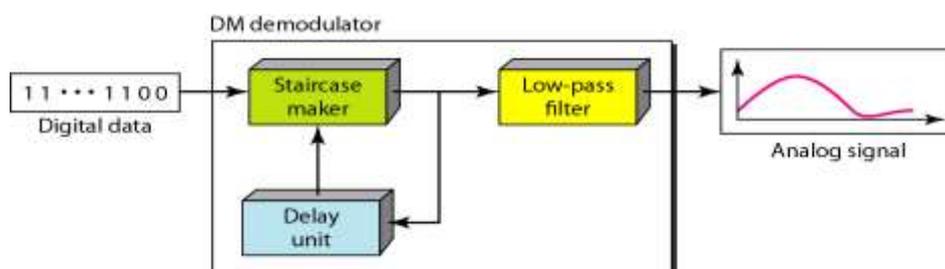


Figure 4.30 Delta demodulation components

TRANSMISSION MODES

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.

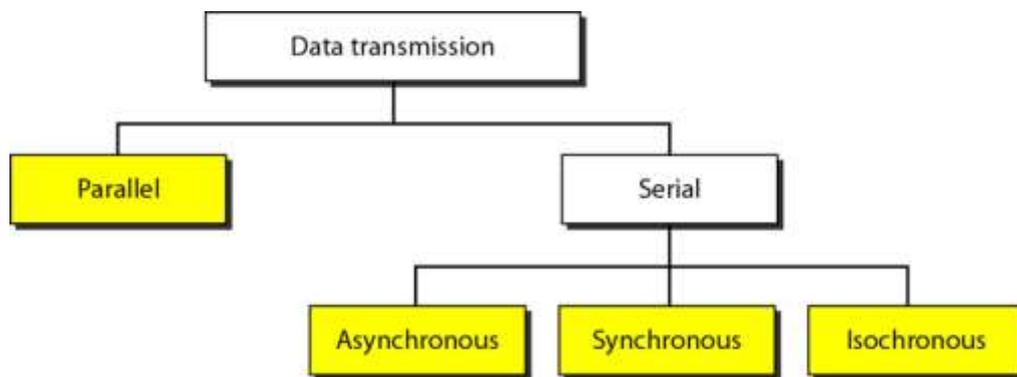


Figure 4.31 Data transmission and modes

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.

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.

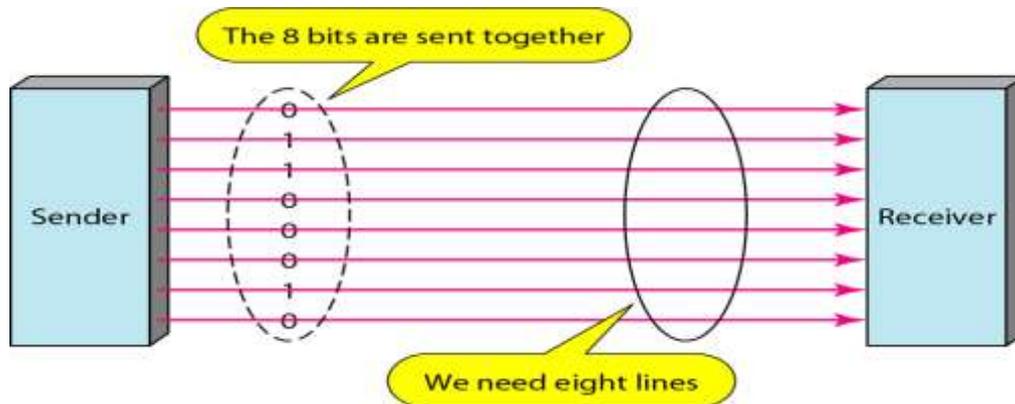


Figure 4.32 Parallel transmission

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). 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.

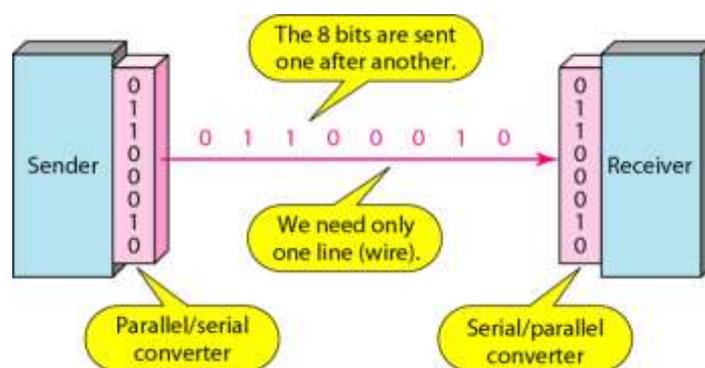


Figure 4.33 Serial transmission

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.

Note

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 each byte. Asynchronous here means “asynchronous at the byte level,” but the bits are still synchronized; their durations are the same.

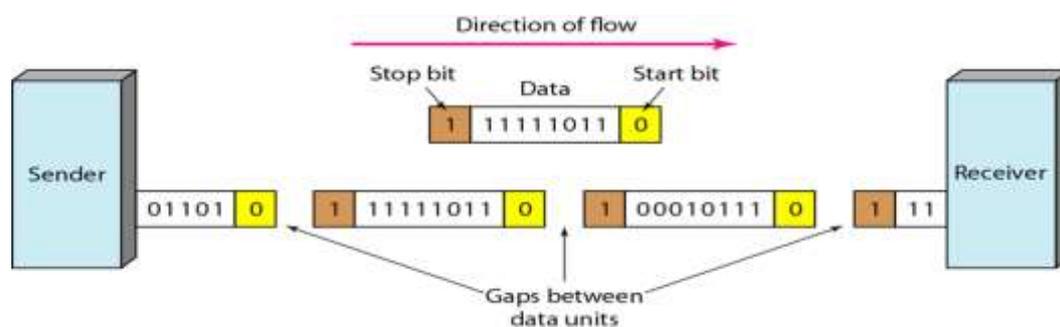


Figure 4.34 Asynchronous transmission

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.

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.

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.

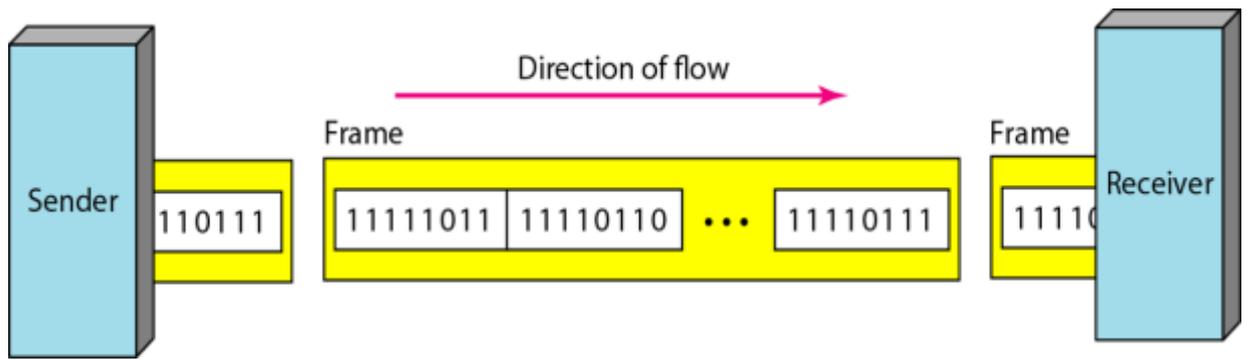


Figure 4.35 Synchronous transmission

متساوي الزمن 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.

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. As discussed in previous lectures, a sine wave is defined by three characteristics:

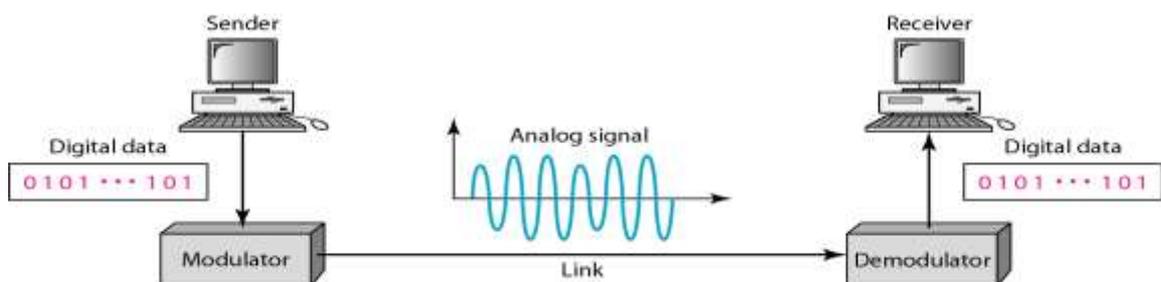


Figure 5.1 Digital-to-analog conversion

Amplitude, Frequency and phase. When we vary anyone 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).

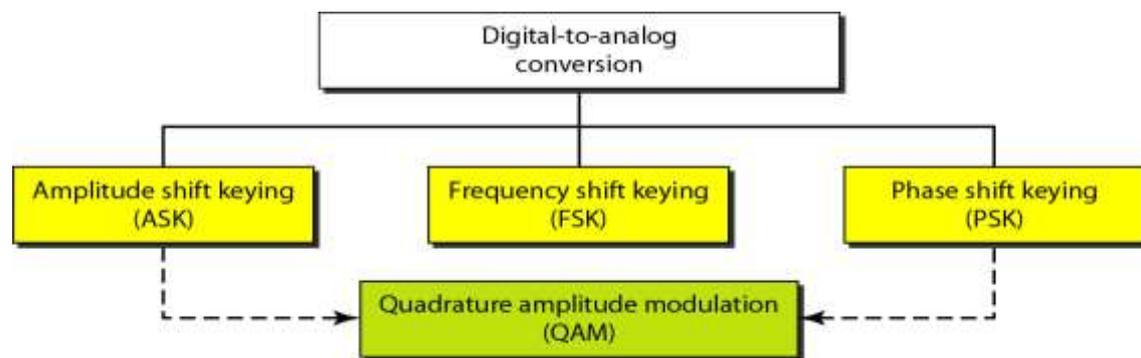


Figure 5.2 *Types of digital-to-analog conversion*

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

$$S = N \times 1/r \text{ baud}$$

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 type of signal element, not the level. The same nomenclature is used to simplify the comparisons.

Note/Bit rate, N , is the number of bits per second (bps). Baud rate is the number of signal elements per second (bauds). In the analog transmission of digital data, the signal or baud rate is less than or equal to the bit rate.

$$S = N \times 1/r \text{ bauds}$$

Where r is the number of data bits per signal element.

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

$$S = N \times \frac{1}{r} \quad \text{or} \quad N = S \times r = 1000 \times 4 = 4000 \text{ bps}$$

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 find first the value of r and then the value of L .

$$\begin{array}{l} S = N \times \frac{1}{r} \quad \longrightarrow \quad r = \frac{N}{S} = \frac{8000}{1000} = 8 \text{ bits/ baud} \\ r = \log_2 L \quad \longrightarrow \quad L = 2^r = 2^8 = 256 \end{array}$$

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).

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 ASKS.

- ASK is implemented by changing the amplitude of a carrier signal to reflect amplitude levels in the digital signal.
- For example: a digital “1” could not affect the signal, whereas a digital “0” would, by making it zero.
- The line encoding will determine the values of the analog waveform to reflect the digital data being carried.
- The bandwidth B of ASK is proportional to the signal rate S .

$$B = (1+d)S$$

- “ d ” is due to modulation and filtering, lies between 0 and 1.

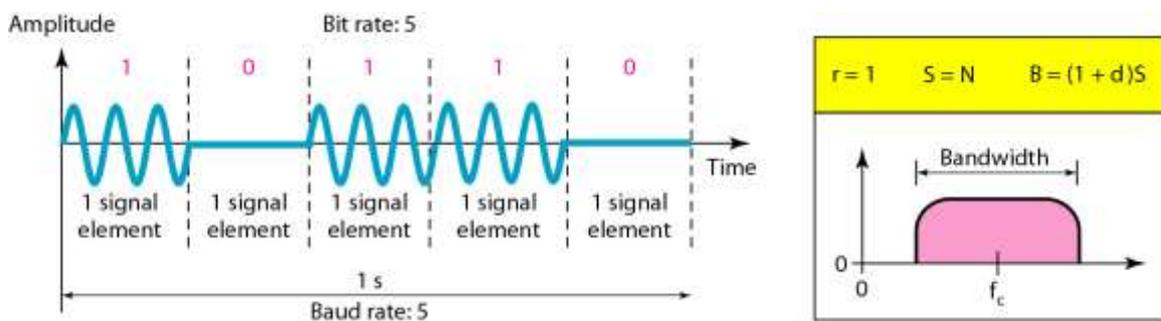


Figure 5.3 Binary amplitude shift keying

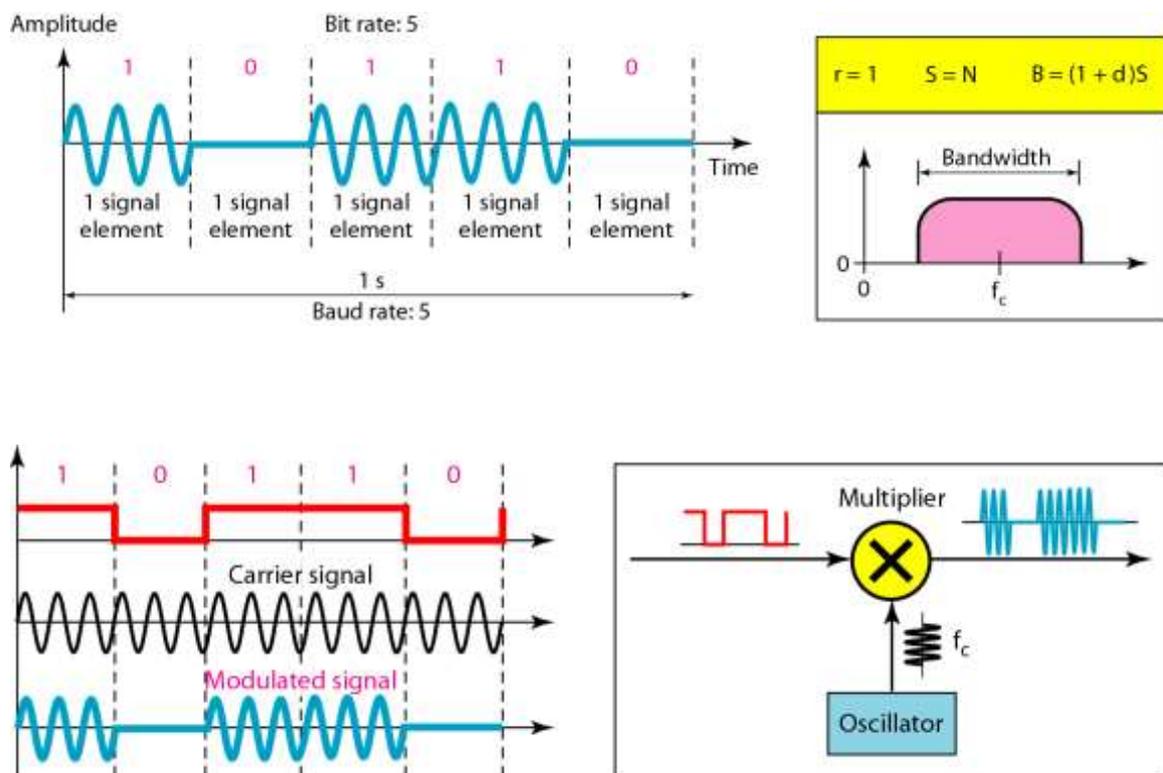
Bandwidth for ASK Figure 5.3 also shows the bandwidth for ASK. Although the carrier signal is only one simple sine wave, the process of modulation produces a nonperiodic composite signal. This signal, as was discussed in Chapter 3, has a continuous set of frequencies. As we expect, **the bandwidth is 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. $B = (1+d)S$

The formula shows that the required bandwidth has a minimum value of 5 and a maximum value of 25. 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.

If digital data are presented as a unipolar NRZ digital signal with a high voltage of 1 V and a low voltage of 0 V, the implementation can be 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 $d = 1$?

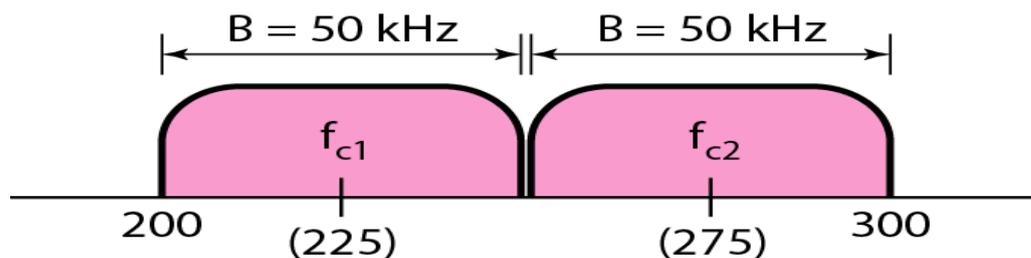
Solution

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 $r = 1$).

$$B = (1 + d) \times S = 2 \times N \times \frac{1}{r} = 2 \times N = 100 \text{ kHz} \quad \rightarrow \quad N = 50 \text{ kbps}$$

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.



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).

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.

Bandwidth of FSK

- If the difference between the two frequencies (f_1 and f_2) is $2\Delta f$, then the required BW B will be:

$$B = (1+d) \times S + 2\Delta f$$

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

$$B = (1 + d) \times S + 2\Delta f = 100 \quad \rightarrow \quad 2S = 50 \text{ kHz} \quad S = 25 \text{ kbaud} \quad N = 25 \text{ kbps}$$

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.

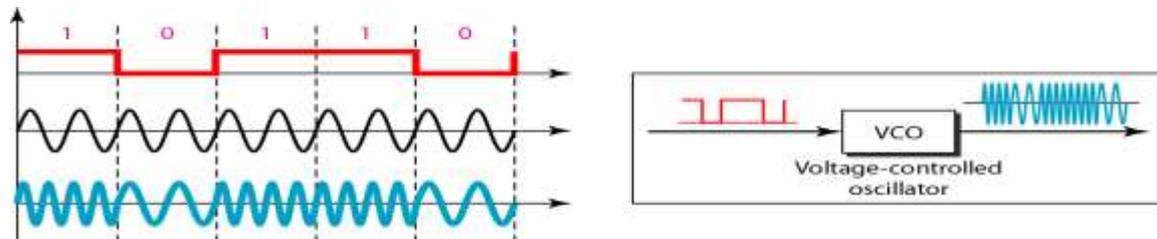


Figure 5.7 Implementation of BFSK

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 with $d = 0$ is

$$B = (1+d) \times S + (L-1)/2\Delta f = L \times S$$

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 $S = 3 \text{ Mbps}/3 = 1 \text{ Mbaud}$. This means that the carrier frequencies must be 1 MHz apart ($2\Delta f = 1 \text{ MHz}$). The bandwidth is $B = 8 \times 1 \text{ M} = 8 \text{ M}$. Figure 5.8 shows the allocation of frequencies and bandwidth.

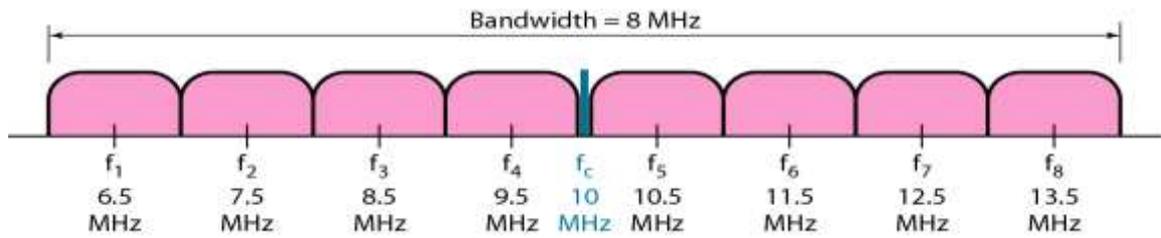


Figure 5.8 Bandwidth of MFSK used in Example 5.6

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, 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 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.

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. $B = (1+d) \times S$

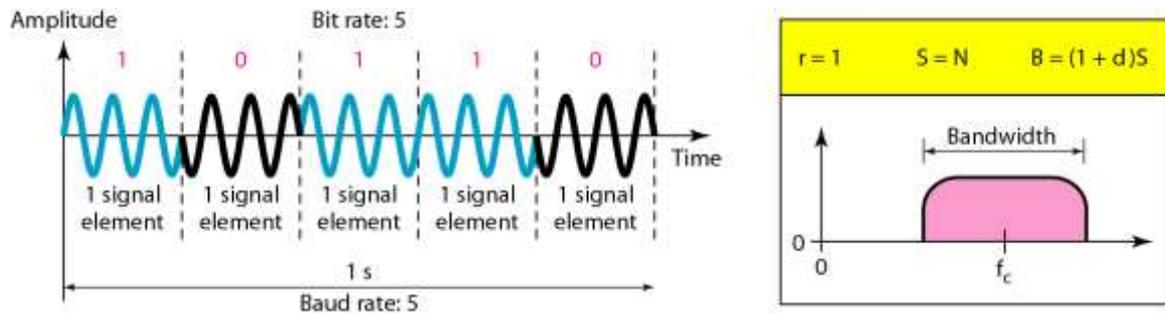


Figure 5.9 Binary phase shift keying

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°.

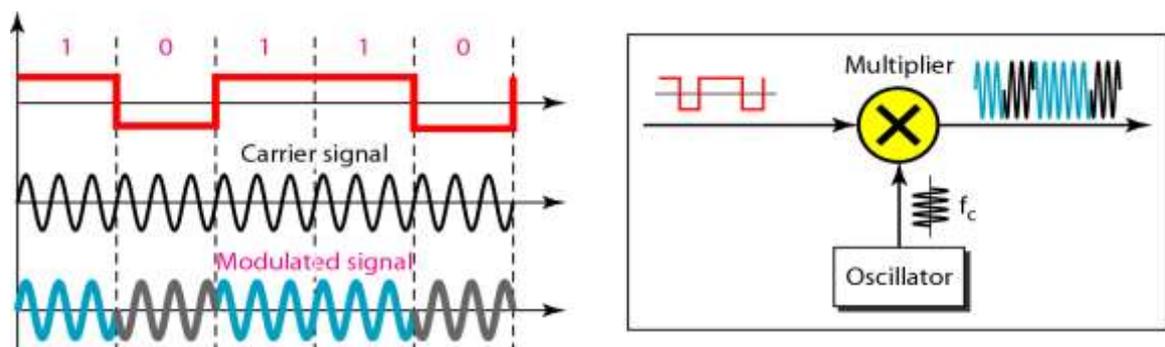
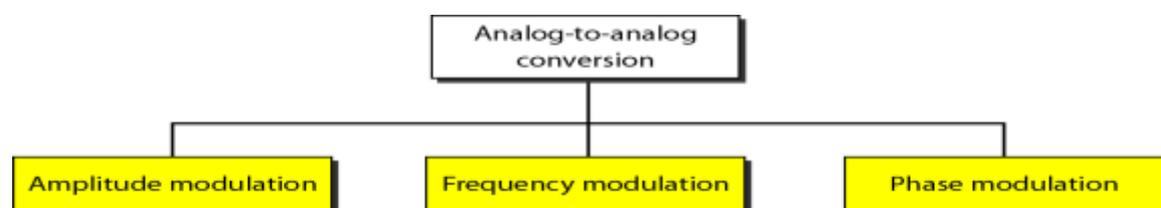


Figure 5.10 Implementation of BASK

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, 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. each to a different range



Amplitude Modulation

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.

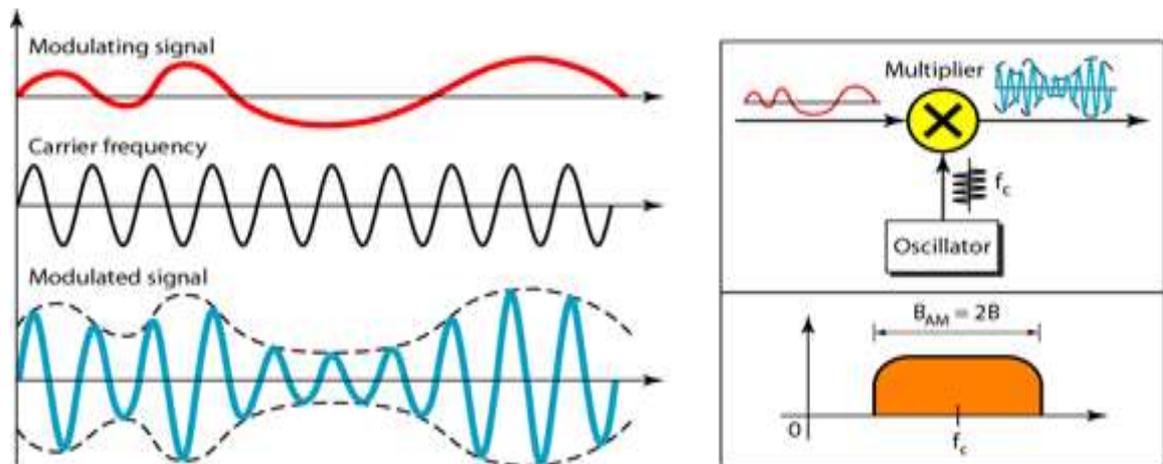


Figure 5.16 Amplitude *modulation*

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.

Frequency Modulation

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.

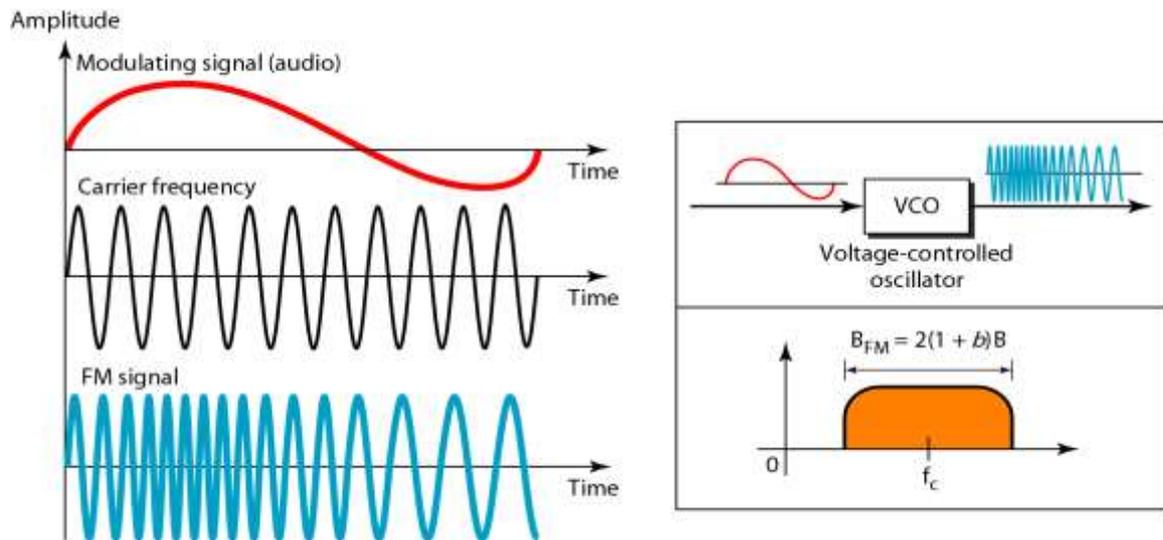
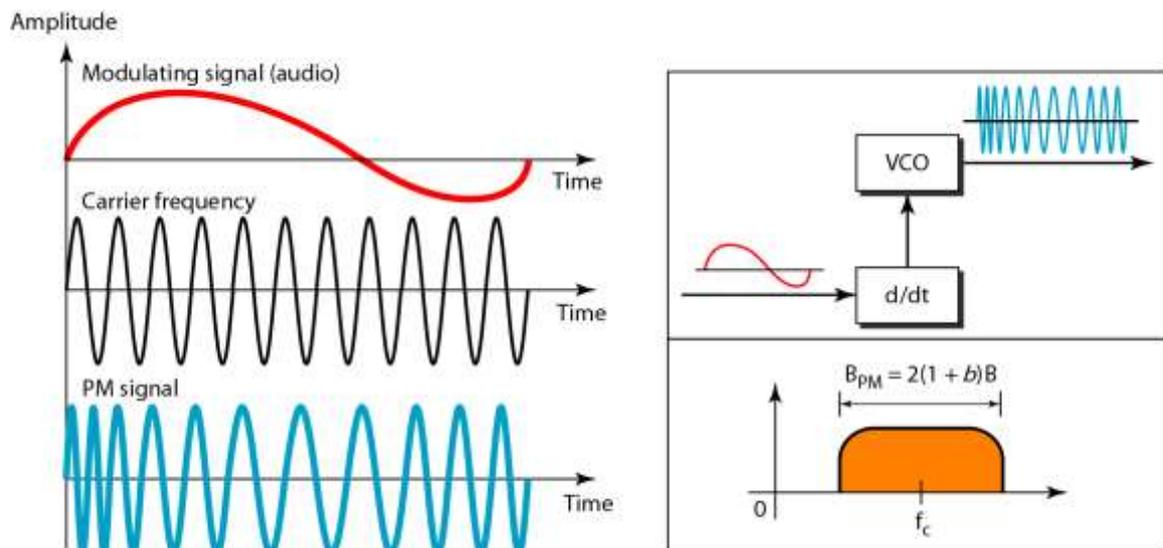


Figure 5.18 *Frequency modulation*

Phase modulation

- The modulating signal only changes the phase of the carrier signal.
- The phase change manifests itself as a frequency change but the instantaneous frequency change is proportional to the derivative of the amplitude.
- The bandwidth is higher than for AM.



MULTIPLEXING الارسال المتعدد

Whenever the bandwidth of a medium linking two devices is greater than the bandwidth needs of the devices, the link can be shared. Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.

today's technology includes high-bandwidth media such as optical fiber and terrestrial and satellite microwaves. Each has a bandwidth far in excess of that needed for the average transmission signal.

Frequency-Division Multiplexing

Frequency-division multiplexing (FDM) is an **analog technique** that can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted.

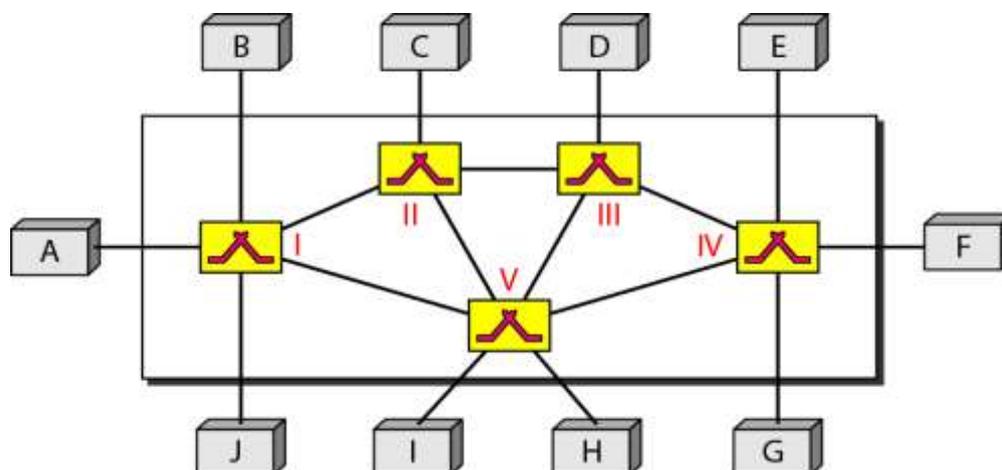
Synchronous Time-Division Multiplexing

Time-division multiplexing (TDM) is a **digital process** that allows several connections to share the high bandwidth of a link. Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link.

Switching

A network is a set of connected devices. Whenever we have multiple devices, we have the problem of how to connect them to make one-to-one communication possible.

A better solution is switching. A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems (computers or telephones, for example). Others are used only for routing. Figure 8.1 shows a switched network.



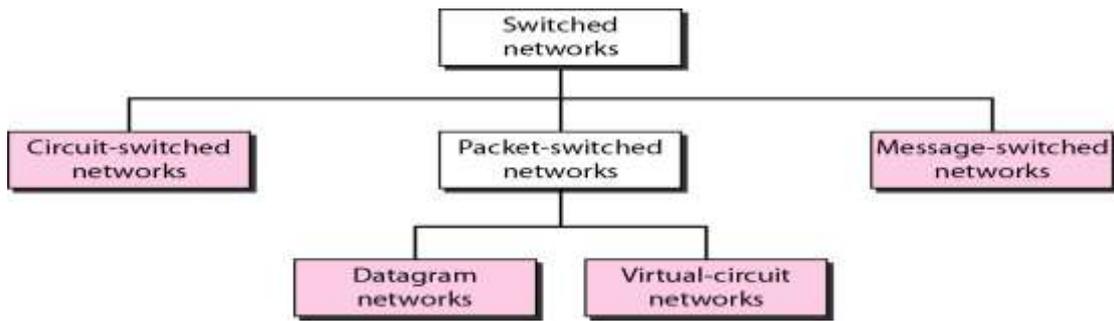


Figure 8.2 Taxonomy of switched networks

Traditionally, three methods of switching have been important: **circuit switching, packet switching, and message switching**. The first two are commonly used today. The third has been phased out in general communications but still has networking applications.

8.1 CIRCUIT-SWITCHED NETWORKS

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Figure 8.3 shows a trivial circuit-switched network with four switches and four links. Each link is divided into n (n is 3 in the figure) channels by using FDM or TDM.

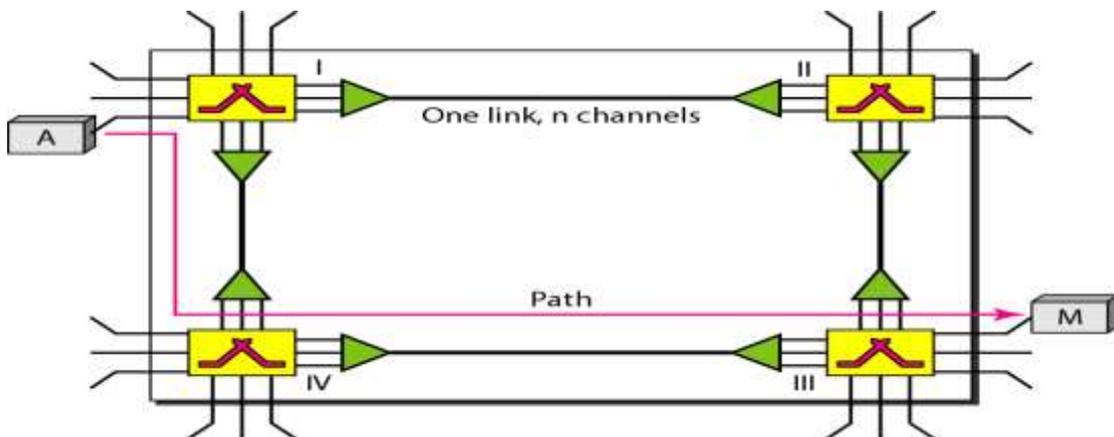


Figure 8.3 A trivial circuit-switched network

We have explicitly shown the multiplexing symbols to emphasize the division of the link into channels even though multiplexing can be implicitly included in the switch fabric.

The end systems, such as computers or telephones, are directly connected to a switch. We have shown only two end systems for simplicity. When end system A needs to communicate with end system M, system A needs

to request a connection to M that must be accepted by all switches as well as by M itself. This is called the setup phase;

a circuit (channel) is reserved on each link, and the combination of circuits or channels defines the dedicated path. After the dedicated path made of connected circuits (channels) is established, data transfer can take place. After all data have been transferred, the circuits are turned down.

We need to emphasize several points here:

- Circuit switching takes place at the physical layer.
- Before starting communication, the stations must make a reservation for the resources to be used during the communication. These resources, such as channels (bandwidth in FDM and time slots in TDM), switch buffers, switch processing time, and switch input/output ports, must remain dedicated during the entire duration of data transfer until the teardown phase *مرحلة الانهاء*.
- Data transferred between the two stations are not packetized (physical layer transfer of the signal). The data are a continuous flow sent by the source station and received by the destination station, although there may be periods of silence.

Three Phases

The actual communication in a circuit-switched network requires three phases: connection setup, data transfer, and connection teardown.

Setup Phase

Before the two parties (or multiple parties in a conference call) can communicate, a dedicated circuit (combination of channels in links) needs to be established. The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches. For example, in Figure 8.3, when system A needs to connect to system M, it sends a setup request that includes the address of system M, to switch I. Switch I finds a channel between itself and switch IV that can be dedicated for this purpose. Switch I then sends the request to switch IV, which finds a dedicated channel between itself and switch III. Switch III informs system M of system A's intention at this time.

In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A. Only after system A receives this acknowledgment is the connection established.

Data Transfer Phase

After the establishment of the dedicated circuit (channels), the two parties can transfer data.

Teardown Phase

When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

Efficiency

It can be argued that circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to other connections. In a telephone network, people normally terminate the communication when they have finished their conversation. However, in computer networks, a computer can be connected to another computer even if there is no activity for a long time. In this case, allowing resources to be dedicated means that other connections are deprived **تحرّم او تمنع**.

Delay

Although a circuit-switched network normally has low efficiency, the delay in this type of network is minimal. During data transfer the data are not delayed at each switch; the resources are allocated for the duration of the connection. delay caused by the setup is the **sum of four parts**: 1. the propagation time of the source computer request, 2. The request signal transfer time, 3. the propagation time of the acknowledgment from the destination computer, 4. and the signal transfer time of the acknowledgment, The delay due to data transfer is the sum of two parts: the propagation time and data transfer time , which can be very long. The third box shows the time needed to tear down the circuit. We have shown the case in which the receiver requests disconnection, which creates the maximum delay.

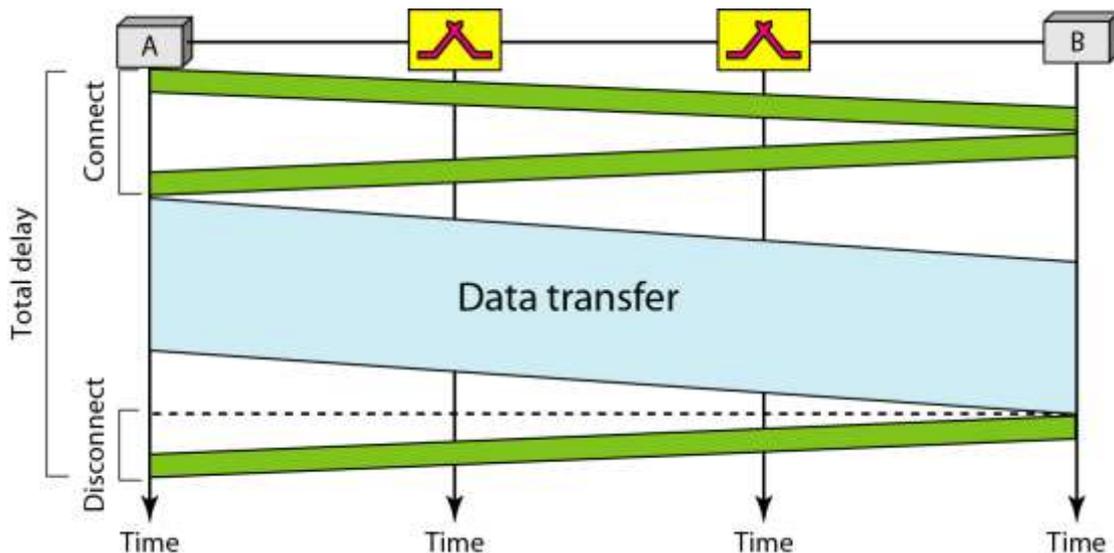


Figure 8.6 *Delay in a circuit-switched network*

Circuit-Switched Technology in Telephone Networks

The telephone companies have previously chosen the circuit switched approach to switching in the physical layer; today the tendency is moving toward other switching techniques. For example, the telephone number is used as the global address, and a signaling system (called SS7 telephony protocol) is used for the setup and teardown phases.

DATAGRAM NETWORKS

In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol.

In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet. Resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed. As with other systems in our daily life, this lack of reservation may create delay. For example, if we do not have a reservation at a restaurant, we might have to wait.

In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multipacket transmission, the network treats

it as though it existed alone. Packets in this approach are referred to as datagrams. **Datagram switching is normally done at the network layer.** We briefly discuss datagram networks here as a comparison with circuit-switched and virtual-circuits switched networks. Figure 8.7 shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers. That is why we use a different symbol for the switches in the figure.

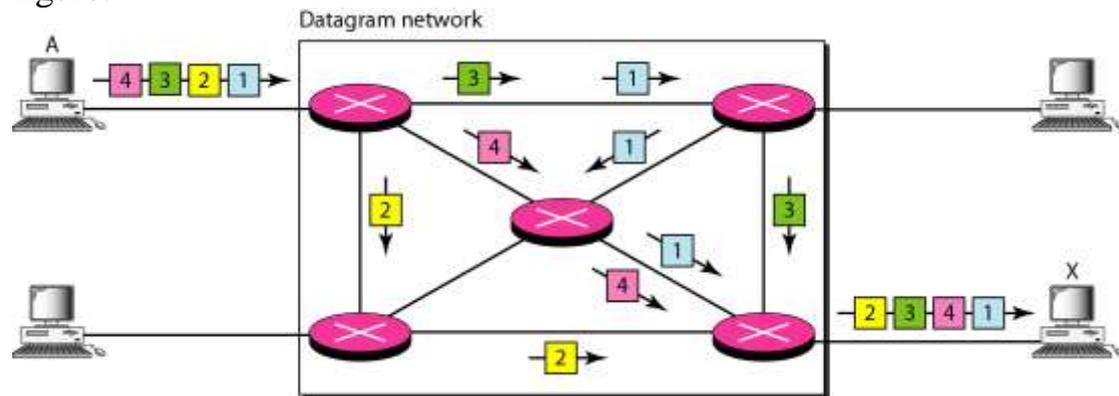


Figure 8.7 A datagram network with four switches (routers)

In this example, all four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination. This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X. This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between the packets. Packets may also be lost or dropped because of a lack of resources. In most protocols, it is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application. The datagram networks are sometimes referred to as connectionless networks. The term *connectionless* here means that the switch (packet switch) does not keep information about the connection state. **There are no setup or teardown phases.** Each packet is treated the same by a switch regardless of its source or destination.

Routing Table

If there are no setup or teardown phases, how are the packets routed to their destinations in a datagram network? In this type of network, each switch (or packet switch) has a routing table which is based on the destination address. The routing tables are dynamic and are updated periodically. The destination addresses and the corresponding forwarding output ports are recorded in the tables. This is different from the table of a circuit switched network in which each entry is created when the setup

phase is completed and deleted when the teardown phase is over. Figure 8.8 shows the routing table for a switch.

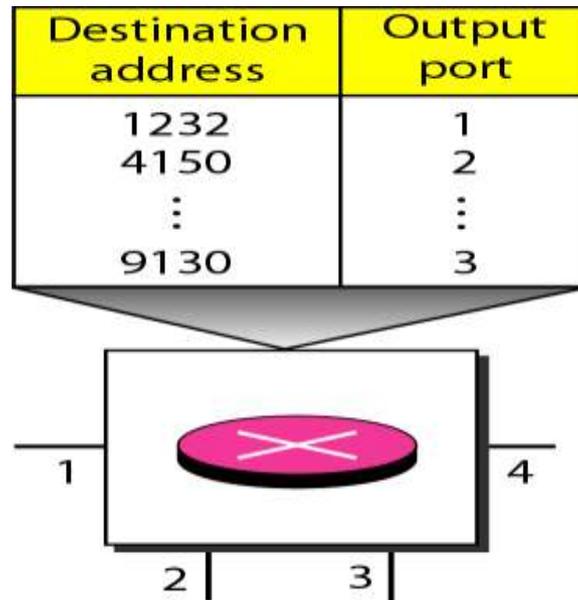


Figure 8.8 *Routing table in a datagram network*

Destination Address

Every packet in a datagram network carries a header that contains, among other information, the destination address of the packet. When the switch receives the packet, this destination address is examined; the routing table is consulted to find the corresponding port through which the packet should be forwarded. This address, unlike the address in a virtual-circuit-switched network, remains the same during the entire journey of the packet.

Efficiency

The efficiency of a datagram network is better than that of a circuit-switched network; resources are allocated only when there are packets to be transferred. If a source sends a packet and there is a delay of a few minutes before another packet can be sent, the resources can be reallocated during these minutes for other packets from other sources.

Delay

There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition,

since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message.

Datagram Networks in the Internet

The Internet has chosen the datagram approach to switching at the network layer. It uses the universal addresses defined in the network layer to route packets from the source to the destination.

VIRTUAL-CIRCUIT NETWORKS

A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both

1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
3. As in a datagram network, data are packetized and each packet carries an address in the header.
4. As in a circuit-switched network, all packets follow the same path established during the connection.
5. A virtual-circuit network is normally implemented in the data link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer. But this may change in the future.

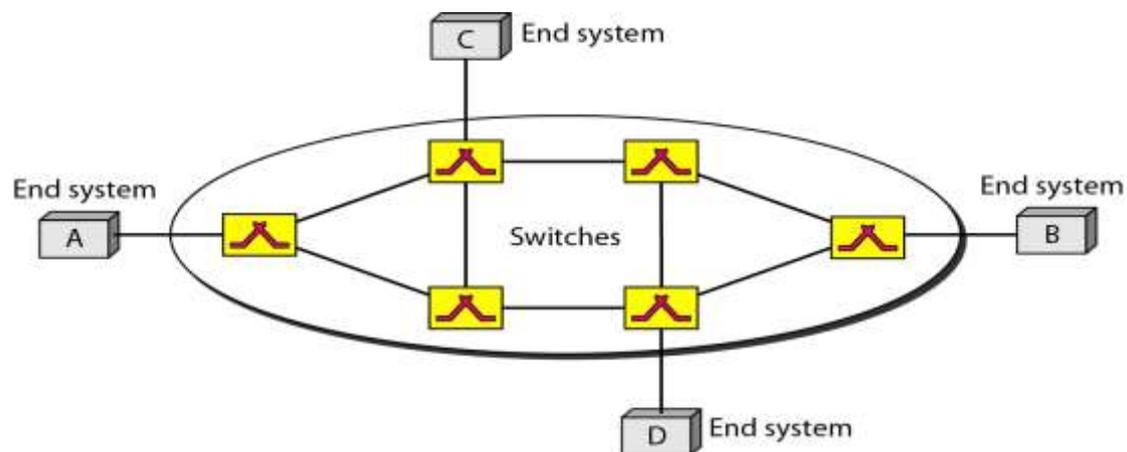


Figure 8.10 *Virtual-circuit network*

Figure 8.10 is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.

Addressing

In a virtual-circuit network, two types of addressing are involved: global and local (virtual-circuit identifier).

Global Addressing

A source or a destination needs to have a global address—an address that can be unique in the scope of the network or internationally if the network is part of an international network. However, we will see that a global address in virtual-circuit networks is used only to create a virtual-circuit identifier, as discussed next.

Virtual-Circuit Identifier

The identifier that is actually used for data transfer is called the virtual-circuit identifier (VCI). A VCI, unlike a global address, is a small number that has only switch scope; it is used by a frame between two switches. When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCI. Figure 8.11 shows how the VCI in a data frame changes from one switch to another. Note that a VCI does not need to be a large number since each switch can use its own unique set of VCIs.

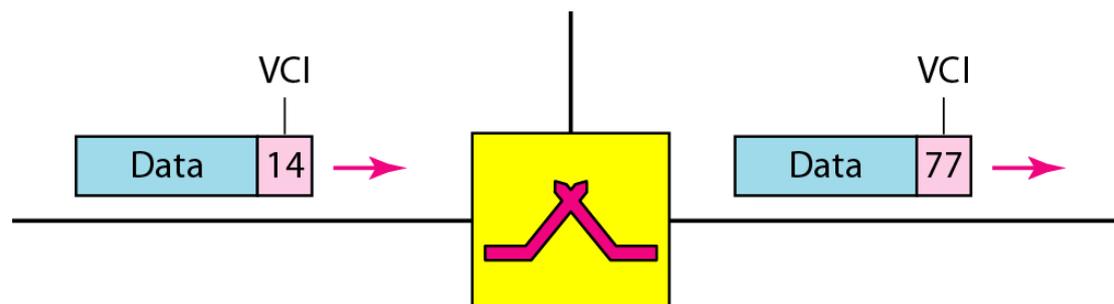


Figure 8.11 Virtual-circuit identifier

Three Phases

As in a circuit-switched network, a source and destination need to go through three phases in a virtual-circuit network: **setup, data transfer, and teardown**. In the setup phase, the source and destination use their global addresses to help switches make table entries for the connection. In the teardown phase, the source and destination inform the switches to

delete the corresponding entry. Data transfer occurs between these two phases.

Efficiency

As we said before, resource reservation in a virtual-circuit network can be made during the setup or can be on demand during the data transfer phase. In the first case, the delay for each packet is the same; in the second case, each packet may encounter different delays. There is one big advantage in a virtual-circuit network even if resource allocation is on demand. The source can check the availability of the resources, without actually reserving it. Consider a family that wants to dine at a restaurant. Although the restaurant may not accept reservations (allocation of the tables is on demand), the family can call and find out the waiting time. This can save the family time and effort.

Delay in Virtual-Circuit Networks

In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets. Figure 8.16 shows the delay for a packet traveling through two switches in a virtual-circuit network.

The packet is traveling through two switches (routers). There are three transmission times ($3T$), three propagation times ($3t$), data transfer depicted by the sloping lines, a setup delay (which includes transmission and propagation in two directions), and a teardown delay (which includes transmission and propagation in one direction). We ignore the processing time in each switch. The total delay time is

$$\text{Total delay} = 3T + 3t + \text{setup delay} + \text{teardown delay}$$

Circuit-Switched Technology in WANs

Virtual-circuit networks are used in switched WANs such as Frame Relay and ATM networks. The data link layer of these technologies is well suited to the virtual-circuit technology.

STRUCTURE OF A SWITCH

We use switches in circuit-switched and packet-switched networks. In this section, we discuss the structures of the switches used in each type of network

Structure of Circuit Switches

Circuit switching today can use either of two technologies: the space-division switch or the time-division switch.

Space-Division Switch

In space-division switching, the paths in the circuit are separated from one another spatially. This technology was originally designed for use in analog networks but is used currently in both analog and digital networks. It has evolved through a long history of many designs.

Crossbar Switch

A crossbar switch connects n inputs to m outputs in a grid; using electronic microswitches (transistors) at each crosspoint (see Figure 8.17). The major limitation of this design is the number of crosspoints required. To connect n inputs to m outputs using a crossbar switch requires $n \times m$ crosspoints. For example, to connect 1000 inputs to 1000 outputs requires a switch with 1,000,000 crosspoints. A crossbar with this number of crosspoints is impractical. Such a switch is also inefficient because statistics show that, in practice, fewer than 25 percent of the crosspoints are in use at any given time. The rest are idle.

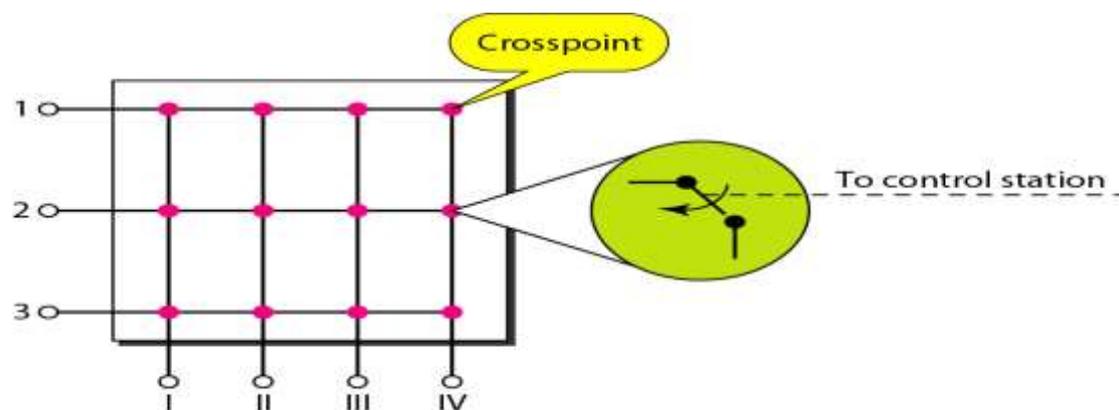


Figure 8.17 Crossbar switch with three inputs and four outputs

Multistage Switch

The solution to the limitations of the crossbar switch is the multistage switch, which combines crossbar switches in several (normally three) stages, as shown in Figure 8.18. In a single crossbar switch, only one row or column (one path) is active for any connection. So we need $N \times N$ crosspoints. If we can allow multiple paths inside the switch, we can

decrease the number of crosspoints. Each crosspoint in the middle stage can be accessed by multiple crosspoints in the first or third stage.

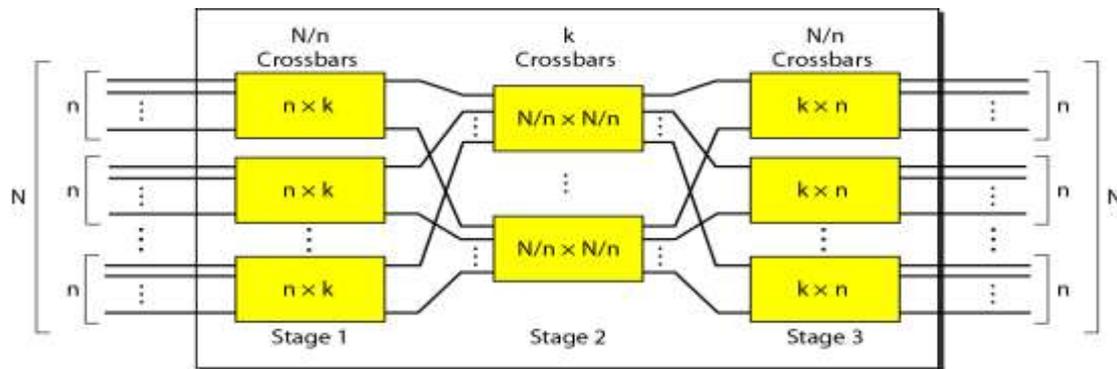


Figure 8.18 Multistage switch

To design a three-stage switch, we follow these steps:

1. We divide the N input lines into groups, each of n lines. For each group, we use one crossbar of size $n \times k$, where k is the number of crossbars in the middle stage. In other words, the first stage has N/n crossbars of $n \times k$ crosspoints.
2. We use k crossbars, each of size $(N/n) \times (N/n)$ in the middle stage.
3. We use N/n crossbars, each of size $k \times n$ at the third stage.

We can calculate the total number of crosspoints as follows:

$$2kN + k(N/n)^2$$

Example 8.3

Design a three-stage, 200×200 switch ($N = 200$) with $k = 4$ and $n = 20$.

Solution

In the first stage we have N/n or 10 crossbars, each of size 20×4 . In the second stage, we have 4 crossbars, each of size 10×10 . In the third stage, we have 10 crossbars, each of size 4×20 . The total number of

crosspoints is $2kN + k(N/n)^2$, or 2000 crosspoints. This is 5 percent of the number of crosspoints in a single-stage switch ($200 \times 200 = 40,000$).

Time-Division Switch

Time-division switching uses time-division multiplexing (TDM) inside a switch. The most popular technology is called the time-slot interchange (TSI).

Time-Slot Interchange

Figure 8.19 shows a system connecting four input lines to four output lines. Imagine that each input line wants to send data to an output line according to the following pattern:



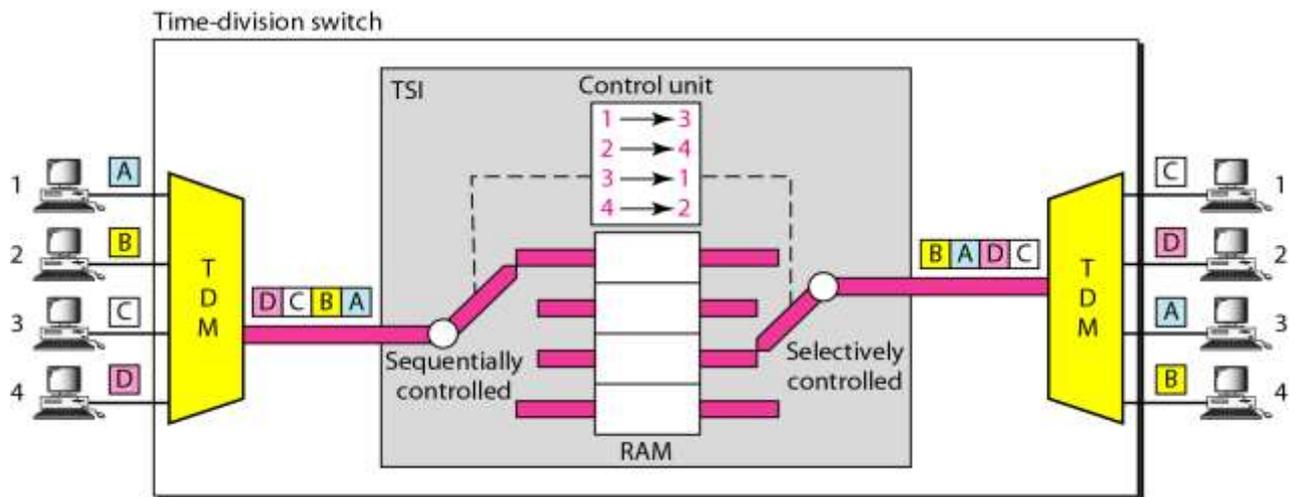


Figure 8.19 *Time-slot interchange*

The figure combines a TDM multiplexer, a TDM demultiplexer, and a TSI consisting of random access memory (RAM) with several memory locations. The size of each location is the same as the size of a single time slot. The number of locations is the same as the number of inputs (in most cases, the numbers of inputs and outputs are equal). The RAM fills up with incoming data from time slots in the order received. Slots are then sent out in an order based on the decisions of a control unit.

Time- and Space-Division Switch Combinations

When we compare space-division and time-division switching, some interesting facts emerge. The advantage of space-division switching is that it is instantaneous. Its disadvantage is the number of crosspoints required to make space-division switching acceptable in terms of blocking. The advantage of time-division switching is that it needs no crosspoints. Its disadvantage,

in the case of TSI, is that processing each connection creates delays. Each time slot must be stored by the RAM, then retrieved and passed on.

In a third option, we combine space-division and time-division technologies to take advantage of the best of both. Combining the two results in switches that are optimized both physically (the number of crosspoints) and temporally (the amount of delay). Multistage switches of this sort can be designed as time-space-time (TST) switch.

Figure 8.20 shows a simple TST switch that consists of two time stages and one

space stage and has 12 inputs and 12 outputs. Instead of one time-division switch, it

divides the inputs into three groups (of four inputs each) and directs them to three timeslot interchanges. The result is that the average delay is one-third of what would result from using one time-slot interchange to handle

all 12 inputs. The last stage is a mirror image of the first stage. The middle stage is a space division switch (crossbar) that connects the TSI groups to allow connectivity between all possible input and output pairs (e.g., to connect input 3 of the first group to output 7 of the second group).

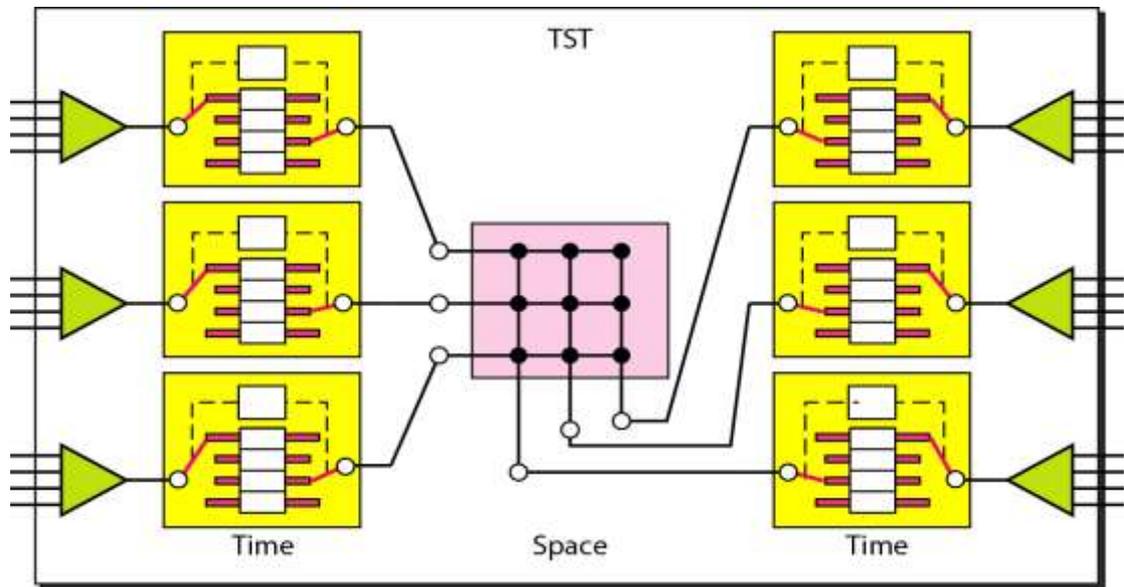
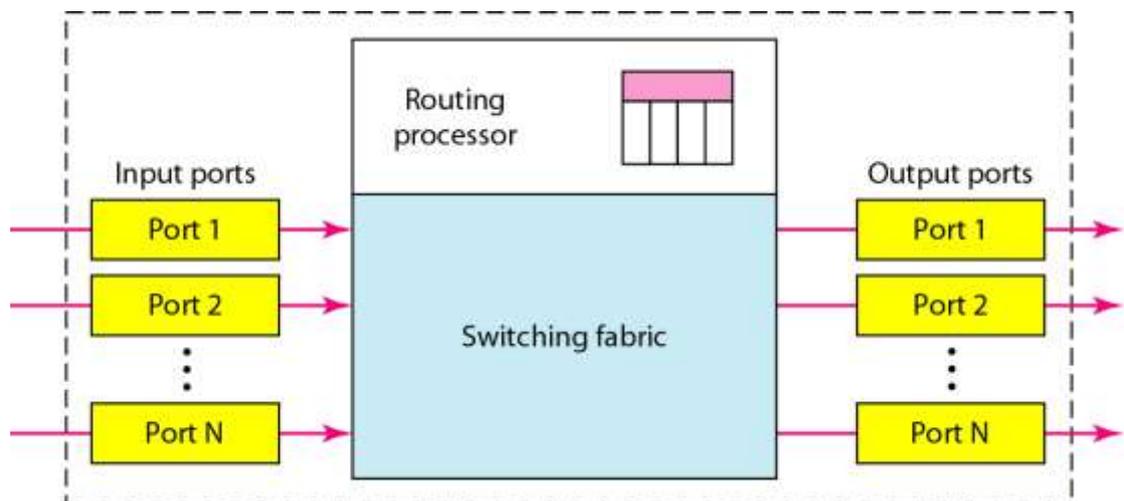


Figure 8.20 Time-space-time switch

Structure of Packet Switches

A switch used in a packet-switched network has a different structure from a switch used in a circuit-switched network. We can say that a packet switch has four components: input ports, output ports, the routing processor, and the switching fabric, as shown in Figure 8.21.

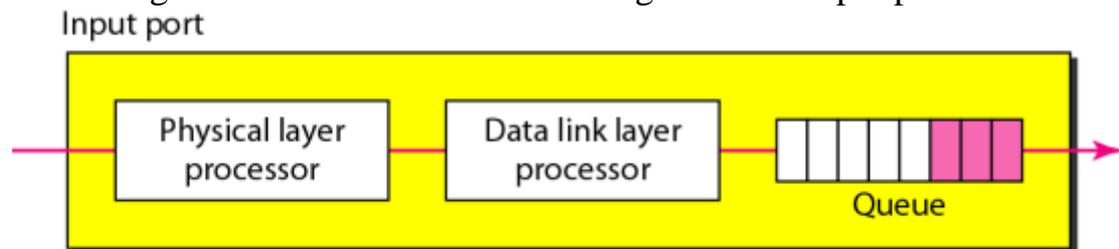


1

Figure 8.21 Packet switch components

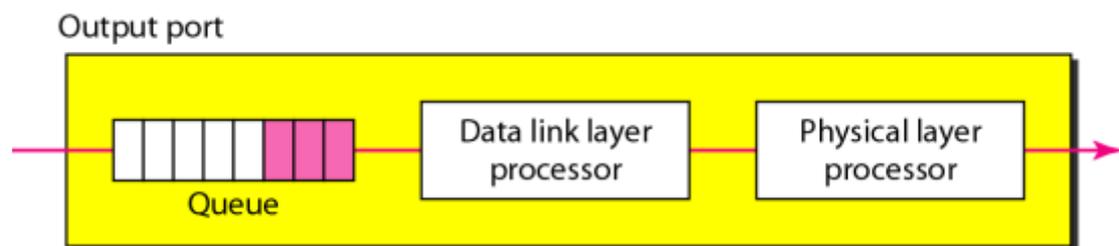
Input Ports

An input port performs the physical and data link functions of the packet switch. The bits are constructed from the received signal. The packet is decapsulated from the frame. Errors are detected and corrected. The packet is now ready to be routed by the network layer. In addition to a physical layer processor and a data link processor, the input port has buffers (queues) to hold the packet before it is directed to the switching fabric. Figure 8.22 shows a schematic diagram of an input port.



Output Port

The output port performs the same functions as the input port, but in the reverse order. First the outgoing packets are queued, then the packet is encapsulated in a frame, and finally the physical layer functions are applied to the frame to create the signal to be sent on the line. Figure 8.23 shows a schematic diagram of an output port.



Routing Processor

The routing processor performs the functions of the network layer. The destination address is used to find the address of the next hop and, at the same time, the output port number from which the packet is sent out. This activity is sometimes referred to as **table lookup** because the routing processor searches the routing table. In the newer packet switches, this function of the routing processor is being moved to the input ports to facilitate and expedite الاسراع the process.

Switching "Fabrics"

The most difficult task in a packet switch is to move the packet from the input queue to the output queue. The speed with which this is done affects the size of the input/output queue and the overall delay in packet delivery. In the past, when a packet switch was actually a dedicated computer, the

memory of the computer or a bus was used as the switching fabric. The input port stored the packet in memory; the output port retrieved the packet from memory. Today, packet switches are specialized mechanisms that use a variety of switching fabrics.

DIAL UP MODEMS

Traditional telephone lines can carry frequencies between 300 and 3300 Hz, giving them a bandwidth of 3000 Hz. All this range is used for transmitting voice, where a great deal of interference and distortion can be accepted without loss of intelligibility.

As we have seen, however, data signals require a higher degree of accuracy to ensure integrity. For safety's sake, therefore, the edges of this range are not used for data communications.

In general, we can say that the signal bandwidth must be smaller than the cable bandwidth. The effective bandwidth of a telephone line being used for data transmission is 2400 Hz, covering the range from 600 to 3000 Hz. Note that today some telephone lines are capable of handling greater bandwidth than traditional lines. However, modem design is still based on traditional capability (see Figure 9.6).



Figure 9.6 Telephone line bandwidth

The term modem is a composite word that refers to the two functional entities that make up the device: a signal modulator and a signal demodulator. A modulator creates a bandpass analog signal from binary data. A demodulator recovers the binary data from the modulated signal.

Figure 9.7 shows the relationship of modems to a communications line. The computer on the left sends a digital signal to the modulator portion of the modem; the data are sent as an analog signal on the telephone lines. The modem on the right receives the analog signal, demodulates it through its demodulator, and delivers data to the computer on the right. The communication can be bidirectional, which means the computer on the right can simultaneously send data to the computer on the left, using the same modulation/demodulation processes.

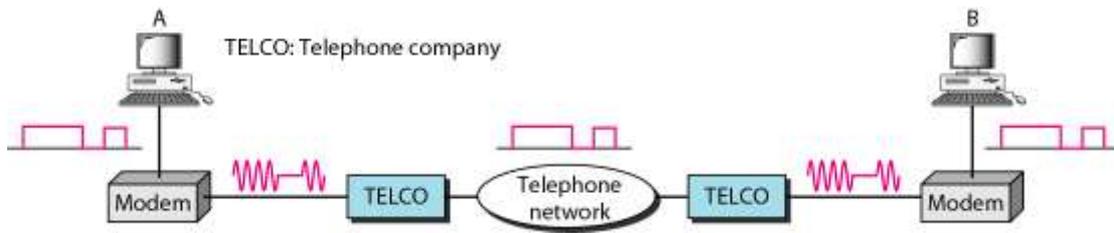


Figure 9.7 *Modulation/demodulation*

DIGITAL SUBSCRIBER LINE

After traditional modems reached their peak data rate, telephone companies developed another technology, DSL, to provide higher-speed access to the Internet. Digital subscriber line (DSL) technology is one of the most promising for supporting high-speed digital communication over the existing local loops. DSL technology is a set of technologies, each differing in the first letter (ADSL, VDSL, HDSL, and SDSL). The set is often referred to as xDSL, where *x* can be replaced by A, V, H, or S.

ADSL

The first technology in the set is asymmetric DSL (ADSL). ADSL, like a 56K modem, provides higher speed (bit rate) in the downstream direction (from the Internet to the resident) than in the upstream direction (from the resident to the Internet). That is the reason it is called asymmetric. Unlike the asymmetry in 56K modems, the designers of ADSL specifically divided the available bandwidth of the local loop unevenly for the residential customer. The service is not suitable for business customers who need a large bandwidth in both directions.

Using Existing Local Loops

One interesting point is that ADSL uses the existing local loops. But how does ADSL reach a data rate that was never achieved with traditional modems? The answer is that the twisted-pair local loop is actually capable of handling bandwidths up to 1.1 MHz, but the filter installed at the end office of the telephone company where each local loop terminates limits the bandwidth to 4 kHz (sufficient for voice communication). If the filter is removed, however, the entire 1.1 MHz is available for data and voice communications.

Adaptive Technology

Unfortunately, 1.1 MHz is just the theoretical bandwidth of the local loop. Factors such as the distance between the residence and the switching office, the size of the cable, the signaling used, and so on affect the bandwidth. The designers of ADSL technology were aware of this problem and used an adaptive technology that tests the condition and

bandwidth availability of the line before settling on a data rate. The data rate of ADSL is not fixed; it changes based on the condition and type of the local loop cable.

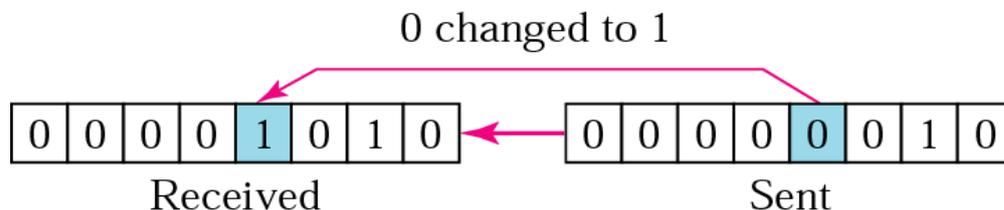
Error Detection and Correction

Networks must be able to transfer data from one device to another with acceptable accuracy. For most applications, a system must guarantee that the data received are identical to the data transmitted. Any time data are transmitted from one node to the next, they can become corrupted in passage. Many factors can alter one or more bits of a message. Some applications require a mechanism for detecting and correcting errors.

Types of Errors

- *Single-Bit Error*

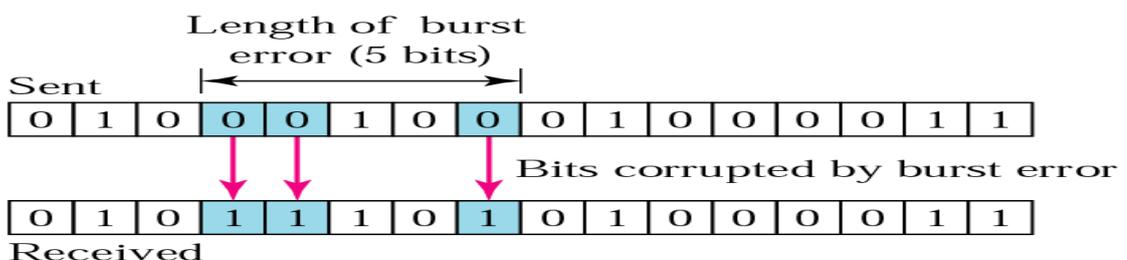
The term *single-bit error* means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1. Single-bit errors are the least likely type of error in serial data transmission. To understand why, imagine data sent at 1 Mbps. This means that each bit lasts only $1/1,000,000$ s, or 1 μ s. For a single-bit error to occur, the noise must have a duration of only 1 μ s, which is very rare; noise normally lasts much longer than this.



10.1 Single-bit error

- *Burst Error*

the term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.



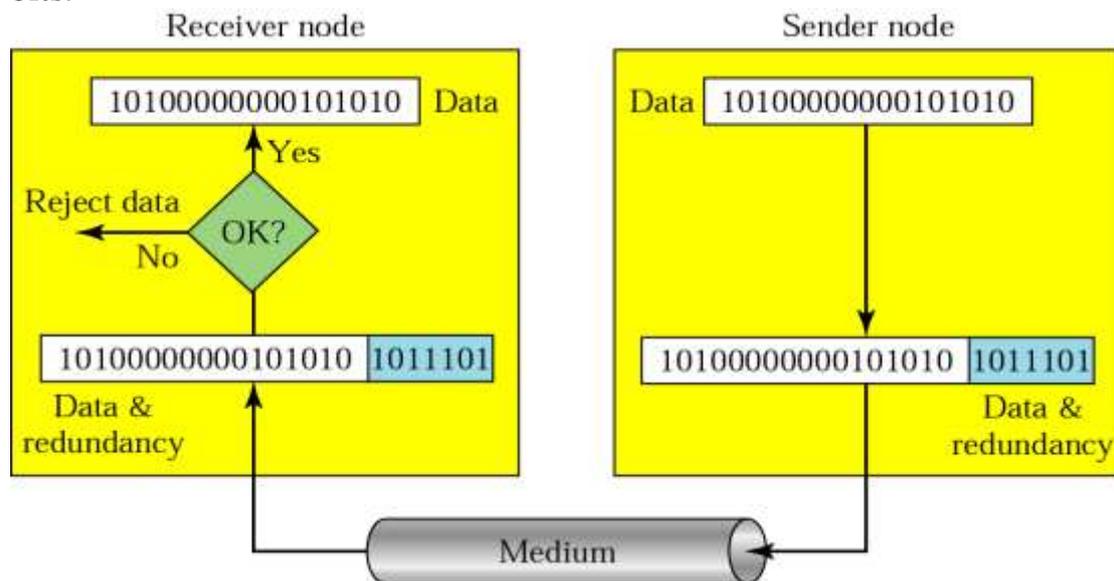
10.2 Burst error of length 5

Figure 10.2 shows the effect of a burst error on a data unit. In this case, 0100010001000011 was sent, but 0101110101100011 was received. Note that a burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at 1 kbps, a noise of 11100 s can affect 10 bits; if we are sending data at 1 Mbps, the same noise can affect 10,000 bits.

Redundancy

The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.



Fig()Redundancy

Detection Versus Correction

The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error.

In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The

number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

Forward Error Correction Versus Retransmission

There are two main methods of error correction. Forward error correction is the process in which the receiver tries to guess the message by using redundant bits. This is possible, as we see later, if the number of errors is small. Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error-free (usually, not all errors can be detected).

Coding

Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits. The receiver checks the relationships between the two sets of bits to detect or correct the errors. The ratio of redundant bits to the data bits and the robustness of the process are important factors in any coding scheme. Figure 10.3 shows the general idea of coding.

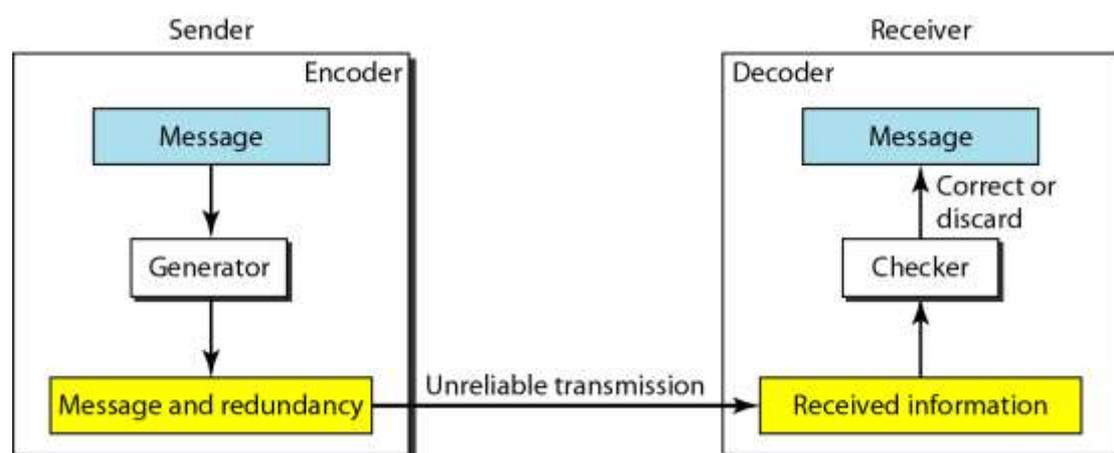


Figure 10.3 *The structure of encoder and decoder*

We can divide coding schemes into two broad categories: block coding and convolution coding. In this book, we concentrate on block coding;

convolution coding is more complex and beyond the scope of these lectures.

Modular Arithmetic

Before we finish this section, let us briefly discuss a concept basic to computer science in general and to error detection and correction in particular: modular arithmetic. Our intent here is not to delve deeply into the mathematics of this topic; we present just enough information to provide a background to materials discussed in this chapter. In modular arithmetic, we use only a limited range of integers. We define an upper limit, called a modulus N . We then use only the integers 0 to $N - 1$, inclusive. This is *modulo- N* arithmetic. For example, if the modulus is 12, we use only the integers 0 to 11, inclusive. An example of modulo arithmetic is our clock system. It is based on modulo-12 arithmetic, substituting the number 12 for 0. In a *modulo- N* system, if a number is greater than N , it is divided by N and the remainder is the result. If it is negative, as many N s as needed are added to make it positive. Consider our clock system again. If we start a job at 11 A.M. and the job takes 5 h, we can say that the job is to be finished at 16:00 if we are in the military, or we can say that it will be finished at 4 P.M. (the remainder of $16/12$ is 4).

Addition and subtraction in modulo arithmetic are simple. There is no carry when you add two digits in a column. There is no carry when you subtract one digit from another in a column.

Modulo-2 Arithmetic

Of particular interest is modulo-2 arithmetic. In this arithmetic, the modulus N is 2. We can use only 0 and 1. Operations in this arithmetic are very simple. The following shows how we can add or subtract 2 bits.

Adding $0+0=0$ $0+1=1$ $1+0=1$ $1+1=0$

Subtracting $0-0=0$ $0-1=1$ $1-0=1$ $1-1=0$

Notice particularly that addition and subtraction give the same results. In this arithmetic we use the XOR (exclusive OR) operation for both addition and subtraction. The result of an XOR operation is 0 if two bits

are the same; the result is 1 if two bits are different. Figure 10.4 shows this operation.

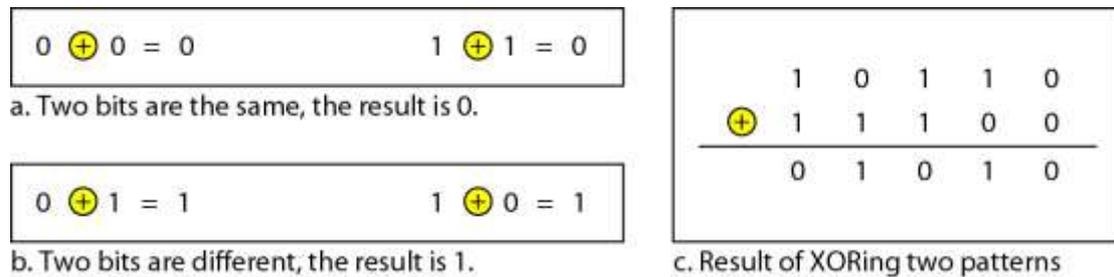


Figure 10.4 XORing of two single bits or two words

BLOCK CODING

In block coding, we divide our message into blocks, each of k bits, called datawords. We add r redundant bits to each block to make the length $n = k + r$. The resulting n -bit blocks are called codewords. How the extra r bits is chosen or calculated is something we will discuss later. For the moment, it is important to know that we have a set of datawords, each of size k , and a set of codewords, each of size of n . With k bits, we can create a combination of 2^k datawords; with n bits, we can create a combination of 2^n codewords. Since $n > k$, the number of possible codewords is larger than the number of possible datawords.

The block coding process is one-to-one; the same dataword is always encoded as the same codeword. This means that we have $2^n - 2^k$ codewords that are not used. We call these codewords invalid or illegal. Figure 10.5 shows the situation.

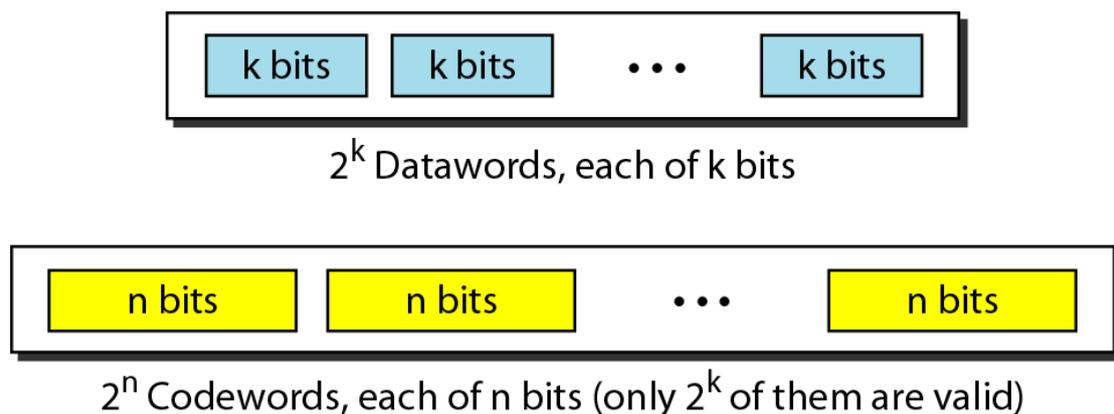


Figure 10.5 Datawords and codewords in block coding

Error Detection

How can errors be detected by using block coding? If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid codewords.
2. The original codeword has changed to an invalid one.

Figure 10.6 shows the role of block coding in error detection.

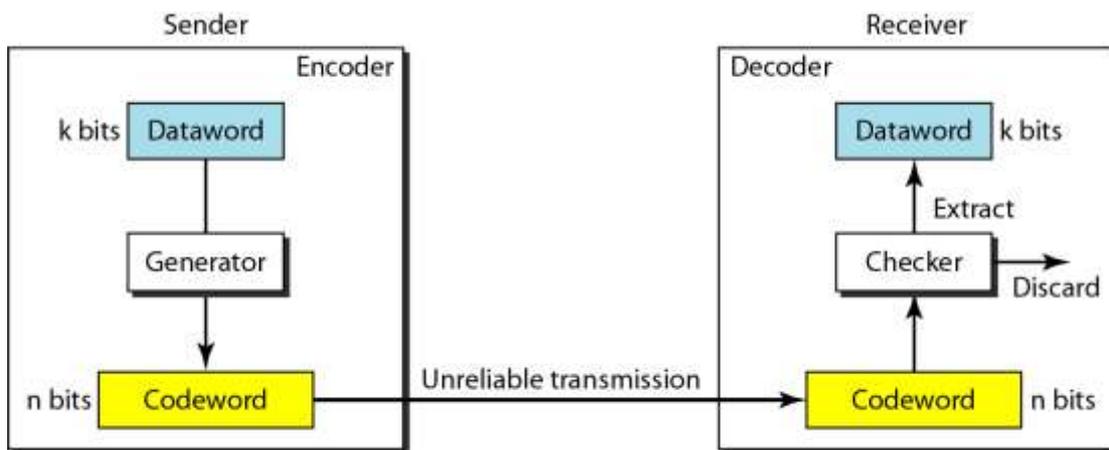


Figure 10.6 Process of error detection in block coding

The sender creates codewords out of datawords by using a generator that applies the rules and procedures of encoding (discussed later). Each codeword sent to the receiver may change during transmission. If the received codeword is the same as one of the valid codewords, the word is accepted; the corresponding dataword is extracted for use. If the received codeword is not valid, it is discarded. However, if the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected. This type of coding can detect only single errors. Two or more errors may remain undetected.

Example

Let us assume that $k = 2$ and $n = 3$. Table 10.1 shows the list of datawords and codewords. Later, we will see how to derive a codeword from a dataword.

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases:

1. The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.
2. The codeword is corrupted during transmission, and 111 is received. This is not a valid codeword and is discarded.
3. The codeword is corrupted during transmission, and 000 is received. This is a valid codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.

Table 10.1 *A code for error detection (Example 10.2)*

<i>Datawords</i>	<i>Codewords</i>
00	000
01	011
10	101
11	110

Error Correction

As we said before, error correction is much more difficult than error detection. In error detection, the receiver needs to know only that the received codeword is invalid; in error correction the receiver needs to find (or guess) the original codeword sent. We can say that we need more redundant bits for error correction than for error detection.

Figure 10.7 shows the role of block coding in error correction. We can see that the idea is the same as error detection but the checker functions are much more complex.

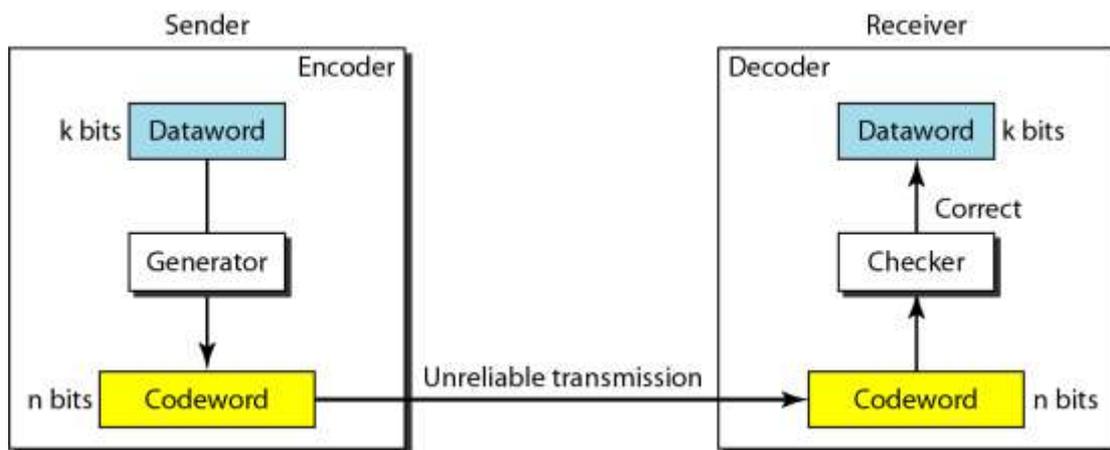


Figure 10.7 Structure of encoder and decoder in error correction

Hamming Distance

One of the central concepts in coding for error control is the idea of the Hamming distance. The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits. We show the Hamming distance between two words x and y as $d(x, y)$.

The Hamming distance can easily be found if we apply the XOR operation on the two words and count the number of 1s in the result. Note that the Hamming distance is a value greater than zero.

Example 10.4

Let us find the Hamming distance between two pairs of words.

1. The Hamming distance $d(000, 011)$ is 2 because

$$000 \oplus 011 \text{ is } 011 \text{ (two 1s)}$$

2. The Hamming distance $d(10101, 11110)$ is 3 because

$$10101 \oplus 11110 \text{ is } 01011 \text{ (three 1s)}$$

Data Link Control

The two main functions of the data link layer are data link control and media access control. The first, data link control, deals with the design and procedures for communication between two adjacent nodes: node-to-node communication.. The second function of the data link layer is media access control, or how to share the link. **Data link control** functions include framing, flow and error control, and software

implemented protocols that provide smooth and reliable transmission of frames between nodes. In this chapter, we first discuss framing, or how to organize the bits that are carried by the physical layer.

FRAMING

Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination. The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing.

The data link layer, on the other hand, needs to pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing. The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility.

Fixed-Size Framing

Frames can be of fixed or variable size. In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network.

Variable-Size Framing

Our main discussion in this chapter concerns variable-size framing, prevalent in local area networks. In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose: **a character-oriented approach and a bit-oriented approach.**

FLOW AND ERROR CONTROL

Data communication requires at least two devices working together, one to send and the other to receive. Even such a basic arrangement requires a great deal of coordination for an intelligible exchange to occur. The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

Flow Control

Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a *buffer*, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

Error Control

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term *error control* refers primarily to methods of error detection and retransmission. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ).

PROTOCOLS

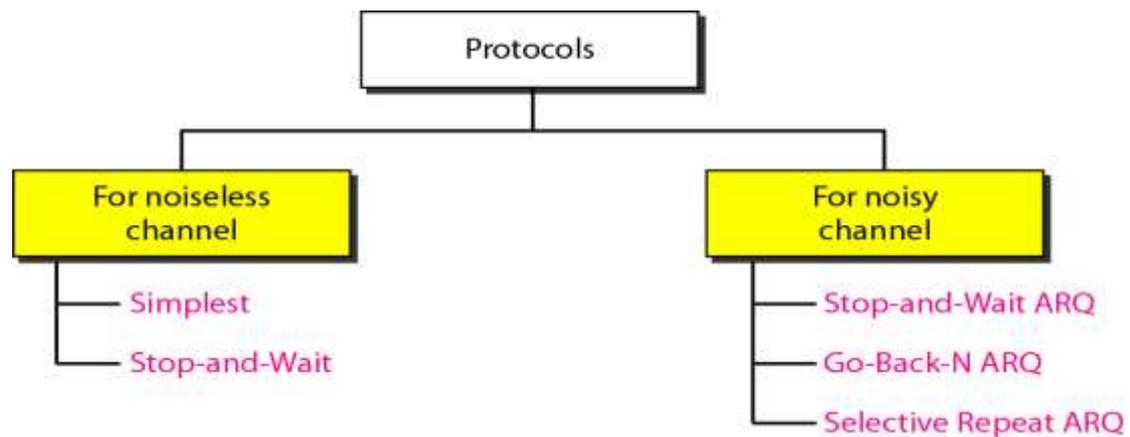


Figure 11.5 *Taxonomy of protocols discussed*

NOISELESS CHANNELS

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel. The first is a protocol that does not use flow control; the second is the one that does.

Simplest Protocol

It is a unidirectional protocol in which data frames are traveling in only one direction—from the sender to receiver. We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately. In other words, the receiver can never be overwhelmed with incoming frames.

Design

There is no need for flow control in this scheme. The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it. The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer. The data link layers of the sender and receiver provide transmission services for their network layers. The data link layers use the services provided by their physical layers (such as

signaling, multiplexing, and so on) for the physical transmission of bits. Figure 11.6 shows a design.

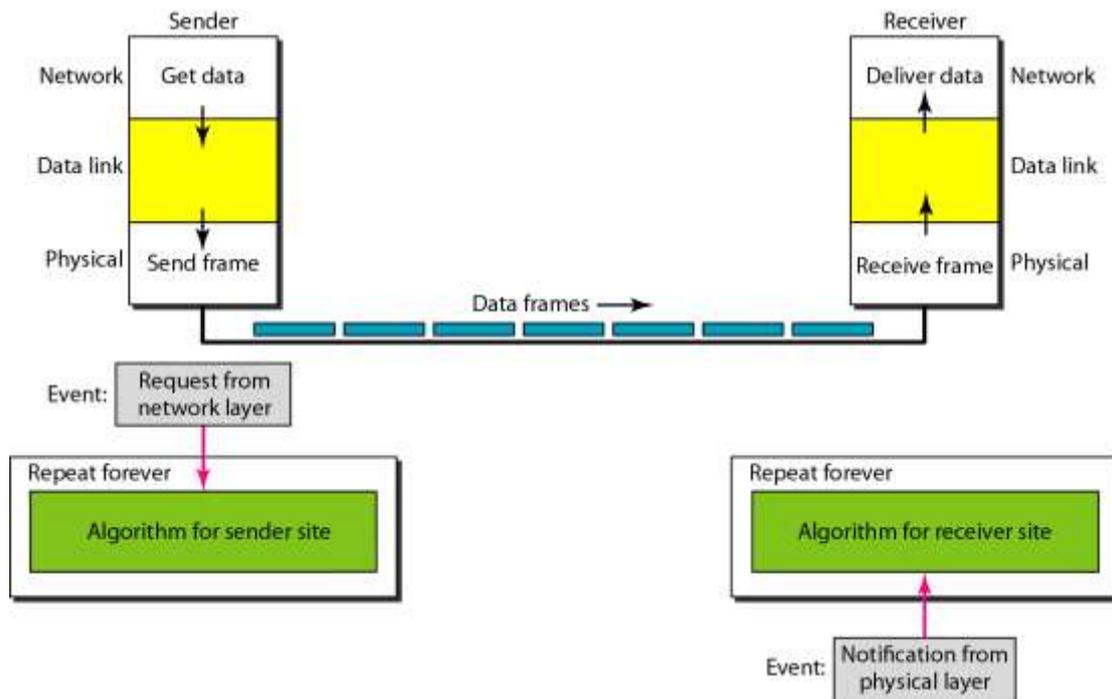


Figure 11.6 *The design of the simplest protocol with no flow or error control*

We need to elaborate on the procedure used by both data link layers. The sender site cannot send a frame until its network layer has a data packet to send. The receiver site cannot deliver a data packet to its network layer until a frame arrives. If the protocol is implemented as a procedure, we need to introduce the idea of events in the protocol. The procedure at the sender site is constantly running; there is no action until there is a request from the network layer. The procedure at the receiver site is also constantly running, but there is no action until notification from the physical layer arrives. Both procedures are constantly running because they do not know when the corresponding events will occur.

Algorithm 11.1 *Sender-site algorithm for the simplest protocol*

```

1 while(true) // Repeat forever
2 {
3   WaitForEvent(); // Sleep until an event occurs
4   if(Event(RequestToSend)) //There is a packet to send
5   {
6     GetData();
7     MakeFrame();
8     SendFrame(); //Send the frame
9   }
10 }

```

Analysis The algorithm has an infinite loop, which means lines 3 to 9 are repeated forever once the program starts. The algorithm is an event-driven one, which means that it *sleeps* (line 3) until an event *wakes it up* (line 4). This means that there may be an undefined span of time between the execution of line 3 and line 4; there is a gap between these actions. When the event, a request from the network layer, occurs, lines 6 through 8 are executed. The program then repeats the loop and again sleeps at line 3 until the next occurrence of the event. We have written pseudocode for the main process. We do not show any details for the modules GetData, MakeFrame, and SendFrame. GetDataO takes a data packet from the network layer, MakeFrameO adds a header and delimiter flags to the data packet to make a frame, and SendFrameO delivers the frame to the physical layer for transmission.

Algorithm 11.2 shows the procedure at the receiver site.

```
1 while(true) // Repeat forever
2 {
3   WaitForEvent(); // Sleep until an event occurs
4   if(Event(ArrivalNotification)) //Data frame arrived
5   {
6     ReceiveFrame();
7     ExtractData();
8     DeliverData(); //Deliver data to network layer
9   }
10 }
```

Stop-and-Wait Protocol

If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service. To prevent the receiver from becoming overwhelmed with frames, we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender. The protocol we discuss now is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens

of acknowledgment) travel from the other direction. We add flow control to our previous protocol.

Design

Figure 11.8 illustrates the mechanism. Comparing this figure with Figure 11.6, we can see the traffic on the forward channel (from sender to receiver) and the reverse channel. At any time, there is either one data frame on the forward channel or one ACK frame on the reverse channel. We therefore need a half-duplex link.

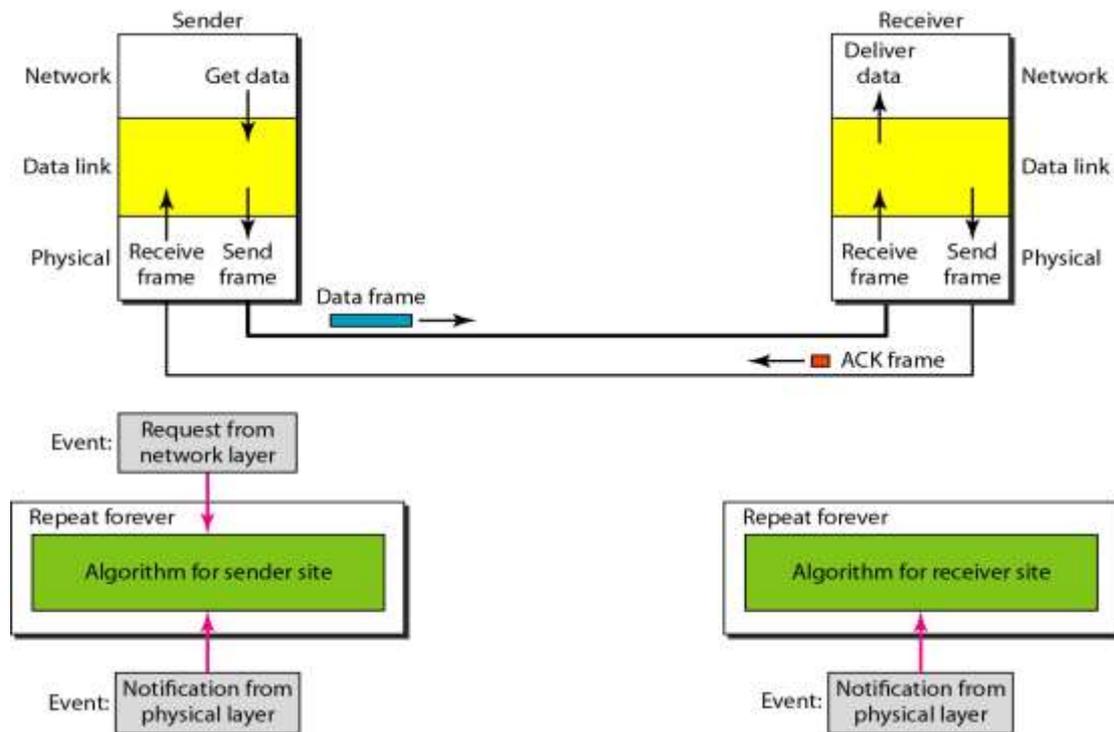


Figure 11.8 Design of Stop-and-Wait Protocol

```

1 while(true) //Repeat forever
2 canSend = true //Allow the first frame to go
3 {
4   WaitForEvent(); // Sleep until an event occurs
5   if(Event(RequestToSend) AND canSend)
6   {
7     GetData();
8     MakeFrame();
9     SendFrame(); //Send the data frame
10    canSend = false; //Cannot send until ACK arrives
11  }
12  WaitForEvent(); // Sleep until an event occurs
13  if(Event(ArrivalNotification) // An ACK has arrived
14  {
15    ReceiveFrame(); //Receive the ACK frame
16    canSend = true;
17  }
18 }

```

Algorithm 11.3 *Sender-site algorithm for Stop-and-Wait Protocol*

Analysis Here two events can occur: a request from the network layer or an arrival notification from the physical layer. The responses to these events must alternate. In other words, after a frame is sent, the algorithm must ignore another network layer request until that frame is acknowledged. We know that two arrival events cannot happen one after another because the channel is error-free and does not duplicate the frames. The requests from the network layer, however, may happen one after another without an arrival event in between. We need somehow to prevent the immediate sending of the data frame. Although there are several methods, we have used a simple *canSend* variable that can either be true or false. When a frame is sent, the variable is set to false to indicate that a new network request cannot be sent until *canSend* is true. When an ACK is received, *canSend* is set to true to allow the sending of the next frame.

Algorithm 11.4 shows the procedure at the receiver site.

```

1 while(true) //Repeat forever
2 {
3   WaitForEvent(); // Sleep until an event occurs
4   if(Event(ArrivalNotification)) //Data frame arrives
5   {
6     ReceiveFrame();
7     ExtractData();
8     Deliver(data); //Deliver data to network layer
9     SendFrame(); //Send an ACK frame
10  }
11 }

```

Analysis This is very similar to Algorithm 11.2 with one exception. After the data frame arrives, the receiver sends an ACK frame (line 9) to acknowledge the receipt and allow the sender to send the next frame.

NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error (as we sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

Stop-and-Wait Automatic Repeat Request

Our first protocol, called the Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ), adds a simple error control mechanism to the Stop-and-Wait Protocol. Let us see how this protocol detects and corrects errors. To detect and correct corrupted frames, we need to add redundancy bits to our data frame (see Chapter 10). When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver. Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated. The completed and lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it

starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network.

Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

Sequence Numbers

As we discussed, the protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame. One important consideration is the range of the sequence numbers. Since we want

to minimize the frame size, we look for the smallest range that provides unambiguous communication. The sequence numbers of course can wrap around. For example, if we decide that the field is m bits long, the sequence numbers start from 0, go to $2^m - 1$, and then are repeated. Let us reason out the range of sequence numbers we need. Assume we have used x as a sequence number; we only need to use $x + 1$ after that. There is no need for $x + 2$. To show this, assume that the sender has sent the frame numbered x . Three things can happen.

1. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment.

The acknowledgment arrives at the sender site, causing the sender to send the next frame numbered $x + 1$.

2. The frame arrives safe and sound at the receiver site; the receiver sends an acknowledgment, but the acknowledgment is corrupted or lost. The sender resends the frame (numbered x) after the time-out. Note that the frame here is a duplicate. The receiver can recognize this fact because it expects frame $x + 1$ but frame x was received.

3. The frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered x) after the time-out.

We can see that there is a need for sequence numbers x and $x + 1$ because the receiver needs to distinguish between case 1 and case 2. But there is no need for a frame to be numbered $x + 2$. In case 1, the frame can be numbered x again because frames x and $x + 1$ are acknowledged and there is no ambiguity at either site. In cases 2 and 3, the new frame is $x + 1$, not $x + 2$. If only x and $x + 1$ are needed, we can let $x = 0$ and $x + 1 = 1$. This means that the sequence is 0, 1, 0, 1, 0, and so on.

Acknowledgment Numbers

Since the sequence numbers must be suitable for both data frames and ACK frames, we use this convention: The acknowledgment numbers always announce the sequence number of the next frame expected by the receiver. For example, if frame 0 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 1 (meaning frame 1 is expected next). If frame 1 has arrived safe and sound, the receiver sends an ACK frame with acknowledgment 0 (meaning frame 0 is expected).

Design

Figure 11.10 shows the design of the Stop-and-Wait ARQ Protocol. The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. A data frame uses a seqNo (sequence number); an ACK frame uses an ackNo (acknowledgment number). The sender has a control variable, which we call S_n (sender, next frame to send), that holds the sequence number for the next frame to be sent (0 or 1).

HDLC

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms we discussed in this chapter.

Configurations and Transfer Modes

HDLC provides two common transfer modes that can be used in different configurations: normal response mode (NRM) and asynchronous balanced mode (ABM).

Normal Response Mode

In normal response mode (NRM), the station configuration is unbalanced. We have one primary station and multiple secondary stations. A primary station can send commands; a secondary station can only respond. The NRM is used for both point-to-point and multiple-point links, as shown in Figure 11.25.

Frames

To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (V-frames). Each type of frame serves as an envelope for the transmission of a different type of message. I-frames are used to transport user data and control information relating to user data (piggybacking). S-frames are used only to transport control information. V-frames are reserved for system management. Information carried by V-frames is intended for managing the link itself.

Frame Format

Each frame in HDLC may contain up to six fields, as shown in Figure 11.27: a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field. In multiple-frame transmissions, the ending flag of one frame can serve as the beginning flag of the next frame.

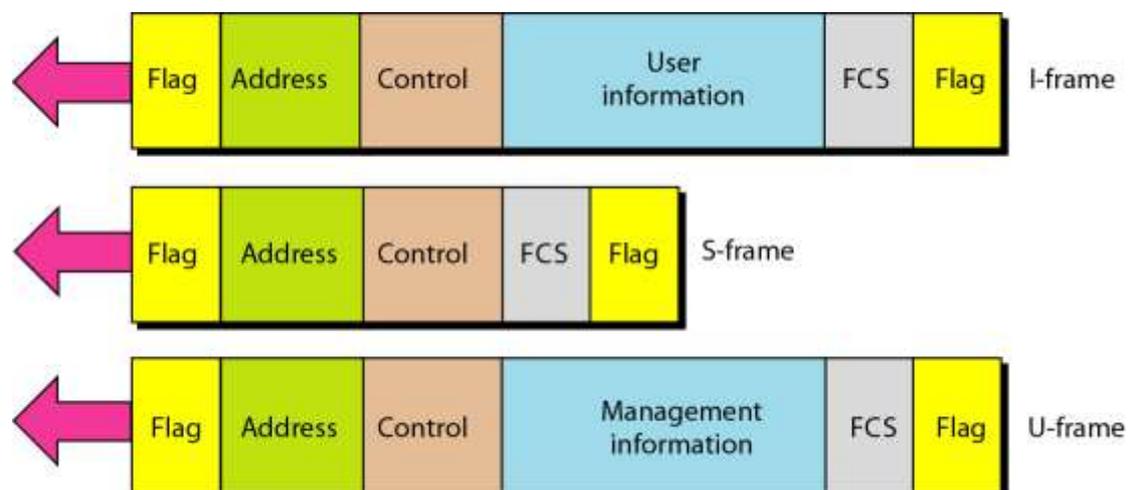


Figure 11.27 *HDLC frames*

POINT-TO-POINT PROTOCOL

Although HDLC is a general protocol that can be used for both point-to-point and multipoint configurations, one of the most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). Today, millions of Internet users who need to connect their home computers to the server of an Internet service provider use PPP. The majority of these users have a traditional modem; they are connected to the Internet through a telephone line, which provides the services of the physical layer. But to control and manage the transfer of data, there is a need for a point-to-point protocol at the data link layer. PPP is by far the most common. PPP provides several services:

1. PPP defines the format of the frame to be exchanged between devices.
2. PPP defines how two devices can negotiate the establishment of the link and the exchange of data.
3. PPP defines how network layer data are encapsulated in the data link frame.
4. PPP defines how two devices can authenticate each other.
5. PPP provides multiple network layer services supporting a variety of network layer protocols.
6. PPP provides connections over multiple links.
7. PPP provides network address configuration. This is particularly useful when a home

user needs a temporary network address to connect to the Internet.

On the other hand, to keep PPP simple, several services are missing:

- I. PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
2. PPP has a very simple mechanism for error control. A CRC field is used to detect errors. If the frame is corrupted, it is silently discarded; the upper-layer protocol needs to take care of the problem. Lack of error control and sequence numbering may cause a packet to be received out of order.

3. PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

Framing

PPP is a byte-oriented protocol. Framing is done according to the discussion of byte-oriented protocols at the beginning of this chapter.

Frame Format

Figure 11.32 shows the format of a PPP frame. The description of each field follows:

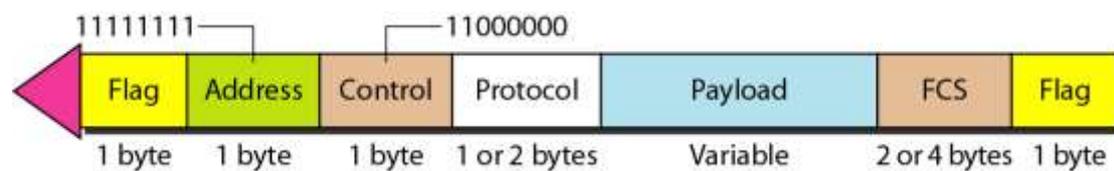


Figure 11.32 PPP frame format

Flag: A PPP frame starts and ends with a 1-byte flag with the bit pattern 01111110. Although this pattern is the same as that used in HDLC, there is a big difference. PPP is a byte-oriented protocol; HDLC is a bit-oriented protocol. The flag is treated as a byte, as we will explain later.

Address: The address field in this protocol is a constant value and set to 11111111 (broadcast address). During negotiation (discussed later), the two parties may agree to omit this byte.

Control: This field is set to the constant value 11000000 (imitating unnumbered frames in HDLC). As we will discuss later, PPP does not provide any flow control. Error control is also limited to error detection. This means that this field is not needed at all, and again, the two parties can agree, during negotiation, to omit this byte.

Protocol: The protocol field defines what is being carried in the data field: either user data or other information. We discuss this field in detail shortly. This field is by default 2 bytes long, but the two parties can agree to use only 1 byte.

Payload field: This field carries either the user data or other information. The data field is a sequence of bytes with the default of a maximum

of 1500 bytes; but this can be changed during negotiation. The data field is bytestuffed if the flag byte pattern appears in this field. Because there is no field defining the size of the data field, padding is needed if the size is less than the maximum default value or the maximum negotiated value.

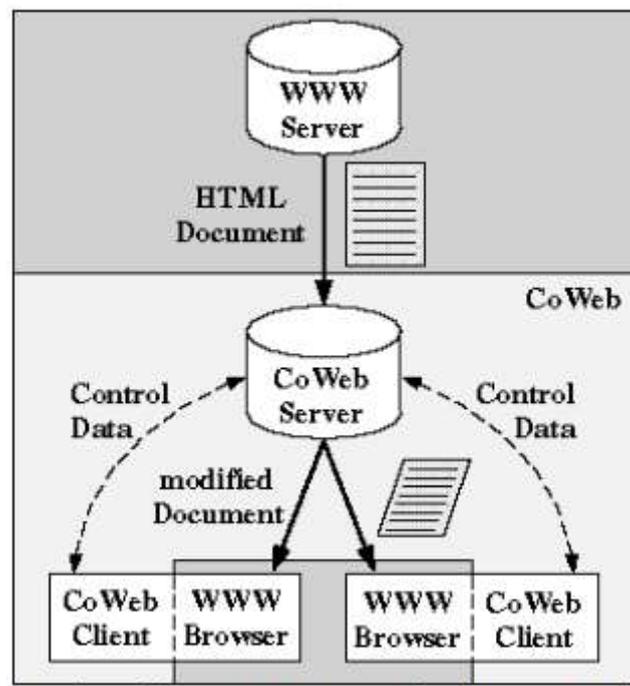
FCS: The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC.

WWW and HTTP

The **World Wide Web** (WWW) is a repository of information linked together from points all over the world. The WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.

ARCHITECTURE

The WWW today is a distributed client -server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called *sites*, as shown in Figure 27.1.



Each site holds one or more documents, referred to as *Web pages*. Each Web page can contain a link to other pages in the same site or at other sites. The pages can be retrieved and viewed by using browsers.

Client (Browser)

A variety of vendors offer commercial browsers that interpret and display a Web document, and all use nearly the same architecture. Each browser usually consists of three parts: a controller, client protocol, and interpreters. The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. The client protocol can be one of the protocols described previously such as FTP or. The interpreter can be HTML, Java, or JavaScript, depending on the type of document. We discuss the use.

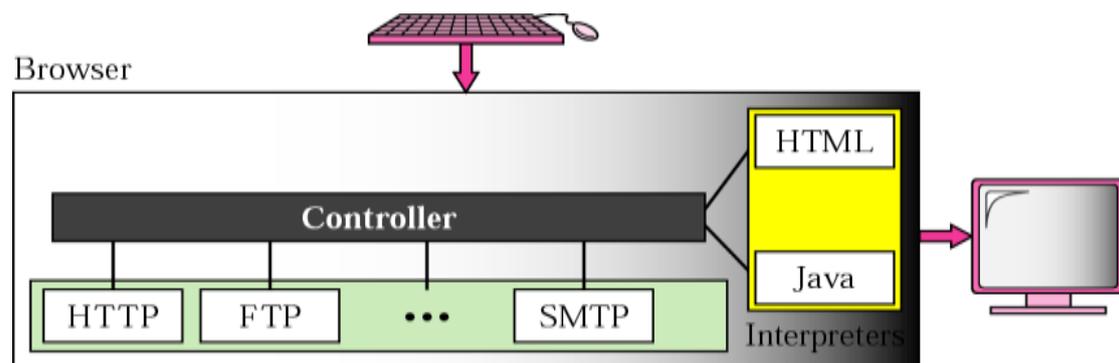


Figure 27.13 Browser architecture

HTTP

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to FTP because it transfers files and uses the services of TCP. However, it is much simpler than FTP because it uses only one TCP connection. There is no separate control connection; only data are transferred between the client and the server.

HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like headers. Unlike SMTP, the HTTP

messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser).

Message Transfer Agent: SMTP

The actual mail transfer is done through message transfer agents. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called the Simple Mail Transfer Protocol (SMTP). As we said before, two pairs of MTA client/server programs are used in the most common situation (fourth scenario). Figure 26.16 shows the range of the SMTP protocol in this scenario.

SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. As we will see shortly, another protocol is needed between the mail server and the receiver. SMTP simply defines how commands and responses must be sent back and forth. Each network is free to choose a software package for implementation. We discuss the mechanism of mail transfer by SMTP in the remainder of the section.

POP3

Post Office Protocol, version 3 (POP3) is simple and limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server. Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one. Figure 26.20 shows an example of downloading using POP3. POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval. The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying. The keep mode is normally used when the user accesses her

mail away from her primary computer (e.g., a laptop). The mail is read but kept in the system for later retrieval and organizing.

IMAP4

Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.

POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. (Of course, the user can create folders on her own computer.) In addition, POP3 does not allow the user to partially check the contents of the mail before downloading. IMAP4 provides the following extra functions:

- o A user can check the e-mail header prior to downloading.
- o A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- o A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- o A user can create, delete, or rename mailboxes on the mail server.
- o A user can create a hierarchy of mailboxes in a folder for e-mail storage.

FILE TRANSFER

Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment. As a matter of fact, the greatest volume of data exchange in the Internet today is due to file transfer. In this section, we discuss one popular protocol involved in transferring files: File Transfer Protocol (*FTP*).

File Transfer Protocol (FTP)

File Transfer Protocol (FTP) is the standard mechanism provided by *TCP/IP* for copying a file from one host to another. Although transferring

files from one system to another seems simple and straightforward, some problems must be dealt with first. For example, two systems may use different file name conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. All these problems have been solved by FTP in a very simple and elegant approach.

FTP differs from other client/server applications in that it establishes two connections between the hosts. One connection is used for data transfer, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient. The control connection uses very simple rules of communication.

We need to transfer only a line of command or a line of response at a time. The data connection, on the other hand, needs more complex rules due to the variety of data types transferred. However, the difference in complexity is at the FTP level, not TCP. For TCP, both connections are treated the same.

FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection.