Objectives

Four major concepts are discussed: data communications, networking, protocols and standards, and networking models.

1. Networks exist so that data may be sent from one place to another-the basic concept of *data communications.*
2. Data communications between remote parties can be achieved through a process called *networking,* Involving
   the connection of computers,
   media, and
   networking devices.
3. Networks are divided into two main categories: local area networks (LANs) and wide area networks (WANs). These two types of networks have different characteristics and different functionalities. The Internet, is a collection of LANs and WANs held together by internetworking devices.
4. *Protocols and standards* are vital to the implementation of data communications and networking.
5. Protocols refer to the rules;
6. a standard is a protocol that has been adopted by vendors and manufacturers.
7. *Network models* serve to organize, unify, and control the hardware and software components of data communications and networking.

Introduction

1. Businesses today rely on computer networks and internetworks.
2. we need to know how networks operate, what types of technologies are available, and which design best fills which set of needs.
3. The development of the personal computer brought about tremendous changes for business, industry, science, and education.
4. A similar revolution is occurring in data communications and networking. Technological advances are making it possible for communications links to carry more and faster signals. As a result, services are evolving to allow use of this expanded capacity.
5. For example, established telephone services such as conference calling, call waiting, voice mail, and caller **ID** have been extended.
6. One goal is to be able to exchange data such as text, audio, and video from all points in the world. We want to access the Internet to download and upload information quickly and accurately and at any time.

This chapter addresses four issues: data communications, networks, the Internet, and protocols and standards.

.

1.1 DATA COMMUNICATIONS

1 When we communicate, we are sharing information. This sharing can be local or remote.

2. Between individuals, local communication usually occurs face to face, while remote communication takes place over distance.

3. The term *telecommunication,* which includes telephony, telegraphy, and television, means communication at a distance *(tele* is Greek for "far").

4. The word *data* refers to information presented in whatever form is agreed upon by the parties creating and using the data.

5. 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).

6. The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

I. Delivery. The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.

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

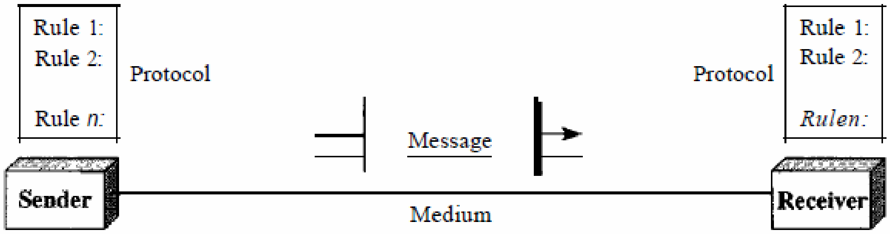
3. Timeliness. The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called *real-time* transmission.

4.. Jitter. Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 3D ms. If some of the packets arrive with 3D-ms delay and others with 4D-ms delay, an uneven quality in the video is the result.

COmponents

A data communications system has five components

Figure 1.1 *Five components ofdata communication*



:

I. Message. The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

2.Sender. The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

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

4.. Transmission medium. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.

5. Protocol. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

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 (Os or Is). 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. Appendix B discusses several different numbering systems.

Images

Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the *resolution.* For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image (better resolution), but more memory is needed to store the image. After an image is divided into pixels, each pixel is assigned a bit pattern. The size and the value of the pattern depend on the image. For an image made of only blackand- white dots (e.g., a chessboard), a I-bit pattern is enough to represent a pixel. If an image is not made of pure white and pure black pixels, you can increase the size of the bit pattern to include gray scale. For example, to show four levels of gray scale, you 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.

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.

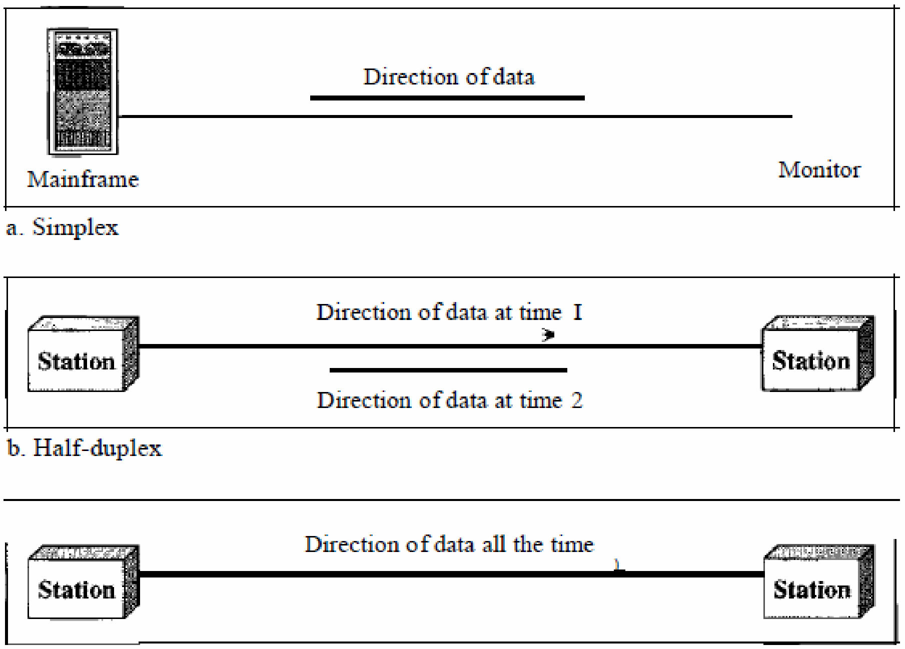
Video

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.

Data Flow

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

Figure 1.2 *Dataflow (simplex, half-duplex, andfull-duplex)*


Direction of data
Mainframe
Monitor
Direction of data at time I
Station
Station
Direction of data at time 2
Direction of data all the time
Station
Station

a. Simplex

b. Half-duplex

c. Full·duplex

Simplex

In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit; the other can only receive (see Figure 1.2a). Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output. The simplex mode can use the entire capacity of the channel to send data in one direction.

Half-Duplex

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

Full-Duplex

In full-duplex m.,lle (als@ called duplex), both stations can transmit and receive simultaneously (see Figure 1.2c). The full-duplex mode is like a tW<D-way street with traffic flowing in both directions at the same time. In full-duplex mode, si~nals going in one direction share th capacity of the link: with signals going in the other din~c~on. This sharing can occur in two ways: Either the link must contain two physically separate t:nmsmissiIDn paths, one for sending and the other

for receiving; or the capacity of the ch:arillilel is divided between signals traveling in both directions. One common example of full-duplex communication is the telephone network.
When two people are communicating by a telephone line, both can talk and listen at the same time. The full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the two directions.

1.2 NETWORKS

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

Distributed Processing

Most networks use distributed processing, in which a task is divided among multiple computers. Instead of one single large machine being responsible for all aspects of a process, separate computers (usually a personal computer or workstation) handle a subset.

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

1. 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.
2. 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.
3. Performance is often evaluated by two networking metrics: throughput and delay.
4. ‘We often need more throughput and less delay.
5. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability

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

Security

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

Physical Structures

Some network attributes.

Type of Connection

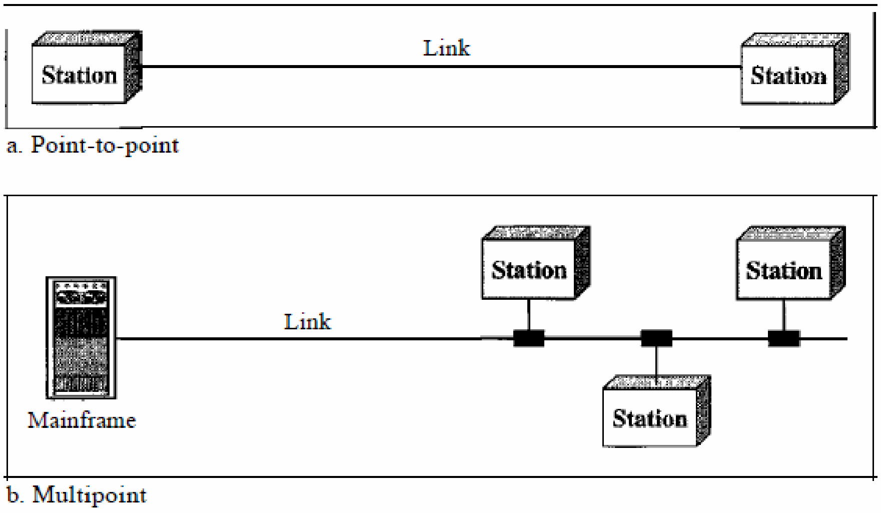
1. A network is two or more devices connected through links.

2. A link is a communications pathway that transfers data from one device to another.
3. For visualization purposes, it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time.
4. There are two possible types of connections: point-to-point and multipoint. Point-to-Poin
5. A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible (see Figure 1.3a). When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.
6. 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.

Physical Topology

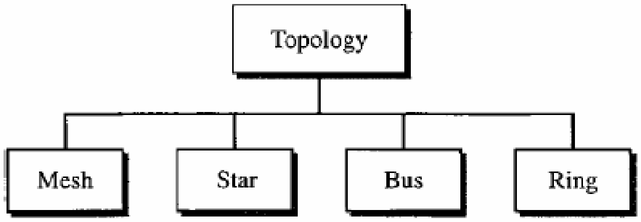
The term *physical topology* refers to the way in which a network is laid out physically.: 1\vo or more devices connect to a link; two or more links form a topology. The topology

Figure 1.3 *Types ofconnections: point-to-point and multipoint*



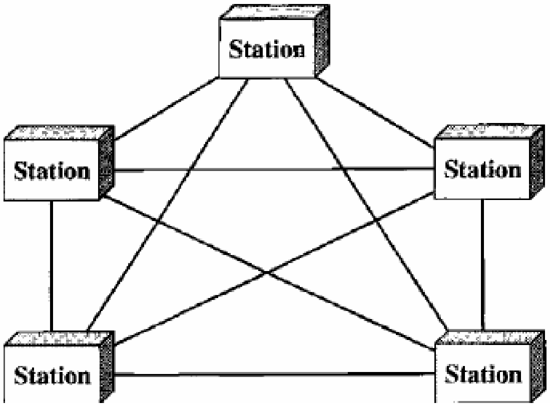
of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. There are four basic topologies possible: mesh, star, bus, and ring (see Figure 1.4).

Figure **1.4** *Categories oftopology*



1. Mesh In a mesh topology, every device has a dedicated point-to-point link to every other device.
2. The term *dedicated* means that the link carries traffic only between the two devices it connects.
3. 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 - I 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.
4. 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.
5. To accommodate that many links, every device on the network must have $n - 1$ input/output *(VO)* ports (see Figure 1.5) to be connected to the other n - 1 stations.

Figure 1.5 *A fully connected mesh topology (five devices)*



A mesh offers several advantages over other network topologies.

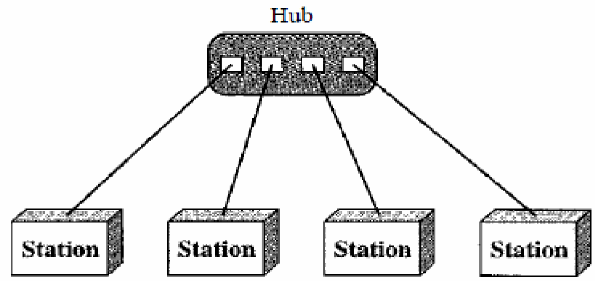
1. First, the use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices.

2. 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.
3. When every message travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages.
4. Finally, point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems.
5. since every device must be connected to every other device, installation and reconnection are difficult.
6. Also the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate.
7. Finally, the hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.
8. 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.
9. One practical example of a mesh topology is the connection of telephone regional offices in which each regional office needs to be connected to every other regional office.

Star Topology

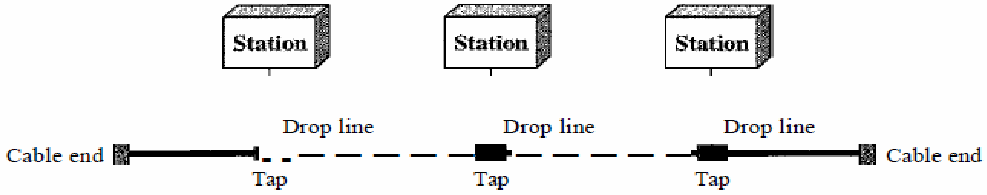
1. In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub.
2. The devices are not directly linked to one another.
3. Unlike a mesh topology, a star topology does not allow direct traffic between devices. The controller acts as an exchange: 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 1.6) . A star topology is less expensive than a mesh topology. In a star, each device needs only one link and one I/O port to connect it to any number of others. This factor also makes it easy to install and reconfigure. Far less cabling needs to be housed, and additions, moves, and deletions involve only one connection: between that device and the hub. Other advantages include robustness. If one link fails, only that link is affected. All other links remain active. This factor also lends itself to easy fault identification and fault isolation.
4. As long as the hub is working, it can be used to monitor link problems and bypass defective links.
5. One big disadvantage of a star topology is the dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead. Although a star requires far less cable than a mesh, each node must be linked to a central hub. For this reason, often more cabling is required in a star than in some other topologies (such as ring or bus).
6. The star topology is used in local-area networks (LANs High-speed LANs often use a star topology with a central hub.

Figure 1.6 *A star topology connecting four stations*



Bus Topology The preceding examples all describe point-to-point connections. A **bus topology,** on the other hand, is multipoint. One long cable acts as a **backbone** to link all the devices in a network (see Figure 1.7).

Figure 1.7 *A bus topology connecting three stations*



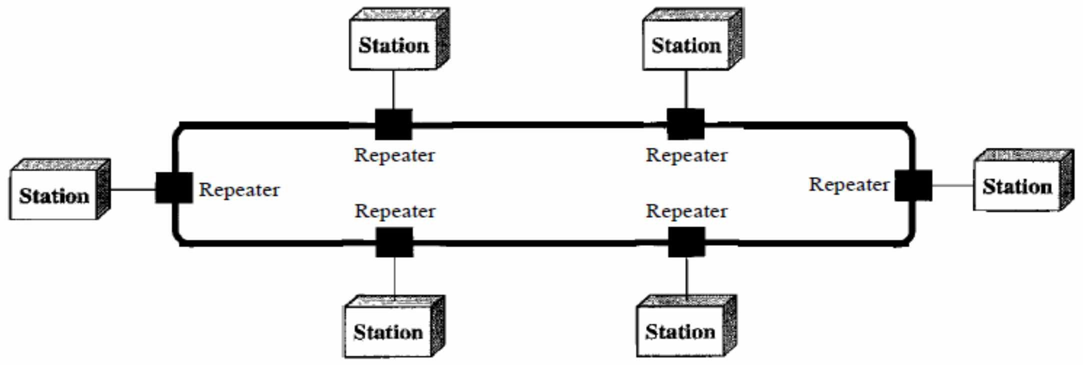
1. Nodes are connected to the bus cable by drop lines and taps.
2. A drop line is a connection running between the device and the main cable.
3. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core. As a signal travels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

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

Disadvantages include difficult reconnection and fault isolation. A bus is usually designed to be optimally efficient at installation. It can therefore be difficult to add new devices. Signal reflection at the taps can cause degradation in quality. This degradation can be controlled by limiting the number and spacing of devices connected to a given length of cable. Adding new devices may therefore require modification or replacement of the backbone. In addition, a fault or break in the bus cable stops all transmission, even between devices on the same side of the problem. The damaged area reflects signals back in the direction of origin, creating noise in both directions. Bus topology was the one of the first topologies used in the design of early localarea networks.

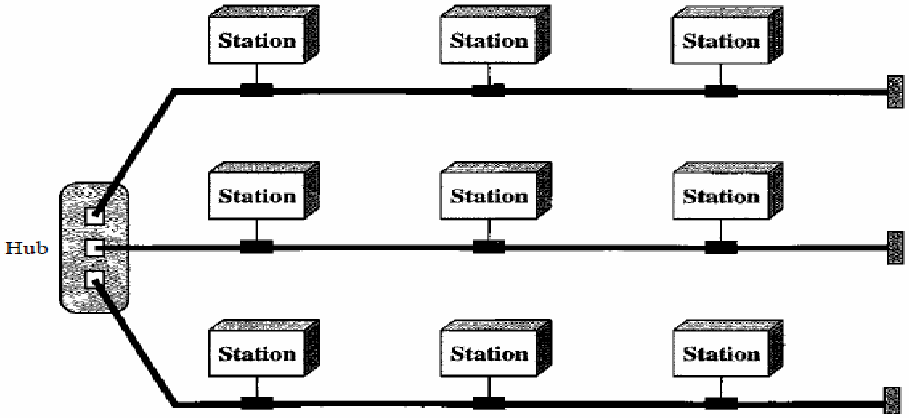
Ring Topology In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along (see Figure 1.8).

Figure 1.8 *A ring topology connecting six stations*



A ring is relatively easy to install and reconfigure. Each device is linked to only its immediate neighbors (either physically or logically). To add or delete a device requires changing only two connections. The only constraints are media and traffic considerations (maximum ring length and number of devices). In addition, fault isolation is simplified. Generally in a ring, a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location. However, unidirectional traffic can be a disadvantage. In a simple ring, a break in the ring (such as a disabled station) can disable the entire network. This weakness can be solved by using a dual ring or a switch capable of closing off the break. Ring topology was prevalent when IBM introduced its local-area network Token Ring. Today, the need for higher-speed LANs has made this topology less popular. Hybrid Topology A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology as shown in Figure 1.9.

Figure 1.9 A *hybrid topology: a star backbone with three bus networks*



Network Models

Computer networks are created by different entities. Standards are needed so that these heterogeneous networks can communicate with one another. The two best-known standards are the OSI model and the Internet model. In Chapter 2 we discuss these two models. The OSI (Open Systems Interconnection) model defines a seven-layer network; the Internet model defines a five-layer network. This book is based on the Internet model with occasional references to the OSI model.

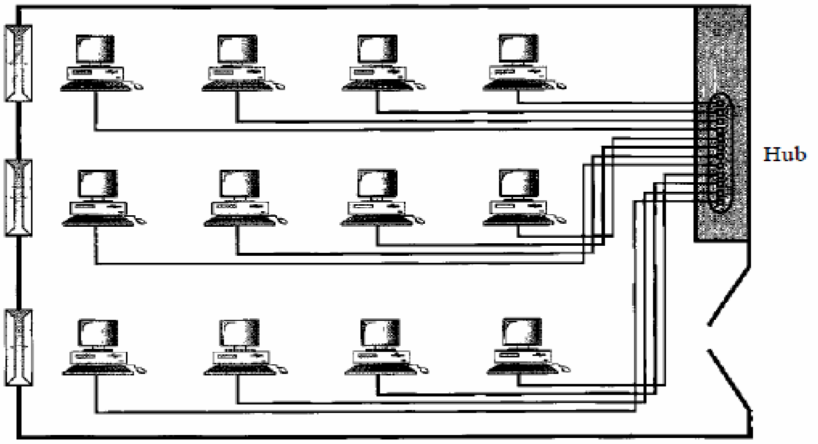
Categories of Networks

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

Local Area Network

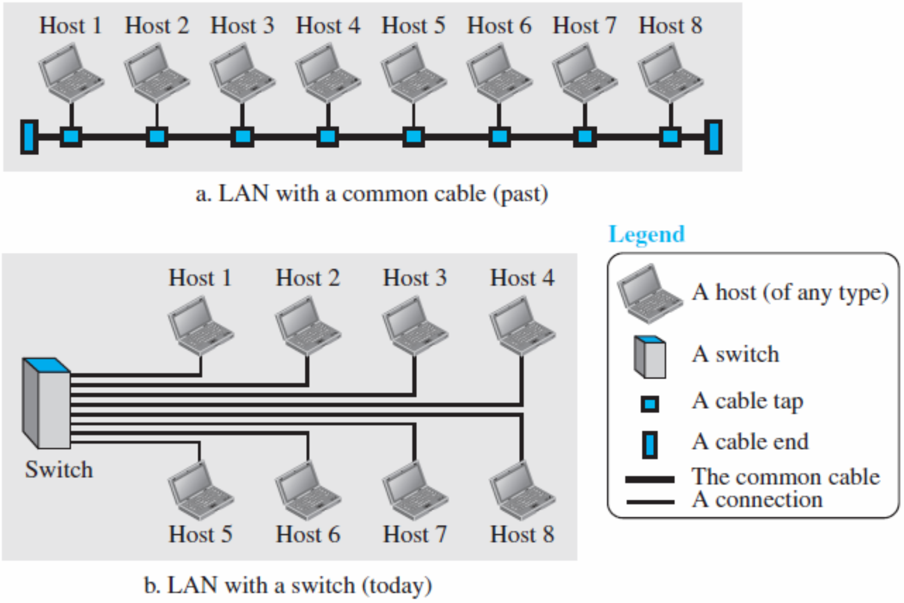
A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus (see Figure 1.10). Depending on the needs of an organization and the type of technology used, a LAN can be as simple as two PCs and a printer in someone's home office; or it can extend throughout a company and include audio and video peripherals. Currently, LAN size is limited to a few kilometers.

Figure 1.10 *An isolated LAN connecting 12 computers to a hub in a closet*



LANs are designed to allow resources to be shared between personal computers or workstations. The resources to be shared can include hardware (e.g., a printer), software (e.g., an application program), or data. A common example of a LAN, found in many business environments, links a workgroup of task-related computers, for example, engineering workstations or accounting PCs. One of the computers may be given a largecapacity disk drive and may become a server to clients. Software can be stored on this central server and used as needed by the whole group. In this example, the size of the LAN may be determined by licensing restrictions on the number of users per copy of software, or by restrictions on the number of users licensed to access the operating system. In addition to size, LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium. The most common LAN topologies are bus, ring, and star. Early LANs had data rates in the 4 to 16 megabits per second (Mbps) range.

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



a. LAN with a common cable (past)

b. LAN with a switch (today)

Wide Area Network

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

Figure 1.9 *A point-to-point WAN*

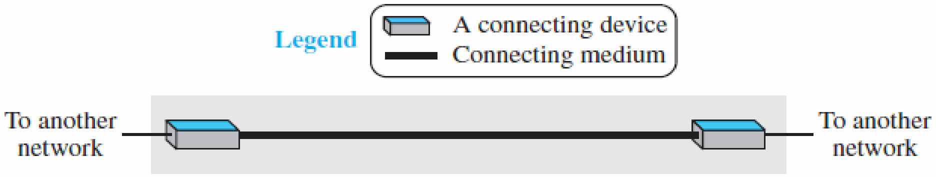
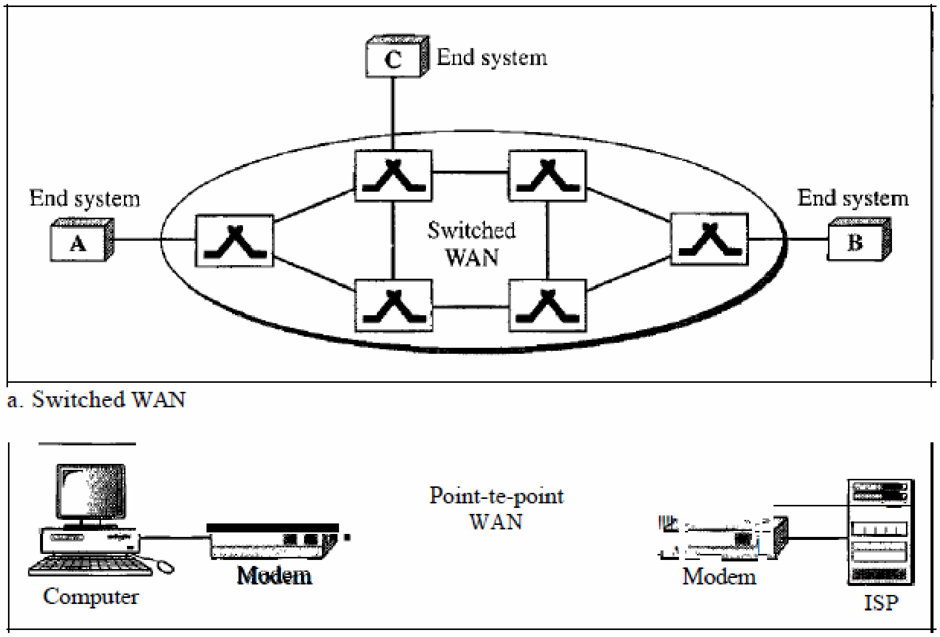


Figure 1.11 *WANs: a switched WAN and a point-to-point WAN*



a. Switched WAN

b. Point-to-point WAN

Switched WAN

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

Figure 1.10 *A switched WAN*

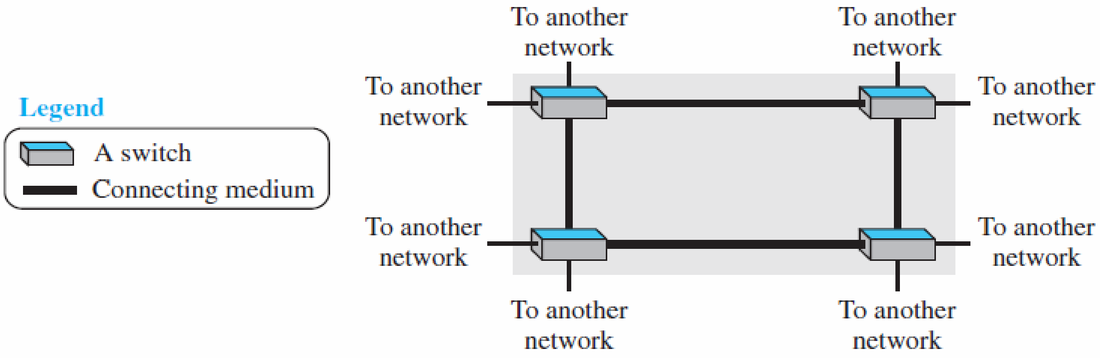
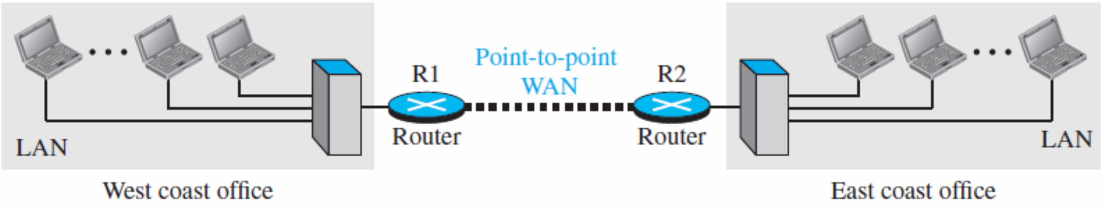
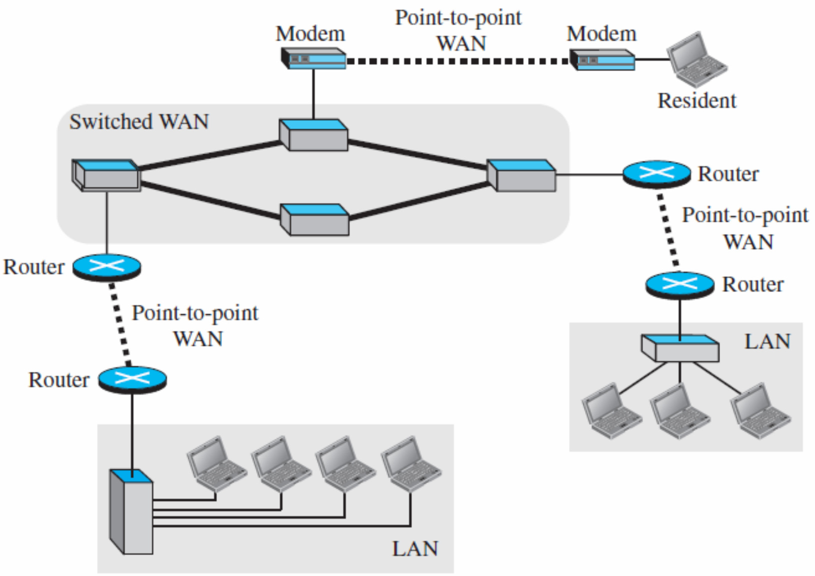


Figure 1.11 *An internetwork made of two LANs and one point-to-point WAN*



An early example of a switched WAN is X.25, a network designed to provide connectivity between end users., X.25 is being gradually replaced by a high-speed, more efficient network called Frame Relay. A good example of a switched WAN is the asynchronous transfer mode (ATM) network, which is a network with fixed-size data unit packets called cells. Another example ofWANs is the wireless WAN that is becoming more and more popular.

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



Metropolitan Area Networks

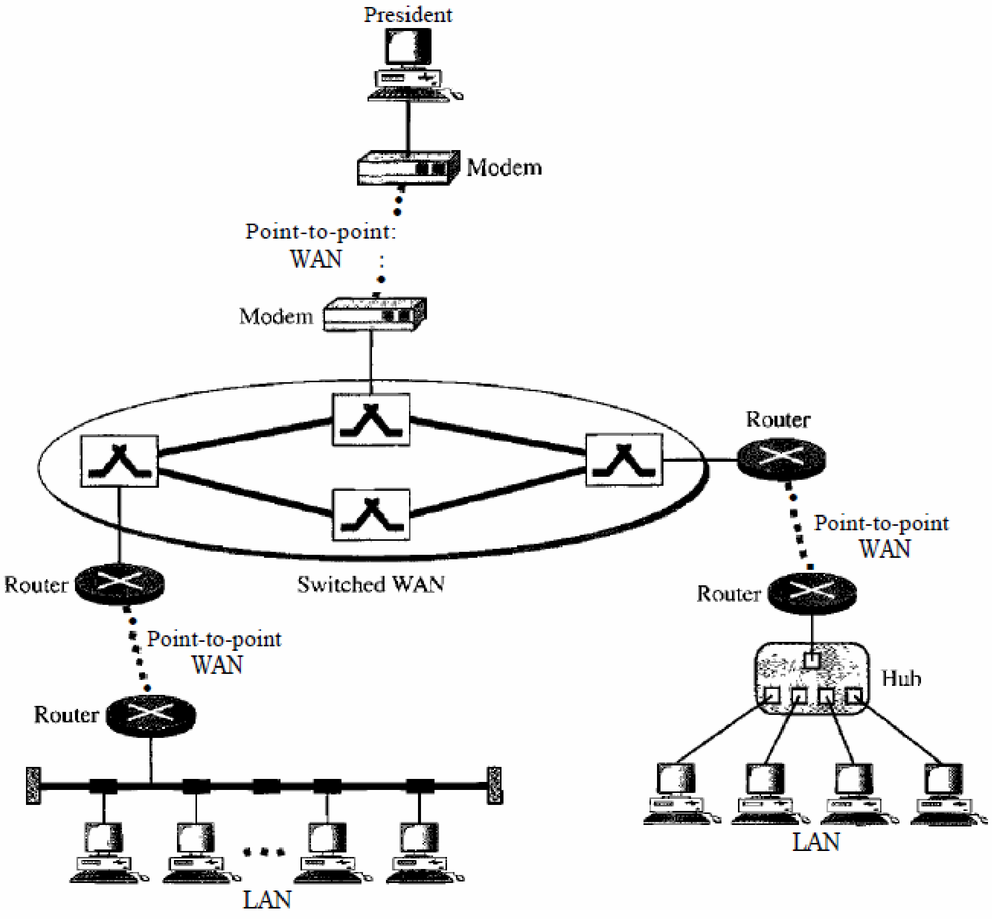
A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city. A good example of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

Interconnection of Networks: Internetwork

When two or more networks are connected, they become 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. The established office on the west coast has a bus topology LAN; the newly opened office on the east coast has a star topology LAN. The president of the company lives somewhere in the middle and needs to have control over the company from her home. To create a backbone WAN for connecting these three entities (two LANs and the president's computer), a switched WAN (operated by a service provider such as a telecom company) has been leased. To connect the LANs to this switched WAN, however, three point-to-point WANs are required. These point-to-point WANs can be a high-speed DSL line offered by a telephone company or a cable modern line offered by a cable TV provider as shown in Figure 1.12.

Figure 1.12 *A heterogeneous network made offour WANs and two LANs*



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.**

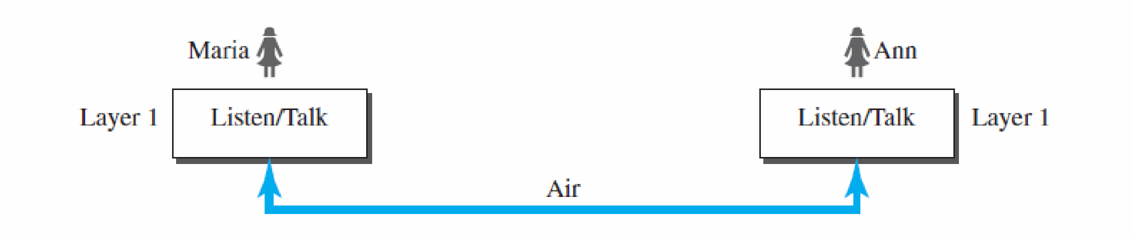
Scenarios

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

First Scenario

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

Figure 2.1 *A single-layer protocol*



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

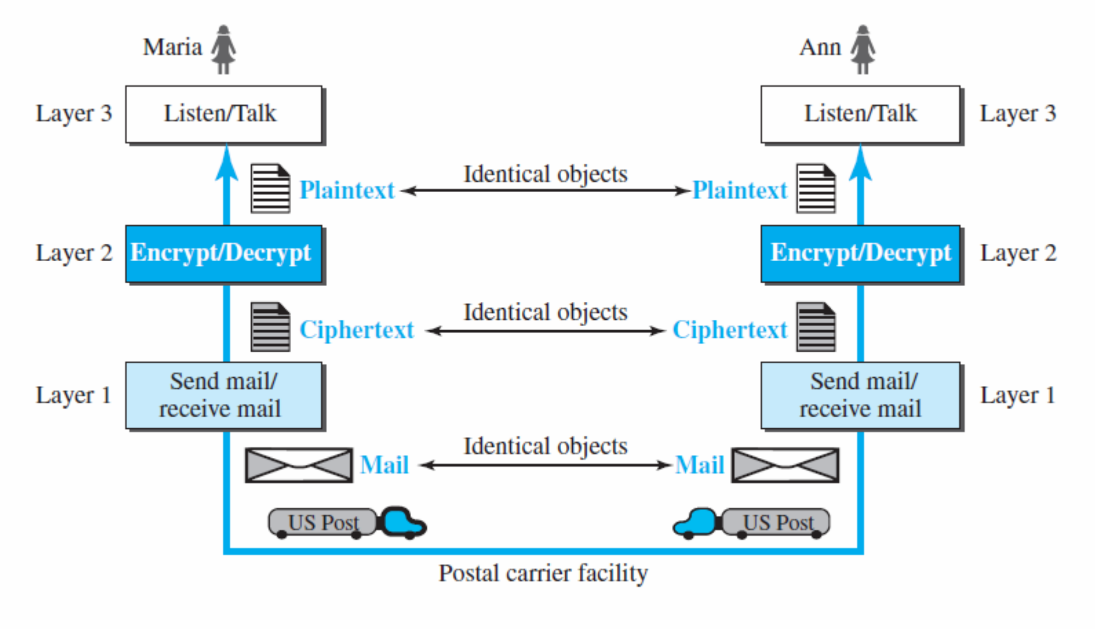
We can see that the protocol used by Maria and Ann is different from the communication between a professor and the students in a lecture hall. The communication in the second case is mostly monolog; the professor talks most of the time unless a student has a question, a situation in which the protocol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

Second Scenario

In the second scenario, we assume that Ann is offered a higher-level position in her company, but needs to move to another branch located in a city very far from Maria. The two friends still want to continue their communication and exchange ideas because they have come up with an innovative project to start a new business when they both
retire. They decide to continue their conversation using regular mail through the post office. However, they do not want their ideas to be revealed by other people if the letters are intercepted. They agree on an encryption/decryption technique. The sender of the letter encrypts it to make it unreadable by an intruder; the receiver of the letter decrypts it to get the original letter., but for the moment we assume that Maria and Ann use one technique that makes it hard to decrypt the letter if one does not have the key for doing so. Now we can say that the communication between Maria and Ann takes place in three layers, as shown in Figure 2.2. We assume that Ann and Maria each have three machines (or robots) that can perform the task at each layer.
Let us assume that Maria sends the first letter to Ann.
Maria talks to the machine at the third layer as though the machine is Ann and is listening to her.
The third layer machine listens to what Maria says and creates the plaintext (a letter in English), which is passed to the second layer machine.
The second layer machine takes the plaintext, encrypts it, and creates the ciphertext, which is passed to the first layer machine.
The first layer machine, presumably a robot, takes the ciphertext, puts it in an envelope, adds the sender and receiver addresses, and mails it.
At Ann's side,

the first layer machine picks up the letter from Ann's mail box, recognizing the letter from Maria by the sender address. The machine takes out the ciphertext from the envelope and delivers it to the second layer machine.
The second layer machine decrypts the message, creates the plaintext, and passes the plaintext to the third-layer machine.
The third layer machine takes the plaintext and reads it as though Maria is speaking.
Figure 2.2 *A three-layer protocol* Maria
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

Figure 2.2 *A three-layer protocol*



change the whole machine. In the present situation, they need to change only the second layer machine; the other two can remain the same. This is referred to as *modularity*.

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

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

Another advantage of protocol layering, which cannot be seen in our simple examples but reveals itself when we discuss protocol layering in the Internet, is that communication does not always use only two end systems; there are intermediate systems that need only some layers, but not all layers. If we did not use protocol layering, we would have to make each intermediate system as complex as the end systems, which makes the whole system more expensive. Is there any disadvantage to protocol layering? One can argue that having a single layer makes the job easier. There is no need for each layer to provide a service to the upper layer and give service to the lower layer. For example, Ann and Maria could find or build one machine that could do all three tasks. However, as mentioned above, if one day they found that their code was broken, each would have to replace the whole machine with a new one instead of just changing the machine in the second layer.

2.1.2 Principles of Protocol Layering

Let us discuss two principles of protocol layering.

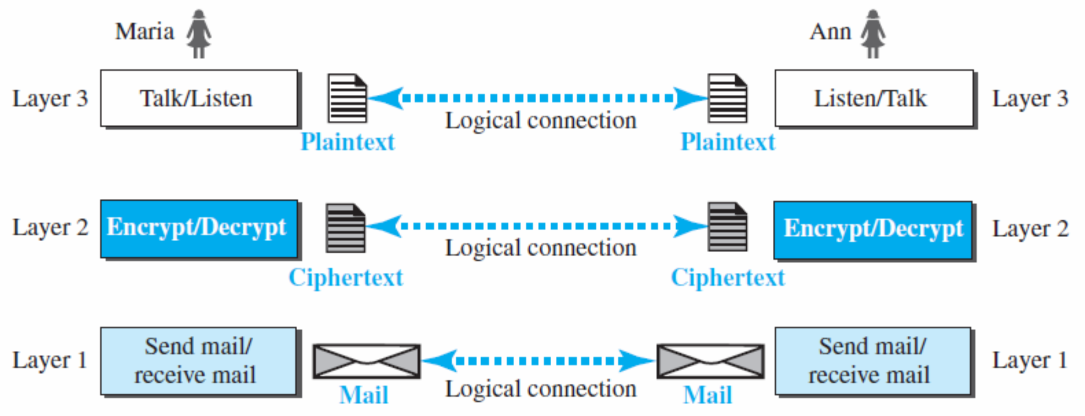
First Principle

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

Second Principle

The second principle that we need to follow in protocol layering is that the two objects under each layer at both sites should be identical. For example, the object under layer 3 at both sites should be a plaintext letter. The object under layer 2 at both sites should be a ciphertext letter. The object under layer 1 at both sites should be a piece of mail.

Figure 2.3 *Logical connection between peer layers*



2.1.3 Logical Connections

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

see that the concept of logical connection will help us better understand the task of layering we encounter in data communication and networking.

2.2 TCP/IP PROTOCOL SUITE

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

2.2.1 Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch. We also assume that the links are connected by one router, as shown in Figure 2.5. shows, we have five communicating devices in this communication: source host (computer A), the link-layer switch in link 1, the router, the link-layer switch in link 2, and the destination host (computer B). Each device is involved with a set of layers depending on the role of the device in the internet. The two hosts are involved in all five layers; the source host needs to create a message in the application layer and send it down the layers so that it is physically sent to the destination host. The destination host needs to receive the communication at the physical layer and then deliver it through the other layers to the application layer. The router is involved in only three layers; there is no transport or application layer in a router as long as the router is used only for routing. Although a router is always involved in one network layer, it is involved in *n* combinations of link and physical layers in which *n* is the number of links the router is connected to. The reason is that each link may use its own data-link or physical protocol. For example, in the above figure, the router is involved in three links, but the message sent from source A to destination B is involved in two links. Each link may be using different link-layer and physical-layer protocols; the router needs to receive a packet from link 1 based on one pair of protocols and deliver it to link 2 based on another pair of protocols.

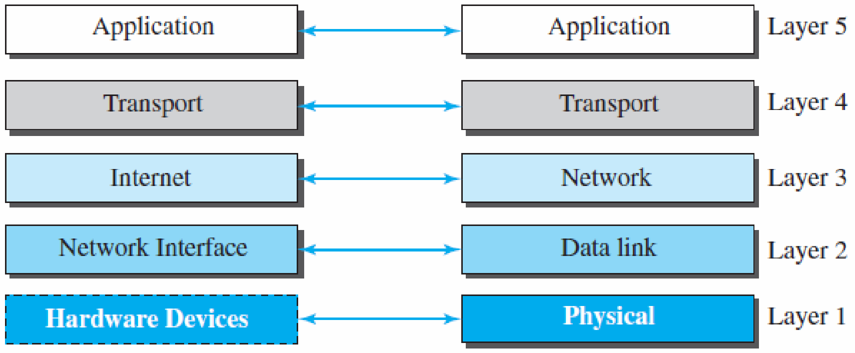
A link-layer switch in a link, however, is involved only in two layers, data-link and physical. Although each switch in the above figure has two different connections, the connections are in the same link, which uses only one set of protocols. This means that, unlike a router, a link-layer switch is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

2.2.2 Layers in the TCP/IP Protocol Suite

After the above introduction, we briefly discuss the functions and duties of layers in the TCP/IP protocol suite. Each layer is discussed in detail in the next five parts of the book. To better understand the duties of each layer, we need to think about the logical connections between layers. Figure 2.6 shows logical connections in our simple
internet. Using logical connections makes it easier for us to think about the duty of each layer. As the figure shows, the duty of the application, transport, and network layers is end-to-end. However, the duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router. In other words, the domain of duty of the top three
layers is the internet, and the domain of duty of the two lower layers is the link. Another way of thinking of the logical connections is to think about the data unit created from each layer. In the top three layers, the data unit (packets) should not be changed by any router or link-layer switch. In the bottom two layers, the packet created
by the host is changed only by the routers, not by the link-layer switches. Figure 2.7 shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device

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

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



a. Original layers

b. Layers used in this book

2.2.3 Description of Each Layer

Physical Layer

We can say that the physical layer is responsible for carrying individual bits in a frame across the link. Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a logical communication because there is another, hidden layer, the transmission media, under the physical layer. Two devices are connected by a transmission medium (cable or air). We need to know that the transmission medium does not carry bits; it carries electrical or optical signals. So the bits received in a frame from the data-link layer are transformed and sent through the transmission media, but we can think that the logical unit between two physical layers in two devices is a *bit*. There are several protocols that transform a bit to a signal. We discuss them in Part II when we discuss the physical layer and the transmission media. We have seen that an internet is made up of several links (LANs and WANs) connected by routers. There may be sev ral overlapping sets of links hat a datagram can travel from the host to the destination. The routers are responsible for choosing the *best* links.

However, when the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link. The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN. We can also have different protocols used with any link type. In each case, the data-link layer is responsible for moving the packet through the link. TCP/IP does not define any specific protocol for the data-link layer. It supports all the standard and proprietary protocols. Any protocol that can take the datagram

Figure 2.5 *Communication through an internet*

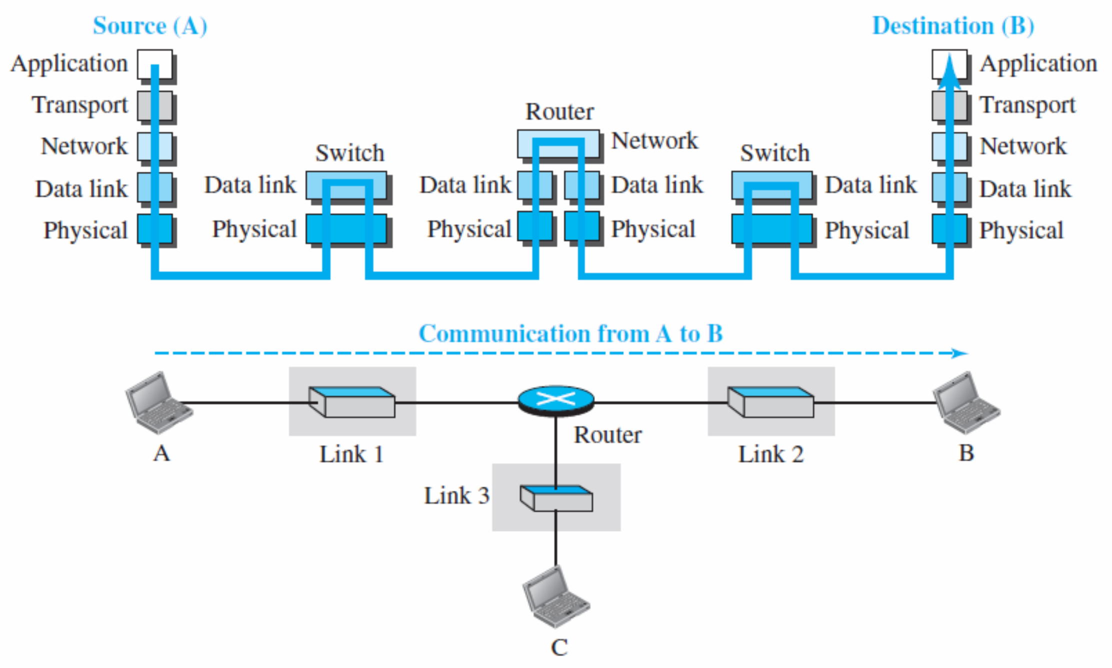
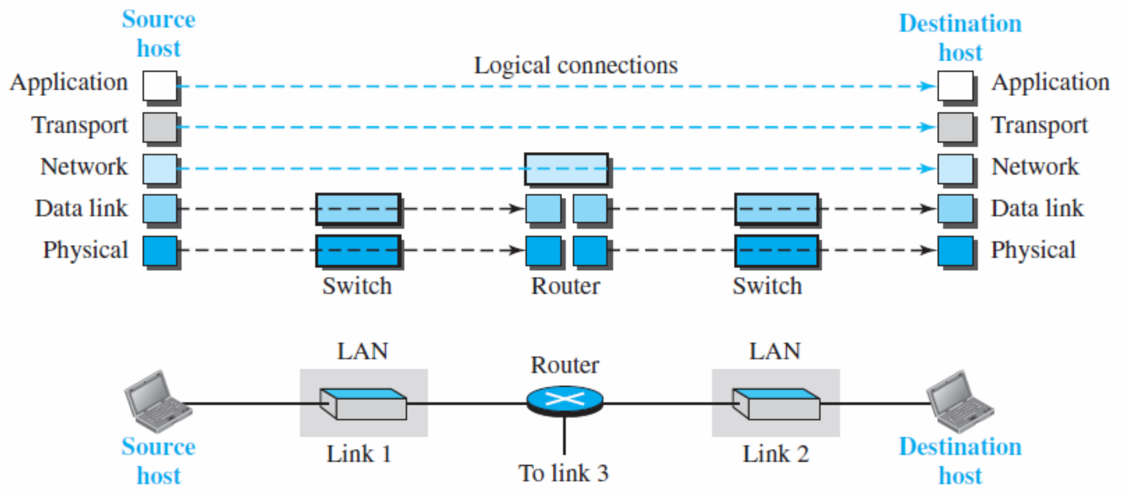


Figure 2.6 *Logical connections between layers of the TCP/IP protocol suite*

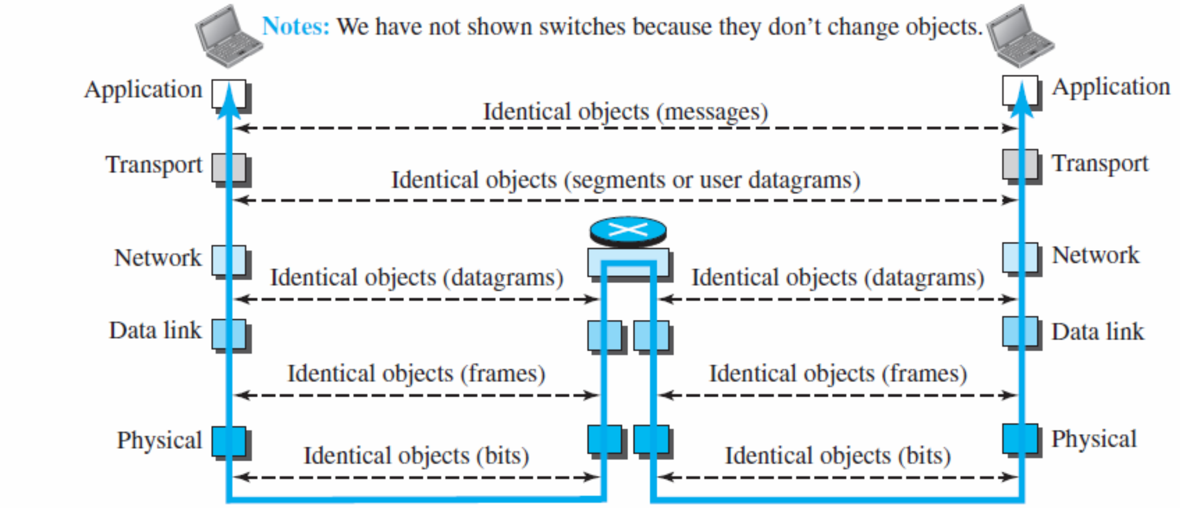


and carry it through the link suffices for the network layer. The data-link layer takes a datagram and encapsulates it in a packet called a *frame*. Each link-layer protocol may provide a different service. Some link-layer protocols
provide complete error detection and correction, some provide only error correction

Network Layer

The network layer is responsible for creating a connection between the source computer and the destination computer. The communication at the network layer is host-to-host. However, since there can be several routers from the source to the destination, the routers in the path are responsible for choosing the best route for each packet. We can say that the network layer is responsible for host-to-host communication and routing the packet through possible routes. Again, we may ask ourselves why we need the network layer. We could have added the routing duty to the transport layer and dropped this layer. One reason, as we said before, is the separation of different tasks between different layers. The second reason is that the routers do not need the application and ransport layers. Separating the tasks allows us to use fewer protocols on the routers. The network layer in the Internet includes the main protocol, Internet Protocol (IP), that defines the format of the packet, called a datagram at the network layer. IP also defines the format and the structure of addresses used in this layer. IP is also responsible for routing a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path. IP is a connectionless protocol that provides no flow control, no error control, and no congestion control services. This means that if any of theses services is required for an application, the application should rely only on the transport-layer protocol. The network layer also includes unicast (one-to-one) and multicast (one-to-many) routing protocols. A routing protocol does not take part in routing (it is the responsibility of IP), but it creates forwarding tables for routers to help them in the routing process. The network layer also has some auxiliary protocols that help IP in its delivery and routing tasks. The Internet Control Message Protocol (ICMP) helps IP to report some problems when routing a packet. The Internet Group Management Protocol (IGMP) is another protocol that helps IP in multitasking. The Dynamic Host Configuration Protocol (DHCP) helps IP to get the network-layer address for a host. T he 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 givenThe logical connection at the transport layer is also end-to-end. The t ransport layer at the source host gets the message from the application layer, encapsulates it in a transportlayer packet (called a *segment* or a *user datagram* in different protocols) and sends it, through the logical (imaginary) connection, to the transport layer at the destination host. In other words, the transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host. e may ask why we need an end-to-end transport layer when we already have an end-to-end application layer. The reason is the separation of tasks and duties, which we discussed earlier. The transport layer should be independent of the application layer. In addition, we will see that we have more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement. As we said, there are a few transport-layer protocols in the Internet, each designed for some specific task. The main protocol, Transmission Control Protocol (TCP), is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data. It creates a logical pipe between two TCPs for transferring a stream of bytes. TCP provides flow control (matching the sending data rate of the source host with the receiving

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



Notes: We have not shown switches because they don't change objects.

Application
Transport
Network
Data link
Physical
Application
Transport
Network
Data link
Physical
Identical objects (messages)
Identical objects (segments or user datagrams)
Identical objects (datagrams)
Identical objects (datagrams)
Identical objects (frames)
Identical objects (frames)
Identical objects (bits)
Identical objects (bits)

data rate of the destination host to prevent overwhelming the destination), error control (to guarantee that the segments arrive at the destination without error and resending the corrupted ones), and congestion control to reduce the loss of segments due to congestion in the network. The other common protocol, User Datagram Protocol (UDP), is a connectionless protocol that transmits user datagrams without first creating a logical connection. In UDP, each user datagram is an independent entity without being related to the previous or the next one (the meaning of the term *connectionless*). UDP is a simple protocol that does not provide flow, error, or congestion control. Its simplicity, which means small overhead, is attractive to an application program that needs to send short messages and cannot afford the retransmission of the packets involved in TCP, when a packet is corrupted or lost. A new protocol, Stream Control Transmission Protocol (SCTP) is designed to respond to new applications that are emerging in the multimedia.

Application Layer

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

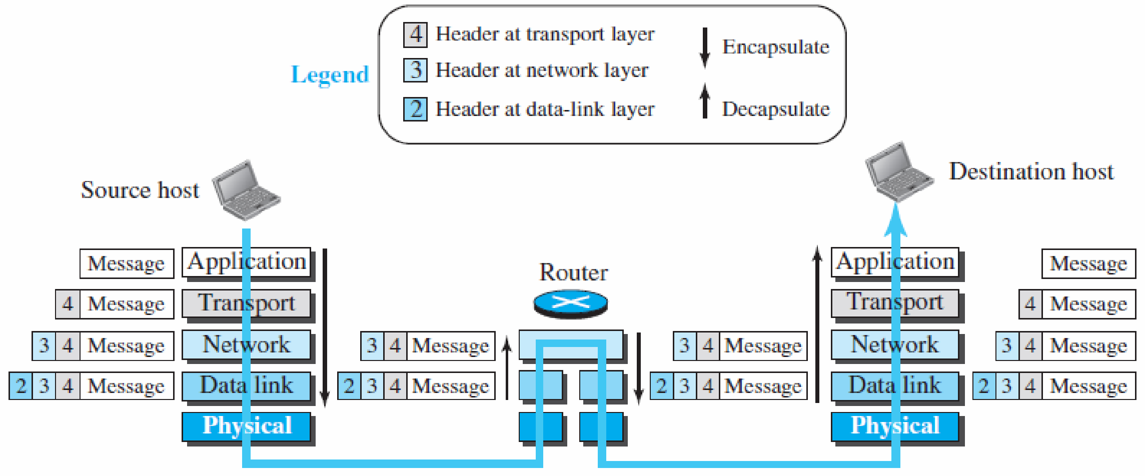
The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW). The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e-mail) service. The Fil e Transfer Protocol (FTP) is used for transferring files from one host to another. The Terminal Network (TELNET) and

Secure Shell (SSH) are used for accessing a site remotely. The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels. The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer. The Internet Group Management Protocol

(IGMP) is used to collect membership in a group. We discuss most of these protocols in Chapter 26 and some in other chapters.

2.2.4 Encapsulation and Decapsulation

Figure 2.8 *Encapsulation/Decapsulation*



One of the important concepts in protocol layering in the Internet is encapsulation/decapsulation. Figure 2.8 shows this concept for the small internet in Figure 2.5. We have not shown the layers for the link-layer switches because no encapsulation/ decapsulation occurs in this device. In Figure 2.8, we show the encapsulation in the source host, decapsulation in the destination host, and encapsulation and decapsulation in the router.

Encapsulation at the Source Host

At the source, we have only encapsulation.

1. At the application layer, the data to be exchanged is referred to as a *message*. A message normally does not contain any header or trailer, but if it does, we refer to the whole as the message. The message is passed to the transport layer.

2. The transport layer takes the message as the payload, the load that the transport layer should take care of. It adds the transport layer header to the payload, which contains the identifiers of the source and destination application programs that

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

Encapsulation at the Source Host

At the source, we have only encapsulation.

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

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

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

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

Decapsulation and Encapsulation at the Router

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

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

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

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

Decapsulation at the Destination Host

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

2.2.5 Addressing

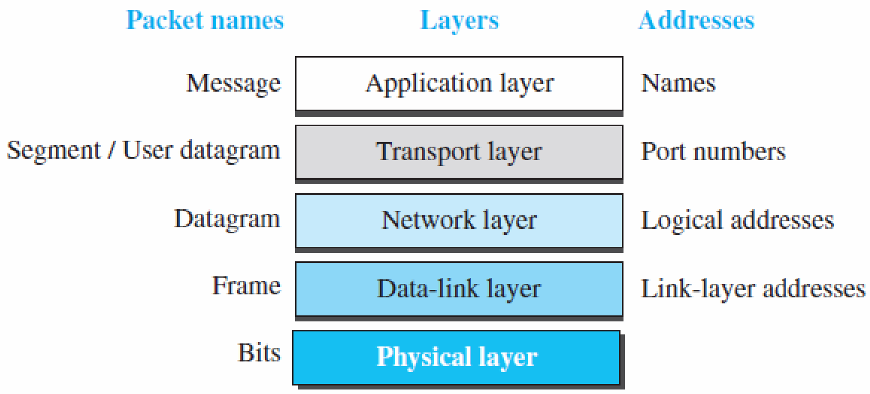
It is worth mentioning another concept related to protocol layering in the Internet, *addressing*. As we discussed before, we have logical communication between pairs of layers in this model. Any communication that involves two parties needs two addresses: source address and destination address. Although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

that layer, and the packet name at that layer.

At the application address, such as *somebody@coldmail.com.* At the transport layer, addresses are called port numbers, and these define the application-layer programs at the source and

destination. Port numbers are local addresses that distinguish between several programs running at the same time. At the network-layer, the addresses are global, with the whole Internet as the scope. A network-layer address uniquely defines the connection of a device to the Internet. The link-layer addresses, sometimes called MAC addresses, are
locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN).

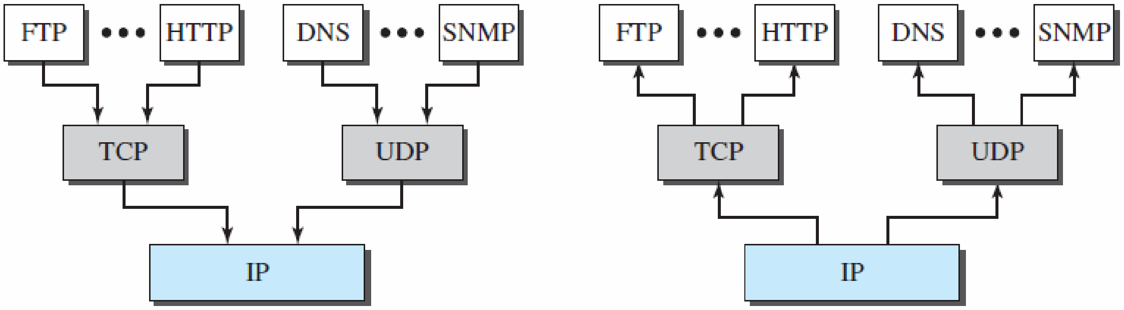
Figure 2.9 *Addressing in the TCP/IP protocol suite*



Packet names	Layers	Addresses
Message	Application layer	Names
Segment / User datagram	Transport layer	Port numbers
Datagram	Network layer	Logical addresses
Frame	Data-link layer	Link-layer addresses
Bits	Physical layer	

2.2.6 Multiplexing and Demultiplexing

Figure 2.10 *Multiplexing and demultiplexing*



a. Multiplexing at source

b. Demultiplexing at destination

2.3 THE OSI MODEL

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

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

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

ISO is the organization; OSI is the model.

2.3.1 OSI versus TCP/IP

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

as shown in Figure 2.12.

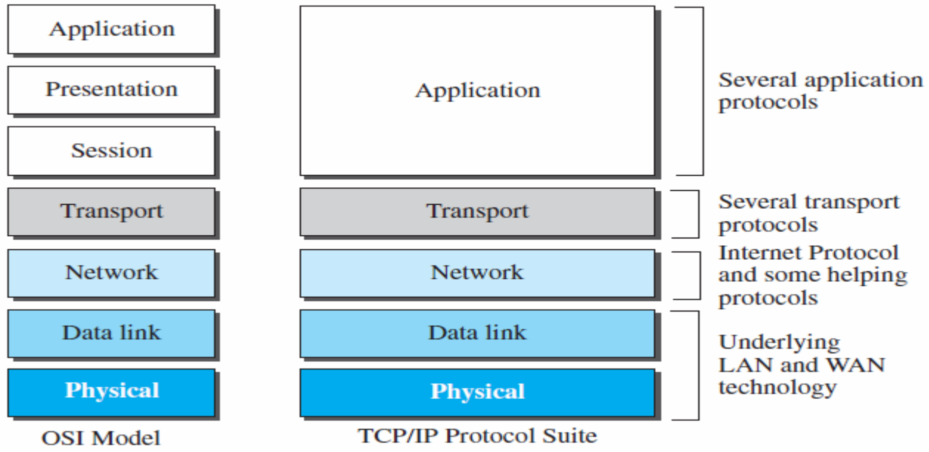
Two reasons were mentioned for this decision. First, TCP/IP has more than one transport-layer protocol. Some of the functionalities of the session layer are available in some of the transport-layer protocols. Second, the application layer is not only one piece of software. Many applications can be developed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

2.3.2 Lack of OSI Model's Success

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

authority to switch from the TCP/IP protocol suite to the OSI model.

Figure 2.12 *TCP/IP and OSI model*



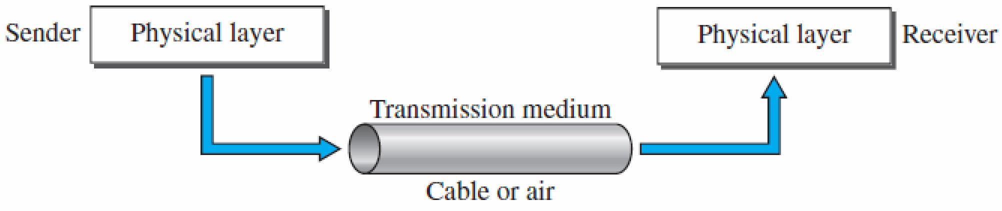
Following are the differences between OSI and TCP/IP Reference Model −

OSI	TCP/IP
OSI represents **Open System Interconnection**.	TCP/IP model represents the Transmission Control Protocol / Internet Protocol.
OSI is a generic, protocol independent standard. It is acting as an interaction gateway between the network and the final-user.	TCP/IP model depends on standard protocols about which the computer network has created. It is a connection protocol that assigns the network of hosts over the internet.
The OSI model was developed first, and then protocols were created to fit the network architecture's needs.	The protocols were created first and then built the TCP/IP model.
It provides quality services.	It does not provide quality services.
The OSI model represents defines administration, interfaces and conventions. It describes clearly which layer provides services.	It does not mention the services, interfaces, and protocols.
The protocols of the OSI model are better unseen and can be returned with another appropriate protocol quickly.	The TCP/IP model protocols are not hidden, and we cannot fit a new protocol stack in it.
It is difficult as distinguished to TCP/IP.	It is simpler than OSI.
It provides both connection and connectionless oriented transmission in the network layer; however, only connection-oriented transmission in the transport layer.	It provides connectionless transmission in the network layer and supports connecting and connectionless-oriented transmission in the transport layer.
It uses a horizontal approach.	It uses a vertical approach.
The smallest size of the OSI header is 5 bytes.	The smallest size of the TCP/IP header is 20 bytes.
Protocols are unknown in the OSI model and are returned while the technology modifies.	In TCP/IP, returning protocol is not difficult.

Transmission Media

Transmission media are actually located below the physical layer and are directly controlled by the physical layer. We could say that transmission media belong to layer zero. Figure 7.1 shows the position of transmission media in relation to the physical layer. A **transmission medium** can be broadly defined as anything that can carry information from a source to a destination. For example, the transmission medium for two people having a dinner conversation is the air. The air can also be used to convey the message in a smoke signal or semaphore. For a written message, the transmission medium might be a mail carrier, a truck, or an airplane. In data communications the definition of the information and the transmission medium is more specific. The transmission medium is usually free space, metallic cable, or fiber-optic cable. The information is usually a signal that is the result of a conversion of data from another form.

Figure 7.1 *Transmission medium and physical layer*



The use of long-distance communication using electric signals started with the invention of the telegraph by Morse in the 19th century. Communication by telegraph was slow and dependent on a metallic medium.

Extending the range of the human voice became possible when the telephone was invented in 1869. Telephone communication at that time also needed a metallic medium to carry the electric signals that were the result of a conversion from the human voice. The communication was, however, unreliable due to the poor quality of the wires. The lines were often noisy and the technology was unsophisticated. Wireless communication started in 1895 when Hertz was able to send highfrequency signals. Later, Marconi devised a method to send telegraph-type messages over the Atlantic Ocean. We have come a long way. Better metallic media have been invented (twisted-pair and coaxial cables, for example). The use of optical fibers has increased the data rate incredibly. Free space (air, vacuum, and water) is used more efficiently, in part due to the technologies (such as modulation and multiplexing) discussed in the previous chapters.

computers and other telecommunication devices use signals to represent data. These signals are transmitted from one device to another in the form of electromagnetic energy, which is propagated through transmission media.

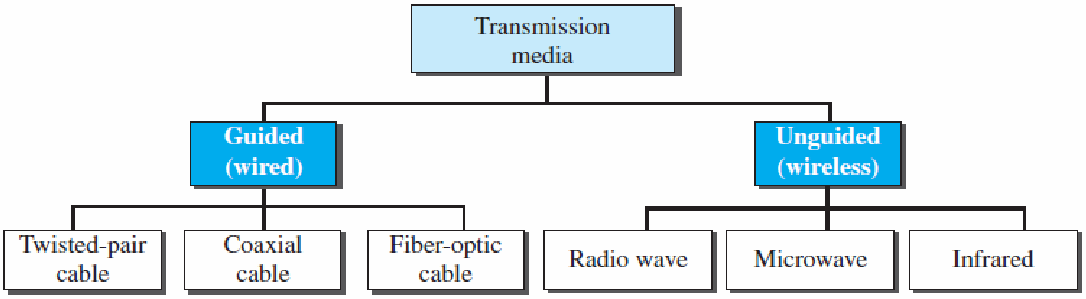
Electromagnetic energy, a combination of electric and magnetic fields vibrating in relation to each other, includes power, radio waves, infrared light, visible light, ultraviolet light, and X, gamma, and cosmic rays. Each of these constitutes a portion of the **electromagnetic spectrum.** Not all portions of the spectrum are currently usable for telecommunications, however. The media to harness those that are usable are also limited to a few types.

In telecommunications, transmission media can be divided into two broad categories: guided and unguided. Guided media include twisted-pair cable, coaxial cable, and fiber-optic cable. Unguided medium is free space. Figure 7.2 shows this taxonomy.

7.2 GUIDED MEDIA

Guided media, which are those that provide a conduit from one device to another, include **twisted-pair cable, coaxial cable,** and **fiber-optic cable.** A signal traveling along any of these media is directed and contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current. **Optical fiber** is a cable that accepts and transports signals in the form of light.

Figure 7.2 *Classes of transmission media*

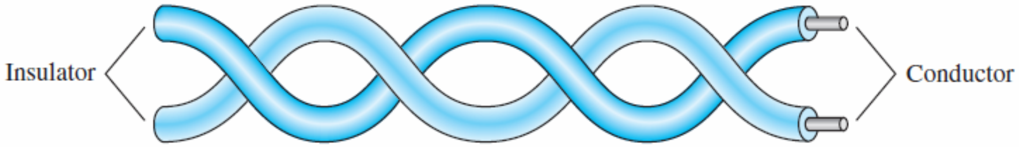


7.2.1 Twisted-Pair Cable

A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together, as shown in Figure 7.3. One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference. The receiver uses the difference between the two.

In addition to the signal sent by the sender on one of the wires, interference (noise) and crosstalk may affect both wires and create unwanted signals. If the two wires are parallel, the effect of these unwanted signals is not the same in both wires because they are at different locations relative to the noise or crosstalk sources (e.g., one is closer and the other is farther). This results in a difference at the receiver. By twisting the pairs, a bala ce is maintained. For example, suppose in one twist, one wire is closer to the noise source and the other is farther; in the next twist, the reverse is true. Twisting makes it probable that both wires are equally affected by external influences (noise or crosstalk). This means that the receiver, which calculates the difference between the two, receives no unwanted signals. The unwanted signals are mostly canceled out. From the above discussion, it is clear that the number of twists per unit of length (e.g., inch) has some effect on the quality of the cable.

Figure 7.3 *Twisted-pair cable*



Unshielded Versus Shielded Twisted-Pair Cable

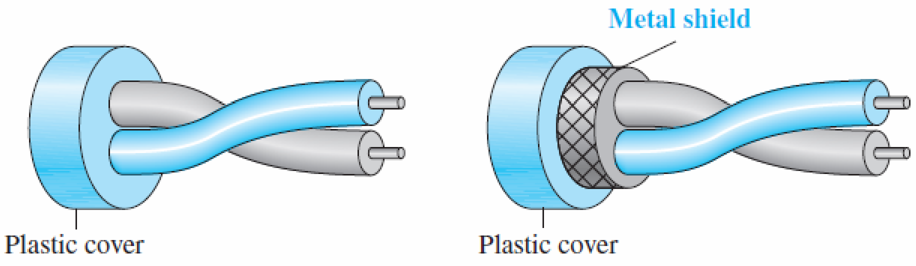
The most common twisted-pair cable used in communications is referred to as ***unshielded twisted-pair* (UTP).** IBM has also produced a version of twisted-pair cable for its use, called ***shielded twisted-pair* (STP).** STP cable has a metal foil or braidedmesh covering that encases each pair of insulated conductors. Although metal casing

improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive. Figure 7.4 shows the difference between UTP and STP.

Categories

The Electronic Industries Association (EIA) has developed standards to classify unshielded twisted-pair cable into seven categories. Categories are determined by cable quality, with 1 as the lowest and 7 as the highest. Each EIA category is suitable for specific uses. Table 7.1 shows these categories.

Figure 7.4 *UTP and STP cables*

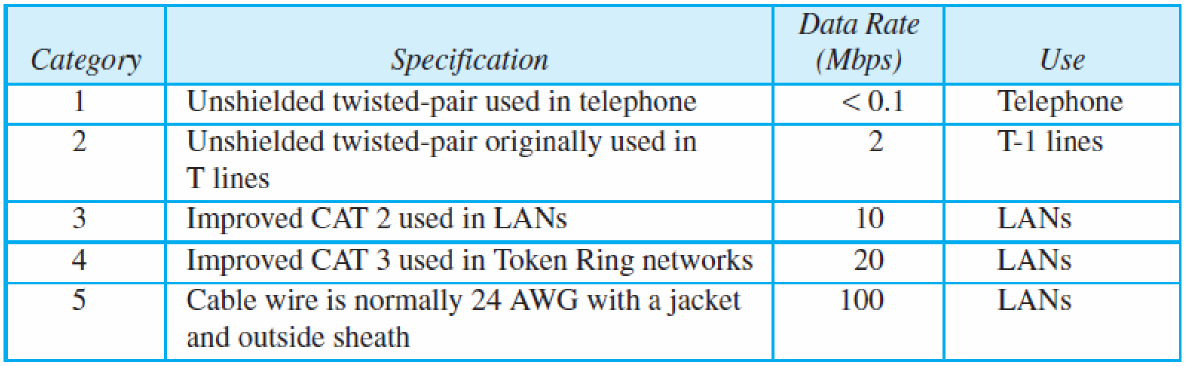


a. UTP

b. STP

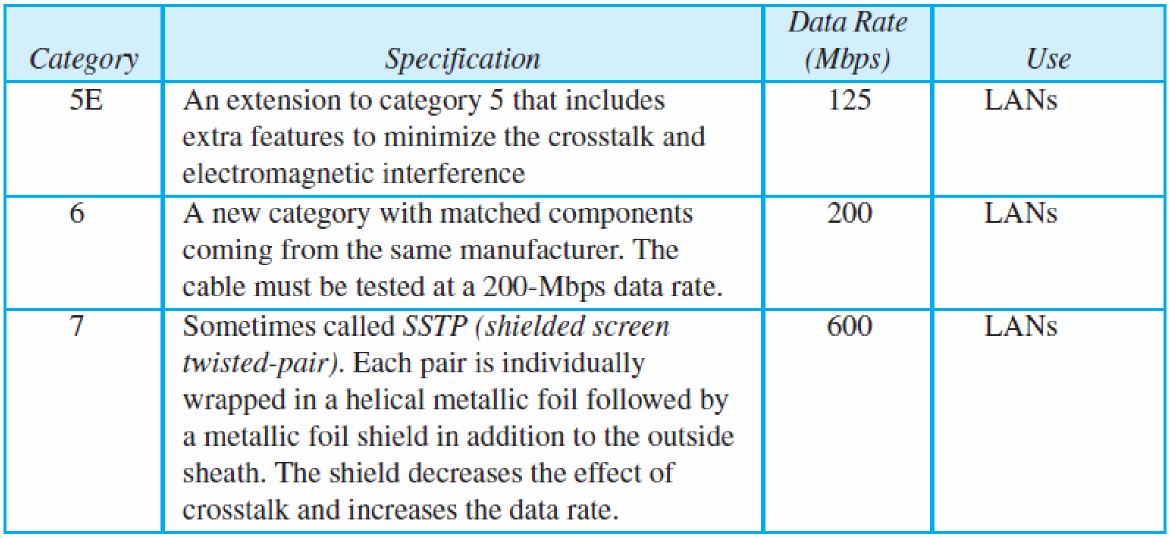
Connectors

Table 7.1 *Categories of unshielded twisted-pair cables*



Category	*Specification*	*Data Rate (Mbps)*	*Use*
1	Unshielded twisted-pair used in telephone	< 0.1	Telephone
2	Unshielded twisted-pair originally used in T lines	2	T-1 lines
3	Improved CAT 2 used in LANs	10	LANs
4	Improved CAT 3 used in Token Ring networks	20	LANs
5	Cable wire is normally 24 AWG with a jacket and outside sheath	100	LANs

Table 7.1 *Categories of unshielded twisted-pair cables (continued)*



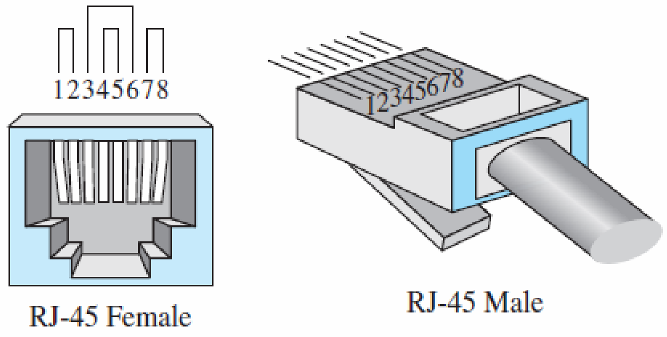
Category	*Specification*	*Data Rate (Mbps)*	*Use*
5E	An extension to category 5 that includes extra features to minimize the crosstalk and electromagnetic interference	125	LANs
6	A new category with matched components coming from the same manufacturer. The cable must be tested at a 200-Mbps data rate.	200	LANs
7	Sometimes called *SSTP (shielded screen twisted-pair)*. Each pair is individually wrapped in a helical metallic foil followed by a metallic foil shield in addition to the outside sheath. The shield decreases the effect of crosstalk and increases the data rate.	600	LANs

The most common UTP connector is **RJ45** (RJ stands for registered jack), as shown in Figure 7.5. The RJ45 is a keyed connector, meaning the connector can be inserted in only one way.

Performance

One way to measure the performance of twisted-pair cable is to compare attenuation versus frequency and distance. A twisted-pair cable can pass a wide range of frequencies. However, Figure 7.6 shows that with increasing frequency, the attenuation, measured in decibels per kilometer (dB/km), sharply increases with frequencies above 100 kHz. Note that ***gauge*** is a measure of the thickness of the wire.

Figure 7.5 *UTP connector*

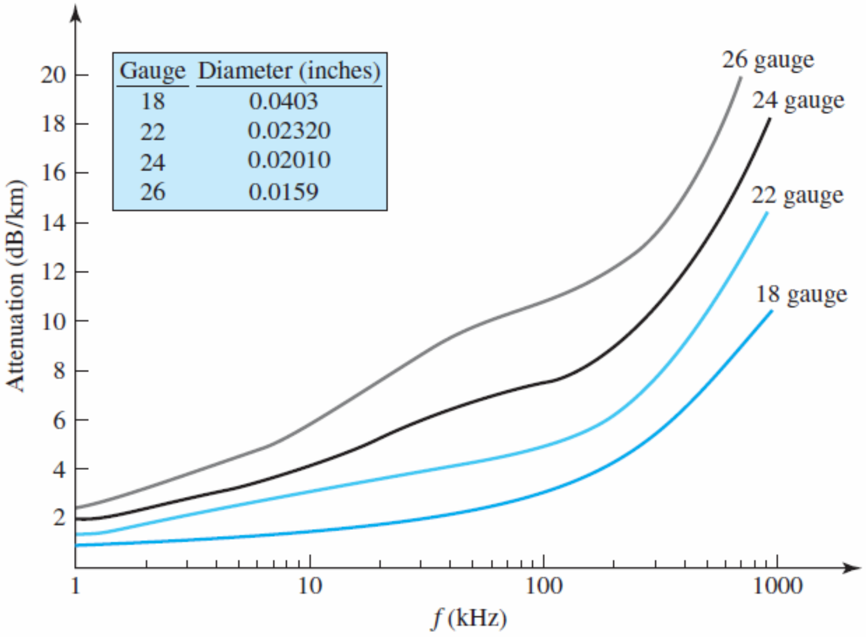


Applications

Twisted-pair cables are used in telephone lines to provide voice and data channels. The local loop—the line that connects subscribers to the central telephone office— commonly consists of unshielded twisted-pair cables. We discuss telephone networks in Chapter 14.

The DSL lines that are used by the telephone companies to provide high-data-rate connections also use the high-bandwidth capability of unshielded twisted-pair cables. Local-area networks, such as 10Base-T and 100Base-T, also use twisted-pair cables.

Figure 7.6 *UTP performance*



Gauge	Diameter (inches)
18	0.0403
22	0.02320
24	0.02010
26	0.0159

7.2.2 Coaxial Cable

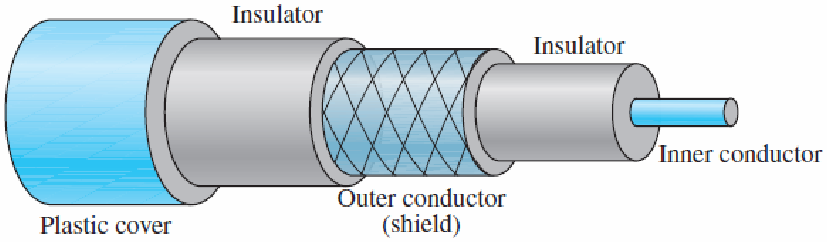
Coaxial cable (or *coax*) carries signals of higher frequency ranges than those in twistedpair cable, in part because the two media are constructed quite differently. Instead of having two wires, coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor

of metal foil, braid, or a combination of the two. The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit. This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover (see Figure 7.7).

Coaxial Cable Standards

Coaxial cables are categorized by their **Radio Government (RG)** ratings. Each RG number denotes a unique set of physical specifications, including the wire gauge of the inner conductor, the thickness and type of the inner insulator, the construction of the shield, and the size and type of the outer casing. Each cable defined by an RG rating is adapted for a specialized function, as shown in Table 7.2.

Figure 7.7 *Coaxial cable*



Coaxial Cable Connectors

To connect coaxial cable to devices, we need coaxial connectors. The most common type of connector used today is the **Bayonet Neill-Concelman (BNC)** connector. Figure 7.8 shows three popular types of these connectors: the BNC connector, the BNC T connector, and the BNC terminator. The BNC connector is used to connect the end of the cable to a device, such as a TV set. The BNC T connector is used in Ethernet networks to branch out to a connection to a computer or other device. The BNC terminator is used at the end of the cable to prevent the reflection of the signal.

Performance

As we did with twisted-pair cable, we can measure the performance of a coaxial cable. We notice in Figure 7.9 that the attenuation is much higher in coaxial cable than in twisted-pair cable. In other words, although coaxial cable has a much higher bandwidth, the signal weakens rapidly and requires the frequent use of repeaters.

Applications

Coaxial cable was widely used in analog telephone networks where a single coaxial network could carry 10,000 voice signals. Later it was used in digital telephone networks where a single coaxial cable could carry digital data up to 600 Mbps. However, coaxial cable in telephone networks has largely been replaced today with fiberoptic cable. Cable TV networks (see Chapter 14) also use coaxial cables. In the traditional cable TV network, the entire network used coaxial cable. Later, however, cable TV providers replaced most of the media with fiber-optic cable; hybrid networks use coaxial cable only at the network boundaries, near the consumer premises. Cable TV uses RG-59
coaxial cable.

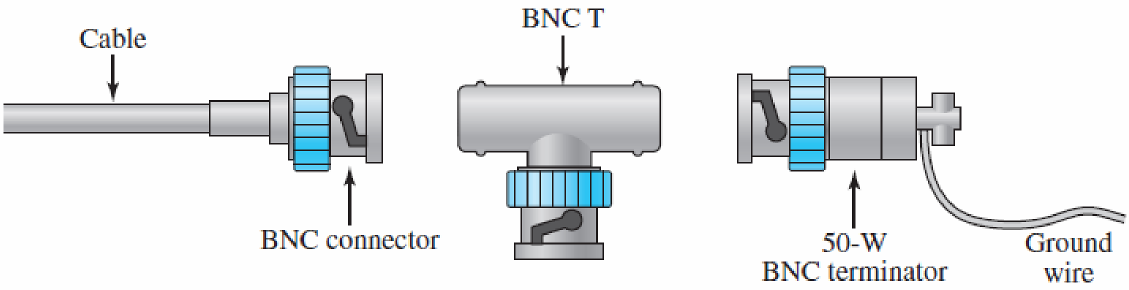
Table 7.2 *Categories of coaxial cables*

Category	*Impedance*	*Use*
RG-59	75 Ω	Cable TV
RG-58	50 Ω	Thin Ethernet
RG-11	50 Ω	Thick Ethernet

Another common application of coaxial cable is in traditional Ethernet LANs (see Chapter 13). Because of its high bandwidth, and consequently high data rate, coaxial cable was chosen for digital transmission in early Ethernet LANs. The 10Base-2, or Thin Ethernet, uses RG-58 coaxial cable with BNC connectors to transmit data at 10 Mbps with a range of 185 m. The 10Base5, or Thick Ethernet, uses RG-11 (thick coaxial cable) to transmit 10

Mbps with a range of 5000 m. Thick Ethernet has specialized connectors.

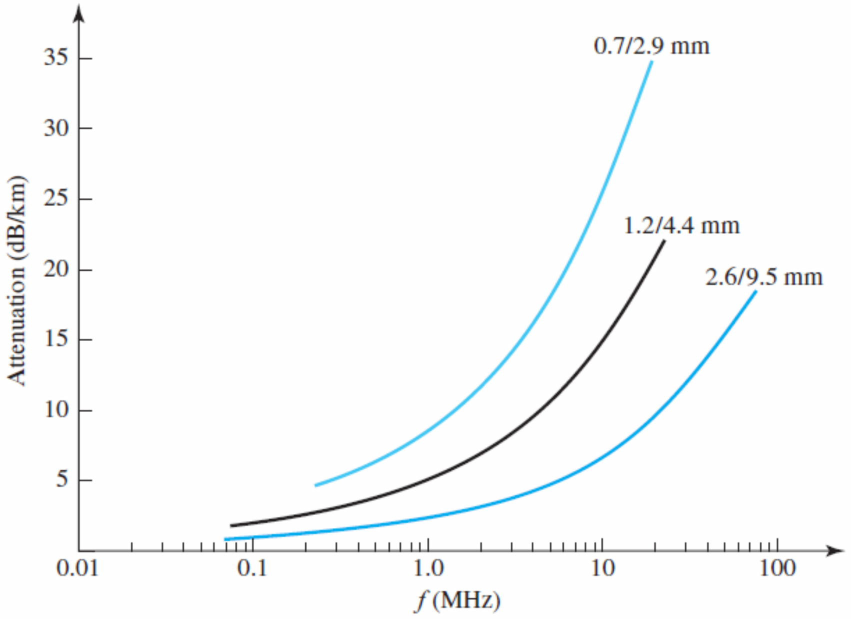
Figure 7.8 *BNC connectors*



7.2.3 Fiber-Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. To understand optical fiber, we first need to explore several aspects of the nature of light. Light travels in a straight line as long as it is moving through a single uniform substance. If a ray of light traveling through one substance suddenly enters another substance (of a different density), the ray changes direction. Figure 7.10 shows how a ray of light changes direction when going from a more dense to a less dense substance. As the figure shows, if the **angle of incidence** *I* (the angle the ray makes with the line perpendicular to the interface between the two substances) is less than the **critical**
angle, the ray **refracts** and moves closer to the surface. If the angle of incidence is equal to the critical angle, the light bends along the interface. If the angle is greater than the critical angle, the ray **reflects** (makes a turn) and travels again in the denser substance. Note that the critical angle is a property of the substance, and its value differs
from one substance to another.

Figure 7.9 *Coaxial cable performance*



Optical fibers use reflection to guide light through a channel. A glass or plastic **core** is surrounded by a **cladding** of less dense glass or plastic. The difference in density of the two materials must be such that a beam of light moving through the core is reflected off the cladding instead of being refracted into it. See Figure 7.11.

Propagation Modes

Current technology supports two modes (multimode and single mode) for propagating light along optical channels, each requiring fiber with different physical characteristics. Multimode can be implemented in two forms: step-index or graded-index (see Figure 7.12). ***Multimode*** Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure

of the core, as shown in Figure 7.13. In **multimode step-index fiber,** the density of the core remains constant from the center to the edges. A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change due to a lower density; this alters the angle of the beam's motion. The term *step-index* refers to the suddenness of this change, which contributes to the distortion of the signal as it passes through the fiber. A second type of fiber, called **multimode graded-index fiber,** decreases this distortion of the signal through the cable. The word *index* here refers to the index of refraction. As we saw above, the index of refraction is related to density. A gradedindex fiber, therefore, is one with varying densities. Density is highest at the center of he core and decreases gradually to its lowest at the edge. Figure 7.13 shows the impact of this variable density on the propagation of light beams.

Figure 7.10 *Bending of light ray*

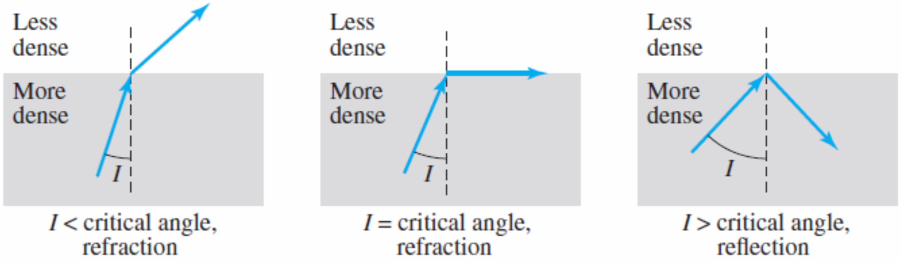


Figure 7.11 *Optical fiber*

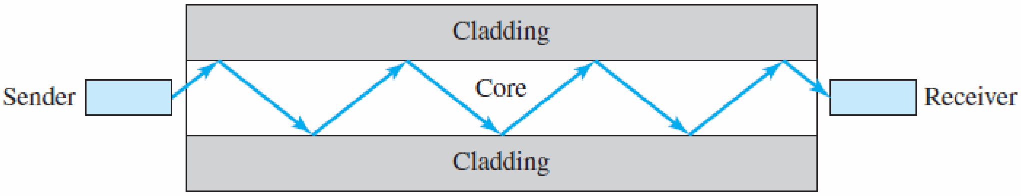
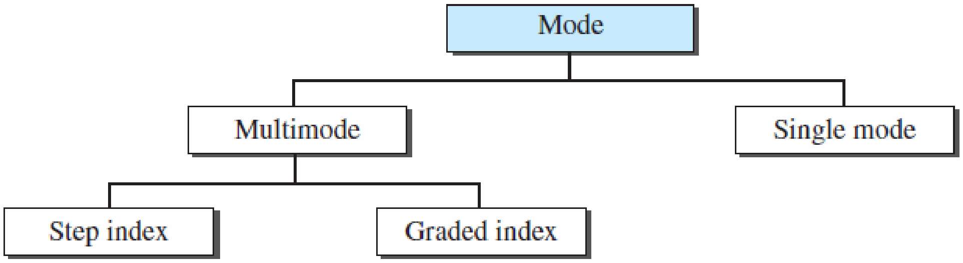


Figure 7.12 *Propagation modes*



Single-Mode

Single-mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal. The **single-mode fiber** itself is manufactured with a much smaller diameter than that of multimode fiber, and with substantially lower density (index of refraction). The decrease in density results in

a critical angle that is close enough to 90° to make the propagation of beams almost horizontal. In this case, propagation of different beams is almost identical, and delays are negligible. All the beams arrive at the destination "together" and can be recombined with little distortion to the signal (see Figure 7.13).

Fiber Sizes

Optical fibers are defined by the ratio of the diameter of their core to the diameter of their cladding, both expressed in micrometers. The common sizes are shown in Table 7.3. Note that the last size listed is for single-mode only. ***Cable Composition***

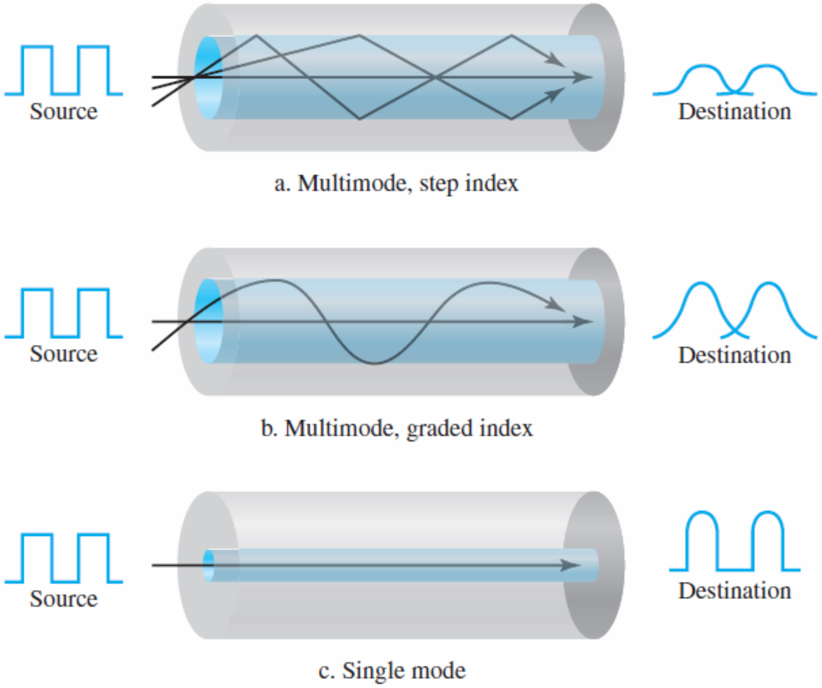
Figure 7.14 shows the composition of a typical fiber-optic cable. The outer jacket is made of either PVC or Teflon. Inside the jacket are Kevlar strands to strengthen the cable. Kevlar is a strong material used in the fabrication of bulletproof vests. Below the Kevlar is another plastic coating to cushion the fiber. The fiber is at the center of the cable, and it consists of cladding and core.

Fiber-Optic Cable Connectors

There are three types of connectors for fiber-optic cables, as shown in Figure 7.15. The **subscriber channel (SC) connector** is used for cable TV. It uses a push/pull locking system. The **straight-tip (ST) connector** is used for connecting cable to networking devices. It uses a bayonet locking system and is more reliable than SC. **MT-RJ** is a

connector that is the same size as RJ45.

Figure 7.13 *Modes*



a. Multimode, step index

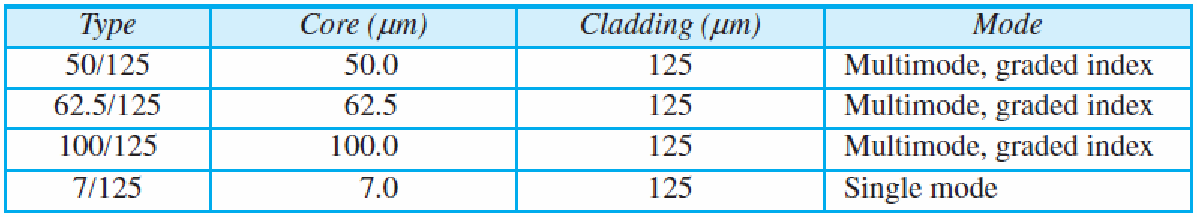
b. Multimode, graded index

c. Single mode

Performance

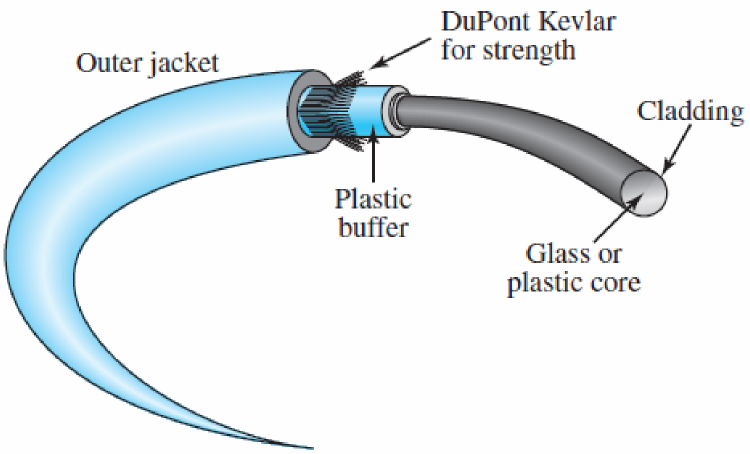
The plot of attenuation versus wavelength in Figure 7.16 shows a very interesting phenomenon in fiber-optic cable. Attenuation is flatter than in the case of twisted-pair cable and coaxial cable. The performance is such that we need fewer (actually onetenth as many) repeaters when we use fiber-optic cable.

Table 7.3 *Fiber types*



Type	*Core (μm)*	*Cladding (μm)*	*Mode*
50/125	50.0	125	Multimode, graded index
62.5/125	62.5	125	Multimode, graded index
100/125	100.0	125	Multimode, graded index
7/125	7.0	125	Single mode

Figure 7.14 *Fiber construction*



Fiber-Optic Cable Connectors

There are three types of connectors for fiber-optic cables, as shown in Figure 7.15. The **subscriber channel (SC) connector** is used for cable TV. It uses a push/pull locking system. The **straight-tip (ST) connector** is used for connecting cable to networking devices. It uses a bayonet locking system and is more reliable than SC. **MT-RJ** is a
connector that is the same size as RJ45.

Performance

The plot of attenuation versus wavelength in Figure 7.16 shows a very interesting phenomenon in fiber-optic cable. Attenuation is flatter than in the case of twisted-pair cable and coaxial cable. The performance is such that we need fewer (actually onetenth as many) repeaters when we use fiber-optic cable.

Applications

Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective. Today, with wavelength-division multiplexing (WDM), we can transfer data at a rate of 1600 Gbps. The SONET network that we discuss in Chapter 14 provides such a backbone. Some cable TV companies use a combination of optical fiber and coaxial cable, thus creating a hybrid network. Optical fiber provides the backbone structure while coaxial cable provides the connection to the user premises. This is a cost-effective configuration since the narrow bandwidth requirement at the user end does not justify the use of optical fiber. Local-area networks such as 100Base-FX network (Fast Ethernet) and 1000Base-X also use fiber-optic cable.

Figure 7.15 *Fiber-optic cable connectors*

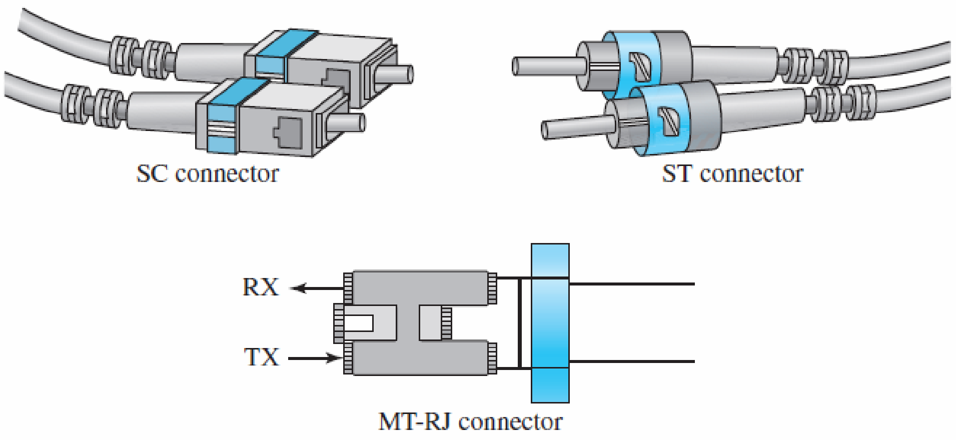
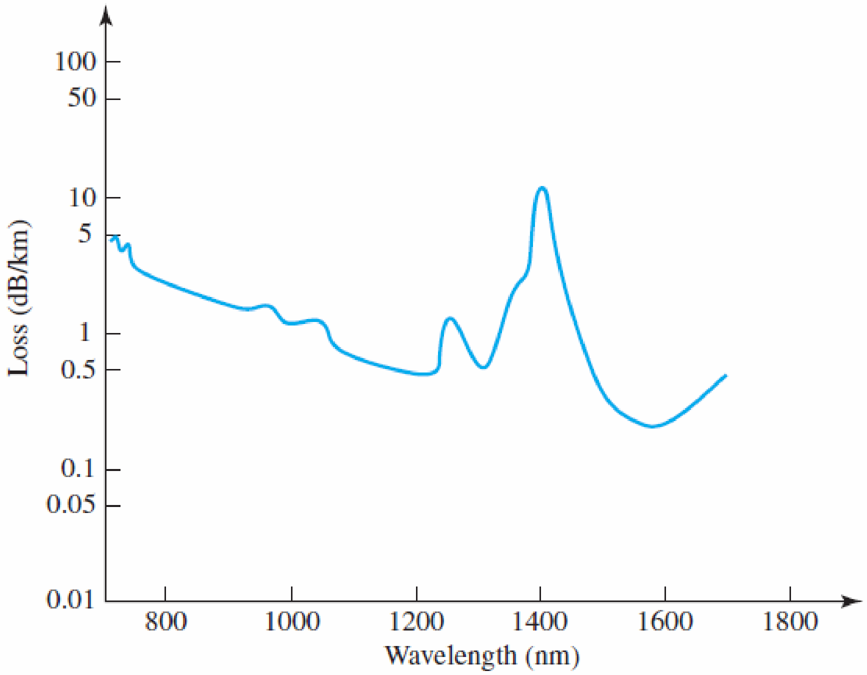


Figure 7.16 *Optical fiber performance*



Advantages and Disadvantages of Optical Fiber

Advantages

Fiber-optic cable has several advantages over metallic cable (twisted-pair or coaxial). ❑ **Higher bandwidth.** Fiber-optic cable can support dramatically higher bandwidths (and hence data rates) than either twisted-pair or coaxial cable. Currently, data rates and bandwidth utilization over fiber-optic cable are limited not by the medium but
by the signal generation and reception technology available.

❑ **Less signal attenuation.** Fiber-optic transmission distance is significantly greater than that of other guided media. A signal can run for 50 km without requiring regeneration. We need repeaters every 5 km for coaxial or twisted-pair cable.

- **Immunity to electromagnetic interference.** Electromagnetic noise cannot affect fiber-optic cables.
- **Resistance to corrosive materials.** Glass is more resistant to corrosive materials than copper.
- **Light weight.** Fiber-optic cables are much lighter than copper cables.
- **Greater immunity to tapping.** Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antenna effects that can easily be tapped.

Disadvantages

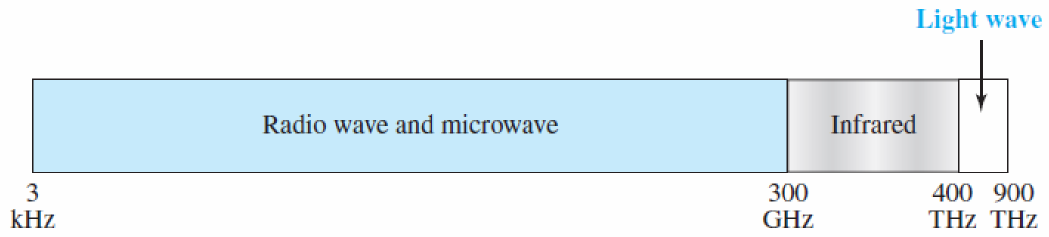
There are some disadvantages in the use of optical fiber.

- **Installation and maintenance.** Fiber-optic cable is a relatively new technology. Its installation and maintenance require expertise that is not yet available everywhere.
- **Unidirectional light propagation.** Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.
- **Cost.** The cable and the interfaces are relatively more expensive than those of other guided media. If the demand for bandwidth is not high, often the use of optical fiber cannot be justified.

7.3 UNGUIDED MEDIA: WIRELESS

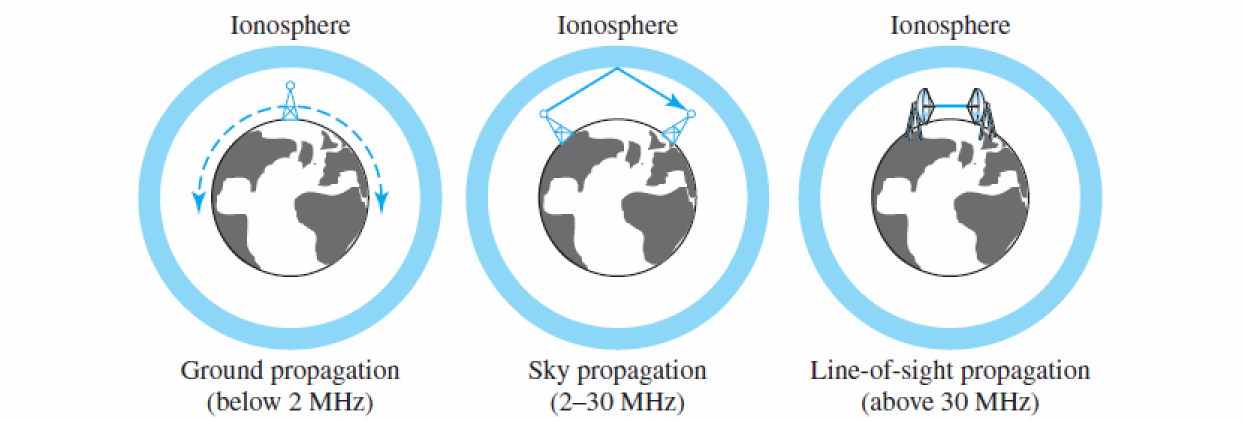
Unguided medium transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as ***wireless communication.*** Signals are normally broadcast through free space and thus are available to anyone who has a device capable of receiving them. Figure 7.17 shows the part of the electromagnetic spectrum, ranging from 3 kHz to 900 THz, used for wireless communication. Unguided signals can travel from the source to the destination in several ways: ground propagation, sky propagation, and line-of-sight propagation, as shown in Figure 7.18. In **ground propagation,** radio waves travel through the lowest portion of the atmosphere, hugging the earth. These low-frequency signals emanate in all directions from the transmitting antenna and follow the curvature of the planet. Distance depends on the amount of power in the signal: The greater the power, the greater the distance. In **sky propagation,** higher-frequency radio waves radiate upward into the ionosphere (the layer of atmosphere where particles exist as ions) where they are reflected back to earth. This type of transmission allows for greater

Figure 7.17 *Electromagnetic spectrum for wireless communication*



distances with lower output power. In **line-of-sight**

Figure 7.18 *Propagation methods*



propagation, very high-frequency signals are transmitted in straight lines directly from antenna to antenna. Antennas must be directional, facing each other, and either tall enough or close enough together not to be affected by the curvature of the earth. Line-ofsight propagation is tricky because radio transmissions cannot be completely focused. The section of the electromagnetic spectrum defined as radio waves and microwaves is divided into eight ranges, called *bands,* each regulated by government authorities.

These bands are rated from *very low frequency* (VLF) to *extremely high frequency* (EHF). Table 7.4 lists these bands, their ranges, propagation methods, and some applications. We can divide wireless transmission into three broad groups: radio waves, microwaves, and infrared waves.

Table 7.4 *Bands*

Band	*Range*	*Propagation*	*Application*
very low frequency (VLF)	3–30 kHz	Ground	Long-range radio navigation
low frequency (LF)	30–300 kHz	Ground	Radio beacons and navigational locators

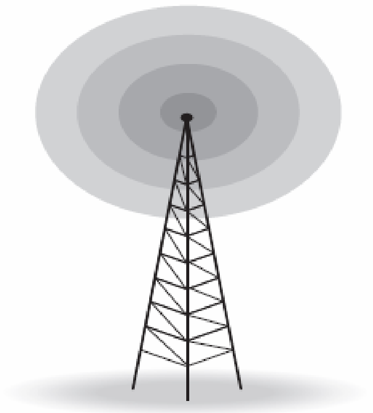
Table 7.4 *Bands (continued)*

Band	*Range*	*Propagation*	*Application*
middle frequency (MF)	300 kHz–3 MHz	Sky	AM radio
high frequency (HF)	3–30 MHz	Sky	Citizens band (CB), ship/aircraft
very high frequency (VHF)	30–300 MHz	Sky and line-of-sight	VHF TV, FM radio
ultrahigh frequency (UHF)	300 MHz–3 GHz	Line-of-sight	UHF TV, cellular phones, paging, satellite
superhigh frequency (SF)	3–30 GHz	Line-of-sight	Satellite
extremely high frequency (EHF)	30–300 GHz	Line-of-sight	Radar, satellite

7.3.1 Radio Waves

Although there is no clear-cut demarcation between radio waves and microwaves, electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called **radio waves;** waves ranging in frequencies between 1 and 300 GHz are called **microwaves.** However, the behavior of the waves, rather than the frequencies, is a better criterion for classification. Radio waves, for the most part, are omnidirectional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves that can be received by any receiving antenna. The omnidirectional property has a disadvantage, too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band. Radio waves, particularly those waves that propagate in the sky mode, can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio. Radio waves, particularly those of low and medium frequencies, can penetrate walls. This characteristic can be both an advantage and a disadvantage. It is an advantage because, for example, an AM radio can receive signals inside a building. It is a disadvantage because we cannot isolate a communication to just inside or outside a building. The radio wave band is relatively narrow, just under 1 GHz, compared to the microwave band. When this band is divided into subbands, the subbands are also narrow, leading to a low data rate for digital communications. Almost the entire band is regulated by authorities (e.g., the FCC in the United States). Using any part of the band requires permission from the authorities.

Figure 7.19 *Omnidirectional antenna*



Omnidirectional Antenna

Radio waves use **omnidirectional antennas** that send out signals in all directions. Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas. Figure 7.19 shows an omnidirectional antenna.

Applications

The omnidirectional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

7.3.2 Microwaves

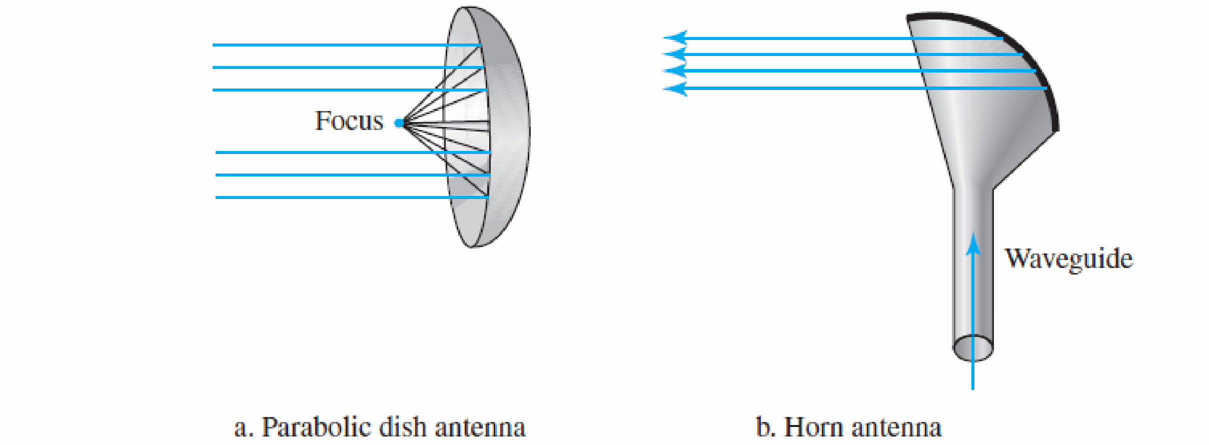
Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves. Microwaves are unidirectional. When an antenna transmits microwaves, they can be narrowly focused. This means that the sending and receiving antennas need to be aligned. The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas. The following describes some characteristics of microwave propagation:

- Microwave propagation is line-of-sight. Since the towers with the mounted antennas need to be in direct sight of each other, towers that are far apart need to be very tall. The curvature of the earth as well as other blocking obstacles do not allow two short towers to communicate by using microwaves. Repeaters are often needed for longdistance communication.
- Very high-frequency microwaves cannot penetrate walls. This characteristic can be a disadvantage if receivers are inside buildings.

❑ The microwave band is relatively wide, almost 299 GHz. Therefore wider subbands can be assigned, and a high data rate is possible.

❑ Use of certain portions of the band requires permission from authorities.

Figure 7.20 *Unidirectional antennas*



a. Parabolic dish antenna b. Horn antenna

Unidirectional Antenna

Microwaves need **unidirectional antennas** that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the horn (see Figure 7.20). A **parabolic dish antenna** is based on the geometry of a parabola: Every line parallel to the line of symmetry (line of sight) reflects off the curve at angles such that all the lines intersect in a common point called the focus. The parabolic dish works as a

funnel, catching a wide range of waves and directing them to a common point. In this way, more of the signal is recovered than would be possible with a single-point receiver. Outgoing transmissions are broadcast through a horn aimed at the dish. The microwaves hit the dish and are deflected outward in a reversal of the receipt path.

A **horn antenna** looks like a gigantic scoop. Outgoing transmissions are broadcast up a stem (resembling a handle) and deflected outward in a series of narrow parallel beams by the curved head. Received transmissions are collected by the scooped shape of the horn, in a manner similar to the parabolic dish, and are deflected down into the stem.

Applications

Microwaves, due to their unidirectional properties, are very useful when unicast (oneto- one) communication is needed between the sender and the receiver. They are used in cellular phones (Chapter 16), satellite networks (Chapter 16), and wireless LANs (Chapter 15).

7.3.3 Infrared

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range communication. Infrared waves, having high frequencies, cannot penetrate walls. This advantageous characteristic prevents interference between one system and another; a short-range communication system in one

room cannot be affected by another system in the next room. When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors. However, this same characteristic makes infrared signals useless for long-range communication. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication

Applications The infrared band, almost 400 THz, has an excellent potential for data transmission. Such a wide bandwidth can be used to transmit digital data with a very high data rate. The *Infrared Data Association* (IrDA), an association for sponsoring the use of infrared waves, has established standards for using these signals for communication between devices such as keyboards, mice, PCs, and printers. For example, some manufacturers

provide a special port called the **IrDA port** that allows a wireless keyboard to communicate with a PC. The standard originally defined a data rate of 75 kbps for a distance up to 8 m. The recent standard defines a data rate of 4 Mbps.
Infrared signals defined by IrDA transmit through line of sight; the IrDA port on the keyboard needs to point to the PC for transmission to occur.

Data-Link Layer

Contents

❑ The first section introduces the data-link layer. It starts with defining the concept of links and nodes. The section then lists and briefly describes the services provided by the data-link layer. It next defines two categories of links: point-to-point and broadcast links. The section finally defines two sublayers at the data-link layer

❑ The second section discusses link-layer addressing. It first explains the rationale behind the existence of an addressing mechanism at the data-link layer. It then describes three types of link-layer addresses to be found in some link-layer protocols.
The section discusses the Address Resolution Protocol (ARP), which maps the addresses at the network layer to addresses at the data-link layer. This protocol helps a packet at the network layer find the link-layer address of the next node for delivery of the frame that encapsulates the packet. To show how the network layer helps us to find the data-link-layer addresses, a long example is included in this section that shows what happens at each node when a packet is travelling through the Internet.

9.1 INTRODUCTION

The Internet is a combination of networks glued together by connecting devices (routers or switches). If a packet is to travel from a host to another host, it needs to pass through these networks. Figure 9.1 shows the scenario, but we are now interested in communication at the data-link layer.

1. Communication at the data-link layer is made up of five separate logical connections between the data-link layers in the path
2. The data-link layer at Alice’s computer communicates with the data-link layer at router R2.
3. The data-link layer at router R2 communicates with the data-link layer at router R4, and so on.
4. Finally, the data-link layer at router R7 communicates with the data-link layer at Bob’s computer.
5. Only one data-link layer is involved at the source or the destination, but two data-link layers are involved at each router.
6. The reason is that Alice’s and Bob’s computers are each connected to a single network, but each router takes input from one network and sends output to another network.

Note that although switches are also involved in the data-link-layer communication, for simplicity we have not shown them in the figure
Let us see the difference between router and switch:

Router	Switch
The main objective of router is to connect various networks simultaneously.	While the main objective of switch is to connect various devices simultaneously.
It works in network layer.	While it works in data link layer.
Router is used by LAN as well as MAN.	While switch is used by only LAN.

Through the router, data is sent in the form of packets.	While through switch data is sent in the form of frame.
There is less collision taking place in the router.	While there is no collision taking place in full duplex switch.
The types of routing are: Adaptive and Non-adaptive routing.	The types of switching are: Circuit, Packet,and Message Switching.

Let’s see the difference between LAN and WAN:

S.NO	LAN	WAN
1.	LAN stands for Local Area Network.	Whereas WAN stands for Wide Area Network.
2.	LAN’s ownership is private.	But WAN’s ownership can be private or public.
3.	The speed of LAN is high(more than WAN).	While the speed of WAN is slower than LAN.
4.	The propagation delay is short in LAN.	Whereas the propagation delay in WAN is long(longer than LAN).
6.	There is more fault tolerance in LAN. LANs tend to have fewer problems associated with them, as there are smaller number of systems to deal with.	While there is less fault tolerance in WAN as they consist of large number of systems..
7.	LAN’s design and maintenance is easy.	While it’s design and maintenance is difficult than WAN.
8.	LAN covers small area i.e. within the building.	While WAN covers large geographical area.
9.	LAN operates on the principle of broadcasting.	While WAN works on the principle of point to point.
10.	Transmission medium used in LAN is co-axial or UTP cable.	Whereas WAN uses PSTN or satellite link as a transmission or communication medium.
11.	LAN has a higher data transfer rate. LAN is a computer network that covers a small geographic area, like a home, office, or group of buildings, while WAN is a computer network that covers a broader area. The speed of LAN is high whereas the speed of WAN is slower	WAN has a lower data transfer rate as compared to LAN.

	than LAN	
12.	LANs technologies used like ethernet and token.	WANs technologies used like Frame Relay and X.25 for connectivity for longer distances.
13.	LANs technologies is data transfer rate is **1000mbps**	WANs technologies data transfer rate 150mbps
14.	LANs is cheaply compared to WAN	WAN is costly compared to LAN.

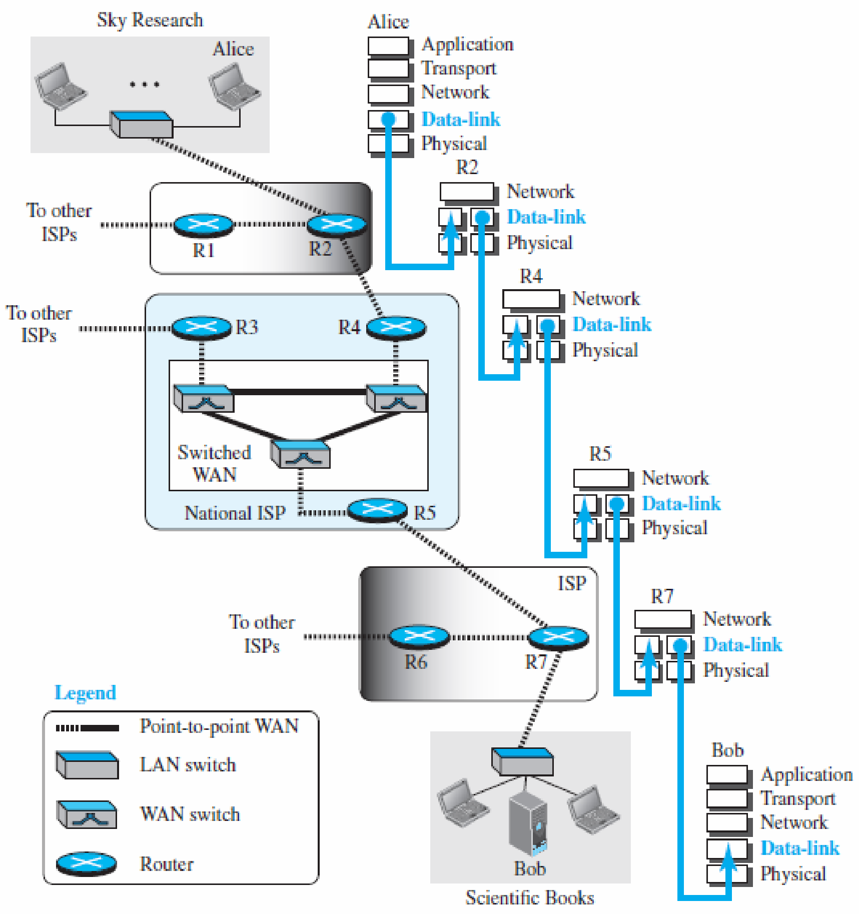
9.1.1 Nodes and Links

1. Communication at the data-link layer is node-to-node.
2. A data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point.
3. Theses LANs and WANs are connected by routers.
4. It is customary to refer to the two end hosts and the routers as ***nodes*** and the networks in between as ***links*.**
5. Figure 9.2 is a simple representation of links and nodes when the path of the data unit is only six nodes.
6. The first node is the source host; the last node is the destination host.
7. The other four nodes are four routers. The first, the third, and the fifth links represent the three LANs; the second and the fourth links represent the two WANs.

9.1.2 Services

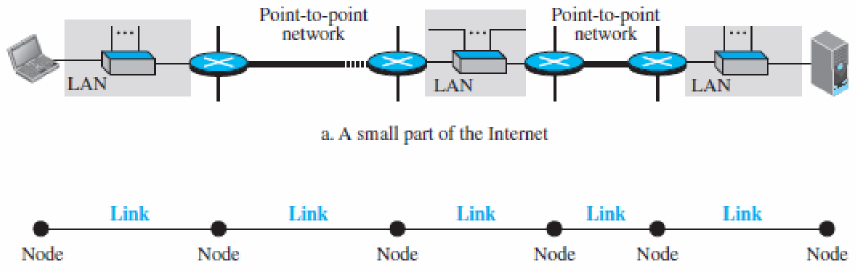
1. The data-link layer is located between the physical and the network layers. The datalink layer provides services to the network layer; it receives services from the physical layer.
2. The duty scope of the data-link layer is node-to-node. When a packet is travelling in the Internet, the data-link layer of a node (host or router) is responsible for delivering a datagram to the next node in the path.
3. For this purpose, the data-link layer of the sending node needs to encapsulate the datagram received from the network in a frame,and the data-link layer of the receiving node needs to decapsulate the datagram from the frame. 4. In other words, the data-link layer of the source host needs only to encapsulate, the data-link layer of the destination host needs to decapsulate, but each intermediate node needs to both encapsulate and decapsulate.
5. Each link may be using a different protocol with a different frame format.
6. Even if one link and the next are using the same protocol, encapsulation and decapsulation are needed because the link-layer addresses are normally different. Assume a person needs to travel from her home to her friend’s home in another city. The traveller can use three transportation tools. She can take a taxi to go to the train station in her own city, then travel on the train from her own city to the city where her friend lives, and finally reach her friend’s home using another taxi.
8. Here we have a source node, a destination node, and two intermediate nodes. The traveller needs to get into the taxi at the source node, get out of the taxi and get into the train at the first intermediate node (train station in the city

Figure 9.1 *Communication at the data-link layer*



1.

Figure 9.2 *Nodes and Links*



a. A small part of the Internet

b. Nodes and links

where she lives), get out of the train and get into another taxi at the second intermediate node (train station in the city where her friend lives), and finally get out of the taxi when she arrives at her destination.
9. A kind of encapsulation occurs at the source node, encapsulation and decapsulation occur at the intermediate nodes, and decapsulation occurs at the destination node. Our traveler is the same, but she uses three transporting tools to reach the destination.

Figure 9.3 *A communication with only three nodes*

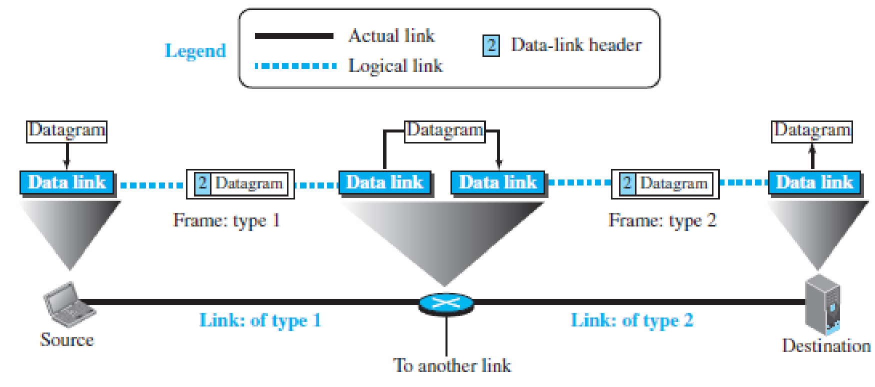


Figure 9.3 shows the encapsulation and decapsulation at the data-link layer. For simplicity, we have assumed that we have only one router between the source and destination. The datagram received by the data-link layer of the source host is encapsulated in a frame. The frame is logically transported from the source host to the router. The frame is decapsulated at the data-link layer of the router and encapsulated at another frame. The new frame is logically transported from the router to the destination host.

.

Framing

Definitely, the first service provided by the data-link layer is **framing**. The data-link layer at each node needs to encapsulate the datagram (packet received from the network layer) in a **frame** before sending it to the next node. The node also needs to decapsulate the datagram from the frame received on the logical channel. Although we have shown only a header for a frame, we will see in future chapters that a frame may have both a header and a trailer. Different data-link layers have different formats for framing.

Flow Control

Whenever we have a producer and a consumer, we need to think about flow control. If the producer produces items that cannot be consumed, accumulation of items occurs. The sending data-link layer at the end of a link is a producer of frames; the receiving data-link layer at the other end of a link is a consumer. If the rate of produced frames is higher than the rate of consumed frames, frames at the receiving end need to be buffered while waiting to be consumed (processed). Definitely, we cannot have an unlimited buffer size at the receiving side. We have two choices. The first choice is to let the receiving data-link layer drop the frames if its buffer is full. The second choice is to let the receiving data-link layer send a feedback to the sending data-link layer to ask it to stop or slow down. Different data-link-layer protocols use different strategies for flow control. Since flow control also occurs at the transport layer, with a higher degree of importance,

Error Control

At the sending node, a frame in a data-link layer needs to be changed to bits, transformed to electromagnetic signals, and transmitted through the transmission media. At the receiving node, electromagnetic signals are received, transformed to bits, and put together to create a frame. Since electromagnetic signals are susceptible to error, a frame is susceptible to error. The error needs first to be detected. After detection, it needs to be either corrected at the receiver node or discarded and retransmitted by the sending node. Since error detection and correction is an issue in every layer (node-tonode or host-to-host)

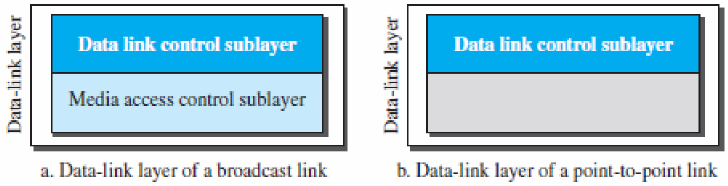
Congestion Control

Although a link may be congested with frames, which may result in frame loss, most data-link-layer protocols do not directly use a congestion control to alleviate congestion, although some wide-area networks do. In general, congestion control is considered an issue in the network layer or the transport layer because of its end-to-end nature. We will discuss congestion control in the network layer and the transport layer in later chapters.

9.1.3 Two Categories of Links

Although two nodes are physically connected by a transmission medium such as cable or air, we need to remember that the data-link layer controls how the medium is used. We can have a data-link layer that uses the whole capacity of the medium; we can also have a data-link layer that uses only part of the capacity of the link. In other words, we can have a *point-to-point link* or a *broadcast link*. In a point-to-point link, the link is dedicated to the two devices; in a broadcast link, the link is shared between several pairs of devices. For example, when two friends use the raditional home phones to chat, they are using a point-to-point link; when the same two friends use their cellular phones, they are using a br oadcast link (the air is shared among many cell phone users).

Figure 9.4 *Dividing the data-link layer into two sublayers*


Data-link layer
Data link control sublayer
Media access control sublayer
Data-link layer
Data link control sublayer

a. Data-link layer of a broadcast link

b. Data-link layer of a point-to-point link

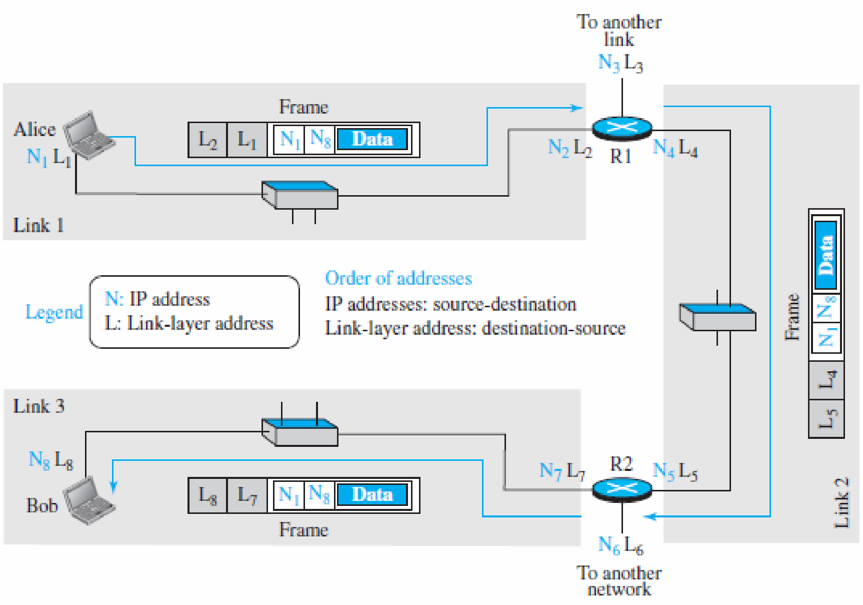
9.1.4 Two Sublayers

To better understand the functionality of and the services provided by the link layer, we can divide the data-link layer into two sublayers: **data link control (DLC)** and **media access control (MAC).** This is not unusual because, as we will see in later chapters, LAN protocols actually use the same strategy. The data link control sublayer deals with all issues common to both point-to-point and broadcast links; the media access control sublayer deals only with issues specific to broadcast links. In other words, we separate these two types of links at the data-link layer, as shown in Figure 9.4.

9.2 LINK-LAYER ADDRESSING

The next issue we need to discuss about the data-link layer is the link-layer addresses. In Chapter 18, we will discuss IP addresses as the identifiers at the network layer that define the exact points in the Internet where the source and destination hosts are connected. However, in a connectionless internetwork such as the Internet we cannot make a datagram reach its destination using only IP addresses. The reason is that each datagram in the Internet, from the same source host to the same destination host, may take a different path. The source and destination IP addresses define the two ends but cannot define which links the datagram should pass through. We need to remember that the IP addresses in a datagram should not be changed. If the destination IP address in a datagram changes, the packet never reaches its destination; if the source IP address in a datagram changes, the destination host or a router can never communicate with the source if a response needs to be sent back or an error needs to be reported back to the source (see ICMP in Chapter 19) The above discussion shows that we need another addressing mechanism in a connectionless internetwork: the link-layer addresses of the two nodes. A *link-layer address* is sometimes called a *link address*, sometimes a *physical address*, and sometimes a *MAC address*. Since a link is controlled at the data-link layer, the addresses need to belong to the data-link layer. When a datagr m passes from the network layer to the data-link layer, the datagram will be encapsulated in a frame and two data-link addresses are added to the frame header. These two addresses are changed every time the frame moves from one link to another. Figure 9.5 demonstrates the concept in a small internet. In the internet in Figure 9.5, we have three links and two routers. We also have

Figure 9.5 *IP addresses and link-layer addresses in a small internet*



shown only two hosts: Alice (source) and Bob (destination). For each host, we have shown two addresses, the IP addresses (N) and the link-layer addresses (L). Note that a router has as many pairs of addresses as the number of links the router is connected to. We have shown three frames, one in each link. Each frame carries the same datagram with the same source and destination addresses (**N1** and **N8**), but the link-layer addresses of the frame change from link to link. In link 1, the link-layer addresses are L_1 and L_2. In link 2, they are L_4 and L_5. In link 3, they are L_7 and L_8. Note that the IP addresses and the link-layer addresses are not in the same order. For IP addresses, the source address comes before the destination address; for link-layer addresses, the destination address comes before the source. The datagrams and frames are designed in this way, and we follow the design. We may raise several questions:

❑ If the IP address of a router does not appear in any datagram sent from a source to a destination, why do we need to assign IP addresses to routers? The answer is that in some protocols a router may act as a sender or receiver of a datagram. For example, in routing protocols we will discuss in Chapters 20 and 21, a router is a sender or a receiver of a message. The communications in these protocols are between routers.

❑ Why do we need more than one IP address in a router, one for each interface? The answer is that an interface is a connection of a router to a link. We will see that an IP address defines a point in the Internet at which a device is connected. A router with *n* interfaces is connected to the Internet at *n* points. This is the situation of a house at the corner of a street with two gates; each gate has the address related to the corresponding street.

❑ How are the source and destination IP addresses in a packet determined? The answer is that the host should know its own IP address, which becomes the source IP address in the packet. As we will discuss in Chapter 26, the application layer uses the services of DNS to find the destination address of the packet and passes it to the network layer to be inserted in the packet.

❑ How are the source and destination link-layer addresses determined for each link? Again, each hop (router or host) should know its own link-layer address, The destination link-layer address is determined by using the Address Resolution Protocol, which we discuss shortly.

❑ What is the size of link-layer addresses? The answer is that it depends on the protocol used by the link. Although we have only one IP protocol for the whole Internet, we may be using different data-link protocols in different links. This means that we can define the size of the address when we discuss different link-layer protocols.

9.2.1 Three Types of addresses

Some link-layer protocols define three types of addresses: unicast, multicast, and broadcast.

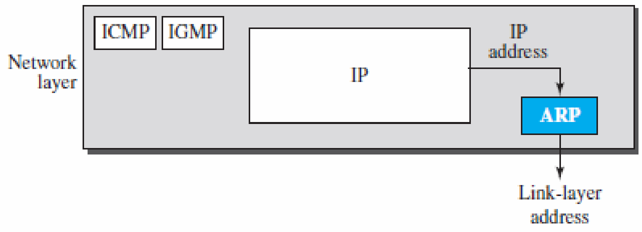
Unicast Address

Each host or each interface of a router is assigned a unicast address. Unicasting means one-to-one communication. A frame with a unicast address destination is destined only for one entity in the link.

Example 9.1

As we will see in Chapter 13, the unicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons; for example, the following is a link-layer address of a computer.

Figure 9.6 *Position of ARP in TCP/IP protocol suite*



Multicast Address

Some link-layer protocols define multicast addresses. Multicasting means one-to-many communication. However, the jurisdiction is local (inside the link).

Example 9.2

As we will see in Chapter 13, the multicast link-layer addresses in the most common LAN, Ethernet, are 48 bits (six bytes) that are presented as 12 hexadecimal digits separated by colons. The second digit, however, needs to be an even number in hexadecimal. The following shows a multicast address:

Broadcast Address

Some link-layer protocols define a broadcast address. Broadcasting means one-to-all communication. A frame with a destination broadcast address is sent to all entities in the link.

Example 9.3

As we will see in Chapter 13, the broadcast link-layer addresses in the most common LAN, Ethernet, are 48 bits, all 1s, that are presented as 12 hexadecimal digits separated by colons. The following shows a broadcast address:

9.2.2 Address Resolution Protocol (ARP)

1. Anytime a node has an IP datagram to send to another node in a link, it has the IP address of the receiving node. The source host knows the IP address of the default router.

2. Each router except the last one in the path gets the IP address of the next router by using its forwarding table. The last router knows the IP address of the destination host. however, the IP address of the next node is not helpful in moving a frame through a link; we need the link-layer address of the next node. This is the time when the **Address Resolution Protocol (ARP)** becomes helpful.

3. The ARP protocol is one of the auxiliary protocols defined in the network layer, as shown in Figure 9.6. It belongs to the network layer, but we discuss it in this chapter because it maps an IP address to a logical-link address. ARP accepts an IP address from the IP protocol, maps the address to the corresponding link-layer address, and passes it to the data-link layer.

4. Anytime a host or a router needs to find the link-layer address of another host or router in its network, it sends an ARP request packet. The packet includes the link-layer and IP addresses of the sender and the IP address of the receiver.

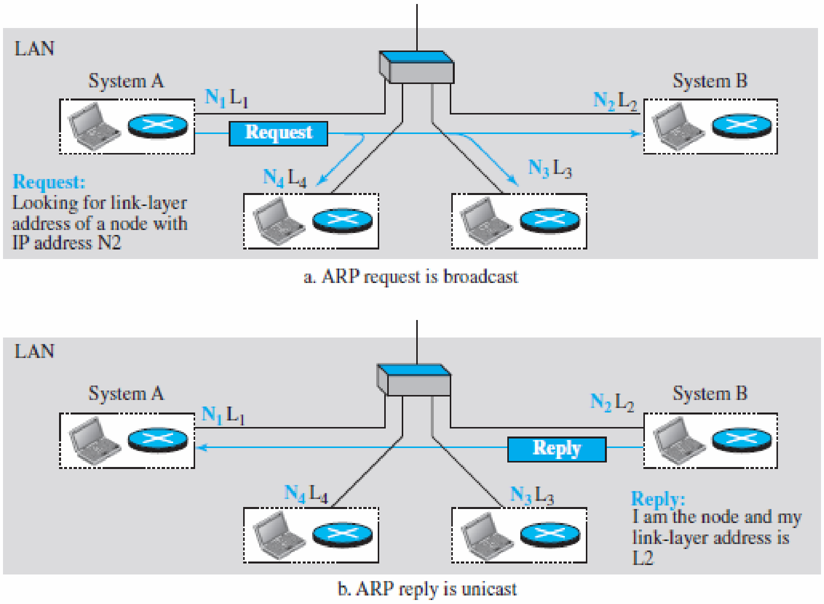
5. Because the sender does not know the link-layer address of the receiver, the query is broadcast over the link using the link-layer broadcast address, which we discuss for each protocol later (see Figure 9.7).

6. Every host or router on the network receives and processes the ARP request packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and link-layer addresses.

7. The packet is unicast directly to the node that sent the request packet. In Figure 9.7a, the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address N2. System A needs to pass the packet to its data-link layer for the actual delivery, but it does not know the physical

address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of N2.

Figure 9.7 *ARP operation*



Request:
Looking for link-layer
address of a node with
IP address N2

a. ARP request is broadcast

Reply:
I am the node and my
link-layer address is
L2

b. ARP reply is unicast

This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure 9.7b. System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination using the physical address it received.

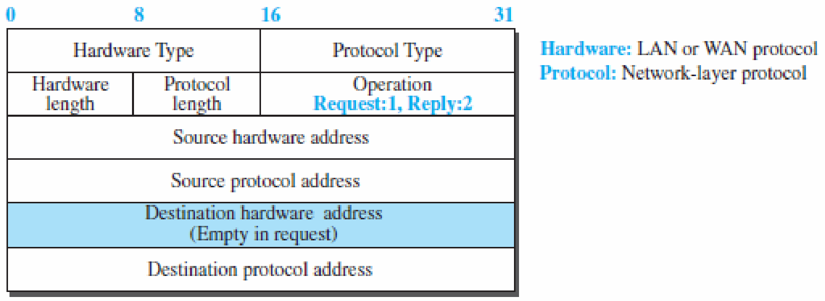
Caching

A question that is often asked is this: If system A can broadcast a frame to find the linklayer address of system B, why can't system A send the datagram for system B using a broadcast frame? In other words, instead of sending one broadcast frame (ARP request), one unicast frame (ARP response), and another unicast frame (for sending the
datagram), system A can encapsulate the datagram and send it to the network. System B receives it and keep it; other systems discard it. To answer the question, we need to think about the efficiency. It is probable that system A has more than one datagram to send to system B in a short period of time. For example, if system B is supposed to receive a long e-mail or a long file, the data do not fit in one datagram. Let us assume that there are 20 systems connected to the network (link): system A, system B, and 18 other systems. We also assume that system A has 10 datagrams to send to system B in one second.

a. Without using ARP, system A needs to send 10 broadcast frames. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the datagram and pass it to their network-layer to find out the datagrams do not belong to them.This means processing and discarding 180 broadcast frames.

b. Using ARP, system A needs to send only one broadcast frame. Each of the 18 other systems need to receive the frames, decapsulate the frames, remove the ARP message and pass the message to their ARP protocol to find that the frame must be discarded. This means processing and discarding only 18 (instead of 180) broadcast frames. After system B responds with its own data-link address, system A can store the link-layer address in its cache memory. The rest of the nine frames are only unicast. Since processing broadcast frames is expensive (time consuming), the first method is preferable.

Figure 9.8 *ARP packet*



Packet Format

Figure 9.8 shows the format of an ARP packet. The names of the fields are selfexplanatory. The *hardware type* field defines the type of the link-layer protocol; Ethernet is given the type 1. The *protocol type* field defines the network-layer protocol: IPv4 protocol is $(0800)_{16}$. The source hardware and source protocol addresses are variable-length

fields defining the link-layer and network-layer addresses of the sender. The destination hardware address and destination protocol address fields define the receiver link-layer and network-layer addresses. An ARP packet is encapsulated directly into a data-link frame. The frame needs to have a field to show that the payload belongs to the ARP and not to the network-layer datagram.

Example 9.4

A host with IP address N1 and MAC address **L1** has a packet to send to another host with IP address N2 and physical address **L2** (which is unknown to the first host). The two hosts are on the same network. Figure 9.9 shows the ARP request and response messages.

9.2.3 An Example of Communication

To show how communication is done at the data-link layer and how link-layer addresses are found, let us go through a simple example. Assume Alice needs to send a datagram to Bob, who is three nodes away in the Internet. How Alice finds the network-layer address of Bob is what we discover in Chapter 26 when we discuss DNS. For the

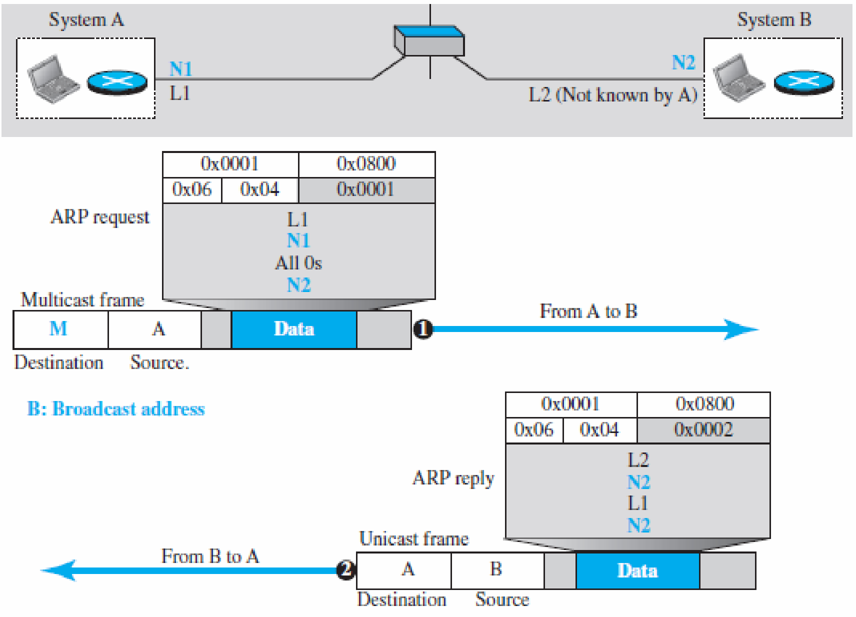
moment, assume that Alice knows the network-layer (IP) address of Bob. In other words, Alice's host is given the data to be sent, the IP address of Bob, and the IP address of Alice's host (each host needs to know its IP address). Figure 9.10 shows the part of the internet for our example

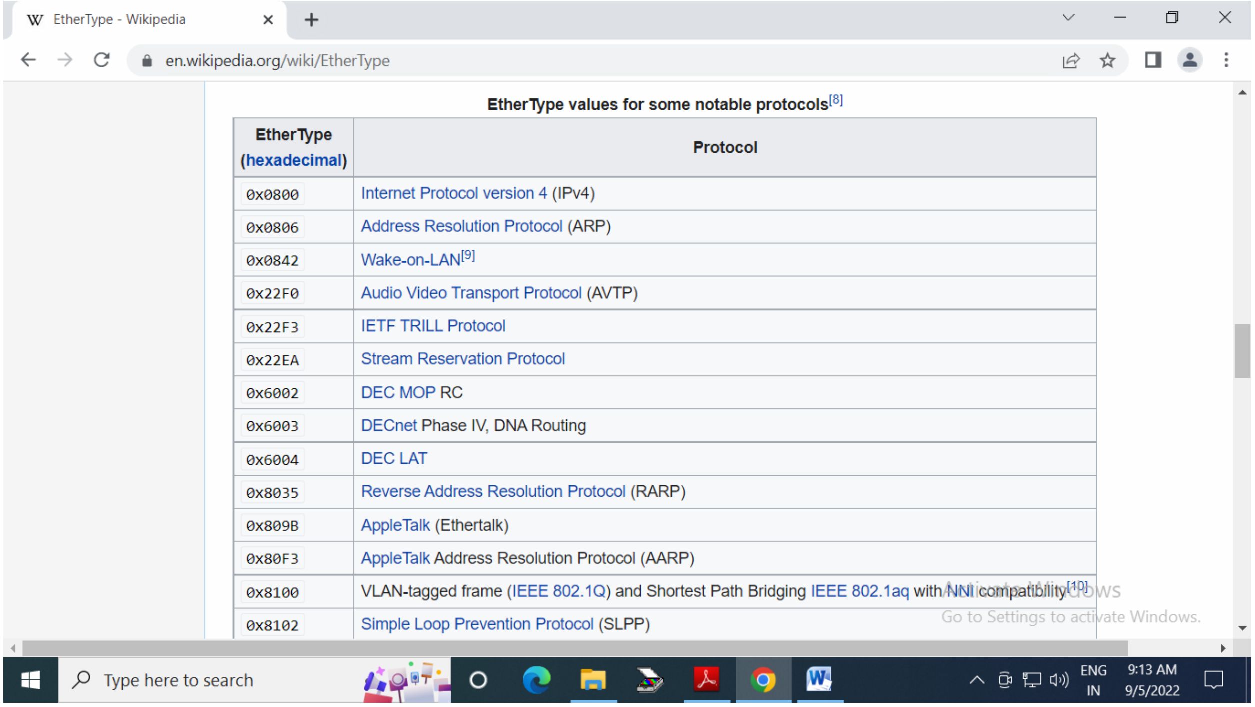
Activities at Alice's Site

We will use symbolic addresses to make the figures more readable. Figure 9.11 shows what happens at Alice's site. The network layer knows it's given N_A, N_B, and the packet, but it needs to find the link-layer address of the next node. The network layer consults its routing table and tries to find which router is next (the default router in this case) for the destination N_B. the routing table gives N_1, but the network layer needs to find the link-layer address of router R1. It uses its ARP to find the link-layer address **L_1**. The network layer can now pass the datagram with the link-layer address to the data-link layer.

The data-link layer knows its own link-layer address, **L_A**. It creates the frame and passes it to the physical layer, where the address is converted to signals and sent through the media.

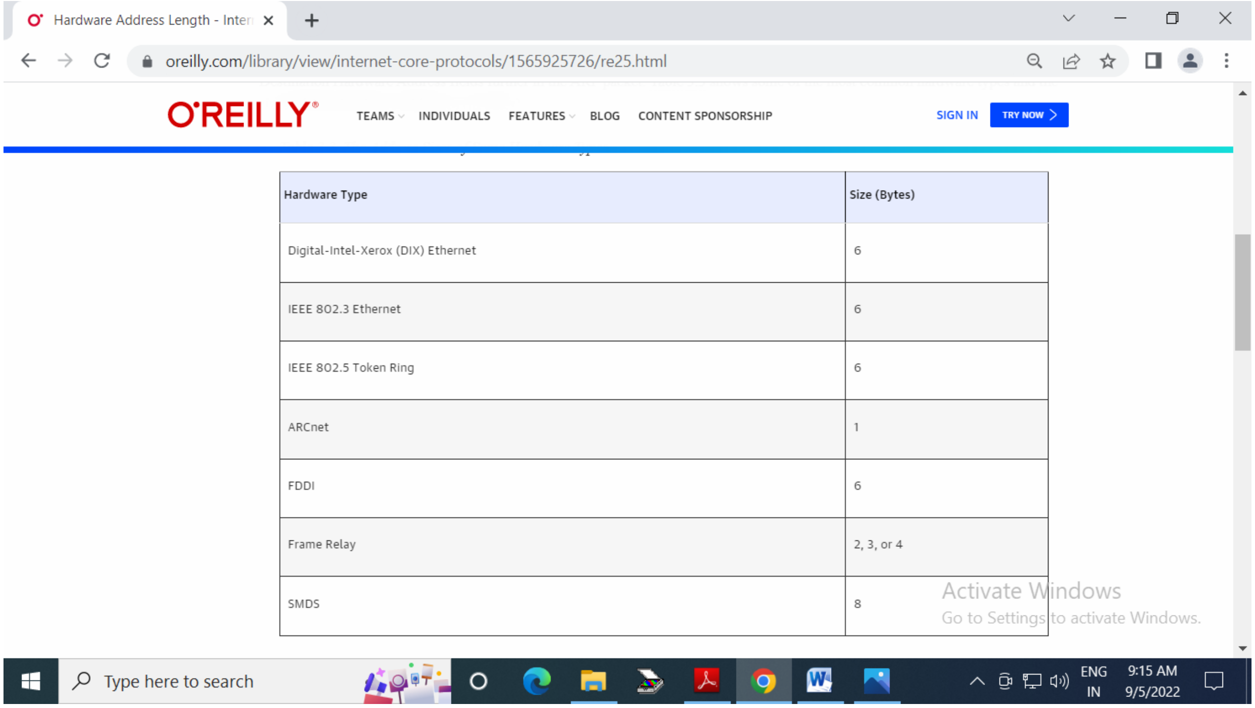
Figure 9.9 *Example 9.4*





EtherType values for some notable protocols[8]

EtherType (hexadecimal)	Protocol
0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x0842	Wake-on-LAN[9]
0x22F0	Audio Video Transport Protocol (AVTP)
0x22F3	IETF TRILL Protocol
0x22EA	Stream Reservation Protocol
0x6002	DEC MOP RC
0x6003	DECnet Phase IV, DNA Routing
0x6004	DEC LAT
0x8035	Reverse Address Resolution Protocol (RARP)
0x809B	AppleTalk (Ethertalk)
0x80F3	AppleTalk Address Resolution Protocol (AARP)
0x8100	VLAN-tagged frame (IEEE 802.1Q) and Shortest Path Bridging IEEE 802.1aq with NNI compatibility[10]
0x8102	Simple Loop Prevention Protocol (SLPP)


Hardware Address Length - Inter
oreilly.com/library/view/internet-core-protocols/1565925726/re25.html
O'REILLY
TEAMS
INDIVIDUALS
FEATURES
BLOG
CONTENT SPONSORSHIP
SIGN IN
TRY NOW
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
IN
9:15 AM
9/5/2022

Hardware Type	Size (Bytes)
Digital-Intel-Xerox (DIX) Ethernet	6
IEEE 802.3 Ethernet	6
IEEE 802.5 Token Ring	6
ARCnet	1
FDDI	6
Frame Relay	2, 3, or 4
SMDS	8

every ARP exchange consists of two distinct packets: the original request and a response to the request. The Source Protocol Address field indicates the sender of this specific ARP packet. If the ARP packet is a request, then this field contains the IP address of the device that is sending the request. If the ARP packet is a response, then this field contains the IP address of the device that is sending the response. The Destination IP Address field indicates the recipient of this specific ARP packet. If the ARP packet is a request, then this field will contain the IP address of the device being looked up. If the ARP packet is a response, then this field will contain the IP address of the device that sent the original request.

Activities at Router R1

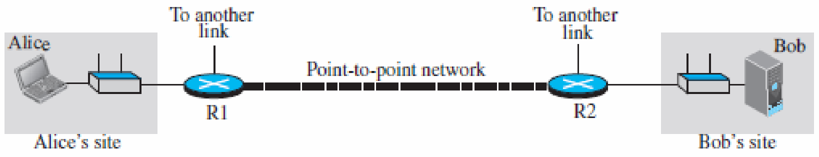
Now let us see what happens at Router R1. Router R1, as we know, has only three lower layers. The packet received needs to go up through these three layers and come down. Figure 9.12 shows the activities. At arrival, the physical layer of the left link creates the frame and passes it to the data-link layer. The data-link layer decapsulates the datagram and passes it to the network layer. The network layer examines the network-layer address of the datagram
and finds that the datagram needs to be delivered to the device with IP address N_B. The network layer consults its routing table to find out which is the next node (router) in the path to N_B. The forwarding table returns N_3. The IP address of router R2 is in the same link with R1. The network layer now uses the ARP to find the link-layer address of this router, which comes up as **L_3.** The network layer passes the datagram and **L_3** to the data-link layer belonging to the link at the right side. The link layer encapsulates the datagram, adds **L3** and **L2** (its own link-layer address), and passes the frame to the physical layer. The physical layer encodes the bits to signals and sends
them through the medium to R2.

Activities at Router R2

Activities at router R2 are almost the same as in R1, as shown in Figure 9.13.

Activities at Bob's Site Now let us see what happens at Bob's site. Figure 9.14 shows how the signals at
Bob's site are changed to a message. At Bob's site there are no more addresses or mapping needed. The signal received from the link is changed to a frame. The frame is passed to the data-link layer, which decapsulates the datagram and passes it to th network layer. The network layer decapsulates the message and passes it to the

Figure 9.10 *The internet for our example*



transport layer.

Changes in Addresses

This example shows that the source and destination network-layer addresses, NA and NB, have not been changed during the whole journey. However, all four network-layer addresses of routers R1 and R2 (N1, N2, N3, and N4) are needed to transfer a datagram from Alice's computer to Bob's computer.

Figure 9.11 *Flow of packets at Alice's computer*

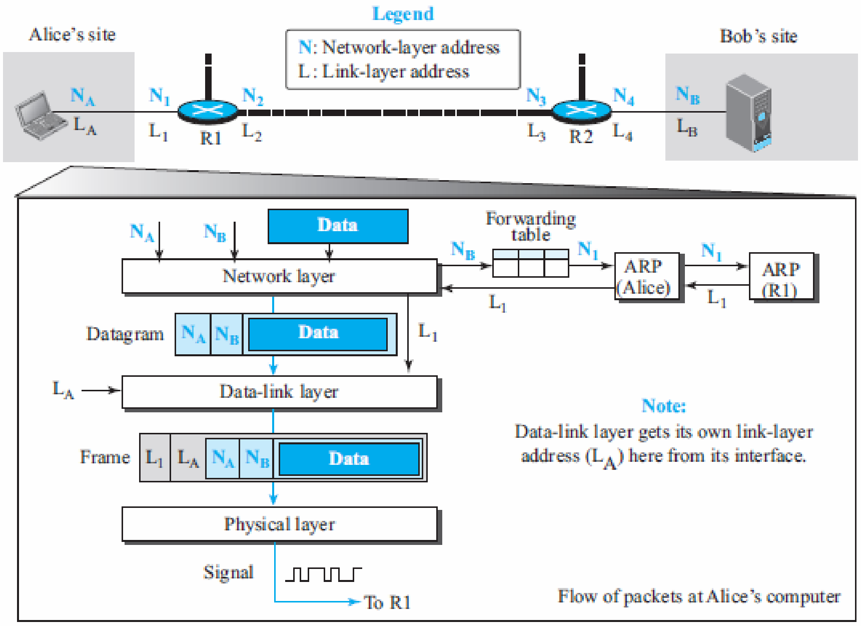


Figure 9.12 *Flow of activities at router R1*

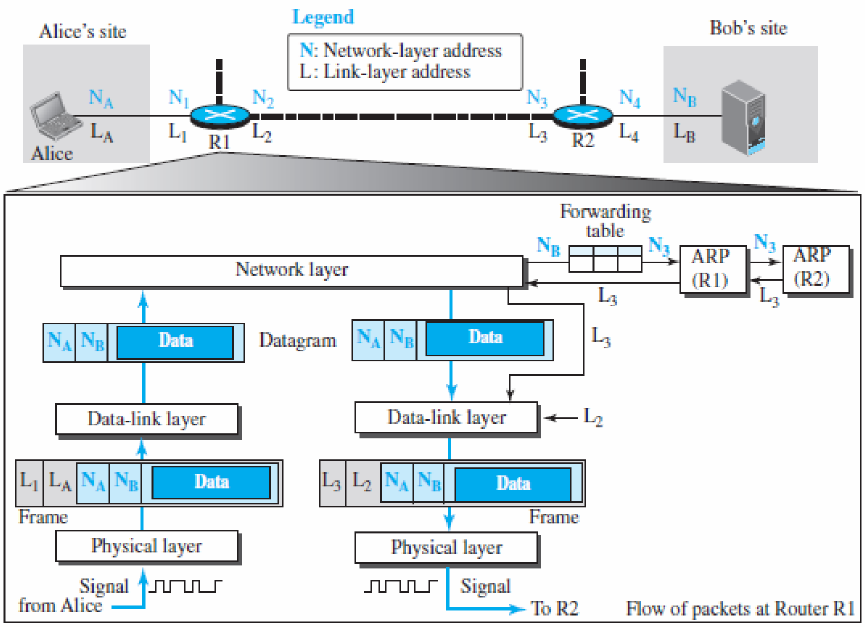


Figure 9.13 *Activities at router R2.*

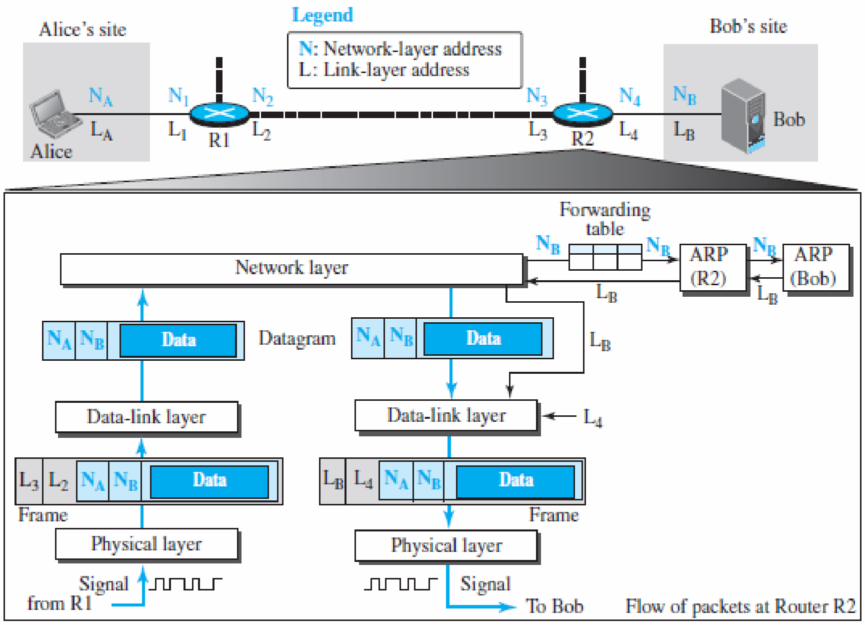
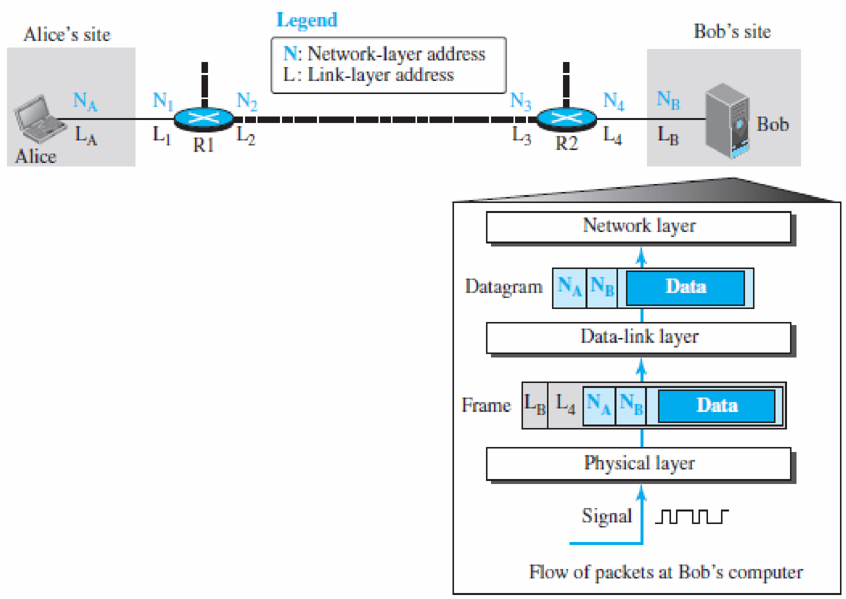


Figure 9.14 *Activities at Bob's site*



10.1 INTRODUCTION

10.1.1 Types of Errors

Whenever bits flow from one point to another, they are subject to unpredictable changes because of **interference.** This interference can change the shape of the signal. The term ***single-bit error*** means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1. The term ***burst error*** means

that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. Figure 10.1 shows the effect of a single-bit and a burst error on a data unit. A burst error is more likely to occur than a single-bit error because the duration of the noise signal is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at 1 kbps, a noise of 1/100 second can affect 10 bits; if we are sending data at 1 Mbps, the same noise can affect 10,000 bits.

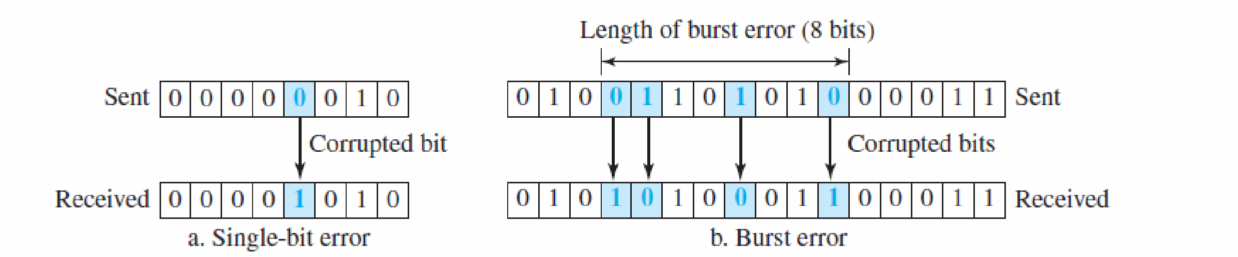
10.1.2 Redundancy

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

10.1.3 Detection versus Correction

The correction of errors is more difficult than the detection. In **error detection**, we are only looking to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of corrupted bits. A single-bit error is the same for us as a burst error. In **error correction**, we need to know the exact number of bits that are corrupted and, more importantly, their location in the message. The number of errors and the size of the message are important factors. If we need to correct a single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two

Figure 10.1 *Single-bit and burst error*



errors in a data unit of the same size, we need to consider 28 (permutation of 8 by 2) possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

10.1.4 Coding

Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits. The receiver checks the relationships between the two sets of bits to detect errors. The ratio of redundant bits to data bits and the robustness of the process

are important factors in any coding scheme. We can divide coding schemes into two broad categories: **block coding** and **convolution coding**.

10.3.1 Cyclic Redundancy Check

We can create cyclic codes to correct errors. However, the theoretical background required is beyond the scope of this book. In this section, we simply discuss a subset of Table 10.3 shows an example of a CRC code. We can see both the linear and cyclic properties of this code. In the encoder, the dataword has *k* bits (4 here); the codeword has *n* bits (7 here). The size of the dataword is augmented by adding *n* *k* (3 here) 0s to the right-hand side

of the word. The *n*-bit result is fed into the generator. The generator uses a divisor of size *n* *k* + 1 (4 here), predefined and agreed upon. The generator divides the augmented dataword by the divisor (modulo-2 division). The quotient of the division is discarded; the remainder ($r_2r_1r_0$) is appended to the dataword to create the codeword.

The decoder receives the codeword (possibly corrupted in transition). A copy of all *n* bits is fed to the checker, which is a replica of the generator. The remainder produced by the checker is a syndrome of *n* *k* (3 here) bits, which is fed to the decision logic analyzer. The analyzer has a simple function. If the syndrome bits are all 0s, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).

Encoder

Let us take a closer look at the encoder. The encoder takes a dataword and augments it with *n* *k* number of 0s. It then divides the augmented dataword by the divisor, as shown in Figure 10.6. The process of modulo-2 binary division is the same as the familiar division process we use for decimal numbers. However, addition and subtraction in this case are the same; we use the XOR operation to do both. As in decimal division, the process is done step by step. In each step, a copy of the divisor is XORed with the 4 bits of the dividend. The result of the XOR operation

(remainder) is 3 bits (in this case), which is used for the next step after 1 extra bit is pulled down to make it 4 bits long. There is one important point we need to remember in this type of division. If the leftmost bit of the dividend (or the part used in each step) is 0, the step cannot use the regular divisor; we need to use an all-0s divisor. When there are no bits left to pull down, we have a result. The 3-bit remainder forms the **check bits** (r_2, r_1, and r_0). They are appended to the dataword to create the codeword.

Decoder

The codeword can change during transmission. The decoder does the same division process as the encoder. The remainder of the division is the syndrome. If the syndrome is all 0s, there is no error with a high probability; the dataword is separated from the received codeword and accepted. Otherwise, everything is discarded. Figure 10.7 shows two cases: The left-hand figure shows the value of the syndrome when no error has occurred; the syndrome is 000. The right-hand part of the figure shows the case in which there is a single error. The syndrome is not all 0s (it is 011).

Divisor

We may be wondering how the divisor 1011 is chosen. This depends on the expectation we have from the code. We will show some standard divisors later in the chapter (Table 10.4) after we discuss polynomials

Table 10.3 *A CRC code with C(7, 4)*

Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Figure 10.5 shows one possible design for the encoder and decoder.

Figure 10.5 *CRC encoder and decoder*

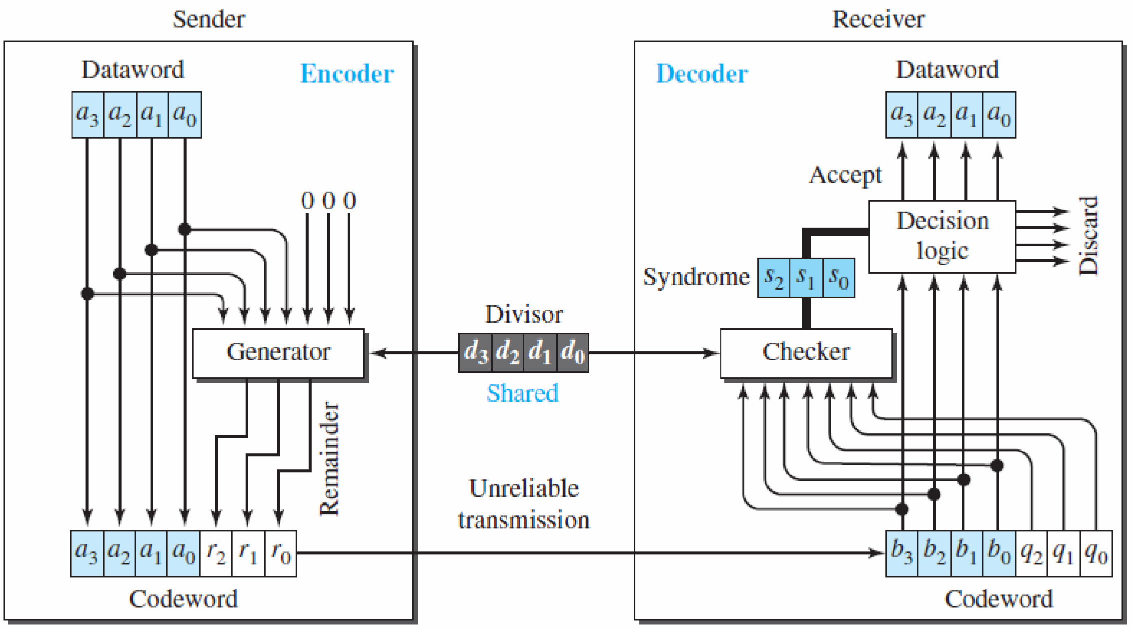


Figure 10.6 *Division in CRC encoder*

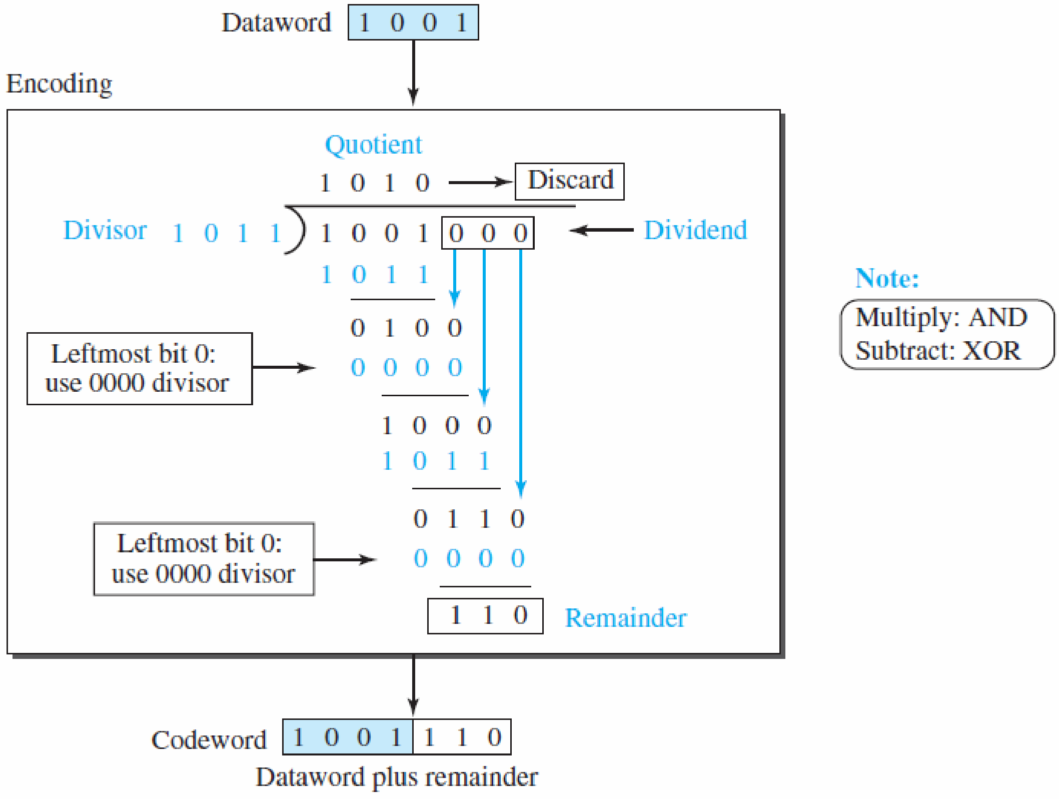
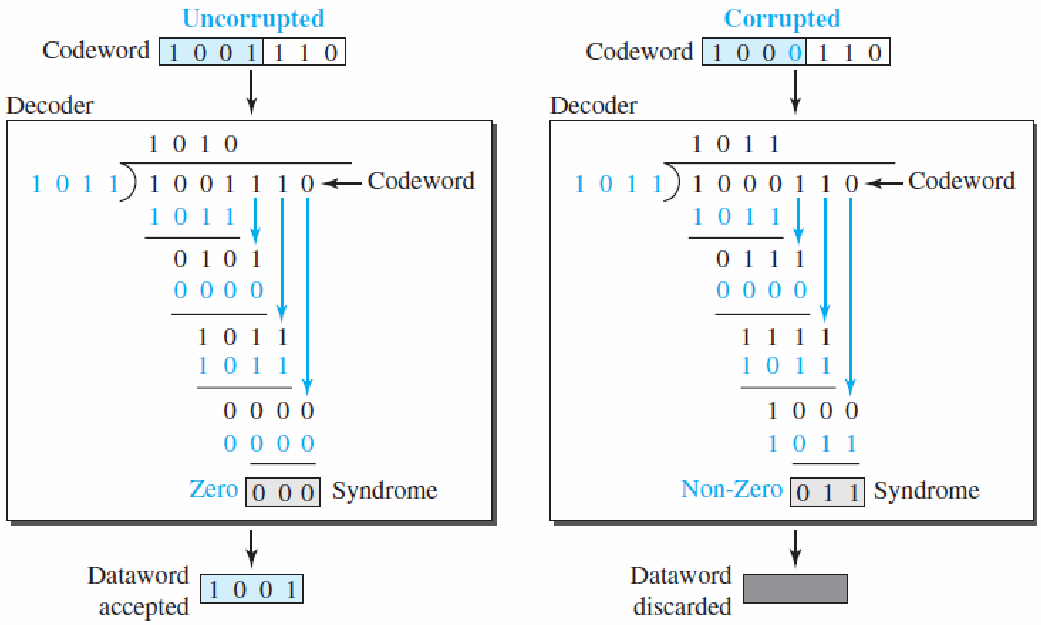


Figure 10.7 *Division in the CRC decoder for two cases*


Uncorrupted
Codeword 1 0 0 1 1 1 0
Decoder
1 0 1 0
1 0 1 1
1 0 0 1 1 1 0
Codeword
1 0 1 1
0 1 0 1
0 0 0 0
1 0 1 1
1 0 1 1
0 0 0 0
0 0 0 0
Zero 0 0 0 Syndrome
Dataword accepted 1 0 0 1
Corrupted
Codeword 1 0 0 0 1 1 0
Decoder
1 0 1 1
1 0 1 1
1 0 0 0 1 1 0
Codeword
1 0 1 1
0 1 1 1
0 0 0 0
1 1 1 1
1 0 1 1
1 0 0 0
1 0 1 1
Non-Zero 0 1 1 Syndrome
Dataword discarded