Unit I
Chapter 1

Reference: Data Communication and Networking, Behrouz A Forouzan**,**
McGraw Hill, 5th Edition,
2008.

1. Data Communications

- The term ***telecommunication,*** which includes telephony, telegraphy, and television, means communication at a distance.

- 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. Data Communications contd.

- **Fundamental Characterstics :**
  - **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.
  - **Accuracy.** The system must deliver the data accurately. Data that have been **altered** in transmission and left uncorrected are unusable.
  - **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.**
  - **Jitter**. Jitter refers to the **variation in the packet arrival time**. It is the **uneven delay** in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

1.1.1 Components

- A data communications system has five components:
- **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.

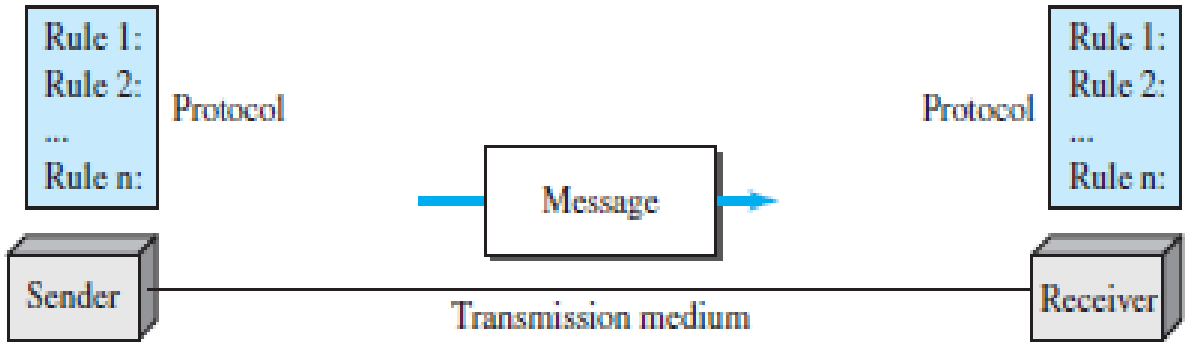
- **3. Receiver.** The receiver is the device that **receives the message**. It can be a computer, workstation, telephone handset, television, and so on.

- **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.

1.1.1 Components Contd.

Figure 1.1 *Five components of data communication*



1.1.2 Data Representation

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

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

- ***Numbers***
  - Numbers are also represented by bit patterns.
  - However, a code such as ASCII is not used to represent numbers; the number is directly converted to a **binary number** to simplify mathematical operations.

1.1.2 Data Representation Contd.

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

1.1.2 Data Representation Contd.

- For an image made of only **black-and-white dots** (e.g., a chessboard), a **1-bit pattern** is enough to represent a pixel.

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

- There are several methods to represent color images. One method is called **RGB**, so called because each color is made of a combination of three primary colors: **red, green, and blue.** The intensity of each color is measured, and a bit pattern is assigned to it.

- Another method is called **YCM**, in which a color is made of a combination of three other primary colors: **yellow, cyan, and magenta**.

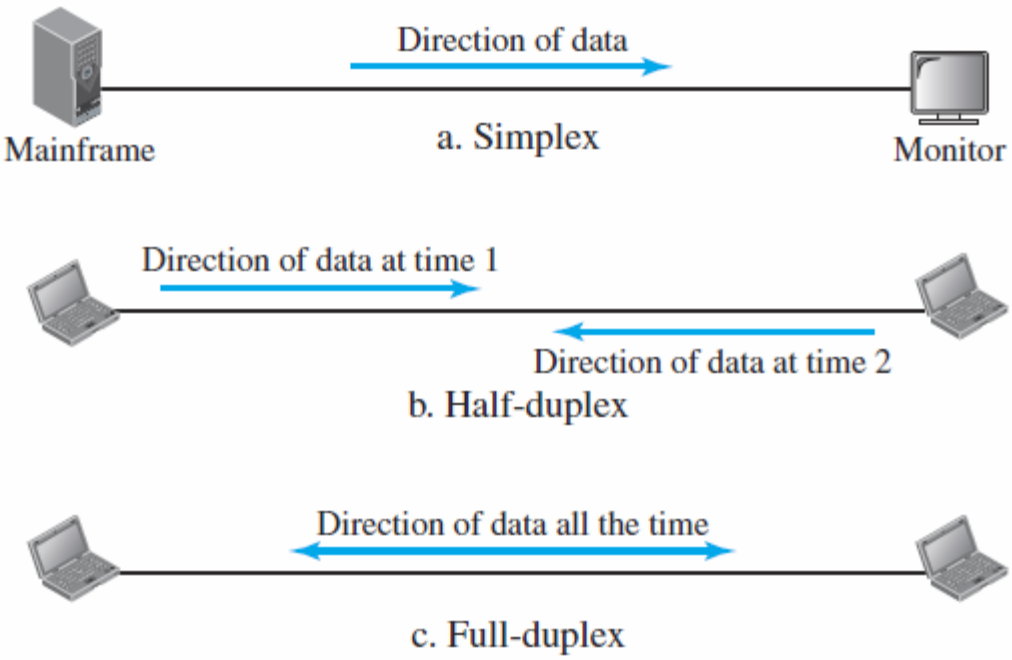
1.1.2 Data Representation Contd.

- ***Audio***
  - Audio refers to the recording or broadcasting of sound or music.
  - Audio is by nature different from text, numbers, or images.
  - It is **continuous**, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal.
- ***Video***
  - Video refers to the recording or broadcasting of a picture or movie.
  - Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion.

1.1.3 Data Flow

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

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



1.1.3 Data Flow Contd.

- ***Simplex***
  - In **simplex mode, the communication is unidirectional.**
  - **Only one** of the two devices on a link **can transmit**; the other can only receive.
  - **Keyboards and traditional monitors** are examples of simplex devices.
  - The keyboard can only introduce input; the monitor can only accept output.
  - The simplex mode can **use the entire capacity of the channel** to send data in one direction.

1.1.3 Data Flow Contd.

- ***Half-Duplex***
  - In **half-duplex mode, each station can both transmit and receive, but not at the same time.**
  - When one device is sending, the other can only receive, and vice versa (see Figure 1.2b).
  - In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time.
  - Walkie-talkies and CB (citizens band) radios are both half-duplex systems.
  - The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

1.1.3 Data Flow Contd.

- ***Full-Duplex***
  - In **full-duplex mode (also called *duplex), both stations can transmit and receive simultaneously*** (see Figure 1.2c).
  - In full-duplex mode, signals going in one direction **share the capacity** of the link with signals going in the other direction.
  - This sharing can occur in **two ways**:
    - Either the link must **contain two physically separate transmission paths**, one for sending and the other for receiving; or the **capacity of the channel is divided** between signals traveling in both directions.
  - One common example of full-duplex communication is the **telephone network.**
  - The full-duplex mode is used when communication in both directions is required all the time.

1.2 NETWORKS

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

1.2.1 Network Criteria

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

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

1.2.1 Network Criteria

- Performance is often evaluated by two networking metrics: **throughput and delay.**
- We often need **more throughput and less delay**.

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

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

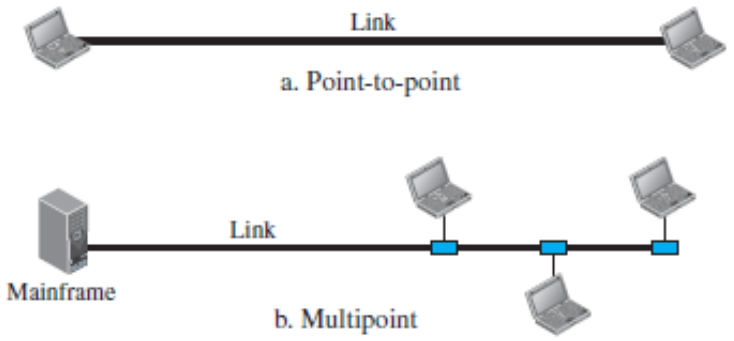
1.2.2 Physical Structures

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

1.2.2 Physical Structures Contd.

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



1.2.2 Physical Structures Contd.

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

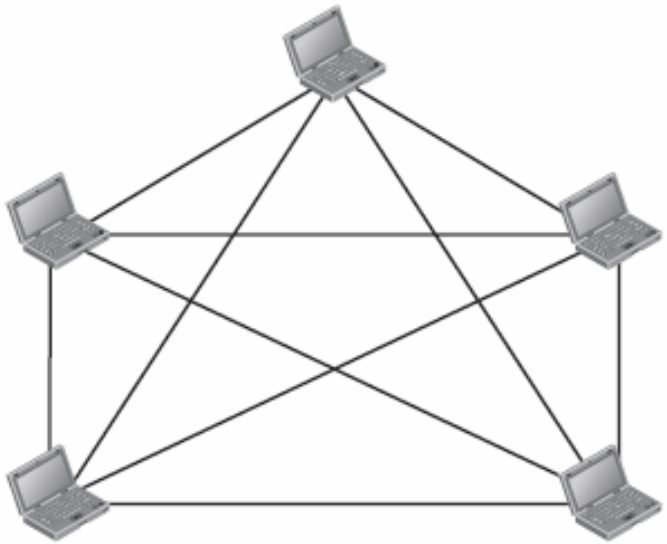
1.2.2 Physical Structures Contd.

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

1.2.2 Physical Structures Contd.

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

$n = 5$
10 links.



1.2.2 Physical Structures Contd.

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

1.2.2 Physical Structures Contd.

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

1.2.2 Physical Structures Contd.

- ***Star Topology***
- In a star topology, **each device has a dedicated point-to-point link only to a central controller,** usually called a **hub.**

- The devices are **not directly linked to one another.** Unlike a mesh topology, a star topology **does not allow direct traffic** between devices.

- The controller acts as an **exchange**: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device (see Figure 1.5) .

- Advantages

  - A star topology is **less expensive** than a mesh topology.

  - In a star, each device needs **only one link and one I/O port** to connect it to any number of others.

  - This factor also makes it **easy to install and reconfigure**.

  - Far **less cabling** needs to be housed, and additions, moves, and deletions involve only one connection: between that device and the hub.

1.2.2 Physical Structures Contd.

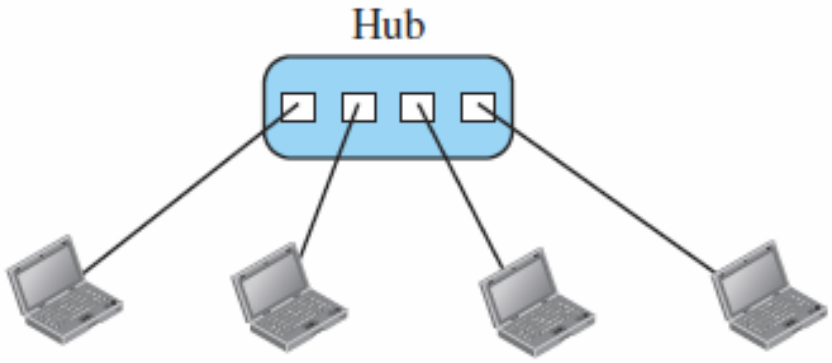
  - It is robust i.e If one link fails, only that link is affected, all other links remain active.
  - This factor also lends itself to easy fault identification and fault isolation. As long as the hub is working, it can be used to monitor link problems and bypass defective links.

- Disadvantage of a star topology
  - Dependency of the whole topology on one single point, the hub.
  - If the hub goes down, the whole system is dead.
  - Although a star requires far less cable than a mesh, each node must be linked to a central hub.
  - For this reason, often **more cabling** is required in a star than in some other topologies (such as **ring or bus**).

- The star topology is used in local-area networks (LANs)
- High-speed LANs often use a star topology with a central hub.

1.2.2 Physical Structures Contd.

Figure 1.5 *A star topology connecting four stations*



1.2.2 Physical Structures Contd.

- ***Bus Topology***
- A **bus topology,** is multipoint.
- One long cable acts as a **backbone to link all the devices in a** network (see Figure 1.6).
- Nodes are connected to the bus cable by **drop lines and taps**.
- A **drop line** is a **connection running between the device and the main cable**.
- A **tap** is a connector that either **splices into the main cable** or punctures the sheathing of a cable to create a contact with the metallic core.
- As a signal travels along the backbone, **some of its energy is transformed into heat**.
- Therefore, it becomes **weaker and weaker** as it travels farther and farther.
- For this reason there is a **limit on the number of taps** a bus can support and on the distance between those taps.

1.2.2 Physical Structures Contd.

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

1.2.2 Physical Structures Contd.

- A bus is usually designed to be optimally efficient at installation. It can therefore **be difficult to add new devices.**

- **Signal reflection** at the taps can cause degradation in quality. This degradation can be controlled by limiting the number and spacing of devices connected to a given length of cable.

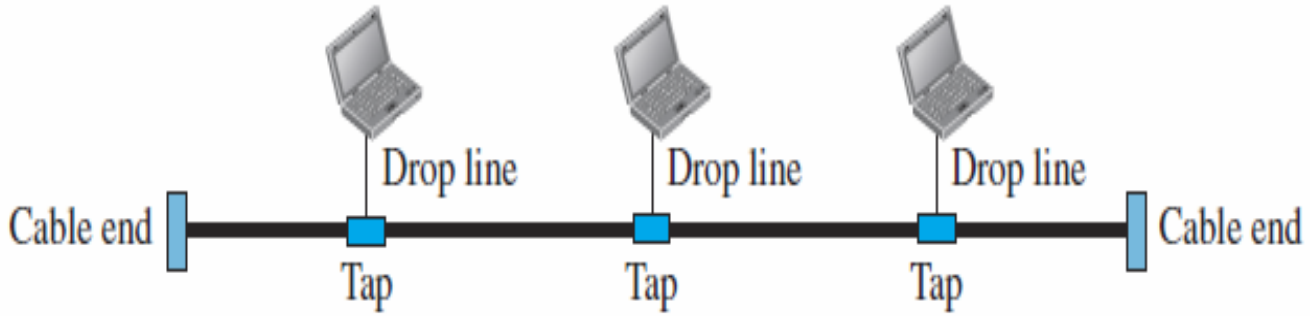
- In addition, a **fault or break in the bus cable stops all transmission,** even between devices on the same side of the problem.

- The **damaged area reflects signals back** in the direction of origin, creating noise in both directions.

- Bus topology was the one of the first topologies used in the design of early local area networks.

1.2.2 Physical Structures Contd.

Figure 1.6 *A bus topology connecting three stations*



1.2.2 Physical Structures Contd.

- ***Ring Topology***

- In a ring topology, **each device has a dedicated point-to-point connection with only the two devices on either side of it.**

- A signal is passed along the ring in **one direction**, from device to device, until it reaches its destination.

- Each device in the ring incorporates a **repeater**.

- When a device receives a signal intended for another device, its repeater **regenerates the bits** and passes them along (see Figure 1.7).

- A ring is relatively **easy to install and reconfigure**.

- Each device is **linked to only its immediate neighbors** (either physically or logically).

1.2.2 Physical Structures Contd.

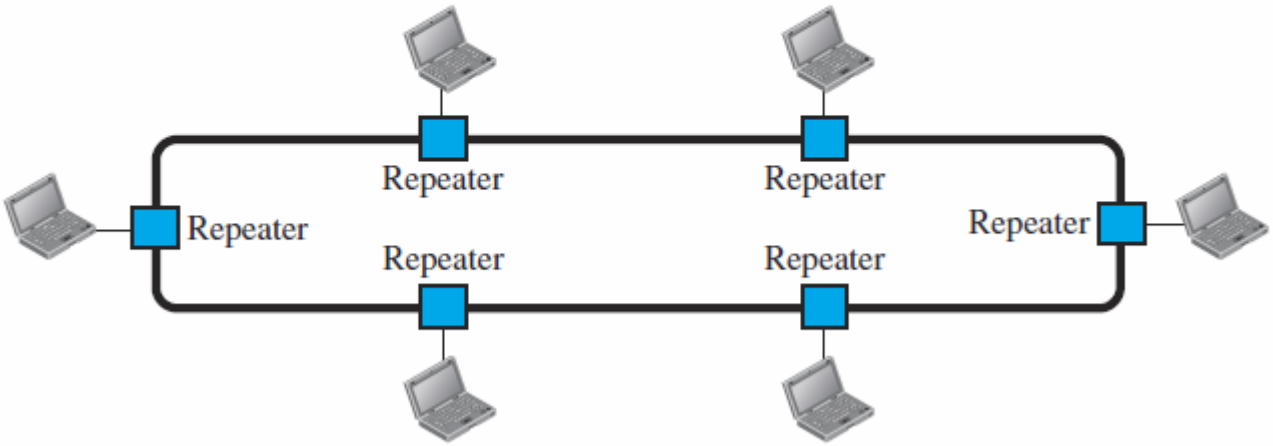
- To add or delete a device requires **changing only two connections**.
- The only constraints are **media and traffic considerations** (maximum ring length and number of devices).
- In addition, **fault isolation is simplified**.
- Generally, in a ring a signal is circulating at all times. If one device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

1.2.2 Physical Structures Contd.

- However, unidirectional traffic can be a disadvantage.
- In a simple ring, a **break in the ring** (such as a disabled station) can disable the entire network.
- This weakness can be **solved by using a dual ring** or a switch capable of closing off the break.
- Ring topology was prevalent when IBM introduced its local-area network, **Token Ring**.
- Today, the need for higher-speed LANs has made this topology less popular.

1.2.2 Physical Structures Contd.

Figure 1.7 *A ring topology connecting six stations*



1.3 NETWORK TYPES

- Criteria of distinguishing one type of network from another is size, geographical coverage, and ownership.
  - Local Area Network
  - Wide Area Network
  - Switching
  - The Internet
  - Accessing the Internet

1.3.1 Local Area Network

- A local area network (LAN) is usually **privately owned** and connects some hosts in a **single office, building, or campus.**

- Depending on the needs of an organization, a LAN can be as simple as two PCs and a printer in someone's **home office**, or it can extend throughout a company and include audio and video devices.

- Each host in a LAN has an identifier, **an address**, that uniquely defines the host in the LAN.

- A packet sent by a host to another host carries both the **source host's and the destination host's addresses**.

1.3.1 Local Area Network Contd.

- In the **past**, all hosts in a network were connected through a common cable, which meant that a packet sent from one host to another was **received by all hosts**. The intended recipient kept the packet; the others dropped the packet.

- Today, most LANs use a smart connecting switch, which is able to **recognize the destination address** of the packet and guide the packet to its destination without sending it to all other hosts.

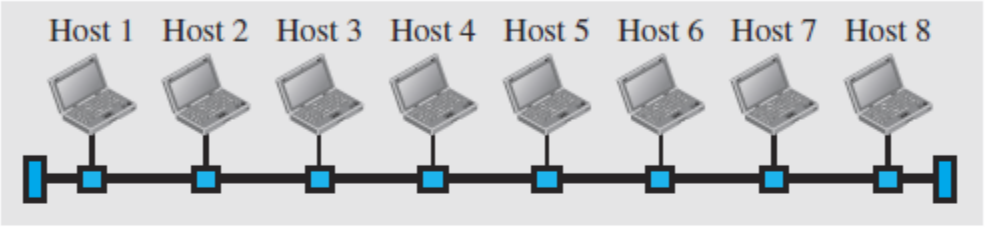
- The **switch** alleviates the traffic in the LAN and **allows more than one pair to communicate with each other at the same time** if there is no common source and destination among them.

- Note that the above definition of a LAN does not define the minimum or maximum number of hosts in a LAN.

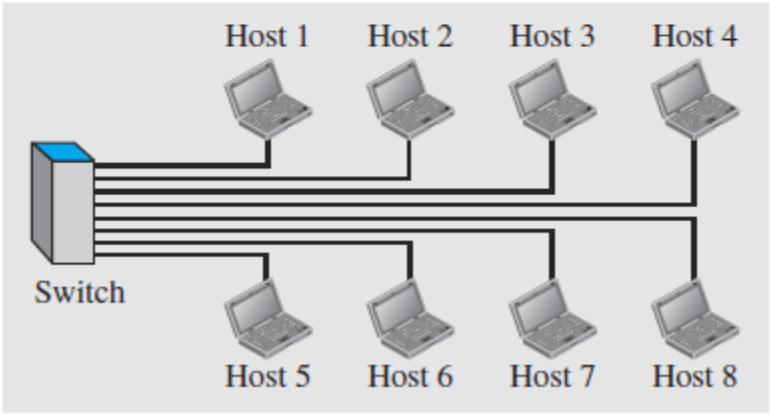
- Figure 1.8 shows a LAN using either a common cable or a switch.

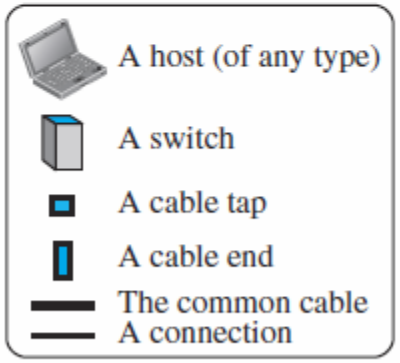
1.3.1 Local Area Network Contd.

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



a. LAN with a common cable (past)





b. LAN with a switch (today)

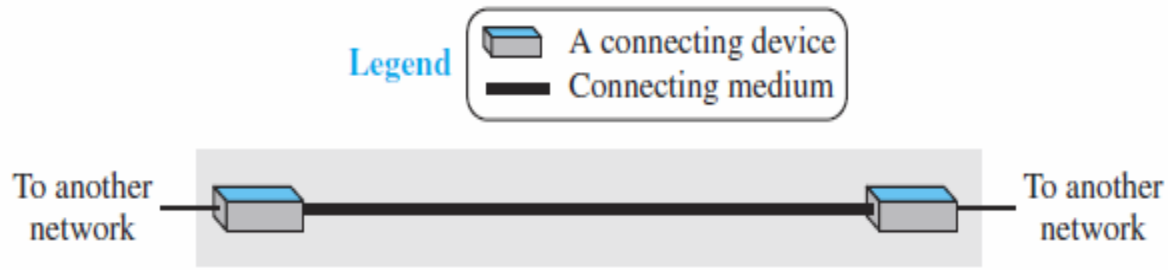
1.3.2 Wide Area Network

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

1.3.2 Wide Area Network Contd.

- ***Point-to-Point WAN***
- A point-to-point WAN is a network that connects two communicating devices through a transmission media (cable or air).
- Figure 1.9 shows an example of a point-to-point WAN.

Figure 1.9 *A point-to-point WAN*



1.3.2 Wide Area Network Contd.

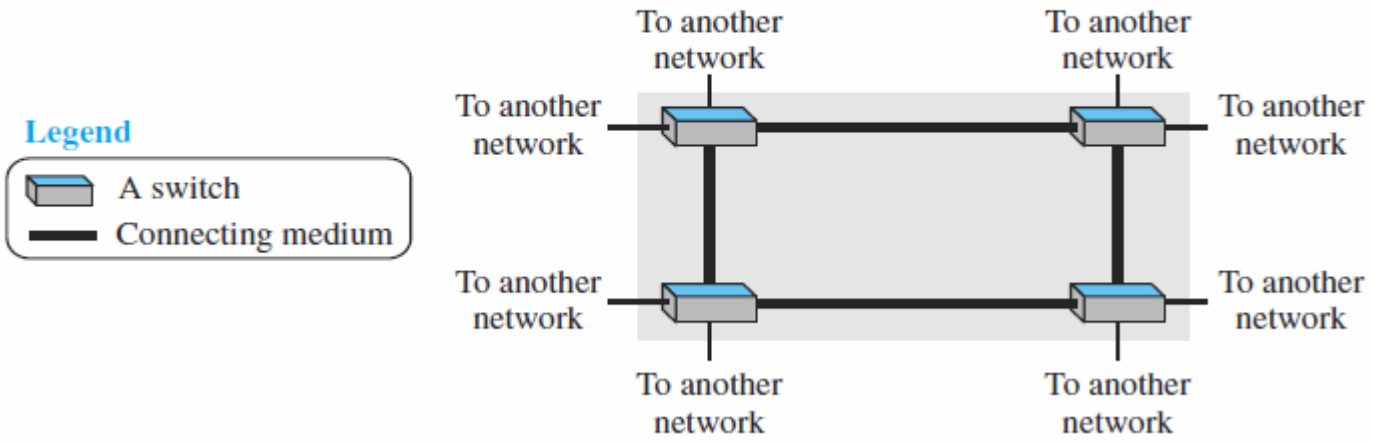
- ***Switched WAN***
- A switched WAN is a network with more than two ends.

- A switched WAN, is used in the backbone of global communication today.

- We can say that a switched WAN is a combination of several point-to-point WANs that are connected by switches.

- Figure 1.10 shows an example of a switched WAN.

Figure 1.10 *A switched WAN*



1.3.2 Wide Area Network Contd.

- ***Internetwork***
- When **two or more networks are connected**, they make an internetwork, or internet.

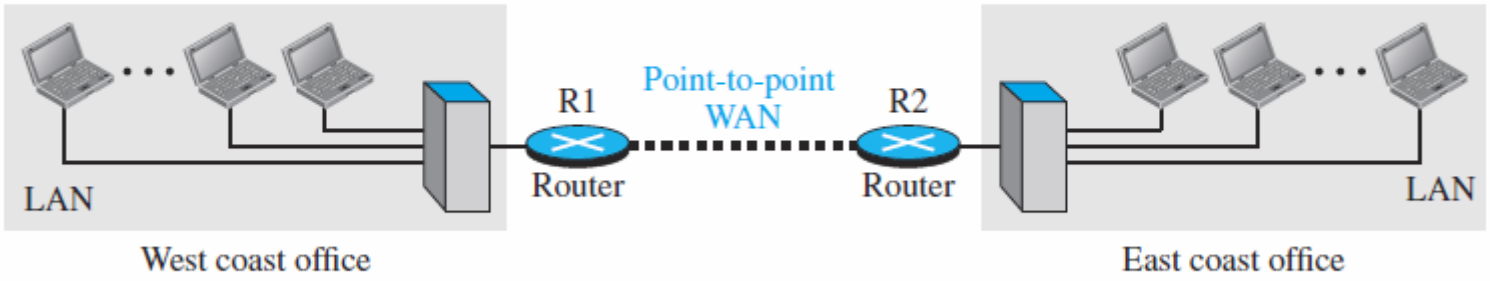
- As an example, assume that an organization has two offices, one on the east coast and the other on the west coast. Each office has a LAN that allows all employees in the office to communicate with each other.

- To make the communication between employees at different offices possible, the management leases a point-to-point dedicated WAN from a service provider, such as a telephone company, and connects the two LANs.

- Now the company has an internetwork, or a **private internet**. *Communication* between offices is now possible. Figure 1.11 shows this internet.

1.3.2 Wide Area Network Contd.

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



1.3.2 Wide Area Network Contd.

- When a host in the west coast office sends a message to another host in the same office, the router blocks the message, but the switch directs the message to the destination.

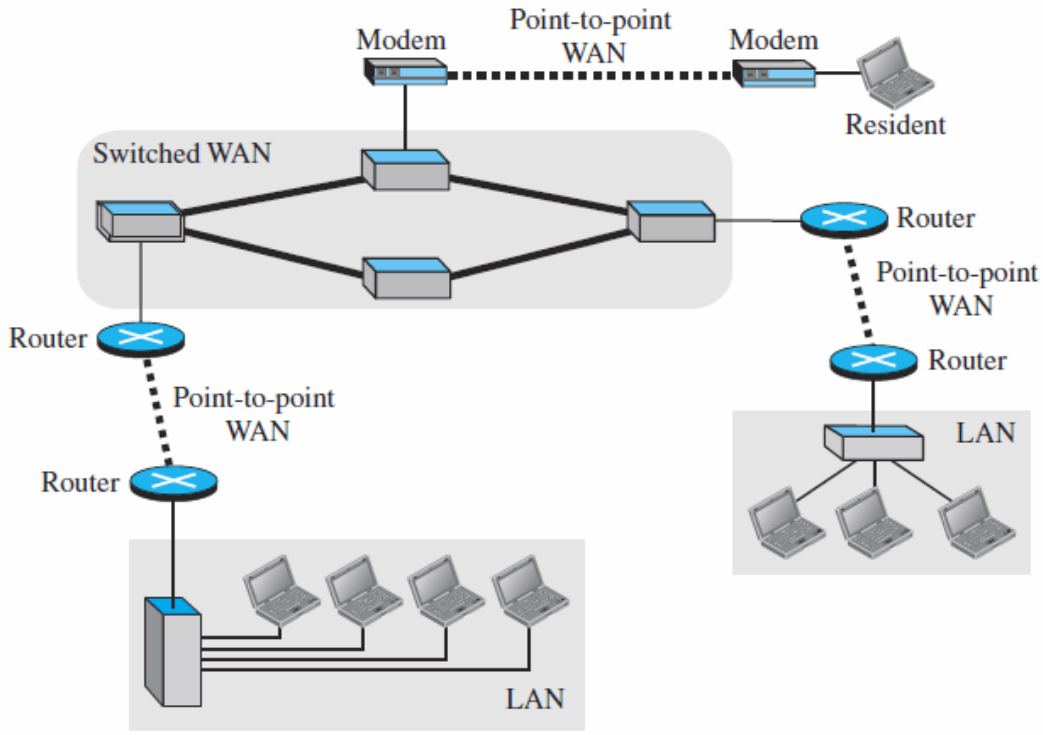
- On the other hand, when a host on the west coast sends a message to a host on the east coast, router R1 routes the packet to router R2, and the packet reaches the destination.

- Figure 1.12 shows another internet with several LANs and WANs connected.

- One of the WANs is a switched WAN with four switches.

1.3.2 Wide Area Network Contd.

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



1.3.3 Switching

- An internet is a switched network in which a switch connects at least two links together.

- A switch needs to **forward data from a network** to another network when required.

- The two most common types of switched networks are **circuit-switched and packet-switched networks.**

- ***Circuit-Switched Network***
- In a circuit-switched network, **a dedicated connection**, called a **circuit**, is always available between the two end systems; the switch can only make it active or inactive.

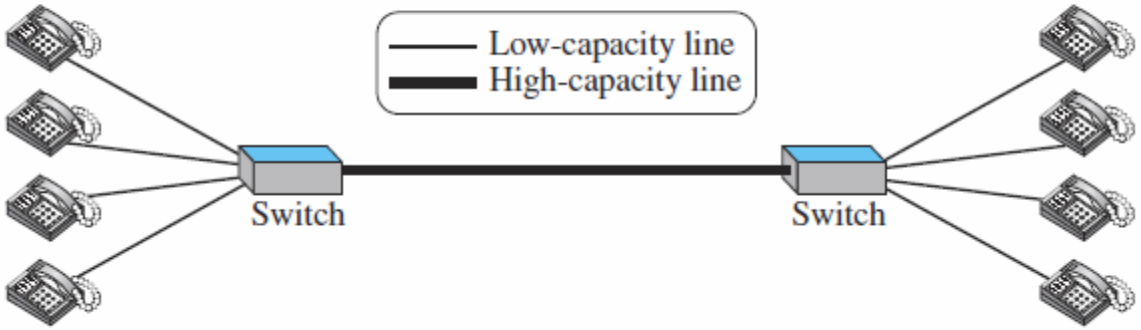
- Figure 1.13 shows a very simple switched network that connects four telephones to each end.

- We have used telephone sets instead of computers as an end system because circuit switching was very common in telephone networks in the past, although part of the telephone network today is a packet-switched network.

1.3.3 Switching Contd.

- In Figure 1.13, the four telephones at each side are connected to a switch.
- The switch connects a telephone set at one side to a telephone line connecting two switches is a high-capacity communication line that can handle four voice communications at the same time; the capacity can be shared between all pairs of telephone sets.
- The switches used in this example have forwarding tasks but no storing capability.

Figure 1.13 *A circuit-switched network*



1.3.3 Switching Contd.

- Let us look at two cases.
  - In the first case, all telephone sets are busy; four people at one site are talking with four people at the other site; the capacity of the thick line is fully used.
  - In the second case, only one telephone set at one side is connected to a telephone set at the other side; only one-fourth of the capacity of the thick line is used.

- This means that a circuit-switched network is **efficient only when it is working at its full capacity;** most of the time, it is inefficient because it is working at partial capacity.

- The reason that we need to make the capacity of the thick line four times the capacity of each voice line is that we do not want communication to fail when all telephone sets at one side want to be connected with all telephone sets at the other side.

1.3.3 Switching Contd.

- ***Packet-Switched Network***
- In a computer network, the communication between the two ends is done in blocks of data called **packets**.

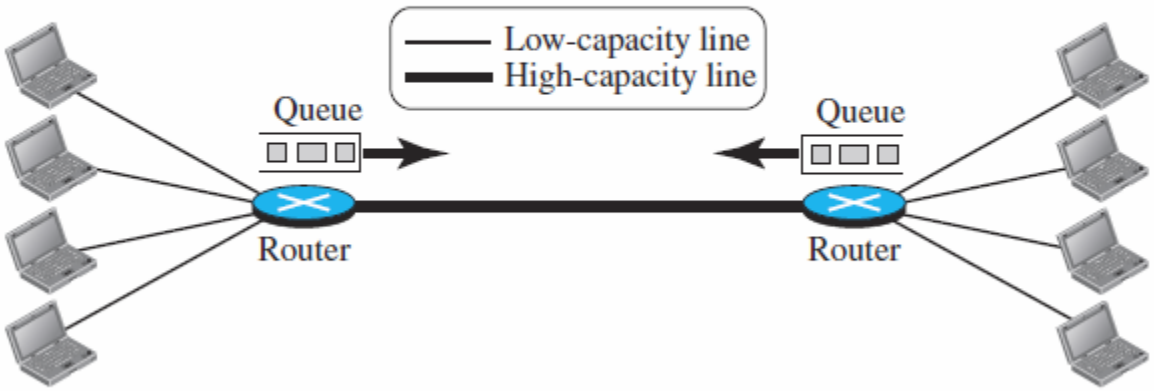
- In other words, instead of the continuous communication we see between two telephone sets when they are being used, we see the **exchange of individual data packets** between the two computers.

- This allows us to make the switches function for both **storing and forwarding** because a packet is an independent entity that can be stored and sent later.

- Figure 1.14 shows a small packet-switched network that connects four computers at one site to four computers at the other site.

1.3.3 Switching Contd.

Figure 1.14 *A packet-switched network*



1.3.3 Switching Contd.

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

1.3.4 The Internet

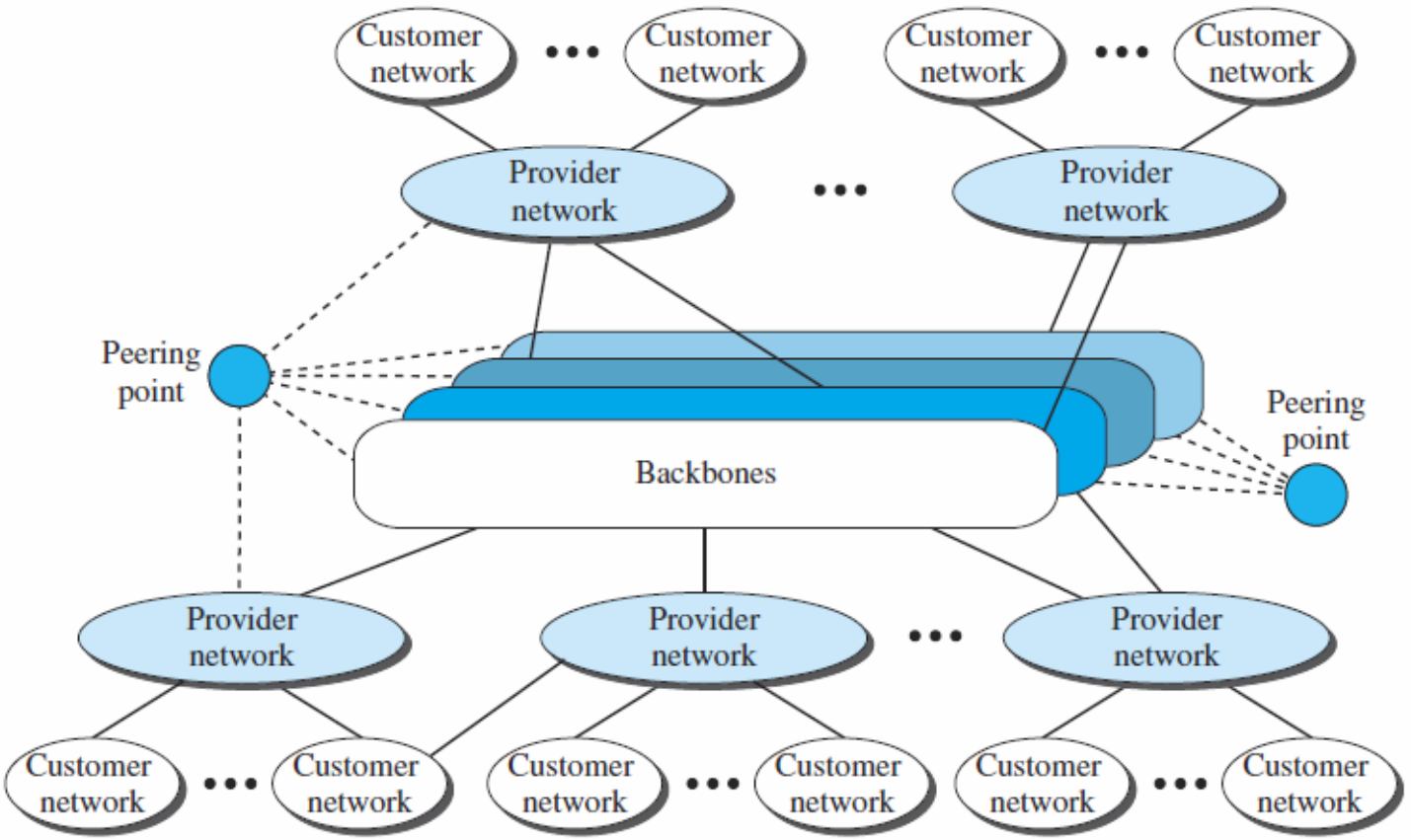
- An **internet** (lowercase i), is two or more networks that can communicate with each other.
- The most notable internet is called the **Internet** (uppercase I), and is composed of thousands of interconnected networks.
- Figure 1.15 shows a conceptual (not geographical) view of the Internet.
- The figure shows the Internet as several **backbones, provider networks, and customer networks.**
- At the top level, the **backbones are large networks owned by some communication companies** such as Sprint, Verizon (MCI), AT&T, and NTT.
- The backbone networks are connected through some **complex switching systems, called peering points**.

1.3.4 The Internet Contd.

- At the **second level,** there are smaller networks, called **provider networks**, that use the services of the backbones for a fee.
- The provider networks are **connected to backbones** and sometimes to **other provider networks**.
- The **customer networks** are networks at the edge of the Internet that actually use the services provided by the Internet.
- They **pay fees to provider networks** for receiving services.
- Backbones and provider networks are also called **Internet Service Providers (ISPs).**
- **The backbones** are often referred to as **international ISPs; the provider networks** are often referred to as **national or regional ISPs**.

1.3.4 The Internet Contd.

Figure 1.15 *The Internet today*



1.3.4 The Internet Contd.

- The Internet today is an internetwork that allows **any user to become part of it.**

- The **user**, however, needs to be **connected to an ISP.**

- ***Using Telephone Networks***
- Today most residences and small businesses have telephone service, which means they are **connected to a telephone network**.

- Since most telephone networks have already connected themselves to the Internet, one option for residences and small businesses to **connect to the Internet is to change the voice line** between the residence or business and the telephone center to a point-to-point WAN.

- This can be done in **two ways**.

1.3.4 The Internet Contd.

- ❑ **Dial-up service.**
  - The first solution is to add to the telephone line a **modem** that converts data to voice.
  - The **software installed on the computer dials the ISP** and imitates making a telephone connection.
  - Unfortunately, the dial-up service is **very slow**, and when the line is used for Internet connection, it **cannot be used for telephone (voice) connection**. It is only useful for small residences.
- ❑ **DSL Service.**
  - Since the advent of the Internet, some telephone companies have upgraded their telephone lines to provide **higher speed Internet services** to residences or small businesses.
  - The DSL service also allows the line to be used **simultaneously for voice and data communication.**

1.3.4 The Internet Contd.

- ***Using Cable Networks***
  - A residence or a small business can be **connected to the Internet** by using **cable service**.
  - It provides a higher speed connection, but the **speed varies depending on the number of neighbors** that use the same cable.
- ***Using Wireless Networks***
  - A household or a small business can use a combination of **wireless and wired connections** to access the Internet.
  - With the growing wireless WAN access, a household or a small business can be connected to the Internet through a **wireless WAN**.
- ***Direct Connection to the Internet***
  - A **large organization** or a large corporation can itself become a **local ISP** and be connected to the Internet.
  - This can be done if the organization or the corporation **leases a high-speed WAN** from a carrier provider and connects itself to a regional ISP.
  - For example, a large university with several campuses can create an internetwork and then connect the internetwork to the Internet

Practice Questions

- Identify the five components of a data communications system.
- What are the three criteria necessary for an effective and efficient network?
- What are the advantages of a multipoint connection over a point-to-point one?
- What are the two types of line configuration?
- Categorize the four basic topologies in terms of line configuration.
- What is the difference between half-duplex and full-duplex transmission modes?
- Name the four basic network topologies, and cite an advantage of each type.
- In a LAN with a link-layer switch (Figure 1.8b), Host 1 wants to send a message to Host 3. Since communication is through the link-layer switch, does the switch need to have an address? Explain.

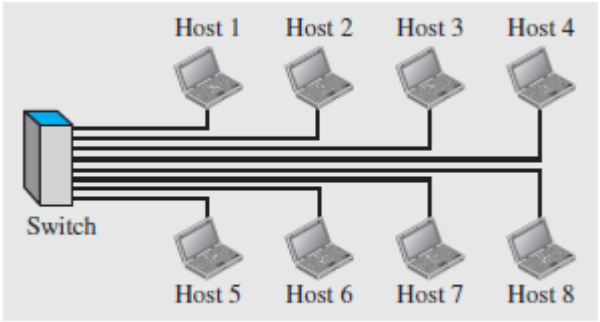


Figure 1.8 b. LAN with a switch (today)

Practice Questions Contd.

- For n devices in a network, what is the number of cable links required for a mesh, ring, bus, and star topology?
- What are some of the factors that determine whether a communication system is a LAN or WAN?
- What is an internet? What is the Internet?
- Why are protocols needed?
- How many point-to-point WANs are needed to connect *n LANs if each LAN* should be able to directly communicate with any other LAN?
- When we use local telephones to talk to a friend, are we using a circuit switched network or a packet-switched network?
- When a resident uses a dial-up or DLS service to connect to the Internet, what is the role of the telephone company?
- What is the first principle we discussed in this chapter for protocol layering that needs to be followed to make the communication bidirectional?

Figure 1.8

Problems

- What is the maximum number of characters or symbols that can be represented by Unicode?
- A color image uses 16 bits to represent a pixel. What is the maximum number of different colors that can be represented?
- Assume six devices are arranged in a mesh topology. How many cables are needed? How many ports are needed for each device?
- For each of the following four networks, discuss the consequences if a connection fails.
  - a. Five devices arranged in a mesh topology
  - b. Five devices arranged in a star topology (not counting the hub)
  - c. Five devices arranged in a bus topology
  - d. Five devices arranged in a ring topology
- We have two computers connected by an Ethernet hub at home. Is this a LAN or a WAN? Explain the reason.

Figure 1.8

Problems Contd.

- In the ring topology, what happens if one of the stations is unplugged?
- In the bus topology, what happens if one of the stations is unplugged?
- Performance is inversely related to delay. When we use the Internet, which of the following applications are more sensitive to delay?
  - a. Sending an e-mail
  - b. Copying a file
  - c. Surfing the Internet
- When a party makes a local telephone call to another party, is this a point-to-point or multipoint connection? Explain the answer.
- Compare the telephone network and the Internet. What are the similarities? What are the differences?

Figure 1.8

END of Chapter 1

Unit I
Chapter 2

Reference: Data Communication and Networking, Behrouz A Forouzan**,**
McGraw Hill, 5th Edition,
2008.

2.1 PROTOCOL LAYERING

- In data communication and networking, a **protocol defines the rules that both the sender and receiver and all intermediate devices need to follow** to be able to communicate effectively.

- When communication is **simple**, we may need only one **simple protocol**; when the communication is **complex**, we may need to divide the task between different layers, in which case we need a protocol at each layer, or **protocol layering.**

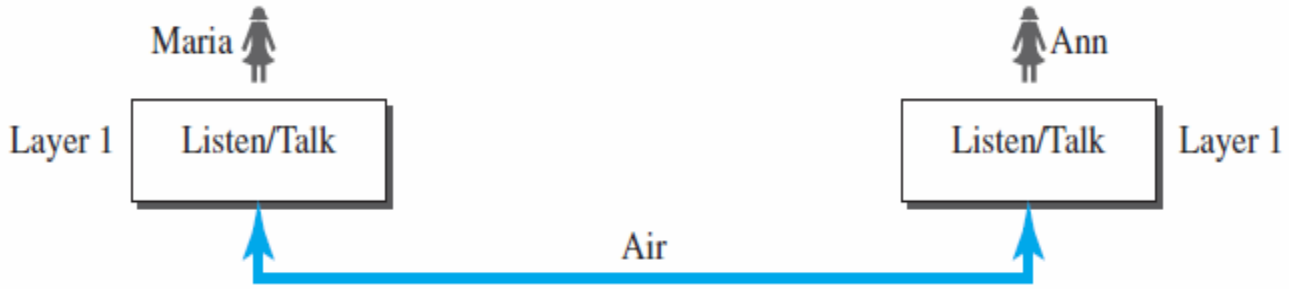
2.1.1 Scenarios

- ***First Scenario***
- In the first scenario, communication is so **simple** that it can occur in only one layer.

- Assume Maria and Ann are **neighbors** with a lot of common ideas.

- Communication between Maria and Ann takes place in **one layer, face to face**, in the same language, as shown in Figure 2.1.

Figure 2.1 *A single-layer protocol*



2.1.1 Scenarios Contd.

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

2.1.1 Scenarios Contd.

- We can see that the protocol used by Maria and Ann is **different** from the **communication** between a **professor and the students** in a lecture hall.

- The communication in the second case is mostly **monolog**; the **professor talks most of the time** unless a student has a question, a situation in which the protocol dictates that she should **raise his/her hand** and wait for **permission to speak**.

- In this case, the communication is normally very **formal** and limited to the subject being taught.

2.1.1 Scenarios Contd.

- ***Second Scenario***
- In the second scenario, we assume that Ann is offered a higher-level position in her company, but needs to **move to another branch** located in a city very far from Maria.

- The two friends still want to continue their communication and **exchange ideas** because they have come up with an **innovative project to start a new business** when they both retire.

- They decide to continue their conversation using **regular mail** through the post office.

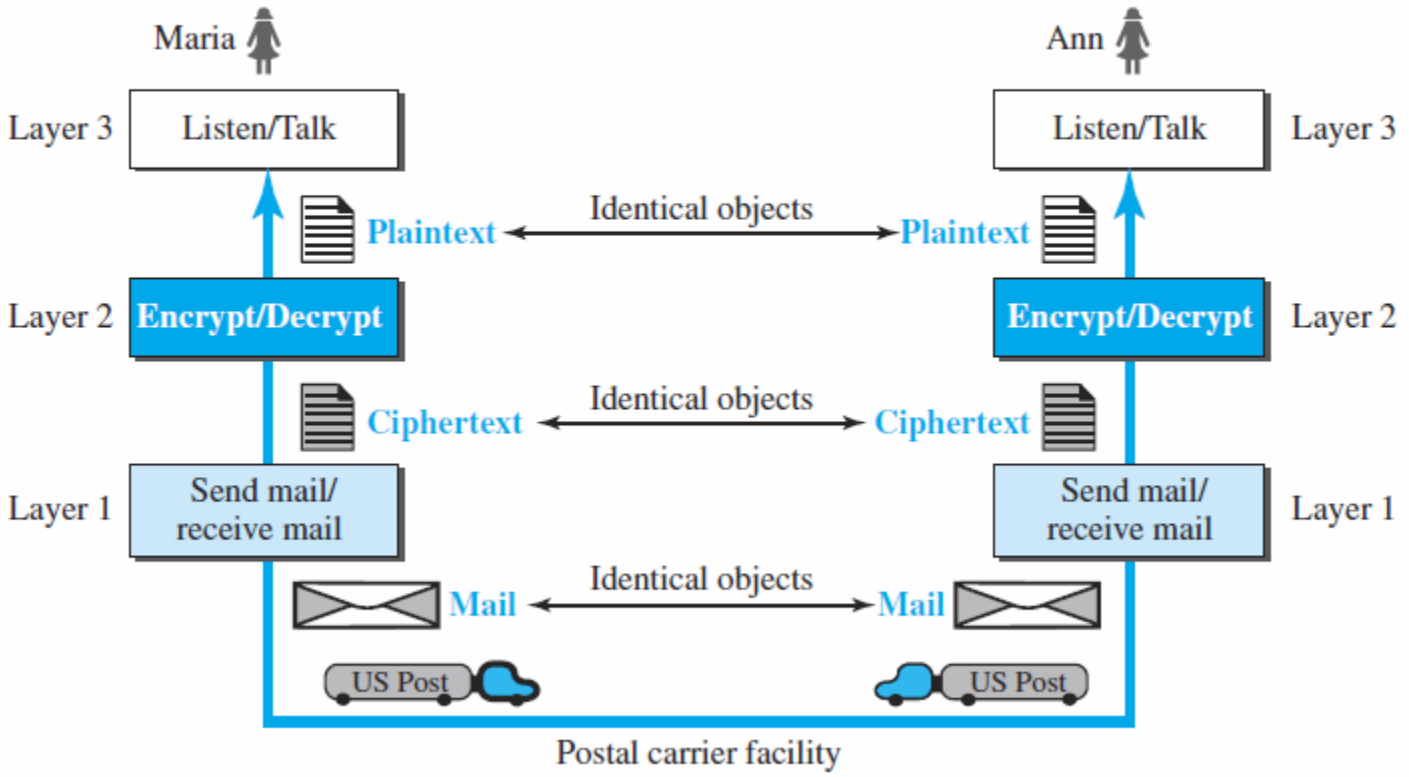
- However, they do not want their **ideas to be revealed by other people** if the letters are intercepted.

2.1.1 Scenarios Contd.

- They agree on an **encryption/decryption** technique.
- The **sender** of the letter **encrypts** it to make it unreadable by an intruder; the **receiver** of the letter **decrypts** it to get the original letter.
- Assume that Maria and Ann use one technique that makes it hard to decrypt the letter if one does not have the key for doing so.
- Now we can say that the communication between Maria and Ann takes place in **three layers**, as shown in Figure 2.2.
- We assume that Ann and Maria each have **three machines** (or robots) that can perform the task at each layer.

2.1.1 Scenarios Contd.

Figure 2.2 *A three-layer protocol*



2.1.1 Scenarios Contd.

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

2.1.1 Scenarios Contd.

- At Ann's side, the **first layer** machine picks up the letter from Ann's **mail box**, recognizing the letter from Maria by the sender address. The machine takes out the ciphertext from the envelope and delivers it to the second layer machine.

- The **second layer machine decrypts** the message, creates the plaintext, and passes the plaintext to the third-layer machine.

- The **third layer machine** takes the plaintext and **reads** it as though Maria is speaking.

2.1.1 Scenarios Contd.

- **Protocol layering** enables us to **divide a complex task** into several smaller and simpler tasks.

- For example, in Figure 2.2, we could have used **only one machine to do the job of all three machines.**

- However, if Maria and Ann decide that the **encryption/ decryption** done by the machine is **not enough to protect** their secrecy, they would have to **change the whole machine.**

- In the present situation, they need to **change only the second layer machine**; the other two can remain the same.

- This is referred to as ***modularity***.

2.1.1 Scenarios Contd.

- Modularity in this case means **independent layers**.
- A layer (**module**) can be defined as a **black box** with inputs and outputs, without concern about how inputs are changed to outputs.
- If two machines provide the same outputs when given the same inputs, they can **replace** each other.
- For example, Ann and Maria **can buy the second layer machine from two different manufacturers**.
- As long as the two machines create the same ciphertext from the same plaintext and vice versa, they do the job.

2.1.1 Scenarios Contd.

- One of the **advantages** of protocol layering is that it allows us to **separate the services from the implementation.**

- A layer needs to be able to receive a set of services from the lower layer and to give the services to the upper layer; we don't care about **how the layer is implemented.**

- For example, Maria may decide not to buy the machine (robot) for the first layer; she can do the job herself. As long as Maria can do the tasks provided by the first layer, in both directions, the communication system works.

- Communication does not always use only two end systems; there are **intermediate systems** that need only some layers, but not all layers.

- If we did not use protocol layering, we would have to make each intermediate system as complex as the end systems, which makes the whole system more expensive.

2.1.1 Scenarios Contd.

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

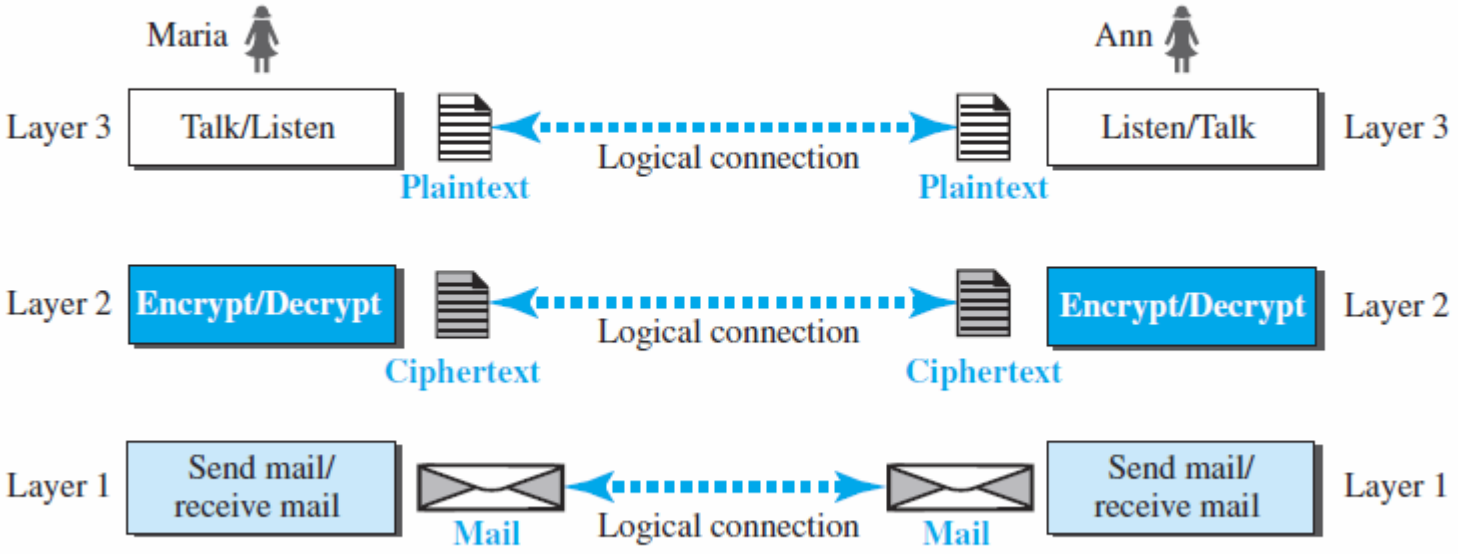
2.1.2 Principles of Protocol Layering

- ***First Principle***
- If we want bidirectional communication, we need to make each layer so that it is able to perform **two opposite tasks**, one in each direction.
  - For example, the third layer task is to **listen** (in one direction) and ***talk*** *(in the other* direction).
  - The second layer needs to be able to **encrypt and decrypt**.
  - The first layer needs to **send and receive mail.**
- ***Second Principle***
- The two objects under each layer at both sites should be **identical**.
  - For example, the object under layer 3 at both sites should be a **plaintext letter**.
  - The object under layer 2 at both sites should be a **ciphertext letter**.
  - The object under layer 1 at both sites should be a **piece of mail.**

2.1.3 Logical Connections

- After following the above two principles, we can think about logical connection between each layer as shown in Figure 2.3.
- This means that we have **layer-to-layer communication**.
- Maria and Ann can think that there is a logical (imaginary) connection at each layer through which they can send the object created from that layer.

Figure 2.3 *Logical connection between peer layers*



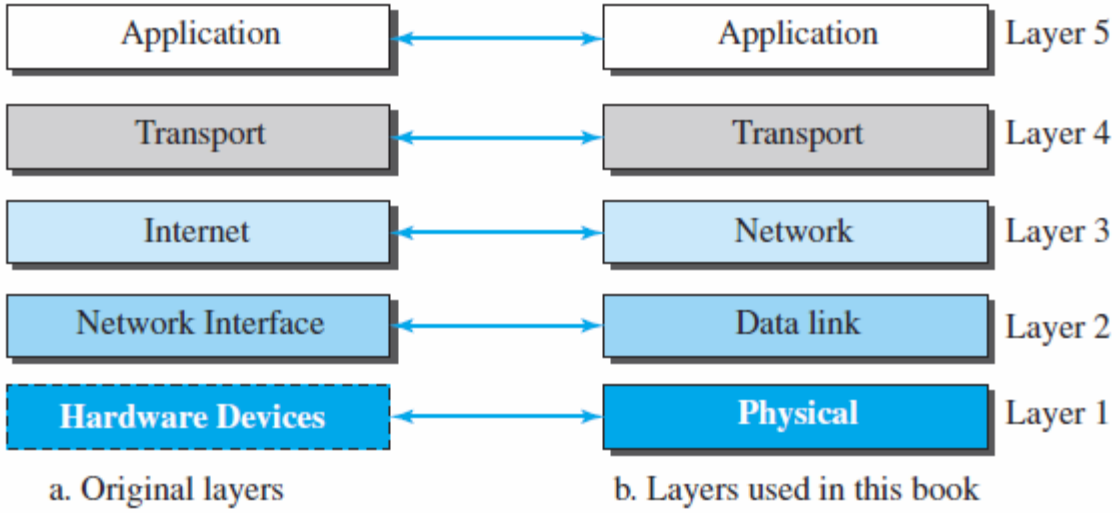
2.2 TCP/IP PROTOCOL SUITE

- **TCP/IP is a protocol suite** (a set of protocols organized in different layers) used in the Internet today.
- It is a hierarchical protocol made up of **interactive modules**, each of which provides a specific functionality.

- The term **hierarchical** means that each **upper level protocol is supported by the services provided by** one or more **lower level protocols**.

- TCP/IP is thought of as a **five-layer model.** Figure 2.4 shows both configurations.

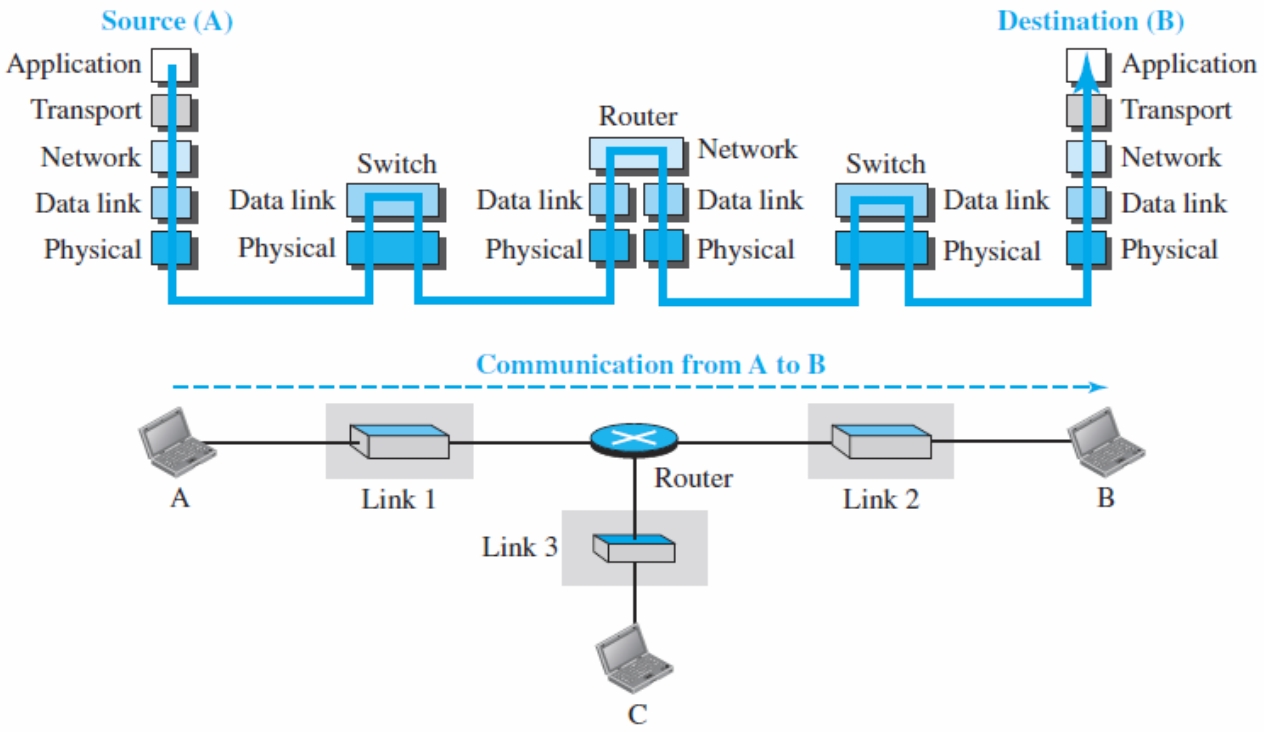
Figure 2.4 *Layers in the TCP/IP protocol suite*



2.2.1 Layered Architecture

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

Figure 2.5 *Communication through an internet*



2.2.1 Layered Architecture Contd.

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

2.2.1 Layered Architecture Contd.

- The **router is involved in only three layers**; there is no transport or application layer in a router as long as the router is used only for routing.

- Although a router is always involved in one network layer, it is **involved in n combinations of link** and physical layers in which n is the number of links the router is connected to.

- The reason is that each link may use its own data-link or physical protocol.

- For example, in the above figure, the router is involved in three links, but the message sent from source A to destination B is involved in two links.

- **Each link may be using different link-layer and physical-layer protocols**; the router needs to receive a packet from link 1 based on one pair of protocols and deliver it to link 2 based on another pair of protocols.

- **A link-layer switch** in a link, however, is involved only in **two layers**, data-link and physical.

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

2.2.3 Description of Each Layer

- ***Physical Layer***
- The physical layer is responsible for **carrying individual bits** in a frame across the link.

- Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a **logical communication because there is another, hidden layer, the transmission media**, under the physical layer.

- Two devices are connected by a **transmission medium (cable or air).**

- We need to know that the transmission medium does not carry bits; it carries **electrical or optical signals**.

- So the bits received in a frame from the data-link layer are transformed and sent through the transmission media, but we can think that the logical unit between two physical layers in two devices is a **bit**.

- There are several protocols that **transform a bit to a signal**.

2.2.3 Description of Each Layer Contd.

- ***Data-link Layer***
- The routers are responsible for choosing the *best links.* However, when the next link to travel is determined by the router, the **data-link layer is responsible for taking the datagram and moving it across the link.**

- The **link can** be a **wired LAN** with a link-layer switch, a **wireless LAN**, a **wired WAN**, or a **wireless WAN**.

- We can also have **different protocols** used with any link type.

- In each case, the data-link layer is responsible for **moving the packet through the link.**

- The data-link layer takes a datagram and encapsulates it in a packet called a ***frame***.

- Each **link-layer protocol** may provide a **different service.**

- Some link-layer protocols provide complete **error detection and correction**, some provide only error correction.

2.2.3 Description of Each Layer Contd.

- ***Network Layer***
- The network layer is responsible for creating a connection between the source computer and the destination computer.

- The communication at the network layer is **host-to-host**.

- However, since there can be several routers from the source to the destination, **the routers in the path are responsible for choosing the best route for each packet**.

- The network layer is responsible for host-to-host communication and routing the packet through possible routes.

- why we need a separate network layer?
  - **Separation of different tasks** between different layers.
  - The **routers do not need the application and transport layers**.
  - Separating the tasks allows us to **use fewer protocols** on the routers.

2.2.3 Description of Each Layer Contd.

- The network layer in the Internet includes the main protocol, **Internet Protocol** (IP), that defines the format of the packet, called a datagram at the network layer.
- IP also defines the **format** and the **structure of addresses** used in this layer.
- IP is also responsible for **routing** a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path.
- IP is a **connectionless protocol** that provides **no flow control, no error control, and no congestion control services.**
- This means that if any of theses services is required for an application, the application should rely only on the transport-layer protocol.
- The network layer also includes **unicast** (one-to-one) and **multicast** (one-to-many) routing protocols.
- A routing protocol creates **forwarding tables** for routers to help them in the routing process.

2.2.3 Description of Each Layer Contd.

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

2.2.3 Description of Each Layer Contd.

- ***Transport Layer***
- The **logical connection** at the transport layer is also **end-to-end**.

- The transport layer at the source host gets the message from the application layer, encapsulates it in a transport layer packet (called a ***segment*** *or a* ***user datagram*** *in different protocols) and sends it,* through the logical connection, to the transport layer at the destination host.

- In other words, the transport layer is responsible for **giving services to the application layer** to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host.

- why we need an end-to-end transport layer ?
  - The reason is the **separation of tasks and duties.**
  - The transport layer should be **independent** of the application layer.
  - Each application program can **use the protocol that best matches its requirement**.

2.2.3 Description of Each Layer Contd.

- The main protocol, **Transmission Control Protocol (TCP**), is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data.

- It creates a **logical pipe** between two TCPs for transferring a stream of bytes.

- TCP provides
  - **flow control** (matching the sending data rate of the source host with the receiving data rate of the destination host to prevent overwhelming the destination),
  - **error control** (to guarantee that the segments arrive at the destination without error and resending the corrupted ones), and
  - **congestion control** to reduce the loss of segments due to congestion in the network.

- The other common protocol, **User Datagram Protocol (UDP),** is a connectionless protocol that transmits user datagrams without first creating a logical connection.

2.2.3 Description of Each Layer Contd.

- ***Application Layer***
- As Figure 2.6 shows, the logical connection between the two application layers is **end-to- end**.

- Communication at the application layer is between two ***processes*** *(two programs* running at this layer).

- To communicate, a process sends a request to the other process and receives a response.

- **Process-to-process communication** is the duty of the application layer.

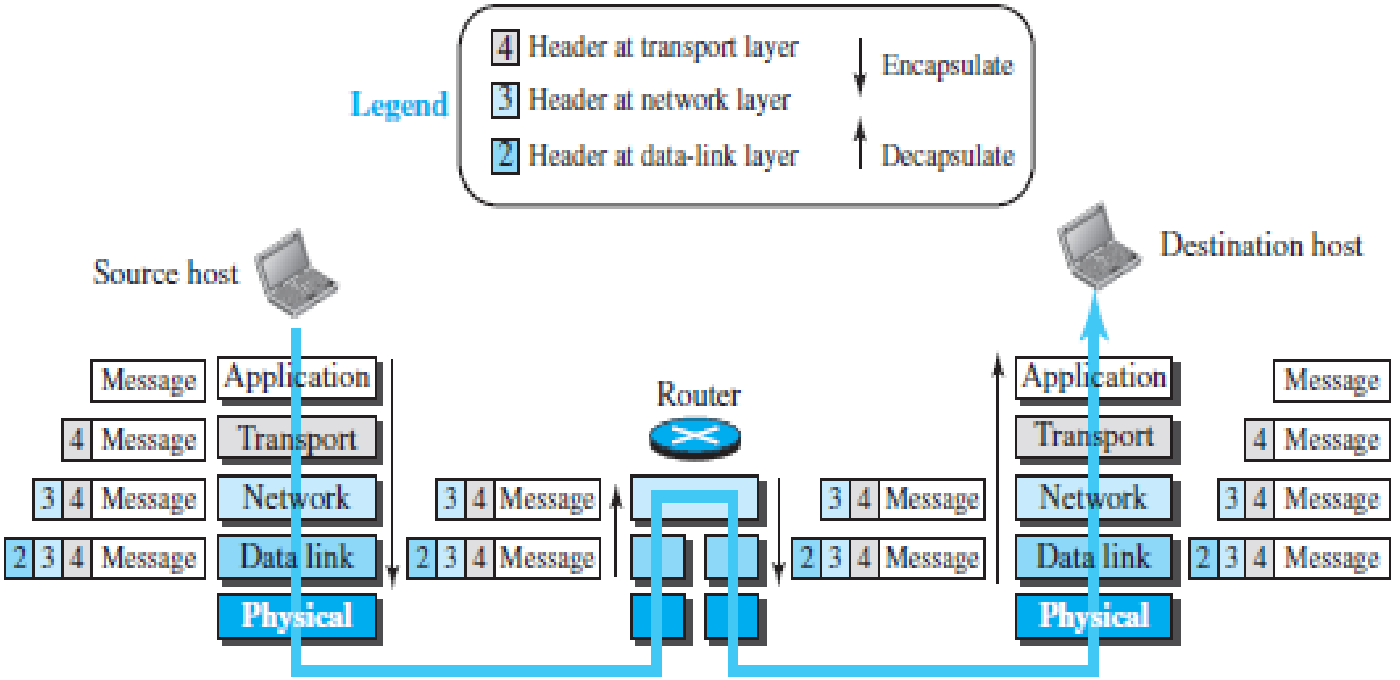
- The application layer in the Internet includes many **predefined protocols**, but a **user can also create a pair of processes** to be run at the two hosts.

2.2.3 Description of Each Layer Contd.

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

2.2.4 Encapsulation and Decapsulation

Figure 2.8 *Encapsulation/Decapsulation*



2.2.4 Encapsulation and Decapsulation Contd.

- Figure 2.8 shows the concept of encapsulation and decapsulation for small internet.
- In Figure 2.8, we show the encapsulation in the source host, decapsulation in the destination host, and encapsulation and decapsulation in the router.

- **Encapsulation at the Source Host**
- At the source, we have only encapsulation.
  - 1. At the application layer, the data to be exchanged is referred to as a message. A message normally does not contain any header or trailer, but if it does, we refer to the whole as the **message**. The message is passed to the transport layer.

  - 2. The transport layer takes the message as the payload, the load that the transport layer should take care of. It adds the transport layer header to the payload, which contains the identifiers of the source and destination application programs that want to communicate plus some more information that is needed for the end-to-end delivery of the message, such as information needed for flow, error control, or congestion control. The result is the transport-layer packet, which is called the **segment** (in TCP) and the user **datagram** (in UDP). The transport layer then passes the packet to the network layer.

2.2.4 Encapsulation and Decapsulation Contd.

- 3. The network layer takes the transport-layer packet as data or payload and adds its own header to the payload. The header contains the addresses of the source and destination hosts and some more information used for error checking of the header, fragmentation information, and so on. The result is the network-layer packet, called a **datagram**. The network layer then passes the packet to the data-link layer.

- 4. The data-link layer takes the network-layer packet as data or payload and adds its own header, which contains the link-layer addresses of the host or the next hop (the router). The result is the link-layer packet, which is called a frame. The **frame** is passed to the physical layer for transmission.

2.2.4 Encapsulation and Decapsulation Contd.

- **Decapsulation and Encapsulation at the Router**

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

  - 1. After the set of bits are delivered to the data-link layer, this layer decapsulates the datagram from the frame and passes it to the network layer.

  - 2. The network layer only inspects the source and destination addresses in the datagram header and consults its forwarding table to find the next hop to which the datagram is to be delivered. The contents of the datagram should not be changed by the network layer in the router unless there is a need to fragment the datagram if it is too big to be passed through the next link. The datagram is then passed to the data-link layer of the next link.

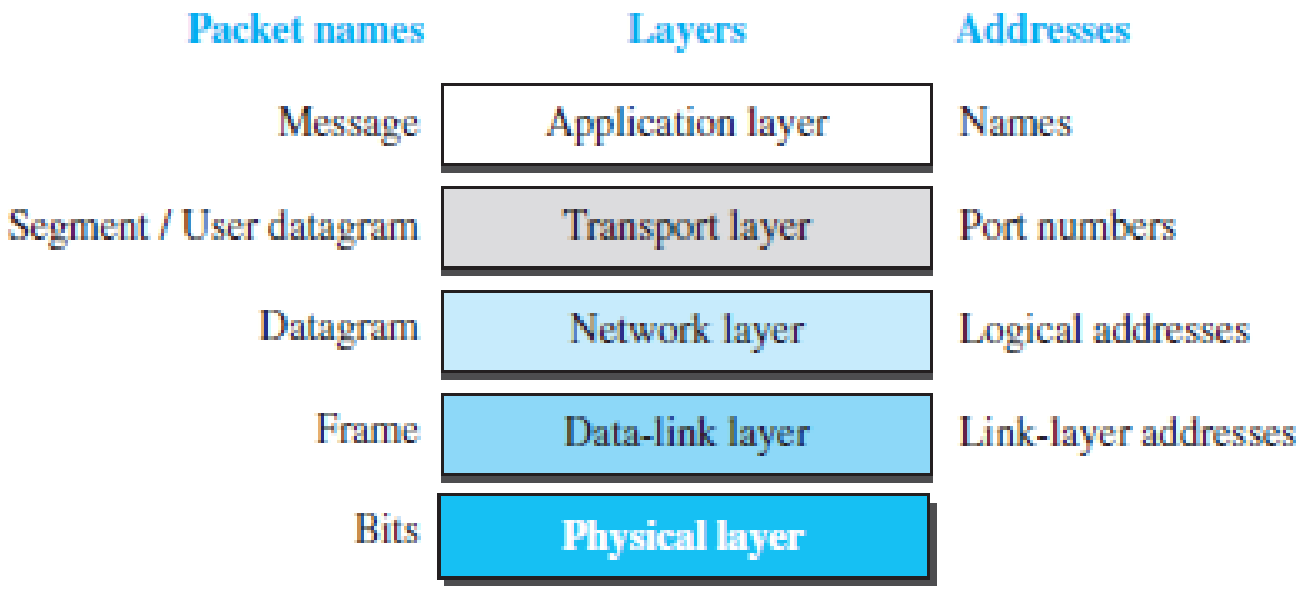
  - 3. The data-link layer of the next link encapsulates the datagram in a frame and passes it to the physical layer for transmission.

2.2.4 Encapsulation and Decapsulation Contd.

- **Decapsulation at the Destination Host**
  - At the destination host, each layer only decapsulates the packet received, removes the payload, and delivers the payload to the next-higher layer protocol until the message reaches the application layer. Decapsulation in the host involves error checking.

2.2.5 Addressing

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



2.2.5 Addressing Contd.

- Any communication that involves two parties needs two addresses: source address and destination address.
- The physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address.
- Figure 2.9 shows the addressing at each layer.
- As the figure shows, there is a relationship between the layer, the address used in that layer, and the packet name at that layer.
- At the application layer, we normally use names to define the site that provides services, such as someorg.com, or the e-mail address, such as somebody@coldmail.com.

2.2.5 Addressing Contd.

- At the transport layer, addresses are called port numbers, and these define the application-layer programs at the source and destination.

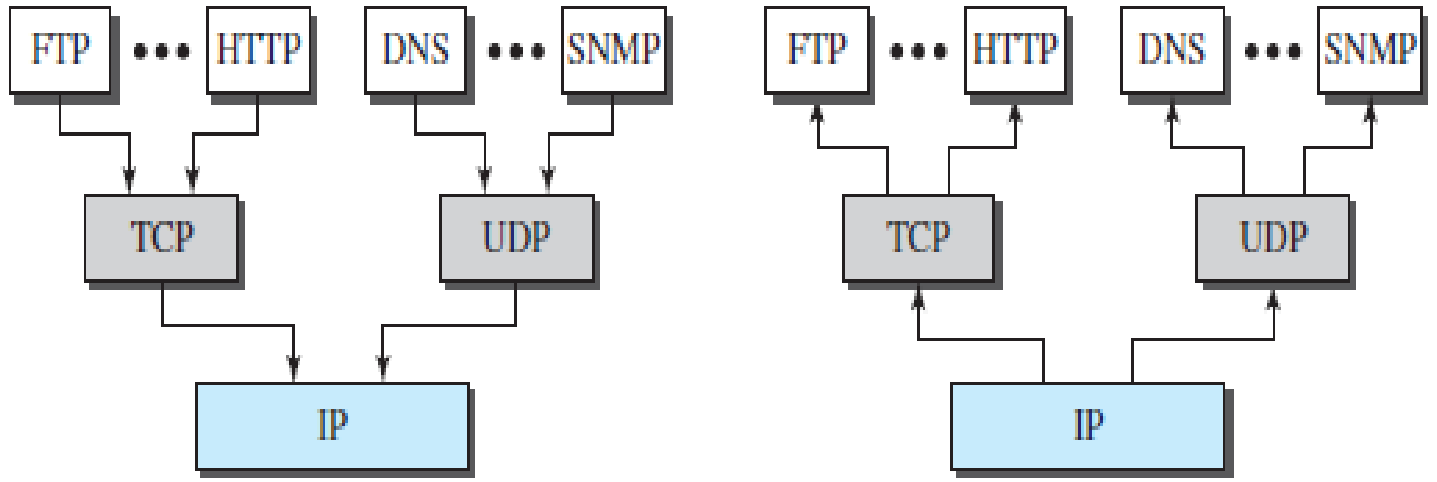
- Port numbers are local addresses that distinguish between several programs running at the same time.

- At the network-layer, the addresses are global, with the whole Internet as the scope.

- A network-layer address uniquely defines the connection of a device to the Internet. The link-layer addresses, sometimes called MAC addresses, are locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN).

2.2.6 Multiplexing and Demultiplexing

Figure 2.10 *Multiplexing and demultiplexing*



a. Multiplexing at source

b. Demultiplexing at destination

2.2.6 Multiplexing and Demultiplexing Contd.

- Since the TCP/IP protocol suite uses several protocols at some layers, we can say that we have multiplexing at the source and demultiplexing at the destination.

- Multiplexing in this case means that a protocol at a layer can encapsulate a packet from several next-higher layer protocols (one at a time)

- Demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time).

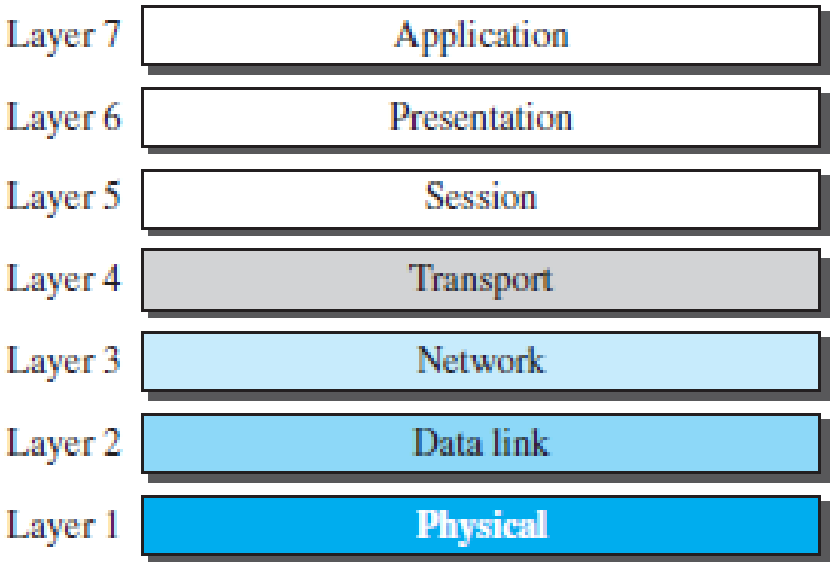
- Figure 2.10 shows the concept of multiplexing and demultiplexing at the three upper layers.

2.2.6 Multiplexing and Demultiplexing Contd.

- To be able to multiplex and demultiplex, a protocol needs to have a field in its header to identify to which protocol the encapsulated packets belong.
- At the transport layer, either UDP or TCP can accept a message from several application-layer protocols.
- At the network layer, IP can accept a segment from TCP or a user datagram from UDP.
- IP can also accept a packet from other protocols such as ICMP, IGMP, and so on.
- At the data-link layer, a frame may carry the payload coming from IP or other protocols such as ARP.

2.3 THE OSI MODEL

Figure 2.11 *The OSI model*



2.3 THE OSI MODEL Contd.

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

2.3 THE OSI MODEL Contd.

- The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software.

- The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.

- The OSI model was intended to be the basis for the creation of the protocols in the OSI stack.

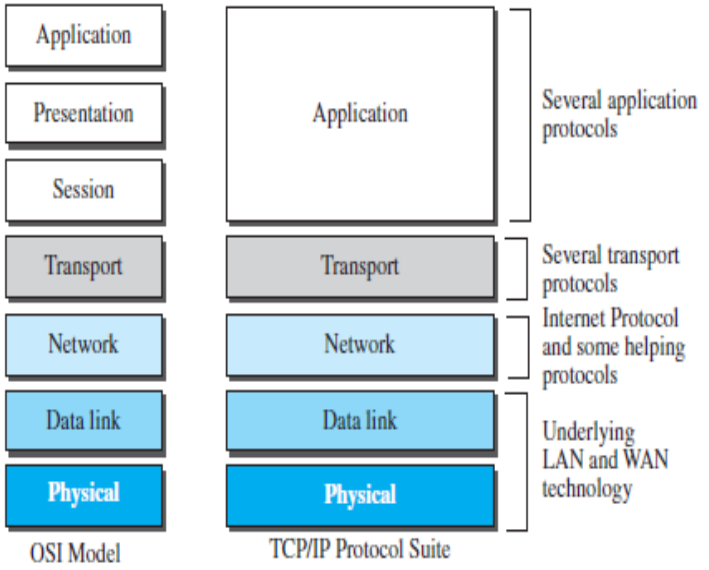
- The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems.

- It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network .

2.3.1 OSI versus TCP/IP

- The application layer in TCP/IP protocol suite is usually considered to be the combination of three layers in the OSI model, as shown in Figure 2.12
- Two reasons were mentioned for this decision. First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport-layer protocols.
- Second, the application layer is not only one piece of software. Many applications can be developed at this layer. If some of the functionalities mentioned in the session and presentation layers are needed for a particular application, they can be included in the development of that piece of software.

Figure 2.12 *TCP/IP and OSI model*



2.3.2 Lack of OSI Model's Success

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

END

Unit I
Chapter 3
Physical Layer

Reference: Data Communication and Networking, Behrouz A Forouzan**,**
McGraw Hill, 5th Edition,
2008.

3.5 DATA RATE LIMITS

- A very important consideration in data communications is how fast we can send data, in bits per second, over a channel.

- Data rate depends on three factors:
  - **1. The bandwidth available**
  - **2. The level of the signals we use**
  - **3. The quality of the channel (the level of noise)**

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

3.5.1 Noiseless Channel: Nyquist Bit Rate

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

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

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

- According to the formula, we might think that, given a specific bandwidth, we can have any bit rate we want by increasing the number of signal levels. Although the idea is theoretically correct, practically there is a limit.

- When we increase the number of signal levels, we impose a burden on the receiver. If the number of levels in a signal is just 2, the receiver can easily distinguish between a 0 and a 1. If the level of a signal is 64, the receiver must be very sophisticated to distinguish between 64 different levels.

- In other words, increasing the levels of a signal reduces the reliability of the system.

3.5.1 Noiseless Channel: Nyquist Bit Rate Contd.

- **Example 3.34**
- Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. The maximum bit rate can be calculated as

3.5.1 Noiseless Channel: Nyquist Bit Rate Contd.

- **Example 3.34**
- Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. The maximum bit rate can be calculated as

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

3.5.1 Noiseless Channel: Nyquist Bit Rate Contd.

- **Example 3.35**
- Consider the same noiseless channel transmitting a signal with four signal levels (for each level, we send 2 bits). The maximum bit rate can be calculated as

3.5.1 Noiseless Channel: Nyquist Bit Rate Contd.

- **Example 3.35**
- Consider the same noiseless channel transmitting a signal with four signal levels (for each level, we send 2 bits). The maximum bit rate can be calculated as

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

3.5.1 Noiseless Channel: Nyquist Bit Rate Contd.

- **Example 3.36**
- We need to send 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signal levels do we need?

3.5.1 Noiseless Channel: Nyquist Bit Rate Contd.

- **Example 3.36**
- We need to send 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signal levels do we need?

- Solution
- We can use the Nyquist formula as shown:

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

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

3.5.2 Noisy Channel: Shannon Capacity

- In reality, we cannot have a noiseless channel; the channel is always noisy.

- In 1944, Claude Shannon introduced a formula, called the **Shannon capacity, to determine the** theoretical highest data rate for a noisy channel:

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

- In this formula, bandwidth is the bandwidth of the channel, SNR is the signal-to-noise ratio, and capacity is the capacity of the channel in bits per second.

- Note that in the Shannon formula there is no indication of the signal level, which means that no matter how many levels we have, we cannot achieve a data rate higher than the capacity of the channel.

3.5.2 Noisy Channel: Shannon Capacity Contd.

- **Example 3.37**
- Consider an extremely noisy channel in which the value of the signal-to-noise ratio is almost zero. In other words, the noise is so strong that the signal is faint. For this channel the capacity *C* is calculated as

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

- This means that the capacity of this channel is zero regardless of the bandwidth.

- In other words, we cannot receive any data through this channel.

3.5.2 Noisy Channel: Shannon Capacity Contd.

- **Example 3.38**
- We can calculate the theoretical highest bit rate of a regular telephone line. A telephone line normally has a bandwidth of 3000 Hz (300 to 3300 Hz) assigned for data communications.

- The signal-to-noise ratio is usually 3162. For this channel the capacity is calculated as

$$C = B \log_2 (1 + \text{SNR}) = 3000 \log_2 (1 + 3162) = 3000 \log_2 3163$$
$$= 3000 \times 11.62 = 34{,}860 \text{ bps}$$

- This means that the highest bit rate for a telephone line is 34.860 kbps.

- If we want to send data faster than this, we can either increase the bandwidth of the line or improve the signal-to-noise ratio.

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

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

Ans: Given, bandwidth, B = 4000 KHz, SNR = 1000

So, We know the Shannon capacity $C = B \log_2 (1 + SNR)$

$\therefore C = 4000 \text{ x } 10\text{^}3 \log_2 (1 + 1000) \approx 40$ Mbps

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

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

SNR = (10)^2 / (5x10^-3) ^2 =4 x 10^6

C = 4 x 10^3 x log2 (1+4 x 10^6)
= 87.72 kbps

3.5.2 Noisy Channel: Shannon Capacity Contd.

- **Example 3.39**
- The signal-to-noise ratio is often given in decibels. Assume that SNRdB = 36 and the channel bandwidth is 2 MHz.

- The theoretical channel capacity can be calculated as

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

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

3.5.2 Noisy Channel: Shannon Capacity Contd.

- **Example 3.40**
- When the SNR is very high, we can assume that SNR + 1 is almost the same as SNR. In these cases, the theoretical channel capacity can be simplified to

$$C = B \times \frac{\text{SNR}_{\text{dB}}}{3}$$

- ***For example, we*** can calculate the theoretical capacity of the previous example as

$$C = 2 \text{ MHz} \times \frac{36}{3} = 24 \text{ Mbps}$$

- **What is the theoretical capacity of a channel in each of the following cases?**
  - **a. Bandwidth: 20 KHz SNRdB = 40**
  - **b. Bandwidth: 200 KHz SNRdB = 4**
  - **c. Bandwidth: 1 MHz SNRdB = 20**

- **What is the theoretical capacity of a channel in each of the following cases?**
  - **a. Bandwidth: 20 KHz SNRdB = 40**
  - **b. Bandwidth: 200 KHz SNRdB = 4**
  - **c. Bandwidth: 1 MHz SNRdB = 20**

Ans: We can approximately calculate the capacity as

a. $C = B * (SNR_{dB} /3) = 20 * (40 /3) = 267$ Kbps

b. $C = B * (SNR_{dB} /3) = 200 * (4 /3) = 267$ Kbps

c. $C = B * (SNR_{dB} /3) = 1 * (20 /3) = 6.67$ Mbps

- **We have a channel with 4 KHz bandwidth. If we want to send data at 100Kbps, what is the minimum SNRdB? What is the SNR?**

- **We have a channel with 4 KHz bandwidth. If we want to send data at 100Kbps, what is the minimum SNRdB? What is the SNR?**

Ans: Given, bandwidth = 4 KHz, Data rate = 100 Kbps

We know, $C = B * (SNR_{dB} / 3)$

$\Rightarrow SNR_{dB} = (3 * C) / B$

$\therefore$ minimum of $SNR_{dB} = (3 * 100) / 4 = 75$

So, the minimum $SNR = 10^{SNR_{dB}/10} = 10^{7.5}$

≈ 31622776

3.5.3 Using Both Limits

- We have a channel with a 1-MHz bandwidth. The SNR for this channel is 63. What are the appropriate bit rate and signal level?
- First, we use the Shannon formula to find the upper limit.

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

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

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

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

3.6 PERFORMANCE

- One important issue in networking is the performance of the network—how good is it?
  - Bandwidth
  - Throughput
  - Latency (Delay)
  - Bandwidth-Delay Product
  - Jitter

3.6.1 Bandwidth

- One characteristic that measures network performance is bandwidth.

- However, the term can be used in two different contexts with two different measuring values: bandwidth in hertz and bandwidth in bits per second.

Bandwidth in Hertz

- Bandwidth in hertz is the range of frequencies contained in a composite signal or the range of frequencies a channel can pass.

- For example, we can say the bandwidth of a subscriber telephone line is 4 kHz.

3.6.1 Bandwidth Contd.

Bandwidth in Bits per Seconds

- The term *bandwidth can also refer to the number of bits per second that a channel, a* link, or even a network can transmit.

- For example, one can say the bandwidth of a Fast Ethernet network (or the links in this network) is a maximum of 100 Mbps.

- This means that this network can send 100 Mbps.

Relationship

- There is an explicit relationship between the bandwidth in hertz and bandwidth in bits per second.

- Basically, an increase in bandwidth in hertz means an increase in bandwidth in bits per second.

3.6.1 Bandwidth Contd.

Example 3.42

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

Example 3.43

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

3.6.2 Throughput

- The throughput is a measure of how fast we can actually send data through a network.

- A link may have a bandwidth of *B bps, but we can only send T bps* through this link with *T always less than B.*

- *In other words, the bandwidth is a potential* measurement of a link; the throughput is an actual measurement of how fast we can send data.

- For example, we may have a link with a bandwidth of 1 Mbps, but the devices connected to the end of the link may handle only 200 kbps.

- This means that we cannot send more than 200 kbps through this link.

3.6.2 Throughput Contd.

- **Example 3.44**
- A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?

- **Solution**
- We can calculate the throughput as

$$\text{Throughput} = \frac{12{,}000 \times 10{,}000}{60} = 2 \text{ Mbps}$$

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

3.6.3 Latency (Delay)

- The latency or delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source.

- We can say that latency is made of four components: propagation time, transmission time, queuing time and processing delay.

Latency = propagation time + transmission time + queuing time + processing delay

3.6.3 Latency (Delay) Contd.

- ***Propagation Time***

- Propagation time measures the time required for a bit to travel from the source to the destination.

- The propagation time is calculated by dividing the distance by the propagation speed.

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

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

3.6.3 Latency (Delay) Contd.

- **Example 3.45**
- What is the propagation time if the distance between the two points is 12,000 km? Assume the propagation speed to be 2.4 × 10^8 m/s in cable.

- **Solution**
- We can calculate the propagation time as

$$\text{Propagation time} = \frac{12{,}000 \times 1000}{2.4 \times 10^8} = 50 \text{ ms}$$

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

3.6.3 Latency (Delay) Contd.

- ***Transmission Time***
- In data communications we don't send just 1 bit, we send a message.

- The first bit may take a time equal to the propagation time to reach its destination; the last bit also may take the same amount of time.

- However, there is a time between the first bit leaving the sender and the last bit arriving at the receiver. The first bit leaves earlier and arrives earlier; the last bit leaves later and arrives later.

- The **transmission time of a message** depends on the size of the message and the bandwidth of the channel.

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

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

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

Ans: Here, Bandwidth = 5 Kbps, Frame = 1,00,000 bits

$\therefore$ 1,00,000 bits / 5 Kbps = 20 s

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

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

Ans: Given, packet length = 1 million bytes = 1000000 * 8 bits = 8000000 bits

Bandwidth = 200 Kbps = 200000 bps

∴ We know, transmission time = (packet length)/(bandwidth)

= 8000000 / 200000 = 40 s

- **Example 3.46**
- What are the propagation time and the transmission time for a 2.5-KB (kilobyte) message (an email) if the bandwidth of the network is 1 Gbps? Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4 × 10^8 m/s.

- **Example 3.46**
- What are the propagation time and the transmission time for a 2.5-KB (kilobyte) message (an email) if the bandwidth of the network is 1 Gbps? Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4 × 10^8 m/s.

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

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

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

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

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

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

3.6.3 Latency (Delay) Contd.

Queuing Time

- The third component in latency is the **queuing time**, the time needed for each intermediate or end device to hold the message before it can be processed.

- The queuing time is not a fixed factor; it changes with the load imposed on the network.

- When there is heavy traffic on the network, the queuing time increases.

- An intermediate device, such as a router, queues the arrived messages and processes them one by one. If there are many messages, each message will have to wait.

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

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

Ans: Given,

Processing time = $10 \times 1\ \mu s = 10\ \mu s = 10^{-6}$ s

Queuing time = $10 \times 2\ \mu s = 20\ \mu s = 20*10^{-6}$ s

Transmission time = frame of size / bandwidth = $5000000 / (5*10^6) = 1$ s

Propagation time = distance / speed = $(2000*10^3) / (2 \times 10^8) = 0.01$ s

We know that,

Latency = processing time + queuing time + transmission time + propagation time

$= 10^{-6} + 20*10^{-6} + 1 + 0.01$

$= 1.01000030$ s

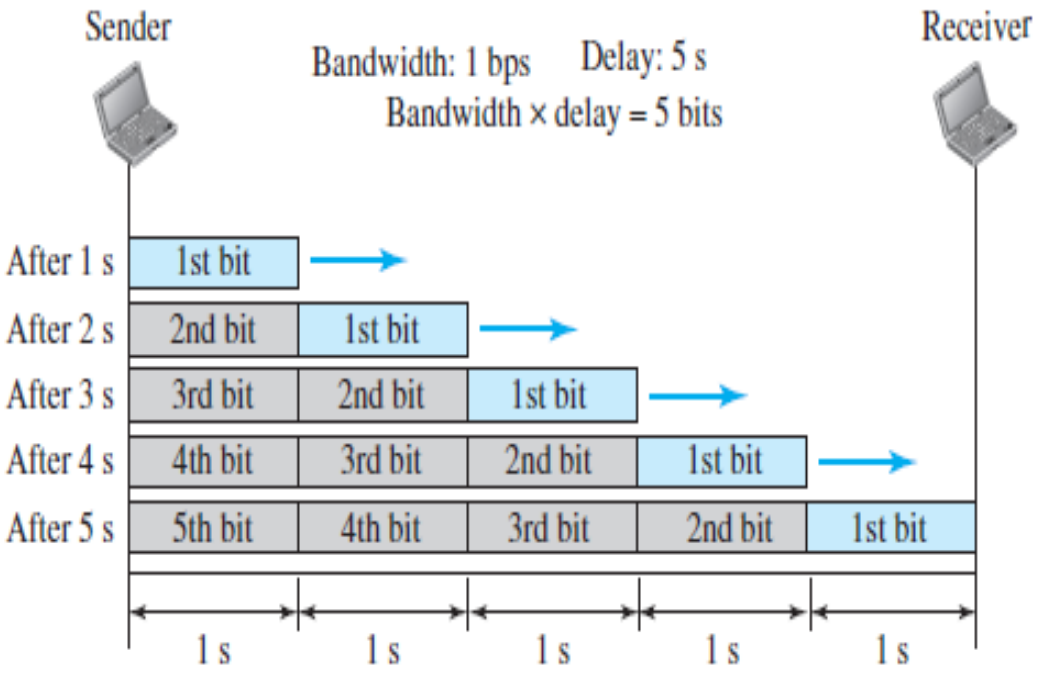
The transmission time is dominant here because the packet size is huge.

3.6.4 Bandwidth-Delay Product

- Bandwidth and delay are two performance metrics of a link.
- In data communications the product of the two, the bandwidth-delay product is important.
- Let us elaborate on this issue, using two hypothetical cases as examples.
- ❑ **Case 1.** Figure 3.32 shows case 1.
  - Let us assume that we have a link with a bandwidth of 1 bps.
  - We also assume that the delay of the link is 5
  - We want to see what the bandwidth-delay product means in this case.
  - Looking at the figure, we can say that this product 1 × 5 is the maximum number of bits that can fill the link. There can be no more than 5 bits at any time on the link.

3.6.4 Bandwidth-Delay Product Contd.

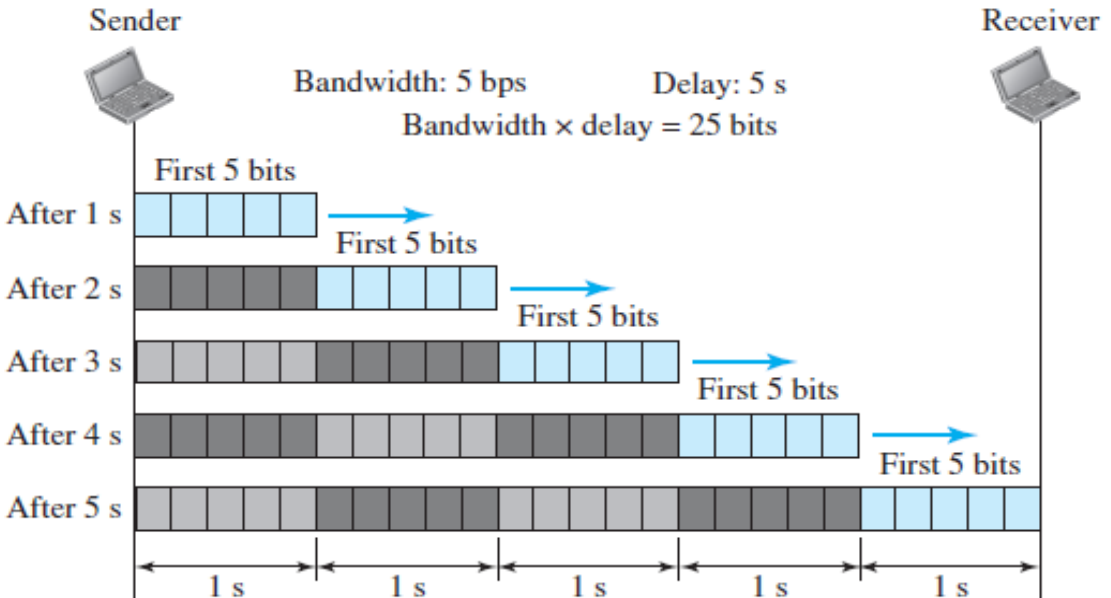
Figure 3.32 *Filling the link with bits for case 1*



3.6.4 Bandwidth-Delay Product Contd.

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

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



3.6.4 Bandwidth-Delay Product Contd.

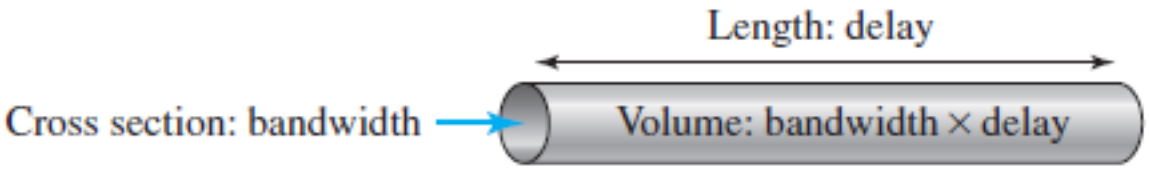
- The above two cases show that the product of bandwidth and delay is the number of bits that can fill the link.
- This measurement is important if we need to send data in bursts and wait for the acknowledgment of each burst before sending the next one.
- To use the maximum capability of the link, we need to make the size of our burst 2 times the product of bandwidth and delay; we need to fill up the full-duplex channel (two directions).
- The sender should send a burst of data of (2 × bandwidth × delay) bits.
- The sender then waits for receiver acknowledgment for part of the burst before sending another burst.
- The amount 2 × bandwidth × delay is the number of bits that can be in transition at any time.

3.6.4 Bandwidth-Delay Product Contd.

- **Example 3.48**
- We can think about the link between two points as a pipe. The cross section of the pipe represents the bandwidth, and the length of the pipe represents the delay.

- We can say the volume of the pipe defines the bandwidth-delay product, as shown in Figure 3.34.

Figure 3.34 *Concept of bandwidth-delay product*



3.6.5 Jitter

- Another performance issue that is related to delay is **jitter.**

- We can roughly say that jitter is a problem if different packets of data encounter different delays and the application using the data at the receiver site is time-sensitive (audio and video data, for example).

- If the delay for the first packet is 20 ms, for the second is 45 ms, and for the third is 40 ms, then the real-time application that uses the packets endures jitter.

END of Chapter 3

Chapter 4
Digital Transmission

Reference:

Data Communication and Networking, Behrouz A.Forouzan, McGraw Hill, 5th Edition, 2008

4.1 DIGITAL-TO-DIGITAL CONVERSION

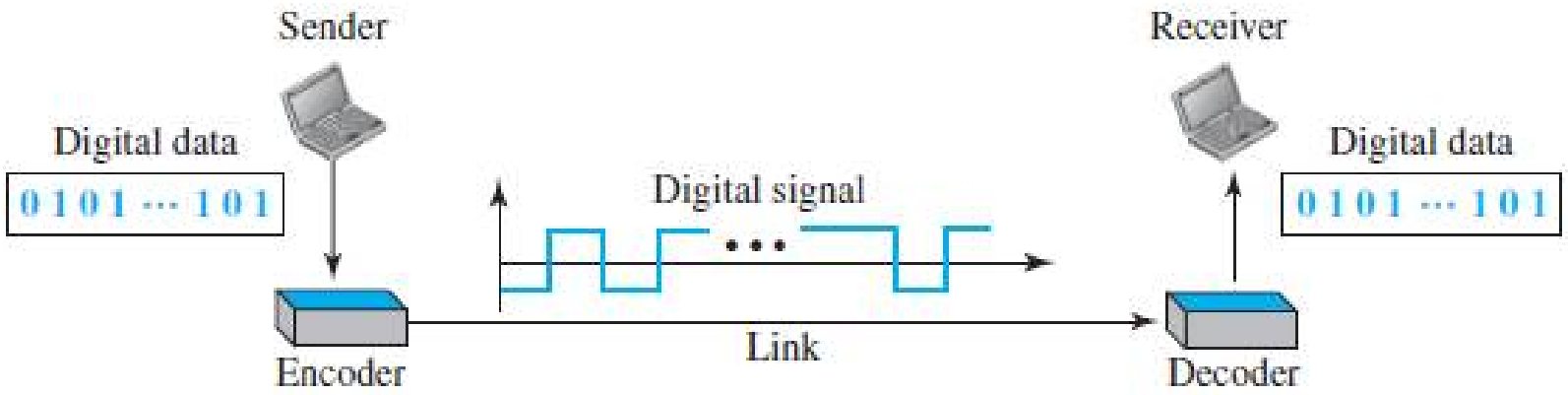
- Represent digital data by using digital signals.
- The conversion involves three techniques:
  - line coding,
  - block coding, and
  - scrambling.
- Line coding is always needed; block coding and scrambling may or may not be needed.

4.1.1 Line Coding

- Line coding is the process of converting digital data to digital signals.
- We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits
- 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.

4.1.1 Line Coding Contd.

Figure 4.1 *Line coding and decoding*



4.1.1 Line Coding Contd.

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

- ***Signal Element Versus Data Element***
  - In data communications, our goal is to send data elements.
  - A data element is the smallest entity that can represent a piece of information: this is the bit.

  - In digital data communications, a signal element carries data elements.
  - A signal element is the shortest unit (timewise) of a digital signal.

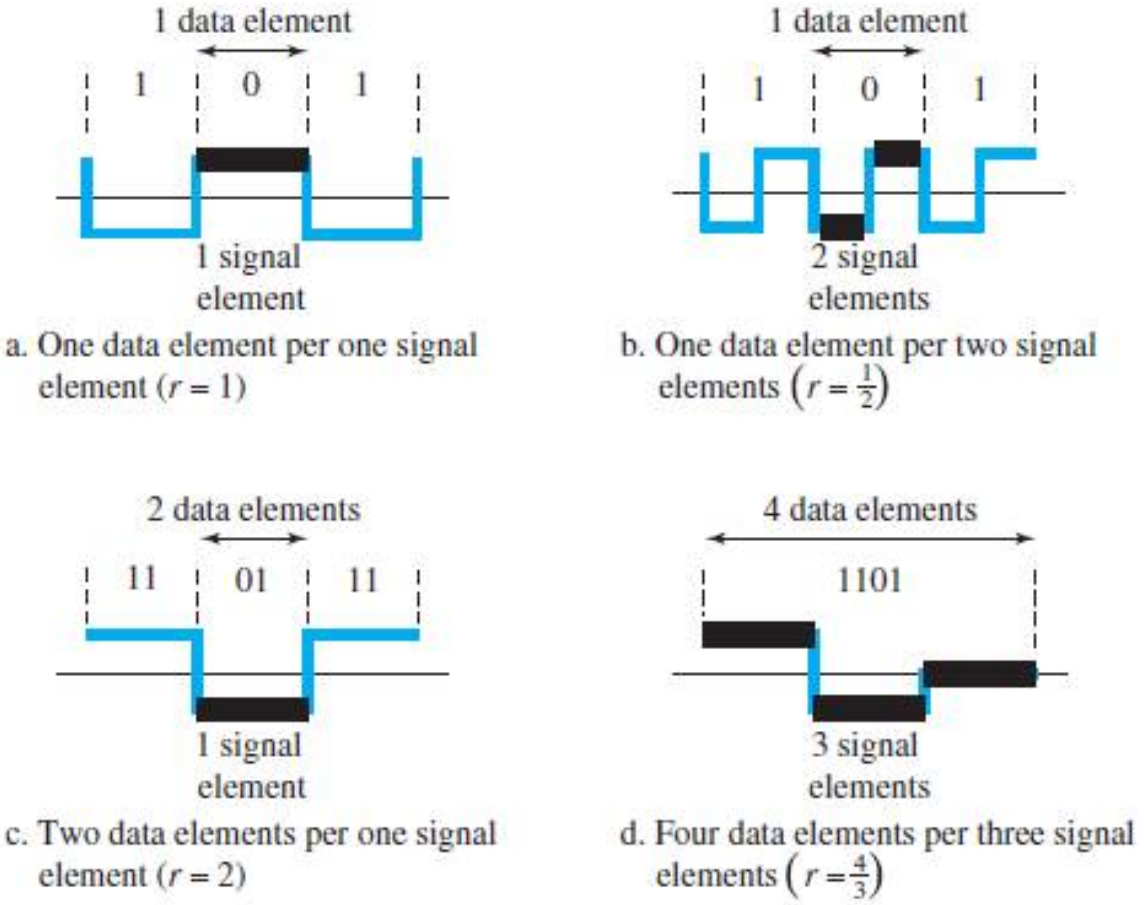
  - In other words, data elements are what we need to send; signal elements are what we can send.
  - Data elements are being carried; signal elements are the carriers.

4.1.1 Line Coding Contd.

- We define a ratio *r which is the number of data elements carried by each signal element.*
- Figure 4.2 shows several situations with different values of *r.*
- In part a of the figure, one data element is carried by one signal element (*r = 1).*
- *In* part b of the figure, we need two signal elements (two transitions) to carry each data element (*r = 1/2).*
- 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 = 4/3).*

4.1.1 Line Coding Contd.

Figure 4.2 *Signal element versus data element*



4.1.1 Line Coding Contd.

- ***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.

  - 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.

4.1.1 Line Coding Contd.

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

$$S = N/r$$

  in which r has been previously defined.

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

4.1.1 Line Coding Contd.

- To derive a formula for the relationship, we need to define three cases: the worst, best, and average.

- The worst case is when we need the maximum signal rate; the best case is when we need the minimum. In data communications, we are usually interested in the average case.

- We can formulate the relationship between data rate and signal rate as

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

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

4.1.1 Line Coding Contd.

- **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?*

4.1.1 Line Coding Contd.

- **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?*

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

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

4.1.1 Line Coding Contd.

- ***Bandwidth***
  - Digital signal that carries information is nonperiodic.
  - 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.

4.1.1 Line Coding Contd.

- We can say that the baud rate, not the bit rate, determines the required bandwidth for a digital signal.
- More changes in the signal mean injecting more frequencies into the signal.
- There is a relationship between the baud rate (signal rate) and the bandwidth. Bandwidth (range of frequencies) is proportional to the signal rate (baud rate).
- The minimum bandwidth can be given as

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

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

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

4.1.1 Line Coding Contd.

- **Example 4.2**
- The maximum data rate of a channel is Nmax = 2 × B × log2 L (defined by the Nyquist formula). Does this agree with the previous formula for Nmax?

- **Solution**
- A signal with L levels actually can carry log2 L bits per level. If each level corresponds to one signal element and we assume the average case (c = 1/2), then we have

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

4.1.1 Line Coding Contd.

- ***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.

4.1.1 Line Coding Contd.

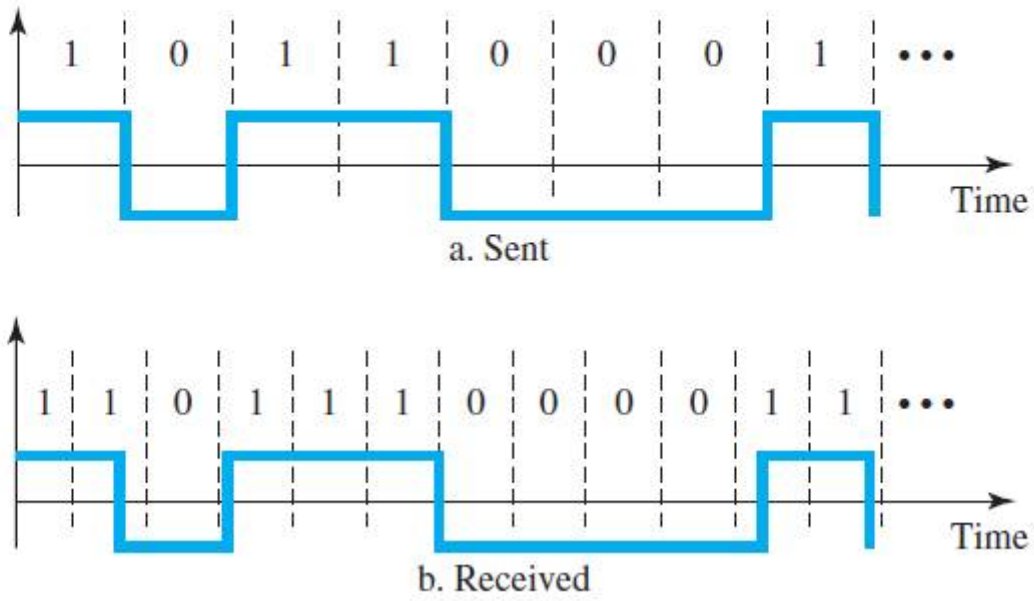
- ***DC Components***
  - When the voltage level in a digital signal is constant for a while, the spectrum creates very low frequencies (results of Fourier analysis).
  - These frequencies around zero, called DC (direct-current) *components, present problems for a system that cannot* pass low frequencies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

4.1.1 Line Coding Contd.

- ***Self-synchronization***
  - To correctly interpret the signals received from the sender, the receiver's bit intervals must correspond exactly to the sender's bit intervals.
  - If the receiver clock is faster or slower, the bit intervals are not matched and the receiver might misinterpret the signals.
  - Figure 4.3 shows a situation in which the receiver has a shorter bit duration.
  - The sender sends 10110001, while the receiver receives 110111000011.
  - A self-synchronizing digital signal includes timing information in the data being transmitted.
  - This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle, or end of the pulse.
  - If the receiver's clock is out of synchronization, these points can reset the clock.

4.1.1 Line Coding Contd.

Figure 4.3 *Effect of lack of synchronization*



4.1.1 Line Coding Contd.

- **Example 4.3**
- In a digital transmission, the receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 kbps? How many if the data rate is 1 Mbps?

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

1000 bits sent	→	1001 bits received	→	1 extra bps

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

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

- In a digital transmission, the sender clock is 0.2 percent faster than the receiver clock. How many extra bits per second does the sender send if the data rate is 1 Mbps?

- In a digital transmission, the sender clock is 0.2 percent faster than the receiver clock. How many extra bits per second does the sender send if the data rate is 1 Mbps?

Ans: Given, sender clock faster = 0.2 % = 0.2/100 = 0.002

The data rate = 1 Mbps = 10^6 bps

$\therefore$ extra bits = $0.002 \times 10^6 = 2000$ bits

4.1.1 Line Coding Contd.

Built-in Error Detection

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

Immunity to Noise and Interference

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

Complexity

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

4.1.2 Line Coding Schemes

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

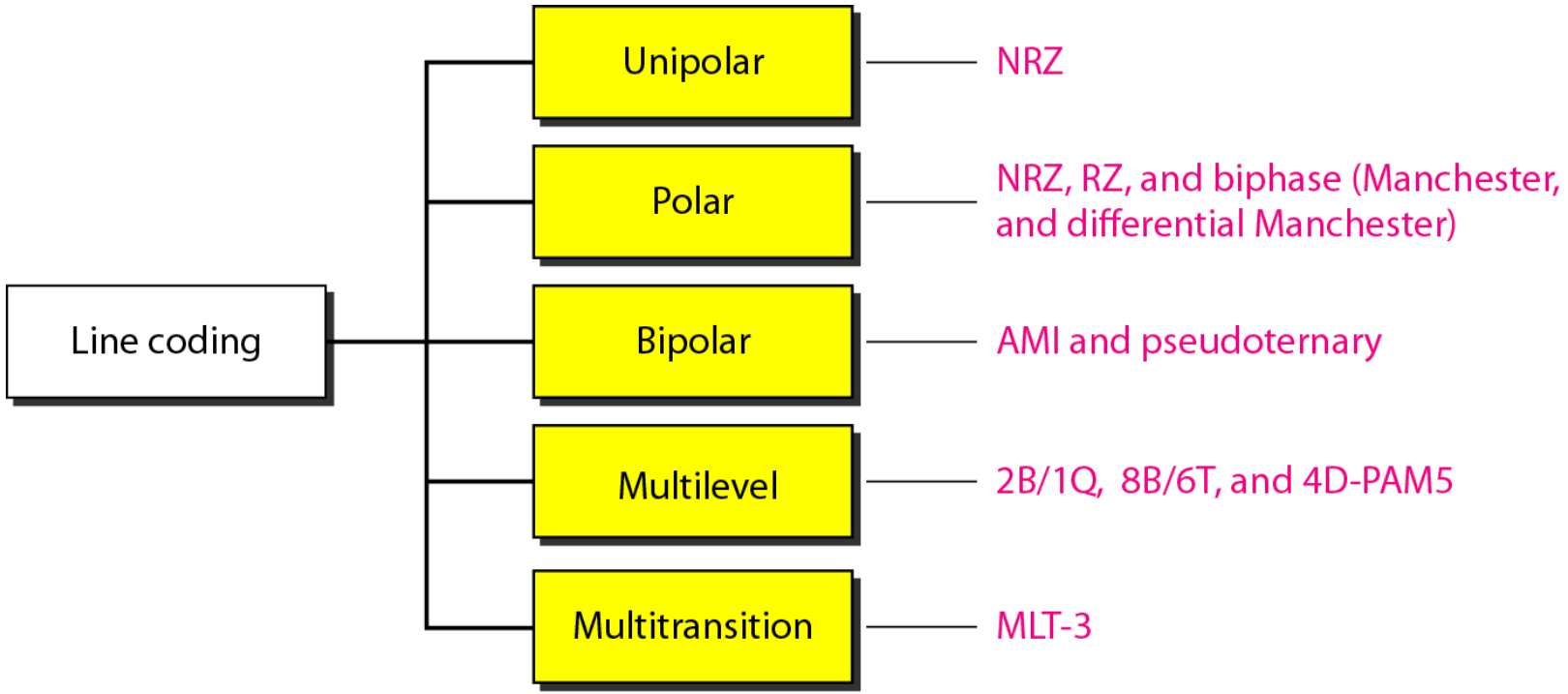


Figure 4.4 Line coding schemes

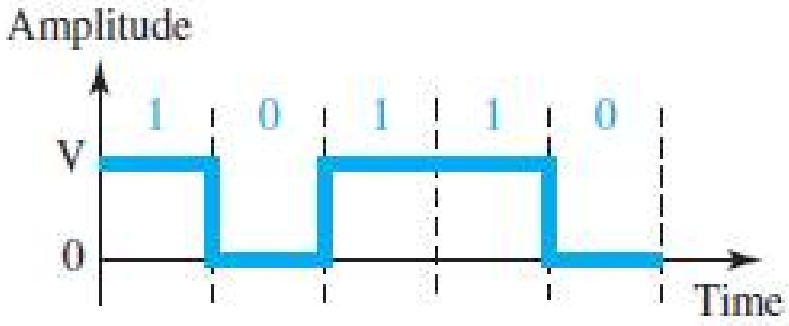
4.1.2 Line Coding Schemes Contd.

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

- **NRZ (Non-Return-to-Zero)**
  - Traditionally, a unipolar scheme was designed as a non-return-to-zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0.
  - It is called NRZ because the signal does not return to zero at the middle of the bit. Figure 4.5 shows a unipolar NRZ scheme.
  - Compared with its polar counterpart, this scheme is very costly.
  - The normalized power (the power needed to send 1 bit per unit line resistance) is double than that for polar NRZ. For this reason, this scheme is normally not used in data communications today.

4.1.2 Line Coding Schemes Contd.

Figure 4.5 *Unipolar NRZ scheme*



4.1.2 Line Coding Schemes Contd.

Polar Schemes

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

Non-Return-to-Zero (NRZ)

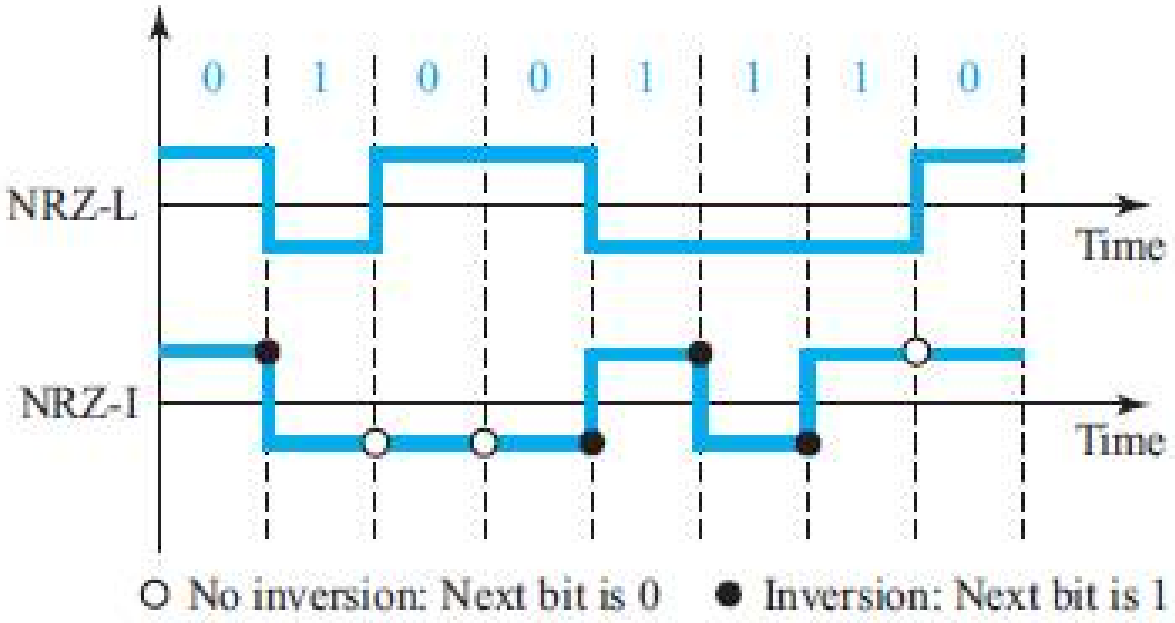
- In polar NRZ encoding, we use two levels of voltage amplitude.
- We can have two versions of polar NRZ: NRZ-L and NRZ-I, as shown in Figure 4.6.

- In the first variation, NRZ-L (NRZ-Level), the level of the voltage determines the value of the bit.

- In the second variation, NRZ-I (NRZ-Invert), the change or lack of change in the level of the voltage determines the value of the bit.
- If there is no change, the bit is 0; if there is a change, the bit is 1.

4.1.2 Line Coding Schemes Contd.

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

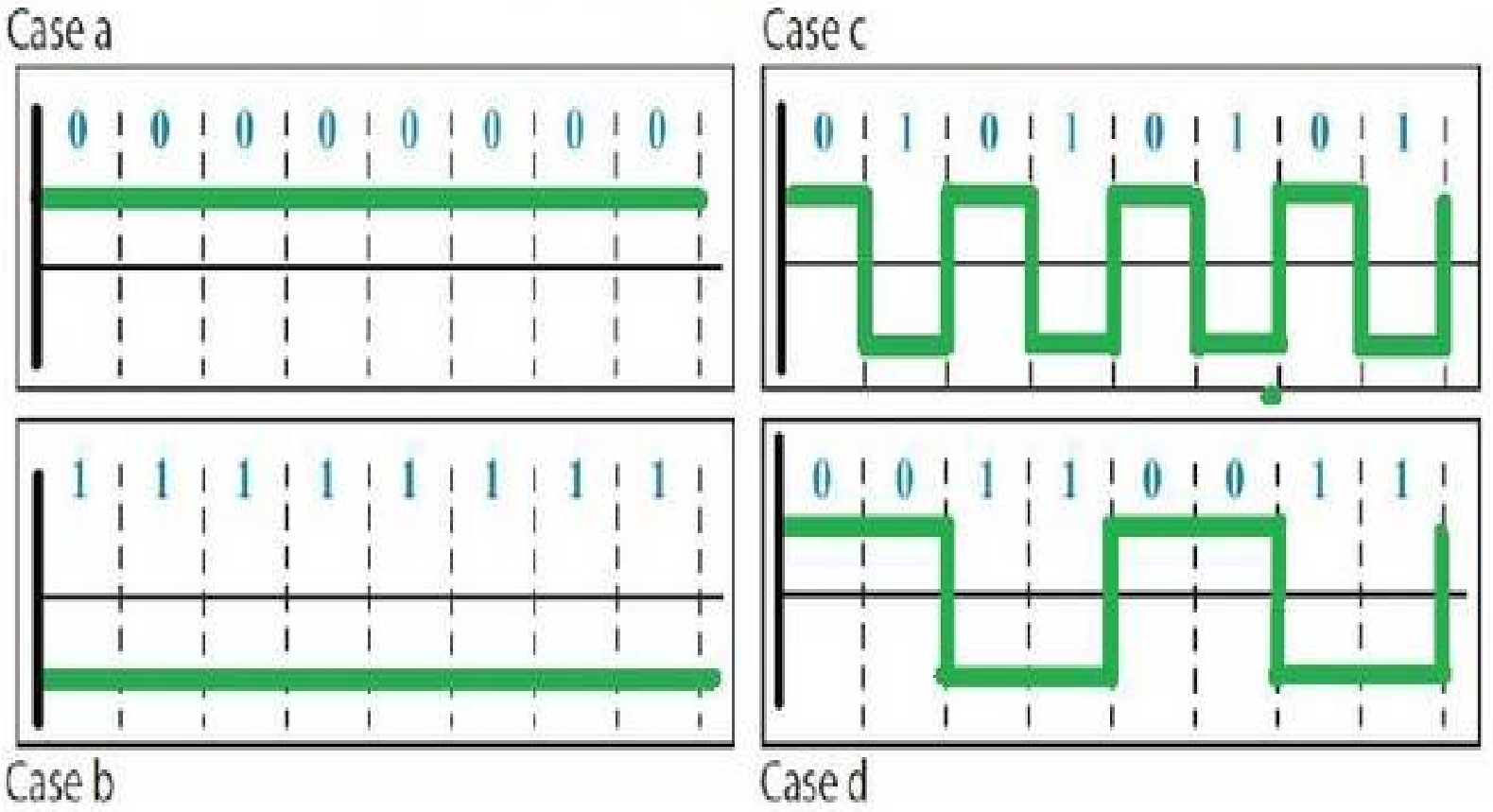


- **Draw the graph of the NRZ-L, NRZ-I, using each of the following data streams, assuming that the last signal level has been positive**

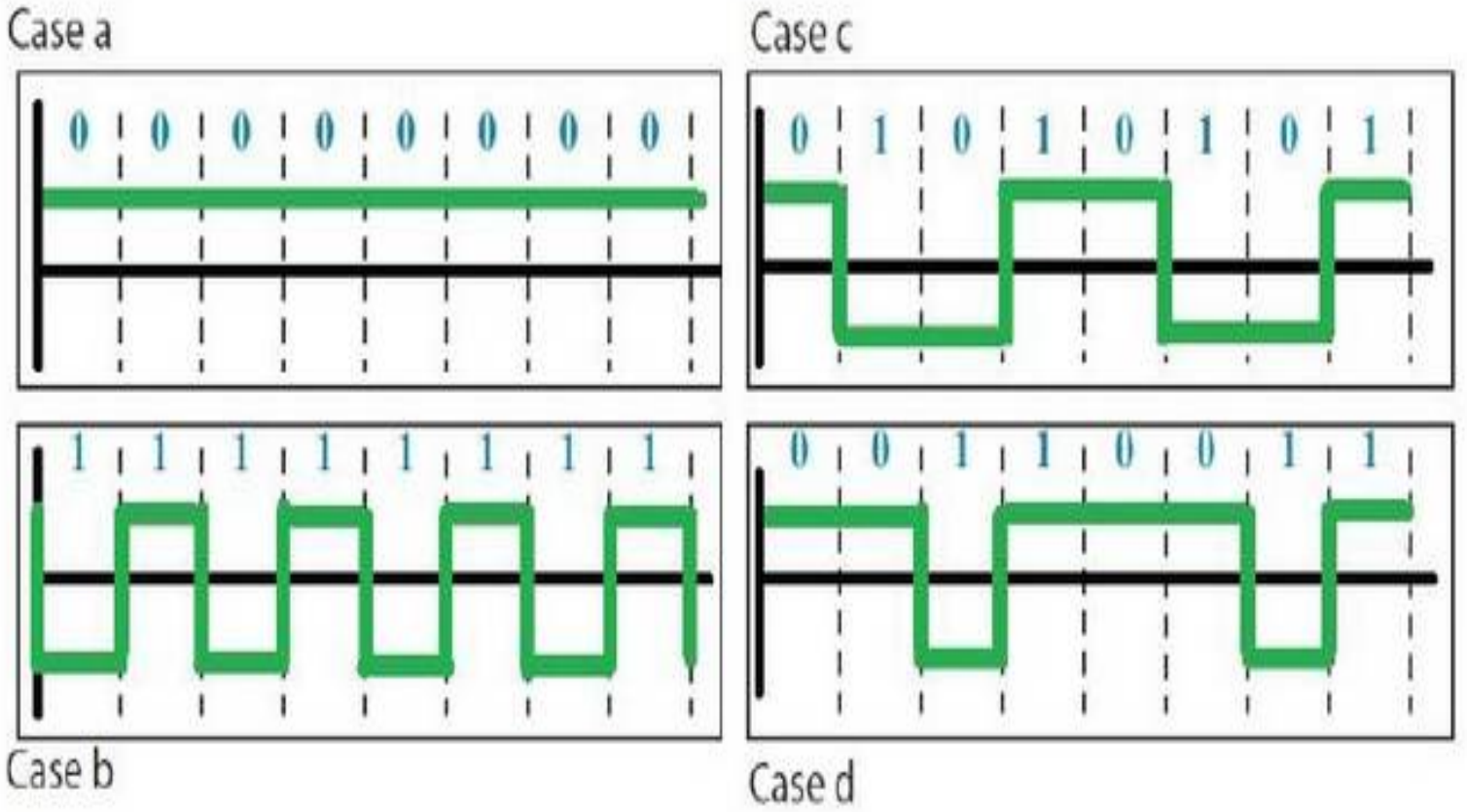
  a) **00000000**
  b) **11111111**
  c) **01010101**
  d) **00110011**

NRZ- L

Ans:


Case a
0 0 0 0 0 0 0 0
Case c
0 1 0 1 0 1 0 1
1 1 1 1 1 1 1 1
0 0 1 1 0 0 1 1
Case b
Case d

NRZ - I


Case a
0 0 0 0 0 0 0 0
Case c
0 1 0 1 0 1 0 1
1 1 1 1 1 1 1 1
Case b
0 0 1 1 0 0 1 1
Case d

4.1.2 Line Coding Schemes Contd.

- Comparison of these two schemes

- Although baseline wandering is a problem for both variations, it is twice as severe in NRZ-L.
- If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes skewed.
- The receiver might have difficulty discerning the bit value.
- In NRZ-I this problem occurs only for a long sequence of 0s.
- If somehow we can eliminate the long sequence of 0s, we can avoid baseline wandering.

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

4.1.2 Line Coding Schemes Contd.

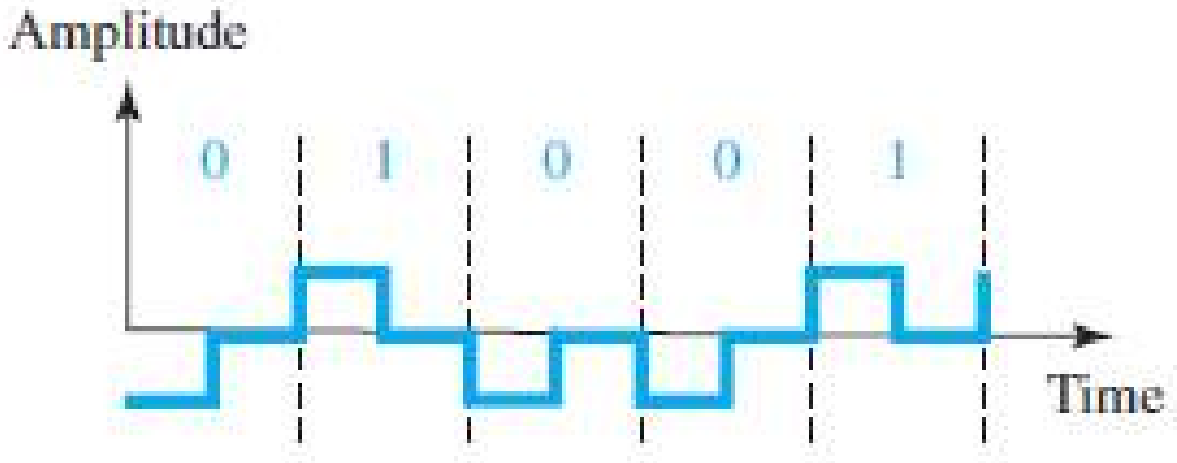
- Another problem with NRZ-L occurs when there is a sudden change of polarity in the system.
- For example, if twisted-pair cable is the medium, a change in the polarity of the wire results in all 0s interpreted as 1s and all 1s interpreted as 0s.
- NRZ-I does not have this problem.
- Both schemes have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

4.1.2 Line Coding Schemes Contd.

- ***Return-to-Zero (RZ)***
  - The main problem with NRZ encoding occurs when the sender and receiver clocks are not synchronized. The receiver does not know when one bit has ended and the next bit is starting.
  - One solution is the return-to-zero (RZ) scheme, which uses three values: positive, negative, and zero.
  - In RZ, the signal changes not between bits but during the bit.
  - In Figure 4.7 we see that the signal goes to 0 in the middle of each bit.
  - It remains there until the beginning of the next bit.

4.1.2 Line Coding Schemes Contd.

Figure 4.7 *Polar RZ scheme*



4.1.2 Line Coding Schemes Contd.

- The main disadvantage of RZ encoding is that it requires two signal changes to encode a bit and therefore occupies greater bandwidth.
- Sudden change of polarity resulting in all 0s interpreted as 1s and all 1s interpreted as 0s, still exists here, but there is no DC component problem.
- Another problem is the complexity: RZ uses three levels of voltage, which is more complex to create and discern.
- As a result of all these deficiencies, the scheme is not used today.
- Instead, it has been replaced by the better-performing Manchester and differential Manchester schemes

4.1.2 Line Coding Schemes Contd.

- **Biphase: Manchester and Differential Manchester**

- The idea of RZ (transition at the middle of the bit) and the idea of NRZ-L are combined into the Manchester scheme.

- In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half.

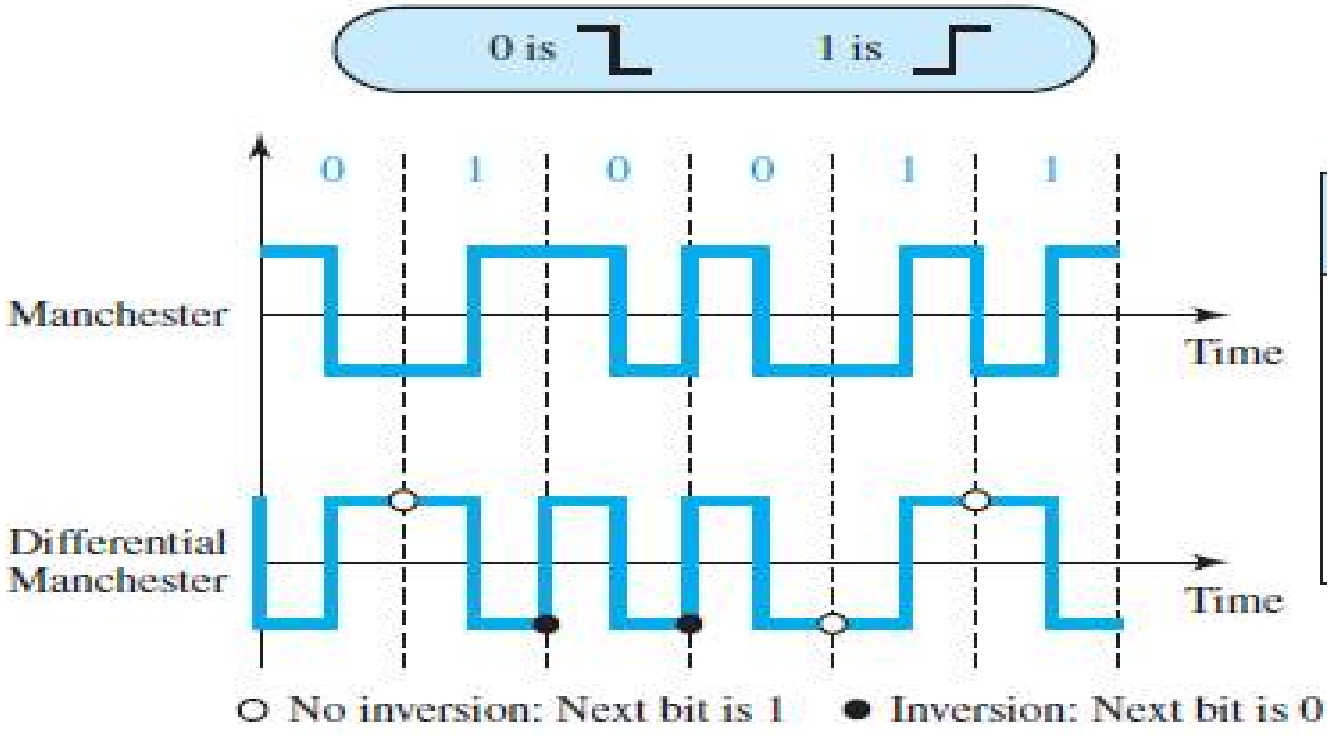
- The transition at the middle of the bit provides synchronization.

4.1.2 Line Coding Schemes Contd.

- Differential Manchester, on the other hand, combines the ideas of RZ and NRZ-I.
- There is always a transition at the middle of the bit, but the bit values are determined at the beginning of the bit.
- If the next bit is 0, there is a transition; if the next bit is 1, there is none.
- Figure 4.8 shows both Manchester and differential Manchester encoding.

4.1.2 Line Coding Schemes Contd.

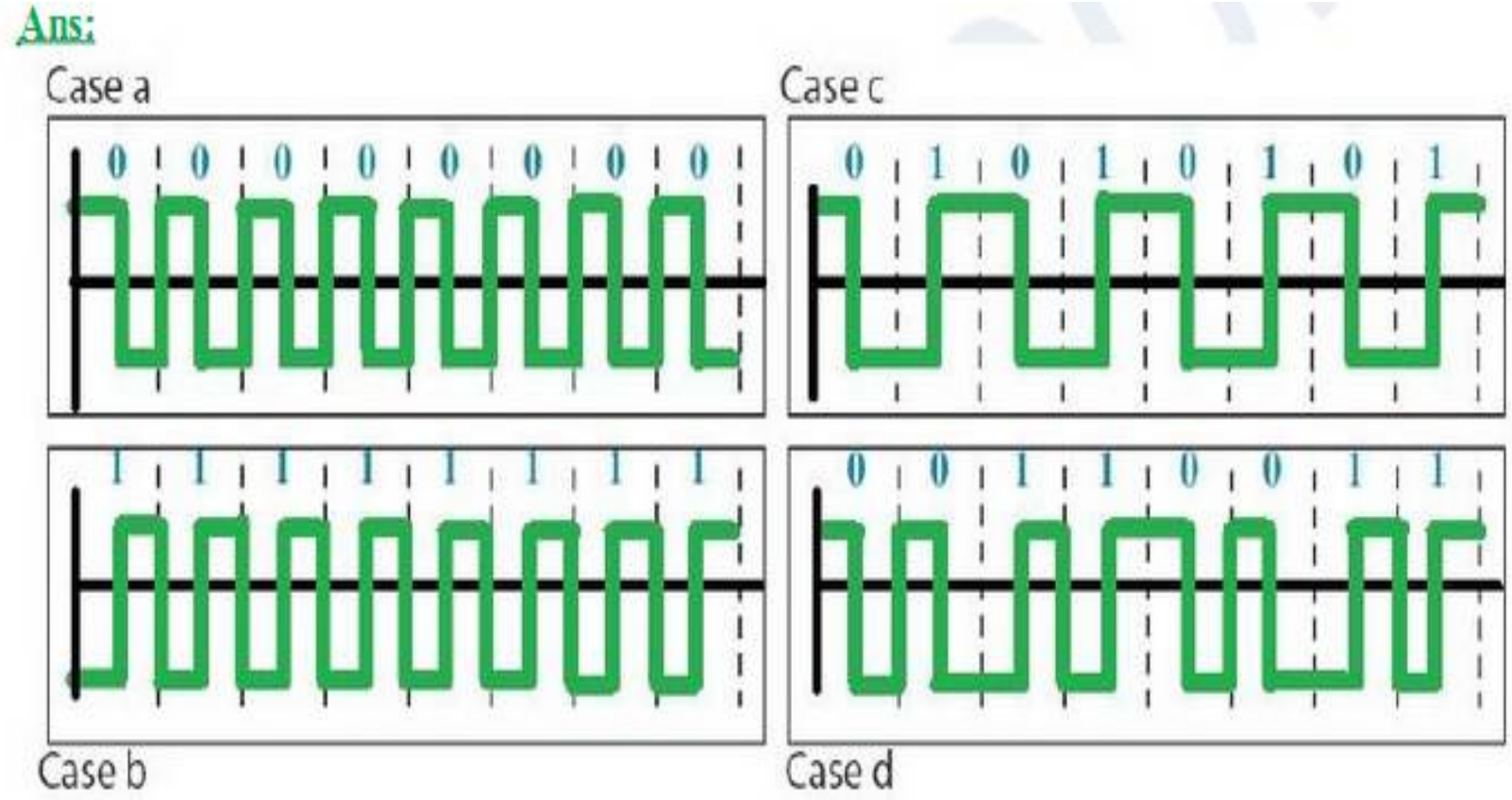
Figure 4.8 *Polar biphase: Manchester and differential Manchester schemes*



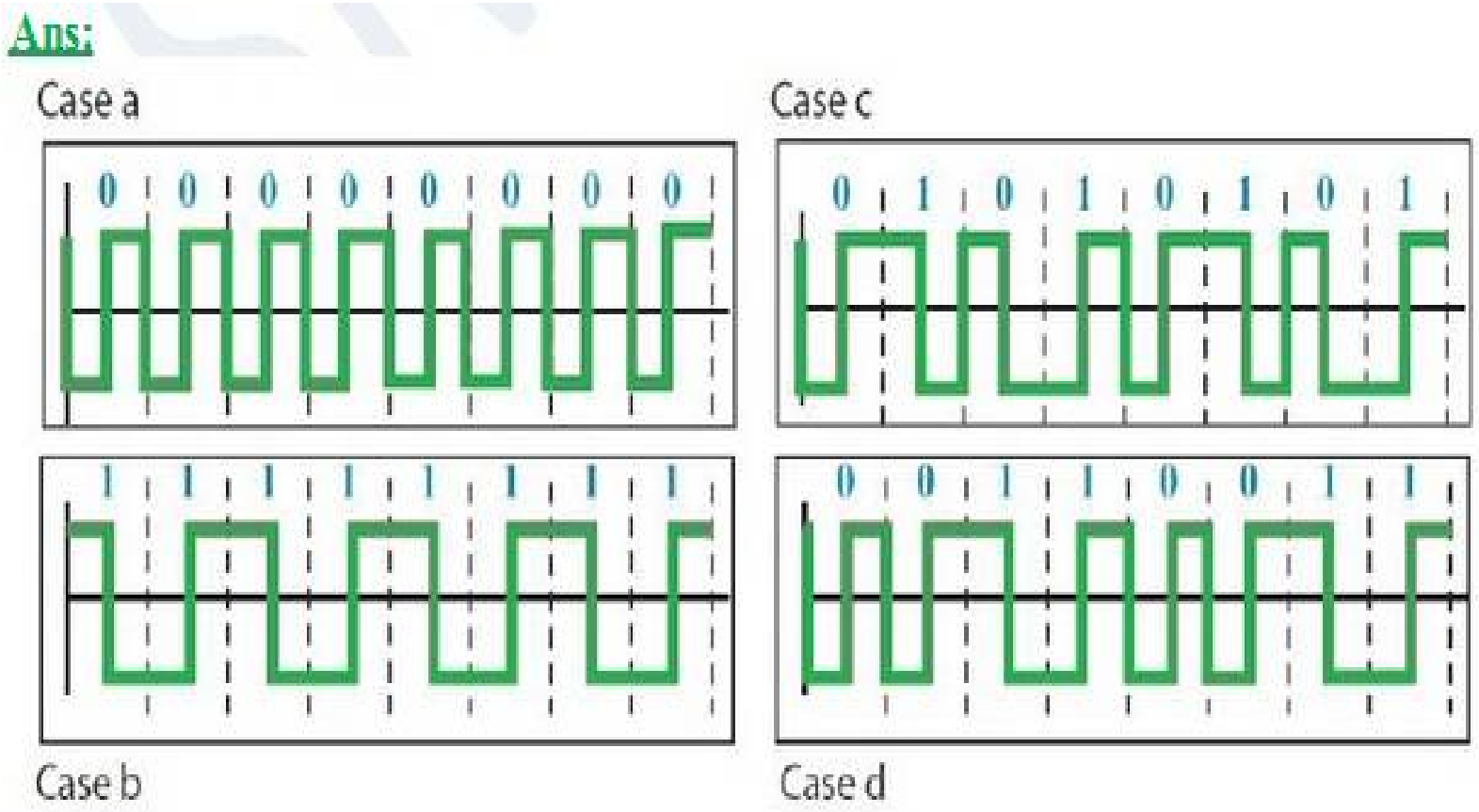
- **Draw the graph of the Manchester, Differential Manchester, using each of the following data streams, assuming that the last signal level has been positive**

  a) **00000000**
  b) **11111111**
  c) **01010101**
  d) **00110011**

Manchester Scheme



Differential Manchester Scheme



4.1.2 Line Coding Schemes Contd.

- The Manchester scheme overcomes several problems associated with NRZ-L, and differential Manchester overcomes several problems associated with NRZ-I.
- First, there is no baseline wandering.
- There is no DC component because each bit has a positive and negative voltage contribution.
- The only drawback is the signal rate.
- The signal rate for Manchester and differential Manchester is double that for NRZ.
- The reason is that there is always one transition at the middle of the bit and maybe one transition at the end of each bit.
- Note that Manchester and differential Manchester schemes are also called biphase schemes.
- The minimum bandwidth of Manchester and differential Manchester is 2 times that of NRZ.

4.1.3 Block Coding

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

4.1.3 Block Coding Contd.

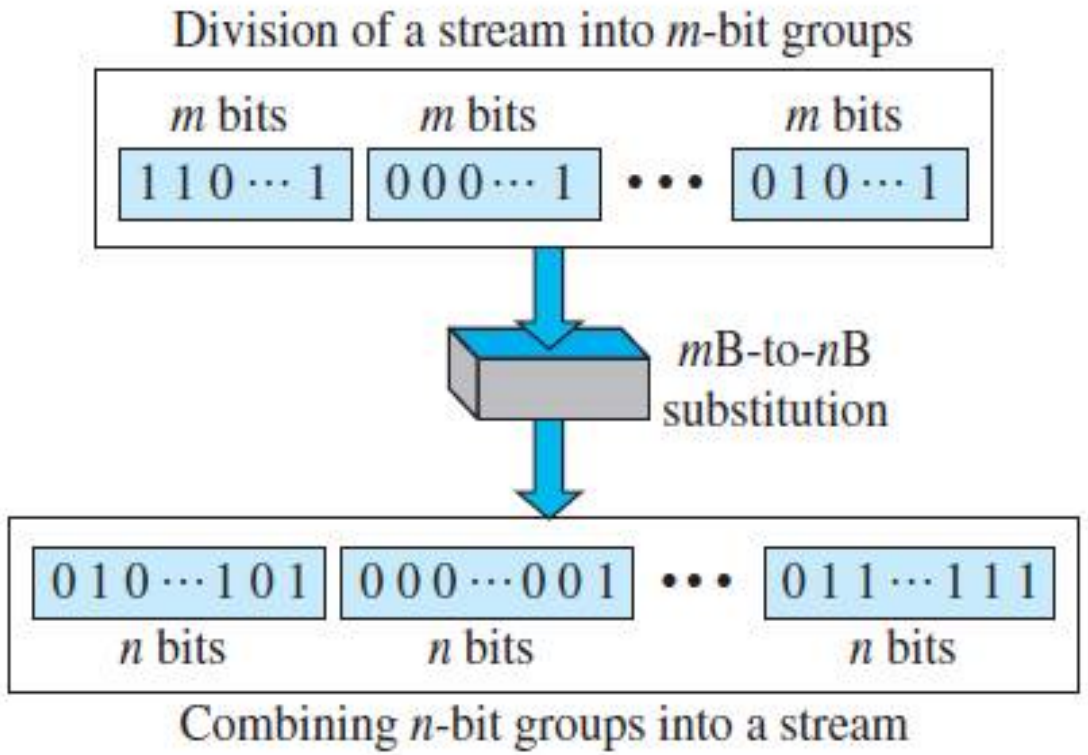
- The slash in block encoding (for example, 4B/5B) distinguishes block encoding from multilevel encoding (for example, 8B6T), which is written without a slash.
- Block coding normally involves three steps: division, substitution, and combination.
- In the division step, a sequence of bits is divided into groups of m bits.
- For example, in 4B/5B encoding, the original bit sequence is divided into 4-bit groups.

4.1.3 Block Coding Contd.

- The heart of block coding is the substitution step.
- In this step, we substitute an m-bit group with an n-bit group.
- For example, in 4B/5B encoding we substitute a 4-bit group with a 5-bit group.
- Finally, the n-bit groups are combined to form a stream.
- The new stream has more bits than the original bits. Figure 4.14 shows the procedure.

4.1.3 Block Coding Contd.

Figure 4.14 *Block coding concept*



4.1.3 Block Coding Contd.

- ***4B/5B***

- The four binary/five binary (4B/5B) coding scheme was designed to be used in combination with NRZ-I.

- Recall that NRZ-I has a good signal rate, one-half that of the biphase, but it has a synchronization problem.

- A long sequence of 0s can make the receiver clock lose synchronization.

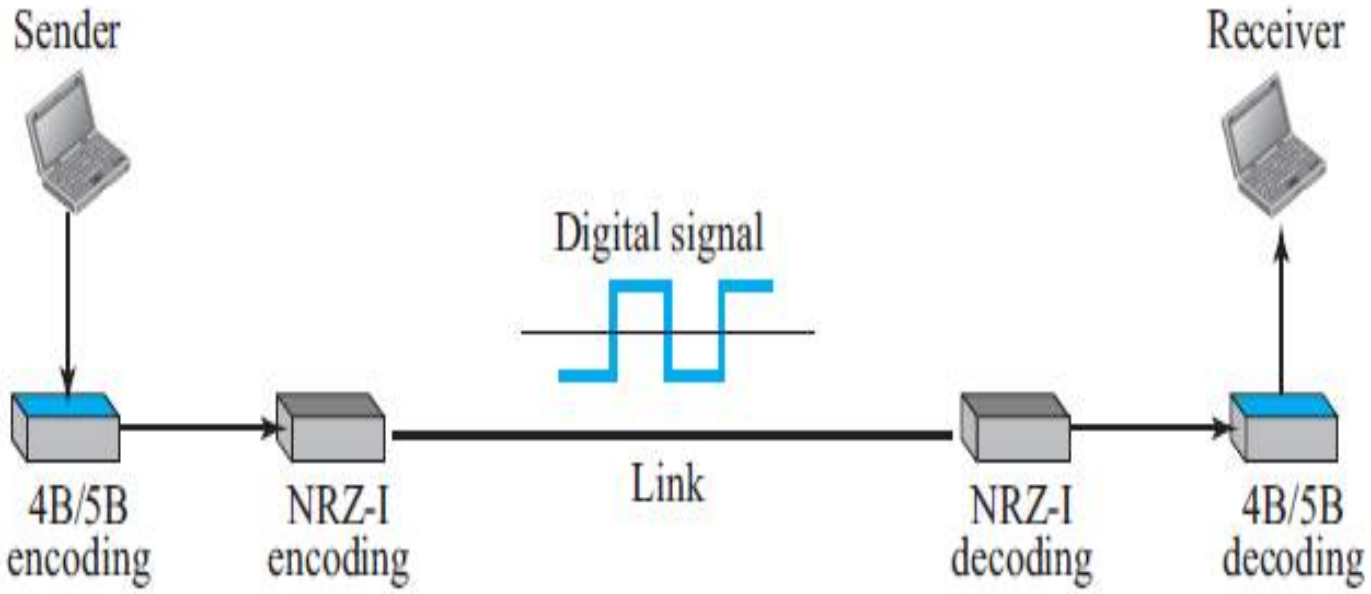
- One solution is to change the bit stream, prior to encoding with NRZ-I, so that it does not have a long stream of 0s.

4.1.3 Block Coding Contd.

- The 4B/5B scheme achieves this goal.
- The block-coded stream does not have more than three consecutive 0s.
- At the receiver, the NRZ-I encoded digital signal is first decoded into a stream of bits and then decoded to remove the redundancy.
- Figure 4.15 shows the idea.

4.1.3 Block Coding Contd.

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



4.1.3 Block Coding Contd.

- In 4B/5B, the 5-bit output that replaces the 4-bit input has no more than one leading zero (left bit) and no more than two trailing zeros (right bits).
- So when different groups are combined to make a new sequence, there are never more than three consecutive 0s.
- Table 4.2 shows the corresponding pairs used in 4B/5B encoding.
- Note that the first two columns pair a 4-bit group with a 5-bit group.
- A group of 4 bits can have only 16 different combinations while a group of 5 bits can have 32 different combinations.

4.1.3 Block Coding Contd.

- This means that there are 16 groups that are not used for 4B/5B encoding.
- Some of these unused groups are used for control purposes; the others are not used at all.
- The latter provide a kind of error detection.
- If a 5-bit group arrives that belongs to the unused portion of the table, the receiver knows that there is an error in the transmission.

4.1.3 Block Coding Contd.

Table 4.2 *4B/5B mapping codes*

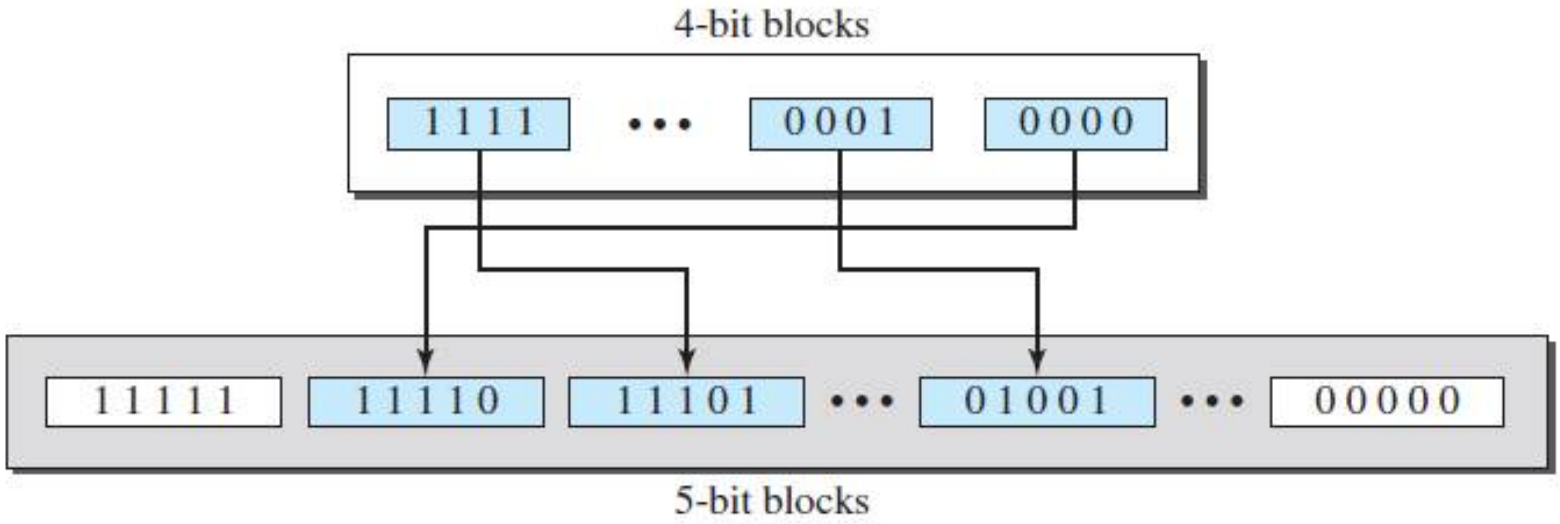
Data Sequence	*Encoded Sequence*	*Control Sequence*	*Encoded Sequence*
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)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

4.1.3 Block Coding Contd.

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

4.1.3 Block Coding Contd.

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



- **The input stream to a 4B/5B block encoder is**

0100 0000 0000 0000 0000 0001 Answer the following questions:

- **What is the output stream?**
- **What is the length of the longest consecutive sequence of 0s in the input?**
- **What is the length of the longest consecutive sequence of 0s in the output?**

- **The input stream to a 4B/5B block encoder is**

0100 0000 0000 0000 0000 0001 Answer the following questions:

- **What is the output stream?**
- **What is the length of the longest consecutive sequence of 0s in the input?**
- **What is the length of the longest consecutive sequence of 0s in the output?**

Ans: a. The output stream is 01010 11110 11110 11110 11110 01001.
b. The maximum length of consecutive 0s in the input stream is 21.
c. The maximum length of consecutive 0s in the output stream is 2.

- **How many invalid (unused) code sequences can we have in 5B/6B encoding? How many in 3B/4B encoding?**

- **How many invalid (unused) code sequences can we have in 5B/6B encoding? How many in 3B/4B encoding?**

Ans: In 5B/6B, we have $2^5 = 32$ data sequences and $2^6 = 64$ code sequences. The number of unused code sequences is $64 - 32 = 32$. In 3B/4B, we have $2^3 = 8$ data sequences and $2^4 = 16$ code sequences. The number of unused code sequences is $16 - 8 = 8$.

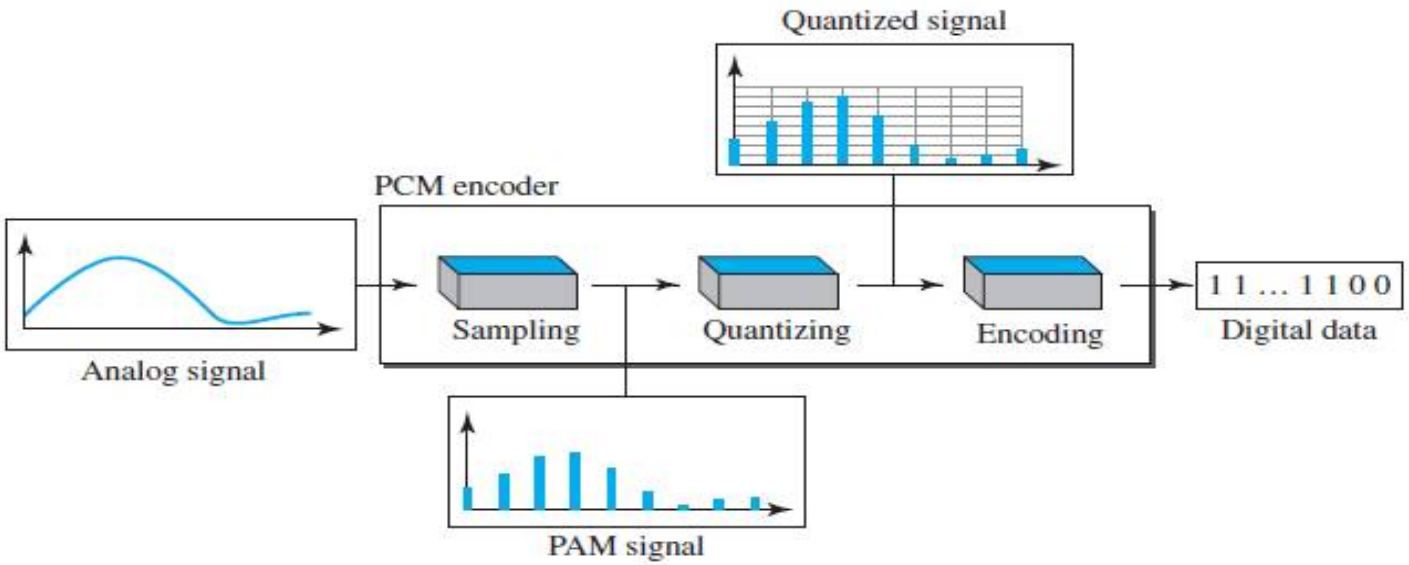
4.2 ANALOG-TO-DIGITAL CONVERSION

- Sometimes we have an analog signal such as one created by a microphone or camera.
- A digital signal is superior to an analog signal.
- The tendency today is to change an analog signal to digital data.
- Two techniques used are pulse code modulation and delta modulation.
- After the digital data are created (digitization), we can use one of the techniques like line coding, block coding and scrambling to convert the digital data to a digital signal.

4.2.1 Pulse Code Modulation (PCM)

- Changes an analog signal to digital data
- A PCM encoder has three processes
  - 1. The analog signal is sampled.
  - 2. The sampled signal is quantized.
  - 3. The quantized values are encoded as streams of bits.

Figure 4.21 *Components of PCM encoder*



4.2.1 Pulse Code Modulation Contd.

- **Sampling**
- The first step in PCM is sampling. Also referred to as pulse amplitude Modulation (PAM).
- The analog signal is sampled every Ts seconds, where Ts is the sample interval or period.
- There are three sampling methods—ideal, natural, and flat-top
- According to the Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal.

4.2.1 Pulse Code Modulation Contd.

- **Quantization**
- The result of sampling is a series of pulses with amplitude values between the maximum and minimum amplitudes of the signal.

- The set of amplitudes can be infinite with nonintegral values between the two limits.

- These values cannot be used in the encoding process.

- The following are the steps in quantization:
  - 1. We assume that the original analog signal has instantaneous amplitudes between Vmin and Vmax.

4.2.1 Pulse Code Modulation Contd.

- As a simple example, assume that we have a sampled signal and the sample amplitudes are between –20 and +20 V.

- We decide to have eight levels (*L = 8). This means that* Δ = 5 V.

- **Quantization Levels**
- The choice of *L, the* number of levels, depends on the range of the amplitudes of the analog signal and how accurately we need to recover the signal.

- If the amplitude of a signal fluctuates between two values only, we need only two levels; if the signal, like voice, has many amplitude values, we need more quantization levels.

- Choosing lower values of *L increases the* quantization error if there is a lot of fluctuation in the signal.

4.2.1 Pulse Code Modulation Contd.

- **Quantization Error**
- The input values to the quantizer are the real values; the output values are the approximated values.
- The output values are chosen to be the middle value in the zone.
- If the input value is also at the middle of the zone, there is no quantization error; otherwise, there is an error.
- The value of the error for any sample is less than $\Delta/2$. In other words, we have $-\Delta/2 \leq \text{error} \leq \Delta/2$.

4.2.1 Pulse Code Modulation Contd.

- The quantization error changes the signal-to-noise ratio of the signal, which in turn reduces the upper limit capacity according to Shannon.

- It can be proven that the contribution of the quantization error to the SNRdB of the signal depends on the number of quantization levels L, or the bits per sample nb, as shown in the following formula:

$$\mathrm{SNR}_{\mathrm{dB}} = 6.02 n_b + 1.76 \quad \mathrm{dB}$$

4.2.1 Pulse Code Modulation Contd.

- **Uniform Versus Nonuniform Quantization**

- For many applications, the distribution of the instantaneous amplitudes in the analog signal is not uniform.

- Changes in amplitude often occur more frequently in the lower amplitudes than in the higher ones.

- For these types of applications it is better to use nonuniform zones.

- In other words, the height of Δ is not fixed; it is greater near the lower amplitudes and less near the higher amplitudes.

4.2.1 Pulse Code Modulation Contd.

- Nonuniform quantization can also be achieved by using a process called companding and expanding.
- The signal is companded at the sender before conversion; it is expanded at the receiver after conversion.
- Companding means reducing the instantaneous voltage amplitude for large values; expanding is the opposite process.
- Companding gives greater weight to strong signals and less weight to weak ones.

4.2.1 Pulse Code Modulation Contd.

- **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.

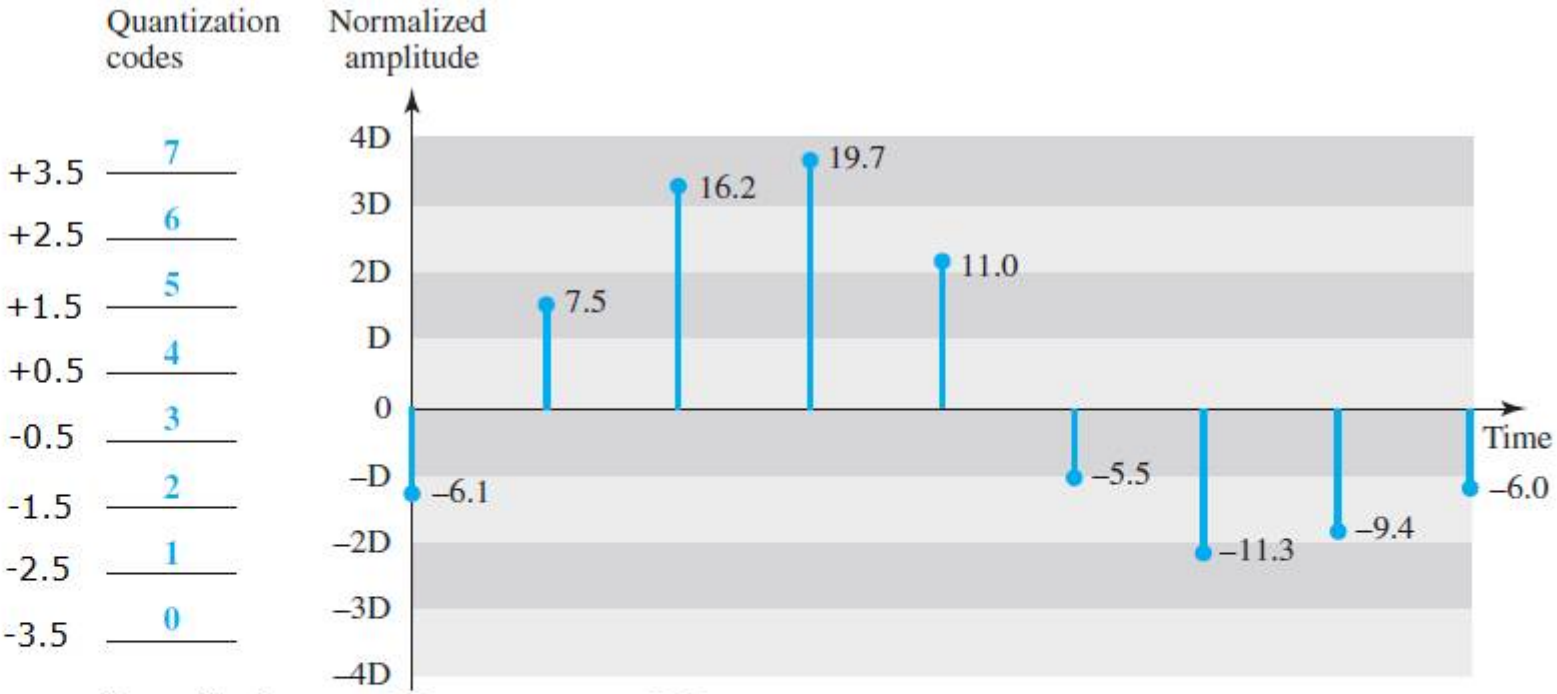
- A quantization code of 2 is encoded as 010; 5 is encoded as 101; and so on.

- Note that the number of bits for each sample is determined from the number of quantization levels i.e nb = log2 L. In our example L is 8 and nb is therefore 3.

- The bit rate can be found from the formula

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

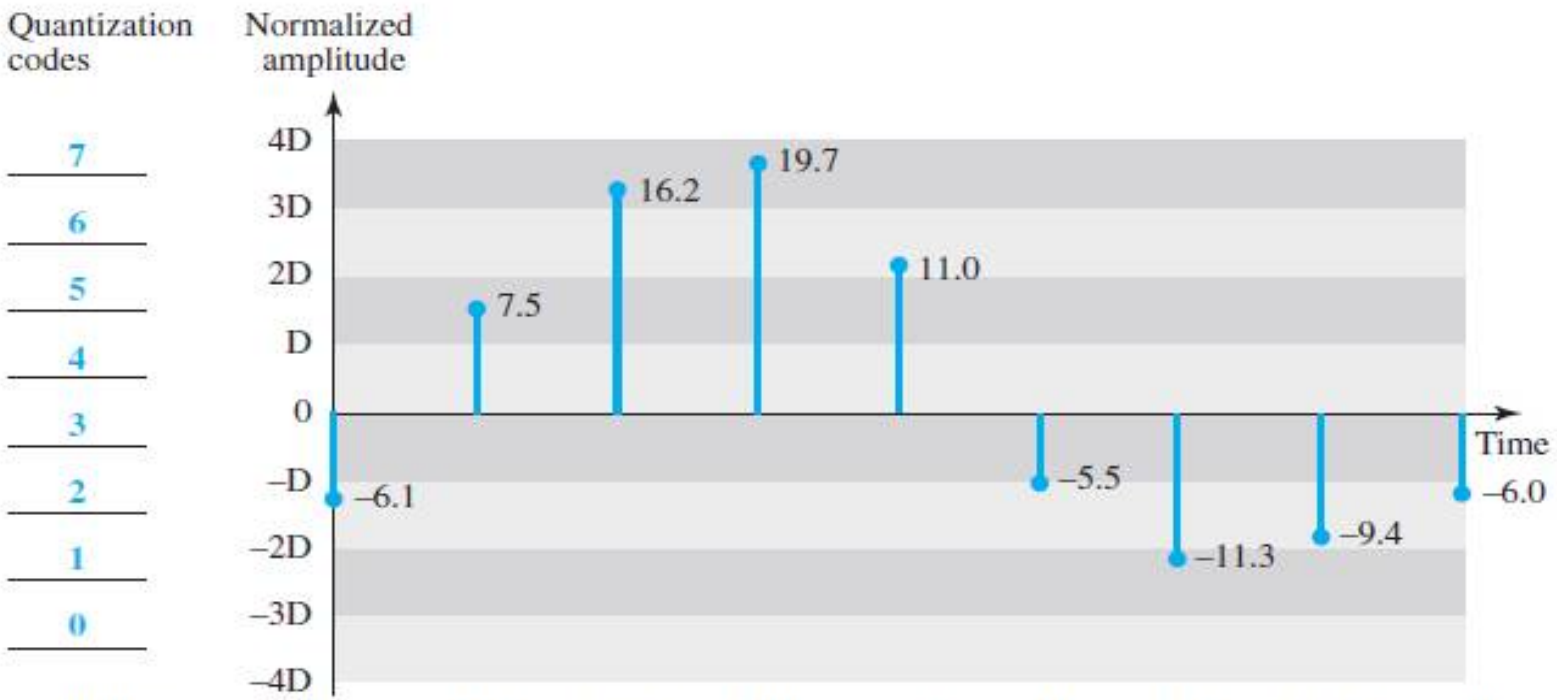
4.2.1 Pulse Code Modulation Contd.



(Amplitude/delta)	Normalized PAM values	–1.22	3.24
(mid value)	Normalized quantized values	–1.50	3.50
(Quantized-PAM values)	Normalized error	-0.28	+0.26
	Quantization code	2	7
	Encoded words	010	111

4.2.1 Pulse Code Modulation Contd.

Figure 4.26 *Quantization and encoding of a sampled signal*



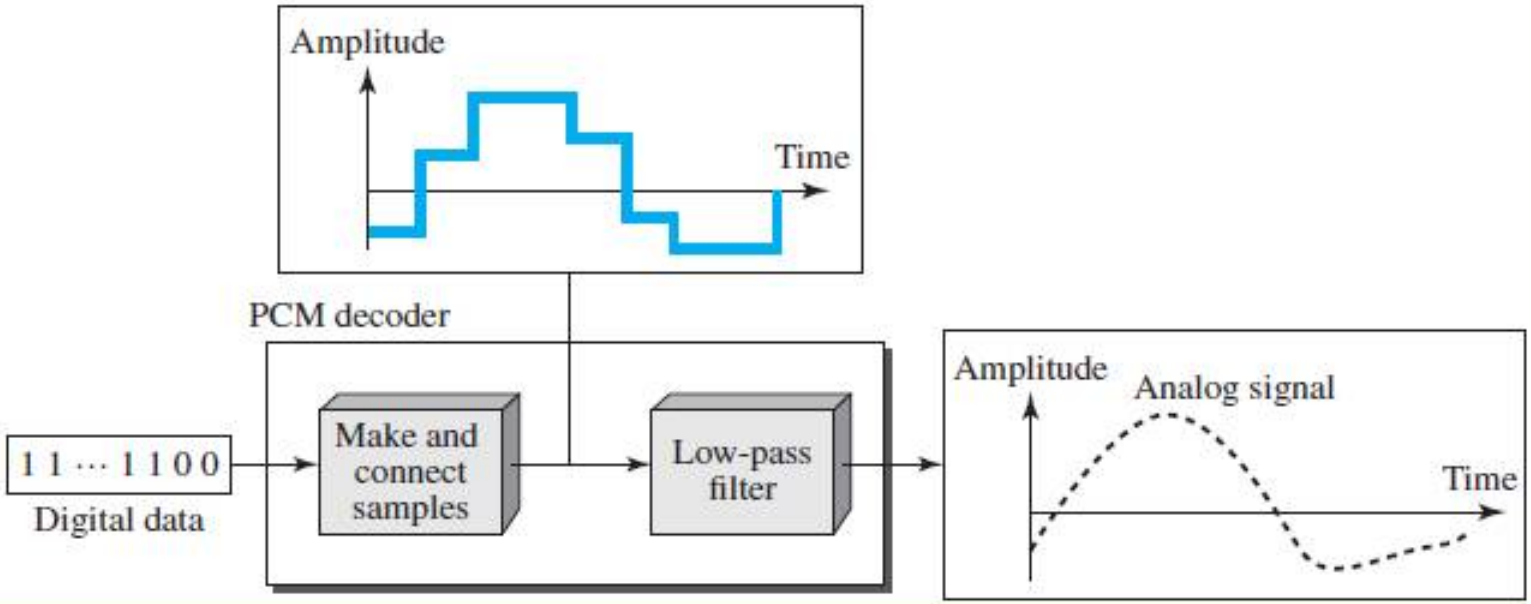
Normalized PAM values	–1.22	1.50	3.24	3.94	2.20	–1.10	–2.26	–1.88	–1.20
Normalized quantized values	–1.50	1.50	3.50	3.50	2.50	–1.50	–2.50	–1.50	–1.50
Normalized error	–0.38	0	+0.26	–0.44	+0.30	–0.40	–0.24	+0.38	–0.30
Quantization code	2	5	7	7	6	2	1	2	2
Encoded words	010	101	111	111	110	010	001	010	010

4.2.1 Pulse Code Modulation Contd.

- **Original Signal Recovery**
- The recovery of the original signal requires the PCM decoder.
- The decoder first uses circuitry to convert the code words into a pulse that holds the amplitude until the next pulse.
- After the staircase signal is completed, it is passed through a low-pass filter to smooth the staircase signal into an analog signal.
- The filter has the same cutoff frequency as the original signal at the sender.
- If the signal has been sampled at the Nyquist sampling rate and if there are enough quantization levels, the original signal will be recreated.

4.2.1 Pulse Code Modulation Contd.

Figure 4.27 *Components of a PCM decoder*



END OF CHAPTER 4

Chapter 5
Analog Transmission

Reference:

Data Communication and Networking, Behrouz A.Forouzan, McGraw Hill, 5th Edition, 2008

Note to Students : ppt is for revision purpose only. Answers in internals and exams should be written elaborately as given in the prescribed text book

Introduction

- **Baseband Transmission**
  - Baseband transmission means sending a digital signal over a channel without changing the digital signal to an analog signal
  - Baseband transmission requires that we have a **low-pass channel,** a channel with a bandwidth that starts from zero

- **Broadband transmission**
  - Broadband transmission means changing the digital signal to an analog signal for transmission.
  - Broadband allows us to use a **bandpass channel—a channel** with a bandwidth that does not start from zero

- Converting digital data to a bandpass analog signal is traditionally called digital-to-analog conversion – ASK, FSK, PSK and QAM

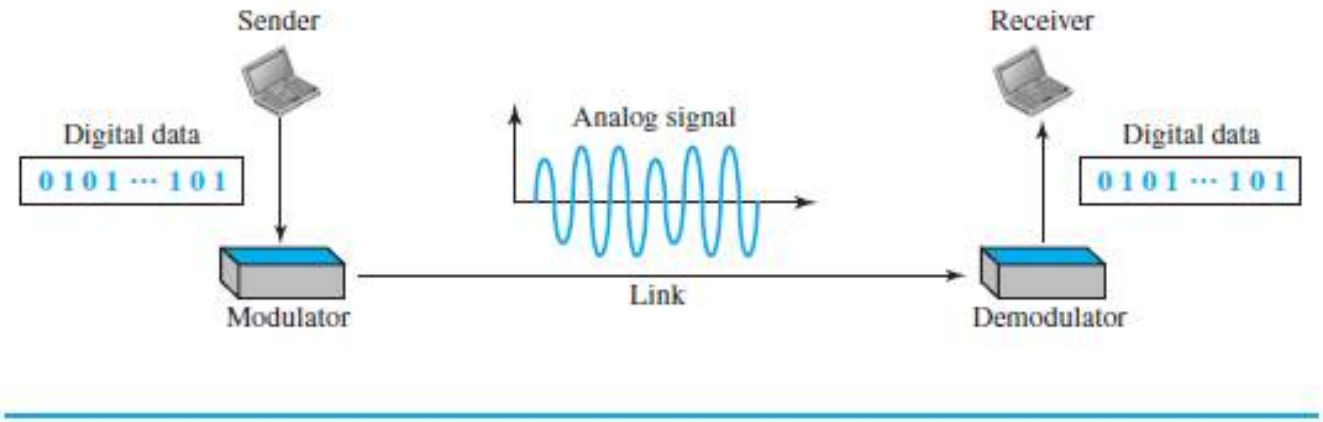
- Converting a low-pass analog signal to a bandpass analog signal is traditionally called analog-to-analog conversion – AM, FM and PM

5.1 DIGITAL-TO-ANALOG CONVERSION

- **process of changing one of the characteristics of** an analog signal based on the information in digital data.

- Figure 5.1 shows the relationship between the digital information, the digital-to-analog modulating process, and the resultant analog signal.

Figure 5.1 *Digital-to-analog conversion*



5.1 DIGITAL-TO-ANALOG CONVERSION Contd..

- A sine wave is defined by three characteristics: amplitude, frequency, and phase.

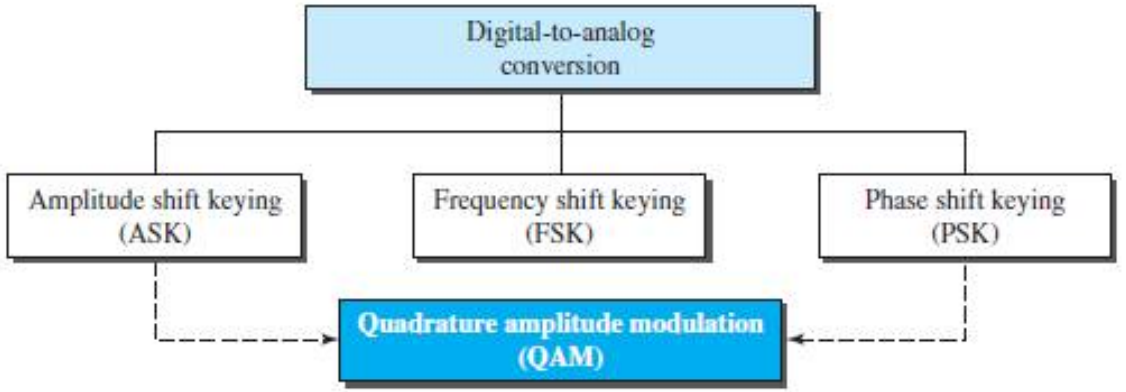
- By changing one characteristic of a simple electric signal, we can use it to represent digital data.

- 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).

5.1 DIGITAL-TO-ANALOG CONVERSION Contd..

- In addition, there is a fourth 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

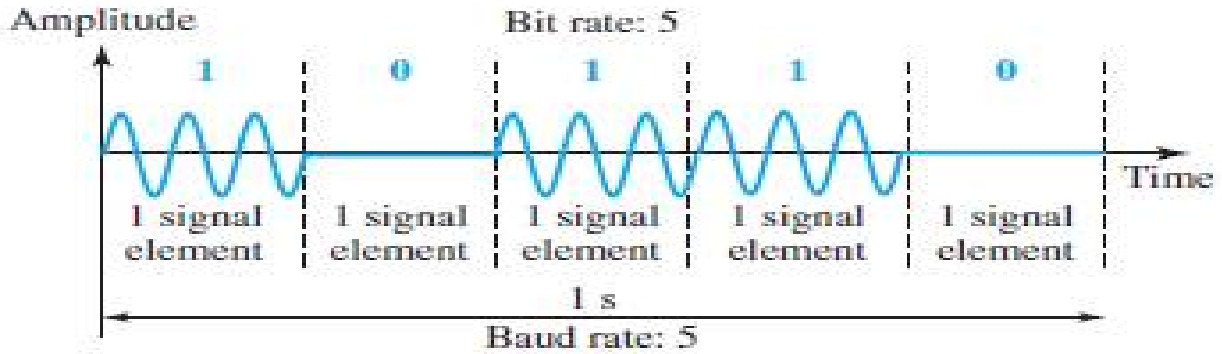
Figure 5.2 *Types of digital-to-analog conversion*



5.1.2 Amplitude Shift Keying

- In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements.
- Both frequency and phase remain constant while the amplitude changes.
- ***Binary ASK (BASK)***
  - 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

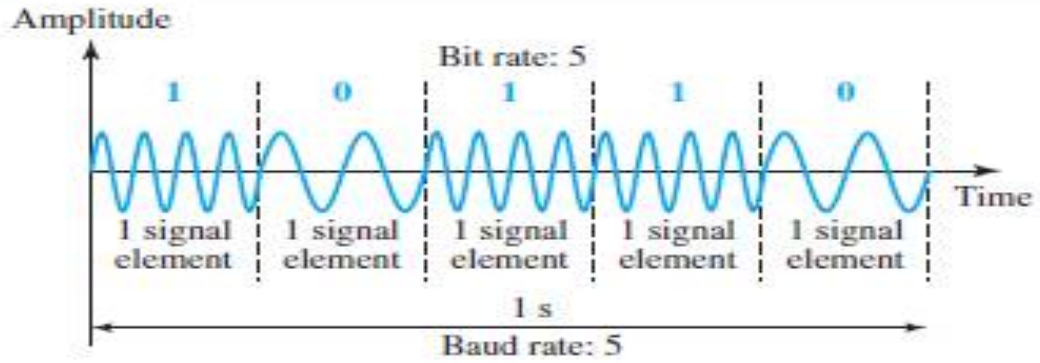
Figure 5.3 *Binary amplitude shift keying*



5.1.3 Frequency Shift Keying

- In frequency shift keying, the frequency of the carrier signal is varied to represent data. Both peak amplitude and phase remain constant for all signal elements.
- 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.
- ***Binary FSK (BFSK)***
- We use the first carrier frequency f1 if the data element is 0; we use the second carrier frequency if the data element is 1.

Figure 5.6 *Binary frequency shift keying*



5.1.4 Phase Shift Keying

- In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements.
- Both peak amplitude and frequency remain constant as the phase changes.
- **Binary PSK (BPSK)**
  - We have only two signal elements, one with a phase of 0°, and the other with a phase of 180°.

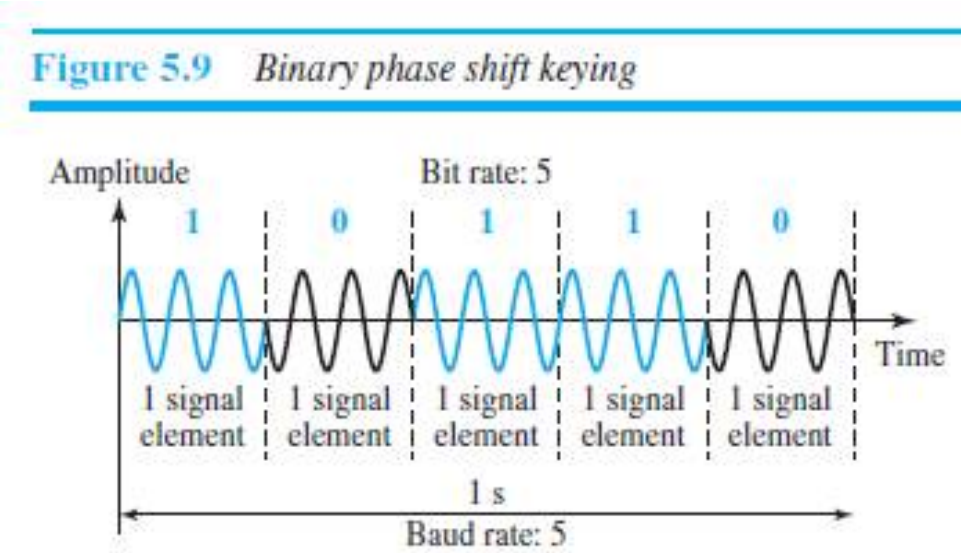


Figure 5.9 *Binary phase shift keying*

5.2 ANALOG-TO-ANALOG CONVERSION

- Analog-to-analog conversion, or analog modulation, is the representation of analog information by an analog signal.

- Modulation is needed if the medium is bandpass in nature or if only a bandpass channel is available to us.

- An example is radio. The government assigns a narrow bandwidth to each radio station.

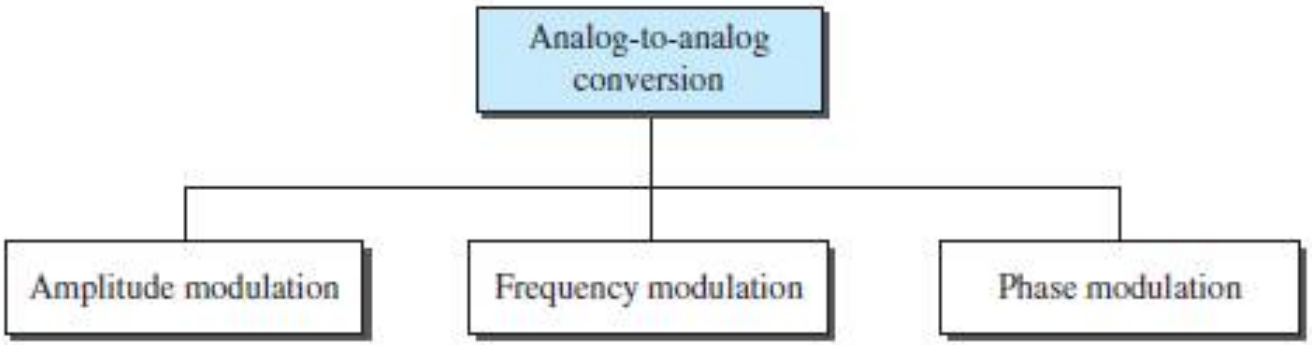
- The analog signal produced by each station is a low-pass signal, all in the same range.

- To be able to listen to different stations, the low-pass signals need to be shifted, each to a different range.

ANALOG-TO-ANALOG CONVERSION Contd..

- Analog-to-analog conversion can be accomplished in three ways:
  - Amplitude modulation (AM),
  - Frequency modulation (FM), and
  - Phase modulation (PM).

Figure 5.15 *Types of analog-to-analog modulation*

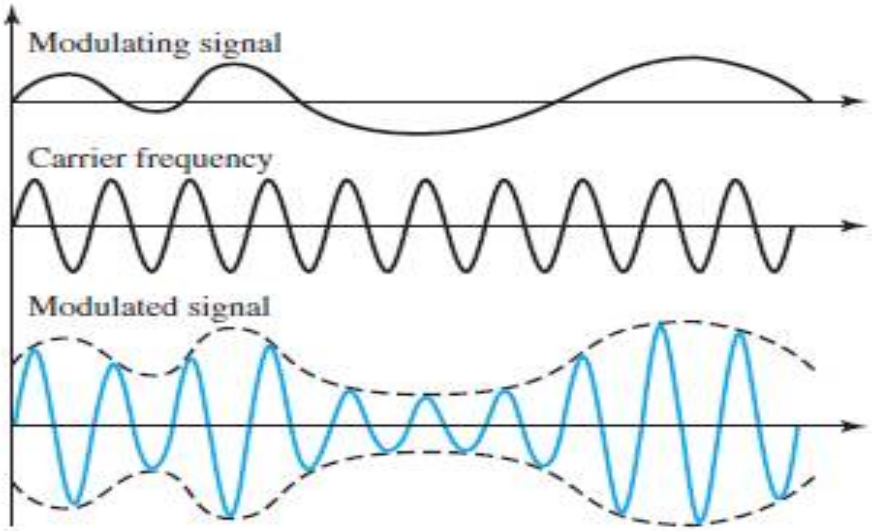


5.2.1 Amplitude Modulation

- 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 *Amplitude modulation*

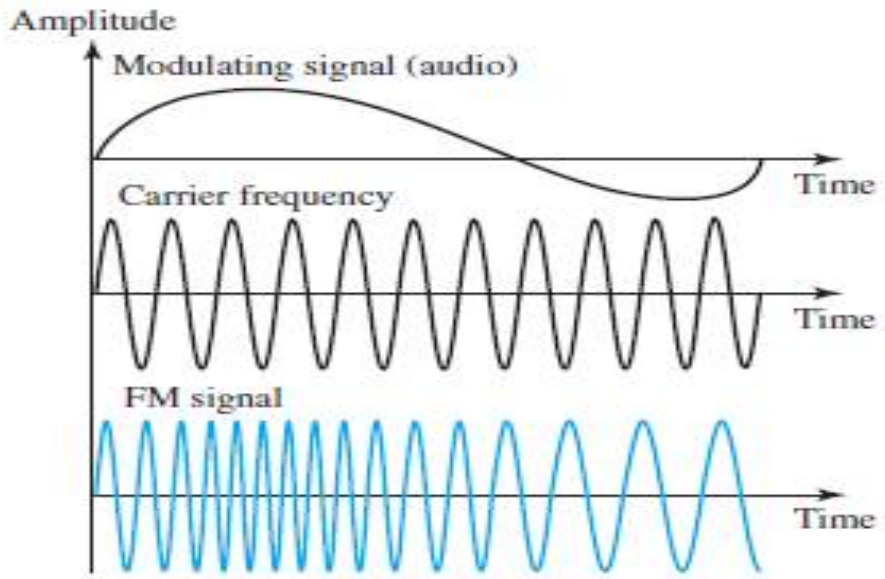


5.2.2 Frequency Modulation

- In FM transmission, the frequency of the carrier signal is modulated to follow the changing voltage level 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.

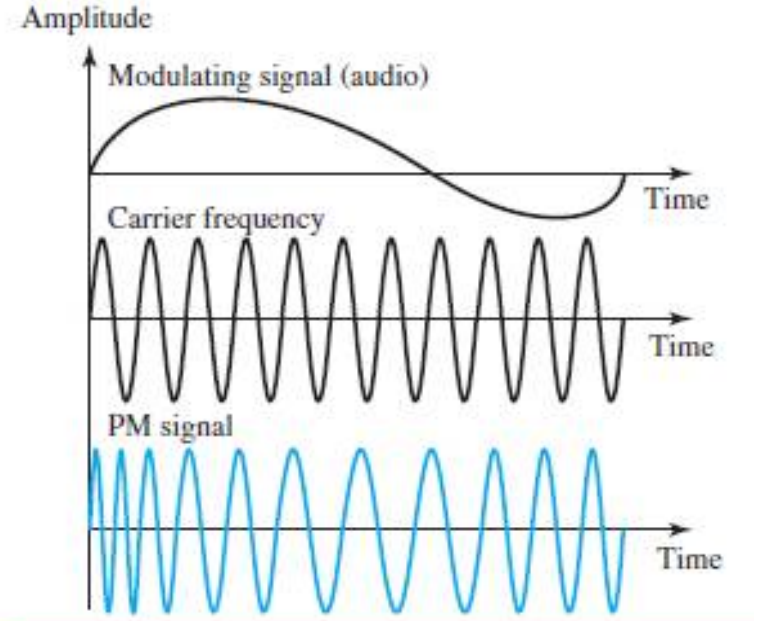
Figure 5.18 *Frequency modulation*



5.2.3 Phase Modulation

- In PM transmission, the phase of the carrier signal is modulated to follow the changing voltage level (amplitude) of the modulating signal.
- The peak amplitude and frequency of the carrier signal remain constant, but as the amplitude of the information signal changes, the phase of the carrier changes correspondingly.

Figure 5.20 *Phase modulation*



END OF CHAPTER 5

Chapter 6
Bandwidth Utilization: Multiplexing and Spectrum Spreading

Reference:

Data Communication and Networking, Behrouz A.Forouzan, McGraw Hill, 5th Edition, 2008

Note to Students : ppt is for revision purpose only. Answers in internals and exams should be written elaborately as given in the prescribed text book

6.1 Multiplexing

- Sometimes we need to expand the bandwidth of a channel to achieve goals such as privacy and anti jamming.
- Two broad categories of bandwidth utilization: multiplexing and spectrum spreading.
- In multiplexing, our goal is efficiency; we combine several low-bandwidth channels to make use of one channel with a larger bandwidth.
- In spectrum spreading, we expand the bandwidth of a channel to achieve goals such as privacy and anti jamming.
- 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 allow the simultaneous transmission of multiple signals across a single data link.

6.1 Multiplexing Contd..

- In a multiplexed system, n lines share the bandwidth of one link.
- Figure 6.1 shows the basic format of a multiplexed system.
- The lines on the left direct their transmission streams to a multiplexer (MUX), which combines them into a single stream (many-to one).
- At the receiving end, that stream is fed into a demultiplexer (DEMUX), which separates the stream back into its component transmissions (one-to-many) and directs them to their corresponding lines.
- In the figure, the word link refers to the physical path.
- The word channel refers to the portion of a link that carries a transmission between a given pair of lines.
- One link can have many (n) channels.

6.1 Multiplexing Contd..

Figure 6.1 *Dividing a link into channels*

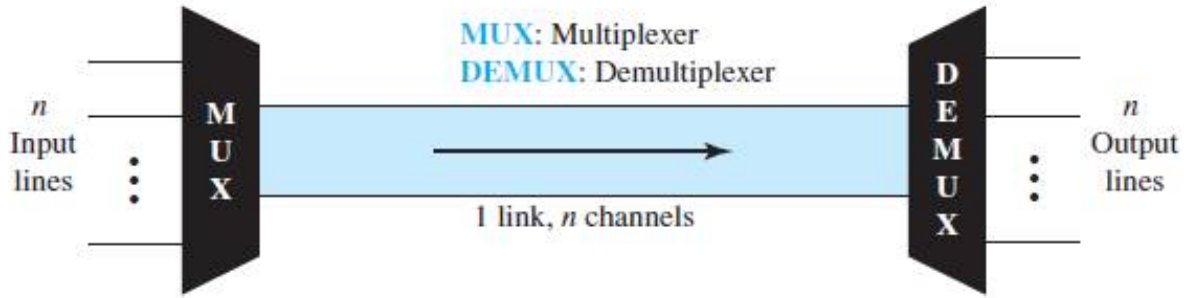
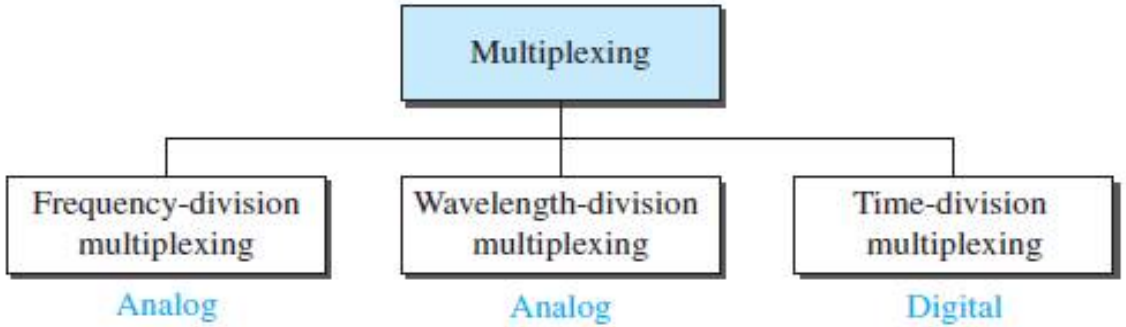


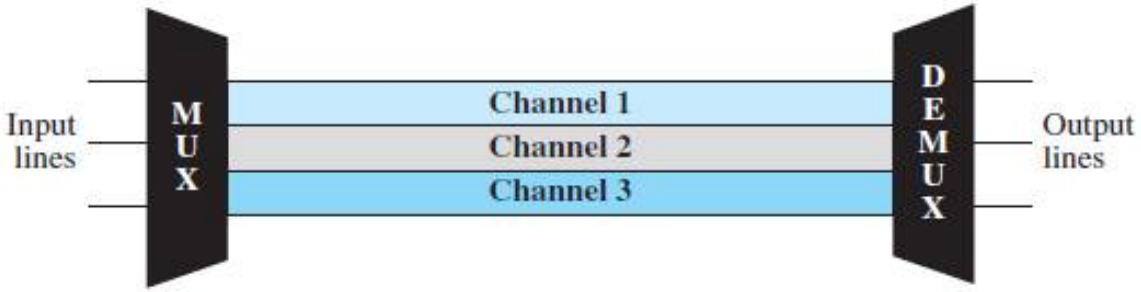
Figure 6.2 *Categories of multiplexing*



Frequency-division multiplexing (1)

- FDM is an analog technique that can be applied when the bandwidth of a link is greater than the combined bandwidths of the signals to be transmitted.
- In FDM, signals generated by each sending device modulate different carrier frequencies.
- These modulated signals are then combined into a single composite signal that can be transported by the link.
- Channels can be separated by strips of unused bandwidth—guard bands—to prevent signals from overlapping.
- Figure 6.3 gives a conceptual view of FDM.
- In this illustration, the transmission path is divided into three parts, each representing a channel that carries one transmission.

Figure 6.3 *Frequency-division multiplexing*

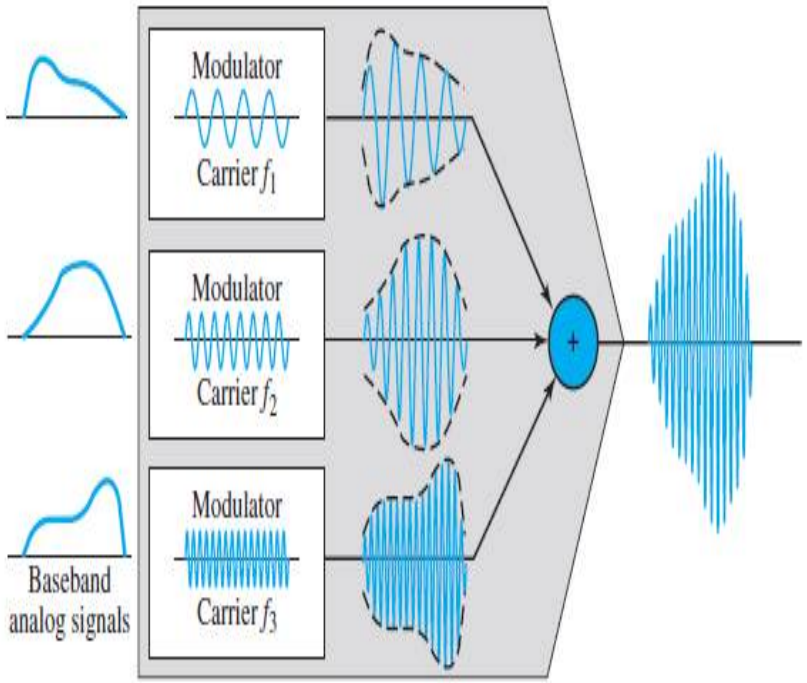


Frequency-division multiplexing (2)

- **Multiplexing Process**
  - Figure 6.4 is a conceptual illustration of the multiplexing process.
  - Each source generates a signal of a similar frequency range.
  - Inside the multiplexer, these similar signals modulate different carrier frequencies (f1, f2, and f3).
  - The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.
- **Demultiplexing Process**
  - The demultiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals.
  - The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines.
  - Figure 6.5 is a conceptual illustration of demultiplexing process.

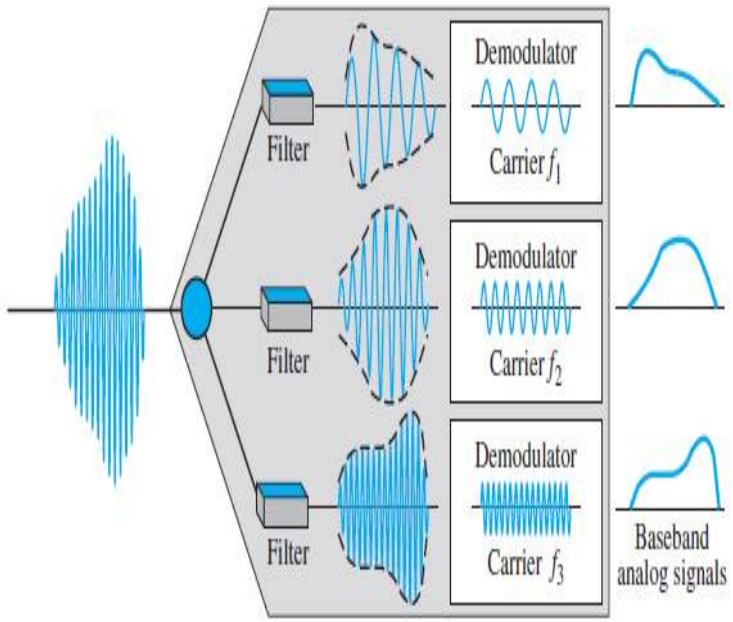
Frequency-division multiplexing (3)

Figure 6.4 *FDM process*



FDM Multiplexing

Figure 6.5 *FDM demultiplexing example*



FDM De-Multiplexing

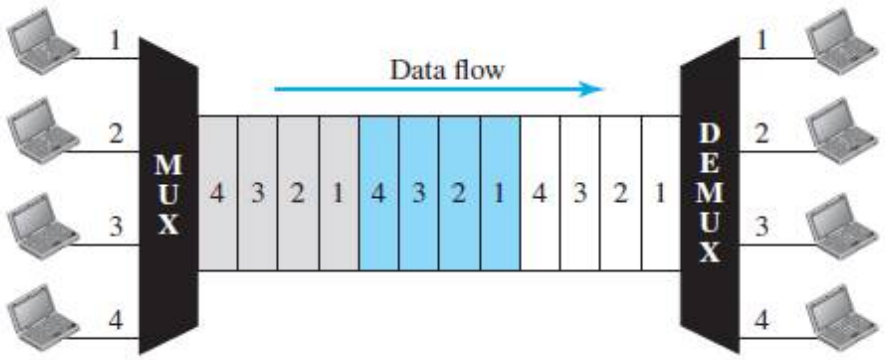
Frequency-division multiplexing (4)

- **Other Applications of FDM**
  - A very common application of FDM is **AM and FM radio broadcasting**.
  - Radio uses the air as the transmission medium.
  - The signal that goes to the air is a combination of signals.
  - A receiver receives all these signals, but filters (by tuning) only the one which is desired.
  - The situation is similar in **FM broadcasting**.
  - Another common use of FDM is in **television broadcasting.**

Time-Division Multiplexing (1)

- **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.
- Figure 6.12 gives a conceptual view of TDM.
- In the figure, portions of signals 1, 2, 3, and 4 occupy the link sequentially.
- This means that all the data in a message from source 1 always go to one specific destination, be it 1, 2, 3, or 4.
- TDM is a digital multiplexing technique i.e digital data from different sources are combined into one timeshared link.
- However, analog data can be sampled, changed to digital data, and then multiplexed by using TDM.

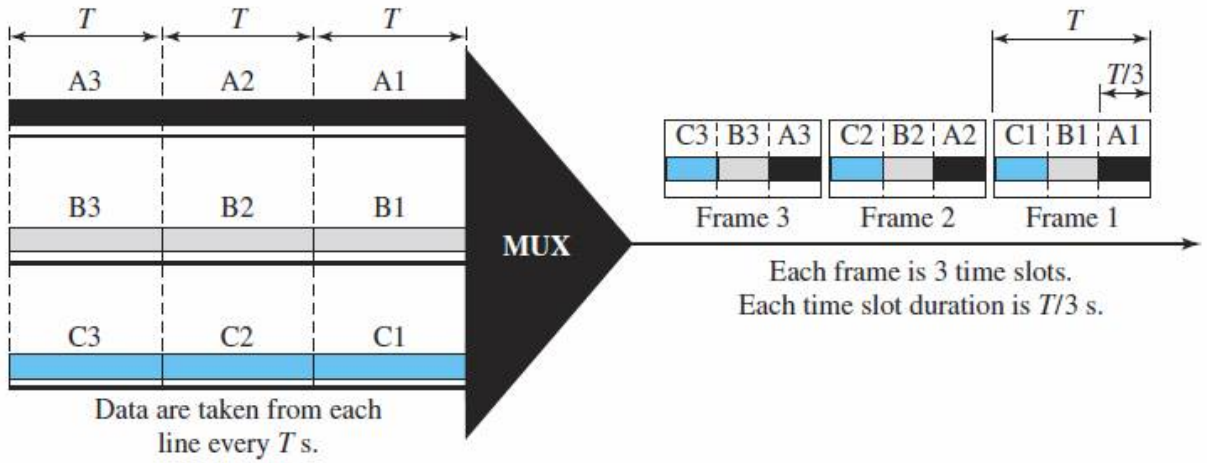
Figure 6.12 *TDM*



Time-Division Multiplexing (2)

- **Synchronous TDM**
- In synchronous TDM, each input connection has an allotment in the output even if it is not sending data.
- **Time Slots and Frames**
- In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot.
- A unit can be 1 bit, one character, or one block of data.
- Each input unit becomes one output unit and occupies one output time slot.
- However, the duration of an output time slot is n times shorter than the duration of an input time slot.
- If an input time slot is T s, the output time slot is T/n s, where n is the number of connections.
- Figure 6.13 shows an example of synchronous TDM where *n* is 3.

Figure 6.13 *Synchronous time-division multiplexing*



Time-Division Multiplexing (3)

- **Example 6.5**

  In Figure 6.13, the data rate for each input connection is 1 kbps. If 1 bit at a time is multiplexed (a unit is 1 bit), what is the duration of

  1. each input slot,
  2. each output slot, and
  3. each frame?

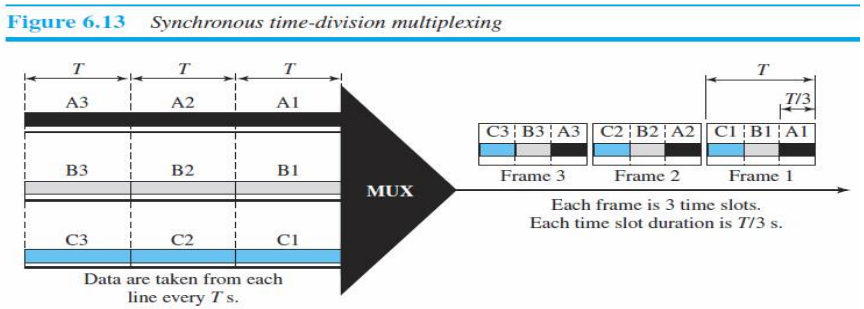


Figure 6.13 *Synchronous time-division multiplexing*

- **Solution**

  We can answer the questions as follows:

  1. The data rate of each input connection is 1 kbps. This means that the bit duration is 1/1000 s or 1 ms. The duration of the input time slot is 1 ms (same as bit duration).

  2. The duration of each output time slot is one-third of the input time slot. This means that the duration of the output time slot is 1/3 ms.

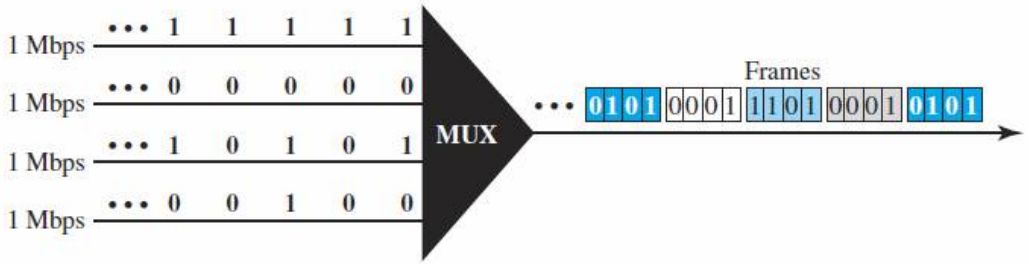
  3. Each frame carries three output time slots. So the duration of a frame is 3 × 1/3 ms, or 1 ms. The duration of a frame is the same as the duration of an input unit.

Time-Division Multiplexing (4)

- **Example 6.6**

  Figure 6.14 shows synchronous TDM with a data stream for each input and one data stream for the output. The unit of data is 1 bit. Find (1) the input bit duration, (2) the output bit duration, (3) the output bit rate, and (4) the output frame rate.

Figure 6.14 *Example 6.6*



- **Solution**

  We can answer the questions as follows:

  1. The input bit duration is the inverse of the bit rate: 1/1 Mbps = 1 μs.
  2. The output bit duration is one-fourth of the input bit duration, or 1/4μs.
  3. The output bit rate is the inverse of the output bit duration, or 1/4 μs, or 4 Mbps. This can also be deduced from the fact that the output rate is 4 times as fast as any input rate; so the output rate = 4 × 1 Mbps = 4 Mbps.
  4. The frame rate is always the same as any input rate. So the frame rate is 1,000,000 frames per second.

Time-Division Multiplexing (5)

- **Example 6.7**

  Four 1-kbps connections are multiplexed together. A unit is 1 bit. Find (1) the duration of 1 bit before multiplexing, (2) the transmission rate of the link, (3) the duration of a time slot, and (4) the duration of a frame.

- **Solution**

  We can answer the questions as follows:

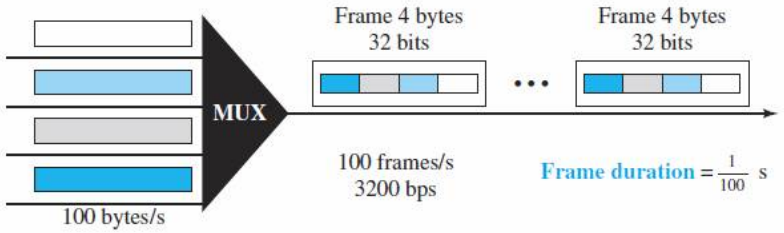
  1. The duration of 1 bit before multiplexing is 1/1 kbps, or 0.001 s (1 ms).
  2. The rate of the link is 4 times the rate of a connection, or 4 kbps.
  3. The duration of each time slot is one-fourth of the duration of each bit before multiplexing, or 1/4 ms or 250 μs.
  4. The duration of a frame is always the same as the duration of a unit before multiplexing, or 1 ms.

Time-Division Multiplexing (6)

- **Example 6.8**

  Four channels are multiplexed using TDM. If each channel sends 100 bytes/s and we multiplex 1 byte per channel, show the frame traveling on the link, the size of the frame, the duration of a frame, the frame rate, and the bit rate for the link.

Figure 6.16 *Example 6.8*

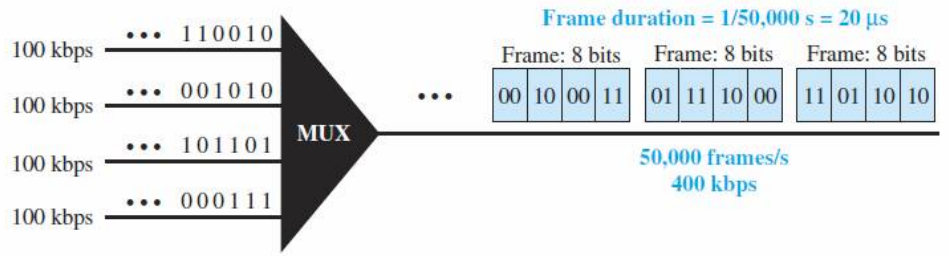


- **Solution**
  - The multiplexer is shown in Figure 6.16.
  - Each frame carries 1 byte from each channel; the size of each frame, therefore, is 4 bytes, or 32 bits.
  - Because each channel is sending 100 bytes/s and a frame carries 1 byte from each channel, the frame rate must be 100 frames per second.
  - The duration of a frame is therefore 1/100 s.
  - The link is carrying 100 frames per second, and since each frame contains 32 bits, the bit rate is 100 × 32, or 3200 bps.
  - This is actually 4 times the bit rate of each channel, which is 100 × 8 = 800 bps.

Time-Division Multiplexing (7)

- **Example 6.9**
- A multiplexer combines four 100-kbps channels using a time slot of 2 bits. Show the output with four arbitrary inputs. What is the frame rate? What is the frame duration? What is the bit rate? What is the bit duration?

Figure 6.17 *Example 6.9*

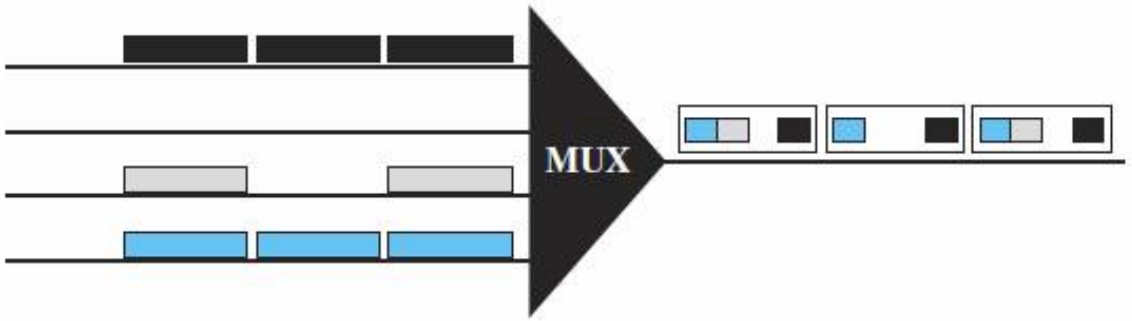


- **Solution**
  - Figure 6.17 shows the output for four arbitrary inputs.
  - The link carries 50,000 frames per second since each frame contains 2 bits per channel. The frame duration is therefore 1/50,000 s or 20 µs.
  - The frame rate is 50,000 frames per second, and each frame carries 8 bits; the bit rate is 50,000 × 8 = 400,000 bits or 400 kbps. The bit duration is 1/400,000 s, or 2.5 µs.
  - Note that the frame duration is 8 times the bit duration because each frame is carrying 8 bits.

Time-Division Multiplexing (8)

- **Empty Slots**
- Synchronous TDM is not as efficient as it could be.
- If a source does not have data to send, the corresponding slot in the output frame is empty.
- Figure 6.18 shows a case in which one of the input lines has no data to send and one slot in another input line has discontinuous data.
- The first output frame has three slots filled, the second frame has two slots filled, and the third frame has three slots filled. No frame is full.

Figure 6.18 *Empty slots*



Time-Division Multiplexing (9)

- **Data Rate Management**
- One problem with TDM is how to handle a disparity in the input data rates.

- In all our discussion so far, we assumed that the data rates of all input lines were the same.

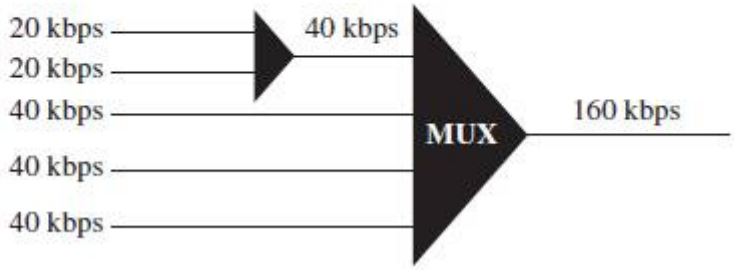
- However, if data rates are not the same, three strategies, or a combination of them, can be used.

- We call these three strategies multilevel multiplexing, multiple-slot allocation, and pulse stuffing.

Time-Division Multiplexing (10)

- Multilevel Multiplexing
  - Multilevel multiplexing is a technique used when the data rate of an input line is a multiple of others.
  - For example, in Figure 6.19, we have two inputs of 20 kbps and three inputs of 40 kbps. The first two input lines can be multiplexed together to provide a data rate equal to the last three.
  - A second level of multiplexing can create an output of 160 kbps.

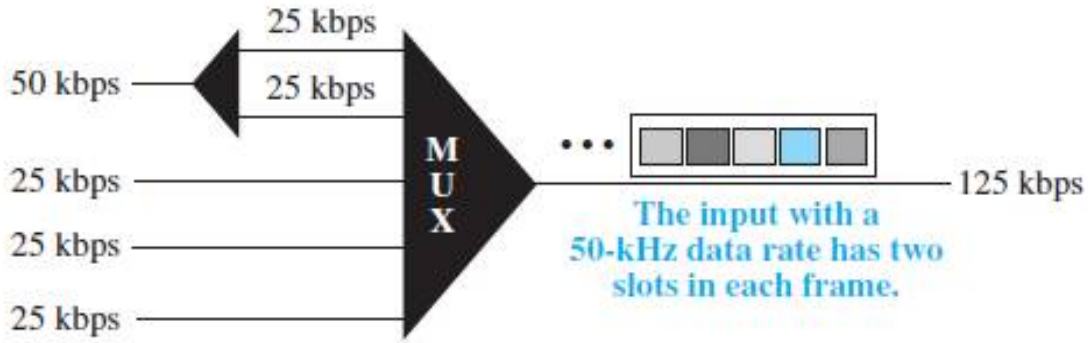
Figure 6.19 *Multilevel multiplexing*



Time-Division Multiplexing (11)

- Multiple-Slot Allocation
  - Sometimes it is more efficient to allot more than one slot in a frame to a single input line.
  - For example, we might have an input line that has a data rate that is a multiple of another input.
  - In Figure 6.20, the input line with a 50-kbps data rate can be given two slots in the output.
  - We insert a demultiplexer in the line to make two inputs out of one.

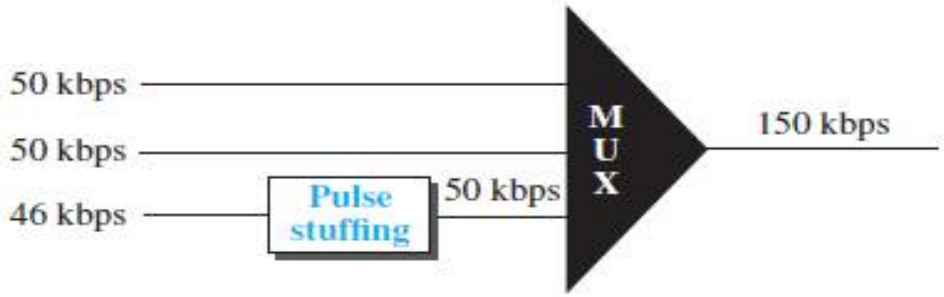
Figure 6.20 *Multiple-slot multiplexing*



Time-Division Multiplexing (12)

- Pulse Stuffing
  - Sometimes the bit rates of sources are not multiple integers of each other.
  - Therefore, neither of the above two techniques can be applied.
  - One solution is to make the highest input data rate the dominant data rate and then add dummy bits to the input lines with lower rates.
  - This will increase their rates.
  - This technique is called pulse stuffing, bit padding, or bit stuffing.
  - The idea is shown in Figure 6.21.
  - The input with a data rate of 46 is pulse-stuffed to increase the rate to 50 kbps. Now multiplexing can take place.

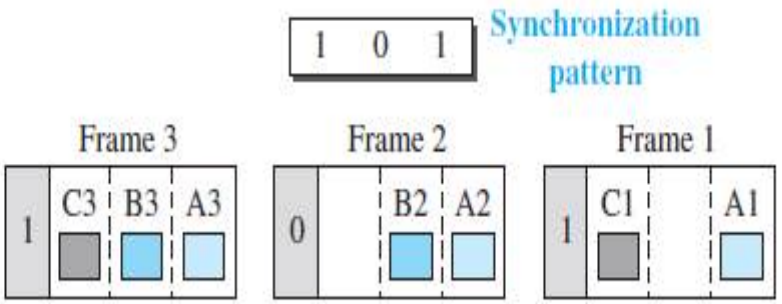
Figure 6.21 *Pulse stuffing*



Time-Division Multiplexing (13)

- Frame Synchronizing
  - Synchronization between the multiplexer and demultiplexer is a major issue.
  - If the multiplexer and the demultiplexer are not synchronized, a bit belonging to one channel may be received by the wrong channel.
  - For this reason, one or more synchronization bits are usually added to the beginning of each frame.
  - These bits, called framing bits, follow a pattern, frame to frame, that allows the demultiplexer to synchronize with the incoming stream so that it can separate the time slots accurately.
  - In most cases, this synchronization information consists of 1 bit per frame, alternating between 0 and 1, as shown in Figure 6.22.

Figure 6.22 *Framing bits*



Time-Division Multiplexing (14)

- **Example 6.10**

  We have four sources, each creating 250 characters per second. If the interleaved unit is a character and 1 synchronizing bit is added to each frame, find (1) the data rate of each source, (2) the duration of each character in each source, (3) the frame rate, (4) the duration of each frame, (5) the number of bits in each frame, and (6) the data rate of the link.

- **Solution**
  - 1. The data rate of each source is 250 × 8 = 2000 bps = 2 kbps.
  - 2. Each source sends 250 characters per second; therefore, the duration of a character is 1/250 s, or 4 ms.
  - 3. Each frame has one character from each source, which means the link needs to send 250 frames per second to keep the transmission rate of each source.
  - 4. The duration of each frame is 1/250 s, or 4 ms. Note that the duration of each frame is the same as the duration of each character coming from each source.
  - 5. Each frame carries 4 characters and 1 extra synchronizing bit. This means that each frame is 4 × 8 + 1 = 33 bits.
  - 6. The link sends 250 frames per second, and each frame contains 33 bits. This means that the data rate of the link is 250 × 33, or 8250 bps.
  - Note that the bit rate of the link is greater than the combined bit rates of the four channels.
  - If we add the bit rates of four channels, we get 8000 bps. Because 250 frames are traveling per second and each contains 1 extra bit for synchronizing, we need to add 250 to the sum to get 8250 bps.

Time-Division Multiplexing (15)

- **Example 6.11**
- Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be multiplexed. How this can be achieved? What is the frame rate? What is the frame duration? What is the bit rate of the link?

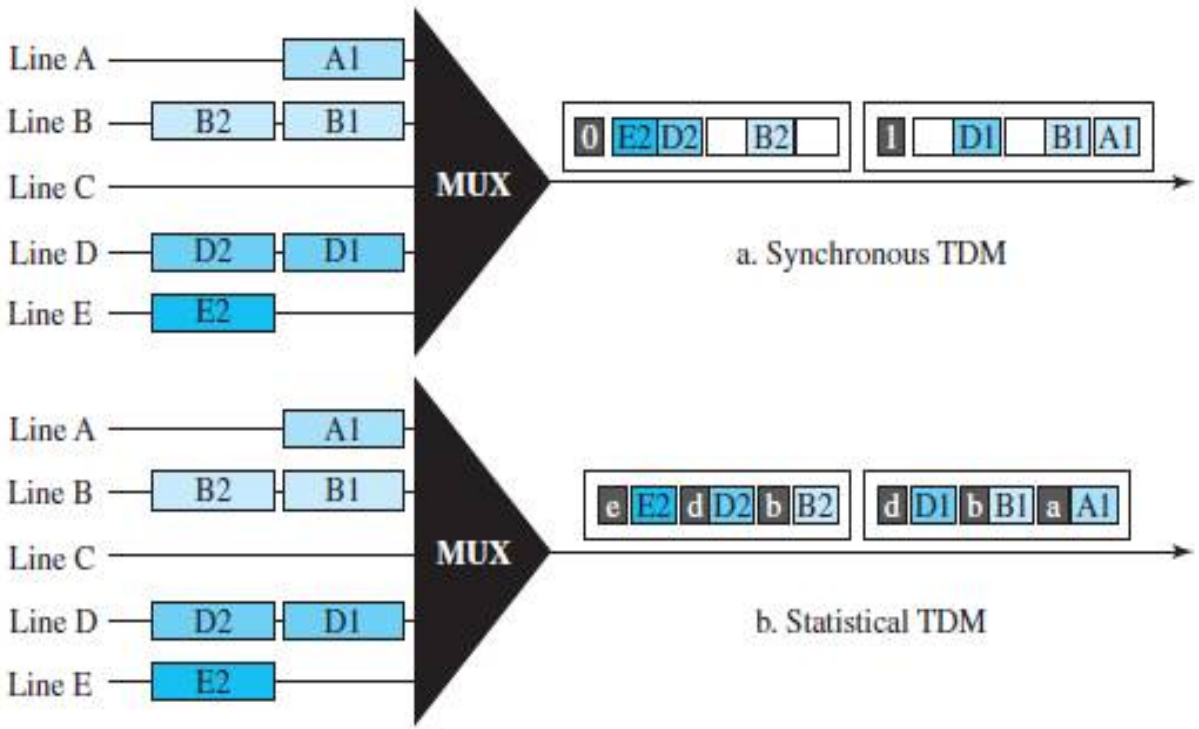
- **Solution**
- We can allocate one slot to the first channel and two slots to the second channel.
- Each frame carries 3 bits.
- The frame rate is 100,000 frames per second because it carries 1 bit from the first channel.
- The frame duration is 1/100,000 s, or 10 ms.
- The bit rate is 100,000 frames/s × 3 bits per frame, or 300 kbps.
- Note that because each frame carries 1 bit from the first channel, the bit rate for the first channel is preserved.
- The bit rate for the second channel is also preserved because each frame carries 2 bits from the second channel

Time-Division Multiplexing (16)

- **Statistical Time-Division Multiplexing**
- In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency.
- Only when an input line has a slot's worth of data to send is it given a slot in the output frame.
- In statistical multiplexing, the number of slots in each frame is less than the number of input lines.
- The multiplexer checks each input line in round robin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.
- Figure 6.26 shows a synchronous and a statistical TDM example.
- In the former, some slots are empty because the corresponding line does not have data to send.
- In the latter, however, no slot is left empty as long as there are data to be sent by any input line

Time-Division Multiplexing (17)

Figure 6.26 *TDM slot comparison*



Time-Division Multiplexing (18)

- **Addressing**
- Figure 6.26 also shows a major difference between slots in synchronous TDM and statistical TDM.
- An output slot in synchronous TDM is totally occupied by data; in statistical TDM, a slot needs to carry data as well as the address of the destination.
- In synchronous TDM, there is no need for addressing; synchronization and preassigned relationships between the inputs and outputs serve as an address.
- In statistical multiplexing, there is no fixed relationship between the inputs and outputs because there are no preassigned or reserved slots.
- We need to include the address of the receiver inside each slot to show where it is to be delivered.
- The addressing in its simplest form can be n bits to define N different output lines with n = log2 N. For example, for eight different output lines, we need a 3-bit address.

Time-Division Multiplexing (19)

- **Slot Size**
  - Since a slot carries both data and an address in statistical TDM, the ratio of the data size to address size must be reasonable to make transmission efficient.
  - For example, it would be inefficient to send 1 bit per slot as data when the address is 3 bits.
  - This would mean an overhead of 300 percent.
  - In statistical TDM, a block of data is usually many bytes while the address is just a few bytes.
- **No Synchronization Bit**
  - The frames in statistical TDM need not be synchronized, so we do not need synchronization bits.
- **Bandwidth**
  - In statistical TDM, the capacity of the link is normally less than the sum of the capacities of each channel.

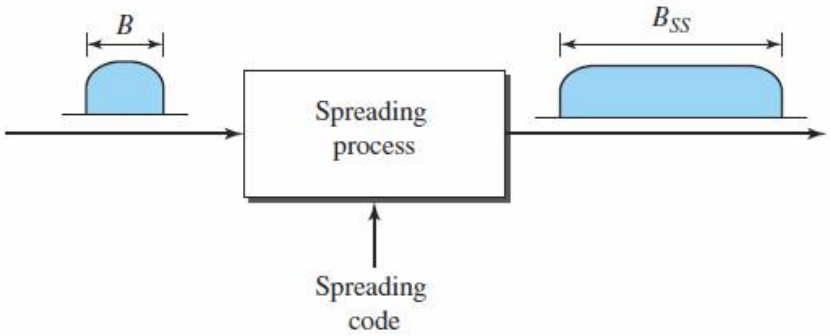
Spread Spectrum (1)

- Spread spectrum is designed to be used in wireless applications (LANs and WANs).
- In wireless applications, all stations use air (or a vacuum) as the medium for communication.
- Stations must be able to share this medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder (in military operations, for example).
- To achieve these goals, spread spectrum techniques add redundancy; they spread the original spectrum needed for each station.
- If the required bandwidth for each station is B, spread spectrum expands it to Bss, such that Bss >> B.
- The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission.

Spread Spectrum (2)

- Figure 6.27 shows the idea of spread spectrum. Spread spectrum achieves its goals through two principles:
  - 1. The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.
  - 2. The expanding of the original bandwidth B to the bandwidth Bss must be done by a process that is independent of the original signal.
- After the signal is created by the source, the spreading process uses a spreading code and spreads the bandwidth.
- There are two techniques to spread the bandwidth: frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).

Figure 6.27 *Spread spectrum*

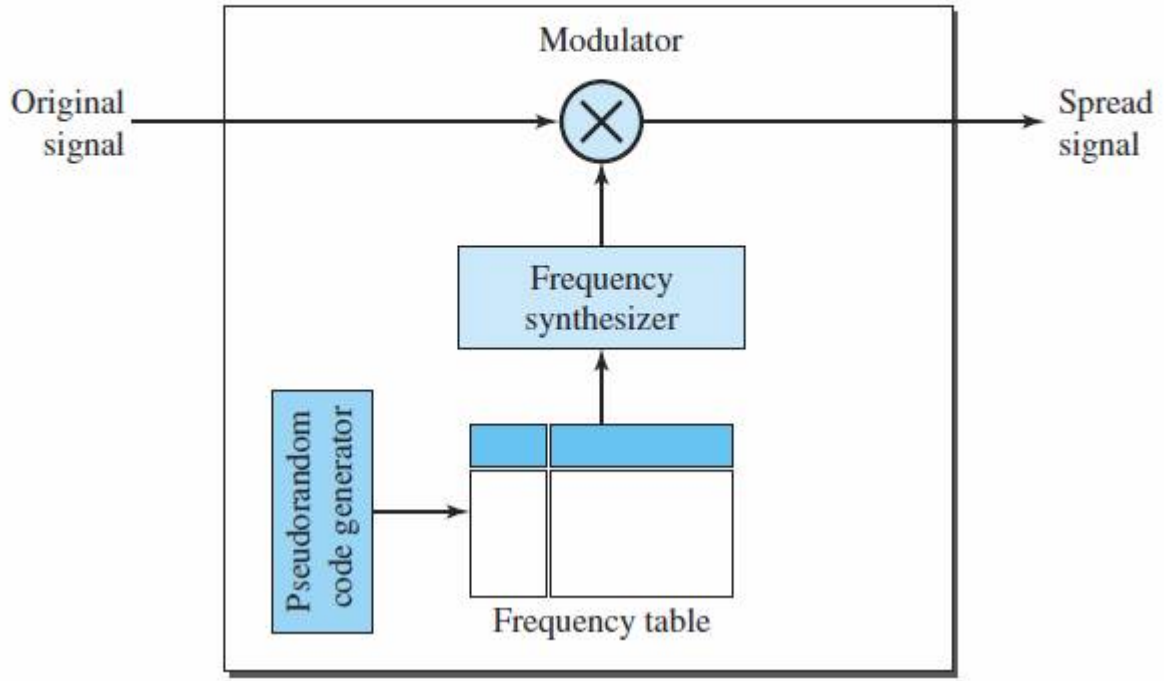


Frequency Hopping Spread Spectrum (1)

- The frequency hopping spread spectrum (FHSS) technique uses M different carrier frequencies that are modulated by the source signal.
- At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency.
- Although the modulation is done using one carrier frequency at a time, M frequencies are used in the long run. The bandwidth occupied by a source after spreading is BFHSS >> B.
- Figure 6.28 shows the general layout for FHSS. A pseudorandom code generator, called pseudorandom noise (PN), creates a k-bit pattern for every hopping period Th.
- The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer.
- The frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal.

Frequency Hopping Spread Spectrum (2)

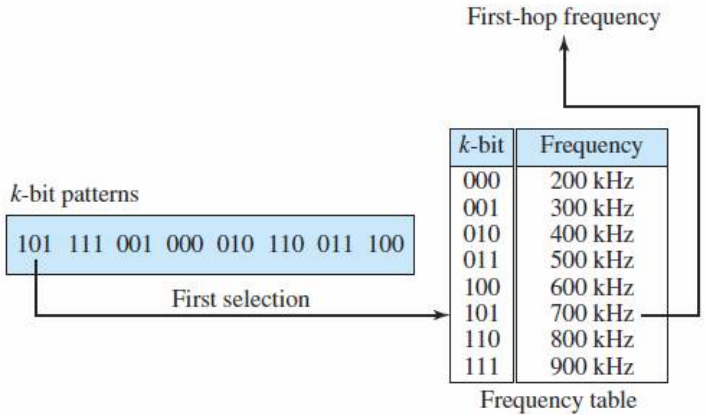
Figure 6.28 *Frequency hopping spread spectrum (FHSS)*



Frequency Hopping Spread Spectrum (3)

- Suppose we have decided to have eight hopping frequencies.
- This is extremely low for real applications and is just for illustration.
- In this case, *M is 8 and k is 3.*
- *The pseudorandom* code generator will create eight different 3-bit patterns.
- These are mapped to eight different frequencies in the frequency table .
- The pattern for this station is 101, 111, 001, 000, 010, 011, 100.
- Note that the pattern is pseudorandom; it is repeated after eight hoppings.
- This means that at hopping period 1, the pattern is 101.
- The frequency selected is 700 kHz; the source signal modulates this carrier frequency.
- The second k-bit pattern selected is 111, which selects the 900-kHz carrier; the eighth pattern is 100, and the frequency is 600 kHz. After eight hoppings, the pattern repeats, starting from 101 again. Figure 6.30 shows how the signal

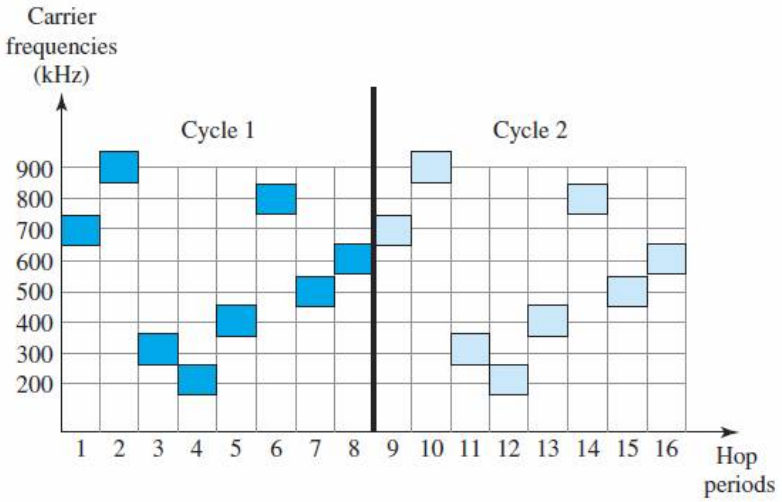
Figure 6.29 *Frequency selection in FHSS*



Frequency Hopping Spread Spectrum (4)

- If there are many k-bit patterns and the hopping period is short, a sender and receiver can have privacy.
- If an intruder tries to intercept the transmitted signal, she can only access a small piece of data because she does not know the spreading sequence to quickly adapt herself to the next hop.
- The scheme also has an antijamming effect.
- A malicious sender may be able to send noise to jam the signal for one hopping period (randomly), but not for the whole period.

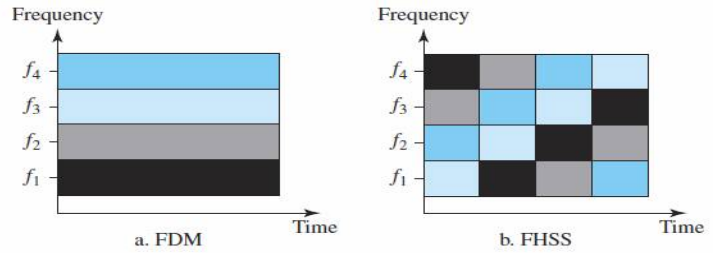
Figure 6.30 *FHSS cycles*



Frequency Hopping Spread Spectrum (5)

- **Bandwidth Sharing**
  - If the number of hopping frequencies is M, we can multiplex M channels into one by using the same Bss bandwidth.
  - This is possible because a station uses just one frequency in each hopping period; M – 1 other frequencies can be used by M – 1 other stations.
  - In other words, M different stations can use the same Bss if an appropriate modulationtechnique such as multiple FSK (MFSK) is used.
  - FHSS is similar to FDM.
  - Figure 6.31 shows an example of four channels using FDM and four channels using FHSS.
  - In FDM, each station uses 1/M of the bandwidth, but the allocation is fixed; in FHSS, each station uses 1/M of the bandwidth, but the allocation changes hop to hop.

Figure 6.31 *Bandwidth sharing*



Direct Sequence Spread Spectrum (1)

- In DSSS, we replace each data bit with n bits using a spreading code.
- In other words, each bit is assigned a code of n bits, called chips, where the chip rate is n times that of the data bit.
- Figure 6.32 shows the concept of DSSS.
- As an example, let us consider the sequence used in a wireless LAN, the famous Barker sequence, where n is 11.
- We assume that the original signal and the chips in the chip generator use polar NRZ encoding.
- Figure 6.33 shows the chips and the result of multiplying the original data by the chips to get the spread signal.
- In Figure 6.33, the spreading code is 11 chips having the pattern 10110111000 (in this case).
- If the original signal rate is N, the rate of the spread signal is 11N.
- This means that the required bandwidth for the spread signal is 11 times larger than the bandwidth of the original signal.
- The spread signal can provide privacy if the intruder does not know the code.
- It can also provide immunity against interference if each station uses a different code.

Direct Sequence Spread Spectrum (2)

Figure 6.32 *DSSS*

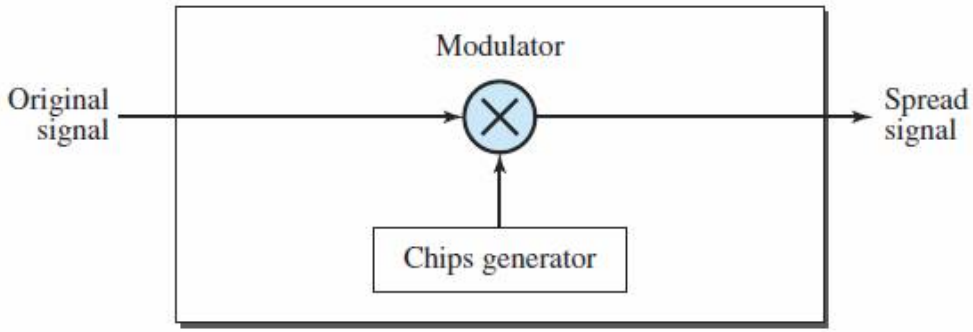
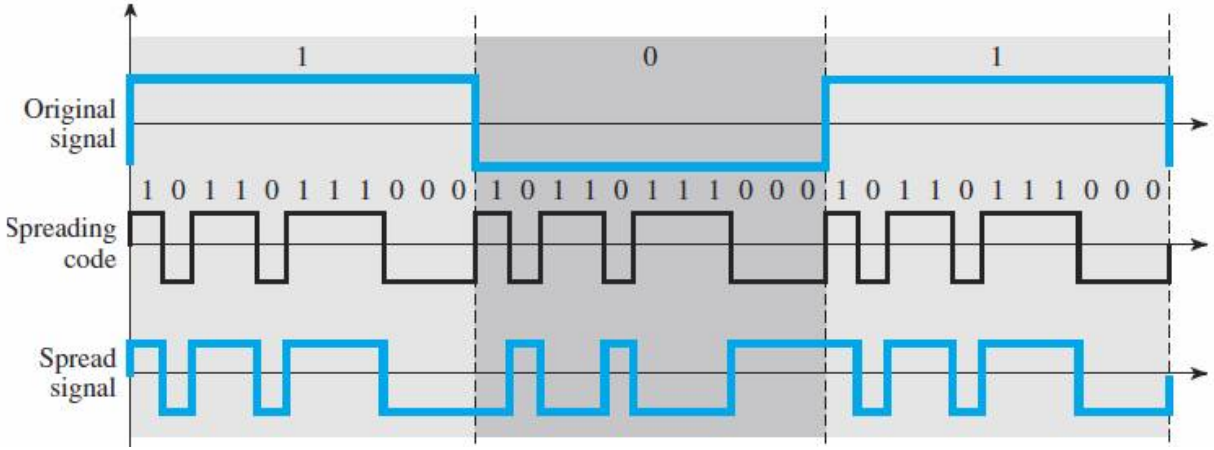


Figure 6.33 *DSSS example*



END OF CHAPTER 6

Chapter 8
Switching

Reference:

Data Communication and Networking, Behrouz A.Forouzan, McGraw Hill, 5th Edition, 2008

Note to Students : ppt is for revision purpose only. Answers in internals and exams should be written elaborately as given in the prescribed text book

Introduction (1)

- 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.
- Figure 8.1 shows a switched network.
- The end systems (communicating devices) are labeled A, B, C, D, and so on, and the switches are labeled I, II, III, IV, and V.
- Each switch is connected to multiple links.

Figure 8.1 *Switched network*

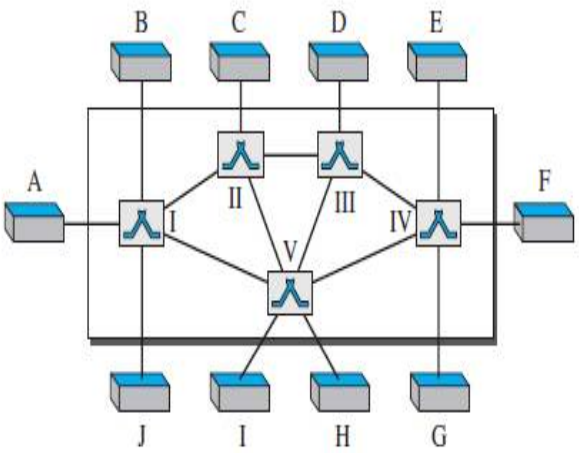
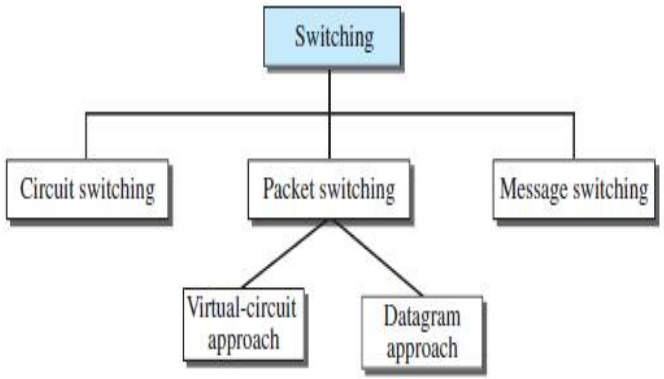


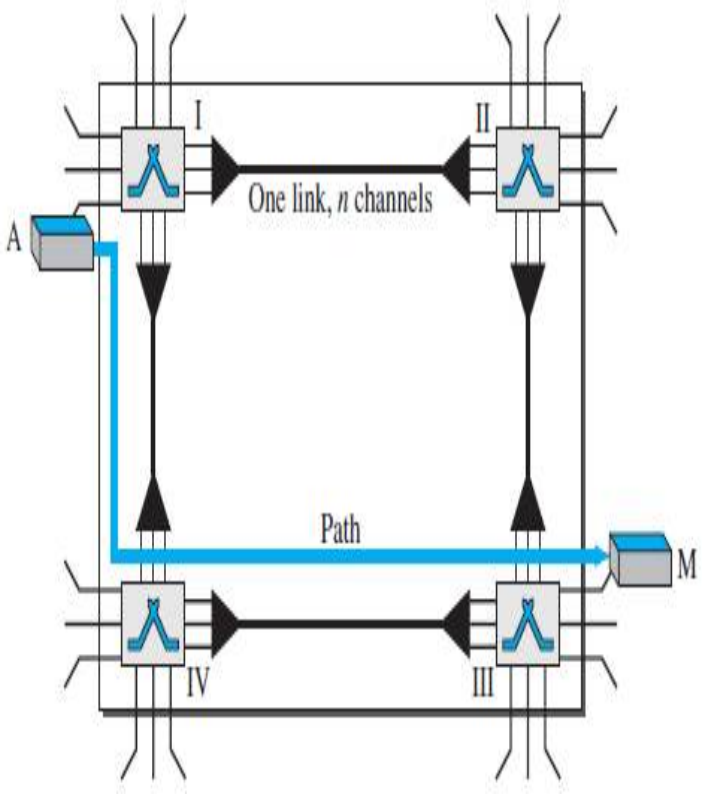
Figure 8.2 *Taxonomy of switched networks*



Circuit-switched network (1)

- 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.
- Each link is normally divided into *n channels by using FDM or TDM*
- 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*



Circuit-switched network (2)

- The end systems, such as computers or telephones, are directly connected to a switch.
- 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, the **data-transfer phase can take place.**
- **After all data have been transferred,** the circuits are torn down.

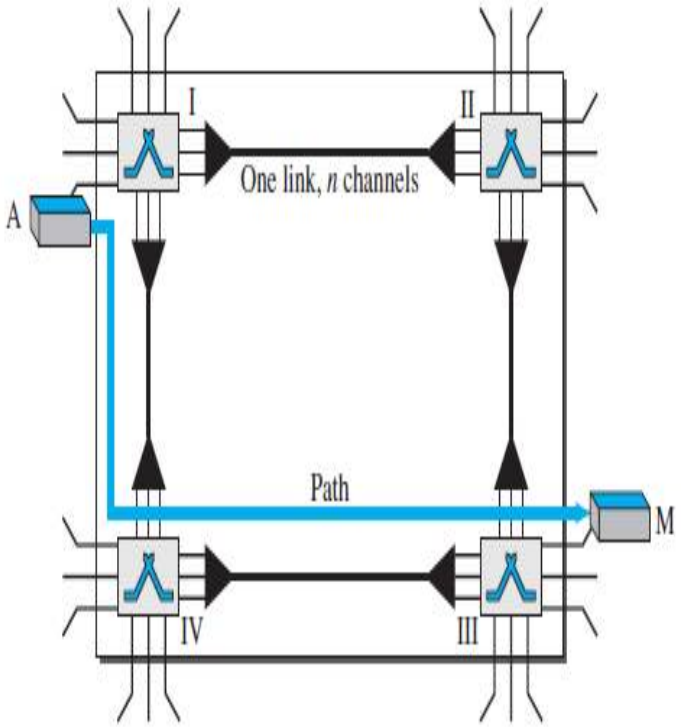
Circuit-switched network (3)

- Some important points :
  - 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.**
  - 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.
  - There is no addressing involved during data transfer. The switches route the data based on their occupied band (FDM) or time slot (TDM). Of course, there is end-to-end addressing used during the setup phase.

Circuit-switched network (4)

- **Three Phases :** setup, data transfer, 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.

Figure 8.3 *A trivial circuit-switched network*



Circuit-switched network (5)

- 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.

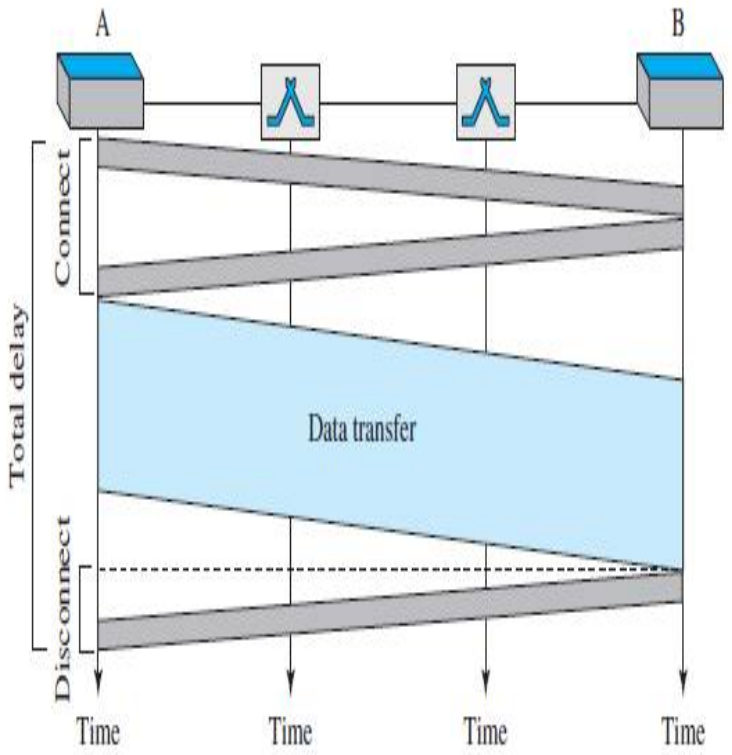
Circuit-switched network (6)

- **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.
- **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.
  - Figure 8.6 shows the idea of delay in a circuit-switched network when only two switches are involved.

Circuit-switched network (7)

- There is no waiting time at each switch.
- The total delay is due to the time needed to create the connection, transfer data, and disconnect the circuit.
- The delay caused by the setup is the sum of four parts: the propagation time of the source computer request, the request signal transfer time, the propagation time of the acknowledgment from the destination computer, 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.
- In this case the receiver requests disconnection, which creates the maximum delay.

Figure 8.6 *Delay in a circuit-switched network*



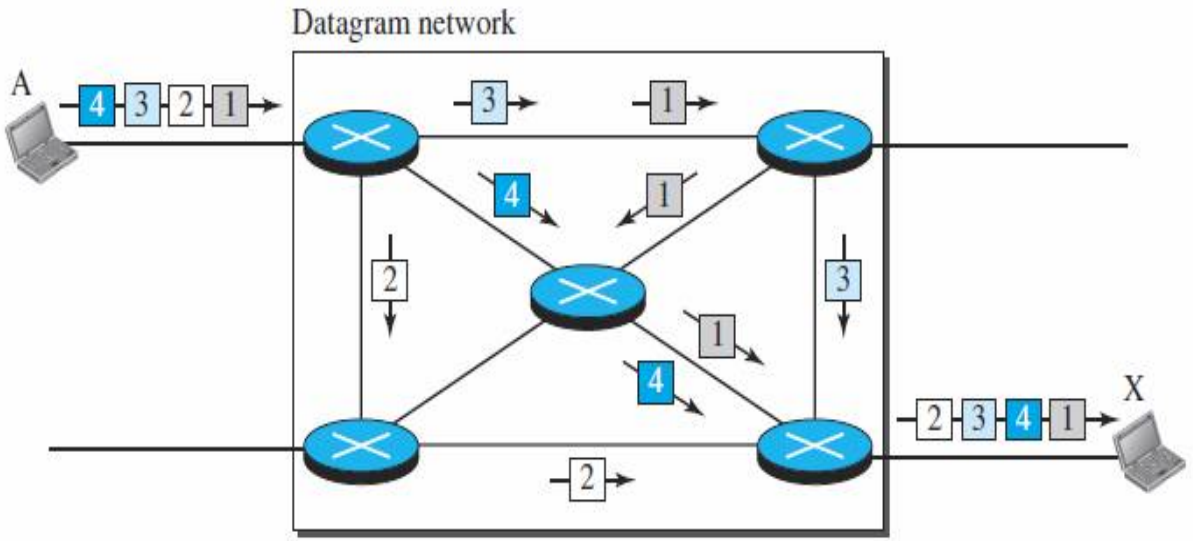
Packet Switching

- 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 firstcome, first-served basis.
- When a switch receives a packet, no matter what the source or destination is, the packet must wait if there are other packets being processed.
- 2 types of packet switched networks – datagram networks and virtual circuits.

Datagram Networks (1)

- In a datagram network, each packet is treated independently of all others.
- Even if a packet is part of a multi-packet 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.
- Figure 8.7 shows how the datagram approach is used to deliver four packets from station A to station X.

Figure 8.7 *A datagram network with four switches (routers)*



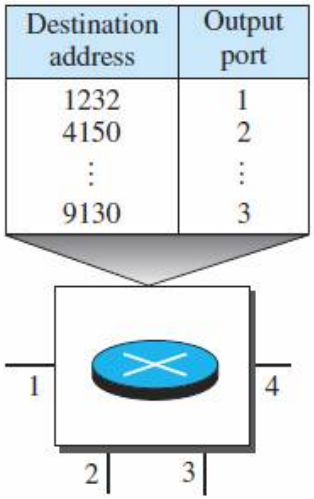
Datagram Networks (2)

- The switches in a datagram network are traditionally referred to as routers.
- In this example, all four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination.
- 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.

Datagram Networks (3)

- Routing Table
- 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.
- Figure 8.8 shows the routing table for a switch.

Figure 8.8 *Routing table in a datagram network*



Datagram Networks (4)

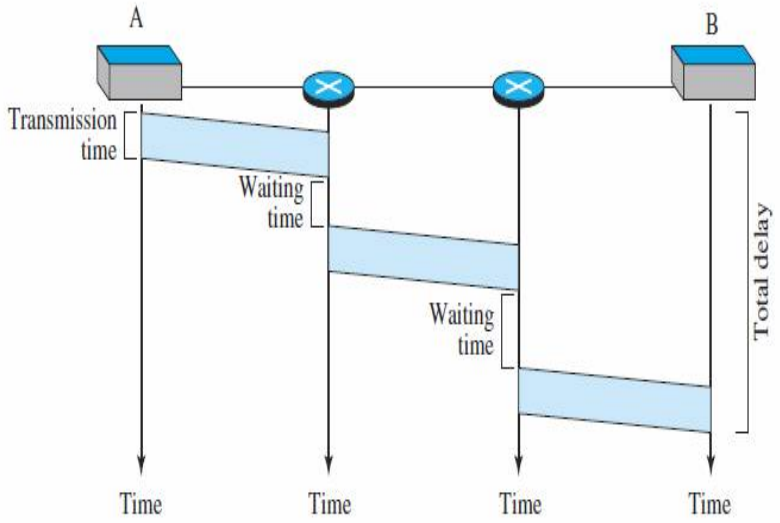
- Destination Address
  - Every packet in a datagram network carries a header that contains, among other information, the destination address of the packet.
  - When the switch receives the packet, this destination address is examined; the routing table is consulted to find the corresponding port through which the packet should be forwarded.
  - This address, 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.

Datagram Networks (5)

- 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.
  - Figure 8.9 gives an example of delay in a datagram network for one packet.
  - The packet travels through two switches.
  - There are three transmission times (3T), three propagation delays (slopes 3τ of the lines), and two waiting times (w1 + w2).
  - We ignore the processing time in each switch. The total delay is

$$\text{Total delay} = 3T + 3\tau + w_1 + w_2$$

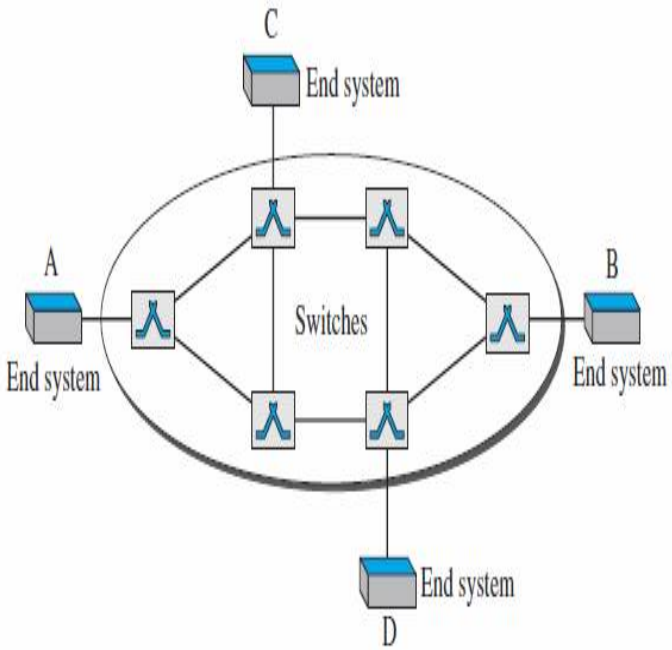
Figure 8.9 *Delay in a datagram network*



Virtual Circuit Networks (1)

- 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.
  - 2. 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. However, the address in the header has local jurisdiction, not end-to-end jurisdiction.
  - 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.
- Figure 8.10 is an example of a virtual-circuit network.

Figure 8.10 *Virtual-circuit network*



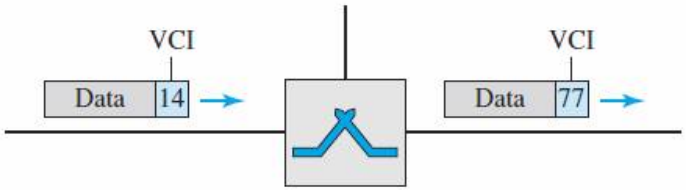
Virtual Circuit Networks (2)

- **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.
  - However, we will see that a global address in virtual-circuit networks is used only to create a virtual-circuit identifier.

- **Virtual-Circuit Identifier**
  - The identifier that is actually used for data transfer is called the virtual-circuit identifier (VCI) or the label.
  - 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*



Virtual Circuit Networks (3)

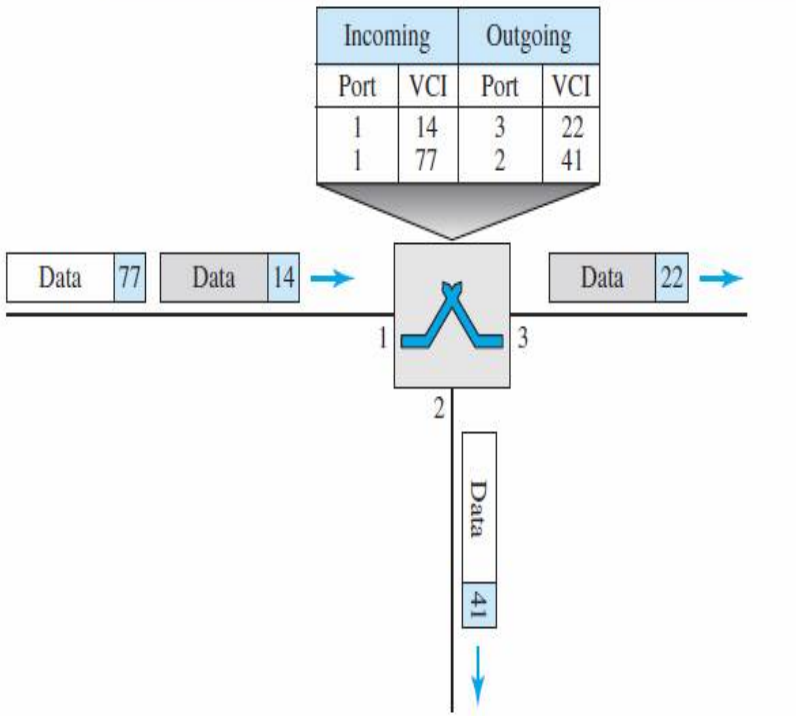
- ***Three Phases***
  - Three phases in a virtual-circuit network: setup, data transfer, and teardown.
  - In the setup phase, the source and destination use their global addresses to help switches make table entries for the connection.
  - In the teardown phase, the source and destination inform the switches to delete the corresponding entry.
  - Data transfer occurs between these two phases.

- ***Data-Transfer Phase***
  - To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit.
  - The table has four columns.
  - This means that the switch holds four pieces of information for each virtual circuit that is already set up.
  - Each switch has a table with entries for all active virtual circuits.
  - Figure 8.12 shows such a switch and its corresponding table.

Virtual Circuit Networks (4)

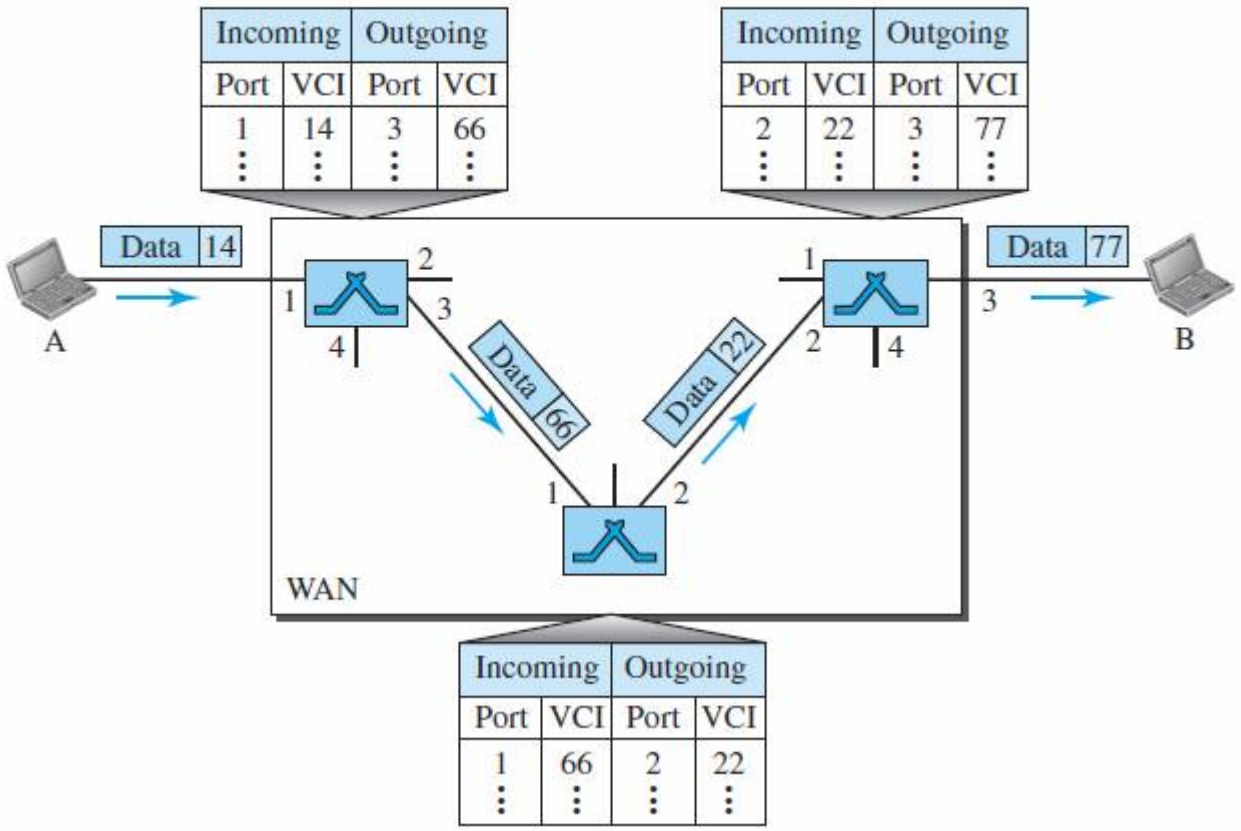
- Figure 8.12 shows a frame arriving at port 1 with a VCI of 14.
- When the frame arrives, the switch looks in its table to find port 1 and a VCI of 14.
- When it is found, the switch knows to change the VCI to 22 and send out the frame from port 3.
- Figure 8.13 shows how a frame from source A reaches destination B and how its VCI changes during the trip.
- Each switch changes the VCI and routes the frame.
- The data-transfer phase is active until the source sends all its frames to the destination.

Figure 8.12 *Switch and tables in a virtual-circuit network*



Virtual Circuit Networks (5)

Figure 8.13 *Source-to-destination data transfer in a virtual-circuit network*



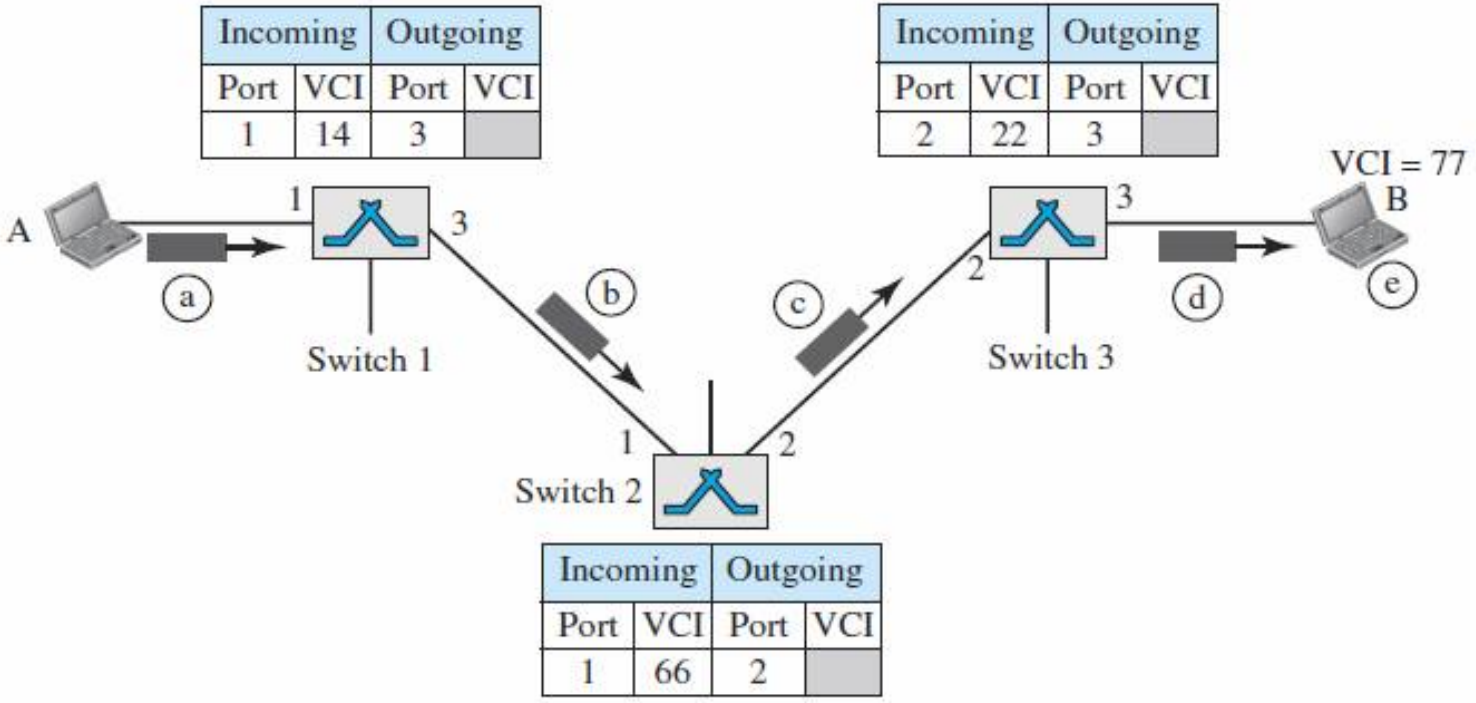
Virtual Circuit Networks (6)

- ***Setup Phase***
  - In the setup phase, a switch creates an entry for a virtual circuit.
  - For example, suppose source A needs to create a virtual circuit to B.
  - Two steps are required: the setup request and the acknowledgment.

- ***Setup Phase***
- A setup request frame is sent from the source to the destination. Figure 8.14 shows the process.
  - Source A sends a setup frame to switch 1.
  - Switch 1 receives the setup request frame. It knows that a frame going from A to B goes out through port 3. The switch, in the setup phase, acts as a packet switch; it has a routing table which is different from the switching table. The switch creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The switch assigns the incoming port (1) and chooses an available incoming VCI (14) and the outgoing port (3). It does not yet know the outgoing VCI, which will be found during the acknowledgment step. The switch then forwards the frame through port 3 to switch 2.
  - Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (1), incoming VCI (66), and outgoing port (2).
  - Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).
  - Destination B receives the setup frame, and if it is ready to receive frames from A, it assigns a VCI to the incoming frames that come from A, in this case 77. This VCI lets the destination know that the frames come from A, and not other sources.

Virtual Circuit Networks (7)

Figure 8.14 *Setup request in a virtual-circuit network*



Virtual Circuit Networks (8)

- ***Acknowledgment***
- A special frame, called the acknowledgment frame, completes the entries in the switching tables. Figure 8.15 shows the process

  - The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.

  - Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.

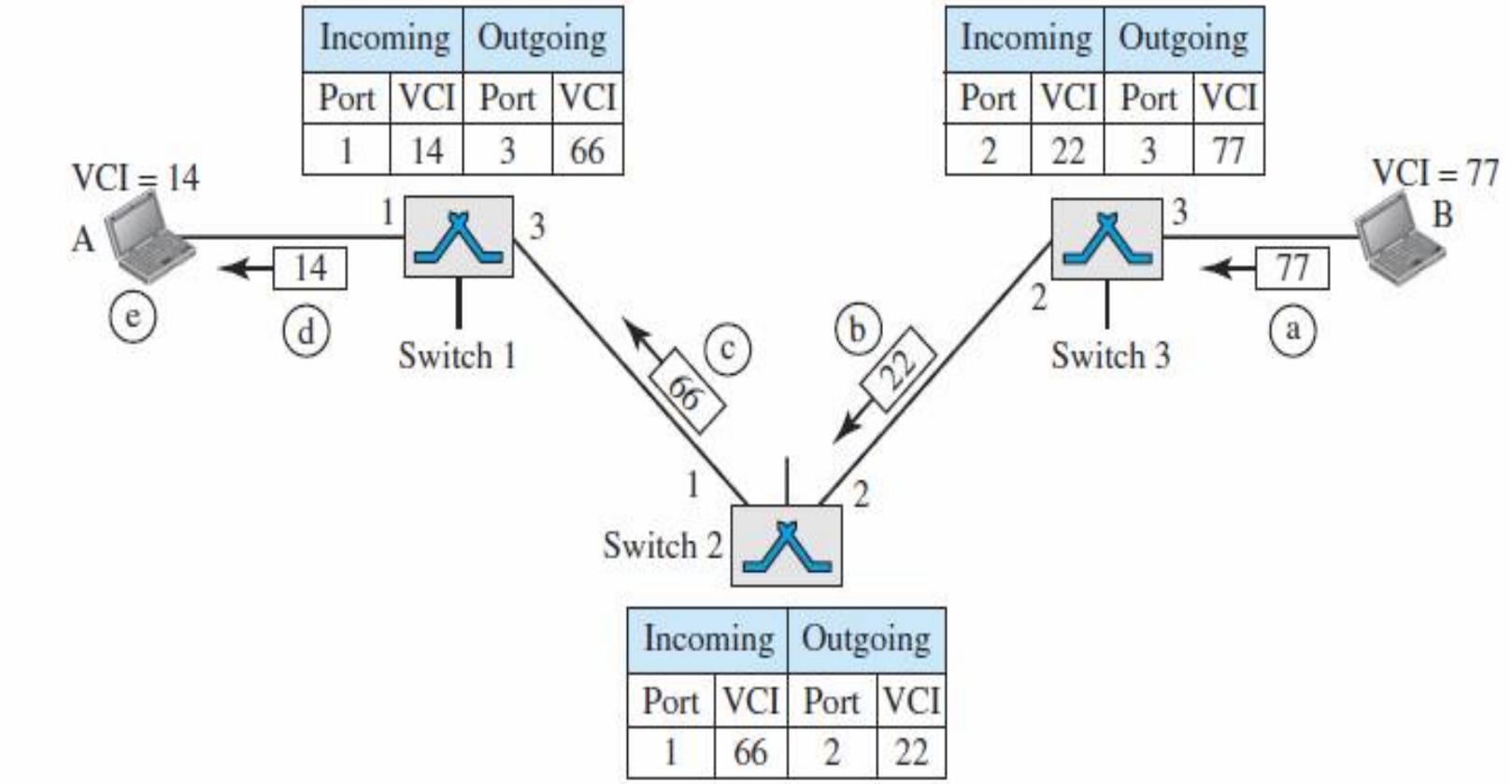
  - Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.

  - Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.

  - The source uses this as the outgoing VCI for the data frames to be sent to destination Bs. Figure 8.15 shows the process.

Virtual Circuit Networks (9)

Figure 8.15 *Setup acknowledgment in a virtual-circuit network*



Virtual Circuit Networks (10)

- ***Teardown Phase***
  - In this phase, source A, after sending all frames to B, sends a special frame called a teardown request.
  - Destination B responds with a teardown confirmation frame.
  - All switches delete the corresponding entry from their tables.

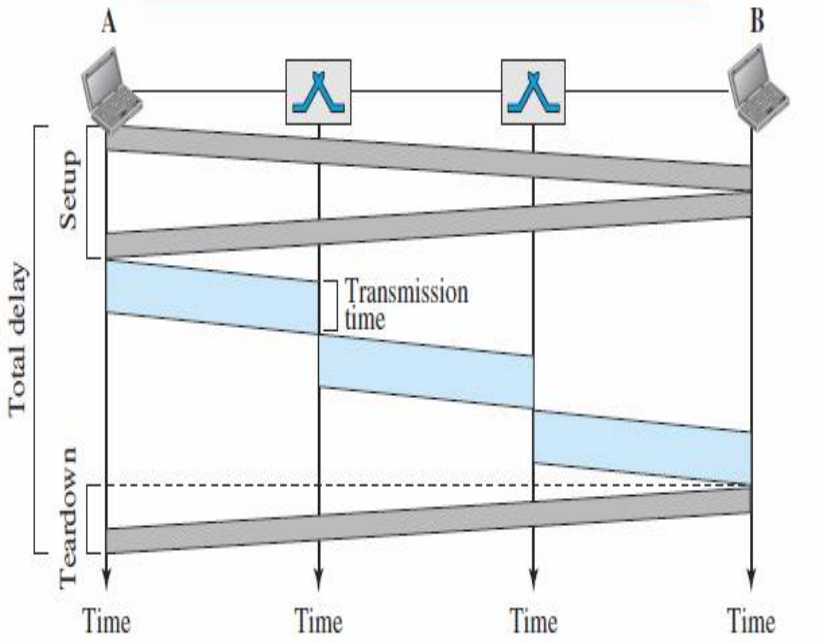
- ***Efficiency***
  - Resource reservation in a virtual-circuit network can be made during the setup or can be on demand during the data-transfer phase.
  - In the first case, the delay for each packet is the same; in the second case, each packet may encounter different delays.
  - There is one big 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.

Virtual Circuit Networks (11)

- **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.
- There are three transmission times (3T), three propagation times (3τ), 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).
- The total delay time is

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

Figure 8.16 *Delay in a virtual-circuit network*



END OF CHAPTER 8