DJA3A - COMPUTER NETWORKS

UNIT- V **148-170**

TCP/IP Protocol suite Part I: Overview of TCP/IP – Network layer – Addressing.

TCP/IP Protocol suite Part II: File Transfer Protocol – Telnet – SMTP – HTTP.

Network Security: Four Aspects of Security – Digital signature – PGP – Access Authorization.

Text Book:

Data Communications and Networking - Behrouz A. Forouzon

2nd edition Tata McGraw Hill edition.

UNIT- I

1. DATA COMMUNICATION

1.1 DATA COMMUNICATION

When we communicate, we are sharing information. This sharing can be local or remote. Between individuals, local-communication usually occurs face to face, while remote communication takes place over distance. The term telecommunication, which includes telephony, telegraphy, and television, means communication at a distance. The word tele come from Greek.

The word data refers to facts, concepts, and instructions presented in whatever form is agreed upon by the parties creating and using the data. In the context of computer information systems, data are represented by binary information (or bits) produced and consumed in the form of 0s and ls.

In computer information systems, data are represented by binary information units (or bits) produced and consumed in the form of 0s and ls.

Data communication is the exchange of data (in the form of 0s and 1s) between two devices via some form of transmission medium (such as a wire cable). Data communication is considered local if the communicating devices are in the same building or a similarly restricted geographical area, and is considered remote if the devices are farther apart.

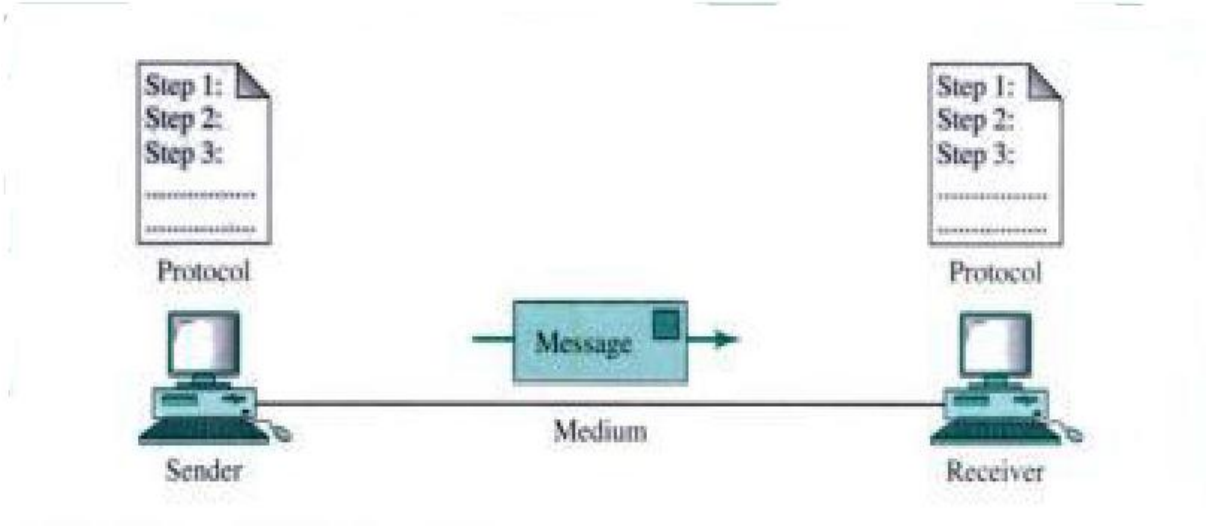
For data` communication to occur, the communicating devices must be part of a communication system made up of a combination of hardware and software. The effectiveness of a data communication system depends on three fundamental characteristics:

1. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.
2. **Accuracy.** The system must deliver 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, audio, and voice data, 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.

Components

A data communication system is made up of five components (Figure 1.1)

Figure 1.1 Data communication system components



1. **Message.** The **message** is the information (data) to be communicated. It can consist of text, numbers, pictures, sound, or video or any combination of these.
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. **Medium.** The transmission **medium** is the physical path by which a message travels from sender to receiver. It can consist of twisted pair wire, coaxial cable, fiber-optic cable, laser, or radio waves (terrestrial or satellite microwave).
5. **Protocol.** A **protocol** is a set of rules that govern data communication. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating.

1.2NETWORKS

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

Distributed Processing

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

Advantages of distributed processing include the following:

- **Security/encapsulation.** A system designer can limit the kinds of interactions that a given user can have with the entire system. For example, a bank can allow users access to

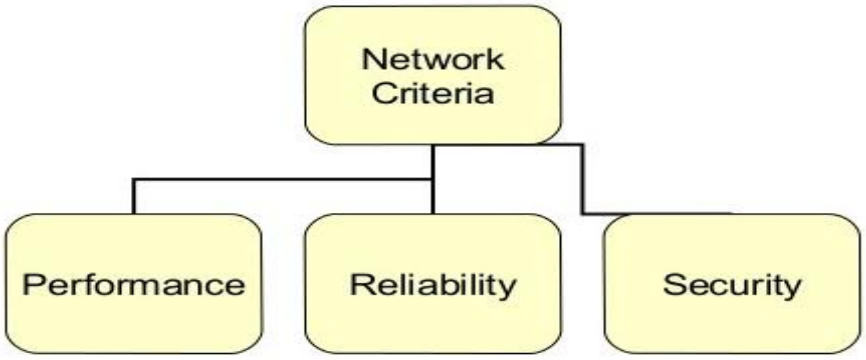
their own accounts through an automated teller machine (ATM) without allowing them access to the bank's entire database.

- **Distributed databases.** No one system needs to provide storage capacity for the entire database. For example, the World Wide Web gives users access to information that may be actually stored and manipulated anywhere on the Internet.
- **Faster problem solving.** Multiple computers working on parts of a problem concurrently often can solve the problem faster than a single machine working alone. For example, networks of PCs have broken encryption codes that were presumed to be unbreakable because of the amount of time it would take a single computer to crack them.
- **Security through redundancy.** Multiple computers running the same program at the same time can provide security through redundancy. For example, in the space shuttle, three computers run the same program so that if one has a hardware error, the other two can override it.
- **Collaborative processing.** Both multiple computers and multiple users may inter-act on a task. For example, in multiuser network games the actions of each player are visible to and affect all the others.

Network Criteria

To be considered effective and efficient, a network must meet a number of criteria. The most important of these are performance, reliability, and security.(Figure 1.2)

Figure 1.2 Network criteria



Performance

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response.

The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hard ware, and the efficiency of the software.

- **Number of users.** Having a large number of concurrent users can slow response time in a network not designed to coordinate heavy traffic loads. The design of a given network is based on an assessment of the average number of users that will be communicating at any

one time. In peak load periods, however, the actual number of users can exceed the average and thereby decrease performance. How a network responds to loading is a measure of its performance.

- T**ype of transmission medium.** The medium defines the speed at which data can travel through a connection (the data rate). Today's networks are moving to faster transmission media, such as fiber-optic cabling. A medium that can carry data at 100 megabits per second is 10 times more powerful than a medium that can carry data at only 10 megabits per second. However, the speed of light imposes an upper bound on the data rate.
- **Hardware.** The types of hardware included in a network affect both the speed and capacity of transmission. A higher-speed computer with greater storage capacity provides better performance.
- **Software.** The software used to process data at the sender, receiver, and intermediate nodes also affects network performance. Moving a message from node to node through a network requires processing to transform the raw data into transmittable signals, to the proper destination, to ensure error free delivery, and to recast the signals into a form the receiver can use. The software that provides these services affects both the speed and the reliability of a network link. Well-designed software can speed the process and make transmission more effective and efficient.

Reliability

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

- **Frequency of failure.** All networks fail occasionally. A network that fails often however, is of little value to a user.
- **Recovery time of a network after a failure** is the time taken to restore service. A network that recovers quickly is more useful one.
- **Catastrophe.** Networks must be protected from catastrophic events such as fire, earthquake, or theft. One protection against unforeseen damage is a reliable system to back up network software.

Security

Security Network security issues include protecting data from unauthorized access and viruses.

- **Unauthorized access.** For a network to be useful, sensitive data must be protected from unauthorized access. Protection can be accomplished at a number of levels. At the lowest level are user identification codes and passwords. At a higher level are encryption techniques. In these mechanisms, data are systematically altered in such a way that if they are intercepted by an unauthorized user, they will be unintelligible.
- **Viruses.** Because a network is accessible from many points, it can be susceptible to computer viruses. A virus is an illicitly introduced code that damages the sys-tem. A good network is protected from viruses by hardware and software designed specifically for that purpose.

Applications

In the short time they have been around, data communication networks have become an indispensable part of business, industry, and entertainment. Some of the network applications in different fields are the following:

- **Marketing and. sales.** Computer networks are used extensively in both marketing and sales organizations. Marketing professionals use them to collect, exchange, and analyze data relating to customer needs and product development cycles. Sales applications include teleshopping, which uses order-entry computers or telephones connected to an order-processing network, and on-line reservation services for hotels, airlines, and so on.
- **Financial services.** Today's financial services are totally dependent on computer networks. Applications include credit history searches, foreign exchange and investment services, and electronic funds transfer (EFT), which allows a user to transfer money without going into a bank (e.g. Automated teller machine is a kind of electronic funds transfer; automatic paycheck deposit).
- **Manufacturing.** Computer networks are used today in many aspects of manufacturing, including the manufacturing process itself. Two applications that use networks to provide essential services are computer-assisted design (CAD) and computer-assisted manufacturing (CAM), both of which allow multiple users to work on a project simultaneously.
- **Electronic messaging.** Probably the most widely used network application is electronic mail (e-mail).
- **Directory services.** Directory services allow lists of files to be stored in a central location to speed worldwide search operations.
- **Information services.** Network information services include bulletin boards and data banks. A World Wide Web site offering the technical specifications for a new product is an information service.
- **Electronic data interchange (EDI).** EDI allows business information (including documents such as purchase orders and invoices) to be transferred without using paper.
- **Teleconferencing.** Teleconferencing allows conferences to occur without the participants being in the same place. Applications include simple text conferencing (where participants communicate through their keyboards and computer monitors), voice conferencing (where participants at a number of locations communicate simultaneously over the phone), and video conferencing (where participants can see as well as talk to one another).
- **Cellular telephone.** In the past, two parties wishing to use the services of the telephone company had to be linked by a fixed physical connection. Today's cellular networks make, it possible to maintain wireless phone connections even while traveling over large distances.
- **Cable television.** Future services provided by cable television networks may include video on request, as well as the same information, financial, and communication services currently provided by the telephone companies and computer networks.

1.3 PROTOCOLS AND STANDARDS

Protocols

In computer networks, communication occurs between entities in different systems. An entity is anything capable of sending or receiving information. Examples include application programs, file transfer packages, browsers, database management systems, and electronic mail software. A system is a physical object that contains one or more entities. Examples include computers and terminals. However, two entities cannot simply send bit streams to each other and expect to be understood. For communication to occur, the entities must agree on a protocol. A protocol is a set of rules that govern data communications. A protocol defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics, and timing.

- **Syntax.** The term syntax refers to the structure or format of the data, meaning the order in which they are presented. For example, a simple protocol might expect the first 8 bits of data to be the address of the sender, the second 8 bits to be the address of the receiver, and the rest of the stream to be the message itself.
- **Semantics.** The word semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation? For example, does an address identify the route to be taken or the final destination of the message?
- **Timing**. The term timing refers to two characteristics: when data should be sent and how fast they can be sent. For example, if a sender produces data at 100 Mbps but the receiver can process data at only 1 Mbps, the transmission will overload the receiver and some data will be lost.

Standards. Standards are essential in creating and maintaining an open and competitive market for equipment manufacturers and in guaranteeing national and international interoperability of data and telecommunications technology and processes. Standards provide guidelines to manufacturers, vendors, government agencies, and other service providers to ensure the kind of interconnectivity necessary in today's marketplace and in international communications.

Data communication standards fall into two categories: de facto (meaning "by fact" or "by convention") and de jure (meaning "by law" or "by regulation").

- **De facto**. Standards that have not been approved by an organized body but have been adopted as standards through widespread use are de facto standards. De facto standards are often established originally by manufacturers who seek to define the functionality of a new product or technology.
- **De jure.** Those standards that have been legislated by an officially recognized body are de jure standards.

1.4 STANDARDS ORGANIZATIONS

An association of organizations, governments, manufacturers and users form the standards organizations and are responsible for developing, coordinating and maintaining the standards. The intent is that all data communications equipment manufacturers and users comply with these standards. The primary standards organizations for data communication are:

International Standard Organization (ISO)

ISO is the international organization for standardization on a wide range of subjects. It is comprised mainly of members from the standards committee of various governments throughout the world. It is even responsible for developing models which provides high level of system compatibility, quality enhancement, improved productivity and reduced costs. The ISO is also responsible for endorsing and coordinating the work of the other standards organizations.

International Telecommunications Union-Telecommunication Sector (ITU-T)

ITU-T is one of the four permanent parts of the International Telecommunications Union based in Geneva, Switzerland. It has developed three sets of specifications: the V series for modem interfacing and data transmission over telephone lines, the X series for data transmission over public digital networks, email and directory services; the I and Q series for Integrated Services Digital Network (ISDN) and its extension Broadband ISDN. ITU-T membership consists of government authorities and representatives from many countries and it is the present standards organization for the United Nations.

Institute of Electrical and Electronics Engineers (IEEE)

IEEE is an international professional organization founded in United States and is compromised of electronics, computer and communications engineers. It is currently the world's largest professional society with over 200,000 members. It develops communication and information processing standards with the underlying goal of advancing theory, creativity, and product quality in any field related to electrical engineering.

American National Standards Institute (ANSI)

ANSI is the official standards agency for the United States and is the U.S voting representative for the ISO. ANSI is a completely private, non-profit organization comprised of equipment manufacturers and users of data processing equipment and services. ANSI membership is comprised of people form professional societies, industry associations, governmental and regulatory bodies, and consumer goods.

Electronics Industry Association (EIA)

EIA is a non-profit U.S. trade association that establishes and recommends industrial standards. EIA activities include standards development, increasing public awareness, and lobbying and it is responsible for developing the RS (recommended standard) series of standards for data and communications.

Telecommunications Industry Association (TIA)

TIA is the leading trade association in the communications and information technology industry. It facilitates business development opportunities through market development, trade promotion, trade shows, and standards development. It represents manufacturers of communications and information technology products and also facilitates the convergence of new communications networks.

Internet Architecture Board (IAB)

IAB earlier known as Internet Activities Board is a committee created by ARPA (Advanced Research Projects Agency) so as to analyze the activities of ARPANET whose purpose is to accelerate the advancement of technologies useful for U.S military. IAB is a technical advisory group of the Internet Society and its responsibilities are: I. Oversees the

architecture protocols and procedures used by the Internet. II. Manages the processes used to create Internet Standards and also serves as an appeal board for complaints regarding improper execution of standardization process. III. Responsible for administration of the various Internet assigned numbers IV. Acts as a representative for Internet Society interest in liaison relationships with other organizations. V. Acts as a source of advice and guidance to the board of trustees and officers of Internet Society concerning various aspects of internet and its technologies.

Internet Engineering Task Force (IETF)

The IETF is a large international community of network designers, operators, vendors and researchers concerned with the evolution of the Internet architecture and smooth operation of the Internet.

Internet Research Task Force (IRTF)

The IRTF promotes research of importance to the evolution of the future Internet by creating focused, long-term and small research groups working on topics related to Internet protocols, applications, architecture and technology.

2. BASIC CONCEPTS

2.1. LINE CONFIGURATION

Line configuration refers to the way two or more communication devices attach to a ink. A link is the physical communication pathway that transfers data from one device to another. For the purposes of visualization; it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time. There are two possible line con-figurations: point-to-point and multipoint. (figure 2.1)

Line configuration defines the attachment of communication devices to a link

Point-to-Point.

A point-to-point line configuration provides a dedicated link between two devices. The entire capacity of the 'channel is reserved for transmission between those two devices. Most point-to-point line configurations 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 (Figure 2.2). When you change television channels by infrared remote control, you are establishing a point-to-point line configuration between the remote control and the television's control system.

Figure 2.1 Two categories of line configuration

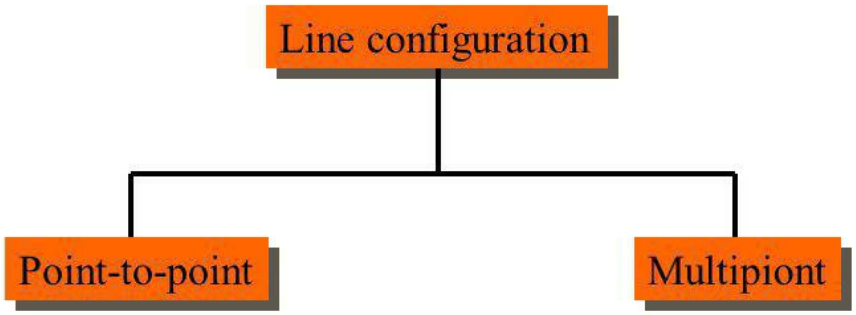
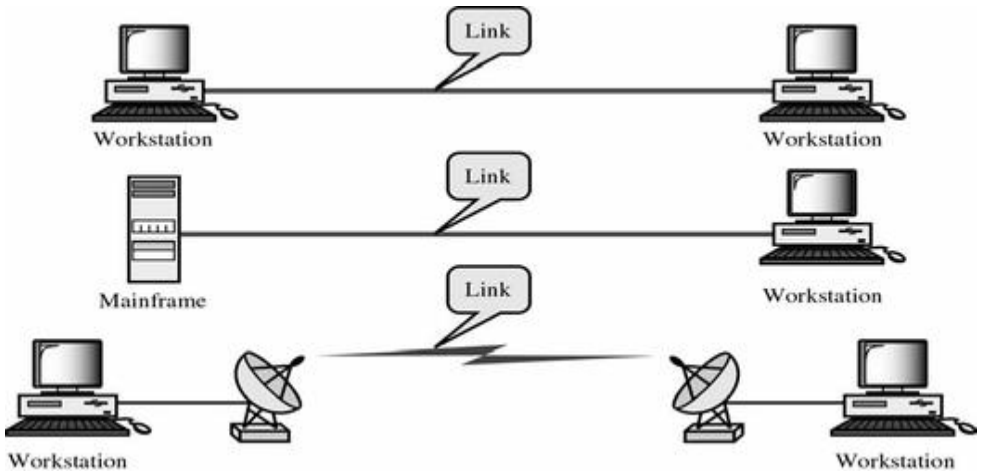


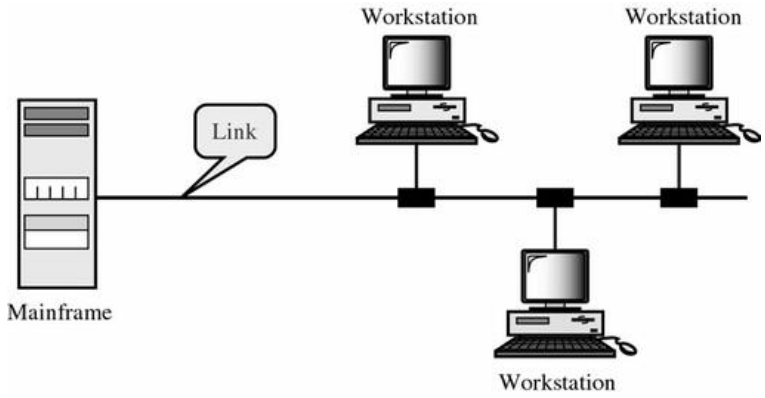
Figure 2.2 Point-to-point line configuration



Multipoint

A multipoint (also called multidrop) line configuration is one in which more than two specific devices share a single link (Figure 2.3). 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 line configuration. If users must take turns, it is a time-shared line configuration.

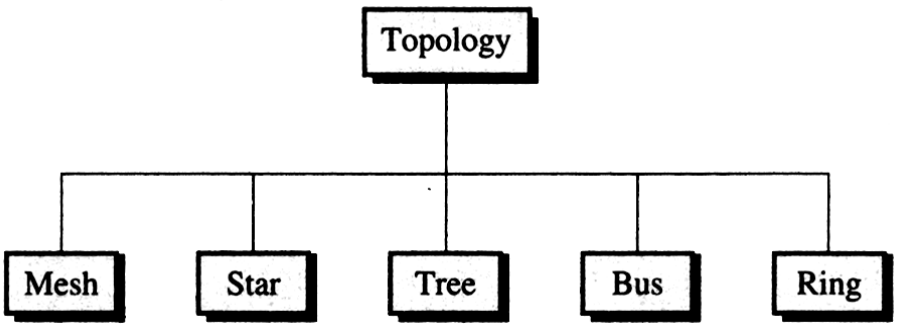
Figure 2.3 Multipoint line configuration



2.2 TOPOLOGY

The term topology refers to the way a network is laid out, either physically or logically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to each other. There are five basic topologies possible: mesh, star, tree, bus, and ring (Figure 2.4).

Figure 2.4 Categories of topology



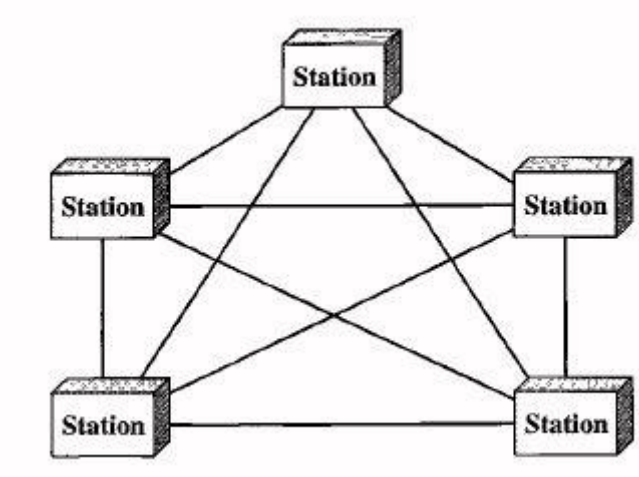
Topology defines physical or logical arrangement of links in a network

These five labels describe how the devices in a network are interconnected rather than their physical arrangement. For example, having a star topology does not mean that all of the computers in the network must be placed physically around a hub in a star shape. A consideration when choosing a topology is the relative status of the devices to be linked. Two relationships are possible: **peer-to-peer**, where the devices share the link equally, and **primary-secondary**, where one device controls traffic and the others must transmit through it. Ring and mesh topologies are more convenient for peer-to-peer transmission, while star and tree are more convenient for primary, secondary. A bus topology is equally convenient for either.

Mesh

In a **mesh topology**, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects. A fully connected mesh network therefore has n (n - 1)/2 physical channels to link n devices. To accommodate that many links, every device on the net-work must have n -1 input/output (1/0) ports (see Figure 2.5).

Figure 2.5 Fully connected mesh topology for five devices



A mesh offers several advantages over other network topologies.

1. 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. A mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system.
3. This topology has privacy or security. When every message sent travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages
4. Point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems. This facility enables the net-work manager to discover the precise location of the fault and aids in finding its cause and solution.

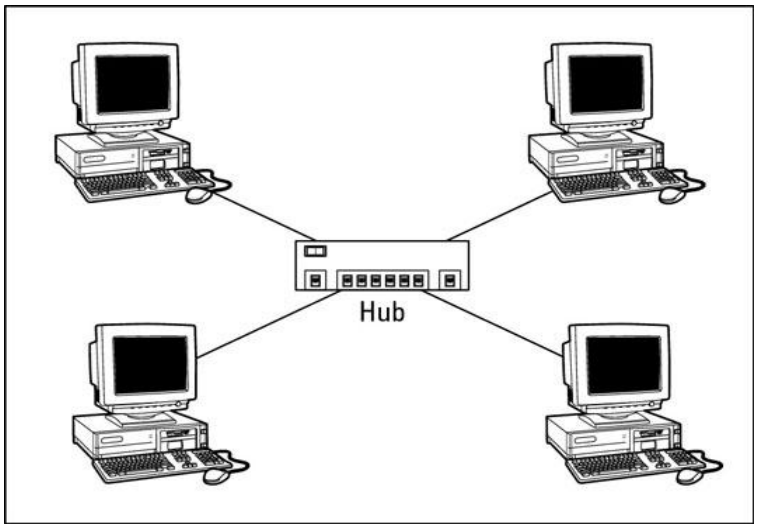
The main disadvantages of a mesh are related to the amount of cabling and the number of I/O ports required.

1. Because every device must be connected to every other device, installation and reconfiguration are difficult.
2. The sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate.
3. The hardware required to connect each link (I/0 ports and cable) can be prohibitively expensive)

Star

In a **star topology**, each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to each other. 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. (figure 2.6)

Figure 2.6 Star topology



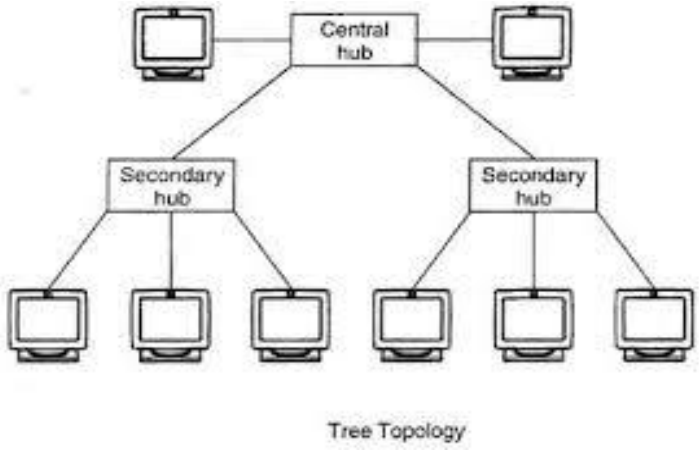
A star topology is less expensive than a mesh topology. In a star, each device needs only one link and one I/O port to connect it to any number of others. This factor also makes it easy to install and reconfigure. Far less cabling needs to be housed, and additions, moves, and deletions involve only one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

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

Tree

Tree topology is a variation of a star. As in a star, nodes in a tree are linked to a central hub that controls the traffic to the network. However, not every device plugs directly into the central hub. The majority of devices connect to a secondary hub that in turn is connected to the central hub (see Figure 2.7).

Figure 2.7 Tree topology



The central hub in the tree is an active hub. An active hub contains a repeater, which is a hardware device that regenerates the received bit patterns before sending them out. Repeating strengthens trans-missions and increases the distance a signal can trave1.

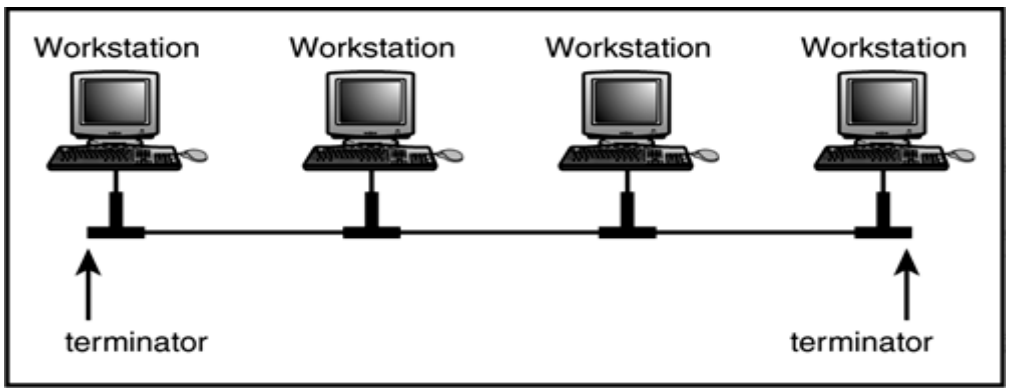
The secondary hubs may be active or passive hubs. A passive hub provides a simple physical connection between the attached devices. The advantages and disadvantages of a tree topology are generally the same as those of a star. The addition of secondary hubs, however, brings two further advantages.

1. It allows more devices to be attached to a single central hub and can therefore increase the distance a signal can travel between devices.
2. It allows the network to isolate and prioritize communications from different computes.

Bus

The preceding examples all describe point-to-point configurations. A bus topology, on the other hand, is multipoint. One long cable acts as a backbone to link all the devices in the network. (Figure. 2.8)

Figure.2.8 Bus topology



Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core. As a signal travels along the backbone, some of its energy is trans-formed into heat. Therefore, it becomes weaker and weaker the farther it has to travel. 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 are

1. Ease of installation.
2. Backbone cable can be laid along the most efficient path, then connected to the nodes by drop lines of various lengths.
3. Bus uses less cabling than mesh, star, or tree topologies.
4. Redundancy is eliminated.
5. Only the back-bone cable stretches through the entire facility. Each drop line has to reach only as far as the nearest point on the backbone

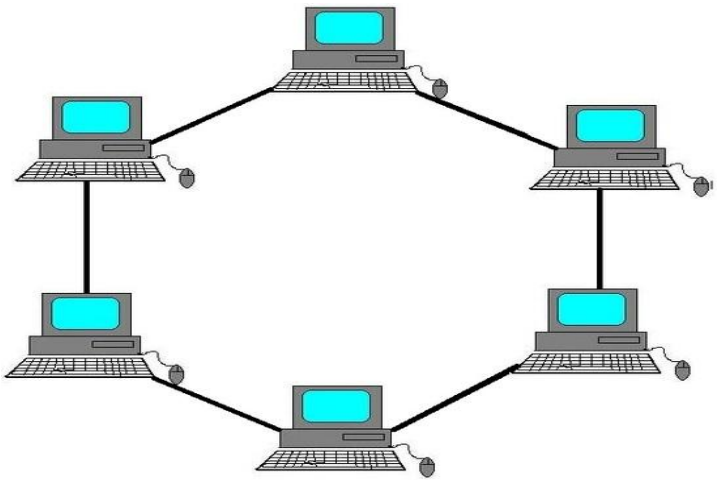
Disadvantages bus topology are

1. Difficult reconfiguration and fault isolation. .
2. It is difficult to add new devices. Adding new devices may therefore require modification or replacement of the backbone.
3. 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. I
4. A fault or break in the bus cable stops all transmission, even between devices on the same side of the problem.
5. The damaged area reflects signals back in the direction of origin, creating noise in both directions.

Ring

In a ring topology, each device has a dedicated point-to-point line configuration only with 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 (Figure 2.9)

Figure 2.9 Ring topology



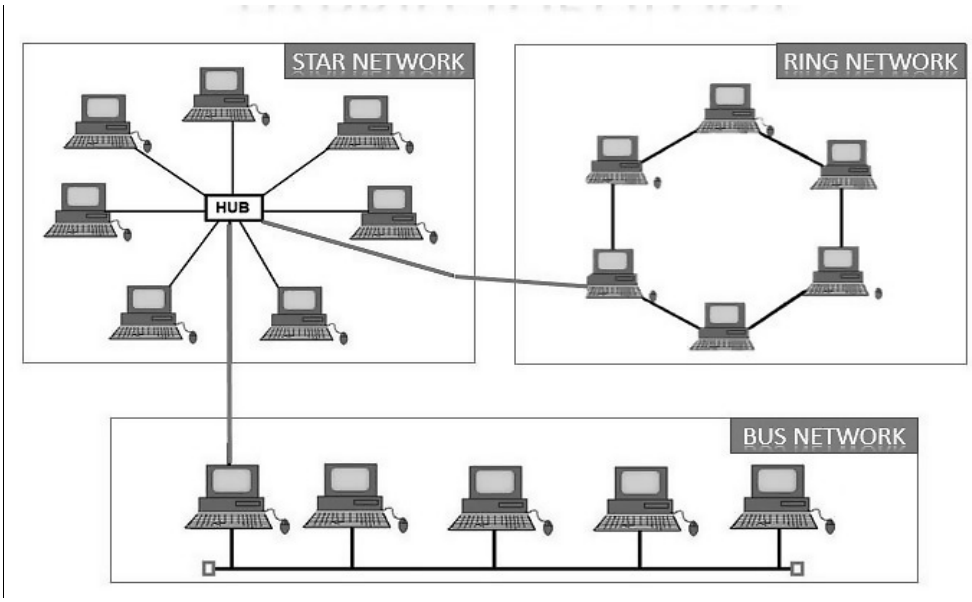
A ring is relatively easy to install and reconfigure. Each device is linked only to its immediate neighbors (either physically or logically). To add or delete a device requires moving only two connections. The only constraints are media and traffic considerations (maximum ring length and number of devices). 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.

Hybrid Topologies

Often a network combines several topologies as subnet works linked together in a larger topology. For instance, one department of a business may have decided to use a bus topology while another department h a ring. The two can be connected to each other via a central controller in a star topology.(Figure 2.10)

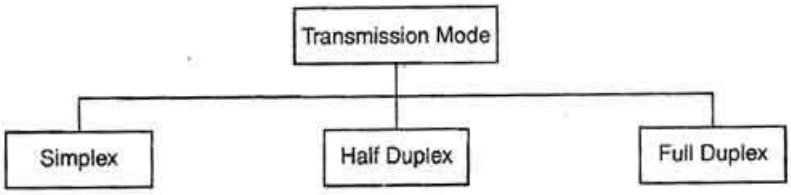
Figure 2.10 Hybrid topology



2.3 TRANSMISSION MODE

The term transmission mode is used to define the direction of signal flow between two linked devices. There are three types of transmission modes: simplex, half-duplex, and full-duplex.(Figure 2.11)

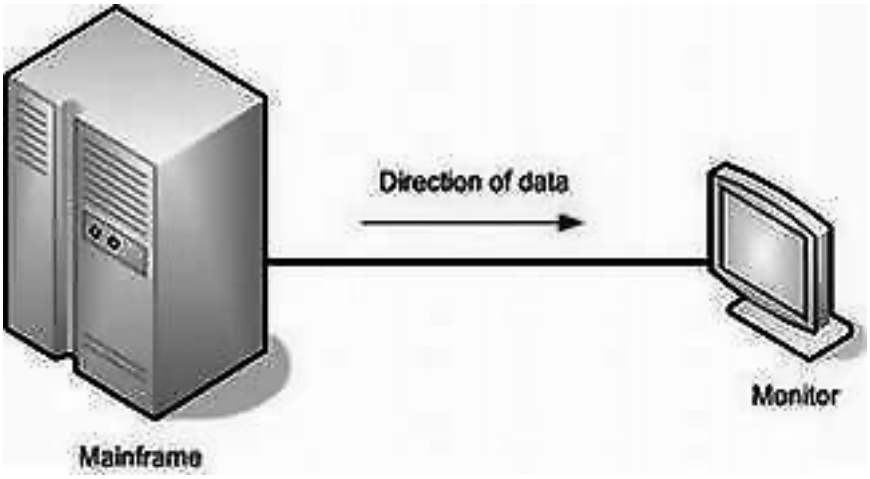
Figure 2.11 Transmission modes



Simplex

Simplex In simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two stations on a link can transmit; the other can only receive (Figure 2.12).

Figure 2.12 Simplex



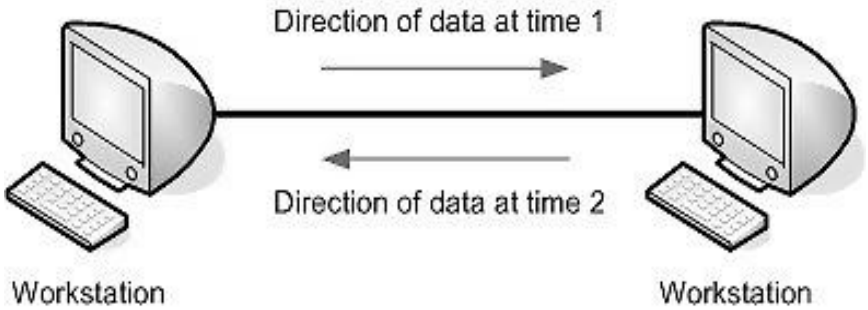
The term transmission mode refers to the direction of information flow between two devices.

Keyboards and traditional monitors are both examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output.

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 (Figure 2.13).

Figure 2.13 Half-duplex

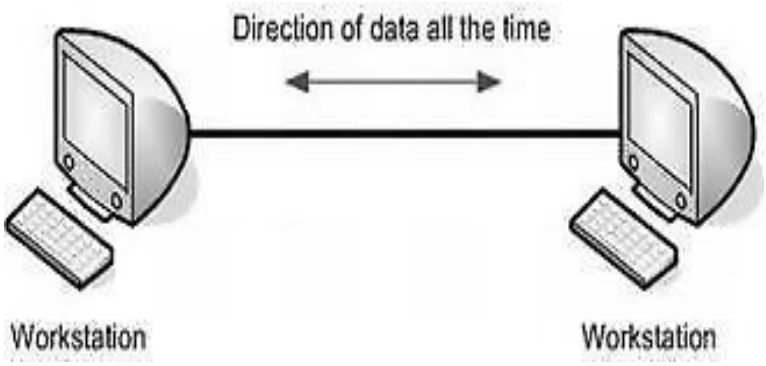


The half-duplex mode is like a one-lane road with two-directional traffic. While cars are traveling one direction, cars going the other way must wait. In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkie-talkies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

Full-Duplex

In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously (Figure 2.14).

Figure 2.14 Full-duplex



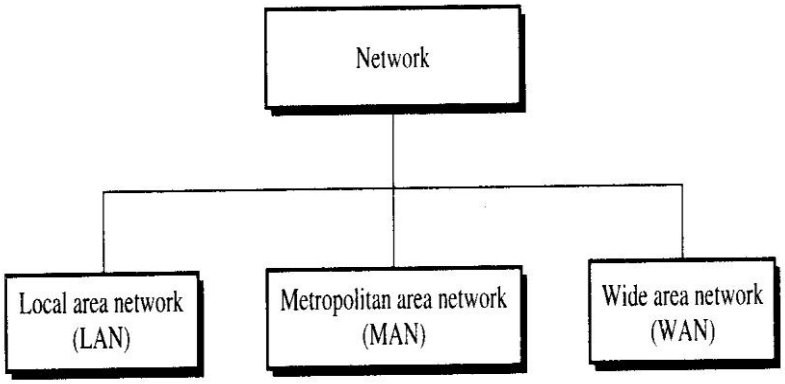
The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in either direction share the capacity of the link. This sharing can occur in two ways either the link must contain two physically separate transmission paths, one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

2.4 CATEGORIES OF NETWORKS

Networks referring to three primary categories: local area networks, metropolitan area networks, and wide area networks. Into which category a network falls is determined by its size, its ownership, the distance it covers, and its physical architecture.(Figure 2.15)

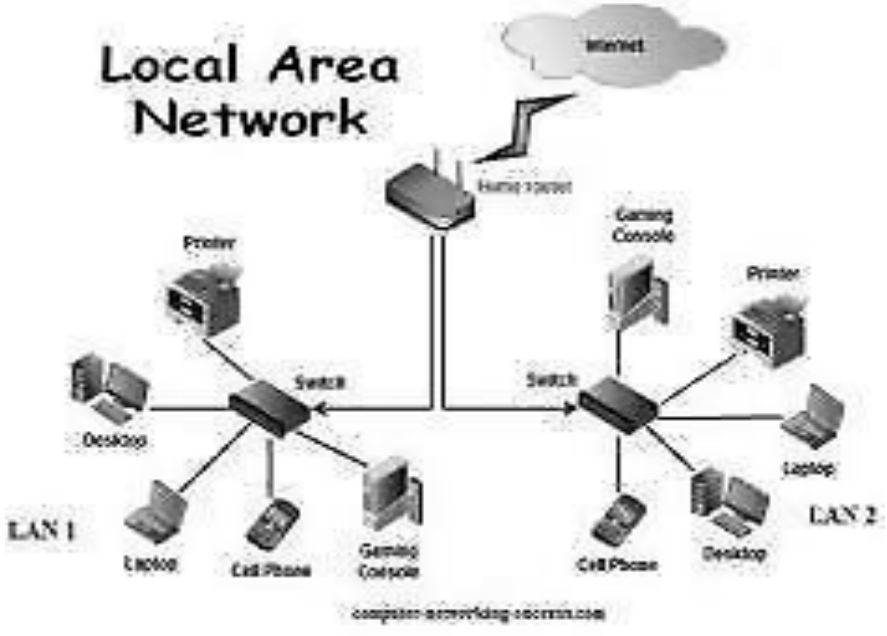
Figure 2.15 Categories of networks



Local Area Network (LAN)

A local area network (LAN) is usually privately owned and links the devices in a sin-g e office, building, or campus (Figure 2.16). 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 voice, sound, and video peripherals. Currently, LAN size is limited to a few kilometers.

Figure 2.16 LAN



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), soft-ware (e.g., an application program), or data. A common example of a LAN, found in many business environments, links a work group of task-related computers, for example, engineering workstations or accounting PCs. One of the computers may be given a large-capacity disk drive and become a server to the other clients. Software can be stored on this central server and used as needed by the whole group.

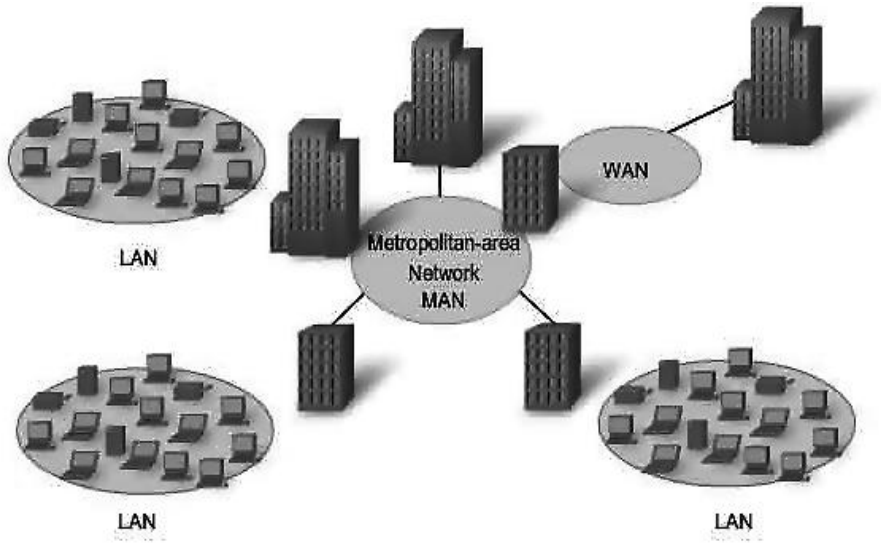
In addition to size, LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium. The most common LAN topologies are bus, ring, and star.

Traditionally, LANs have data rates in the 4 to 16 Mbps range.

Metropolitan Area Network (MAN)

A metropolitan area network (MAN) is designed to extend over an entire city. It may be a single network such as a cable television network, or it may be a means of connecting a number of LANs into a larger network so that resources may be shared LAN-to-LAN as well as device-to-device. For example, a company can use a MAN to connect the LANs in all of its offices throughout a city. (Figure 2.17)

Figure 2.17 MAN

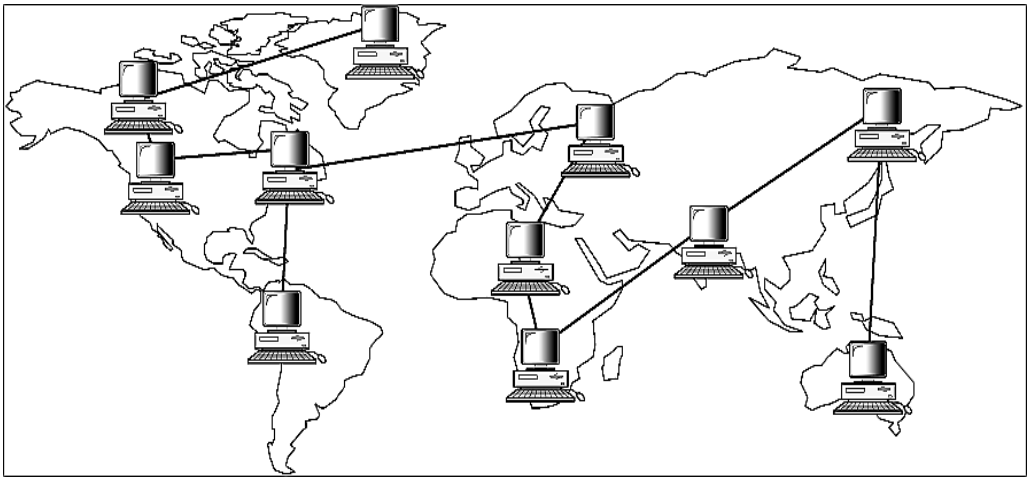


MAN may be wholly owned and operated by a private company, or it may be a service provided by a public company, such as a local telephone company. Many telephone companies provide a popular MAN service called Switched Multi-megabit Data Services..

Wide Area Network (WAN)

A wide area network (WAN) provides long-distance transmission of data, voice, image, and video information over large geographical areas that may comprise a country, a continent, or even the whole world (Figure 2.18). In contrast to LANs (which depend on their own hardware for transmission), WANs may utilize public, leased, or private communication devices, usually in combinations, and can therefore span an unlimited number of miles.

Figure 2.18 WAN

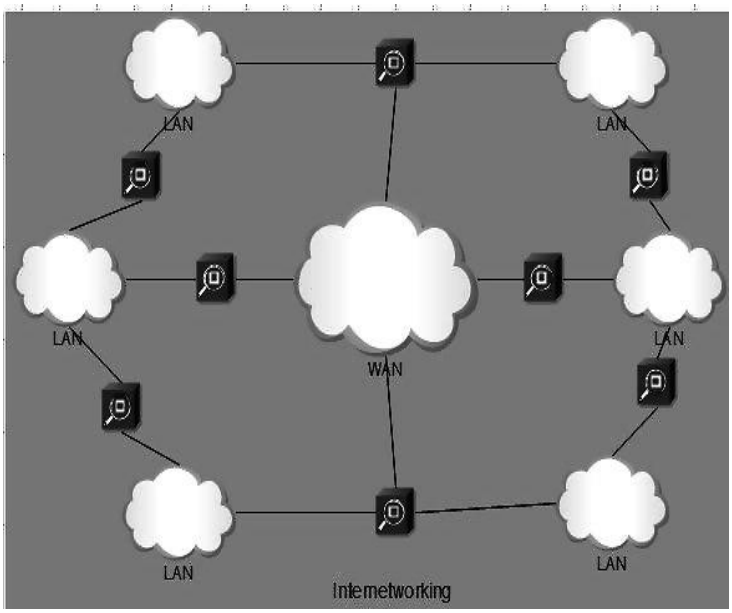


A WAN that is wholly owned and used by a single company is often referred to as an enterprise network.

2.5 INTERNETWORKS

When two or more networks are connected, they become an **internetwork**, or **internet.** In the Figure 2.19, the boxes labeled R represent routers). Individual networks are joined into internetworks by the use of internetworking device. The term internet (lowercase i) should not be confused with the Internet (uppercase 1). The first is a generic term used to mean an interconnection of networks. The second is the name of a specific worldwide network.

Figure 2.19 Internetwork (internet)



UNIT II

3. THE OSI MODEL

3.1 THE MODEL

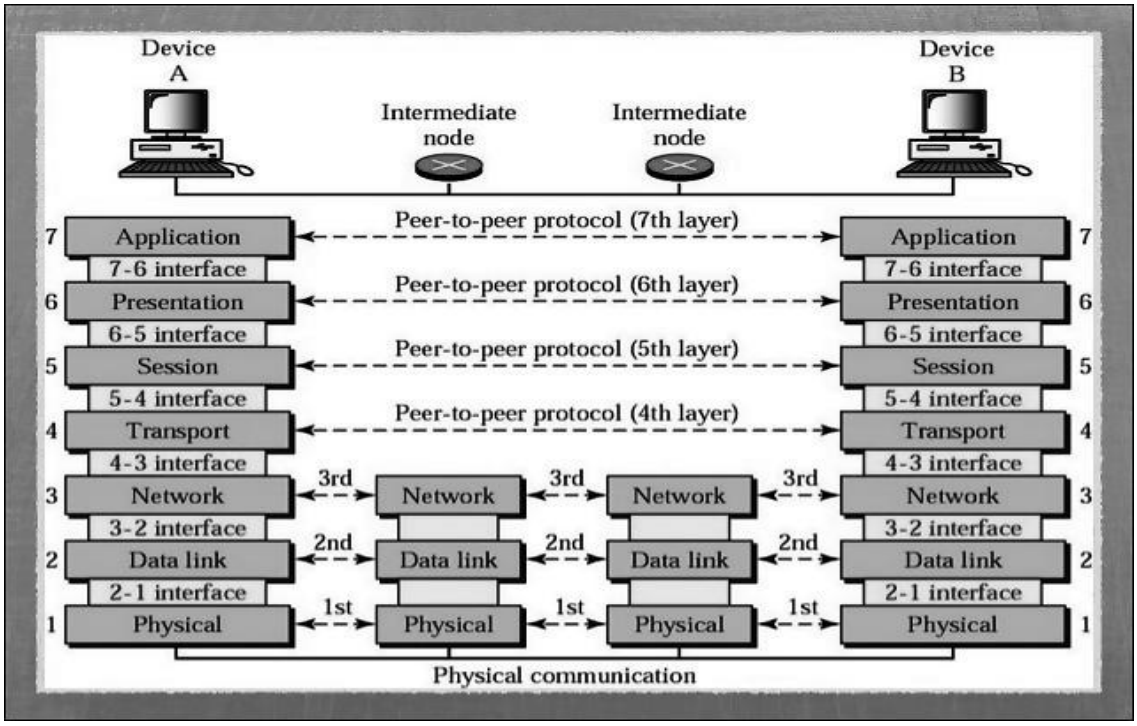
The Open Systems Interconnection model is a layered framework for the design of net work systems that allows for communication across all types of computer systems. It consists of seven separate but related layers, each of which defines a segment of the process of moving information across a network. (Figure 3.1) Understanding the fundamentals of the OSI model provides a solid basis for exploration of data communication.

Figure 3.1 The OSI model

Layered Architecture

The OSI model is built of seven ordered layers Physical(layer1), data link(layer2), Network(layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7). (Figure 3.2) shows the layers involved when a message is sent from device A to device B. As the message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes usually involve only the first three layers of the OSI model.

Figure 3.2 OSI layers



Peer-to-Peer

Within a single machine, each layer calls upon the services of the layer just below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4. Between machines, layer x on one machine communicates with layer x on another machine. This communication is governed by an agreed-upon series of rule and conventions called protocols. The processes on each machine that communicate aa given layer are called **peer-to-peer processes**. Communication between machines i therefore a peer-to-peer process using the protocols appropriate to a given layer. At the physical layer, communication is direct: Machine A sends a stream of bits to machine B.

At the higher layers, however, communication must move down through the layers on machine A, over to machine B, and then back up through the layers. Each layer in the sending machine adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below it. This information is added in the form of headers or trailers.

Headers are added to through message at layers 6, 5, 4, 3, and 2. A trailer is added at layer 2.

Interfaces between Layers

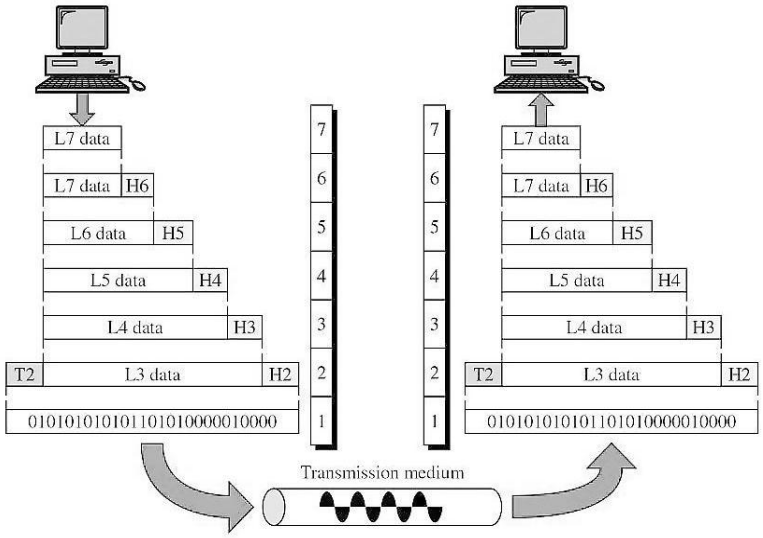
The passing of the data and network information down through the layers of the sending machine and back up through the layers of the receiving machine is made possible by an interface between each pair of adjacent layers. Each interface defines what information

and services a layer must provide for the layer above it. Well-defined interfaces and layer functions provide modularity to a network.

Organization of the Layers

The seven layers can be thought of as belonging to three subgroups. Layers 1, 2, and 3-physical, data link, and network-are the network support layers; they deal with the physical aspects of moving data from one device to another (such as electrical specifications, physical connections, physical addressing, and transport timing and reliability). Layers 5, 6, and 7—session, presentation, and application can he thought of as the user support layers; they allow interoperability among unrelated software systems. Layer 4, the transport layer, ensures end-to-end reliable data trans-mission while layer 2 ensures reliable transmission on a single link. The upper OSI layers are almost always implemented in software; lower layers are a combination of hardware and software, except for the physical layer, which is mostly hardware

Figure 3.3 An exchange using the OSI model



In Figure 3.3, which gives an overall view of the OSI layers, L7 data means the data unit at layer 7, L6 data means the data unit at layer 6, and so on. The process starts out at layer 7 (the application layer), then moves from layer to layer in descending sequential order. At each layer (except layers 7 and 1), a header is added to the data unit. At layer 2, a trailer is added as well. When the formatted data unit passes through the physical layer (laver1), it is changed into an electromagnetic signal and transported along a physical link.

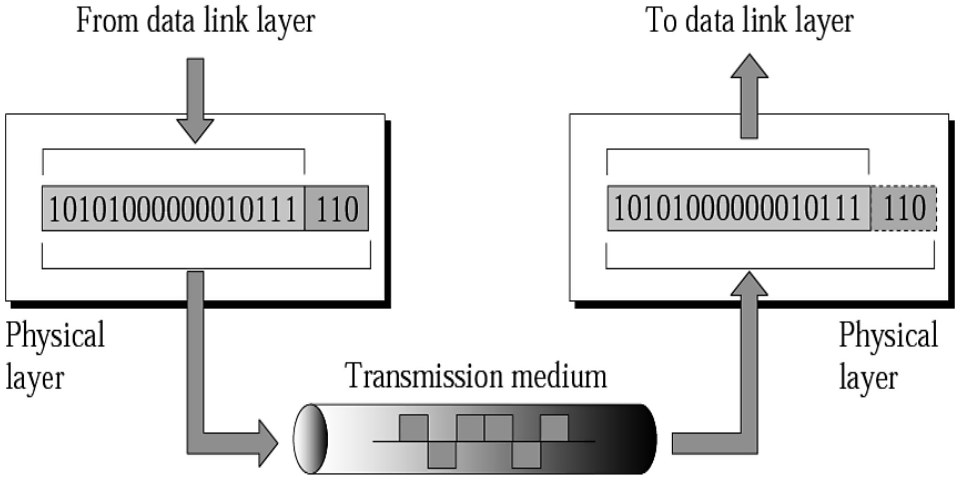
3.2 FUNCTIONS OF THE LAYERS

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

Physical Layer

The physical layer coordinates the functions required to transmit a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the inter-face and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur. Figure 3.4 shows the position of the physical layer with respect to the transmission medium and the data link layer.

Figure 3.4 Physical layer



The physical layer is concerned with the following

- **Physical characteristics of interfaces and media**. The physical layer defines the characteristics of the interface between the devices and the transmission medium.
- **Representation of bits.** The physical layer data consist of a stream of bits (sequence of 0s and 1s) without any interpretation. To be transmitted, bits must be encoded into signals-electrical or optical. The physical layer defines the type of encoding (how 0s and I s are changed to signals.
- **Data rate.** The **transmission rate**-the number of bits sent each second-is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.
- **Synchronization of bits**. The sender and receiver must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.
- **Line configuration**. The physical layer is concerned with the connection of devices to the medium. In a point-to-paint configuration, two devices are connected together through a dedicated link. In a multipoint configuration, a link is shared between several devices.
- **Physical topology.** The physical topology defines how devices are connected to make a network. Devices can be connected using a mesh topology (every device connected

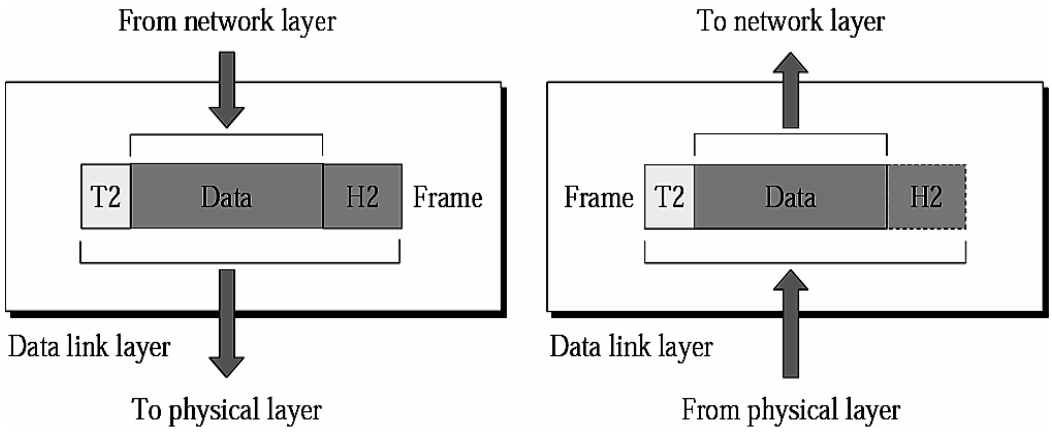
to every other device), a star topology (devices are connected through a central device), a ring topology (every device is connected to the next, forming a ring), or a bus topology (every device on a common link).

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

Data Link Layer

The **data link layer** transforms the physical layer, a raw transmission facility, to a reliable link and is responsible for **node-to-node** delivery. It makes the physical layer appear error free to the upper layer (network layer). Figure 3.5 shows the relationship of the data link layer to the network and physical layers.

Figure 3.5 Data link layer



Specific responsibilities of the data link layer include the following:

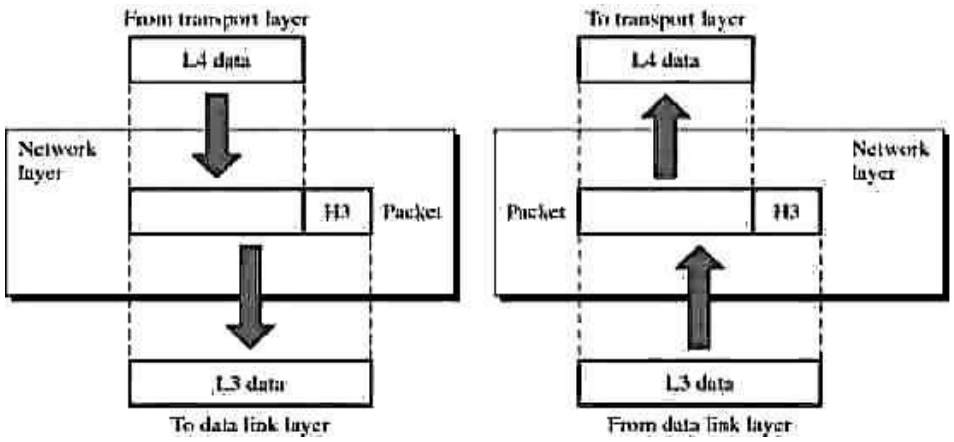
- **Framing**. The data link layer divides the stream of bits received from the network layer into manageable data units called **frames.**
- **Physical addressing**. If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the **physical address** of the sender **(source address)** and/or receiver **(destination address)** of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects one network to the next.
- **Flow control**. If the rate at which the data are absorbed by the receiver is less than the rate produced in the sender, the data link layer imposes a flow control mechanism to prevent overwhelming the receiver.

- **Error control**. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to prevent duplication of frames. Error control is normally achieved through a trailer added to the end of the frame.
- **Access control**. When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Network Layer

The **network layer** is responsible for the source-to-destination delivery of a packet possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination. If two systems are connected to the same link, there is usually no need for a net-work layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery. Figure 3.7 shows the relationship of the network layer to the data link and transport layers.

Figure 3.7 Network layer



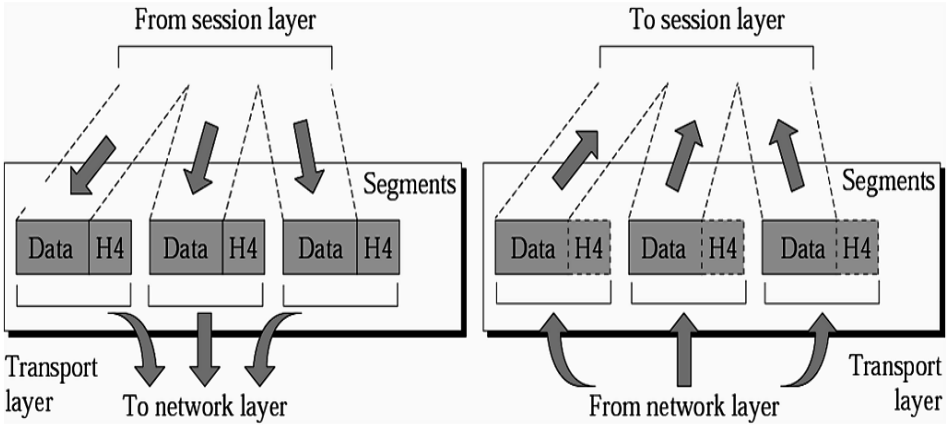
Specific responsibilities of the network layer include the following:

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

Transport Layer

The transport layer is responsible for **source-to-destination** (end-to-end) **delivery** of the entire message. Whereas the network layer oversees end-to-end delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level. Figure 3.8 shows the relationship of the transport layer to the net-work and session layers.

Figure 3.8 Transport layer



For added security, the transport layer may create a connection between the two end ports. By confining transmission of all packets to a single pathway, the transport layer has more control over sequencing, flow, and error detection and correction. Specific responsibilities of the transport layer include the following:

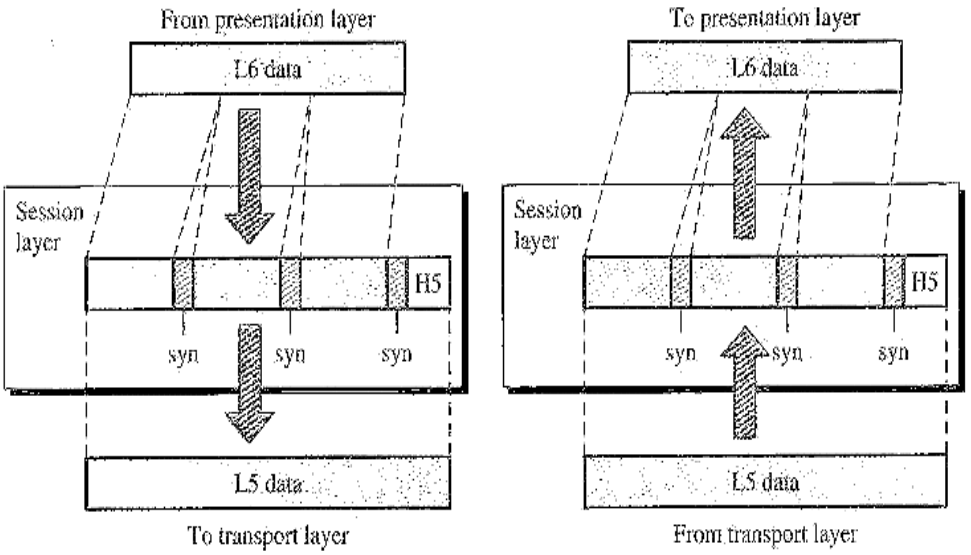
- **Service-point addressing**. Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header therefore must include a type of address called a service-point address (or **port address**). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly.** A message is divided into transmittable segments, each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in the transmission.
- **Connection control.** The transport layer can be either connectionless or connection-oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.

- **Flow control.** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- **Error control.** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed end to end rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without **error** (damage, loss, or duplication). Error correction is usually achieved through retransmission.

Session Layer

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

Figure 3.9 Session layer



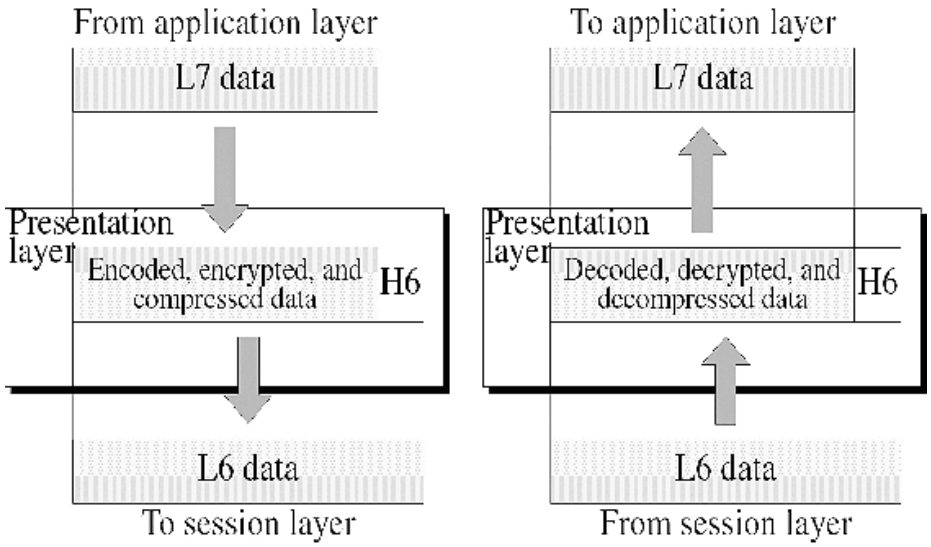
Specific responsibilities of the session layer include the following:

- **Dialog control**. The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place either in half-.duplex (one way at a time) or full-duplex (two ways at a time). For example, the dialog between a terminal connected to a mainframe can be half-duplex.
- **Synchronization.** The session layer allows a process to add checkpoints (synchronization points) into a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, retransmission begins at page 501. Pages 1 to 500 need not be retransmitted. Figure 3.9 illustrates the relationship of the session layer to the transport and presentation layers.

Presentation layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. The figure 3.10 shows the relationship between the presentation layer, application layer and the session layer.

Figure 3.10 Presentation layer



information should be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

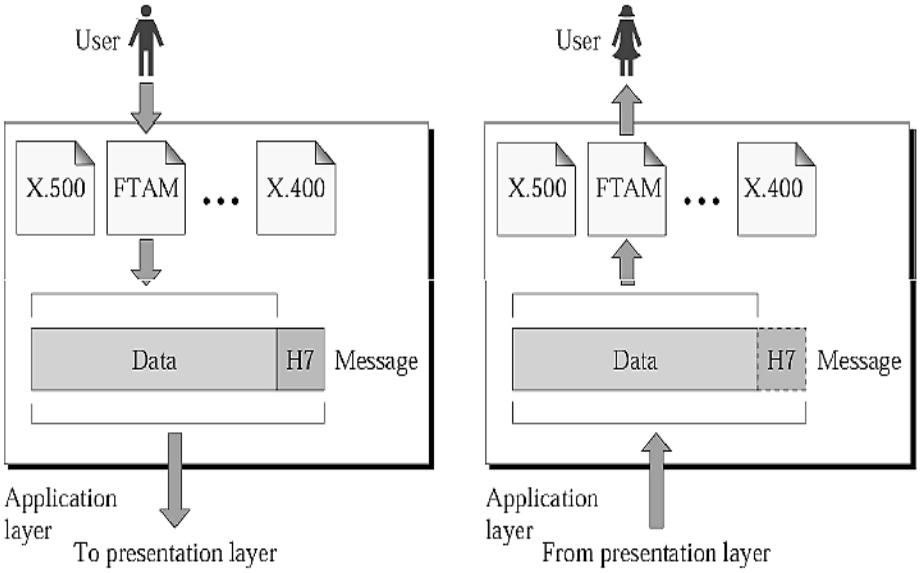
- Encryption. To carry sensitive information, a system must be able to assure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its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

Application Layer

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

Figure 3.11 shows the relationship of the application layer to the user and the presentation layer.

Figure 3.11 Application layer



Specific services provided by the application layer include the following:

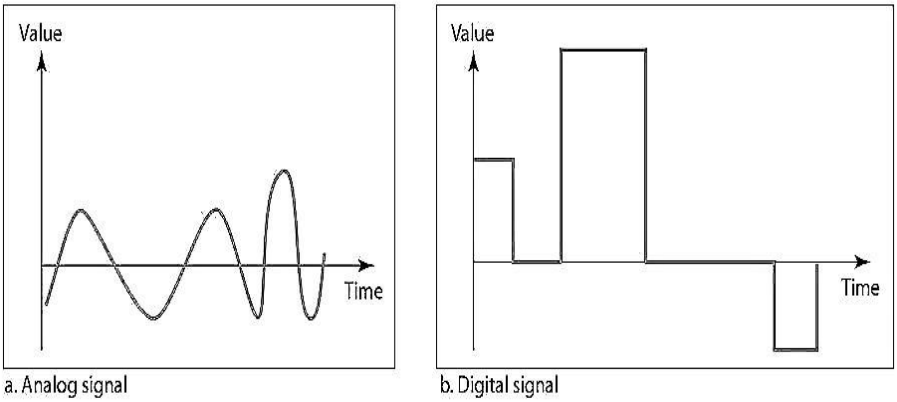
- **Network virtual terminal**. A network virtual terminal is a software version of a physical terminal and allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user' computer talks to the software terminal, which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminal and allows you to log on.
- **File transfer, access, and management (FTAM)**. This application allows a use to access files in a remote computer (to make changes or read data), to retrieve file from a remote computer; and to manage or control files in a remote computer.
- **Mail services.** This application provides the basis for e-mail forwarding an storage.
- **Directory services.** This application provides distributed database sources an access for global information about various objects and services.

4. SIGNALS

4.1 ANALOG AND DIGITAL

Both data and the signals that represent them can take either analog or digital form. **Analog** refers to something that is continuous – a set of specific points of data and all possible points between. **Digital** refers to something that is discrete – a set of specific points of data with no other points in between.

Figure 4.1 Comparison of analog and digital signals



Analog and Digital data

Data can be analog or digital. An example of analog data is the human voice. When somebody speaks a continuous wave is created in the air. This can be captured by the microphone and converted to an analog signal.

An example of digital data is data stored in the memory of a computer in the form of 0's an 1's.

Analog and digital signals

An analog signal is a continuous wave form that changes smoothly over time. As the wave moves from value A to value B it passes through and includes an infinite number of values along its path. A digital signal on the other hand is discrete. It can have only limited number of defined values as 1's and 0's.

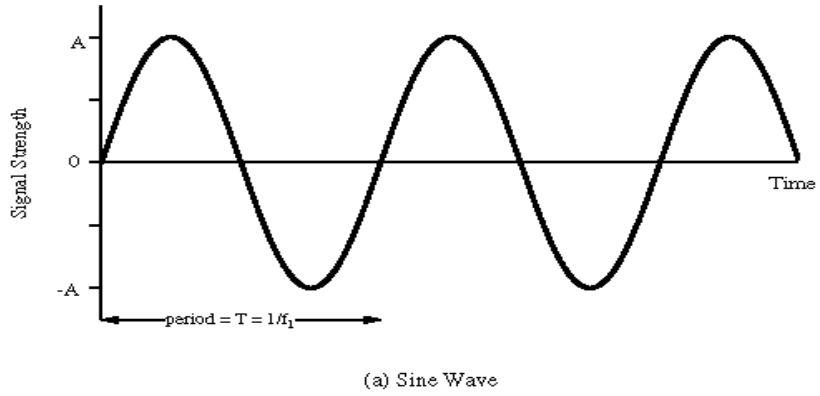
4.2.ANALOG SIGNALS

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

Simple Analog Signals

The sine wave is the most fundamental form of a periodic analog signal. Visualized as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow. Figure 4.2 shows a sine wave. Each cycle consists of a single arc

Figure 4.2 A sine wave



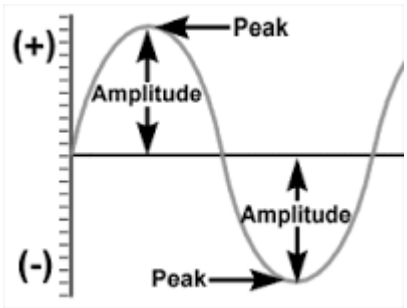
(a) Sine Wave

above the time axis followed by a single arc below it. Sine waves can be fully described by three characteristics: amplitude, period or frequency, and phase.

Amplitude

On a graph, the **amplitude** of a signal is the value of the signal at any point on the wave. It is equal to the vertical distance from a given point on the wave form to the horizontal axis. The maximum amplitude of a sine wave is equal to the highest value it reaches on the vertical axis (Figure 4.3).

Figure 4.3 Amplitude



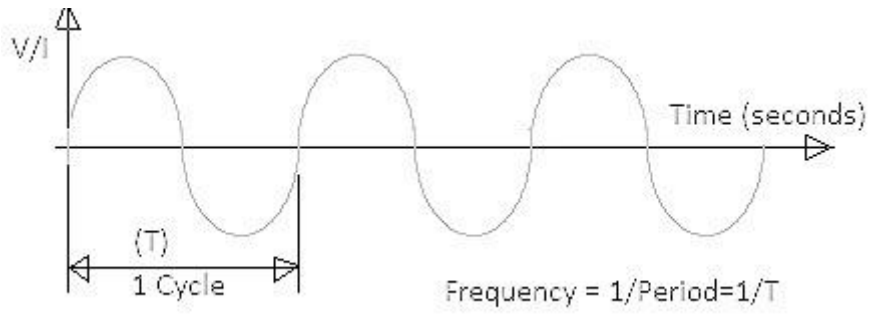
Amplitude is measured in either volts, amperes, or watts, depending on the type of signal. Volts refer to voltage; amperes refer to current; and watts refer to power.

Amplitude refers to the height of the signal. The unit for amplitude depends on the type of the signal. For electrical signals, the unit is normally volts, amperes, or watts.

Period and Frequency

Period refers to the amount of time, in seconds, a signal needs to complete one cycle. Frequency refers to the number of periods in one second. The frequency of a signal is its number of cycles per second. Figure 4.4 shows the concept of period and frequency.

Figure 4.4 Period and frequency



Unit of Period Period is expressed in seconds. The communications industry uses five units to measure period : second (s), **millisecond** (ms = 10-3 S), **microsecond** (μs= 10^{-6} s), **nanosecond** (ns = 10^{-9}), and **picosecond** (ps = 10^{-12} s). See Table 4.1.

Table 4.1 Units of periods

Unit	Equivalent
Second	1 S
Milliseconds (ms)	10^{-3} S
Microseconds (μs)	10^{-6} S
Nanoseconds (ns)	10^{-9} S
Picoseconds (ps)	10^{-12} S

Period is the amount of time it takes a signal to complete one cycle ; frequency is the number of cycles per second. Frequency and period are inverses of each other f=1/T and T=1/f.

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

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

Phase

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

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

Phase is measured in degrees or radians (360 degrees is 2π radian). A phase shift of 360 degrees corresponds to a shift of a complete period; a phase shift of 180 degrees corresponds to a shift of half a period; and a phase shift of 90 degrees corresponds to a shift of a quarter of a period (see Figure 4.7).

4.3 DIGITAL SIGNALS

In addition to being represented by an analog signal, data also can be represented by a digital signal. For example, a 1 can be encoded as a positive voltage and a 0 as zero voltage (Figure 4.5).

Figure 4.5 A digital signal

Bit Interval and Bit Rate

Most digital signals are aperiodic and, thus, period or frequency is not appropriate. Two new terms, bit interval (instead of period) and bit rate (instead of frequency) are used to describe digital signals. the**bit interval** is the time required to send one single bit. The **bit rate** is the number of bit intervals per second. This means that the bit rate is the number of bits sent in one second, usually expressed in **bits per second (bps).**

Decomposition of a Digital Signal

A digital signal can be decomposed into an infinite number of simple sine waves called **harmonics,** each with a different amplitude, frequency, and phase. This means that when we send a digital signal along a transmission medium, we are sending an infinite number of simple signals. To receive and exact replica of the digital signal, all of the frequency components must be faithfully transferred through the transmission medium. If some of the components are not passed through the medium, corruption of the signal at the receiver is the result. Since no practical medium (such as a cable) is capable of transferring the entire range of frequencies, we always have corruption.

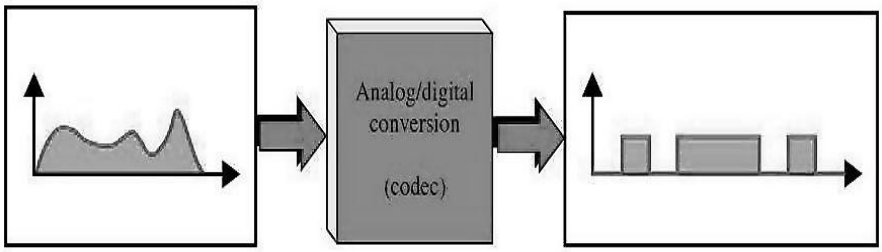
Although the frequency spectrum of a digital signal contains an infinite number of frequencies with different amplitudes, if we send only those components whose amplitudes are significant (able an acceptable threshold), we can still recreate the digital signal with reasonable accuracy at the receiver (minimum distortion). We call this part of the infinite spectrum the significant spectrum, and its bandwidth the significant bandwidth.

5. ENCODING AND MODULATING

5.1 ANALOG-TO-DIGITAL CONVERSION

We sometimes need to digitize an analog signal. For example, to send human voice a long distance, we need to digitize it since digital signals are less prone to noise. This is called an analog-to-digital conversion or digitizing an analog signal. This requires a reduction of the potentially infinite number of values in an analog message; so that they can be represented as a digital stream with a minimum loss of information. Figure 5.1 shows the analog-to-digital converter, called a codec (coder-decoder).

Figure 5.1 Analog-to-digital conversion



In analog-to-digital conversion, we are representing the information contained in a continuous wave form as a series of digital pulses (ls or Os).

The structure of the transporting signal is not the problem. Instead, the problem is how to translate information from an infinite number of values to a discrete number of values without sacrificing sense 1.yr quality.

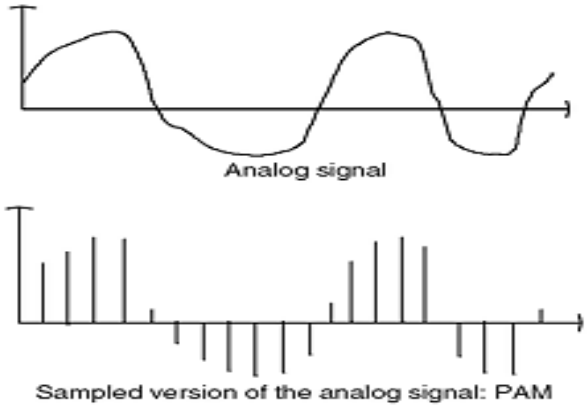
Pulse Amplitude Modulation (PAM)

The first step in analog-to-digital conversion is called **pulse amplitude modulation (PAM).** This technique takes an analog signal, samples it, and generates a series of pulses based on the results of the sampling. The term sampling means measuring the amplitude of the signal at equal intervals.

The method of sampling used in PAM is more useful to other areas of engineering than it is to data communication. However, PAM is the foundation of an important analog-to-digital conversion method called **pulse code modulation (PCM).**

In PAM, the original signal is sampled at equal intervals as shown in Figure 5.2. PAM uses a technique called sample and hold. At a given moment, the signal level is read, then held briefly. The sampled value occurs only instantaneously in the actual wave form, but is generalized over a still short but measurable period in the PAM result

Figure 5.2 PAM



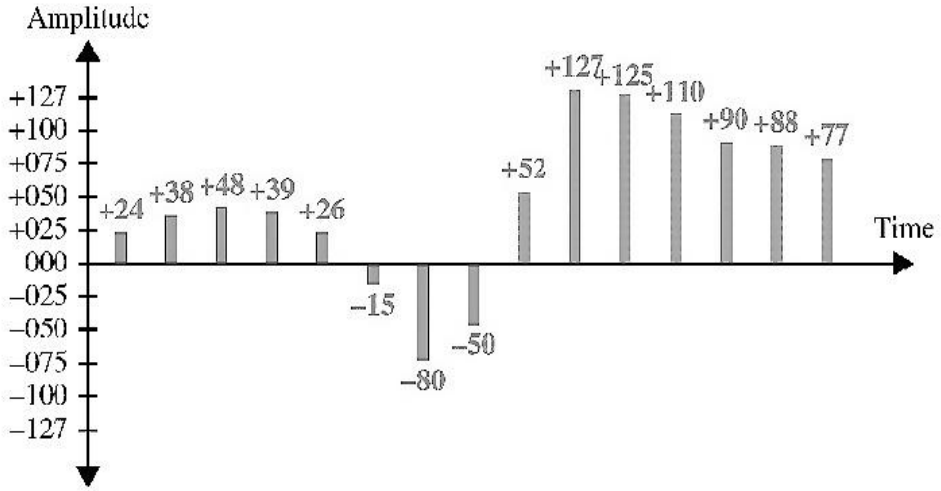
The reason PAM is not useful to data communications is that, although it translates the original wave form to a series of pulses, these pulses are still of any amplitude (still an analog signal, not digital). To make them digital, we must modify them by using **pulse code modulation (PCM).**

Pulse amplitude modulation (PAM) has some applications, but it is not used by itself in data communication. However, it is the first step in another very popular conversion method called pulse code modulation (PCM).

Pulse Code Modulation (PCM)

PCM modifies the pulses created by PAM to create a completely digital signal. To do so, PCM first quantizes the PAM pulses. Quantization is a method of assigning integral values in a specific range to sampled instances. The result of quantization is presented in Figure 5.3.

Figure 5.3 Quantized PAM Signal



The binary digits are then transformed into a digital signal using one of the digital-to-analog techniques. The result of the pulse code modulation of the original signal encoded finally into a unipolar signal.

The binary digits are then transformed into a digital signal using one of the digital to digital encoding techniques. The result of the pulse code modulation of the original signal encoded finally into a unipolar signal.

PCM is actually made up of four separate processes : PAM, quantization, binary encoding, and digital – to – digital encoding.

Sampling Rate

As you can tell from the preceding figures, the accuracy of any digital reproduction of an analog signal depends on the number of samples taken. Using PAM and PCM, we can reproduce the wave form exactly by taking infinite samples, or we can reproduce the barest generalization of its direction of change by taking three samples. Obviously we prefer to find a number somewhere between these two extremes. So the question is, How many samples are sufficient?

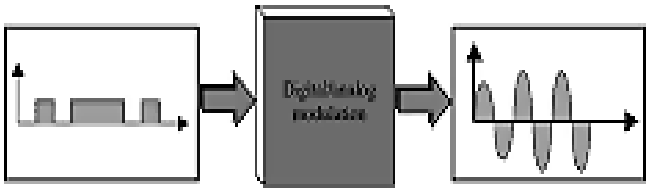
Actually, it requires remarkably little information for the receiving device to reconstruct an analog signal. According to the **Nyquist theorem,** to ensure the accurate reproduction of an original analog signal using PAM, the sampling rate must be at least twice the highest frequency of the original signal. So if we want to sample telephone voice with maximum frequency 4000 Hz, we need a sampling rate of 8000 samples per second.

A sampling rate of twice the frequency of x HZ means that the signal must be sampled every ½ x seconds. .

5.3 DIGITAL-TO-ANALOG CONVERSION

Digital-to-analog conversion or digital-to-analog modulation is the process of changing one of the characteristics of an analog signal based on the information in a digital signal (0s and 1s). When you transmit data from one computer to another across a public access phone line, for example, the original data are digital, but because telephone services use analog signals, the data must be converted. The digital data must be modulated on an analog signal that has been manipulated to look like two distinct values corresponding to binary 1 and binary 0. Figure 5.4 shows the relationship between the digital information, the digital-to-analog modulating hardware, and the resultant analog signal.

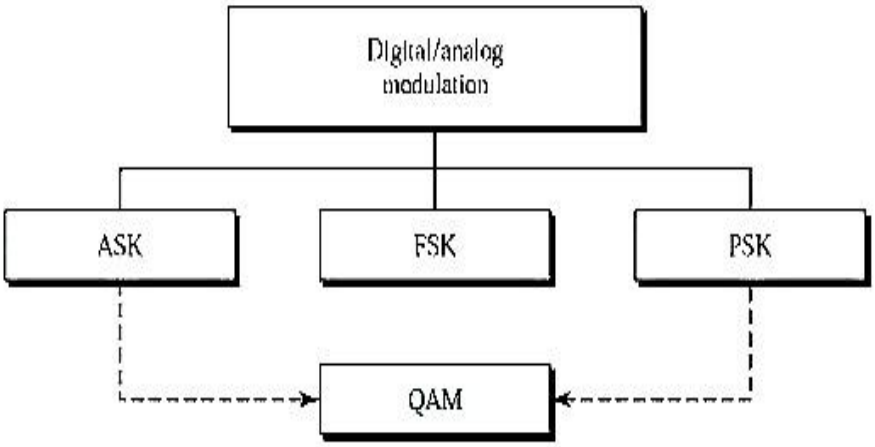
Figure 5.4 Digital-to-analog modulation



Of the many mechanisms for digital-to-analog modulation, we will discuss only those most useful for data communications.

A sine wave is defined by three characteristics: **amplitude, frequency, and phase**. When we vary any one of these characteristics, we create a second version of that wave. If we then say that the original wave represents binary 1, the variation can represent binary 0, or vice versa. So, by changing one aspect of a simple electrical signal back and forth, we can use it to represent digital data. Any of the three characteristics listed above can be altered in this way, giving us at least three mechanisms for modulating digital data into an analog signal: amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). In addition, there is a fourth (and better) mechanism that combines changes in both amplitude and phase called quadrature amplitude modulation (QAM). QAM is the most efficient of these options aid is the mechanism used in all modern modems. (Figure 5.5).

Figure 5.5 Types of digital-to-analog modulation



Aspects of Digital-to-Analog Conversion

Before we discuss specific methods of digital-to-analog modulation, two basic issues must be defined: bit/baud rate and carrier signal.

Bit Rate and Baud Rate

Two terms used frequently in data communication are bit rate and baud rate. Bit rate is the number of bits transmitted during one second. Baud rate refers to the number of signal units per second that are required to represent those bits. In discussions of computer efficiency, the bit rate is the more important-we want to know how long it takes to process each piece of information. In data transmission, however, we are more concerned with how efficiently we can move those data from place to place, whether in pieces or blocks. The fewer signal units required, the more efficient the system and the less bandwidth required to transmit more bits; so we are more concerned with baud rate. The baud rate determines the bandwidth required to send the signal. Bit rate equals the baud rate times the number of bits represented by each signal unit. The baud rate equals the bit rate divided by the number of bits represented by each signal shift. Bit rate is always greater than or equal to the baud rate.

Bit rate is the number of bits per second. Baud rate is the number of signal units per second. Baud rate is less than or equal to the bit rate.

. Similarly, the number of bauds determines the required bandwidth, not the number of bits.

Carrier Signal

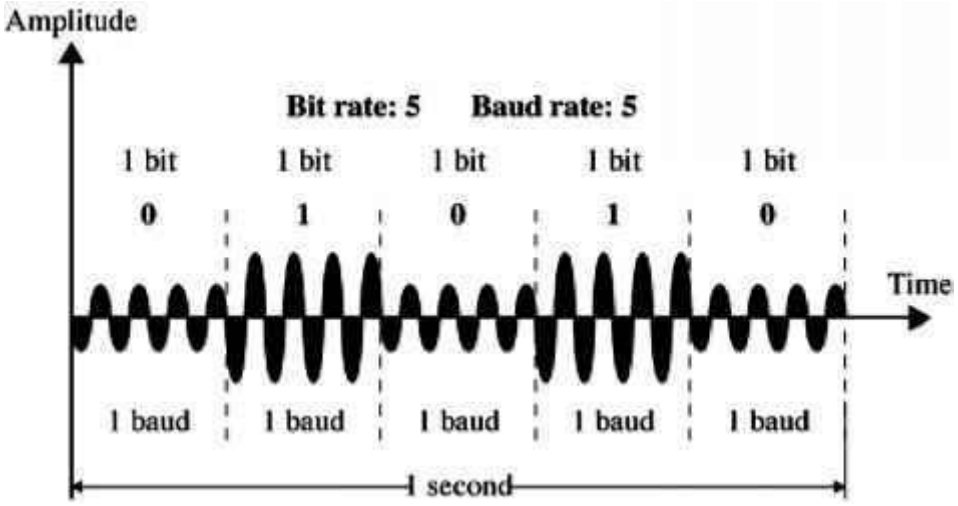
In analog transmission, the sending device produces a high-frequency signal that acts as a basis for the information signal. This base signal is called the carrier signal or car-rier frequency. The receiving device is tuned to the frequency of the carrier signal that it expects from the sender. Digital information is then modulated on the carrier signal by modifying one or more of its characteristics (amplitude, frequency, phase). This kind of modification is called modulation (or shift keying) and the information signal is called a modulating signal.

Amplitude Shift Keying (ASK)

In amplitude shift keying (ASK), the strength of the carrier signal is varied to represent binary 1 or 0. Both frequency and phase remain constant while the amplitude changes. Which voltage represents 1 and which represents 0 is left to the system designers. A bit duration is the period of time that defines one bit. The peak amplitude of the signal during each bit duration is constant and its value depends on the bit (0 or 1). The speed of transmission using ASK is limited by the physical characteristics of the transmission medium. Figure 5.6 gives a conceptual view of ASK.

.

Figure 5.6 ASK



Unfortunately, ASK transmission is highly susceptible to noise interference. The term noise refers to unintentional voltages introduced onto a line by various phenomena such as heat or electromagnetic induction created by other sources. These unintentional voltages combine with the signal to change the amplitude. A 0 can be changed to 1, and a 1 to 0. You can see how noise would be especially problematic for ASK, which relies solely on

amplitude for recognition. Noise usually affects the amplitude; therefore, ASK is the modulating method most affected by noise.

A popular ASK technique is called on – off – keying (OOK). In OOK one of the bit values is represented by no voltage. The advantage is a reduction in the amount of energy required to transmit information.

Bandwidth for ASK

As you will recall from Chapter 4, the bandwidth of a signal is the total range of frequencies occupied by that signal. When we decompose an ASK – modulated signal, we get a spectrum of many simple frequencies. However, the most significant ones are those between f_c-N_{baud} /2 and fc+ N_{baud}/2 with the carrier frequency, $f_{,c9}$ at the middle.

Frequency Shift Keying (IFSK)

In frequency shift keying (FSK), the frequency of the carrier signal is varied to represent binary 1 or 0. The frequency of the signal during each bit duration is constant and its value depends on the bit (0 or 1): both peak amplitude and phase remain constant. .

Bandwidth for FSK

Although FSK shifts between two carrier frequencies, it is easier to analyze as two coexisting frequencies. We can say that the FSK spectrum is the combination of two ASK spectra centered around f_c0 and f_{c1}. The bandwidth required for FSK transmission is equal to the baud rate of the signal plus the frequency shift (difference between the two carrier frequencies): BW = (f_{c1}-f_{c0})+ N_{baud}.

Although there are only two carrier frequencies, the process of modulation produces a compos signal that is a combination of many simple signals, each with a different frequency.

Phase Shift Keying (PSK)

In phase shift keying (PSK), the phase of the carrier is varied to represent binary 1 or 0. Both peak amplitude and frequency remain constant as the phase changes. For example, if we start with a phase of 0 degrees to represent binary 0, then we can change the phase to 180 degrees to send binary 1. The phase of the signal during each bit duration is constant and its value depends on the bit (0 or 1). Figure 5.29 gives a conceptual view of PSK.

The above method is often called 2-PSK, or binary PSK, because two different phases (0 and 180 degrees) are used. Figure 5.30 makes this point clearer by showing the relationship of phase to bit value. A second diagram, called a constellation or phase-state diagram, shows the same relationship by illustrating only the phases.

PSK is not susceptible to the noise degradation that affects ASK, nor to the band-width limitations of FSK. This means that smaller variations in the signal can be detected reliably by the receiver. Therefore, instead of utilizing only two variations of a signal, each representing one bit, we can use four variations and let each phase shift represent two bits (see Figure 5.31).

The constellation diagram for the signal in Figure 5.31 is given in Figure 5.32. A phase of 0 degrees now represents 00; 90 degrees represents 01; 180 degrees represents 10; and 270 degrees represents 11. This technique is called 4-PSK or Q-PSK. The pair of bits

represented by each phase is called a dibit. We can transmit data two times as fast using 4-PSK as we can using 2-PSK.

We can extend this idea to 8-PSK. Instead of 90 degrees, we now vary the signal by shifts of 45 degrees. With eight different phases, each shift can represent three bits (one tribit) at a time. (As you can see, the relationship of number of bits per shift to numberof phases is a power of two. When we have four possible phases, we can send two bits at a time-22 equals 4. When we have eight possible phases, we can send three bits at a time-23 equals 8). Figure 5.33 shows the relationships between the phase shifts and the tribits each one represents: 8-PSK is three times faster than 2-PSK.

Bandwidth for PSK

The minimum bandwidth required for PSK transmission is the same as that required for ASK transmission—and for the same reasons. As we have seen, the maximum bit rate in PSK transmission, however, is potentially much greater than that of ASK. So while the maximum baud rates of ASK and PSK are the same for a given bandwidth, PSK bit rates using the same bandwidth can be two or more times greater (see Figure 5.34).

Quardrature Amplitude Modulation (QAM)

PSK is limited by the ability of the equipment to distinguish small differences in phase. This factor limits its potential bit rate.

Bandwidth limitations make combinations of FSK with other changes practically useless. Then we could have x variations in phase and y variations in amplitude, giving us x times y possible variations and the corresponding number of bits per variation. Quadrature amplitude modulation (QAM) does just that. The term quadrature is derived from the restrictions required for minimum performance and is related to trigonometry.

Quadrature amplitude modulation (QAM) means combining ASK and PSK in such a way that we have maximum contrast between each bit, dibit, tribit, quadbit, and so on.

Possible variations of QAM are numerous. Theoretically, any measurable number of changes in amplitude can be combined with any measurable number of changes in phase. Figure 5.35 shows two possible configurations, 4-QAM and 8-QAM. In both cases, the number of amplitude shifts is fewer than the number of phase shifts. Because amplitude changes are susceptible to noise and require greater shift differences than do phase changes, the number of phase shifts used by a QAM system is always larger than the number of amplitude shifts. In general, therefore, a second advantage of QAM over ASK is its lower susceptibility to noise.

Bandwidth for QAM

The minimum bandwidth required for QAM transmission is the same as that required for ASK and PSK transmission. QAM has the same advantages as PSK over ASK.

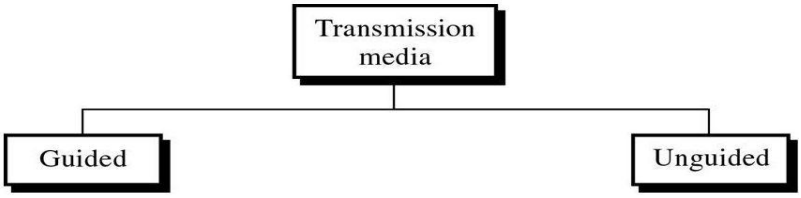
Bit/Baud Comparison

Assuming that an FSK signal over voice-grade phone lines can send 1200 bits per second, the bit rate is 1200 bps. Each frequency shift represents a single bit; so it requires 1200 signal elements to send 1200 bits. Its baud rate, therefore, is also 1200 bps. Each signal variation in an 8-QAM system, however, represents three bits. So a bit rate of 1200 bps, using 8-QAM, has a baud rate of only 400.

UNIT- III
6. TRANSMISSION MEDIA

Electromagnetic energy, a combination of electrical and magnetic fields vibrating in relation to each other, includes power, voice, radio waves, infrared light, visible light, ultraviolet light, and X, gamma, and cosmic rays. Each of these constitutes a portion of the electromagnetic spectrum (Figure 6.1). Not all portions of the spectrum are currently usable for telecommunications, however, and media to harness those that are usable are limited to a few types. Voice-band frequencies are generally trans-mitted as current over metal cables, such as twisted-pair or coaxial cable. Radio frequencies can travel through air or space but require specific transmitting and receiving mechanisms. Visible light, the third type of electromagnetic energy currently used for communications, is harnessed using fiber-optic cable.

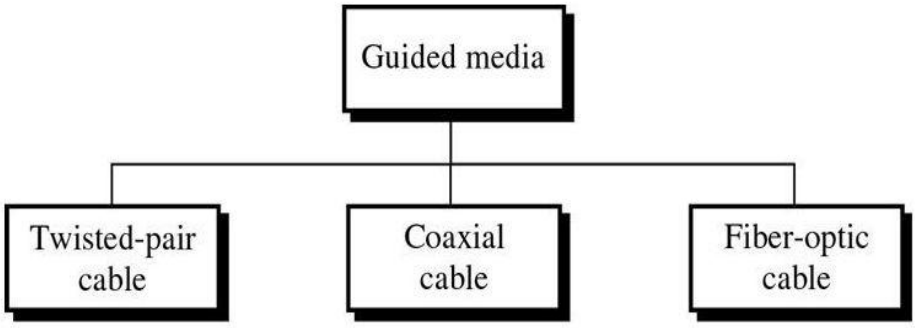
Figure 6.1 Classes of transmission media



6.1 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 (Figure 6.2). 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 electrical current. Optical fiber is a glass or plastic cable that accepts and transports signals in the form of light.

Figure 6.2 Categories of guided media



Twisted-Pair Cable

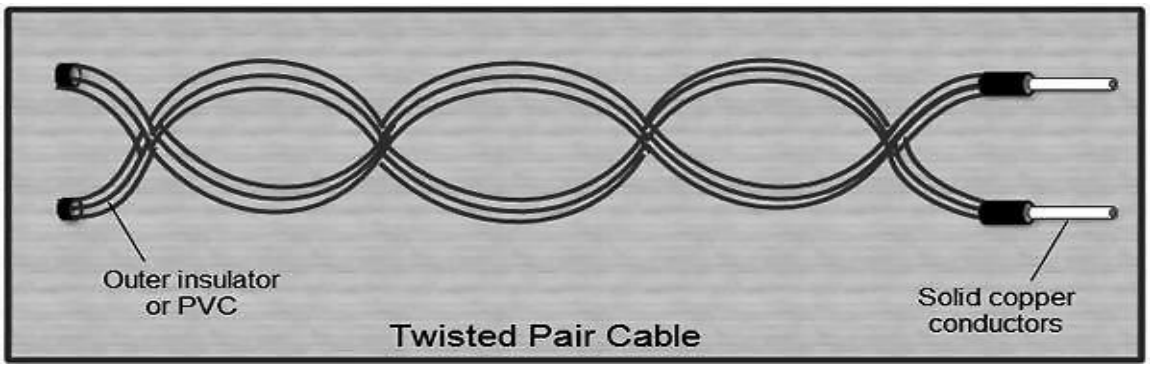
Twisted-pair cable comes in two forms: unshielded and shielded.

Unshielded Twisted-Pair (UTP) Cable

Unshielded twisted-pair (UTP) cable is the most common type of telecommunication medium in use today. Although most familiar from its use in telephone systems, its frequency range is suitable for transmitting both data and voice. A twisted pair consists of two conductors (usually copper), each with its own colored plastic insulation. The plastic insulation is color-banded for identification (Figure 6.3). Colors are used both to identify the specific conductors in a cable and to indicate which wires belong in pairs and how they relate to other pairs in a larger bundle.

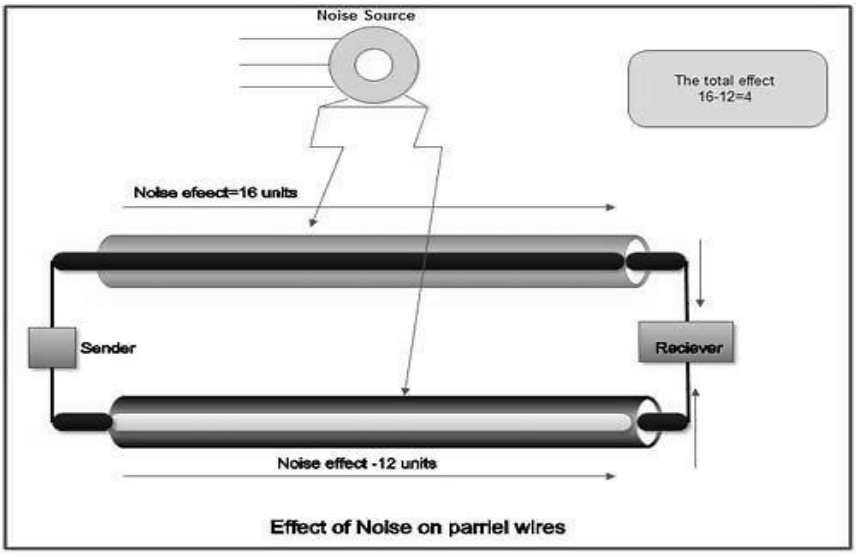
A twisted pair consists of two conductors each surrounded by an insulating material.

Figure 6.3 Twisted pair cable



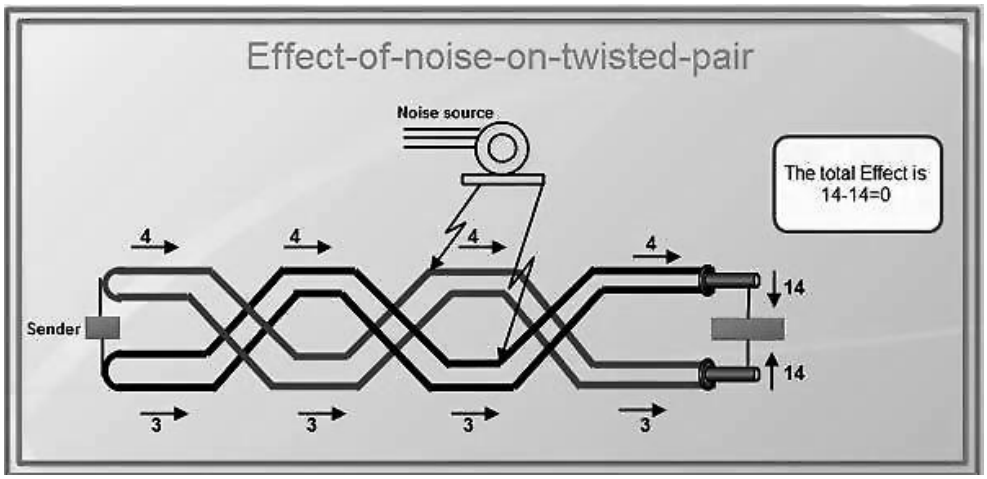
In the past, two parallel flat wires were used for communication. However, electro-magnetic interference from devices such as a motor can create noise over those wires. If the two wires are parallel, the wire closest to the source of the noise gets more interference and ends up with a higher voltage level than the re farther away, which results in an uneven load and a damaged signal.(Figure 6.4)

Figure 6.4 Effect of noise on parallel lines



If however, the two wires are twisted around each other at regular intervals (between 2 and 12 twists per foot), each wire is closer to the noise source for half the time and farther away for the other half. With twisting, therefore, the cumulative effect of the interference is equal on both wires (Figure 6.5). Each section of wire has a "load" of 4 when it is on the top of the twist and 3 when it is on the bottom. The total effect of the noise at the receiver is therefore 0 (14 - 14). Twisting does not always eliminate the impact of noise, but it does significantly reduce it.

Figure 6.5 Effect of noise on twisted pair lines



Advantages of UTP are its cost and ease of use. UTP is cheap, flexible, and easy to install. Higher grades of UTP are used in many LAN technologies, including Ethernet and Token Ring.

The Electronic Industries Association (EIA) has developed standards to grade UTP cable y quality. Categories are determined by cable quality, with 1 as the lowest and 5 as the highest. Each EIA category is suitable for certain uses and not for others:

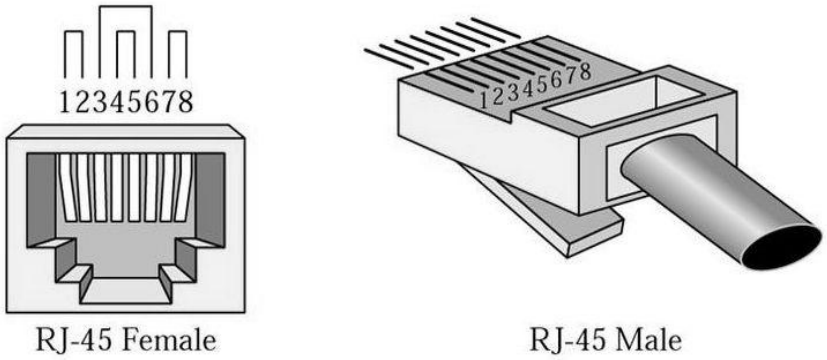
- **Category 1.** The basic twisted-pair cabling used in telephone systems. This level of quality is fine for voice but inadequate for all but low-speed data communication.
- **Category 2**. The next higher grade, suitable for voice and for data transmission of up to 4 Mbps.
- **Category 3**. Required to have at least three twists per foot and can be used for data transmission of up to 10 Mbps. It is now the standard cable for most telephone systems.
- **Category 4.** Must also have at least three twists per foot as well as other conditions to bring the possible transmission rate to 16 Mbps.
- Category 5. Used for data transmission up to 100 Mbps.

UTP Connectors

UTP is most commonly connected to network devices via a type of snap-in plug like that used with telephone jacks. Connectors are either male (the plug) or female (the receptacle). Male connectors snap into female connectors and have a repressible tab (called a key) that locks them in place. Each wire in a cable is attached to one conductor (or pin) in the connector. The most frequently used of these plugs is an RJ45 connector with eight conductors, one for each wire of four twisted pairs. (Figure 6.6)

.

Figure 6.6 UTP connection

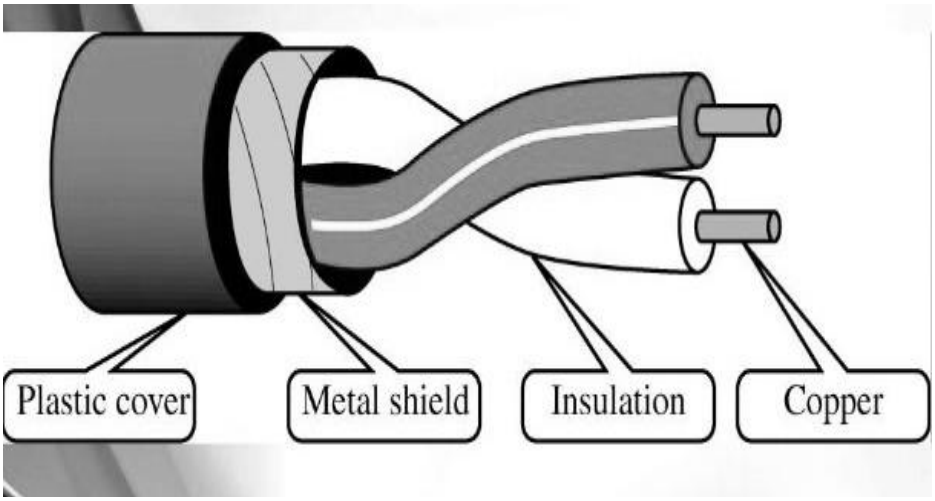


4-conductor

Shielded Twisted-Pair (STP) Cable

Shielded twisted-pair (STP) cable has a metal foil or braided-mesh covering that encases each pair of insulated conductors (Figure 6.7). The metal casing prevents the penetration of electromagnetic noise. It also can eliminate a phenomenon called crosstalk, which is the undesired effect of one circuit (or channel) on another circuit (or channel). It occurs when one line (acting as a kind of receiving antenna) picks up some of the signals traveling down another line (acting as a kind of sending antenna). This effect can be experienced during telephone conversations when one can hear other conversations in the background. Shielding each pair of a twisted-pair cable can eliminate most crosstalk

Figure 6.7 Shielded twisted pair cable

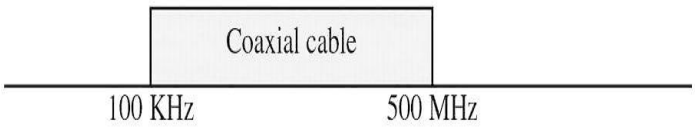


STP has the same quality considerations and uses the same connectors as UTP, but the shield must be connected to a ground. Materials and manufacturing requirements make STP more expensive than UTP but less susceptible to noise.

Coaxial Cable

Coaxial cable (or coax) carries signals of higher frequency ranges than twisted-pair cable (Figure 6.9), 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 (also usually copper). 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.(figure 6.8)

Figure 6.8 Frequency range of coaxial cable



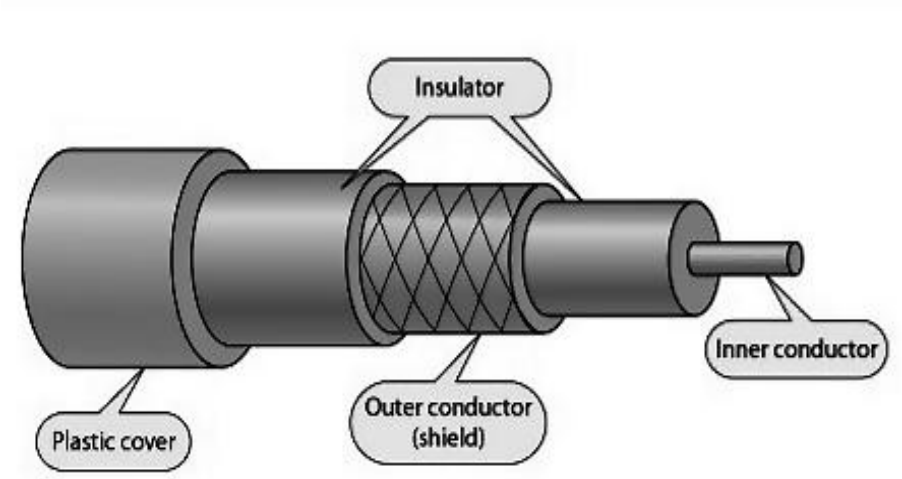
Coaxial Cable standards

Standards Different coaxial cable designs 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 RG ratings is adapted for a specialized function. The following are a few of the common ones:

- **RG-8**. Used in thick Ethernet.
- **RG-9.** Used in thick Ethernet.
- **RG-11.** Used in thick Ethernet
- **RG-58.** Used in thin Ethernet.
- **RG-59.** Used for TV.

Figure 6.9 Coaxial cable



Coaxial Cable Connectors

Over the years, a number of connectors have been designed for use with coaxial cable, usually by manufacturers seeking specific solutions to specific product requirements. A few of the most widely used connector designs have become standardized. The most common of these is called a barrel connector because of its shape. Of the barrel connectors, the most popular is the bayonet network connector (BNC), which pushes on and locks into place with a half turn. Other types of barrel connectors either screw together, and thus require more effort to install, or push on without locking, which is less secure. Generally, a cable terminates in a male connector that plugs or screws onto a corresponding female connector attached to the device. All coaxial connectors have a single pin protruding from the center of the male connector that slides into a ferrule in the female connector. Coaxial connectors are familiar from cable TV and VCR hookups, which employ both threaded and slip-on styles.

Two other commonly used types of connectors are T-connectors and terminators. A T-connector (used in thin Ethernet) allows a secondary cable or cables to branch off from a main line. A cable running from a computer, for example, can branch to connect several terminals. Terminators are required for bus topologies where one main cable acts as a backbone with branches to several devices but does not itself terminate in a device. If the main cable is left unterminated, any signal transmitted over the line echoes back and interferes with the original signal. A terminator absorbs the wave at the end and eliminates echo-back.

Optical Fiber

Up until this point, we have discussed conductive (metal) cables that transmit signals in the form of current. Optical fiber, on the other hand, 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.

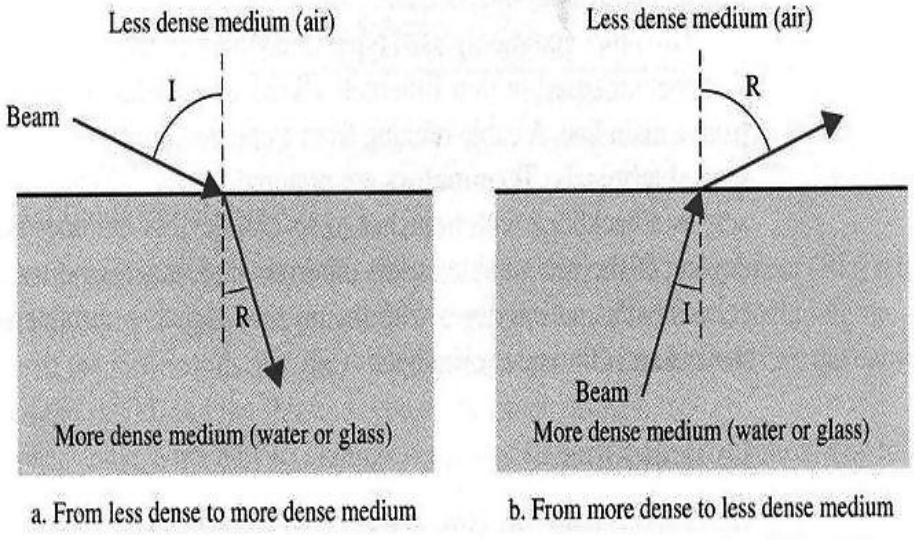
The Nature of Light

Light is a form of electromagnetic energy. It travels at its fastest in a vacuum: 300,000 kilometers/second (approximately 186,000 miles/second). The speed of light depends on the density of the medium through which it is traveling (the higher the density, the slower the speed).

Light, a form of electromagnetic energy, travels at 300,000 kilometers/second, or approximately 186,000 miles/second, in a vacuum. This speed decreases as the medium through which the light travels becomes denser.

Refraction 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 (more or less dense) substance, its speed changes abruptly, causing the ray to change direction. This change is called refraction. A straw sticking out of a glass of water appears bent, or even broken, because the light by which we see it changes direction as it moves from the air to the water.

Figure 6.10 Refraction

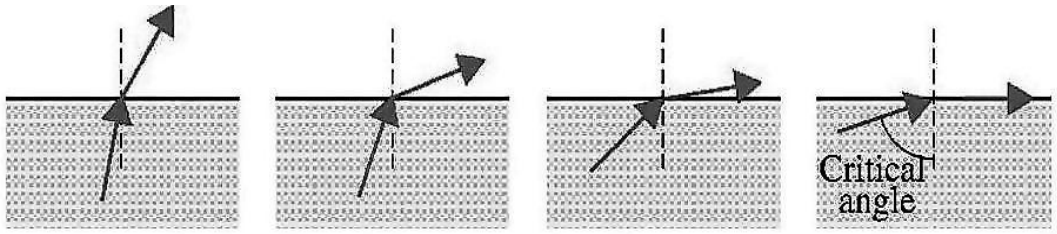


The direction in which a light ray is refracted depends on the change in density encountered. A beam of light moving from a less dense into a more dense medium is bent toward the vertical axis (Figure 6.10). The two angles made by the beam of light in relation to the vertical axis are called I, for incident, and R, for refracted. In Figure 6.10a, the beam travels from a less dense medium into a more dense medium. In this case, angle R is smaller than angle I. In Figure 6.10b, however, the beam travels from a more dense medium into a less dense medium. In this case, the value of I is smaller than the value of R. In other words, when light travels into a more dense medium, the angle of incidence is greater than the angle of refraction; and when light travels into a less dense medium, the angle of incidence is less than the angle of refraction.

Critical Angle Now examine Figure 6.11. Once again we have a beam of light moving from a more dense into a less dense medium. In this example, however, we gradually increase the angle of incidence measured from the vertical. As the angle of incidence increases, so does

the angle of refraction. It, too, moves away from the vertical and closer and closer to the horizontal.

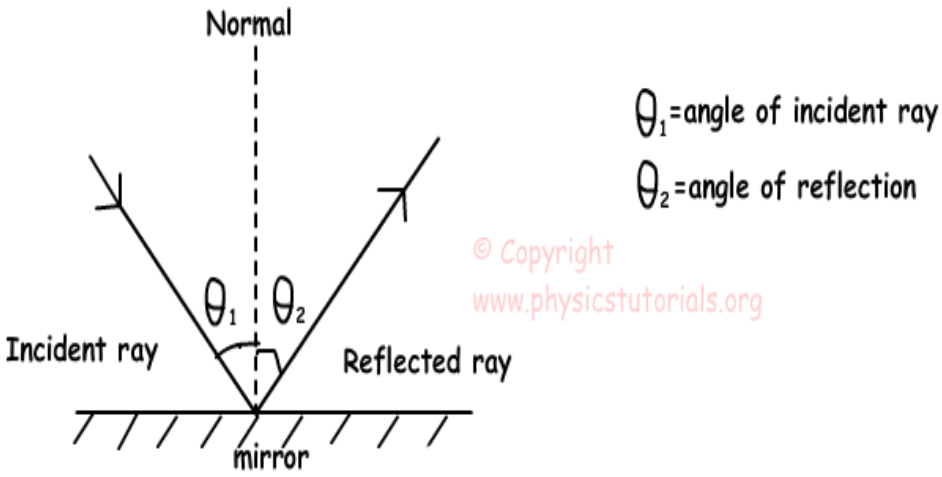
Figure 6.11 Critical Angle



At some point in this process, the change in the incident angle results in a refracted angle of 90 degrees, with the refracted beam now lying along the horizontal. The incident angle at this point is known as the critical angle.

Reflection. When the angle of incidence becomes greater than the critical angle, a new phenomenon occurs called reflection (or, more accurately, complete reflection, because some aspects of reflection always coexist with refraction). Light no longer passes into the less dense medium at all. In this case, the angle of incidence is always equal to the angle of reflection (Figure 6.13).

Figure 6.12 Reflection



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. Information is encoded onto a beam of light as a series of on-off flashes that represent 1 and 0 bits.

Propagation Modes

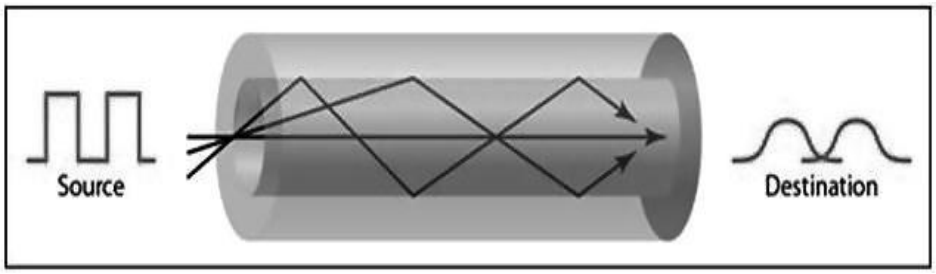
Current technology supports two modes for propagating light along optical channels, each requiring fiber with different physical characteristics: multimode and single mode. Multimode, in turn, can be implemented in two forms: step-index or graded-index.

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.

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 to a lower density that alters the angle of the beam's motion. The term step-index refers to the suddenness of this change.

Figure 6.13 shows various beams (or rays) traveling through a step-index fiber. Some beams in the middle travel in straight lines through the core and reach the destination without reflecting or refracting. Some beams strike the interface of the core and cladding at an angle smaller than the critical angle; these beams penetrate the cladding and are lost. Still others hit the edge of the core at angles greater than the critical angle and reflect back into the core and off the other side, bouncing back and forth down the channel until they reach the destination.

Figure 6.13 Multimode step-index fiber

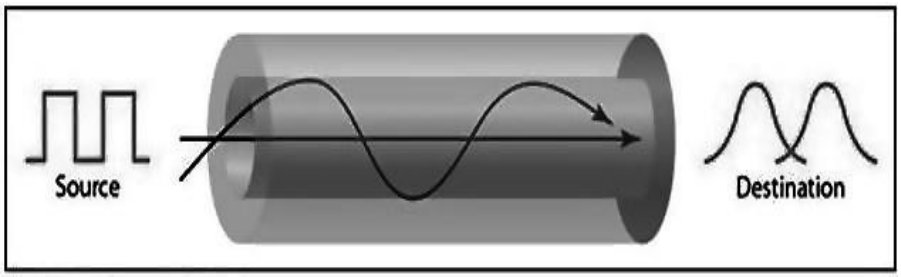


Every beam reflects off the interface at an angle equal to its angle of incidence. The greater the angle of incidence, the wider the angle of reflection. A beam with a smaller angle of incidence will require more bounces to travel the same distance than a beam? with a larger angle of incidence.

Difference in path length means that different beams arrive at the destination at different times. As these different beams are recombined at the receiver, they result in a signal that is no longer an exact replica of the signal that was transmitted. Such a signal has been distorted by propagation delays. This distortion limits the available data rate and makes multimode step-index cable adequate for certain precise application.

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, index of refraction is related to density. A graded-index fiber, therefore, is one with varying densities. Density is highest at the center of the core and decreases gradually to its lowest at the edge. Figure 6.14 shows, the impact of this variable density on the propagation of light beam.

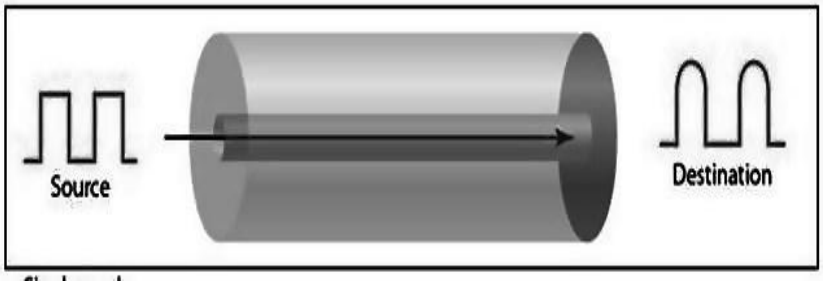
Figure 6.14 Multimode graded index fiber



The signal is introduced at the center of the core. From this point, only the horizontal beam moves in a straight line through the constant density at the center. Beams at other angles move through a series of constantly changing densities. Each density difference causes each beam to refract into a curve. In addition, varying the refraction varies the distance each beam travels in a given period of time, resulting in different beams intersecting at regular intervals. Careful placement of the receiver at one of these inter-sections allows the signal to be reconstructed with far greater precision.

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 fibers, and with substantially lower density (index of refraction). The decrease in density results in a critical angle that is close enough to 90 degrees to make the propagation of beams almost horizontal. In this case, propagation of different beams is almost identical and delays are negligible. All of the beams arrive at the destination "together' and can be recombined without distortion to the signal.(Figure 6.15)

Figure 6.15 Single mode fiber



Fiber sizes

Optical fibers are defined by the ratio of the diameter of their core to the diameter of their cladding, both expressed in microns. The common sizes are shown in table

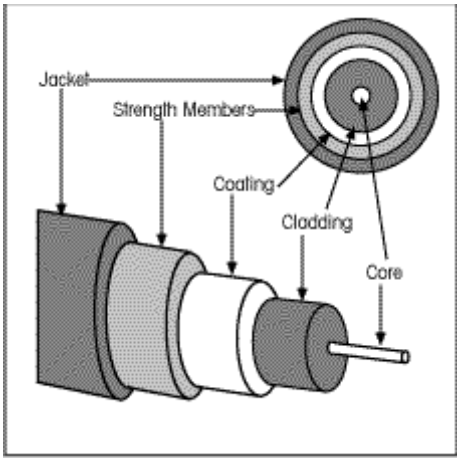
Table 6.1 Fiber Types

Type	Core	Cladding
50/125	50	125
62.5/125	62.5	125
100/125	100	125
7/125	7	125

Cable composition

Figure 6.16 shows the composition of a typical fiber optic cable. A core is surrounded by cladding , forming the fiber. In most cases is covered by a buffer layer that protects from moisture. The entire cable is encased in an outer jacket.

Figure 6.16 Fiber Construction



Both core and cladding are made of glass or plastic. The inner core must be ultrapure and completely regular in size and shape. Chemical differences in material, and even small variations in the size or shape of the channel, alter the angle of reflection and distort the signal. Some applications can handle a certain amount of distortion and their cables can be made more cheaply, but others depend on complete uniformity.

The outer jacket (or sheath) can be made of several materials, including Teflon coating, plastic coating, fibrous plastic, metal tubing, and metal mesh. Each of these jacketing materials has its own purpose. Plastics are lightweight and inexpensive but do not provide structural strength and can emit fumes when burned. Metal tubing provides strength but raises cost. Teflon is lightweight and can be used in open air, but it is expensive and does not increase cable strength. The choice of the material depends on where the cable is to be installed.

Light Sources for Optical Cable

As we have seen, the purpose of fiber-optic cable is to contain and direct a beam of light from source to target. For transmission to occur, the sending device must be equipped with a light source and the receiving device with a photosensitive cell (called a photodiode) capable of translating the received light into current usable by a computer. The light source can be either a light-emitting diode (LED) or an injection laser diode (ILD). LEDs are the cheaper source, but they provide unfocused light that strikes the boundaries of the channel at uncontrollable angles and diffuses over distance. For this reason, LEDs are limited to short-distance use.

Lasers, on the other hand, can be focused to a very narrow range, allowing control over the angle of incidence. Laser signals preserve the character of the signal over considerable distances.

Fiber-Optic Connectors

Connectors for fiber-optic cable must be as precise as the cable itself. With metallic media, connections are not required to be exact as long as both conductors are in physical contact. With optical fiber, on the other hand, any misalignment of one segment of core either with another segment or with a photodiode results_ in the signal reflecting back toward the sender, and any difference in the size of two connected channels results in a change in the angle of the signal. In addition, the connection must be complete yet not overly tight. A gap between two cores results in a dissipated signal; an overly tight connection can compress the two cores and alter the angle of reflection.

Given these constraints, manufacturers have developed several connectors that are both precise and easy to use. All of the popular connectors are barrel shaped and come in male and female versions. The cable is equipped with a male connector that locks or threads into a female connector attached to the device to be connected.

Advantages of Optical Fiber

The major advantages offered by fiber-optic cable over twisted-pair and coaxial cable are noise resistance, less signal attenuation, and higher bandwidth.

- **Noise resistance.** Because fiber-optic transmission uses light rather than electricity, noise is not a factor. External light, the only possible interference, is blocked from the channel by the outer jacket.
- **Less signal attenuation.** Fiber-optic transmission distance is significantly greater than that of other guided media. A signal can run for miles without requiring regeneration.
- **Higher bandwidth.** Fiber-optic cable can support dramatically higher bandwidths (and hence data rates) than either twisted-pair or coaxial cable. Currently, data rate; and bandwidth utilization over fiber-optic cable are limited not by the medium but by the signal generation and reception technology available.

Disadvantages of Optical Fiber

The main disadvantages of fiber optics are cost, installation/maintenance, and fragility.

- **Cost.** Fiber-optic cable is expensive. Because any impurities or imperfections in the core can throw off the signal, manufacturing must be painstakingly precise. Also, a laser light source can cost thousands of dollars, compared to hundreds of dollars for electrical signal generators.
- **Installation/maintenance.** Any roughness or cracking in the core of an optical cable diffuses light and alters the signal. All splices must be polished and precisely fused. All connections must be perfectly aligned and matched for core size and must provide a

completely light-tight seal,, Metallic media connections, on the other hand, can be made by cutting and crimping using relatively unsophisticated tools.

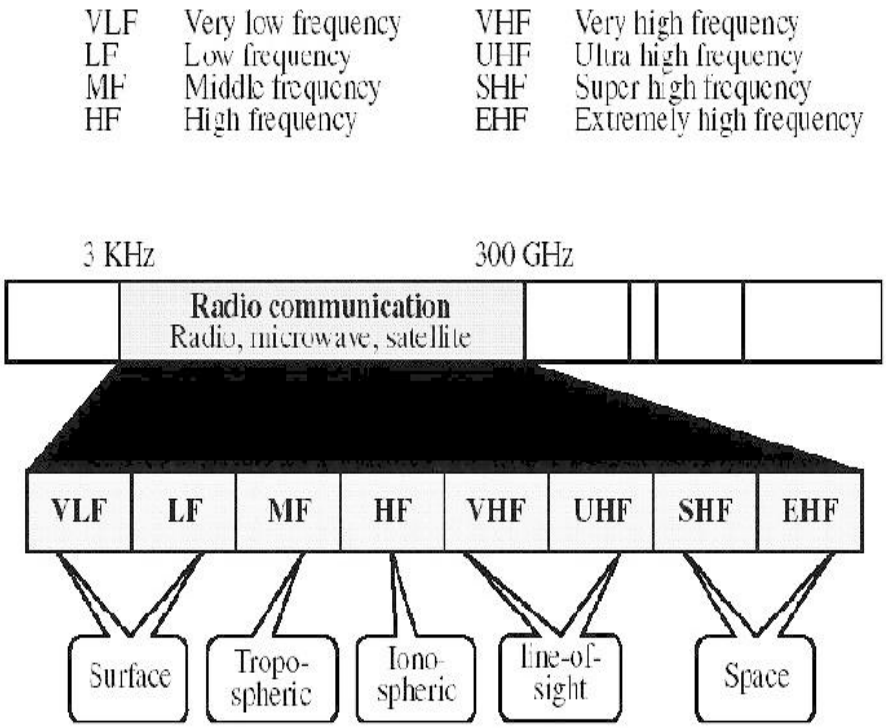
- **Fragility.** Glass fiber is more easily broken than wire, making it less useful for applications where hardware portability is required. As manufacturing techniques have improved and costs have come down, high data rates and immunity to noise have made fiber optics increasingly popular.

6.2 UNGUIDED MEDIA

Unguided media, or wireless communication, transport electromagnetic waves with-out using a physical conductor. Instead, signals are broadcast through air (or, in a few cases, water), and thus are available to anyone who has a device capable of receiving them.

Radio Frequency Allocation The section of the electromagnetic spectrum defined as radio communication 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). Figure 6.17 shows all eight bands and their acronyms.

Figure 6.17 Radio Communication band

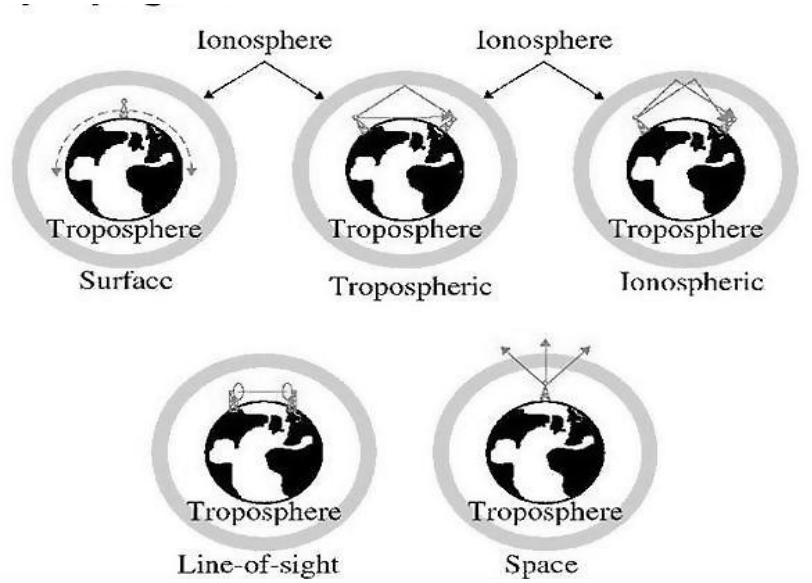


Propagation of Radio Waves

Types of Propagation

Radio wave transmission utilizes five different types of propagation: surface, tropospheric, ionospheric, line-of-sight, and space (Figure 6.18).

Figure 6.18 Types of propagation



Radio technology considers the earth as surrounded by two layers of atmosphere: the troposphere and the ionosphere. The troposphere is the portion of the atmosphere extending outward approximately 30 miles from the earth's surface (in radio terminology, the troposphere includes the high-altitude layer called the stratosphere) and contains what we generally think of as air. Clouds, wind, temperature variations, and weather in general occur in the troposphere, as does jet plane travel. The ionosphere is the layer of atmosphere above the troposphere but below space. It is beyond what we think of as atmosphere and contains free electrically charged particles (hence the name).

Surface Propagation In surface propagation, radio waves travel through the lowest portion of the atmosphere, hugging the earth. At the lowest frequencies, 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. Surface propagation can also take place in seawater.

Tropospheric Propagation Tropospheric propagation can work two ways. Either a signal can be directed in a straight line from antenna to antenna (line-of-sight), or it can be broadcast at an angle into the upper layers of the troposphere where it is reflected back down to the earth's surface. The first method requires that the placement of the receiver and the transmitter be within line-of-sight distances, limited by the curvature of the earth in relation to the height of the antennas. The second method allows greater distances to be covered.

Ionospheric Propagation In ionospheric propagation, higher-frequency radio waves radiate upward into the ionosphere where they are reflected back to earth. The density difference between the troposphere and the ionosphere causes each radio wave to speed up and change direction, bending back to earth. This type of transmission allows for greater distances to be covered with lower power output.

Line-of-Sight Propagation In line-of-sight 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-of-sight propagation is tricky because radio transmissions cannot be completely focused. Waves emanate upward and downward as well as forward and can reflect off the surface of the earth or parts of the atmosphere. Reflected waves that arrive at the receiving antenna later than the direct portion of the transmission can corrupt the received signal.

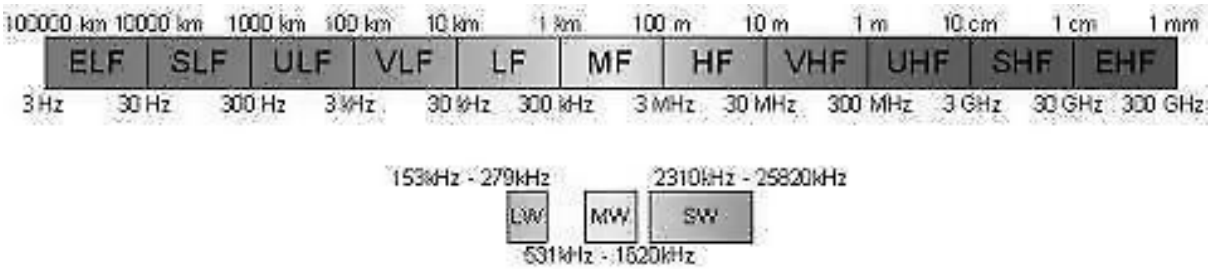
Space Propagation Space propagation utilizes satellite relays in place of atmospheric refraction. A broadcast signal is received by an orbiting satellite, which rebroadcasts the signal to the intended receiver back on the earth. Satellite transmission is basically line-of-sight with an intermediary (the satellite). The distance of the satellite from the earth makes it the equivalent of a super-high-gain antenna and dramatically increases the distance coverable by a signal.

Propagation of Specific Signals

The type of propagation used in radio transmission depends on the frequency (speed) of the signal. Each frequency is suited for a specific layer of the atmosphere and is most efficiently transmitted and received by technologies adapted to that layer.

VLF Very low frequency (VLF) waves are propagated as surface waves, usually through air but sometimes through seawater. VLF waves do not suffer much attenuation in transmission but are susceptible to the high levels of atmospheric noise (heat and electricity) active at low altitudes. VLF waves are used mostly for long-range radio navigation and for submarine communication (Figure 6.19).

Figure 6.19 Frequency range for radio waves



LF Similar to VLF, low frequency (LF) waves are also propagated as surface waves. LF waves are used for long-range radio navigation and for radio beacons or navigational locators (Figure 6.20). Attenuation is greater during the daytime, when absorption of waves by natural obstacles increases.

MF Middle frequency (MF) signals are propagated in the troposphere. These frequencies are absorbed by the ionosphere. The distance they can cover is therefore limited by the angle needed to reflect the signal within the troposphere without entering the ionosphere. Absorption increases during the daytime, but most MF transmissions rely on line-of-sight antennas to increase control and avoid the absorption problem altogether. Uses for MF transmissions include AM radio, maritime radio, radio direction finding (RDF), and emergency frequencies (Figure 6.20).

HF High frequency (HF) signals use ionospheric propagation. These frequencies move into the ionosphere, where the density difference reflects them back to earth. Uses for HF signals include amateursadio (ham radio), citizen's band (CB) radio, inter-national broadcasting, military communication, long distance aircraft and ship communication, telephone, telegraph, and facsimile (Figure 6.20).

VHF Most very high frequency (VHF) waves use line-of-sight propagation. Uses for VHF include VHF television; FM radio, aircraft AM radio, and aircraft navigational aid (see Figure 6.20).

UHF Ultrahigh frequency (UHF) waves always use line-of-sight propagation. Uses for UHF include UHF television, mobile telephone, cellular radio, paging, and micro-wave links (see Figure 6.20). Note that microwave communication begins at 1 GHz in the UH1 band and continues into the SHF and EHF fends.

SHFSuperhigh frequency (SHF) waves are transmitted using mostly line-of-sight and some space propagation. Uses for SHF include terrestrial and satellite microwave and radar communication (see Figure 6.20).

EHF. Extremely high frequency (EHF) waves use space propagation. Uses for EHF are predominantly scientific and include radar, satellite, and experimental communications (see Figure 6.20).

Terrestrial Microwave

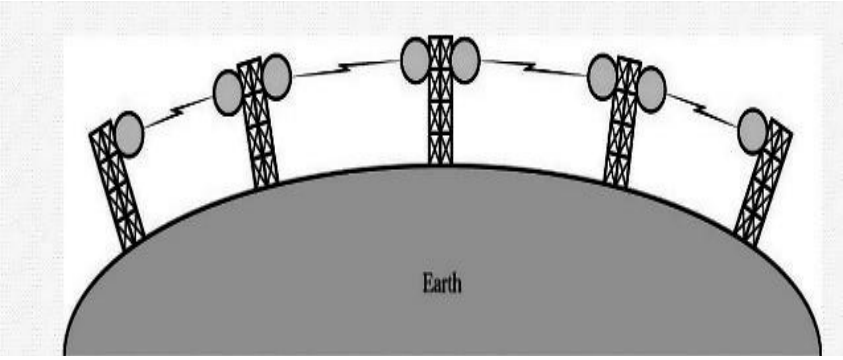
Microwaves do not follow the curvature of the earth and therefore require line-of-sight transmission and reception equipment. The distance coverable by a line-of-sight signal depends to a large extent on the height of the antenna: the taller the antennas, the longer the sight distance. Height allows the signal to travel farther without being stopped by the curvature of the planet and raises the signal above many surface obstacles, such as low hills and tall buildings that would otherwise block transmission. Typically, antennas are mounted on towers that are in turn often mounted on hills or mountains.

Microwave signals propagate in one direction at a time, which means that two frequencies are necessary for two-way communication such as a telephone conversation. One frequency is reserved for microwave transmission in one direction and the other for transmission in the other. Each frequency requires its own transmitter and receiver. Today, both pieces of equipment usually are combined in a single piece of equipment called a transceiver, which allows a single antenna to serve both frequencies and functions.

Repeaters

To increase the distance served by terrestrial microwave, a system of repeaters can be installed with each antenna. A signal received by one antenna can be converted back into transmittable form and relayed to the next antenna (Figure 6.20). The distance required between repeaters varies with the frequency of the signal and the environment in which the antennas are found. A repeater may broadcast the regenerated signal either at the original frequency or at a new frequency, depending on the system.

Figure 6.20 Terrestrial microwave



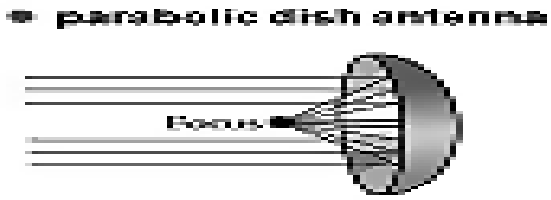
Terrestrial microwave with repeaters provides the basis for most contemporary telephone systems worldwide.

Antennas

Two types of antennas are used for terrestrial microwave communications: parabolic dish and horn.

A parabolic dish antenna is based on the geometry of a parabola: every line parallel to the line of symmetry (line of sight) reflect off the curve at angles such that they intersect in a common point called the focus (Figure 6.21). The parabolic dish works like 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.

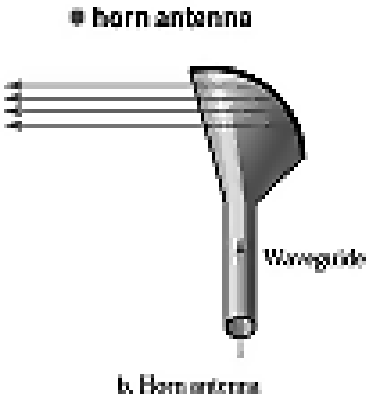
Figure 6.21 Parabolic dish antenna



Outgoing transmissions are broadcast through a horn aimed at the dish. The micro-waves 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 (see Figure 6.22). 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.

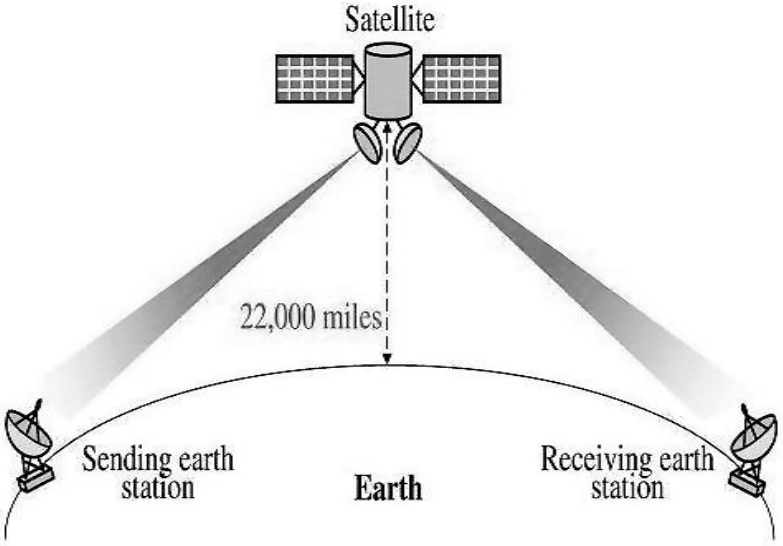
Figure 6.22 Horn antenna



Satellite Communication

Satellite transmission is much like line-of-sight microwave transmission in which one of the stations is a satellite orbiting the earth. The principle is the same as terrestrial microwave, with a satellite acting as a super tall antenna and repeater (see Figure 6.23). Although in satellite transmission signals must still travel in straight lines, the limitations imposed on distance by the curvature of the earth are reduced. In this way, satellite relays allow microwave signals to span continents and oceans with a single bounce.

Figure 6.23 Satellite Communication



Satellite microwave can provide transmission capability to and from any location on earth, no matter how remote. This advantage makes high-quality communication available to undeveloped parts of the world without requiring a huge investment in ground-based infrastructure. Satellites themselves are extremely expensive, of course, but leasing time or frequencies on one can be relatively cheap.

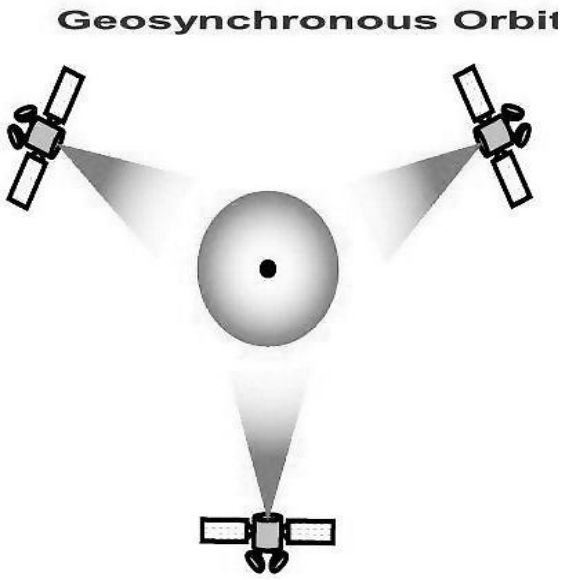
Geosynchronous Satellites

Line-of-sight propagation requires that the sending and receiving antennas be locked onto each other's location at all times (one antenna must have the other in sight). For this reason, a satellite that moves faster or slower than the earth's rotation is useful only for short periods of time (just as a stopped clock is accurate twice a day). To ensure constant communication, the satellite must move at the same speed as the earth so that it seems to remain fixed above a certain spot. Such satellites are called geosynchronous.

Because orbital speed is based on distance from the planet, only one orbit can be geosynchronous. This orbit occurs at the equatorial plane and is approximately 22.000 miles from the surface of the earth. But one geosynchronous satellite cannot cover the whole earth.

One satellite in t has line-of-sight contact with a vast number of stations, but the curvature of the earth still keeps much of the planet out of sight. It takes a minimum of three satellites equidistant from each other in geosynchronous orbit to provide full global transmission. Figure 6.24 shows three satellites, each 120 degrees from another in geosynchronous orbit around the equator. The view is from the North Pole.

Figure 6.24 Satellites in geosynchronous orbit



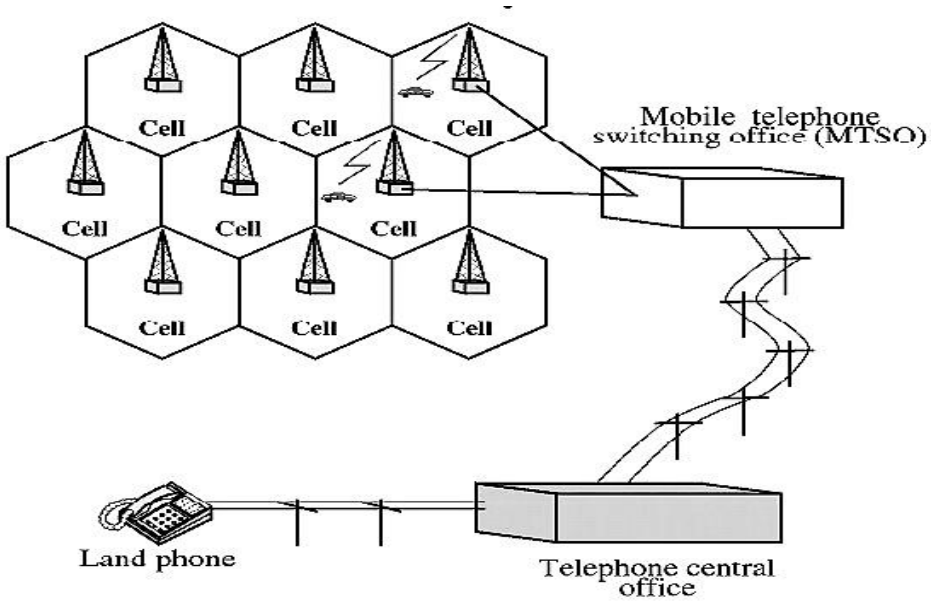
Frequency Bands for Satellite Communication The frequencies reserved for satellite microwave communication are in the gigahertz (GHz) range. Each satellite sends and receives over two different bands. Transmission from the earth to the satellite is called uplink. Transmission from the satellite to the earth is called downlink. Table 6.2 gives the band names and frequencies for each range.

Table 6.2 Satellite frequency band

Band	Downlink	Uplink
C	03.7 - 04.2 GHz	05.925 - 06.425 GHz
Ku	11.7 - 12.2 GHz	14.000 - 14.500 GHz
Ka	17.7 - 21.0 GHz	27.500 - 31.000 GHz

Cellular Telephony Cellular telephony is designed to provide stable communications connections between two moving devices or between one mobile unit and one stationary (land) unit. A service provider must be able to locate and track a caller, assign a channel to the call, and transfer the signal from channel to channel as the caller moves out of the range of one channel and into the range of another. To make this tracking possible, each cellular service area is divided into small regions called cells. Each cell contains an antenna and is controlled by a small office, called the cell office. Each cell office, in turn, is controlled by a switching office called a mobile telephone switching office (MTSO). The MTSO coordinates communication between all of the cell offices and the telephone central office. It is a computerized center that is responsible for connecting calls as well as recording call information and billing (see Figure 6.25).

Figure 6.25 Cellular system



Cell size is not fixed and can be increased or decreased depending on the population of the area. The typical radius of a cell is 1 to 12 miles. High-density areas require-more,

geographically smaller cells to meet traffic demands than do lower density areas. Once determined, cell size is optimized to prevent the interference of adjacent cell signals. The transmission power of each cell is kept low to prevent its signal from interfering with those of other cells.

Cellular Bands

Traditional cellular transmission is analog. To minimize noise, frequency modulation (FM) is used for communication between the mobile telephone itself and the cell office. The FCC has assigned two bands for cellular use. The band between 824 and 849 MHz carries those communications that initiate from mobile phones. The band between 869 and 894 MHz carries those communications that initiate from land phones. Carrier frequencies are spaced every 30 KHz, allowing each band to support up to 833 carriers. However, two carriers are required for full-duplex communication, which doubles the required width of each channel to 60 KHz and leaves only 416 channels available for each band.

Each band, therefore, is divided into 416 FM channels (for a total of 832 channels). Of these, some are reserved for control and setup data rather than voice communication. In addition, to prevent interference, channels are distributed among the cells in such a way that adjacent cells do not use the same channels. This restriction means that each cell normally has access to only 40 channels.

Transmitting

To place a call from a mobile phone, the caller enters a code of 7 or 10 digits (a phone number) and presses the send button. The mobile phone then scans the band, seeking a setup channel with a strong signal, and sends the data (phone number) to the closest cell office using that channel. The cell office relays the data to the MTSO. The MTSO sends the data on to the telephone central office. If the called party is available, a connection is made and the result is relayed back to the MTSO. At this point, the MTSO assigns an unused voice channel to the call and a connection is established. The mobile phone automatically adjusts its tuning to the new channel and voice communication can begin.

Receiving

When a land phone places a call to a mobile phone, the telephone central office sends the number to the MTSO. The MTSO searches for the location of the mobile phone by sending query signals to each cell in a process called paging. Once the mobile phone is found, the MTSO transmits a ringing signal and, when the mobile phone is answered, assigns a voice channel to the call, allowing voice communication to begin.

Handoff

It may happen that, during a conversation, the mobile phone moves from one cell to another. When it does, the signal may become weak. To solve this problem, the MTSO monitors the level of the signal every few seconds. If the strength of the signal diminishes, the MTSO seeks a new cell that can accommodate the communication better. The MTSO then changes the channel carrying the call (hands the signal off from the old channel to a new one). Handoffs are performed so smoothly that most of the time they are transparent to the users.

Digital

Analog (FM) cellular services are based on a standard called analog circuit switched cellular (ACSC). To transmit digital data using an ACSC service requires a modem with a maximum speed of 9600 to 19,200 bps.

Since 1993, however, several service providers have been moving to a cellular data standard called cellular digital packet data (CDPD). CDPD provides low-speed digital service over the existing cellular network. It is based on the OSI model.

To use the existing digital services, such as 56K switched service, CDPD uses what is called a trisector. A trisector is a combination of three cells each using 19.2 Kbps, for a total of 57.6 Kbps (which can be accommodated on a 56K switched line by eliminating some overhead). Under this scheme, the United States is divided into 12,000 trisectors. For every 60 trisectors, there is one router.

Integration with Satellites and PCs

Cellular telephony is moving fast toward integrating the existing system with satellite communication. This integration will make it possible to have mobile communication between any two points on the globe. Another goal is to combine cellular telephony and personal computer communication under a scheme called mobile personal communication to enable people t se small, mobile personal computers to send and receive data, voice, image, and video.

7. ERROR DETECTION AND CORRECTION

Networks must be able to transfer data from one device to another with complete accuracy. Anytime data are transmitted from source to destination, they can become corrupted in passage. In fact, it is more likely that some part of a message will be altered in transit than that the entire contents will arrive intact.

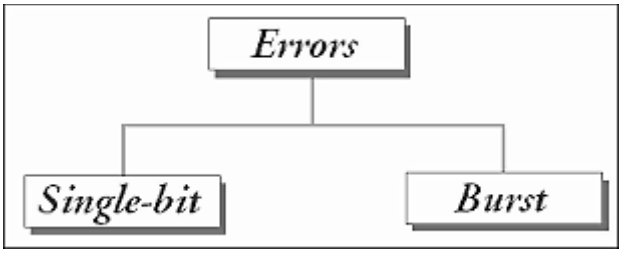
Data can be corrupted during transmission. For reliable communication, errors must be detected and corrected.

Error detection and correction are implemented either at the data link layer or the transport layer of the OS1 model.

7.1 TYPES OF ERRORS

Whenever an electromagnetic signal flows from one point to another, it is subject to unpredictable interference from heat, magnetism, and other forms of electricity. This interference can change the shape or timing of the signal. If the signal is carrying encoded binary data, such changes can alter the meaning of the data. In a single-bit error, a 0 is changed to a 1 or a 1 to a 0. In a burst error, multiple bits are changed. For example, a 0.01-second burst of impulse noise on a transmission with a data rate of 1200 bps might change all or some of 12 bits of information (see Figure 7.1).

Figure 7.1 Types of errors



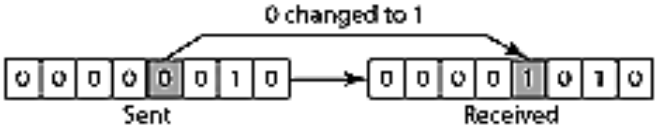
Single-Bit Error

The term single-bit error means that only one bit of a given data unit (such as a byte, character, data unit, or packet) is changed from 1 to 0 or from 0 to 1.

In a single-bit error, only one bit in the data unit has changed.

Figure 7.2 shows the effect of a single-bit error on a data unit. To understand the impact of the change, imagine that each group of eight bits is an ASCII character with a 0 bit added to the left. In the figure, 00000010 (ASCII STX) was sent, meaning start of text, but 00001010 (ASCII LF) was received, meaning line feed.

Figure 7.2 Single-bit error



Single-bit errors are the least likely type of error in serial data transmission. To see why, imagine a sender sends data at 1Mbps. This means that each bit lasts only 1/1,000,000 second, or 1 us. For a single-bit error to occur, the noise must have a duration of only 1 ps, which is very rare; noise normally lasts much longer than this.

However, a single-bit error can happen if we are sending data using parallel transmission. For example, if eight wires are used to send all of the eight bits of a byte at the same time and one of the wires is noisy, one bit can be corrupted in each byte. Think of parallel transmission inside a computer, between CPU and memory, for example.

Burst Error

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

A burst error means that two or more bits in the data unit have changed.

Figure 7.3 shows the effect of a burst error on a data unit. In this case, 0100010001000011 was sent, but 0101110101000011 was received. Note that a burst error does not necessarily mean that the errors occur in consecutive 'bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.

Figure 7.3 Burst error on length five

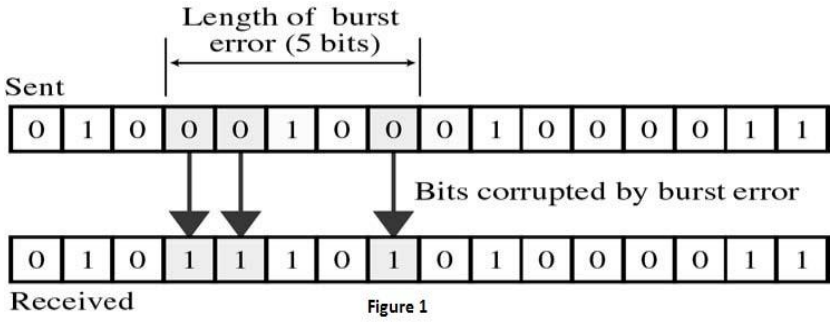


Figure 1

Burst error is most likely to happen in a serial transmission. The duration of noise is normally longer than the duration of a 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 seconds can affect 10 bits; if we are sending data at 1 Mbps, the same noise can affect 10,000 bits. 7)

7.2 DETECTION

For a machine to check for error would be slow, costly, and of questionable value. What we need is a mechanism that is simple and completely objective.

Redundancy

Cone error detection mechanism that would satisfy these requirements would be to send every data unit twice. The receiving device would then be able to do a bit-for-bit comparison between the two versions of the data. Any discrepancy would indicate an error,

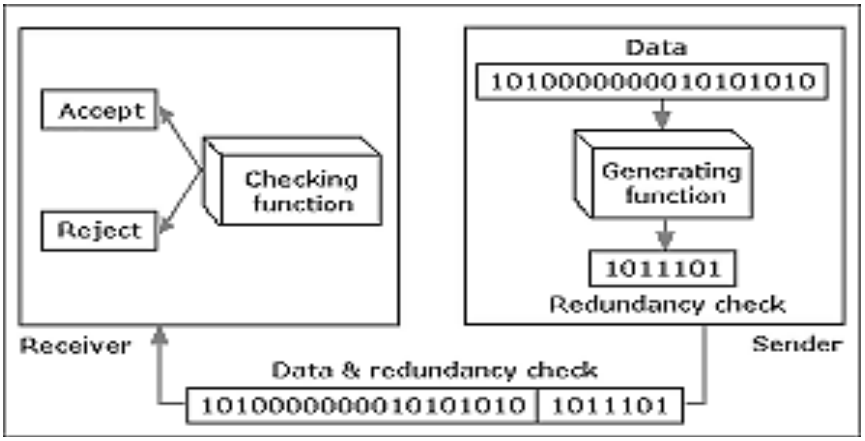
and an appropriate correction mechanism could be set in place. This system would be completely accurate (the odds of errors being introduced onto exactly the same bits in both sets of data are infinitesimally small), but it would also be insupportably slow. Not only would the transmission time double, but the time it takes to compare every unit bit by bit must be added.

The concept of including extra information in the transmission solely for the purposes of comparison is a good one. But instead of repeating the entire data stream, a shorter group of bits may be appended to the end of each unit. This technique is called redundancy because the extra bits are redundant to the information; they are discarded as soon as the accuracy of the transmission has been determined.

Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.

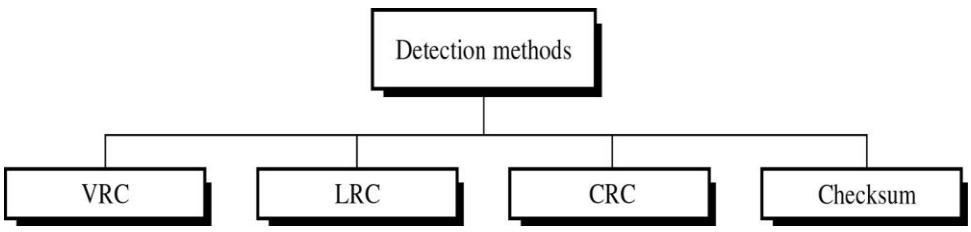
Figure 7.4 shows the process of using redundant bits to check the accuracy of a data unit. Once the data stream has been generated, it passes through a device that analyzes it and adds on an appropriately coded redundancy check. The data unit, now enlarged by several bits (in this illustration, seven), travels over the link to the receiver. The receiver puts the entire stream through a checking function. If the received bit stream passes the check in criteria, the data portion of the data unit is accepted and the redundant bits are discarded

Figure 7.4 Redundancy



Four types of redundancy checks are used in data communications: vertical redundancy check (VRC) (also called parity check), longitudinal redundancy check (LRC), cyclical redundancy check (CRC), and checksum. The first three, VRC, LRC, and CRC, are normally implemented in the physical layer for use in the data link layer. The fourth, checksum, is used primarily by upper layers (see Figure 7.5)

Figure 7.5 Detection Methods



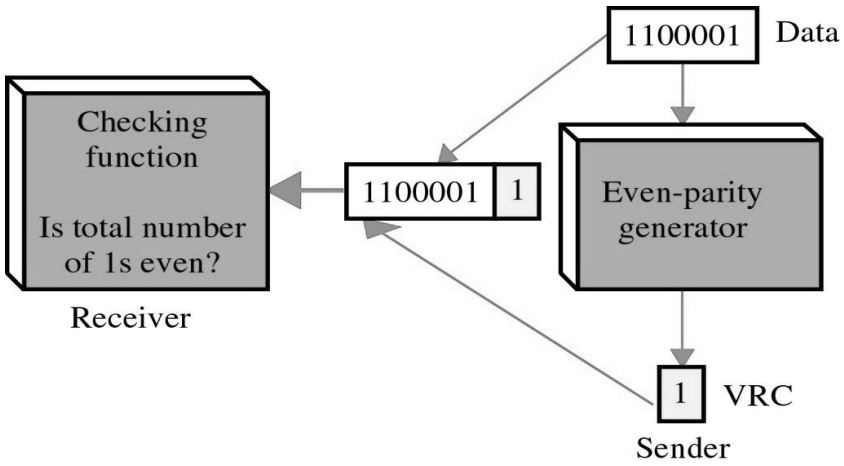
7.3 VERTICAL REDUNDANCY CHECK (VRC)

The most common and least expensive mechanism for error detection is the vertical redundancy check (VRC), often called a parity check. In this technique, a redundant bit, called a parity bit, is appended to every data unit so that the total number of Is in the unit (including the parity bit) becomes even.

Suppose we want to transmit the binary data unit 1100001 [ASCII a (97)]; see Figure 7.6. Adding together the number of 1s gives us 3, an odd number. Before transmitting, we pass the data unit through a parity generator. The parity generator counts the Is and appends the parity bit (a 1 in this case) to the end. The total number of Is is now four, an even number. The system now transmits the entire expanded unit across the network link. When it reaches its destination, the receiver puts all eight bits through an even-parity checking function. If the receiver sees 11100001, it counts four Is, an even number, and the data unit passes. But what if the data unit has been damaged in transit? What if, instead of 11100001, the receiver sees 11100101? Then, when the parity checker counts the 1 s, it gets 5, an odd number. The receiver knows that an error has been introduced into the data somewhere and therefore rejects the whole u

In vertical redundancy check (VRC), a parity bit is added to every data unit so that the total number of ls becomes even.

Figure 7.6 Even parity VRC concept



Performance

VRC can detect all single-bit errors. It can also detect burst errors as long as the total number of bits changed is odd (1, 3, 5, etc.). Let's say we have an even-parity data unit where the total number of 1 s, including the parity bit, is 6: 1000111011. If any three bits change value, the resulting parity will be odd and the error will be detected: 1111111011:9, 0110111011:7, 1100010011:5-all odd. The VRC checker would return a result of 1 and the data unit would be rejected. The same holds true for any odd number of errors.

Suppose, however, that two bits of the data unit are changed: 1110111011:8, 1100011011:6, 1000011010:4. In each case the number of is in the data unit is still even. The VRC checker will add them and return an even number although the data unit contains two errors. VRC cannot detect errors where the total number of bits changed is even. If any two

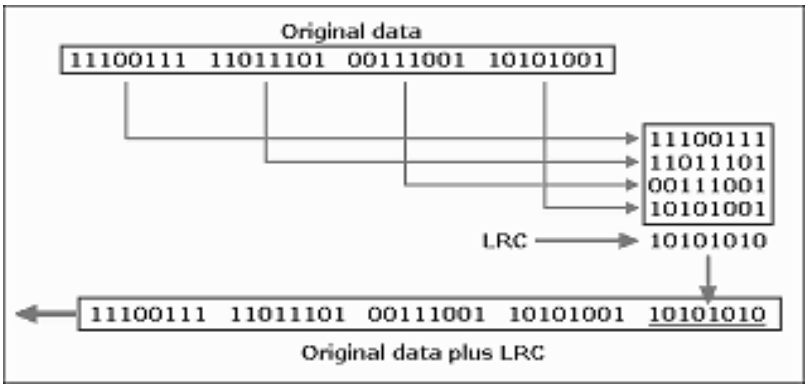
bits change in transmission, the changes cancel each other and the data unit will pass a parity check even though the data unit is damaged. The same holds true for any even number of error

VRC can detect all single-bit errors. It can detect burst errors only if the total number of errors in each data unit is odd.

7.4 LONGITUDINAL REDUNDANCY CHECK (LRC)

In longitudinal redundancy check (LRC), a block of bits is organized in a table (rows and columns). For example, instead of sending a block of 32 bits, we organize them in a table made of four rows and eight columns, as shown in Figure 7.7. We then calculate the parity bit for each column and create a new row of eight bits, which are the parity bits for the whole block. Note that the first parity bit in the fifth row is calculated based on all first bits. The second parity bit is calculated based on all second bits, and so on. We then attach the eight parity bits to the original data and send them to the receive.

Figure 7.7 LRC



In longitudinal redundancy check (LRC), a block of bits is divided into rows and a redundant row of bits is added to the whole block.

Performance

(LAC increases the likelihood of detecting burst errors. As we showed in the previous example, an LRC of n bits can easily detect a burst error of n bits. A burst error of more than n bits is also detected by LRC with a very high probability. There is, however, one pattern of errors that remains elusive. If two bits in one data unit are damaged and two bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error. Consider, for example, two data units: 11110000 and 11000011.4 If the first and last bits in each of them are changed, m ing the data units read 01110001 and 01000010, the errors cannot be detected by LRC.

7.5 CYCLIC REDUNDANCY CHECK (CRC)

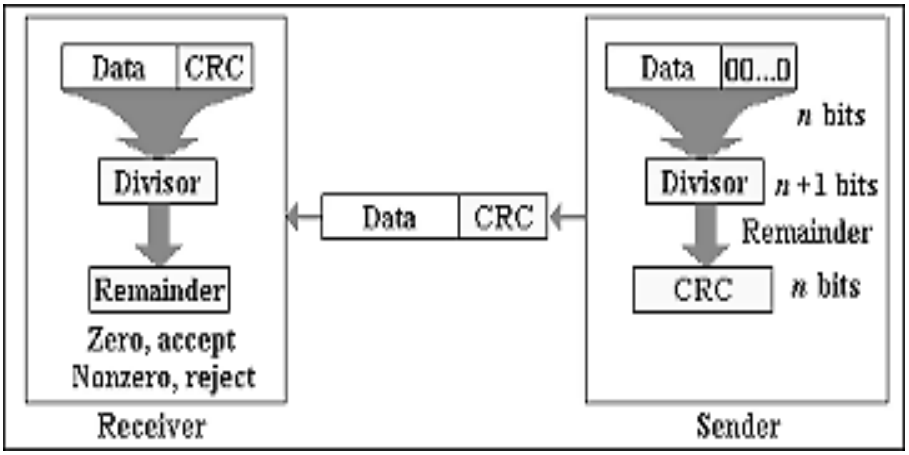
The third and most powerful of the redundancy checking techniques is the cyclic redundancy check (CRC). Unlike VRC and LRC, which are based on addition, CRC is based on binary division. In CRC, instead of adding bits together to achieve a desired parity, a sequence of redundant bits, called the CRC or the CRC remainder, is appended to the end of a data unit so that the resulting data unit becomes exactly divisible by a second,

predetermined binary number. At its destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be intact and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

The redundancy bits used by CRC are derived by dividing the data unit by a predetermined divisor; the remainder is the CRC. To be valid, a CRC must have two qualities: it must have exactly one less bit than the divisor, and appending it to the end of the data string must make the resulting bit sequence exactly divisible by the divisor.

Both the theory and the application of CRC error detection are straightforward. The only complexity is in deriving the CRC. In order to clarify this process, we will start with an overview and add complexity as we go. Figure 7.8 provides an outline of the three basic steps.

Figure 7.8 CRC generator and checker



First, a string of n. than 0s is appended to the data unit. The number n is one less than the number of bits in the predetermined divisor, which is n + 1 bits.

Second, the newly elongated data unit is divided by the divisor using a process called binary division. The remainder resulting from this division is the CRC,

Third, the CRC of n bits derived in step 2 replaces the appended 0s at the end of the data unit. Note that the CRC may consist of all 0s.

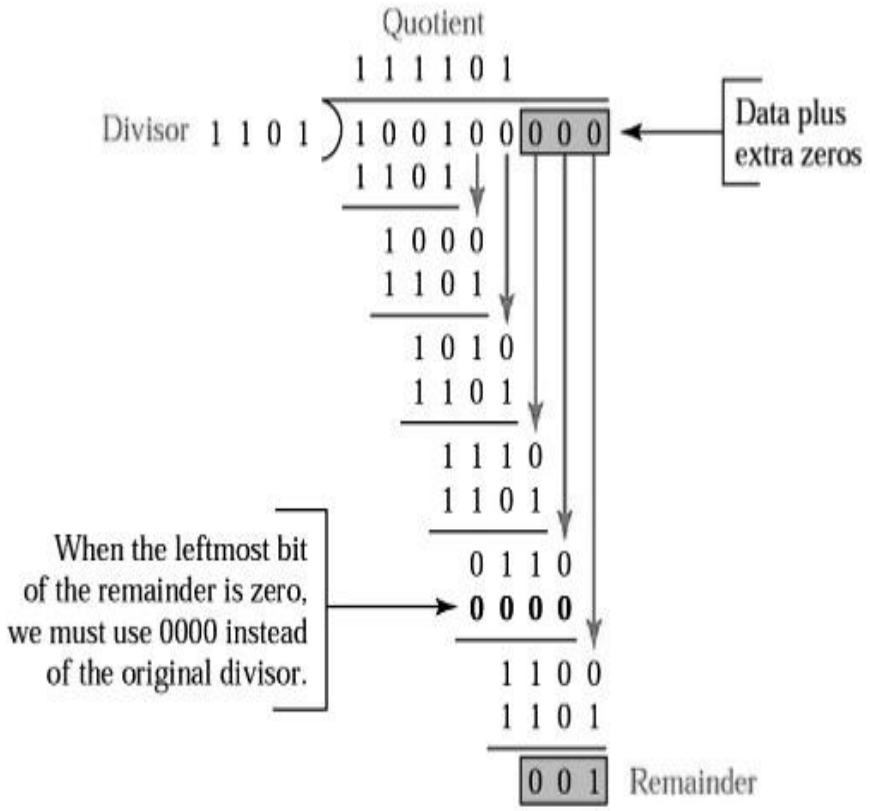
The data unit arrives at the receiver data first, followed by the CRC. The receiver treats the whole string as a unit and divides it by the same divisor that was used to find the CRC remainder.

If the string arrives without error, the CRC checker yields a remainder of zero and the data unit passes. If the string has been changed in transit, the division yields a non-zero remainder and the data unit does not pass.

The CRC Generator

ACRC generator uses modulo-2 division. Figure 7.9 shows this process. In the first step, the four-bit divisor is subtracted from the first four bits of the dividend, Each bit c,' the divisor is subtracted from the corresponding bit of the dividend without disturbing the next higher bit. In our example, the divisor, 1101, is subtracted from the first four bits of the dividend, 1001, yielding 100 (the leading 0 of the remainder is dropped off).

. Figure 7.9 Binary division in CRC generator



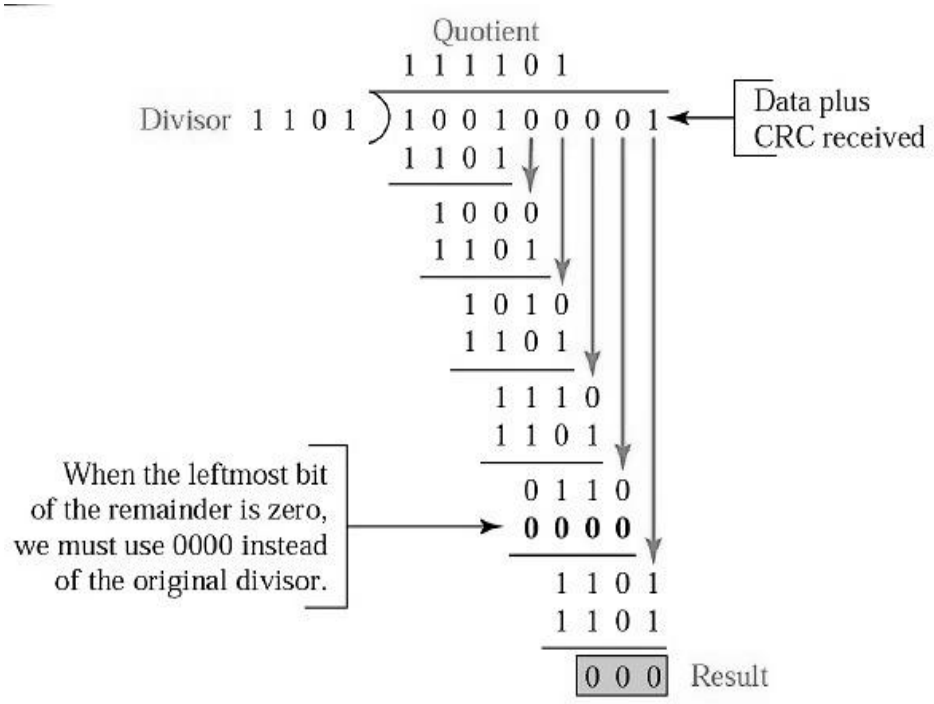
The next unused bit from the dividend is then pulled down to make the number of bits in the remainder equal to the number of bits in the divisor. The next step, therefore, is 1000 – 1101, which yields 101, and so on.

In this process, the divisor always begins with a 1; the divisor is subtracted from a portion of the previous dividend remainder that is equal to it in length; the divisor can only be subtracted from a dividend/remainder whose leftmost bit is 1. Anytime the left-most bit of the dividend/remainder is 0, a string of 0s, of the same length as the divisor, replaces the divisor in that step of the process. For example, if the divisor is four bits long, it is replaced by four 0s. (Remember, we are dealing with bit patterns, not with quantitative values; 0000 is not the same as 0.) This restriction means that, at any step, the leftmost subtraction will be either 0 - 0 or 1- 1, both of which equal 0. c,% after subtraction, the leftmost bit of the remainder will always be a leading zero, which is dropped off, and the next unused bit of the dividend is pulled down to fill out the remainder. Note that only the first bit of the remainder is dropped—if the second bit is also 0, it is retained, and the dividend/remainder for the next step will begin with O. This process repeats until the entire dividend has been used.

The CRC Checker

A CRC checker functions exactly like the generator. After receiving the data appended with the CRC, it does the same modulo-2 division. If the remainder is all 0s, the CRC is dropped and the data accepted; otherwise, the received stream of bits is discarded and data are resent. Figure 7.10 shows the same process of division in the receiver. We assume that there is no error. The remainder is therefore all 0s and the data are accepted.

Figure 7.10 Binary division in CRC checker

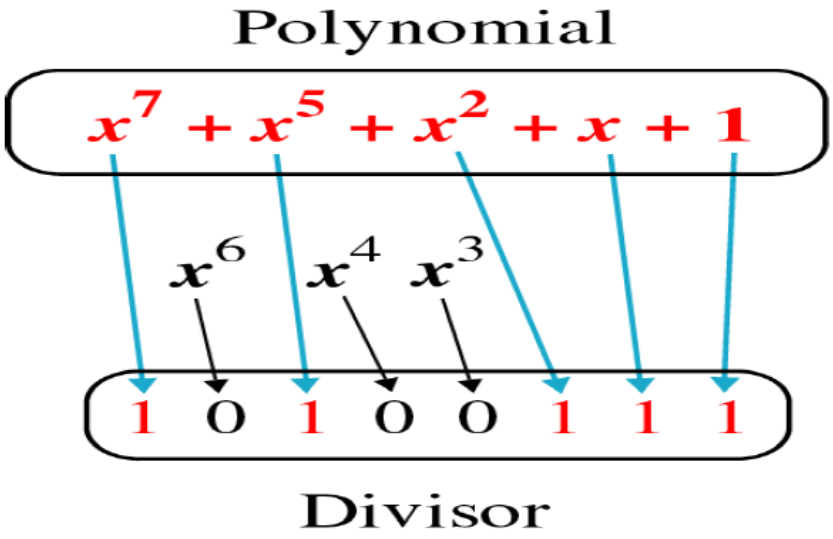


Polynomials

The CRC generator (the divisor) is most often represented not as a string of Is and 0s, t as an algebraic polynomial. The polynomial format is useful for two reasons: It is short, and it can be used to prove the concept mathematically

The relationship of a polynomial to its corresponding binary representation is shown in Figure 7.11.

Figure 7.11.A polynomial representing divisor



A polynomial should be selected to have at least the following properties:

- It should not be divisible by x.
- It should be divisible by (x + 1).

The first condition guarantees that all burst errors of a length equal to the degree of the polynomial are detected. The second condition guarantees that all burst errors affecting an odd number of bits are detected.

Performance

CRC is a very effective error detection method. If the divisor is chosen according to the previously mentioned rules,

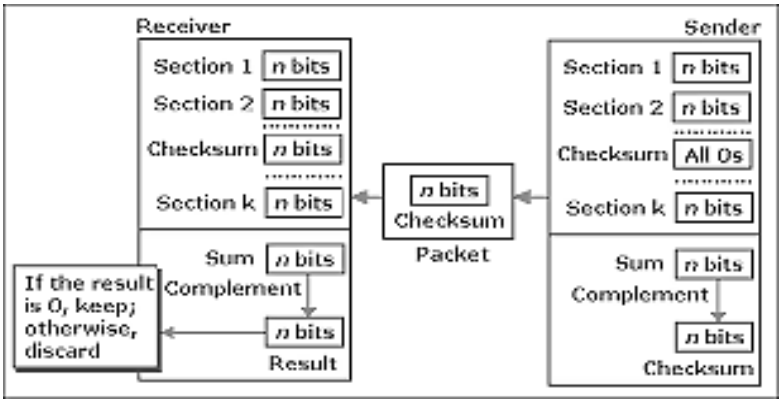
a. CRC can detect all burst errors that affect an odd number of bits.
b. CRC can detect all burst errors of length less than or equal degree. The polynomial.
c. CRC can detect with a very if h probability burst errors of length greater thank. The degree of the polynomial.

7.6 CHECKSUM

The error detection method used by the higher-layer protocols is called checksum. Like VRC, LRC, and CRC, checksum is based on the concept of redundancy.

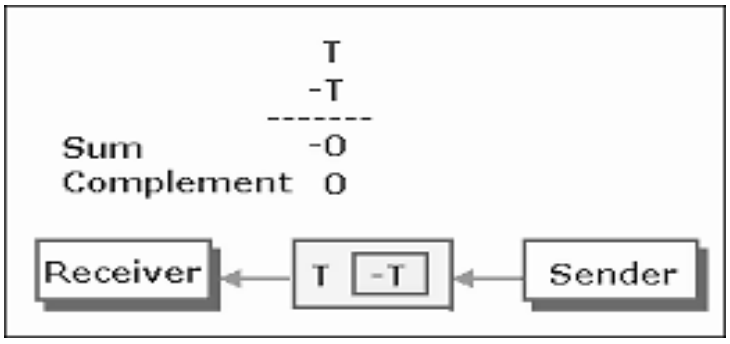
Checksum Generator n the sender, the checksum generator subdivides the data unit into equal segments of n bits (usually 16). These segments are added together using one's com7lemefitt arithmetic (see Appendix C) in such a way that the total is also n bits long. That total (sum) is then complemented and appended to the end of the original data unit as redundancy bits, called the checksum field. The extended data unit is transmitted across the IDA-work. So if the sum of the data segment is T, the checksum will be -T (see Figures 7.14 and 9.15).

Figures 7.14 Checksum



Checksum Checker The receiver subdivides the data unit as above and adds all segments together and complements the result. If the extended data unit is intact, the total value found by adding the data segments and the checksum field should be zero. If the result is not zero, the packet contains an error and the receiver rejects.

Figures 7.15 Data unit and checksum



The sender follows these steps:

- The unit is divided into k sections, each of n bits.
- All sections are added together using one's complement to get the sum.
- The sum is complemented and becomes the checksum.
- The checksum is sent with the data.

The receiver follows these steps:

- The unit is divided into k sections, each of n bits.
- All sections are added together using one's complement to get the sum.
- The sum is complemented.
- If the result is zero, the data are accepted: otherwise, they are rejected.

Performance

The checksum detects all errors involving an odd number of bits, as well as most errors involving an even number of bits. However, if one or more bits of a segment are dam-aged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem. If the last digit of one segment is a 0 and it gets changed to a 1 in transit, then the last 1 in another segment must be changed to a 0 if the error is to go undetected. In LRC, two 0s could both change to 1 s without altering the parity because carries were discarded. Checksum retains all carries; so, although two 0s becoming 1 s would not alter the value of their own column, they would change the value of the next higher column. But anytime a bit inversion is balanced by an opposite bit inversion in the corresponding digit of another data segment, the error is invisible.

7.7 ERROR CORRECTION

The mechanisms that we have covered up to this point detect errors but do not correct them. Error correction can be handled in two ways. In one, when an error is discovered, the receiver can have the sender retransmit the entire data unit. In the other, a receiver can use an error-correcting code, which automatically corrects certain errors. In theory, it is possible to correct any binary code errors automatically. Error correcting codes, however, are more sophisticated than error-detection codes and require more redundancy bits. The number of bits required to correct a multiple-bit or burst error is so high that in most cases it is inefficient to do so. For this reason, most error correction is limited to one-, two-, or three-bit errors.

Single-Bit Error Correction

The concept underlying error correction can be most easily understood by examining the simplest case: single-bit errors.

As we saw earlier, single-bit errors can be detected by the addition of a redundant (parity) bit to the data unit (VRC). A single additional bit can detect single-bit errors in any sequence of bits because it must distinguish between only two conditions: error or no error. A bit has two states (0 and 1). These two states are sufficient for this level of detection.

But what if we want to correct as well as detect single-bit errors? Two states are enough to detect an error but not to correct it. An error occurs when the receiver reads a 1 bit as a 0 or a 0 bit as a 1. To correct the error, the receiver simply reverses the value of the altered bit. To do so, however, it must know which bit is in error. The secret of error correction, therefore, is to locate the invalid bit or bits.

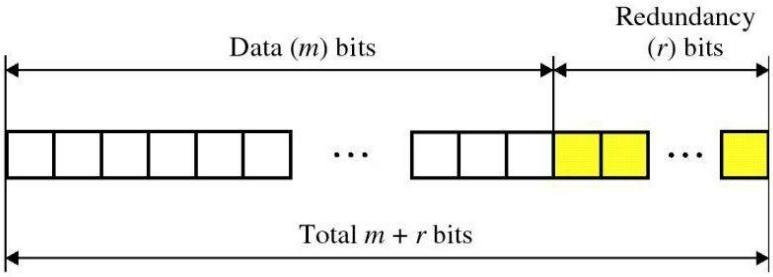
For example, to correct a single-bit error in an ASCII character, the error correction code must determine which of the seven bits has changed. In this case, we have to distinguish between eight different states: no error, error in position 1, error in position 2, and so on, up to error in position 7. To do so requires enough redundancy bits to show all eight states.

At first glance, it looks like a three-bit redundancy code should be adequate because three bits can show eight different states (000 to 111) and can thereforeindicate the locations of eight different possibilities.

Redundancy Bits

To calculate the number of redundancy bits (r) required to correct a given number of data bits (m), we must find a relationship between m and r. Figure 7.16 shows m bits of data with r bits of redundancy added to them. The length of the resulting code is m + r,,

Figure 7.16 Data and redundancy bits



If the total number of bits in a transmittable unit is m + r, then r must be able to indicate at least m + r + 1 different state. Of these, one state means no error and m + r states indicate the location of an error in each of the m + r positions. So, m + r + 1 state must be discoverable by r bits; and r bits can indicate 2^r different states. Therefore, 2^r must be equal to or greater than m + r + 1:

$$2^r \geq m + r + 1$$

The value of r can be determined by plugging in the value of m (the original length of the data unit to be transmitted).

Hamming Code

So far, we have examined the number of bits required to cover all of the possible single-tit error states in a transmission. But how do we manipulate those bits to discover which state has occurred? A technique developed by R. W. Hamming provides a practical solution.

Positioning the Redundancy Bits

The Hamming code can be applied to data units of any length and uses the relationship between data and redundancy bits discussed above. For example, a seven-bit ASCII code requires four redundancy bits that can be added to the end of the data unit or interspersed with the original data bits. In Figure 9.17, these bits are placed in positions 1. 2. 4, and 8 (the positions in an 11-bit sequence that are powers of 2). For clarity in the examples below, we refer to these bits as $r_1, r_2, r_3, and\ r_8$.

In the Hamming code, each r bit is the VRC bit for one combination of data bitsr_1 is the VRC bit for one combination of data bits, $r_2$is the VRC bit for another combination of data bits, and so on. The combinations used to calculate each of the four r values for a seven bit data sequence are as follows:

r_1: bits 1,3,5,7,9,11

r_2: bits 2,3,6,7,10,11

r_3: bits 4,5,6,7

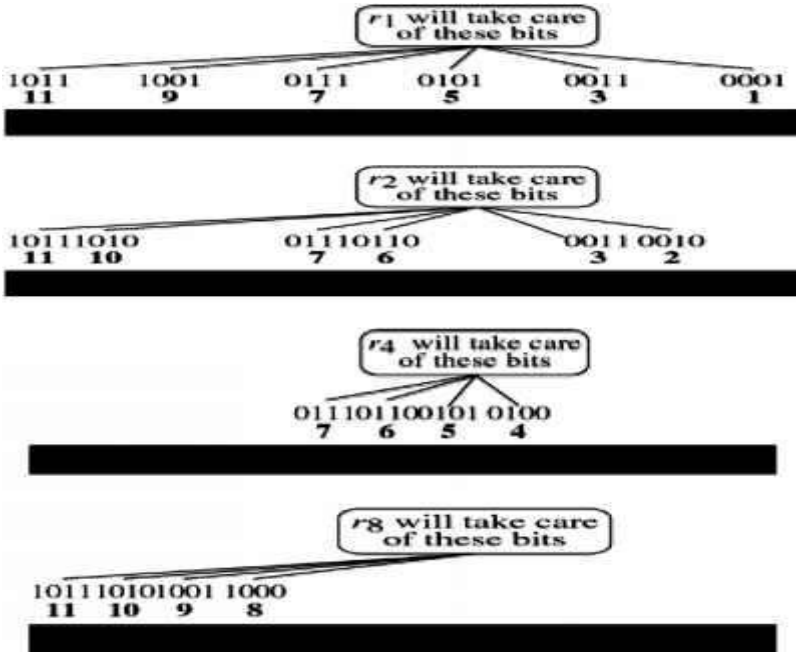
r_8: bits 8,9,10,11

Each data bit may be included in more than one VRC calculation. In the sequences above, for example, each of the original data bits is included in at least two sets. while the r bits are included in only one.

Calculating the r Values

Figure 7.17 shows a Hamming code implementation for an ASCII character. In the first step, we place each bit of the original character in its appropriate position in the 11-bit unit. In the subsequent steps, we calculate the even parities for the various bit combinations. The parity value for each combination is the value of the corresponding r bit. For example, the value of bits calculated to provide even parity for a combination of bits 3, 5, 7, 9, and 11. The value of r2 is calculated to provide even parity with bits 3, 7, 10, and 11, and so on. The final 11-bit code is sent through the transmission line.

Figure 7.17 Redundancy bits calculation



Error Detection and Correction

Now imagine that by the time the above transmission is received, the number 7 bit has been changed from 1 to 0 (see Figure 9.20).

The receiver takes the transmission and recalculates four new VRCs using the same sets of bits used by the sender plus the relevant panty (r) bit for each set (see Figure 9.21). Then it assembles the new parity values into a binary number in order of r position r_8, r_4, r_2, r_1.In our example, this step gives us the binary number 0111 (7 in decimal), which is the precise location of the bit in error. Once the bit is identified, the receiver can reverse its value and correct the error.

Burst Error Correction

A Hamming code can be designed to correct burst errors of certain lengths. The number of redundancy bits required to make these corrections, however, is dramatically higher than that required for single-bit errors. To correct double-bit errors, for example, we must take into consideration that the two bits can be a combination of any two bits in the entire sequence. Three-bit correction means any three bits in the entire sequence, and so on. So the simple strategy used by the Hamming code to correct single-bit errors must be redesigned to be applicable for multiple-bit correction

8. DATA LINK CONTROL

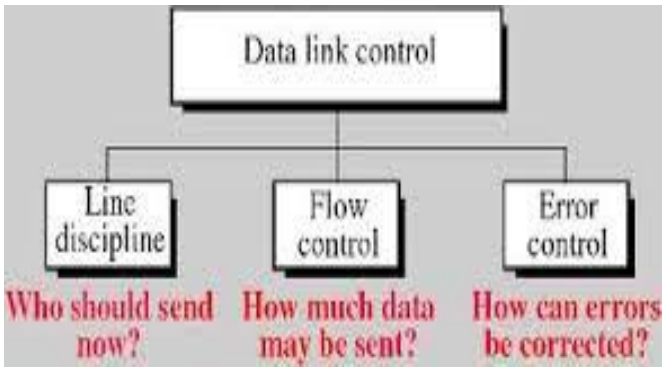
Data Link Control

Communication requires at least two devices working together, one to send and one to receive. Even such a basic arrangement requires a great deal of coordination for an intelligible exchange to occur. For example, in half-duplex transmission, it is essential that only one device transmit at a time. If both ends of the link put signals on the line simultaneously, they collide, leaving nothing on the line but noise. The coordination of half-duplex transmission is part of a procedure called **line discipline**, which is one of the functions included in the second layer of the OSI model, the data link layer.

In addition to line discipline most important functions in the data link layer are **flow control** and **error control**.

- Line discipline coordinates the link systems. It determines which device can send and when it can send.
- Flow control coordinates the amount of data that can be sent before receiving acknowledgment. It also provides the receiver's acknowledgment of frames received intact, and so is linked to error control.
- Error control means error detection and correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender (see Figure 8.1).

Figure 8.1 Data link layer functions



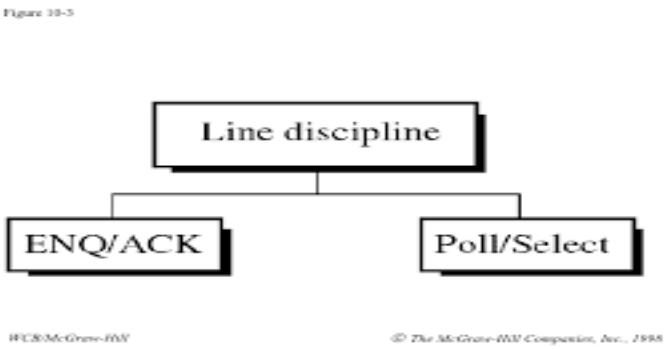
8.1 LINE DISCIPLINE

Whatever the system, no device in it should be allowed to transmit until that device has evidence that the intended receiver is able to receive and is prepared to accept the transmission. What if the receiving device does not expect a transmission, is busy, or is out of commission? With no way to determine the status of the intended receiver, the transmitting device may waste its time sending data to a nonfunctioning receiver or may interfere with signals already on the link. The line discipline functions of the data link layer oversee the establishment of links and the right of a particular device to transmit at a given time.

Line discipline answers the question. Who should send now?

Line discipline can be done in two ways: enquiry/acknowledgment (ENQ/ACK) and poll/select. The first method is wed in peer-to-peer communication the second method is used in primary, secondary communication (see Figure 8.2).

Figure 8.2 Line discipline categories

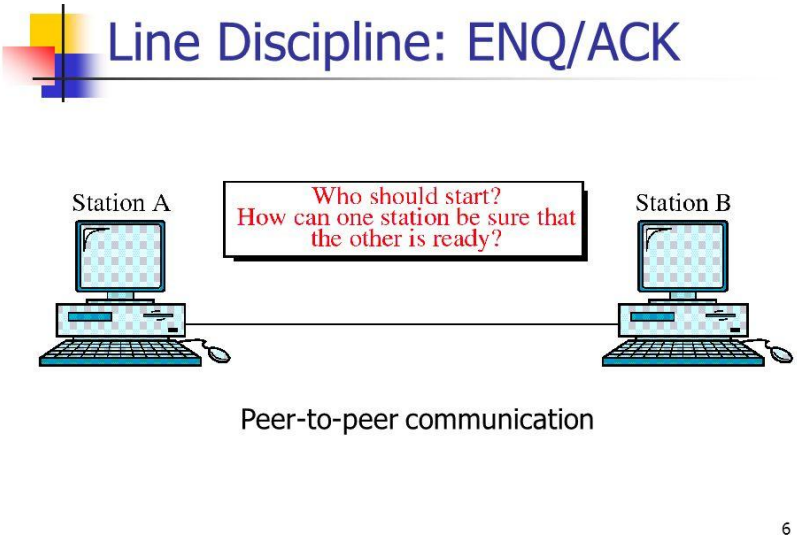


ENQ/ACK

Enquiry/acknowledgment (ENQ/ACK) is used primarily in systems where there is no question of the wrong receiver getting the transmission, that is, when there is a dedicated link between two devices so that the only device capable of receiving the trans-mission is the intended one.

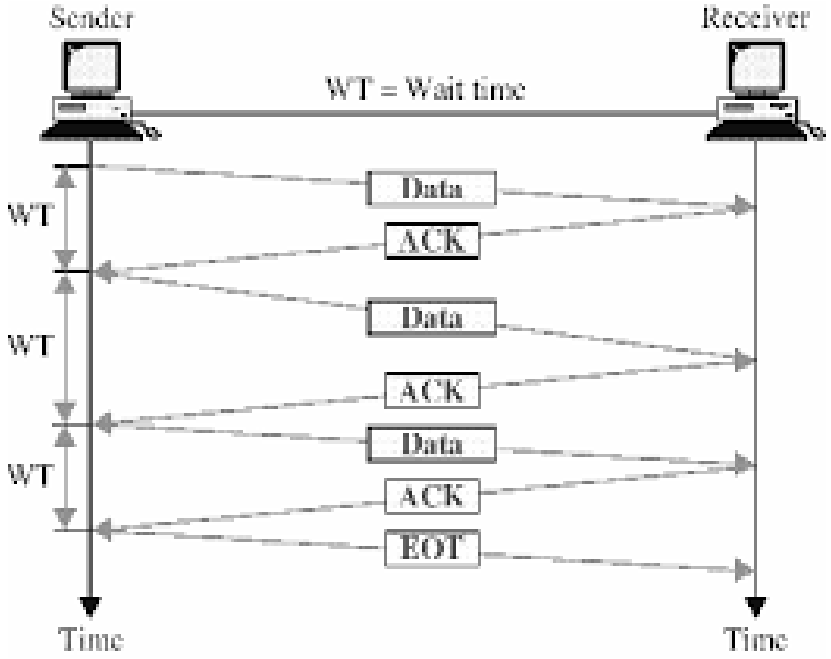
ENQ/ACK coordinates which device may start a transmission and whether or not the intended recipient is ready and enabled (see Figure 8.3). Using ENQ/ACK, a session can be

Figure 8.3 Line discipline concept ENQ/ACK



initiated by either station on a link as long as both are of equal rank-a printer, for example, cannot initiate communication with a CPU.

Figure 8.4 Line discipline concept: ENQ/ACK



In both half-duplex and full-duplex transmission, the initiating device establishes the session. In half-duplex, the initiator then sends its data while the responder waits. The responder may take over the link when the initiator is finished or has requested a response. In full-duplex, both devices can transmit simultaneously once the session has been established.

How It Works The initiator first transmits a frame called an enquiry (ENQ) asking if the receiver is available to receive data. The receiver must answer either with an **acknowledgement (ACK)** frame if it is ready to receive or with a **negative acknowledgement (NAK)** frame if it is not. By requiring a response even if the answer is negative, the initiator knows that its enquiry was in fact received given if the receiver is currently unable to accept a transmission. If neither an ACK nor a NAK is received within a specified time limit, the initiator assumes that the ENQ frame was lost in transit, disconnects, and sends a replacement. An initiating system ordinarily makes three such attempts to establish a link before giving up.

If the response to the ENQ is negative for three attempts, the initiator disconnects and begins the process again at another time. If the response is positive, the initiator is free to send its data. Once all of its data have been transmitted, the sending system finishes with an end of transmission (EOT) frame. This process is illustrated in Figure 8.4.

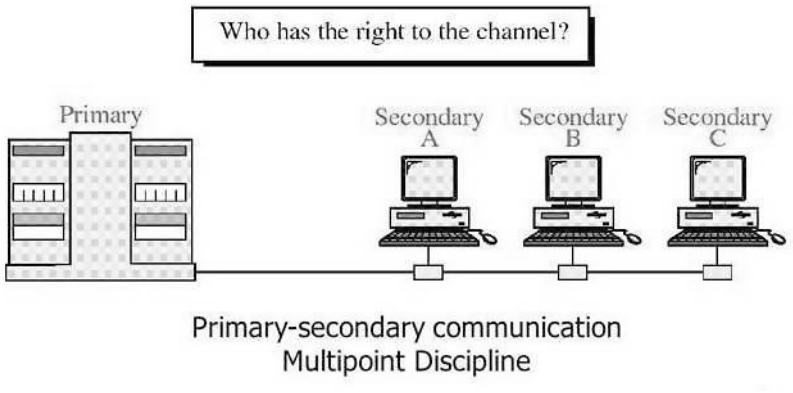
Poll/Select

The **poll/select** method of line discipline works with topologies where one device is designated as a **primary station** and the other devices are **secondary stations**. Multipoint systems must coordinate several nodes, not just two. The question to be deter-mined in these cases, therefore, is more than just, Are you ready? It is also, which of the several nodes has the right to use the channel?

How It Works Whenever a multipoint link consists of a primary device and multiple secondary devices using a single transmission line, all exchanges must be made through the primary device even when the ultimate destination is a secondary device. The primary device controls the link; the secondary devices follow its instructions. It is up to the primary to

determine which device is allowed to use the channel at a given time (see Figure 8.5). The primary, therefore, is always the initiator of a session. If the primary wants to receive data, it asks the secondaries if they have anything to send; this function is called polling. If the primary wants to send data, it tells the target secondary to get ready to receive; this function is called selecting.

Figure 8.5 Poll/select discipline



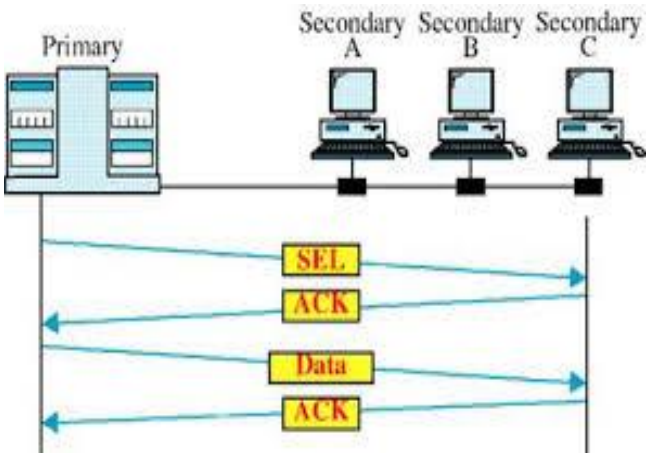
Addresses For point-to-point configurations, there is no need for addressing; any transmission put onto the link by one device can be intended only for the other. For the primary device in a multipoint topology to be able to identify and communicate with a specific secondary device, however, there must be an addressing convention. For this reason, every device on a link has an address that can be used for identification.

Poll/select protocols identify each frame as being either to or from a specific device on the link. Each secondary device has an address that differentiates it from the others. In any transmission, that address will appear in a specified portion of each frame, called an address field or header depending on the protocol. If the transmission comes from the primary device, the address indicates the recipient of the data. If the transmission comes from a secondary device, the address indicates the originator of the data

Select The select mode is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is not either sending or receiving data, it knows the link is available. If it has something to send, it sends it. What it does not know, however, is whether the target device is prepared to receive. So the primary must alert the secondary to the upcoming transmission and wait for an acknowledgment of the secondary's ready status. Before sending data, the primary creates and transmits a select (SEL) frame, one field of which includes the address of the intended secondary. Multipoint topologies use a single link for several devices, which means that any frame on the link is available to every device. As a frame makes its way down the link, each of the secondary devices checks the address field. Only when a device recognizes its own address does it open the frame and read the data. In the case of a SEL frame, the enclosed data consist of an alert that data are forthcoming.

If the secondary is awake and running, it returns an ACK frame to the primary. The primary then sends one or more data frames, each addressed to the intended secondary. Figure 8.6 illustrates this procedure.

Figure 8.6 Select

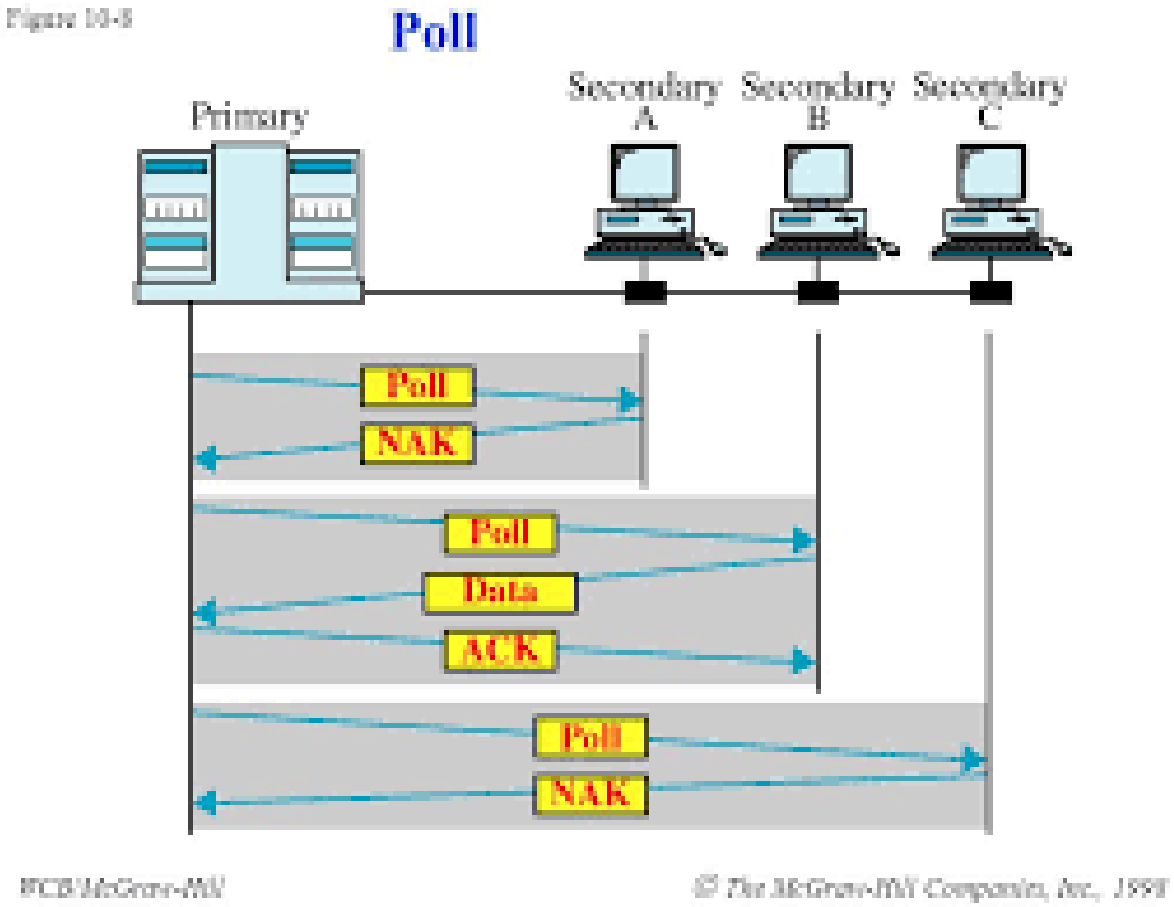


Poll The polling function is used by the primary device to solicit transmissions from the secondary devices. As noted above, the secondaries are not allowed to transmit data unless asked (don't call us we'll call you). By keeping all control with the primary, the multipoint system guarantees that only one transmission can occur at a time, thereby ensuring against signal collisions without requiring elaborate precedence protocols. When the primary is ready to receive data, it must ask (poll) each device in turn if it has anything to send. When the first secondary is approached, it responds either with a NAK frame if it has nothing to send or with data (in the form of a data frame) if it does.

If the response is negative (a NAK frame), the primary then polls the next secondary in the same way until it finds one with data to send. When the response is positive (a data frame), the primary reads the frame and returns an acknowledgment (ACK frame) verifying its receipt. The secondary may send several data frames one after the other, or it may be required to wait for an ACK before sending each one, depending on the protocol being used.

There are two possibilities for terminating the exchange: either the secondary sends all its data, finishing with an end of transmission (EOT) frame, or the primary says, "Time's up." Which of these occurs depends on the protocol and the length of the message. Once a secondary has finished transmitting, the primary can poll the remaining devices. (Figure 8.7)

Figure 8.7 Poll



8.2 FLOW CONTROL

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

Flow control refers to a set of procedures used to restrict the amount of data send before waiting for acknowledgment.

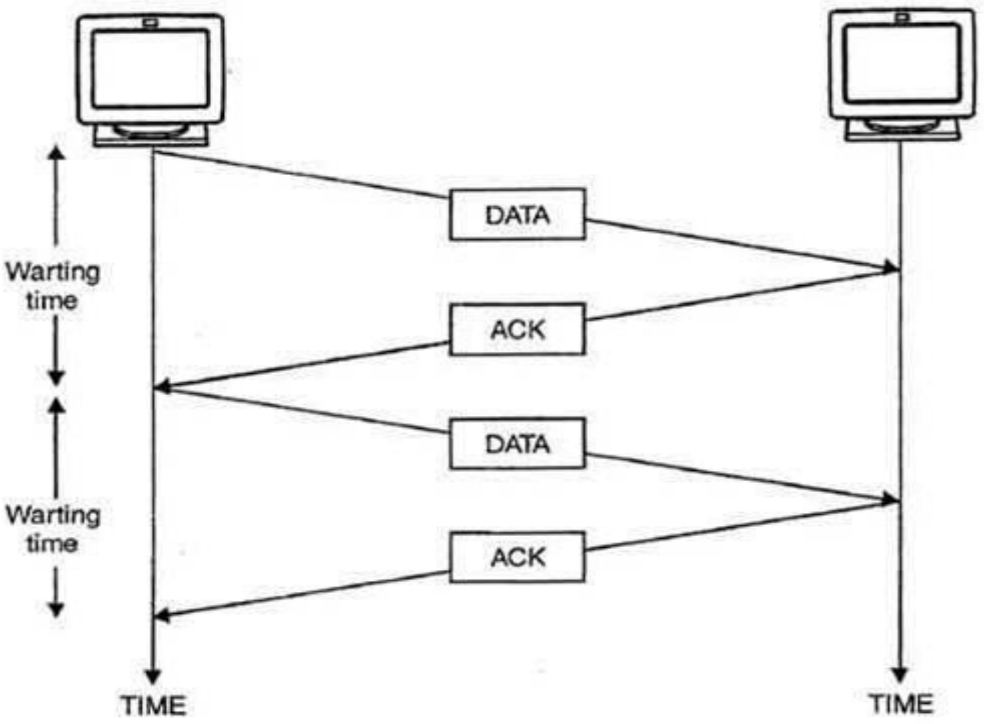
Two methods have been developed to control the flow of data across communications links: stop-and-wait and sliding window.

Stop-and-Wait

In a stop-and-wait method of flow control, the sender waits for an acknowledgment after every frame it sends (see Figure 8.8). Only when an acknowledgment has been received is the next frame sent. This process of alternately sending and waiting repeats until the sender transmits an end of transmission (EOT) frame. Stop-and-wait can be compared to a picky executive giving dictation: she says a word, her assistant says "OK," she says another word, her assistant says “OK,” and so on.

In the stop-and-wait method of flow control, the sender sends one frame and waits for an acknowledgment before sending the next frame.

Figure 8.8 Stop-and-wait



The advantage of stop-and-wait is simplicity: each frame is checked and acknowledged before the next frame is sent. The disadvantage is inefficiency: stop and wait is slow. Each frame must travel all the way to the receiver and an acknowledgment must travel all the way back before the next frame can be sent. In other words, each frame is alone on the line. Each frame sent and received uses the entire time needed to traverse the link. If the distance between devices is long, the time spent waiting for ACKs between each frame can add significantly to the total transmission time.

Sliding Window

In the sliding window method of flow control, the sender can transmit several frames before needing an acknowledgment. Frames can be sent one right after another, meaningthat the link can carry several frames at once and its capacity can be used efficiently. The receiver acknowledges only some of the frames, using a single ACK to confirm the receipt of multiple data frames.

In the sliding window method of flow control, several frames can be in transit at a time.

The sliding window refers to imaginary boxes at both the sender and the receiver. This window can hold frames at either end and provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgment. Frames may be acknowledged at any point without waiting for the window to fill up and may be trans-mitted as long as the window is not yet full. To keep track of which frames have been transmitted and which received, sliding window introduces an identification scheme based on the size of the window. The frames are numbered modulo-n, which means they are numbered from 0 to n - 1. For example, if n = 8, the frames are numbered 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, . . . The size of the window is n - 1 (in this case, 7). In other words, the window cannot cover the whole module (8 frames); it covers one frame less. The reason for this will be discussed at the end of this section.

When the receiver sends an ACK, it includes the number of the next frame it expects to receive. In other words, to acknowledge the receipt of a string of frames ending in frame 4, the receiver sends an ACK containing the number 5. When the sender sees an ACK with the number 5, it knows that all frames up through number 4 have been received.

The window can hold n- 1 frames at either end; therefore, a maximum of n - 1 frames may be sent before an acknowledgment is required.

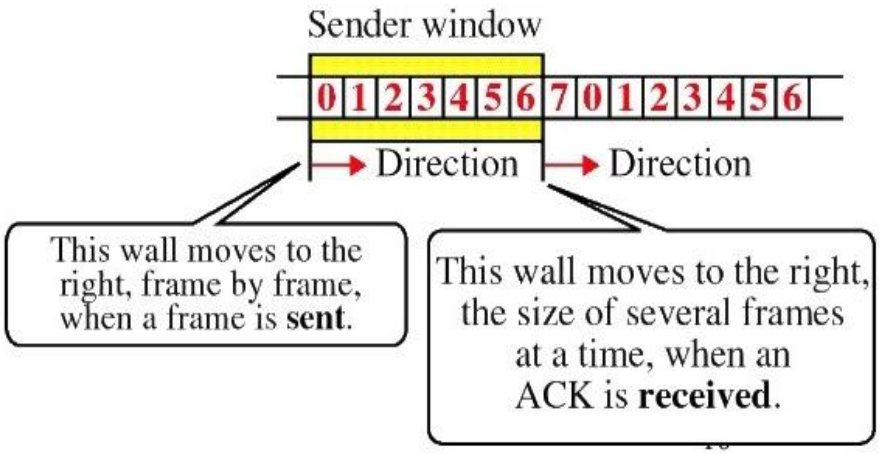
Sender Window

At the beginning of a transmission, the sender's window contains n — l frames. As frames are sent out, the left boundary of the window moves inward, shrinking the size of the window. Given a window of size w, if three frames have been transmitted since the last acknowledgment, then the number of frames left in the window is w 3. Once an ACK arrives, the window expands to allow in a number of new frames equal to the number of frames acknowledged by that ACK. Figure 10.12 shows a sender sliding window of size 7.

Given a window of size 7, as shown in Figure 8.9, if frames 0 through 4 have been sent and no acknowledgment has been received, the sender's window contains two frames (numbers 5 and 6). Now, if an ACK numbered 4 is received, four frames (0 through 3) are known to have arrived undamaged and the sender's window expands to include the next four frames in its buffer. At this point, the sender's window contains six frames (numbers 5, 6, 7, 0, 1, 2). If the received ACK had been numbered 2, the sender's window would have expanded by only two frames, to contain a total of four.

Conceptually, the sliding window of the sender shrinks from the left when frames of data are sent. The sliding window of the sender expands to the right when acknowledgments are received.

Figure 8.9 Sender sliding window

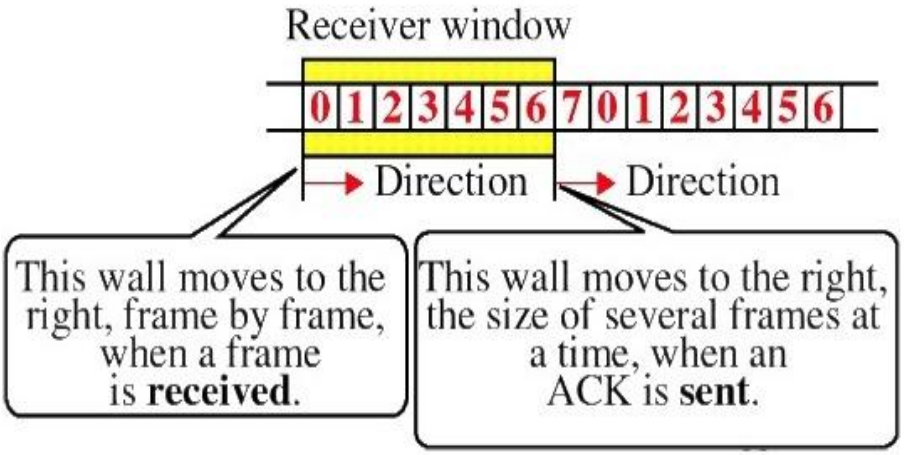


Receiver Window

At the beginning of transmission, the receiver window contains not n - 1 frames but n - 1 spaces for frames. As new frames come in, the size of the receiver window shrinks. The receiver window therefore represents not the number of frames received but the number of frames that may still be received before an ACK must be sent. Given a window of size w, if three frames are received without an acknowledgment being returned, the number of spaces in the window is w - 3. As soon as an acknowledgment is sent, the window expands to include places for a number of frames equal to the number of frames acknowledged.

Figure 8.10 shows a receiving window of size 7. In the figure, the window contains spaces for seven frames, meaning that seven frames may be received before an ACK must be sent. With the arrival of the first frame, the receiving window shrinks, moving the boundary from space 0 to 1. The window has shrunk by one, so the receiver may now accept six frames before it is required to send an ACK. If frames 0 through 3 have arrived but have not been acknowledged, the window will contain three frame spaces.

Figure 8.10 Receiver sliding window



Conceptually, the sliding window of the receiver shrinks from the left when frames of data are received. The sliding window of the receiver expands to the right when acknowledgments are sent.

As each ACK is sent out, the receiving window expands to include as many new placeholders as newly acknowledged frames. The window expands to include a number of new frame spaces equal to the number of the most recently acknowledged frame minus the number of the previously acknowledged frame. In a seven-frame window, if the prior ACK was for frame 2 and the current ACK is for frame 5, the window expands by three (5 - 2). If the prior ACK was for frame 3 and the current ACK is for frame 1. the window expands by six (1 + 8 - 3).

More about Window Size

In the sliding window method of flow control, the size of the window is one less than the modulo range so that there is no ambiguity in the acknowledgment of the received frames. Assume that the frame sequence numbers are modulo-8 and the window size is also 8. Now imagine that frame 0 is sent and ACK 1 is received. The sender expands its window and sends frames 1, 2, 3, 4, 5, 6, 7, and 0. If it now receives an ACK 1 again, it is not sure if this is a duplicate of the previous ACK 1 (duplicated by the network) or a new ACK 1 confirming the most recently sent eight frames. But if the window size is 7 (instead of 8), this scenario could not happen.

8.3 ERROR CONTROL

In the data link layer, the term error control refers primarily to methods of error detection and retransmission.

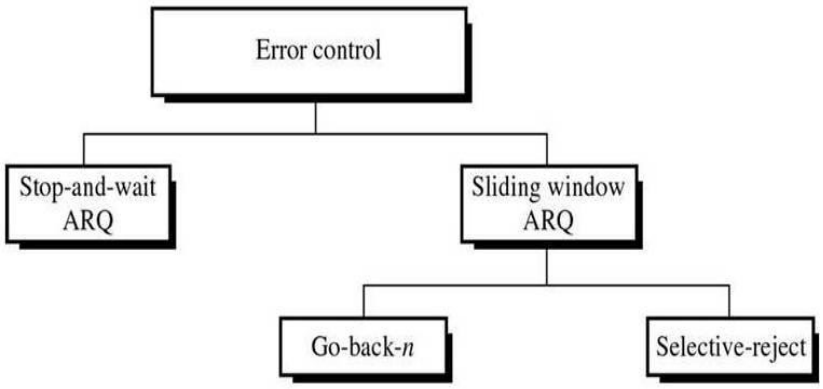
Automatic Repeat Request (ARQ)

Error correction in the data link layer is implemented simply: anytime an error is detected in an exchange, a negative acknowledgment (NAK) is returned and the specified frames are retransmitted. This process is called automatic repeat request (ARQ).

Error control in the data link layer is based on automatic repeat request (ARQ), which means retransmission of data in three cases: damaged frame, lost frame, and lost acknowledgment.

It sometimes happens that a frame is so damaged by noise during transmission that the receiver does not recognize it as a frame at all. In those cases, ARQ allows us to say that the frame has been lost. A second function of ARQ is the automatic retransmission of lost frames, including lost ACK and NAK frames (where the loss is detected by the sender instead of the receiver). ARQ error control is implemented in the data link layer as an adjunct to flow control. In fact, stop-and-wait flow control is usually implemented as stop-and-wait ARQ and sliding window is usually implemented as one of two variants of sliding window ARQ, called go-back-n or selective-reject (see Figure 8.11).

Figure 8.11 Categories of error control



Stop-and-Wait ARQ

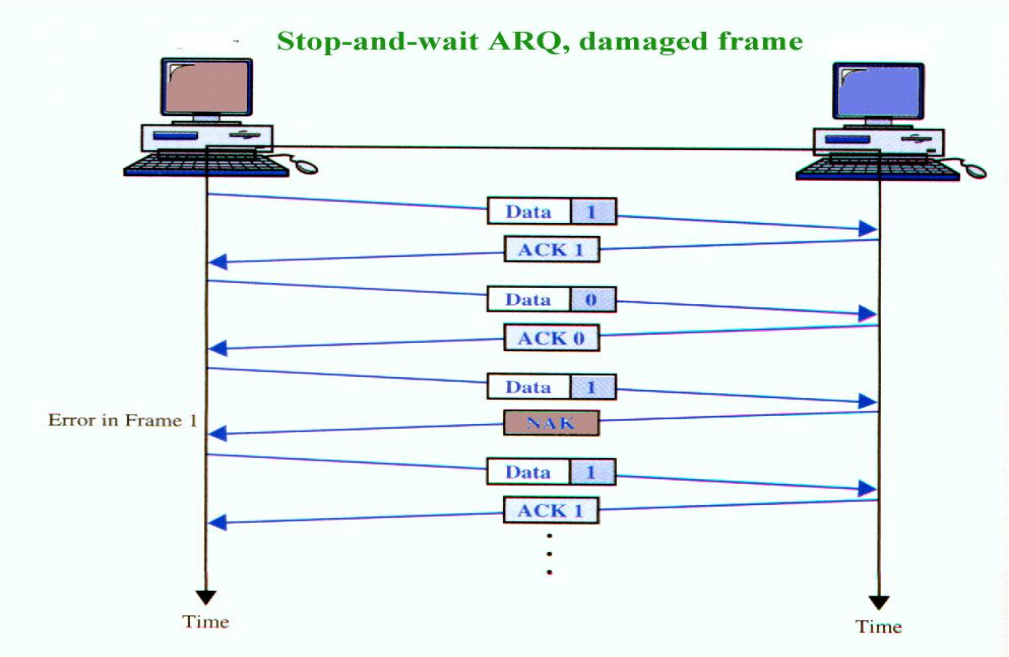
Stop-and-wait ARQ is a form of stop-and-wait flow control extended to include re-transmission of data in case of lost or damaged frames. For retransmission to work, four features are added to the basic flow control mechanism:

- The sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. Keeping a copy allows the sender to retransmit lost or damaged frames until they are received correctly.
- For identification purposes, both data frames and ACK frames are numbered alternately 0 and 1. A data 0 frame is acknowledged by an ACK I frame, indicating that the receiver has gotten data 0 and is now expecting data 1. This numbering allows for identification of data frames in case of duplicate transmission (important in the case of lost acknowledgments, as we will see below).
- If an error is discovered in a data frame, indicating that it has been corrupted in transit, a NAK frame is returned. NAK frames, which are not numbered, tell the sender to retransmit the last frame sent. Stop-and-wait ARQ requires that the sender wait until it receives an acknowledgment for the last frame transmitted before it transmits the next one. When the sending device receives a NAK, it re-sends the frame transmitted after the last acknowledgment, regardless of number.
- The sending device is equipped with a timer. If an expected acknowledgment is not received within an allotted time period, the sender assumes that the last data frame was lost in transit and sends it again.

Damaged Frames

When a frame is discovered by the receiver to contain an error, it returns a NAK frame and the sender retransmits the last frame. For example, in Figure 8.12, the sender transmits a data frame: data 0. The receiver returns an ACK 1, indicating that data 0 arrived undamaged and it is now expecting data L. The sender transmits its next frame: data 1. It arrives undamaged, and the receiver returns ACK 0. The sender transmits its next frame: data 0. The receiver discovers an error in data 0 and returns a NAK. The sender retransmits data 0. This time data 0 arrives intact, and the receiver returns ACK 1.

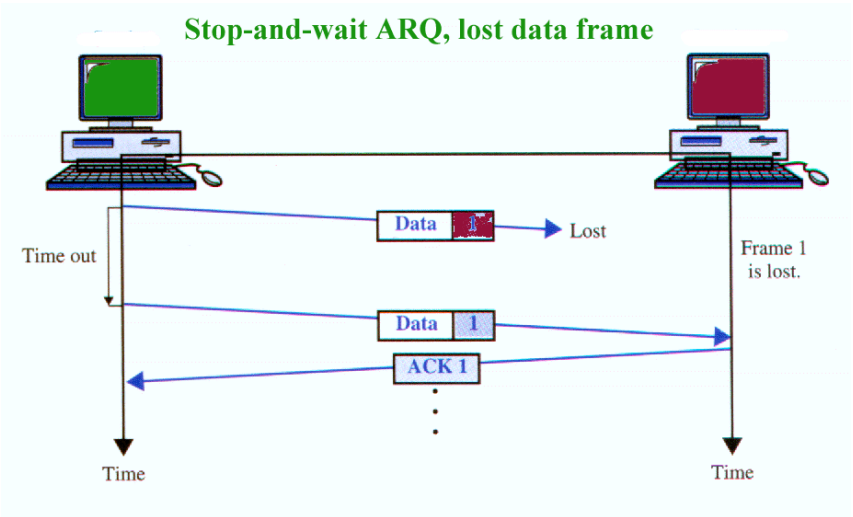
Figure 8.12 Stop-and-wait ARQ damaged frame



Lost Frame
Any of the three frame types can be lost in transit.

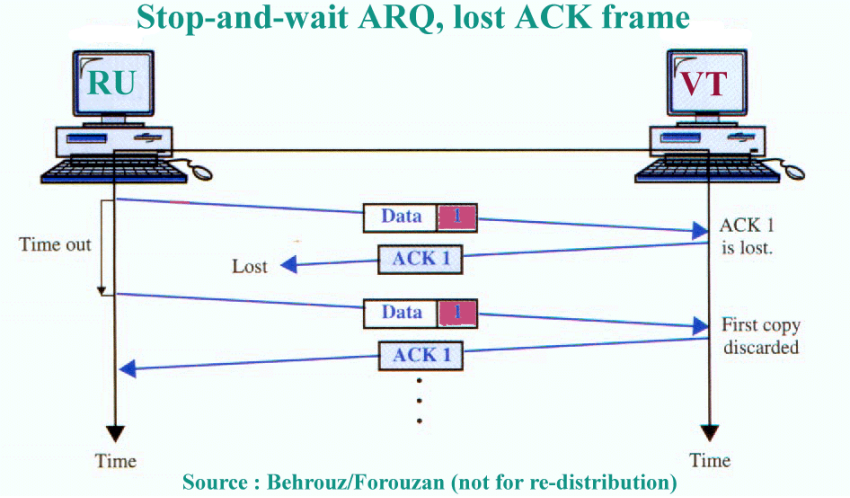
Lost Data Frame Figure 8.13 shows how stop-and-wait ARQ handles the loss of a data frame. As noted above, the sender is equipped with a timer that starts every time a data frame is transmitted. If the frame never makes it to the receiver, the receiver can never acknowledge it, positively or negatively. The sending device waits for an ACK or NAK frame until its timer goes off, at which point it tries again. It retransmits the last data frame, restarts its timer, and waits for an acknowledgment.

Figure 8.13 Stop-and-wait ARQ lost data frame



Lost Acknowledgment in this case, the data frame has made it to the receiver and has been found to be either acceptable or not acceptable. But the ACK or NAK frame returned by the receiver is lost in transit. The sending device waits until its timer goes off, then retransmits the data frame. The receiver checks the number of the new data frame. If the lost frame was a NAK, the receiver accepts the new copy and returns the appropriate ACK (assuming the copy arrives undamaged). If the lost frame was an ACK, the receiver recognizes the new copy as a duplicate, acknowledges. (Figure 8.14)

Figure 8.14 Stop-and-wait ARQ lost ACK frame



Sliding Window ARQ

Among the several popular mechanisms for continuous transmission error control, two protocols are the most popular: go-back-n ARQ and selective-reject ARQ, both based on sliding window flow control. To extend sliding window to cover retransmission of lost or damaged frames, three features are added to the basic flow control mechanism:

- The sending device keeps copies of all transmitted frames until they have been acknowledged. If frames 0 through 6 have been transmitted, and the last acknowledgment was for frame 2 (expecting 3), the sender keeps copies of frames 3 through 6 until it knows that they have been received undamaged.
- In addition to ACK frames, the receiver has the option of returning a NAK frame if the data have been received damaged. The NAK frame tells the sender to retransmit a damaged frame. Because sliding window is a continuous transmission mechanism (as opposed to stop-and-wait), both ACK and NAK frames must be numbered for identification. ACK frames, you will recall, carry the number of the next frame expected. NAK frames, on the other hand, carry the number of the damaged frame itself. In both cases, the message to the sender is the number of the frame that thereceiver expects next. Note that data frames that are received without errors do not have to be acknowledged individually. If the last ACK was numbered 3, an ACK 6 acknowledges the receipt of frames 3 and 4 as well as frame 5. Every damaged frame, however, must be acknowledged. If data frames 4 and 5 are received dam-aged, both NAK 4 and NAK 5 must be returned. However, a NAK 4 tells the sender that all frames received before frame 4 have arrived intact.
- Like stop-and-wait ARQ, the sending device in sliding window ARQ is equipped with a timer to enable it to handle lost acknowledgments. In sliding window ARQ, n -1 frames (the size of the window) may be sent before an acknowledgment must be received. If n -

1 frames are awaiting acknowledgment, the sender starts a timer and waits before sending any more. If the allotted time has run out with no acknowledgment, the sender assumes that the frames were not received and re-transmits one or all of the frames depending on the protocol. Note that as with stop-and-wait ARQ, the sender here has no way of knowing whether the lost frames are data, ACK, or NAK frames. By retransmitting the data frames, two possibilities are covered: lost data and lost NAK. If the lost frame was an ACK frame

Go-Back-n ARQ

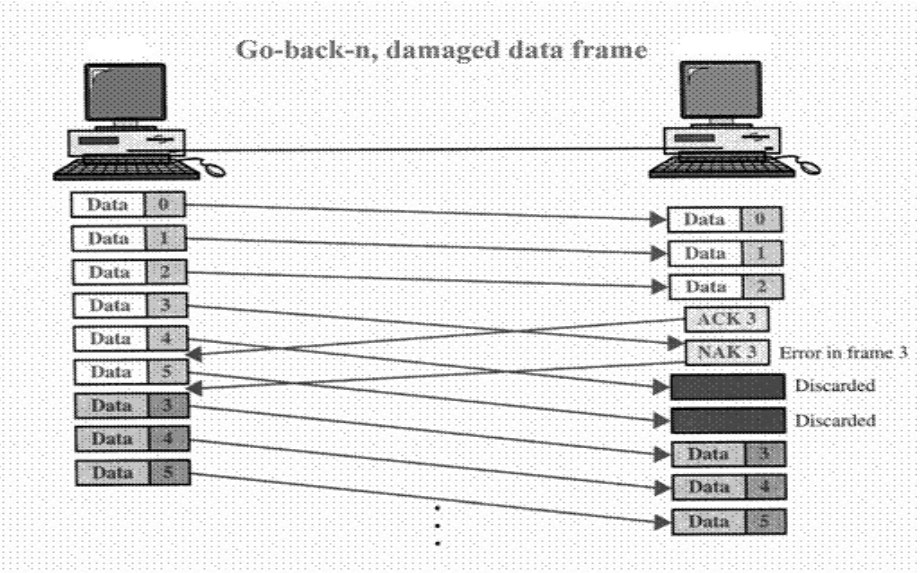
In this sliding window go-back-n ARQ method, if one frame is lost or damaged, all frames sent since the last frame acknowledged are retransmitted.

Damaged Frame What if frames 0, 1, 2, and 3 have been transmitted, but the first acknowledgment received is a NAK 3? Remember that a NAK means two things: (1) a positive acknowledgment of all frames received prior to the damaged frame and (2) a negative acknowledgment of the frame indicated. If the first acknowledgment is a NAK 3, it means that data frames 0, 1, and 2 were all received in good shape. Only frame 3 must be resent.

What if frames 0 through 4 have been transmitted before a NAK is received for frame 2? As soon as the receiver discovers an error, it stops accepting subsequent frames until the damaged frame has been replaced correctly. In the scenario above, data 2 arrives damaged and so is discarded, as are data 3 and data 4 whether or not they have arrived intact. Data 0 and data 1, which were received before the damaged frame, have already been accepted, a fact indicated to the sender by the NAK 2 frame. The retransmission therefore consists of frames 2, 3, and 4.

Figure 8.15 gives an example where six frames have been transmitted before an error is discovered in frame 3. In this case, an ACK 3 has been returned, telling the sender that frames 0, 1, and 2 have all been accepted. In the figure, the ACK 3 is sent before data 3 has arrived. Data 3 is discovered to be damaged, so a NAK 3 is sent immediately and frames 4 and 5 are discarded as they come in. The sending device retransmits all three frames (3, 4, and 5) sent since the last acknowledgment, and the process continues. The receiver discards frames 4 and 5 (as well as any subsequent frames) until it receives a good data 3.

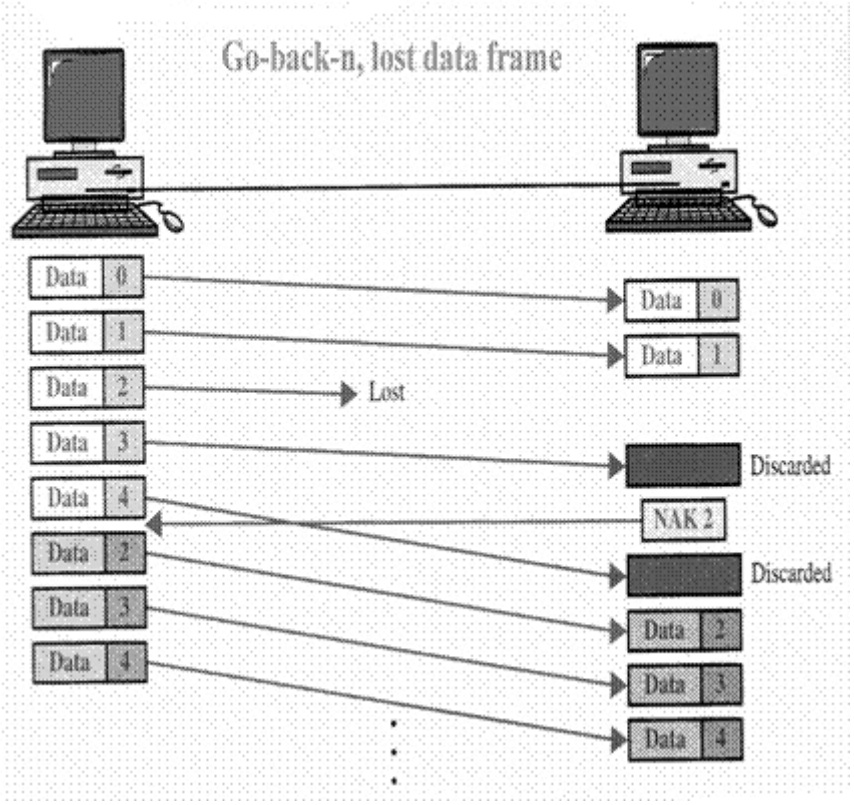
Figure 8.15 Go-back-n damaged data frame



Lost Data Frame Sliding window protocols require that data frames be transmitted sequentially. If one or more frames are so noise corrupted that they become lost in transit, the next frame to arrive at the receiver will be out of sequence. The receiver checks the identifying number on each frame, discovers that one or more have been skipped, and returns a NAK for the first missing frame. A NAK frame does not indicate whether the frame has been lost or damaged, just that it needs to be resent. The sending device then retransmits the frame indicated by the NAK, as well as any frames that it had transmitted after the lost one.

In Figure 8.16, data 0 and data 1 arrive intact but data 2 is lost. The next frame to arrive at the receiver is data 3. The receiver is expecting data 2 and so considers data .3 to be an error, discards it, and returns a NAK 2, indicating that 0 and 1 have been accepted but 2 is in error (in this case lost). In this example, because the sender has transmitted data 4 before receiving the NAK 2, data 4 arrives at the destination out of sequence and is therefore discarded. Once the sender receives the NAK 2, it retransmits all three pending frames (2, 3, and 4).

Figure 8.16 Go-back-n lost data frame



Lost Acknowledgment The sender is not expecting to receive an ACK frame for every data frame it sends. It cannot use the absence of sequential ACK numbers to identify lost ACK or NAK frames. Instead, it uses a timer. The sending device can send as many frames as the window allows before waiting for an acknowledgment. Once that limit has been reached or the sender has no more frames to send, it must wait. 318 CHAPTER 10 DATA LINK CONTROL

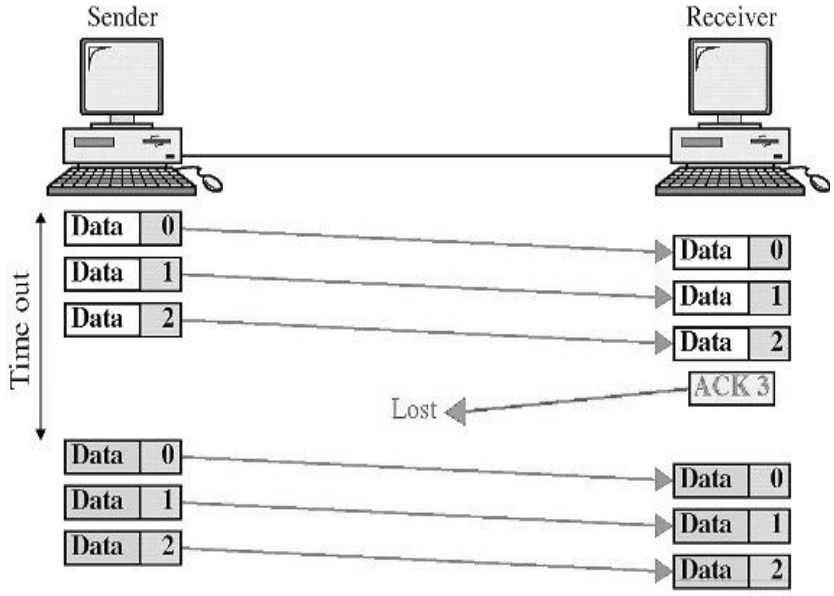
Figure 10.20 Go-back-n, lost data frame

If the ACK (or, especially, if the NAK) sent by the receiver has been lost, the sender could wait forever. To avoid tying up both devices, the sender is equipped with a timer that

begins counting whenever the window capacity is reached. If an acknowledgment has not been received within the time limit, the sender retransmits every frame trans-mitted since the last ACK.

Figure 8.17 shows a situation in which the sender has transmitted all of its frames and is waiting for an acknowledgment that has been lost along the way. The sender waits a predetermined amount of time, then retransmits the unacknowledged frames. The receiver recognizes that the new transmission is a repeat of an earlier one, sends another ACK, and discards the redundant data.

Figure 8.17 Go-back-n lost ACK



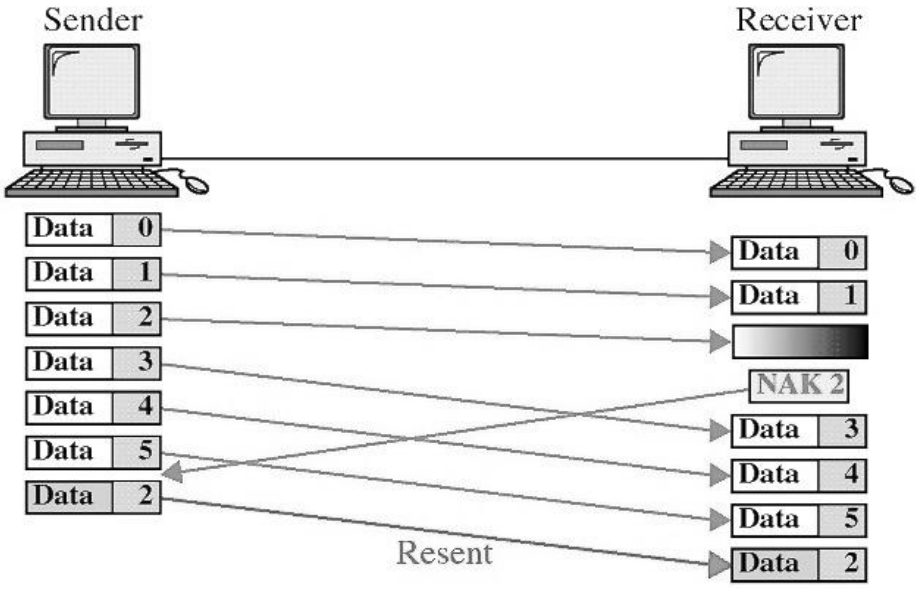
Selective-Reject ARQ

In selective-reject ARQ, only the specific damaged or lost frame is retransmitted. If a frame is corrupted in transit, a NAK is returned and the frame is resent out of sequence. The receiving device must be able to sort the frames it has and insert the retransmitted frame into its proper place in the sequence. To make such selectivity possible, a selective reject ARQ system differs from a go-back-n ARQ system in the following ways:

- The receiving device must contain sorting logic to enable it to reorder frames received out of sequence. It must also be able to store frames received after a NAK has been sent until the damaged frame has been replaced.
- The sending device must contain a searching mechanism that allows it to find and select only the requested frame for retransmission.
- A buffer in the receiver must keep all previously received frames on hold until all retransmissions have been sorted and any duplicate frames have been identified and discarded.
- To aid selectivity, ACK numbers, like NAK numbers, must refer to the frame received (or lost) instead of the next frame expected.
- This complexity requires a smaller window size than is needed by the go-back-n method if it is to work efficiently. It is recommended that the window size be less than or equal to (n + 1)/2, where n - 1 is the go-back-n window size.

Damaged Frames Figure 8.18 shows a situation in which a damaged frame is received. As you can see, frames 0 and 1 are received but not acknowledged. Data 2 arrives and is found to contain an error, so a NAK 2 is returned. Like NAK frames in go-back-n error correction, a NAK here both acknowledges the intact receipt of any previously unacknowledged data frames and indicates an error in the current frame. In the figure, NAK 2 tells the sender that data 0 and data 1 have been accepted, but that data 2 must be resent. Unlike the receiver in a go-back-n system, however, the receiver in a selective-reject system continues to accept new frames while waiting for an error to be corrected. However, because an ACK implies the successful receipt not only of the specific frame indicated but of all previous frames, frames received after the error frame cannot be acknowledged until the damaged frames have been retransmitted. In the figure, the receiver accepts data 3, 4, and 5 while waiting for a new copy of data 2. When the new data 2 arrives, an ACK 5 can be returned, acknowledging the new data 2 and the original frames 3, 4, and 5. Quite a bit of logic is required by the receiver to sort out-of-sequence retransmissions and to keep track of which frames are still missing and which have yet to be acknowledged.

Figure 8.18 Selective-reject damaged lost frame



Lost Frames Although frames can be accepted out of sequence, they cannot be acknowledged out of sequence. If a frame is lost, the next frame will arrive out of sequence. When the receiver tries to reorder the existing frames to include it, it will discover the discrepancy and return a NAK. Of course, the receiver will recognize the omission only if other frames follow. If the lost frame was the last of the transmission, the receiver does nothing and the sender treats the silence like a lost acknowledgment.

Lost Acknowledgment Lost ACK and NAK frames are treated by selective-reject ARQ just as they are by go-back-n ARQ. When the sending device reaches either the capacity of its window or the end of its transmission, it sets a timer. If no acknowledgment arrives in the time allotted, the sender retransmits all of the frames that remain unacknowledged. In most cases, the receiver will recognize any duplication and discard them.

Comparison between Go-Back-n and Selective-Reject

Although retransmitting only specific damaged or lost frames may seem more efficient than resending undamaged frames as well, it is in fact less so. Because of the complexity of the sorting and storage required by the receiver, and the extra logic needed by the sender to select specific frames for retransmission, selective-reject ARQ is expensive and not often used. In other words, selective-reject gives better performance, but in practice it is usually discarded in favor of go-back-n for simplicity of implementation.

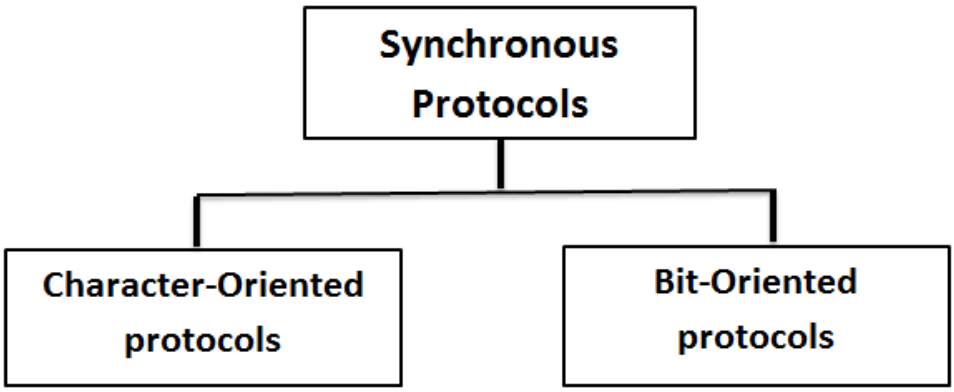
UNIT IV

9. DATA LINK PROTOCOLS

9.1 SYNCHRONOUS PROTOCOLS

The speed of synchronous transmission makes it the better choice, over asynchronous transmission, for LAN, MAN. And WAN technology. Protocols governing synchronous transmission can be divided into two classes: character-oriented protocols and bit-oriented protocols (see Figure 9.1).

Figure 9.1 Synchronous Protocols



Character-oriented protocols (also called byte-oriented protocols) interpret a transmission frame or packet as a succession of characters, each usually composed of one byte (eight bits). All control information is in the form of an existing character encoding system (e.g., ASCII characters).

Bit-oriented protocols interpret a transmission frame or packet as a succession of individual bits, made meaningful by their placement in the frame and by their juxtaposition with other bits. Control information in a bit-oriented proof can be one or multiple bits depending on the information embodied in the pattern)

In a character-oriented protocol, the frame or packet is interpreted as a series of characters. In a bit-oriented protocol, the frame or packet is interpreted as a series of bits.

9.2 CHARACTER-ORIENTED PROTOCOLS

For reasons we will examine later in this section, character-oriented protocols are not as efficient as bit-oriented protocols and therefore are now seldom used. They are, however, easy to comprehend and employ the same logic and organization as the bit-oriented protocols.

An understanding of character-oriented protocols provides an essential foundation for an examination of bit-oriented protocols. In all data link protocols, control information is inserted into the data stream either as separate control frames or as additions to existing data frames. In character-oriented protocols, this information is in the form of code words taken from existing character sets such as ASCII or EBCDIC. These multi bit characters carry information about line discipline, flow control, and error control. Of the several existing character-oriented protocols, the best known is IBM's binary synchronous communication (BSC).

Binary Synchronous Communication (BSC)

Binary synchronous communication (BSC) is a popular character-oriented data link protocol developed by IBM in 1964. Usable in both point-to-point and multipoint configurations, it supports half-duplex transmission using stop-and-wait ARQ flow control and error correction. BSC does not support full-duplex transmission or sliding window protocol.

A popular character-oriented data link protocol is binary synchronous communication (BSC), which specifies half-duplex transmission with stop-and-wait ARQ. It was developed by IBM.

Control Characters

Table 9.1 is a list of standard control characters used in a BSC frame. Note that the character ACK is not used in this protocol. Remember that BSC uses stop-and-wait ARQ: acknowledgments must be either ACK 0 or ACK 1 to specify alternating data frames.

Table 9.1 Control characters for BSC

Character	ASCII Code	Function
ACK 0	DLE and 0	Good even frame received or ready to receive
ACK 1	DLE and 1	Good odd frame received
DLE	DLE	Data transparency marker
ENQ	ENQ	Request for a response
EOT	EOT	Sender terminating
ETB	ETB	End of transmission block : ACK required
ETX	ETX	End of text in a message
ITB	US	End of intermediate block in a multiblock transmission
NAK	NAK	Bad frame received or nothing to send
NUK	NULL	Fill character
RVI	DLE and <	Urgent message from receiver
SOH	SOH	Header information begins
STX	STX	Text begins
SYN	SYN	Alerts receiver to incoming frame
TTD	STX and ENQ	Sender is pausing but not relinquishing the line
WACK	DLE and :	Good frame received but not ready to receive more

ASCII Codes

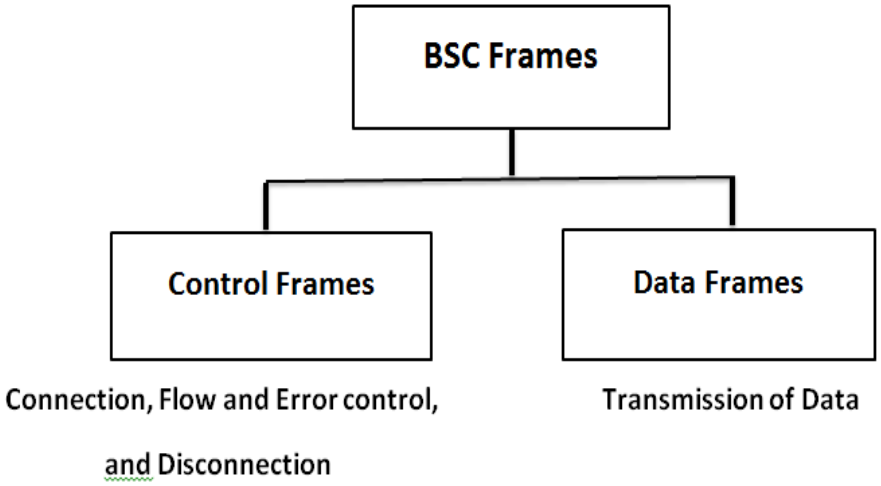
The characters in Table 11.1 are represented differently in different coding systems, and not all of them are available in every system. Whatever the system, not all control characters can be represented by a single character. Often they must be represented by two or three characters. The ASCII codes are also shown in Table 11.1. For a complete list of the ASCII code, see Appendix A.

BSC Frames

The BSC protocol divides a transmission into frames. If a frame is used strictly for control purposes, it is called a control frame. Control frames are used to exchange information between communicating devices, for example, to establish the initial connection, to control the flow of the transmission, to request error corrections, and to disconnect the devices at the close of a session. If a frame contains part or all of the message data itself, it is called a data

frame. Data frames are used to transmit information, but may also contain control information applicable to that information (see Figure 9.2).

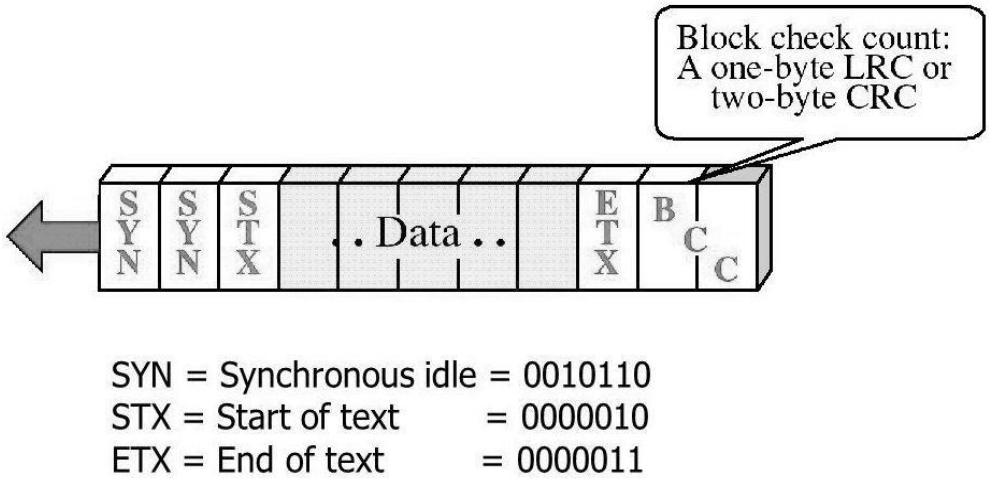
Figure 9.2 BSC frames



Data Frames

Figure 9.3 shows the format of a simple data frame. The arrow shows the direction of transmission. The frame begins with two or more synchronization (SYN) characters, These characters alert the receiver to the arrival of a new frame and provide a bit pattern used by the receiving device to synchronize its timing with that of the sending device.

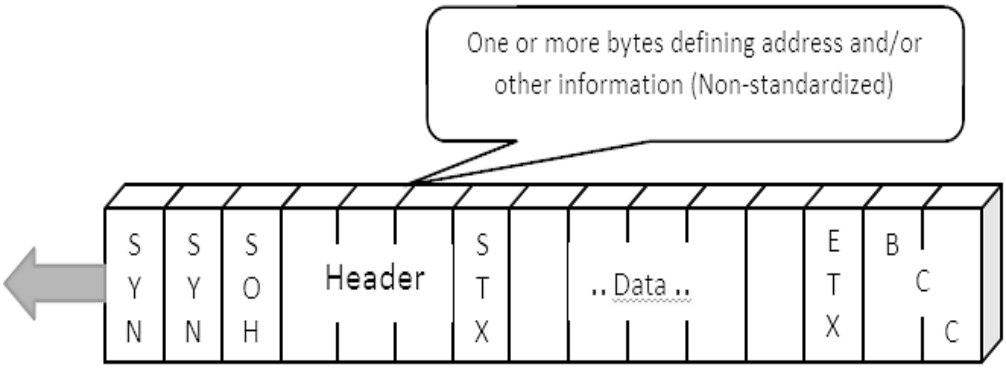
Figure 9.3 A simple BSC data frame



After the two synchronization characters comes a start of text (STX) character. This character signals to the receiver that the control information is ending and the next byte will be data. Data or text can consist of varying numbers of characters. An end of text (ETX) character indicates the transition between text and more control characters. Finally, one or two characters called the block check count (BCC) are included for error detection. A BCC field can be a one-character longitudinal redundancy check (LRC) or a two-character cyclic redundancy check (CRC).

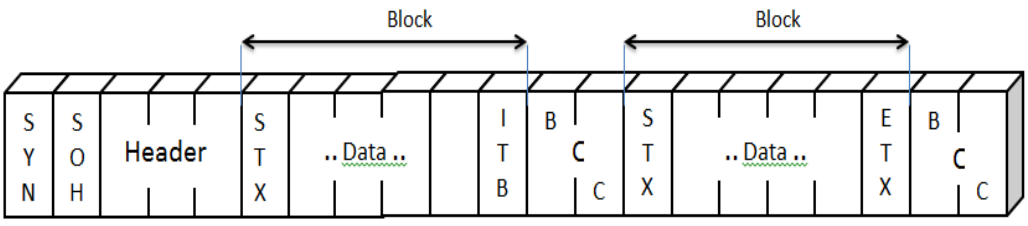
Header Fields A frame as simple as the one described above is seldom useful. Usu-ally we need to include the address of the receiving device, the address of the sending device, and the identifying number of the frame (0 or 1) for stop-and-wait ARQ (see Figure 9.4). This information is included in a special field called a header, which begins with a start of header (SOH) character. The header comes after the SYNs and before the STX character; everything received after the SOH field but before the STX character is header information.

Figure 9.4 S A BSC frame with header



Multiblock Frames The probability of an error in the block of text increases with the length of the frame. The more bits in a frame, the greater the likelihood that one of them will be corrupted in transit, and the greater the likelihood that changes in several bits will cancel each other out and make detection difficult. For this reason, text in a message is often divided between several blocks. Each block, except the last one, starts with an STX character and ends with an intermediate text block (ITB). The last block starts with an STX but ends with an ETX. Immediately after each ITB or ETX is a BCC field. In that way, the receiver can check each block separately for errors, thereby increasing the likelihood of detection. If any block contains an error, however, the entire frame must be retransmitted. After the ETX has been reached and the last BCC checked, the receiver sends a single acknowledgment for the entire frame. (Figure 9.5)

Figure 9.5 A multiblock frame



Multiframe Transmission In the examples explored above, a single frame carries an entire message. After each frame, the message is complete and control of the line passes to the secondary device (half-duplex mode). Some messages, however. may be too long to fit into the format of a single frame. In such cases, the sender can split the message not only among

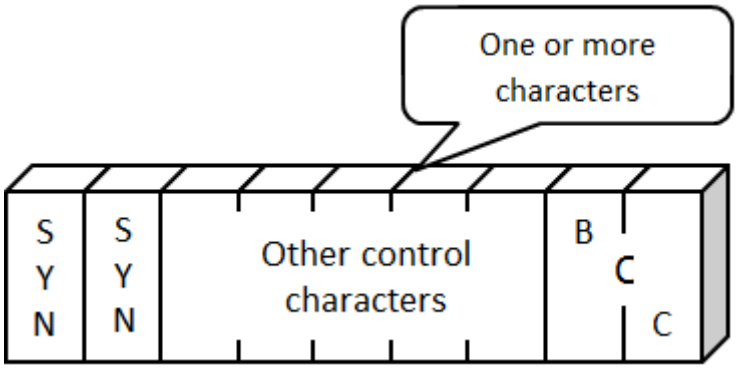
blocks but among frames. Several frames can carry continua-tions of a single message. To let the receiver know that the end of the frame is not the end of the transmission, the ETX character in all frames but the last one is replaced by an end of transmission block (ETB). The receiver must acknowledge each frame separately but can of take over control of the link until it sees the ETX in the last frame.

Control Frames

A control frame should not be confused with a control character. A control frame is used by one device to send commands to, or solicit information from, another device. A control frame contains control characters but no data; it carries information specific to the functioning of the data link layer itself. Figure 9.6 shows the basic format of a BSC control frame.

Control frames serve three purposes: establishing connections, maintaining flow and error control during data transmission, and terminating connections.

Figure 9.6 BSC control frame



Data Transparency

BSC was originally designed to transport only textual messages (words or figures composed of alphanumeric characters). Today, however, a user is just as likely to want to send binary sequences that contain non textual information and commands, like programs and graphics. Unfortunately, messages of this sort can create problems for BSC transmission. If the text field of a transmission includes an eight-bit pattern that looks like a BSC control character, the receiver interprets it as one, destroying the sense of the message. For example, a receiver seeing the bit pattern 000001 I reads it as an ETX character. As we learned from the control frames above, whenever a receiver finds an ETX, it expects the next two bytes to be the BCC and begins an error check. But the pattern 0000011 here is intended as data and not as control information. Confusion between control information and data is called a lack of data transparency.

For a protocol to be useful, it must be transparent-it must be able to carry any combination of bits as data without their being confused with control information.

Data transparency in data communication means we should be able to send any combination of bits as data.

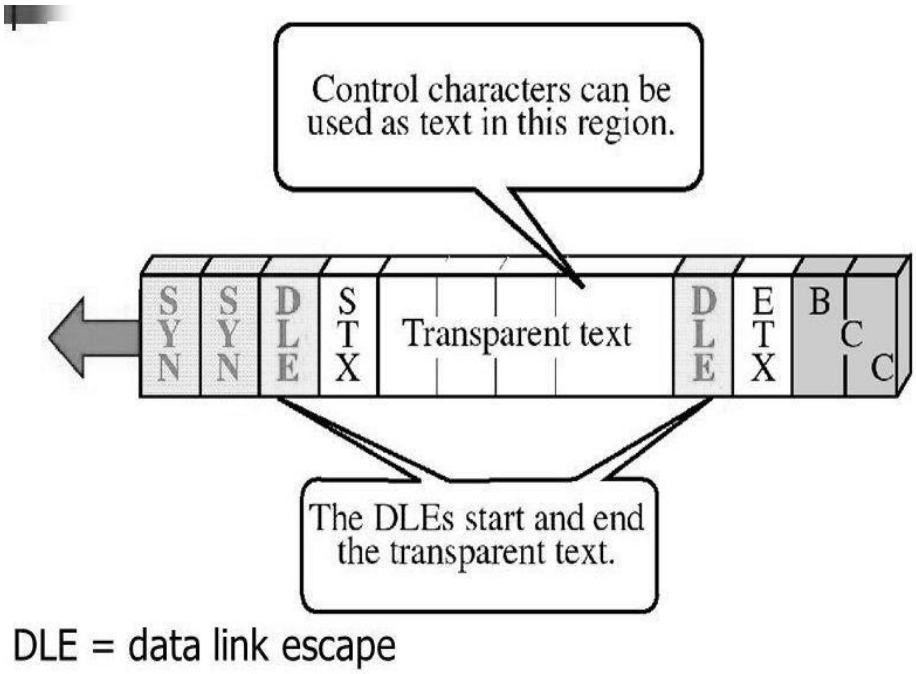
Data transparency in BSC is achieved by a process called **byte stuffing**. It involves two activities: defining the transparent text region with the data link escape (DLE) characters and preceding any DLE character within the transparent region by an extra DLE character.

To define the transparent region, we insert one DLE character just before the STX character at the beginning of the text field and another just before the ETX (or ITB or ETB) character at the end of the text field. The first DLE tells the receiver that the text may contain

control characters and to ignore them. The last DLE tells the receiver that the transparent region has ended.

Problems may still arise if the transparent region contains a DLE character as text. In that case, we insert an additional DLE just before each DLE within the text. Figure 9.7 shows an example of a transparent frame.

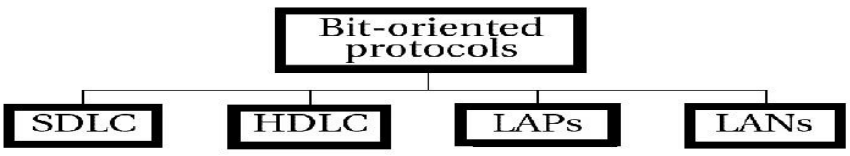
Figure 9.7 Byte stuffing



9.3 BIT-ORIENTED PROTOCOLS

In character-oriented protocols, bits are grouped into predefined patterns forming characters. By comparison, bit-oriented protocols can pack more information into shorter frames and avoid the transparency problems of character-oriented protocols. Given the advantages of bit-oriented protocols and the lack of any preexisting cod-ing system (like ASCII) to tie them to, it is no wonder that over the last two decades many different bit-oriented protocols have been developed, all vying to become the standard (see Figure 9.8). Most of these offerings have been proprietary, designed by manufacturers to support their own products. One of them, HDLC, is the design of the ISO and has become the basis for all bit-oriented protocols in use today.

Figure 9.8 Bit-oriented protocol



In 1975, IBM pioneered the development of bit-oriented protocols with synchronous data link control (SDLC) and lobbied the ISO to make SDLC the standard. In 1979, the ISO answered with high-level data link control (HDLC), which was based on SDLC. Adoption of HDLC by the ISO committees led to its adoption and extension by other organizations. The ITU-T was one of the first organizations to embrace HDLC. Since 1981, ITU-T has developed a series of protocols called link access proto-cols (LAPs: LAPB, LAPD, LAPM, LAPX, etc.), all based on HDLC. Other protocols (such as Frame Relay, PPP, etc.) developed by both ITU-T and ANSI also derive from HDLC, as do most LANs' access control protocols. In short, all hit-oriented protocols in use today either derive from or are sources for HDLC. Through HDLC, therefore, we have a basis for understanding the others.

All bit-oriented protocols are related to high-level data link control (HDLC), a bit-oriented protocol published by ISO. HDLC supports both half-duplex and full-duplex modes in point-to-point and multipoint configurations.

HDLC

HDLC is a bit-oriented data link protocol designed to support both half-duplex and full-duplex communication over point-to-point and multipoint links. Systems using HDLC can be characterized by their station types, their configurations, and their response modes.

Station Types

HDLC differentiates between three types of stations: primary, secondary, and combined.

A **primary station** in HDLC functions in the same way as the primary devices in the discussions of flow control in Chapter 10. The primary is the device in either a point-to-point or multipoint line configuration that has complete control of the link. The primary sends commands to the **secondary stations.** A primary issues commands; a secondary issues responses.

A **combined station** can both command and respond. A combined station is one of a set of connected peer devices programmed to behave either as a primary or as a secondary depending on the nature and direction of the transmission.

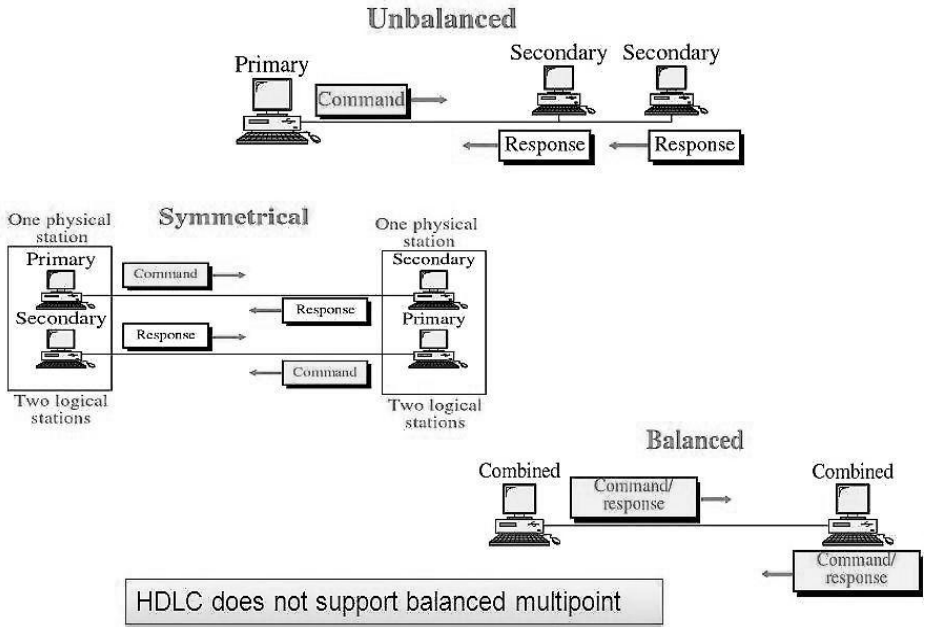
Stations in HDLC are of three types: primary, secondary, and combined. A primary station sends commands. 'A secondary station sends responses. A combined station sends commands and responses.

Configurations

The word configuration refers to the relationship of hardware devices on a link. Primary, secondary, and combined stations can be configured in three ways: unbalanced, symmetrical, and balanced (see Figure 9.9). Any of these configurations can support both half-duplex and full-duplex transmission.

An **unbalanced configuration** (also called a master/slave configuration) is one in which one device is primary and the others are secondary. Unbalanced configurations can be point-to-point if only two devices are involved; more often they are multipoint, with one primary controlling several secondaries.

Figure 9.9 HDLC Configuration



A **symmetrical configuration** is one in which each physical station on a link consists of two logical stations, one a primary and the other a secondary. Separate lines link the primary aspect of one physical station to the secondary aspect of another physical station. A symmetrical configuration behaves like an unbalanced configuration except that control of the link can shift between the two stations.

A **balanced configuration** is one in which both stations in a point-to-point topology are of the combined type. The stations are linked by a single line that can be con-trolled by either station.

HDLC does not support balanced multipoint. This necessitated the invention of media access protocols for LANs.

Modes of Communication

A mode in HDLC is the relationship between two devices involved in an exchange; the mode describes who controls the link. Exchanges over unbalanced configurations are always conducted in normal response mode. Exchanges over symmetrical or balanced configurations can be set to a specific mode using a frame designed to deliver the command. .HDLC supports three modes of communication between stations: normal response mode (NRM), asynchronous response mode (ARM), and asynchronous balanced mode (ABM).

NRM Normal response mode (NRM) refers to the standard primary secondary relationship. In this mode, a secondary device must have permission from the primary device before transmitting. Once permission has been granted, the secondary may initiate a response transmission of one or more frames containing data.

ARM In asynchronous response mode (ARM), a secondary may initiate a trans-mission without permission from the primary whenever the channel is idle. ARM does not alter the primary secondary relationship in any other way. All transmissions from a secondary (even to

another secondary on the same link) must still be made to the primary for relay to a final destination.

ABM In asynchronous balanced mode (ABM), all stations are equal and therefore only combined stations connected in point-to-point are used. Either combined station may initiate transmission with the other combined station without permission. Figure 11.15 shows the relationships between these modes and station types.

Modes:

- Normal response mode (NRM)
- Asynchronous response mode (ARM)
- Asynchronous balanced mode (ABM)

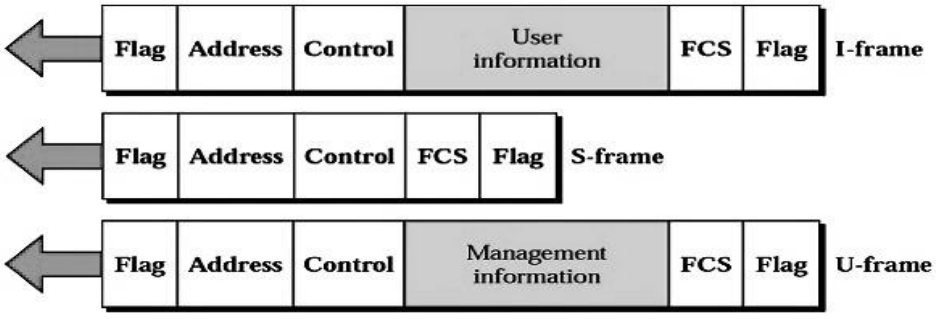
Figure 9.1 HDLC modes

	NRM	ARM	ABM
Station type	Primary & Secondary	Primary & Secondary	combined
Initiator	Primary	Either	Any

Frames

To provide the flexibility necessary to support all of the options possible in the modes and configurations described above, HDLC defines three types of frames: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (U-frames); see Figure 9.9. Each type of frame works as an envelope for the transmission of a different type of message. I-frames are used to transport user data and control information relating to user data. S-frames are used only to transport control information, primarily data link layer flow and error controls'-frames are reserved for system management. Information carried by U-frames is intended for managing the link itself. Each frame in HDLC may contain up to six fields: a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field. In multiple frame transmissions, the ending **flag** of one frame can double as the beginning flag of the next frame.

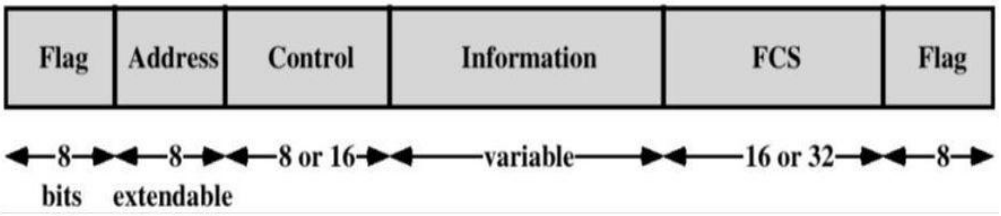
Figure 9.9 HDLC frame types



Flag Field

The flag field of an HDLC frame is an eight-bit sequence with a bit pattern 01111110 that identifies both /the beginning and end of a frame and serves as a synchronization pattern for the receiver. Figure 9.10 shows the placement of the two flag fields in an I-frame.

Figure 9.10 HDLC Flag field



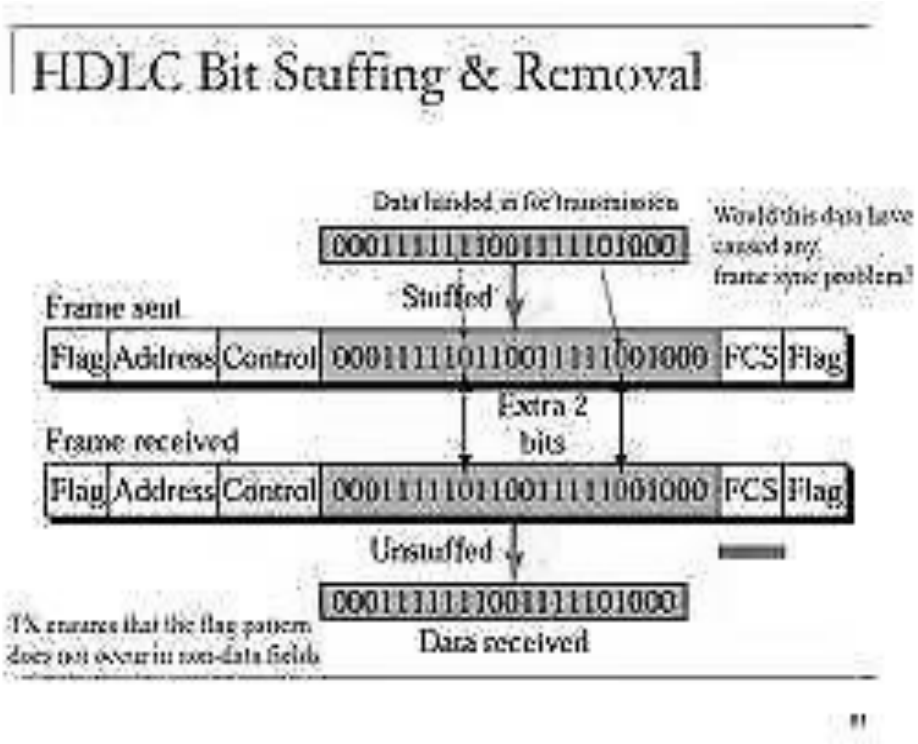
The flag field is the closest that HDLC comes to a control character that might be misread by a receiver. The flag field is also, therefore, HDLC's only potential cause of transparency problems. Once a station finds a flag on the line, determines that the frame is addressed to it, and begins reading the transmission, it is watching for the next flag that signifies the end of the frame. It is always possible that a bit sequence, whether control information or data, might contain the pattern 01111110. If that were to happen in the data, for example, the receiver would find it and assume that the end of the frame had been reached (with disastrous results).

To guarantee that a flag does not appear inadvertently anywhere else in the frame, HDLC uses a process called bit stuffing. Every time a sender wants to transmit a bit sequence having more than five consecutive I s, it inserts (stuffs) one redundant 0 after the fifth 1. For example, the sequence 011111111000 becomes 0111110111000. This extra 0 is inserted regardless of whether the sixth bit is another 1 or not. Its presence tells the receiver that the current sequence is not a flag. Once the receiver has seen the stuffed 0, it is dropped from the data and the original bit stream is restored.

Bit stuffing is the process of adding one extra 0 whenever there are five consecutive is in the data so that the receiver does not mistake the data for a flag.

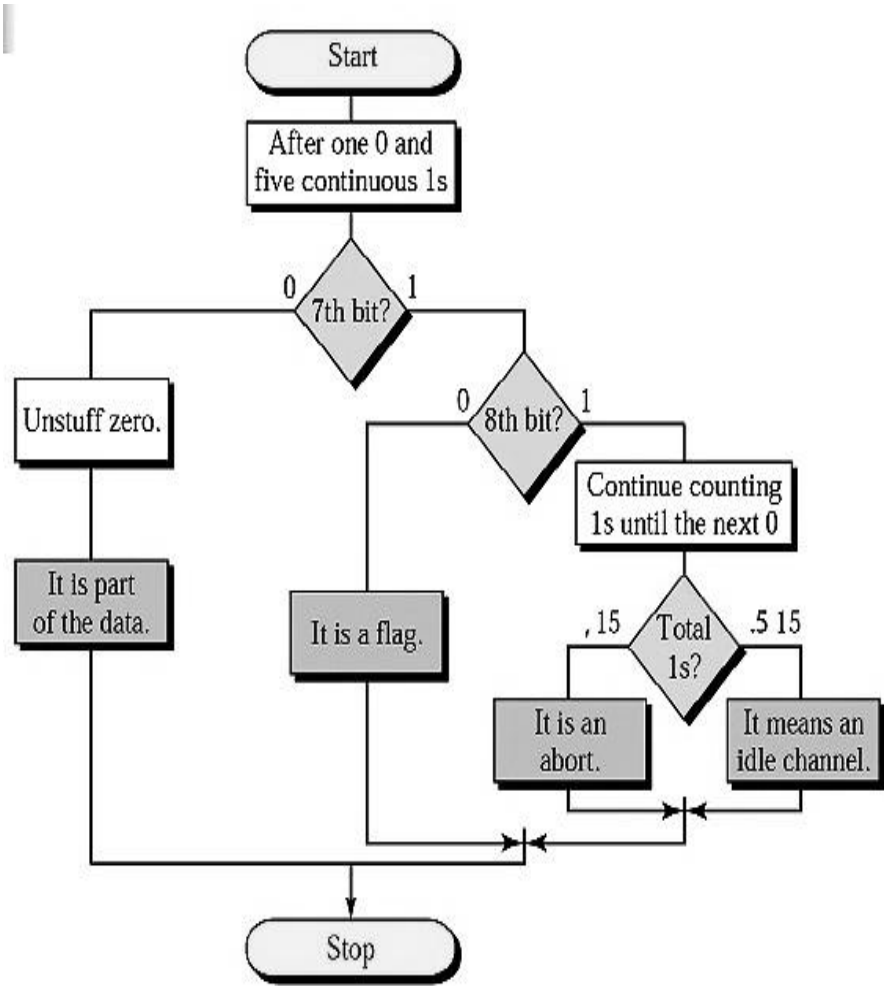
Figure 9.11 shows bit stuffing at the sender and bit removal at the receiver. Note that even if we have a 0 after five Is, we still stuff a 0. The 0 will be removed by the receiver.

Figure 9.11 Bit stuffing and removal



With three exceptions, bit stuffing is required whenever five ls occur consecutively. The exceptions are when the bit sequence really is a flag, when the transmission is being aborted, and when the channel is being put into idle. The flowchart in Figure 9.12 shows the process the receiver follows to identify and discard a stuffed bit. As the receiver reads the incoming bits, it counts ls. When it finds five consecutive is after a 0, it checks the next (seventh) bit. If the seventh bit is a 0, the receiver recognizes it as a stuffed bit, discards it, and resets its counter. If the seventh bit is a 1, the receiver checks the eighth bit. If the eighth bit is a 0, the sequence is recognized as a flag and treated accordingly. If the eighth bit is another 1, the receiver continues counting. A total of 7 to 14 consecutive ls indicates an abort. A total of 15 or more ls indicates an idle channel.

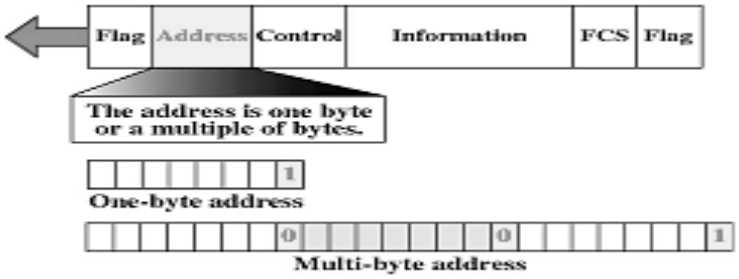
Figure 9.12 Bit stuffing in HDLC



Address Field

The second field of an HDLC frame contains the address of the secondary station that is either the originator or destination of the frame (or the station acting as secondary in the case of combined stations). If a primary station creates a frame, it contains a to address. If a secondary creates the frame, it contains a from address. An address field can be one byte or several bytes long, depending on the needs of the network. One byte can identify up to 128 stations (because one bit is used for another purpose). Larger networks require multiple-byte address fields. Figure 9.13 shows the address field in relation to the rest of the frame.

Figure 9.13 HDLC address field



If the address field is only one byte, the last bit is always a 1. If the address is more than one byte, all bytes but the last one will end with 0; only the last will end with I.

Ending each intermediate byte with 0 indicates to the receiver that there are more address bytes to come.

Control Field

The control field is a one- or two-byte segment of the frame used for flow management. We first discuss the one-byte case and then the two-byte case, called the extended mode.

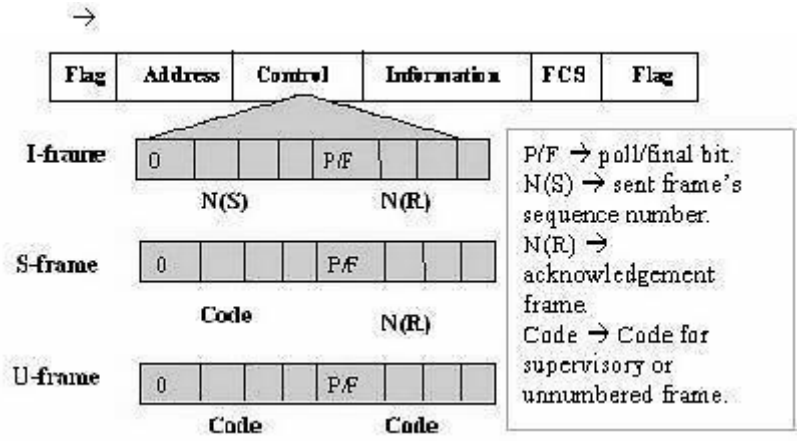
Control fields differ depending on frame type. If the first bit of the control field is 0, the frame is an I-frame. If the first bit is a 1 and the second bit is 0, it is an S-frame. If both the first and second bits are ls, it is a U-frame. The control fields of all three types of frames contain a bit called the poll/final (P/F) bit (discussed below).

An I-frame contains two 3-bit flow and error control sequences, called N(S) and N(R), flanking the P/F bit. N(S) specifies the number of the frame being sent (its own identifying number). N(R) indicates the number of the frame expected in return in a two-way exchange; thus N(R) is the acknowledgment field. If the last frame received was error-free, the N(R) number will be that of the next frame in the sequence. If the last frame was not received correctly, the N(R) number will be the number of the dam-aged frame, indicating the need for its retransmission.

The control field of an S-frame contains an N(R) field but not an N(S) field. S-frames are used to return N(R) when the receiver does not have data of its own to send. Otherwise the acknowledgment is contained in the control field of an I-frame (above). S-frames do not transmit data and so do not require N(S) fields to identify them. The two bits preceding the P/F bit in an S-frame are used to carry coded flow and error control information, which we will discuss later in this chapter.

U-frames have neither N(S) nor N(R) fields, and are not designed for user data exchange or acknowledgment. Instead, U-frames have two code fields, one two bits and the other three, flanking the P/F bit. These codes are used to identify the type of U-frame and its function (e.g., establishing the mode of an exchange). The control fields of all three types of frames are shown in Figure 9.14.

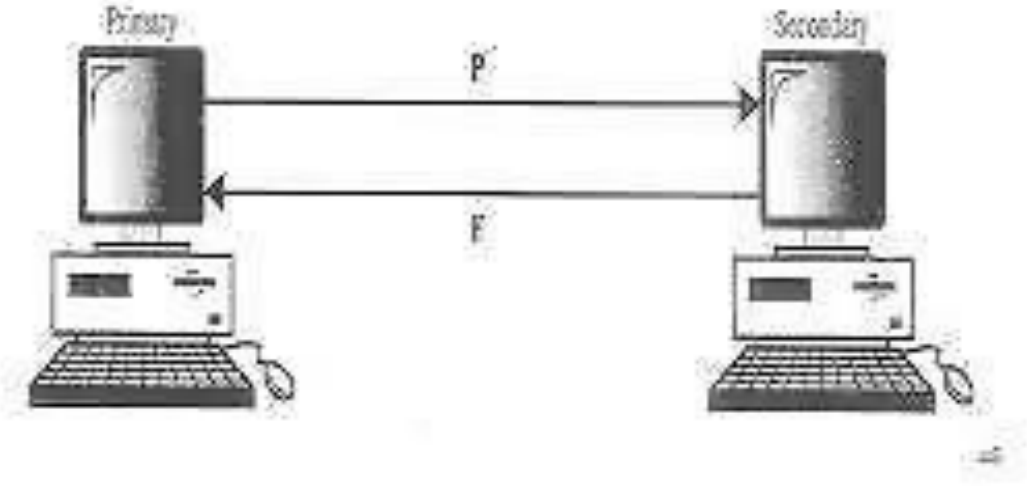
Figure 9.14 HDLC control fields



The control field in the extended mode. Note that in the extended mode, the control field in the I-frame and S-frame is two bytes long to allow seven bits for the sending and receiving sequence number (the sequence number is between 0 and 127). However, the control field in the U-frame is still one byte.

The P/F field is a single bit with a dual purpose. It has meaning only when it is set (bit = 1) and can mean poll or final. It means poll when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver). It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender); see Figure 9.15.

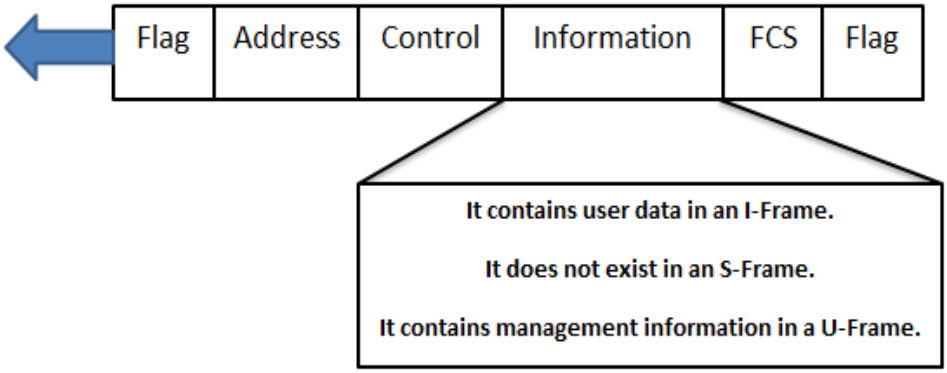
Figure 9.15 Poll/final field in HDLC



Information Field

The information field contains the user's data in an I-frame, and network management information in a U-frame (see Figure 9.16). Its length can vary from one network to another but is always fixed within each network. An S-frame has no information field.

Figure 9.16 Information field in HDLC



As we have seen in the several cases above, it is often possible to include flow, error, and other control information in an I-frame that also contains data. For example, in a two-way exchange of data (either half or full duplex), station 2 can acknowledge receipt of data from station 1 in the control field of its own data frame rather than sending a separate frame just for the acknowledgment. Combining data to be sent with control information this way is called piggybacking.

Piggybacking means combining data to be sent and acknowledgment of the frame received in one single frame.

FCS Field

The frame check sequence (FCS) is HDLC's error detection field. It can contain either a two- or four-byte CRC.

More about Frames

Of the three frames used by HDLC, the I-frame is the most straightforward. I-frames are designed for user information transport and piggybacked acknowledgments and nothing else. For this reason, the range of variation in I-frames is small—all differences relate either to the data (content and CRC), to the identifying number of the frame, or to the acknowledgment of received frames (ACK or NAK). S-frames and U-frames, however, contain subfields within their control fields. As we saw in our discussion of control fields, these subfields carry codes that alter the meaning of the frame. For example, an S-frame coded for selective-reject (SREJ) can-not be used in the same context as an S-frame coded for receive ready (RR). In this section, we will examine the different types of and uses for S- and U-frames.

S-frames

Supervisory frames are used for acknowledgment, flow control, and error control whenever piggybacking that information onto an I-frame is either impossible or inappropriate (when the station either has no data of its own to send or needs to send a command or response other than an acknowledgment). S-frames do not have information fields, yet each one carries messages to the receiving station. These messages are based on the type of the S-frame and the context of the transmission. The type of each S-frame is determined by a two-bit code set into its control field just before the P/F bit. There are four types of S-frames: receive ready (RR), receive not ready (RNR), reject (REJ), and selective-reject (SREJ);

Receive Ready
An S-frame containing the code for RR (00) can be used in four possible ways, each having a different significance.

- **ACK. RR** is used by a receiving station to return a positive acknowledgment of a received I-frame when the receiver has no data of its own to send (no I-frame on which to piggyback the acknowledgment). In this case, the N(R) field of the control frame contains the sequence number of the next frame expected by the receiver. In a one-byte control field, an N(R) field has three bits, allowing up to 8 frames to be acknowledged. In an extended mode control field, an N(R) field has 7 bits, allowing up to 128 frames to be acknowledged.
- **Poll**. When transmitted by the primary (or acting primary in a combined station) with the P/F bit (now functioning as the poll or P bit) set, RR asks the secondary if it has anything to send.
- **Negative response to poll**. When sent by a secondary with the P/F bit (now functioning as the final or F bit) set, RR tells the primary that the secondary has nothing to send. If the secondary does have data to transmit, it responds to the poll with an I-frame, not an S-frame.
- **Positive response to select**. When a secondary is able to receive a transmission from the primary, it returns an RR frame with the P/F (used as the F) bit set to 1. (For a description of selection, see RNR, below.)

Receive Not Ready RNR frames can be used in three different ways:

- **ACK. RNR** returned by a receiver to a sending station acknowledges receipt of all frames up to, but not including, the one indicated in the N(R) field but requests that no more frames be sent until an RR frame is issued.
- **Select.** When a primary wishes to transmit data to a specific secondary, it alerts the secondary by sending an RNR frame with the P/F (used as the P) bit set. The RNR code tells the secondary not to send data of its own, that the frame is a select and not a poll.
- **Negative response to select**. When a selected secondary is unable to receive data, it returns an RNR frame with the P/F (used as the F) bit set.

Reject A third type of S-frame is reject (REJ). REJ is the negative acknowledgment returned by a receiver in a go-back-n ARQ error correction system when the receiver has no data on which to piggyback the response. In an REJ frame, the N(R) field contains the number of the damaged frame to indicate that the frame and all that follow it need to be retransmitted.

Selective-Reject A selective-reject (SREJ) frame is a negative acknowledgment in a selective reject ARQ system. It is sent by the receiver to the sender to indicate that a specific frame (the number in the N(R) field) has been received damaged and must be resent

U-frames
Unnumbered frames are used to exchange session management and control information between connected devices. Unlike S-frames, U-frames contain an information field, but one used for system management information not user data. As with S-frames, however, much of the information carried by U-frames is contained in codes included in the control field. U-frame codes are divided into two sections: a two-bit prefix before the P/F bit and a three-bit suffix after the P/F bit. Together, these two segments (five bits) can be used to create up to 32 different types of U-frames.

The U-frame commands and responses listed in Table 9-2 can be divided into five basic functional categories: mode setting, unnumbered-exchange, disconnection, initialization, and miscellaneous.

Mode Setting, Mode-setting commands are sent by the primary station, or by a combined station wishing to control an exchange, to establish the mode of the session. A mode-setting U-frame tells the receiving station what format the transmission will take. For example, if a combined station wishes to establish a temporary primary-to-secondary relationship with another station, it sends a U-frame containing the code 00 001 (for set, normal response mode). The addressed station understands that it is being selected to receive a transmission (as if from a primary) and adjusts itself accordingly.

Unnumbered-exchange codes are used to send or solicit specific pieces of data link information between devices. The unnumbered poll (UP) code (00 100) is transmitted by the primary station on a link (or the combined station acting as a primary) to establish the send/receive status of the addressed station in an unnumbered exchange. The unnumbered information (UI) code (00 000) is used for the transmission of specific pieces of information such as time/date for synchronization. UI frames can be sent either as commands (e.g., a list of parameters for the coming transmission) or as responses (e.g., a description of the

Table 9-2 U frame control command and response

Command/Response	Meaning
SNRM	Set normal response mode
SNRME	Set Normal Response mode (Extended)
SARM	Set Asynchronous response mode
SARME	Set Asynchronous response Mode (Extended)
SABM	Set Asynchronous Balanced Mode
SABME	Set Asynchronous Balanced Mode (Extended)
UP	Unnumbered Poll
UI	Unnumbered Information
UA	Unnumbered Acknowledgement
RD	Request Disconnect
DISC	Disconnect
DM	Disconnect Mode
RIM	Request Information Mode
SIM	Set Initialization Mode
RSET	Reset
XID	Exchange ID
FRMR	Frame Reject

capabilities of the addressed station to receive data). The unnumbered acknowledgment (UA) code (00 110) is returned by the addressed station in answer to an unnumbered poll, to acknowledge one of the unnumbered request frames.

Disconnection There are three disconnect codes, one a command from the acting primary or combined station, the other two responses from the receiving station. The first of these, disconnect (DISC, 00 010), is sent by the first station to the second to terminate the connection. The second, request disconnect (RD, 00 010) is a request by the second station to the first that a DISC-be issued. The third, disconnect mode (DM, 11 000), is transmitted by

the addressed station to the initiating station as a negative response to a mode-setting command

Initialization Mode The code 10 000, used as a command (first system to second system), means set initialization mode (SIM). SIM prepares the addressed station to initialize its data link control functions. The SIM command is then followed by UI frames containing, for example, a new program or a new parameter set. The same code, 10 000, used as a response (second system to first system), means request initialization mode (RIM) and solicits a SIM command from the first station. It is used to respond to a modesetting command when the second station cannot act upon the command with-out first receiving a SIM (see. Table 9-2).

Miscellaneous Of the final three commands, the first two reset (RSET, 11 001) and exchange ID (XID, 11 101) are commands from the initiating system to the addressed system. The third, frame reject (FRMR, 10 001) is a response sent from the addressed system to the initiating system.

RSET tells the second station the first station is resetting its send sequence numbering and instructs the second system to do likewise. It is usually issued in response to an FRMR.

XID requests an exchange of identifying data from the second station

FRMR tells the first system that a U-frame received by the second system contains a syntax error (This doesn't look like an HDLC frame!). It is returned by the addressed system when, for example frame is identified as an S-frame but contains an information field (see Table 9-2).

10. SWITCHING

SWITCHING

A switched network consists of a series of inter-linked nodes, called switches. Switches are hardware and/or software devices capable of creating temporary connections between two or more devices linked to the switch but not to each other. In a switched network, some of these nodes are connected to the communicating devices. Others are used only for routing.

Each switch is connected to multiple links and is used to complete the connections between them, two at a time. Traditionally, three methods of switching have been important: circuit switching, packet switching, and message switching. The first two are commonly used today. The third has been phased out in general communications but still has net-working applications. New switching strategies are gaining prominence, among them cell relay (ATM) and Frame Relay.

10.1 CIRCUIT SWITCHING

Circuit switching creates a direct physical connection between two devices such as phones or computers. For example, in Figure 10.1, instead of point-to-point connections between the three computers on the left (A, B, and C) to the four computers on the right (D, E, F, and G), requiring 12 links, we can use four switches to reduce the number and the total length of the links. In Figure 10.1, computer A is connected through switches I, IL and III to computer D. By moving the levers of the switches, any computer on the left can be connected to any computer on the right.

A circuit switch is a device with n inputs and m outputs that creates a temporary connection between an input link and an output link (see Figure 10.2). The number of inputs does not have to match the number of outputs.

An n-by-n folded switch can connect n lines in full-duplex mode. For example, it can connect n telephones in such a way that each phone can be connected to every other phone (see Figure 10.3).

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

Figure 10.1 Circuit switched network

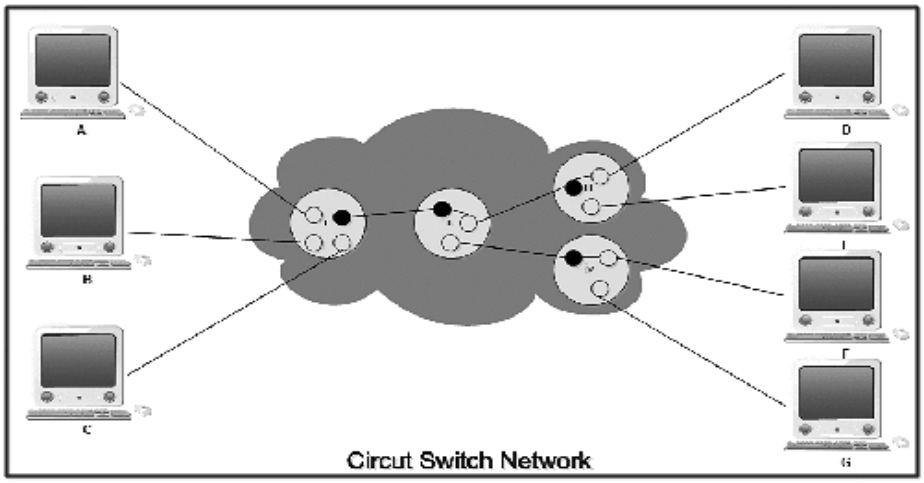


Figure 10.2 A circuit switch

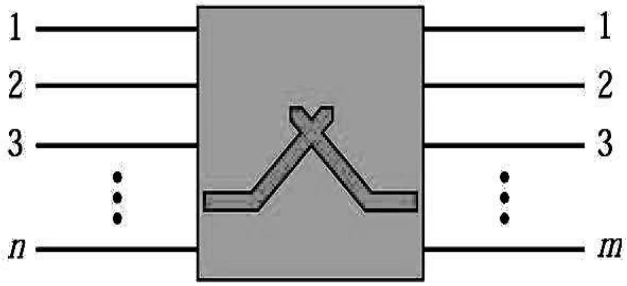
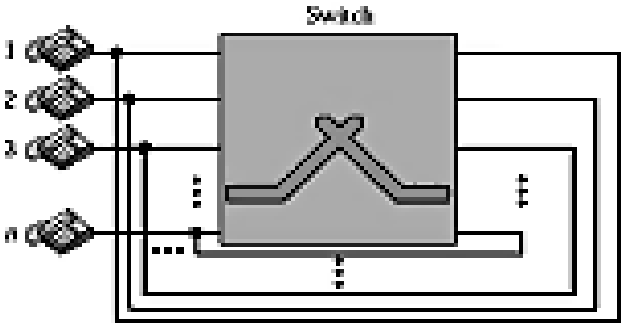


Figure 10.3 A folded switch



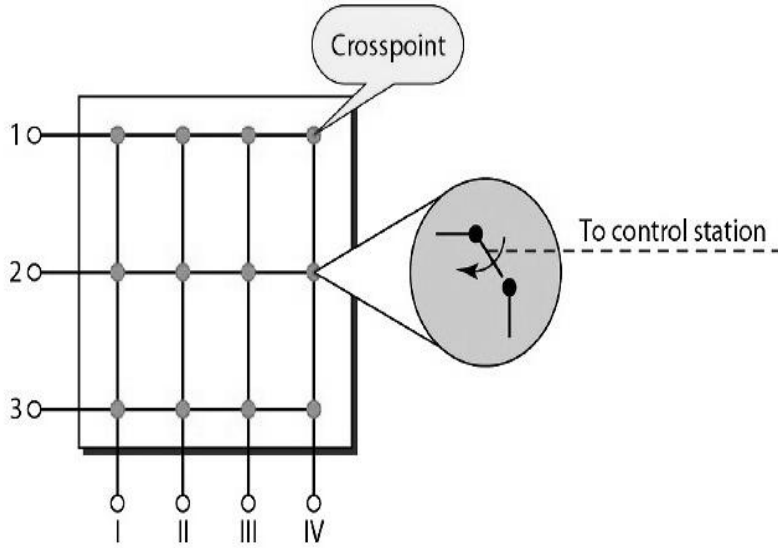
Space-Division Switches

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

Crossbar Switches

A crossbar switch connects n inputs to m outputs in a grid, using electronic micro-switches (transistors) at each cross point (see Figure 10.4). The major limitation of this design is the number of cross points required. Connecting n inputs to m outputs using a crossbar switch requires n x m cross points. For example, to connect 1000 inputs to 1000 outputs requires a crossbar with 1,000,000 cross points. This factor makes the crossbar impractical because it makes the size of the crossbar huge. Such a switch is also inefficient because statistics show that, in practice, fewer than 25 percent of the cross points are in use at a given time. The rest are idle.

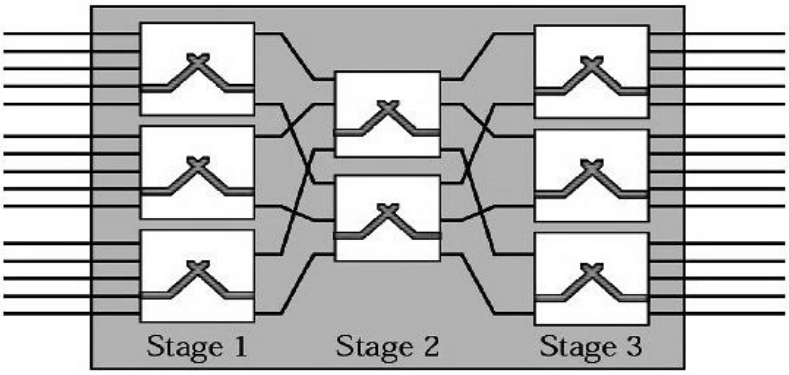
Figure 10.4 Crossbar Switch



Multistage Switches

The solution to the limitations of the crossbar switch is to use multistage switches, which combine crossbar switches in several stages. In multistage switching, devices are linked to switches that, in turn, are linked to a hierarchy of other switches (see Figure 10.5).

Figure 10.5 Multistage switch



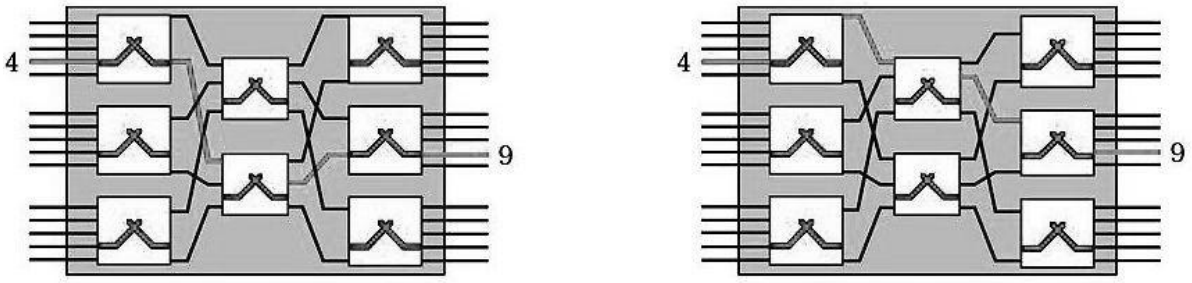
The design of a multistage switch depends on the number of stages and the number of switches required (or desired) in each stage. Normally, the middle stages have fewer switches than do the first and last stages. For example, imagine that we want a multi-stage switch as in Figure 14.8 to do the job of a single 15-by-15 crossbar switch. Assume that we have decided on a three-stage design that uses three switches in the first and final stages and two switches in the middle stage. Because there are three of them, each of the first-stage switches has inputs from one-third of the input devices, giving them five inputs each (5 x 3 = 15).

Next, each of the first-stage switches must have an output to each of the intermediate switches. There are two intermediate switches; therefore, each first-stage switch has two

outputs. Each third-stage switch must have inputs from each of the intermediate switches; two intermediate switches means two inputs. The intermediate switches must connect to all three first-stage switches and all three last-stage switches, and so must have three inputs and three outputs each.

Multiple Paths Multistage switches provide several options for connecting each pair of linked devices. Figure 10.6 shows two ways traffic can move from an input to an out-put using the switch designed in the example above.

Figure 10.6 Switching path



a. First option b. Second option

In Figure 10.6 a, a pathway is established between input line 4 and output line 9. In this instance, the path uses the lower intermediate switch and that switch's center output line to reach the last-stage switch connected to line 9.

Figure 10.6 b shows a pathway between the same input line 4 and the same output line 9 using the upper intermediate switch.

Let us compare the number of cross points in a 15-by-15 single-stage crossbar switch with the 15-by-15 multistage switch that we described above. In the single-stage switch, we need 225 cross points (15 x 15). In the multistage switch, we need

- Three first-stage switches, each with 10 cross points (5 x 2), for a total of 30 cross-points at the first stage.
- Two second-stage switches, each with 9 cross points (3 x 3), for a total of 18 cross-points at the second stage.
- Three third-stage switches, each with 10 cross points (5 x 2), for a total of 30 cross points at the last stage.

The total number of cross points required by our multistage switch is 78. In this example, the multistage switch requires only 35 percent as many cross points as the single-stage switch.

Blocking

This savings comes with a cost, however. The reduction in the number of cross points results in a phenomenon called blocking during periods of heavy traffic. Blocking refers to times when one input cannot be connected to an output because there is no path available between them all of the possible intermediate switches are occupied.

In a single-stage switch, blocking does not occur. Because every combination of input and output has its own cross point, there is always a path. (Cases where two inputs are trying to contact the same output don't count. That path is not blocked; the output is merely busy.)

In the multistage switch described in the example above, however, only two of the first five inputs can use the switch at a time, only two of the second five inputs can use the switch at a time, and so on. The small number of outputs at the middle stage further increases the restriction on the number of available links.

In large systems, such as those having 10,000 inputs and outputs, the number of stages can be increased to cut down the number of cross points required. As the number of stages increases, however, possible blocking increases as well. Many people have experienced blocking on public telephone systems in the wake of a natural disaster when calls being made to check on or reassure relatives far outnumber the ordinary load of the system. In those cases, it is often impossible to get a connection. Under nor-mal circumstances, however, blocking is not usually a problem. In countries that can afford it, the number of switches between lines is calculated to blocking unlikely. The formula for finding this number is based on statistical analysis, which is beyond the scope of this book.

Time-Division Switches

Time-division switching uses time-division multiplexing to achieve switching. There are two popular methods used in time-division multiplexing: the time-slot interchange and the TDM bus.

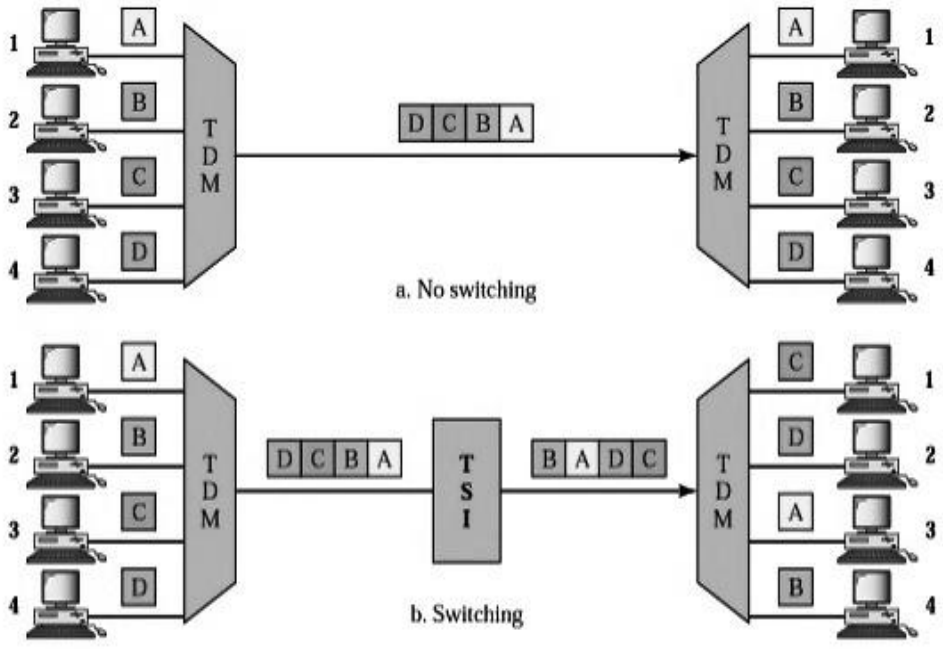
Time-Slot Interchange (TSI)

Figure 10.7 shows a system connecting four input lines to four output lines. Imagine that each input line wants to send data to an output line according to the following pattern:

Figure 10.7a shows the results of ordinary time-division multiplexing. As you can see, the desired task is not accomplished. Data are output in the same order as they are input. Data from 1 go to 1, from 2 go to 2, from 3 go to 3, and from 4 go to 4.

In Figure 10.7b, however, we insert a device called a time-slot interchange (TSI) into the link. A TSI changes the ordering of the slots based on the desired connections. In this case, it changes the order of data from A, B, C, D to C. D. A, B. Now, when the demultiplexer separates the slots, it passes them to the proper outputs.

Figure 10.7 Time division multiplexing without and with a time slot interchange

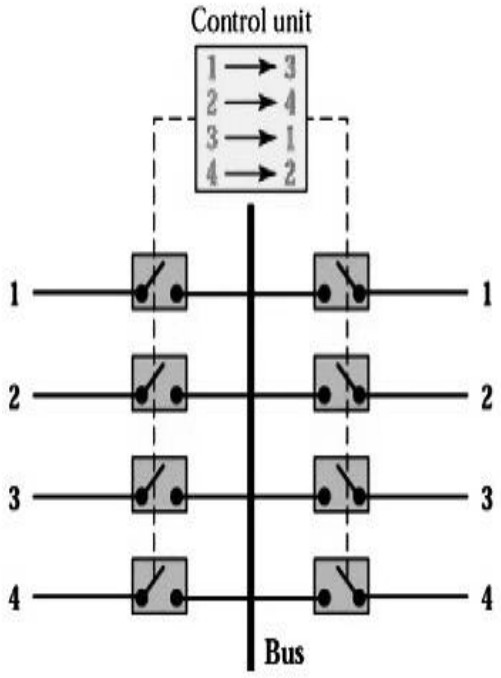


A TSI consists of random access memory (RAM) with several memory locations. The size of each location is the same as the size of a single time slot. The number of locations is the same as the number of inputs (in most cases, the number of inputs and outputs are equal). The RAM fills up with incoming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

TDM Bus

Figure 10.8 shows a very simplified version of a TDM bus. The input and output lines are connected to a high-speed bus through input and output gates (micro switches). Each input gate is closed during one of the four time slots. During the same time slot, only one output gate is also closed. This pair of gates allows a burst of data to be transferred from one specific input line to one specific output line using the bus. The control unit opens and closes the gates according to switching need. For example, in the figure, at the first time slot the input gate 1 and output gate 3 will be closed; during the second time slot, input gate 2 and output gate 4 will be closed; and so on.

Figure 10.8 TDM bus



A folded TDM bus can be made with duplex lines (input and output) and dual gates.

Space- and Time-Division Switching Combinations

When we compare space-division and time-division switching, some interesting facts emerge. The advantage of space-division switching is that it is instantaneous. Its disadvantage is the

number of cross points required to make space-division switching accept-able in terms of blocking.

The advantage of time-division switching is that it needs no cross points. Its disadvantage, in the case of TSI, is that processing each connection creates delays. Each time slot must be stored by the RAM, then retrieved and passed on.

In a third option, we combine space-division and time-division technology to take advantage of the best of both. Combining the two results in switches that are optimized both physically (the number of cross points) and temporally (the amount of delay). Multistage switches of this sort can be designed as time-space-time (TST), time-space--space-time (TSST), space-time-time-space (SITS), or other possible combinations.

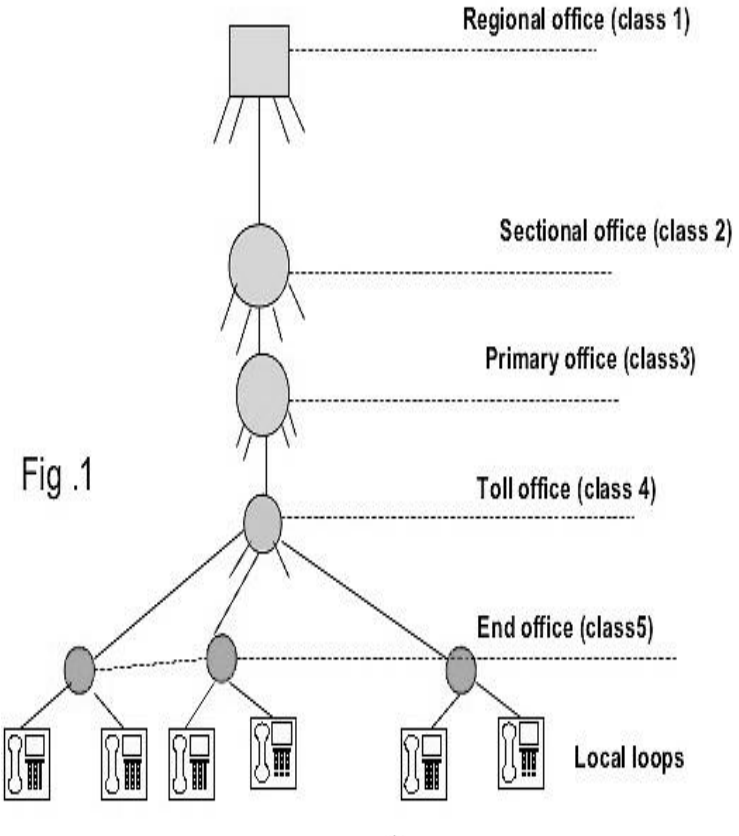
A simple TST switch that consists of two time stages and one space stage and has 12 inputs and 12 outputs. Instead of one time-division switch, it divides the inputs into three groups (of four inputs each) and directs them to three time-slot interchanges. The result in this case is that the average delay is one-third of that which would result from using one time-slot interchange to handle all 12 inputs.

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

Public Switched Telephone Network (PSTN)

An example of a circuit-switched telephone network is the Public Switched Telephone Network (PSTN) in North America. The switching centers are organized into five classes: regional offices (class 1), sectional offices (class 2), primary offices (class 3), toll offices (class 4), and end offices (class 5). Figure 10.9 shows the hierarchical relationship between these offices.

Figure 10.9 PSTN Hierarchy



Subscriber telephones are connected, through local loops, to end offices (or central offices). A small town may have only one end office, but a large city will have several end offices. Many end offices are connected to one toll office. Several toll offices are connected to a primary office. Several primary offices are connected to a sectional office, which normally serves more than one state. And finally several sectional offices are connected to one regional office. All the regional offices are connected using mesh topology.

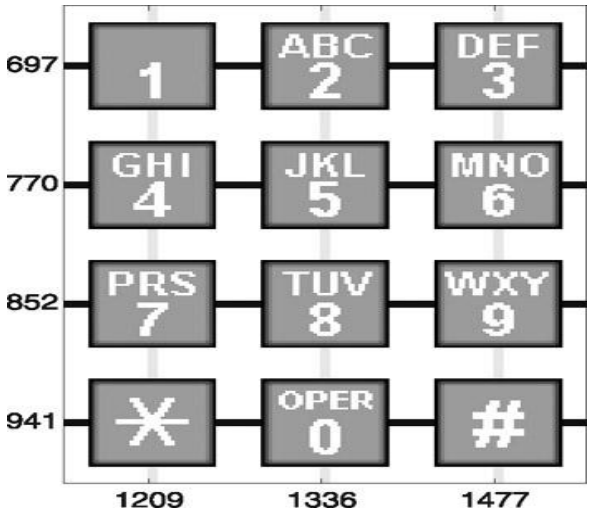
Accessing the switching station at the end offices is accomplished through dialing. In the past, telephones featured rotary or pulse dialing, in which a digital signal was sent to the end office for each number dialed. This type of dialing was prone to errors due to the inconsistency of humans during the dialing process.

Today, dialing is accomplished through the Touch-Tone technique. In this method, instead of sending a digital signal, the user sends two small bursts of analog signals, called dual tone. The frequency of the signals sent depends on the row and column of the pressed pad)

12-pad Touch-Tone dialing system. Note that there is also a variation with an extra column (16-pad Touch-Tone), which is used for special purposes.

In Figure 10.10, when a user dials, for example, the number 8, two bursts of analog signals with frequencies 852 and 1336 Hz are sent to the end office.

Figure 10.10 Touch-Tone dialing



10.2 PACKET SWITCHING

Circuit switching was designed for voice communication. In a telephone conversation, for example, once a circuit is established, it remains connected for the duration of the session. Circuit switching creates temporary (dialed) or permanent (leased) dedicated links that are well suited to this type of communication.

Circuit switching is less well suited to data and other non-voice transmissions. Non-voice transmissions tend to be bursty, meaning that data come in spurts with idle gaps between them. When circuit-switched links are used for data transmission, therefore, the line is often idle and its facilities wasted.

A second weakness of circuit-switched connections for data transmission is in its data rate. A circuit-switched link creates the equivalent of a single cable between two devices and thereby assumes a single data rate for both devices. This assumption limits the flexibility and usefulness of a circuit-switched connection for networks interconnecting a variety of digital devices.

Third, circuit switching is inflexible. Once a circuit has been established, that circuit is the path taken by all parts of the transmission whether or not it remains the most efficient or available.

Finally, circuit switching sees all transmissions as equal. Any request is granted to whatever link is available. But often with data transmission, we want to be able to prioritize: to say, for example, that transmission x can go anytime but transmission z is time dependent and must go immediately.

A better solution for data transmission is **packet switching**. In a **packet-switched network**, data are transmitted in discrete units of potentially variable length blocks called packets. The maximum length of the packet is established by the network. Longer transmissions are broken up into multiple packets. Each packet contains not only data but also a header with control information (such as priority codes and source and destination addresses). The packets are sent over the network node to node. At each node, the packet is stored briefly then routed according to the information in its header.

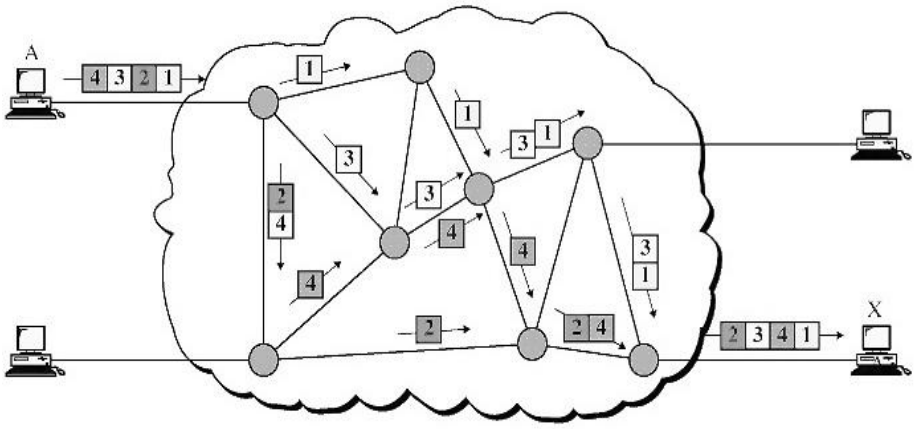
There are two popular approaches to packet switching: datagram and virtual circuit (

Datagram Approach

In the datagram approach to packet switching, each packet is treated-independently from all others. Even when one packet represents just a piece of a multipacket transmission, the network (and network layer functions) treats it as though it existed alone. Packets in-this technology are referred to as **datagrams**.

Figure 10.11 shows how the datagram approach can be used to deliver four packets from station A to station X. In this example, all four packets (or datagrams) belong to the same message, but may go by different paths to reach their destination.

Figure 10.11 Datagram approach

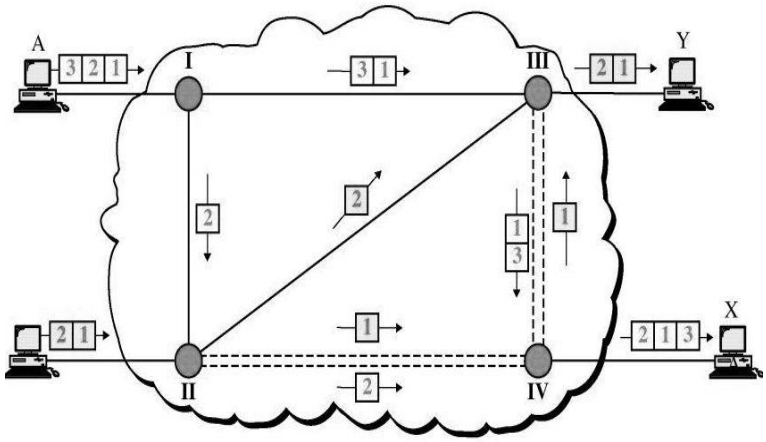


This approach can cause the datagrams of a transmission to arrive at their destination out of order. It is the responsibility of the transport layer in most protocols to reorder the datagrams before passing them on to the destination port.

The link joining each pair of nodes can contain multiple channels. Each of these channels is capable, in turn, of carrying datagrams either from several different sources or from one source. Multiplexing can be done using TDM or FDM (see Figure 14.18).

In Figure 10.12, devices A and B are sending datagrams to devices X and Y. Some paths use one channel while others use more than one. As you can see, the bottom link is carrying two packets from different sources in the same direction. The link on the right, however, is carrying datagrams in two directions

Figure 10.12 Multiple channels in datagram approach



Virtual Circuit Approach

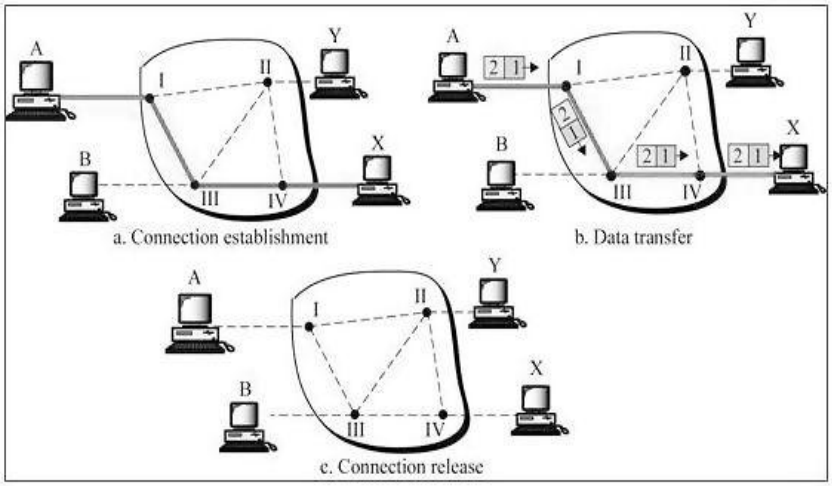
In the virtual circuit approach to packet switching, the relationship between all packets belonging to a message or session is preserved. A single route is chosen between sender and receiver at the beginning of the session. When the data are sent, all packets of the transmission travel one after another along that route.

Today, virtual circuit transmission is implemented in two formats: switched virtual circuit (SVC) and permanent virtual circuit (PVC).

SVC

The switched virtual circuit (SVC) format is comparable conceptually to dial-up lines in circuit switching. In this method, a virtual circuit is created whenever it is needed and exists only for the duration of the specific exchange. For example, imagine that station A wants to send four packets to station X. First, A requests the establishment of a connection to X. Once the connection is in place, the packets are sent one after another and in sequential order. When the last packet has been received and, if necessary, acknowledged, the connection is released and that virtual circuit ceases to exist (see Figure 10.13). Only one single route exists for the duration of transmission, although the network could pick an alternate route in response to failure or congestion.

Figure 10.13 Switched virtual circuit (SVC)

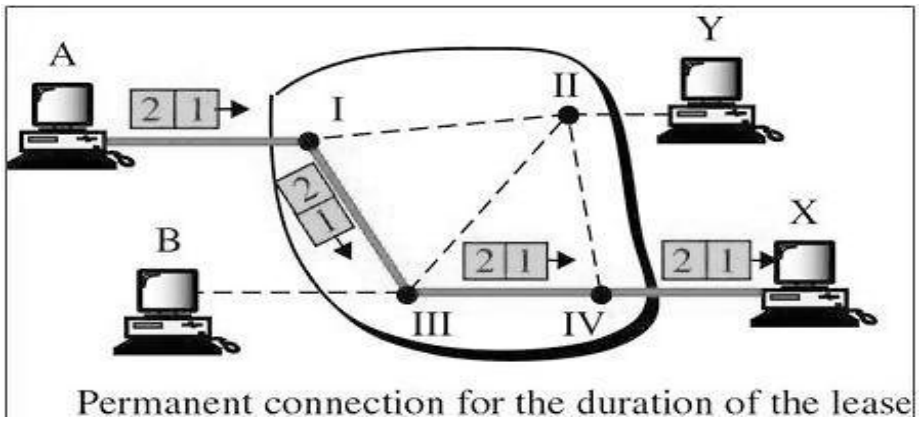


Each time that A wishes to communicate with X, a new route is established. The route may be the same each time, or it may differ in response to varying network conditions.

PVC

Permanent virtual circuits (PVC) are comparable to leased lines in circuit switching. In this method, the same virtual circuit is provided between two users on a continuous basis. The circuit is dedicated to the specific users. No one else can use it and, because it is always in place, it can be used without connection establishment and connection termination. Whereas two SVC users may get a different route every time they request a connection, two PVC users always get the same route (see Figure 10.14).

Figure 10.14 Permanent virtual circuit



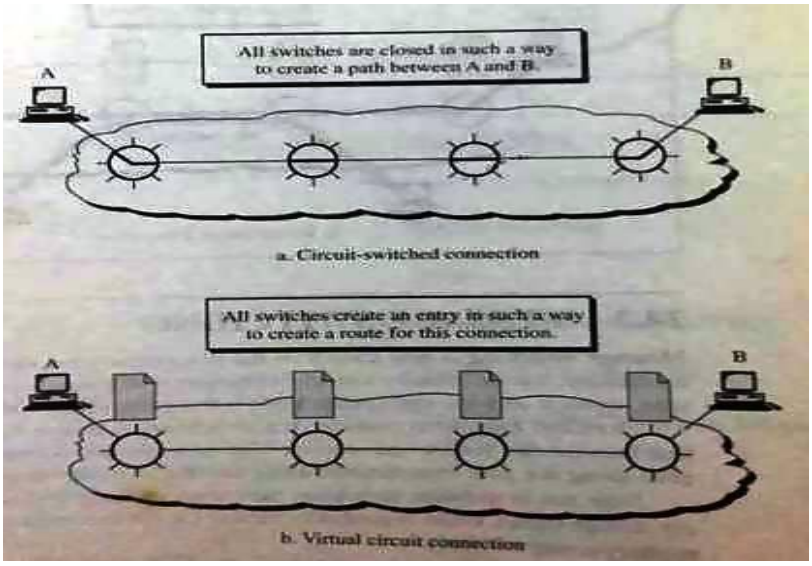
Circuit-Switched Connection versus Virtual-Circuit Connection

Although it seems that a circuit-switched connection and a virtual-circuit connection are the same, there are differences:

- Path versus route. A circuit-switched connection creates a path between two points. The physical path is created by setting the switches for the duration of the dial (dial-up line) or the duration of the lease (leased line). A virtual circuit connection creates a route between two points. This means each switch creates an entry in its routing table (see Chapter 21)

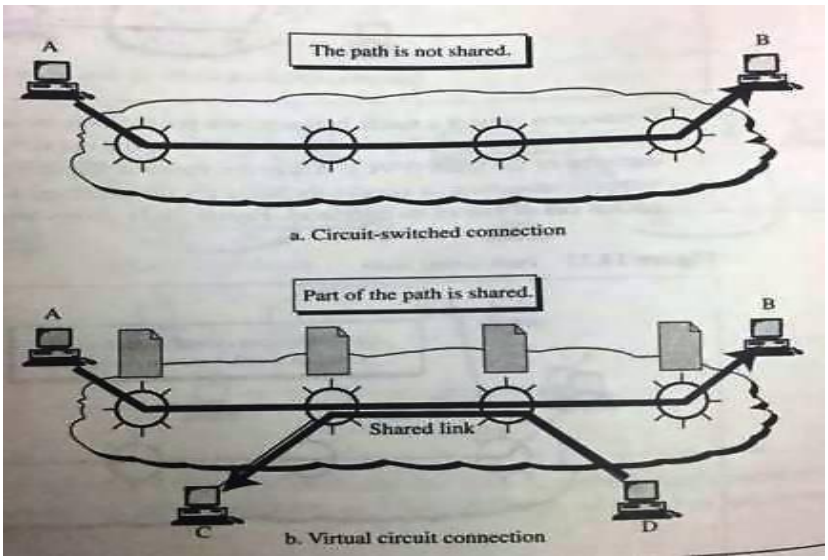
for the duration of the session (SVC) or duration of the lease (PVC). Whenever, the switch receives a packet belonging to a virtual connection, it checks the table for the corresponding entry and routes the packet out of one of its interfaces. Figure 10.15 shows this difference.

Figure 10.15 Path versus route



- **Dedicated versus sharing**. In a circuit-switched connection, the links that make a path are dedicated; they cannot be used by other connections. In a virtual circuit connection, the links that makeroute can be shared by other connections. Figure 10.16 shows this difference

Figure 10.16 Dedicated versus shared



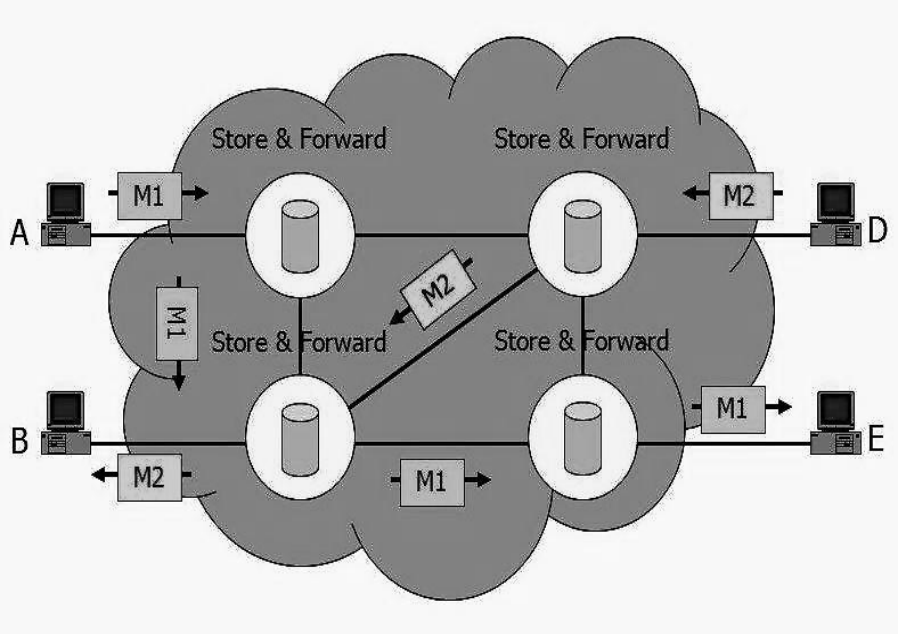
10.3 MESSAGE SWITCHING

Message switching is best known by the descriptive term store and forward. In this mechanism, a node (usually a special computer with a number of disks) receives a message, stores it until the appropriate route is free, then sends it along.

Store and forward is considered a switching technique because there is no direct link between the sender and receiver of a transmission. A message is delivered to the node along one path then rerouted along another to its destination.

Note that in message switching, the messages are stored and relayed from secondary storage (disk), while in packet switching the packets are stored and forwarded from primary storage (RAM).

Figure 10.17 Message Switching



Message switching was common in the 1960s and 1970s. The primary uses have been to provide high-level network services (e.g., delayed delivery, broadcast) forunintelligent devices. Since such devices have been replaced, this type of switch has virtually disappeared. Also, the delays inherent in the process, as well as the requirements for large capacity storage media at each node, make it unpopular for direct communication.

11. LOCAL AREA NETWORKS

A local area network is a data communication system that allows a number of independent devices to communicate directly with each other in a limited geographic area.

11.1 PROJECT 802

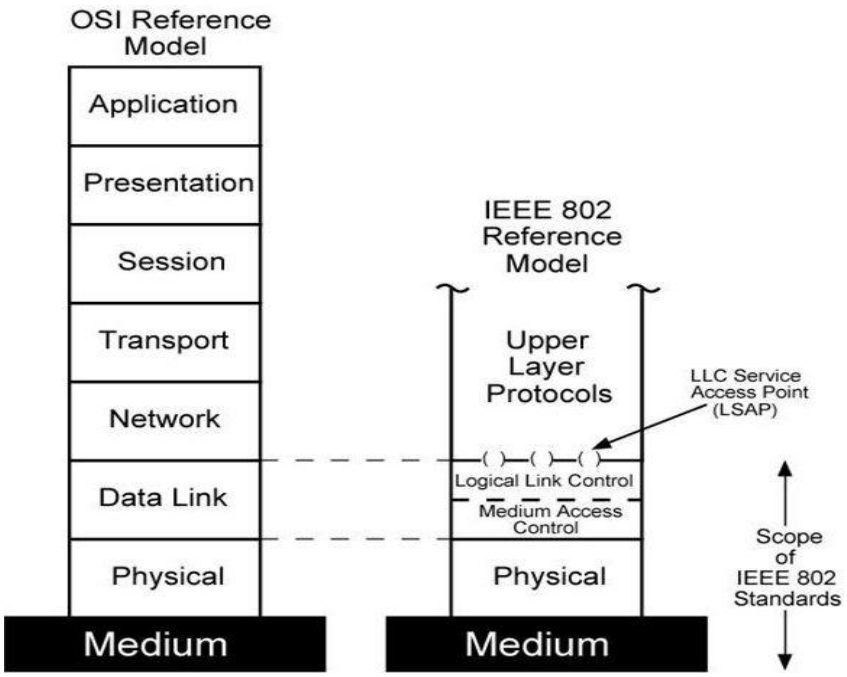
In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication between equipment from a variety of manufacturers. Project 802 does not seek to replace any part of the OSI model. Instead, it is a way of specifying functions of the physical layer, the data link layer, and, to a lesser extent, the network layer to allow for interconnectivity of major LAN protocols.

In 1985, the Computer Society of the IEEE developed Project 802. It covers the first two layers of the OSI model and part of the third level.

The relationship of IEEE Project 802 to the OSI model is shown in Figure 11.1. The IEEE has subdivided the data link layer into two sublayers **logical link control (LLC)** and **medium access control (MAC)**.

The LLC is non-architecture specific; that is, it is the same for all IEEE-defined LANs. The MAC sublayer, on the other hand, contains a number of distinct modules; each carries proprietary information specific to the LAN product being used.

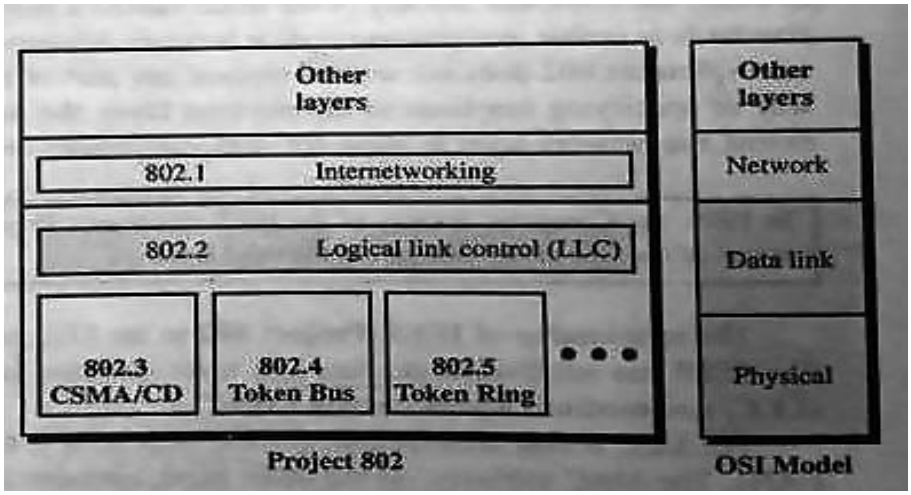
Figure 11.1 LAN compared with the OSI model



Project 802 has split the data link layer into two different sublayers: logical link control (LLC) and media access control (MAC).

In addition to the two sublayers, Project 802 contains a section governing internetworking. This section assures the compatibility of different LANs and MANs across protocols and allows data to be exchanged across otherwise incompatible networks. The strength of Project 802 is modularity. By subdividing the functions necessary for LAN management, the designers were able to standardize those that can be generalized and to isolate those that must remain specific. Each subdivision is identified by a number 802.1 (internetworking); 802.2 (LLC); and the MAC modules 802.3 (CSMA/ CD), 802.4 (Token Bus), 802.5 (Token Ring), and others (see Figure 11.2).
Model.

Figure 11.2 Project 802



IEEE 802.1

IEEE 802.1 is the section of Project 802 devoted to internetworking issues in LANs and MANs. Although not yet complete, it seeks to resolve the, incompatibilitiesbetween network architectures without requiring modifications in existing addressing, access, and error recovery mechanisms, among others.

IIEEE 802.1 is an internetworking standard for LANs.

LLC

In general, the IEEE Project 802 model takes the structure of an HDLC frame and divides it into two sets of functions. One set contains the end-user portions of the frame: the logical addresses, control information, and data. These functions are handled by the IEEE 802.2 logical link control (LLC) protocol. LLC is considered the upper layer of the IEEE .802 data link layer and is common to all LAN protocols.

IEEE 802.2 logical link control (LLC) is the upper sublayer of the data link layer.

MAC

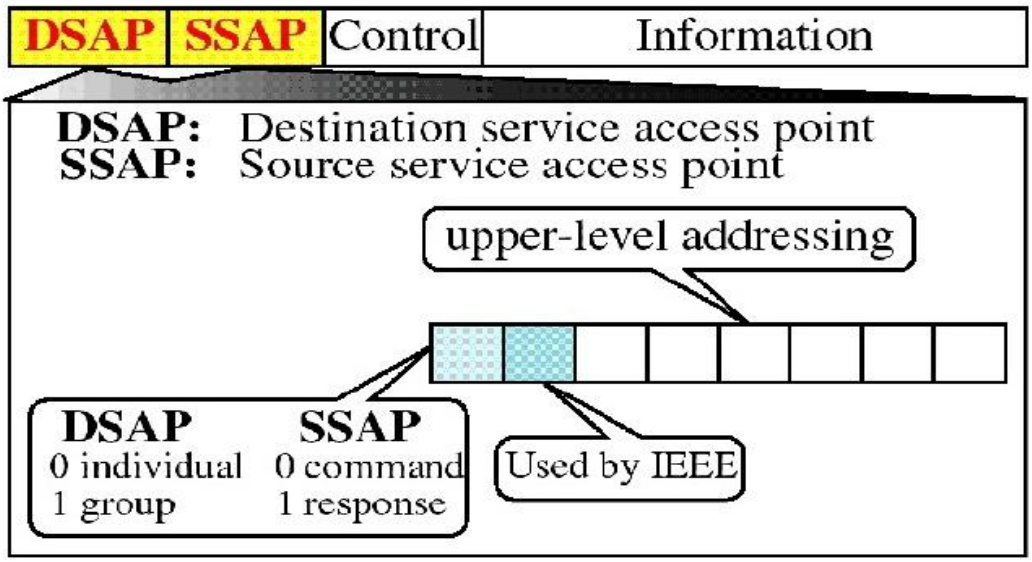
The second set of functions, the medium access control (MAC) sublayer, resolves the contention for the shared media. It contains the synchronization, flag, flow, and error control specifications necessary to move information from one place to another, as well as the physical address of the next station to receive and route a packet. MAC protocols are specific to the LAN using them (Ethernet, Token Ring, and Token Bus, etc.).

Media access control (MAC) is the lower sublayer of the data link layer.

Protocol Data Unit (PDU)

The data unit in the LLC level is called the protocol data unit (PDU), The PDU contains four fields familiar from HDLC: a destination service access point (DSAP), a source ser-vice access point (SSAP), a control field, and an information field (see Figure 11.3).

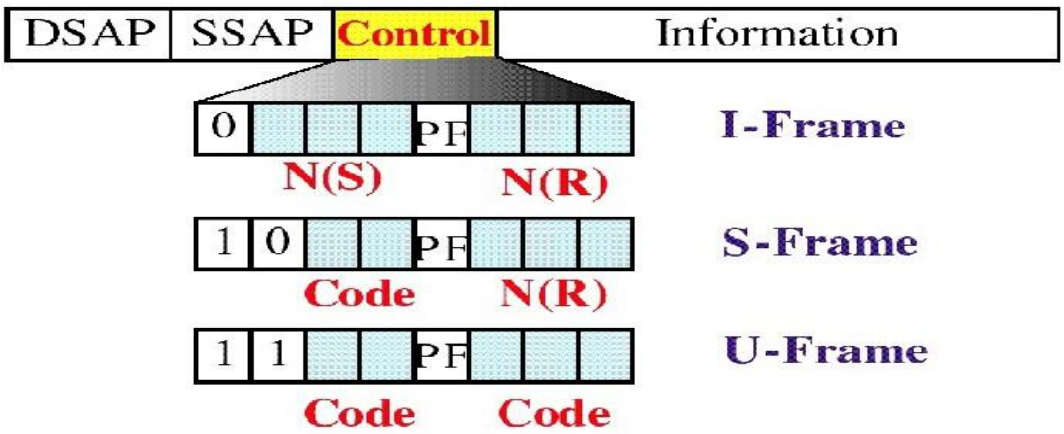
Figure 11.3 PDU format



DSAP and SSAP

The DSAP and SSAP are addresses used by the LLC to identify the protocol stacks on the receiving and sending machines that are generating and using the data. The first bit of the DSAP indicates whether the frame is intended for an individual or a group. The first bit of the SSAP indicates whether the communication is a command or response PDU (see Figure 11.3).

Figure 11.4 Control field in a PDU



Control

The control field of the PDU is identical to the control field in HDLC. As in HDLC. PDU frames can be I-frames, S-frames, or U-frames and carry all of the codes and information that the corresponding HDLC frames carry (see Figure 11.4).

11.2 ETHERNET

I E 802.3 supports a LAN standard originally developed by Xerox and later extended by a joint venture between Digital Equipment Corporation, Intel Corporation, and Xerox. This was called **Etherne**t.

IEEE 802.3 defines two categories: **baseband** and **broadband**, as shown in Fig 11.5. The word base specifies a digital signal (in this case, Manchester encoding). The word broad specifies an analog signal (in this case, PSK encoding). IEEE divides the baseband category into five different standards: **10Base5, 10Base2, 10Base-T, 1Base5, and 100Base-T**. The first number (10, 1, or 100) indicates the data rate in Mbps.

Figure 11.5 IEEE 802.3

The last number or letter (5, 2, 1, or T) indicates the maximum cable length or the type of cable. IEEE defines only one specification for the broadband category:

Access Method: CSMA/CD

Whenever multiple users have unregulated access to a single line, there is a danger of signals overlapping and destroying each other. Such overlaps, which turn the signals into unusable noise, are called collisions. As traffic increases on a multiple-access link, so do collisions. A LAN therefore needs a mechanism to coordinate traffic, minimize the number of collisions that occur, and maximize the number of frames that are delivered successfully. The access mechanism used in an Ethernet is called **carrier sense multiple access with collision detection (CSMA/CD**, standardized in IEEE 802.3).

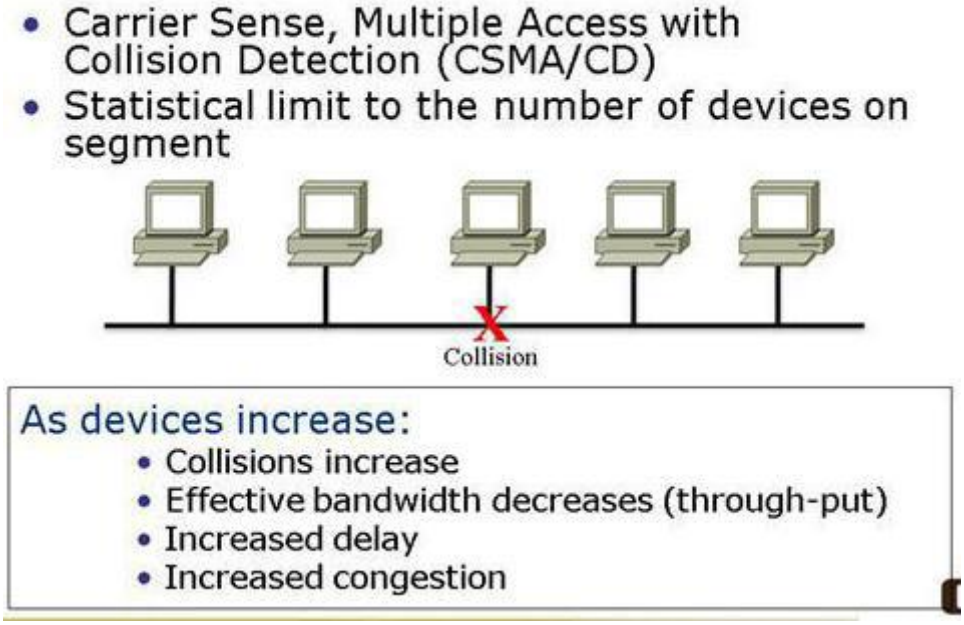
CSMA/CD is the result of an evolution from **multiple access** (MA) to carrier sense multiple access (CSMA), and, finally, to carrier sense multiple access with collision detection (CSMA/CD). The original design was a multiple access method in which every workstation had equal access to a link. In MA, there was no provision for traffic coordination. Any station wishing to transmit did so, then relied on acknowledgments to verify that the transmitted frame had not been destroyed by other traffic on the line.

In a CSMA system, any workstation wishing to transmit must first listen for existing traffic on the line. A device listens by checking for a voltage. If no voltage is detected, the line is considered idle and the transmission is initiated. CSMA cuts down on the number of

collisions but does not eliminate them. Collisions can still occur. If another station has transmitted too recently for its signal to have reached the listening station, the listener assumes the line is idle and introduces its own signal onto the line.

The final step is the addition of collision detection (CD). In CSMA/CD the station wishing to transmit first listens to make certain the link is free, then transmits its data, then listens again. During the data transmission, the station checks the line for the extremely high voltages that indicate a collision. If a collision is detected, the station quits the current transmission and waits a predetermined amount of time for the line to clear, then sends its data again (see Figure 11.6).

Figure 11.6 Collision in CSMA/CD



Addressing

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC usually fits inside the station and provides the station with a six-byte physical address. The number on the NIC is unique.

Electrical Specification

Signaling

The baseband systems use Manchester digital encoding (see Chapter 5). There is one broadband system, 10Broad36. It uses digital/analog conversion (differential PSK).

Data Rate

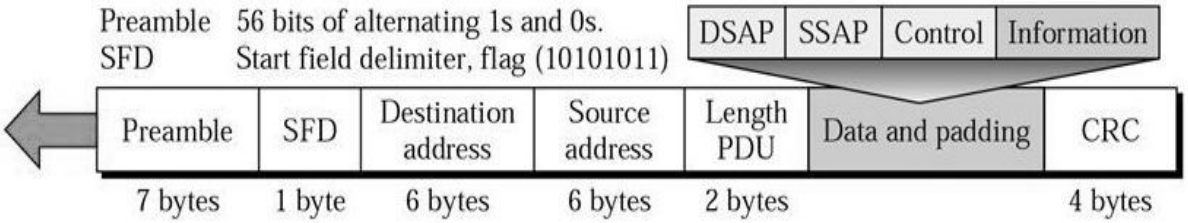
Ethernet LANs can support data rates between 1 and 100 Mbps.

Frame Format

IEEE 802.3 specifies one type of frame containing seven fields: preamble, SFD, DA, SA, length/type of PDU, 802.2 frame, and the CRC. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium.

Acknowledgments must be implemented at the higher layers. The format of the MAC frame in CSMA/CD is shown in Figure 11.7.

Figure 11.7 802.3 MAC frame



- **Preamble.** The first field of the 802.3 frame, the preamble, contains seven bytes (56 bits) of alternating 0s and 1 s that alert the receiving system to the coming frame and enable it to synchronize its input timing. The pattern 1010101 provides only an alert and a timing pulse; it can be too easily aliased to be useful in indicating the beginning of the data stream. HDLC combined the alert, timing, and start synchronization into a single field: the flag. IEEE 802.3 divides these three functions between the preamble and the second field, the start frame delimiter (SFD).
- **Start frame delimiter (SFD).** The second field (one byte: 10101011) of the 802.3 frame signals the beginning of the frame. The SFD tells the receiver that every-thing that follows is data, starting with the addresses.
- **Destination address (DA).** The destination address (DA) field is allotted six bytes and contains the physical address of the packet's next destination. A system's physical address is a bit pattern encoded on its network interface card (NIC). Each NIC has a unique address that distinguishes it from any other NIC. If the packet must cross from one LAN to another to reach its destination, the DA field contains the physical address of the router connecting the current LAN to the next one. When the packet reaches the target network, the DA field contains the physical address of the destination device.
- **Source address (SA).** The source-address (SA) field is also allotted six bytes and contains the physical address of the last device to forward the packet. That device can be the sending station or the most recent router to receive and forward the packet.
- **Length/type of PDU.** These next two bytes indicate the number of bytes in the coming PDU. If the length of the PDU is fixed, this field can be used to indicate type, or as a base for other protocols. For example, Novell and the Internet use it to identify the network layer protocol that is using the PDU.
- **802.2 frame (PDU).** This field of the 802.3 frame contains the entire 802.2 frame as a modular, removable unit. The PDU can be anywhere from 46 to 1500 bytes long, depending on the type of frame and the length of the information field. The PDU is generated by the upper (LLC) sublayer, then linked to the 802.3 frame.
- **CRC.** The last field in the 802.3 frame contains the error detection information, in this case a CRC-32.

Implementation

Although the bulk of the IEEE Project 802 standard focuses on the data link layer of the OSI model, the 802 model also defines some of the physical specifications for each of the protocols defined in the MAC layer. In the 802.3 standard, the IEEE defines the types of cable, connections, and signals that are to be used in each of five different Ethernet implementations. All Ethernet LANs are configured as logical buses, although they may be physically implemented in bus or star topologies. Each frame is transmit-ted to every station on the link but read only by the station to which it is addressed.

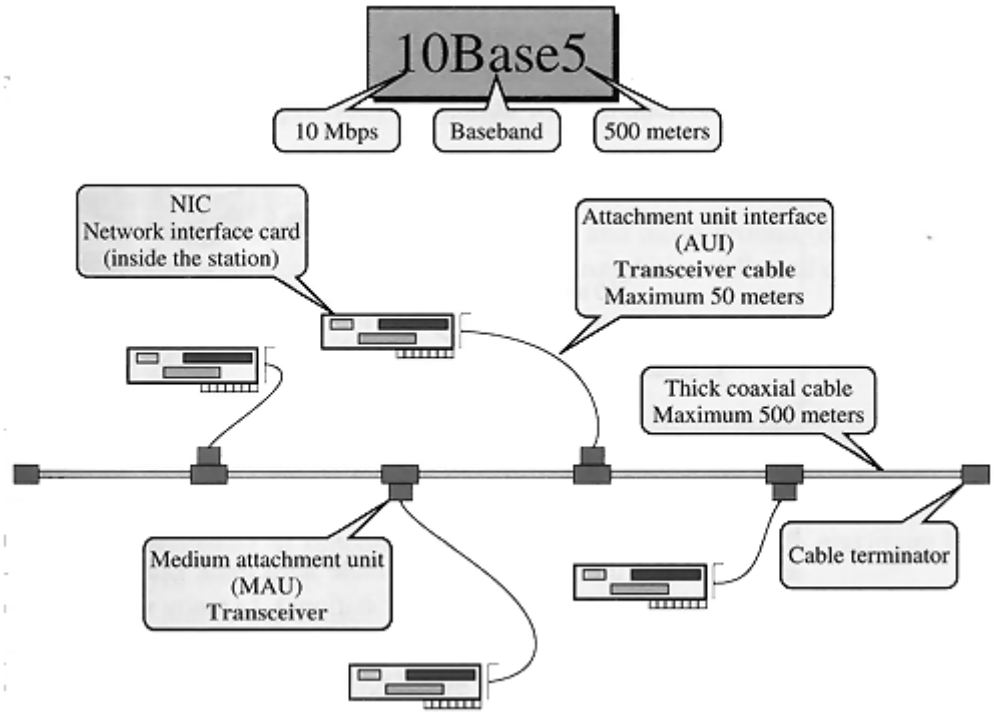
10BASE5: Thick Ethernet

The first of the physical standards defined in the IEEE 802.3 model is called 10Base5, **thick Ethernet**, or **Thicknet**. The nickname derives from the size of the cable, which is roughly the size of garden hose and too stiff to bend with your hands. 10Base5 is a bus topology LAN that uses baseband signaling and has a maximum segment length of 500 meters.

In thick Ethernet, a local area network can be divided into segments by connecting devices. In this case, the length of each segment is limited to 500 meters. However, to reduce collisions, the total length of the bus should not exceed 2500 meters (five segments). Also, the standard demands that each station be separated from each neighbor by 2.5 meters (200 stations per segment and 1000 stations total).

The physical connectors and cables utilized by 10Base5 include coaxial cable, net-work interface cards, transceivers, and attachment unit interface (AUI) cables. The interaction of these components is illustrated in Figure 11.8.

Figure 11.8 Topology of 10BASE5



RG-8 Cable RG-8 cable (RG stands for radio government) is a thick coaxial cable that provides the backbone of the IEEE 802.3 standard.

Transceiver Each station is attached by an AUI cable to an intermediary device called a medium attachment unit (MAU) or, more commonly, a transceiver (short for transmitter-receiver). The transceiver performs the CSMA/CD function of checking for voltages and collisions on the line and may contain a small buffer. It also serves as the connector that attaches a station to the thick coaxial cable itself via a tap..

AUI Cables Each station is linked to its corresponding transceiver by an **attachment unit interface (AUI)**, also called a **transceiver cable**. An AUI is a 15-wire cable with plugs that performs the physical layer interface functions between the station and the transceiver. Each end of an AUI terminates in a DB-15 (15-pin) connector. One connector plugs into a port on the NIC, the other into a port on the transceiver. AUIs are restricted to a maximum length of 50 meters, allowing for some flexibility in placement of stations relative to the 10BASE5 backbone cable.

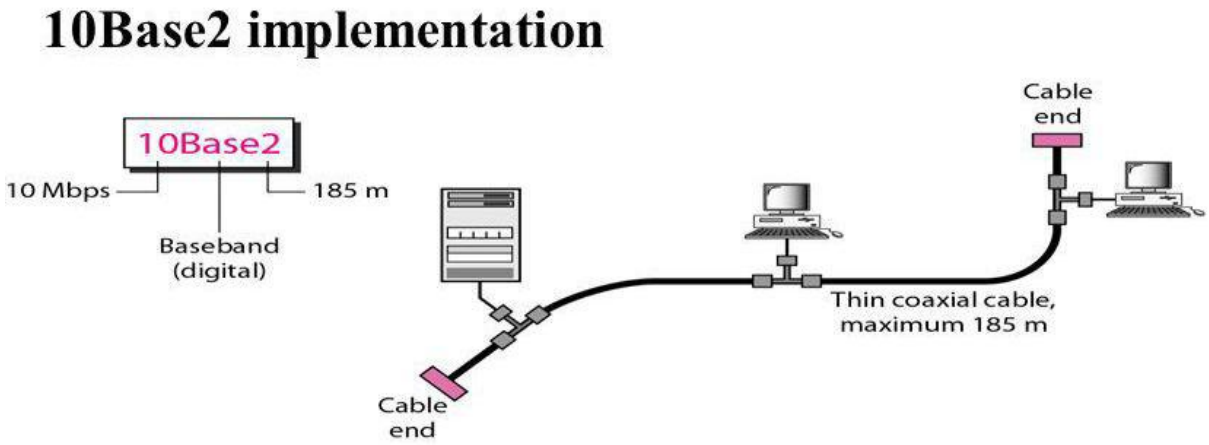
Transceiver Tap Each transceiver contains a connecting mechanism, called a tap because it allows the transceiver to tap into the line at any point. The tap is a thick cable-sized well with a metal spike in the center. The spike is attached to wires inside the transceiver. When the cable is pressed into the well, the spike pierces the jacket and sheathing layers and makes an electrical connection between the transceiver and the cable. This kind of connector is often called a **vampire tap** because it bites the cable.

10BASE2: Thin Ethernet

The second Ethernet implementation defined by the IEEE 802 series is called 10Base2 or thin Ethernet. Thin Ethernet (also called **Thinnet, cheapnet, cheapernet**, and **thin-wire Ethernet**) provides an inexpensive alternative to l0Base5 Ethernet, with the same data rate. Like 10Base5, 10Base2 is a bus topology LAN. The advantages of thin Ether-net are reduced cost and ease of installation (the cable is lighter weight and more flexible than that used in thick Ethernet). The disadvantages are shorter range (185 meters as opposed to the 500 meters available with thick Ethernet) and smaller capacity (the thinner cable accommodates fewer stations).

The physical layout of 10Base2 is illustrated in Figure 11.9. The connectors and cables utilized are: NICs, thin coaxial cable, and BNC-T connectors. In this technology, the transceiver circuitry has moved into the NIC, and the transceiver tap has been replaced by a connector that splices the station directly into the cable, eliminating the need for AUI cables.

Figure 11.9 Topology of 10BASE2



NIC The NICs in a thin Ethernet system provide all of the same functionality as those in a thick Ethernet system, plus the functions of the transceivers. That means that a 10Base2 NIC not only provides the station with an address but also:,becks for volt-ages on the link.

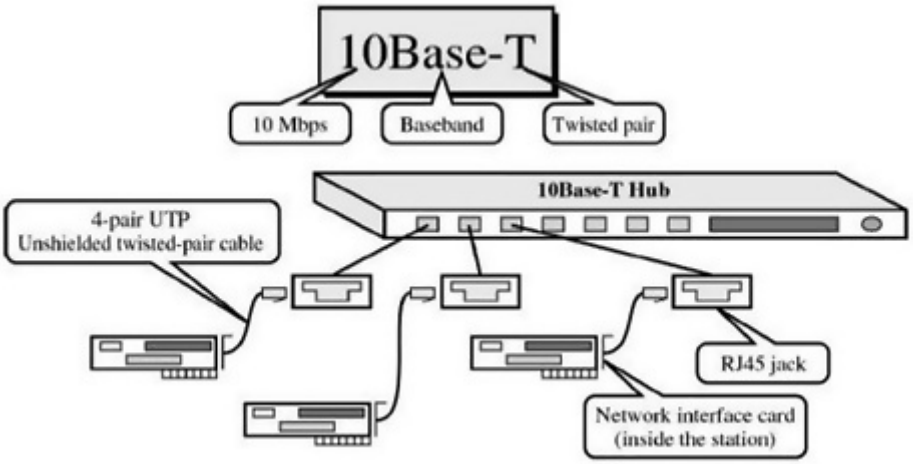
Thin Coaxial Cable The cable required to implement the 10Base2 standard is RG-58. These cables are relatively easy to install and move around (especially inside existing buildings where cabling must be pulled through the walls and ceilings).
BNC-T The BNC-T connector is a T-shaped device with three ports: one for the NIC and one each for the input and output ends of the cable.

10BASE-T:Twisted-Pair Ethernet

The most popular standard defined in the IEEE 802.3 series is 10Base-T (also called twisted-pair Ethernet), a star-topology LAN using unshielded twisted pair (UTP) cable instead of coaxial cable. It supports a data rate of 10 Mbps and has a maximum length (hub to station) of 100 meters.

Instead of individual transceivers, 10Base-T Ethernet places all of its networking operations in an intelligent hub with a port for each station. Stations are linked into the hub by four-pair RJ-45 cable (eight-wire unshielded twisted-pair cable) terminating at each end in a male-type connector much like a telephone jack (see Figure 11.10). The hub fans out any transmitted frame to all of its connected stations. Logic in the NIC assures that the only station to open and read a given frame is the station to which that frame is addressed.

Figure 11.10 10BASE-T topology



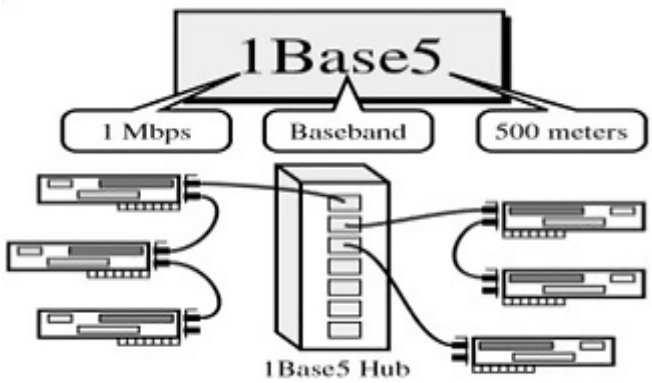
As Figure 11.10 shows, each station contains an NIC. A length of four pair UTP of not more than 100 meters connects the NIC in the station to the appropriate port in the 10 Base T hub.

The weight and flexibility of the cable and the convenience of the RJ-45 jack and plug make to Base T the easiest of the 802.3 LANs to install and reinstall. When a station needs to be replaced, a new station can simply be plugged in.

1Base5: StarLAN

StarLAN is an AT&T product used infrequently today because of its slow speed. At only 1 Mbps, it is 10 times slower than the three standards discussed above.

Figure 11.11 IBase5



What is interesting about StarLAN is its range, which can be increased by a mechanism called daisy chaining. Like 10Base-T, StarLAN uses twisted-pair cable to connect stations to a central intelligent hub. Unlike 10 Base-T, which requires that each station have its own dedicated cable into the hub, Star LAN allows as many as 10 stations to

be linked, each to the next, in a chain in which only the lead device connects to the hub (see Figure 11.11)

11.3 TOKEN BUS

Local area networks have a direct application in factory automation and process control, where the nodes are computers controlling the manufacturing process. In this type of application, real-time processing with minimum delay is needed. Processing must occur at the same speed as the objects moving along the assembly line. Ethernet (IEEE 802.3) is not a suitable protocol for this purpose because the number of collisions is not predictable and the delay in sending data from the control center to the computers along the assembly line is not a fixed value. **Token Ring (IEEE 802.5)** is also not a suitable protocol because an assembly line resembles a bus topology and not a ring. **Token Bus (IEEE 802.4**) combines features of Ethernet and Token Ring. It combines the physical configuration of Ethernet (a bus topology) and the collision-free (predictable delay) feature of Token Ring. Token Bus is a physical bus that operates as a logical ring using **tokens**.

Stations are logically organized into a ring. A token is passed among stations. If a station wants to send data, it must wait and capture the token. However, like Ethernet, stations communicate via a common bus.

Token Bus is limited to factory automation and process control and has no commercial application in data communication. Also, the details of the operation are very involved. For these two reasons, we will not discuss this protocol further.

11.4 TOKEN RING

The network access mechanism used by Ethernet (CSMA/CD) is not infallible and may result in collisions. Stations may attempt to send data multiple times before a transmission makes it onto the link. This redundancy may create delays of indeterminable length if the traffic is heavy. There is no way to predict either the occurrence of collisions or the delays produced by multiple stations attempting to capture the link at the same time.

Token Ring resolves this uncertainty by requiring that stations take turns sending data. Each station may transmit only during its turn and may send only one frame during each turn. The mechanism that coordinates this rotation is called token passing. A token is a simple placeholder frame that is passed from station to station around the ring. A station may send data only when it has possession of the token.

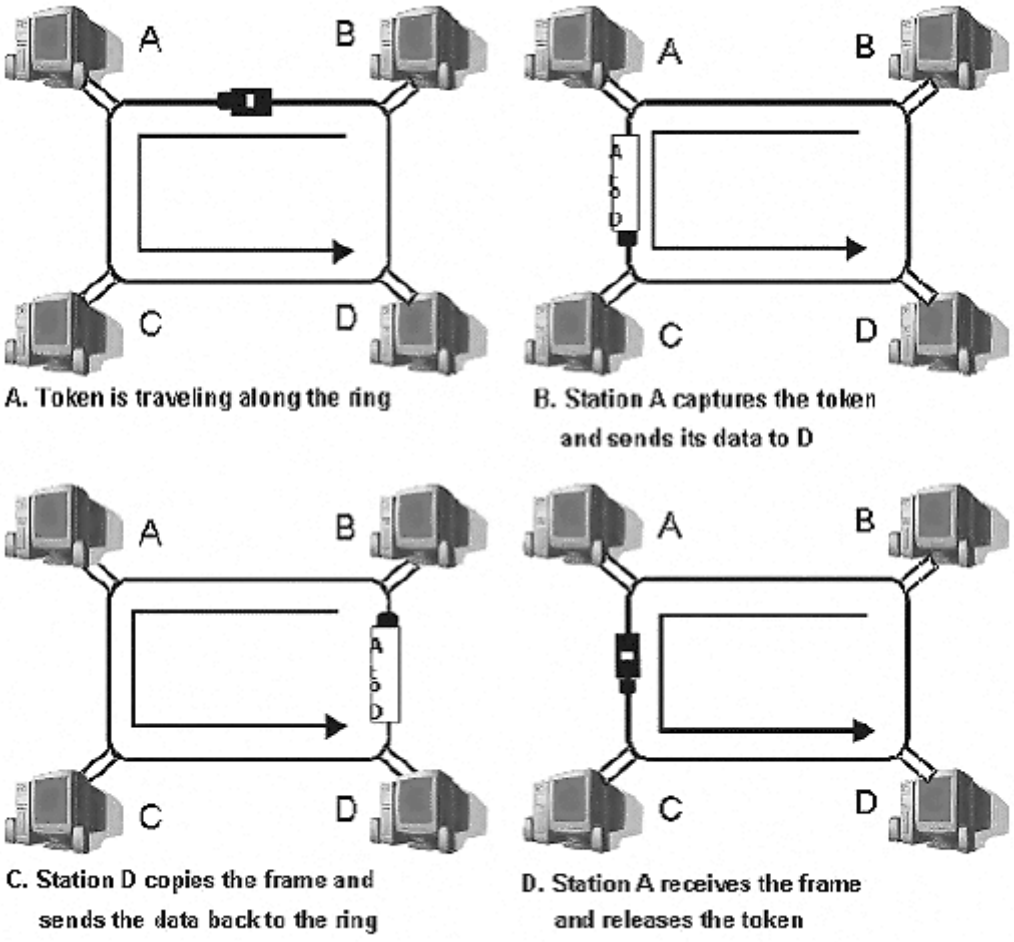
Token Ring allows each station to send one frame per turn.

Access Method: Token Passing

Token passing is illustrated in Figure 11.12. Whenever the network is unoccupied, it circulates a simple three-byte token. This token is passed from NIC to NIC in sequence until it encounters a station with data to send. That station waits for the token to enter its network board. If the token is free, the station may then send a data frame. It keeps the token and sets a bit inside its NIC as a reminder that it has done so, then sends its one data frame.

This data frame proceeds around the ring, being regenerated by each station. Each intermediate station examines the destination address, finds that the frame is addressed to another station, and relays it to its neighbor. The intended recipient recognizes its own address, copies the message, checks for errors, and changes four bits in the last byte of the frame to indicate address recognized and frame copied. The full packet then continues around the ring until it returns to the station that sent it.

Figure 11.12 Token Passing



The sender receives the frame and recognizes itself in the source address field. It then examines the address-recognized bits. If they are set, it knows the frame was received. The sender then discards the used data frame and releases the token back to the ring.

Priority and Reservation

Generally, once a token has been released, the next station on the ring with data to send has the right to take charge of the ring. However, in the IEEE 802.5 model, option is possible. The busy token can be reserved by a station waiting to transmit, regardless of that station's location on the ring. Each station has a priority code. As a frame passes by, a station waiting to transmit may reserve the next open token by entering its priority code in the access control (AC) field of the token or data frame. A station with a higher priority may remove a lower priority reservation and replace it with its own. Among stations of equal priority, the process is first-come, first-served. Through this mechanism, the station holding the reservation gets the opportunity to transmit as soon as the token is free, whether or not it comes next physically on the ring.

Time Limits
To keep traffic moving, Token Ring imposes a time limit on any station wanting to use the ring. A starting delimiter (the first field of either a token or data frame) must reach each station within a specified interval (usually 10 milliseconds). In other words, each station expects to receive frames within regular time intervals

Monitor Stations
Several problems may occur to disrupt the operation of a Token Ring network. In another scenario, a station may neglect to retransmit a token or a token may be destroyed I noise, in which case there is no token on the ring and no station may send data. Another scenario, a sending station may neglect to remove its used data frame from the ring or may not release the token once its turn has ended.

To handle these situations, one station on the ring is designated as a monitor station. The monitor sets a timer each time the token passes. If the token does not reappear in the allotted time, it is presumed to be lost and the monitor generates a new token and introduces it to the ring. The monitor guards against perpetually recirculating data frames by setting a bit in the AC field of each frame. As a frame passes, the monitor checks the status field. If the status bit has been set, it knows that the packet has already been around the ring and should have been discarded. The monitor then destroys the frame and puts a token onto the ring. If the monitor fails, a second station, designate as back-up, takes over.

Addressing
Token Ring uses a six-byte address, which is imprinted on the NIC card similar to Ethernet addresses.

Electrical Specification

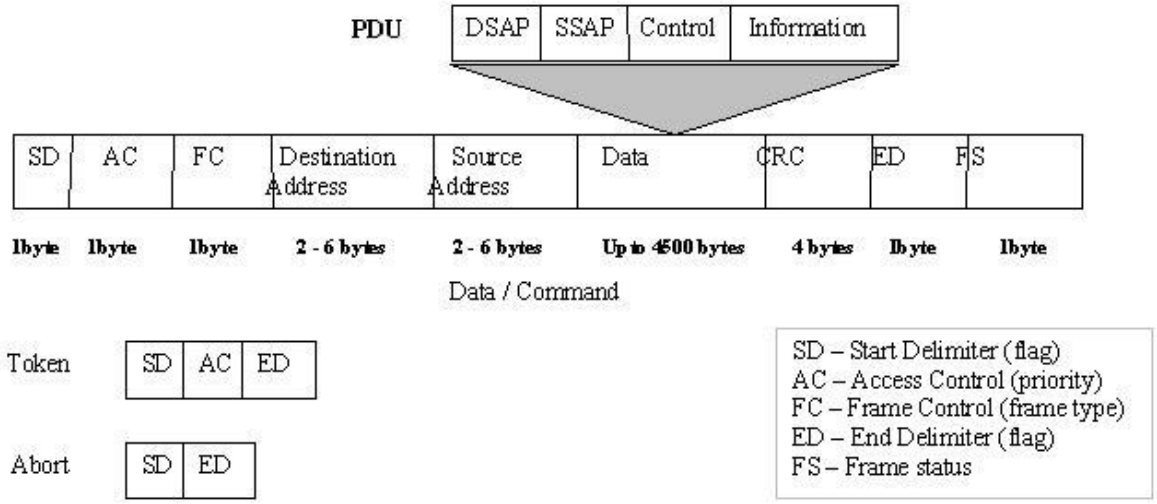
Signaling
Token Ring uses differential Manchester encoding.

Data Rate
Token Ring supports data rates of up to 16 Mbps. (The original specification was 4 Mbps.)

Frame Formats
The Token Ring protocol specifies three types of frames: data/command, token, a abort. The token and abort frames are both truncated data/command frames (Figure 11.13).

Data/Command Frame
In Token Ring, the data/command frame is the only one of the three types of frames that can carry a PDU and is the only one addressed to a specific destination rather than being available to the ring at large. This frame can carry either the user data or the management commands. The nine fields of the frame are start delimiter (SD), access control (AC), frame control (FC), destination address (DA), source address (SA), 80 PDU frame, CRC, end delimiter (ED), and frame status (FS).

Figure 11.13 Token Ring frame



- **Start delimiter (SD).** The first field of the data/command frame, SD, is one byte long and is used to alert the receiving station to the arrival of a frame as well as to allow it to synchronize its retrieval timing.
- **Access control (AC).** The AC field is one byte long and includes four subfields . The first three bits are the priority field. The fourth bit is called the token bit and is set to indicate that the frame is a data/command frame rather than a token or an abort frame. The token hit is followed by a monitor bit. Finally, the last three bits are a reservation field that can be set by stations wishing to reserve access to the ring.
- **Frame control (FC).** The FC field is one byte long and contains two fields. The first is a one-bit field used to indicate the type of information contained in the PDU (whether it is control information or data). The second uses the remaining seven bits of the byte and contains information used by the Token Ring logic
- **Destination address (DA).** The two- to six-byte DA field contains the physical address of the frame's next destination. If its ultimate destination is another net-work, the DA is the address of the router to the next LAN on its path. If its ultimate destination is on the current LAN, the DA is the physical address of the destination station.
- **Source address (SA).** The SA field is also two to six bytes long and contains the physical address of the sending station. If the ultimate destination of the packet is a station on the same network as the originating station, the SA is that of the originating station. If the packet has been routed from another LAN, the SA is the physical address of the most recent router.
- **Data.** The sixth field, data, is allotted 4500 bytes and contains the PDU. A Token Ring frame does not include a PDU length or type field.
- **CRC.** The CRC field is four bytes long and contains a CRC-32 error detection sequence.
- **End delimiter (ED).** The ED is a second flag field of one byte and indicates the end of the sender's data and control information.
- **Frame status (FS).** The last byte of the frame is the FS field. It can be set by the receiver to indicate that the frame has been read, or by the monitor to indicate that the frame has

already been around the ring. This field is not an acknowledgment, but it does tell the sender that the receiving station has copied the frame, which can now be discarded.

Token Frame

Because a token is really a placeholder and reservation frame, it includes only three fields: the SD, AC, and ED. The SD indicates that a frame is coming. The AC indicates that the frame is a token and includes the priority and reservation fields. The ED indicates the end of the frame.

Abort Frame

An abort frame carries no information at all—just starting and ending delimiters. It can be generated either by the sender to stop its own transmission or by the monitor to purge an old transmission from the line.

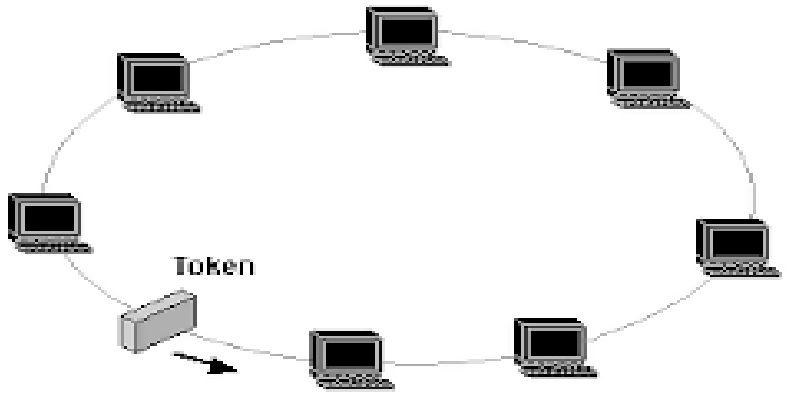
Implementation

Ring

The ring in a Token Ring consists of a series of 150-ohm, shielded twisted-pair sections linking each station to its immediate neighbors (see Figure 11.14). Each section connects an output port on one station to an input port on the next, creating a ring with unidirectional traffic flow. The output from the final station connects to the input of the first to complete the ring. A frame is passed to each station in sequence, where it is examined, regenerated, and then sent on to the next station.

Each station in the Token Ring regenerates the frame.

Figure 11.14 Token Ring



Switch

As Figure 11.14 shows, configuring the network as a ring introduces a potential problem: One disabled or disconnected node could stop the flow of traffic around the entire network. To solve this problem, each station is connected to an automatic switch. This switch can bypass an inactive station. While a station is disabled, the Switch closes the ring without it. When the station comes on, a signal sent by the NIC moves the switch and brings the station into the ring (Figure 11.15).

Each station's NIC has a pair of input and output ports combined in a nine-pin connector. A nine-wire cable connects the NIC to the switch. Of these wires, four are used for data and the remaining five are used to control the switch (to include or bypass a station).

Figure 11.15 Token Ring switch

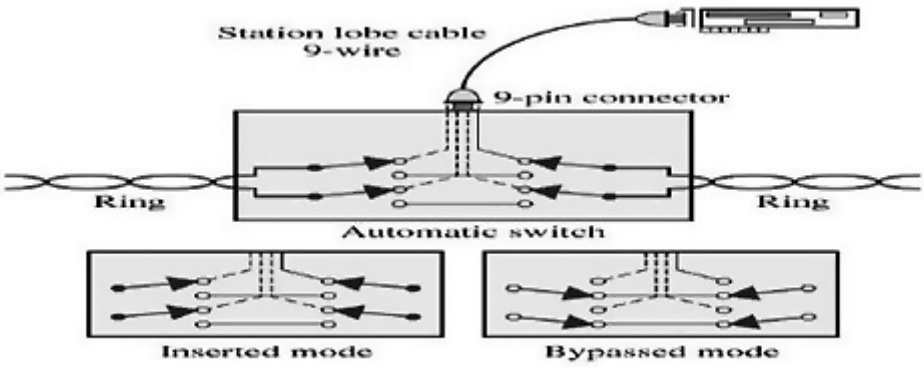


Figure 11.15 shows the two switching modes. In the first part, connections are completed to the station, thereby inserting it into the ring. In the second part, an alternate pair of connections is completed to bypass the station.

Multistation Access Unit (MAU)
For practical purposes, individual automatic switches are combined into a hub called a multistation access unit (MAU). One MAU can support up to eight stations. Looked at from the outside, this system looks like a star with the MAU at the middle.

11.5 FDDI

Fiber distributed data interface (FDDI) is a local area network protocol standardized by ANSI and the ITU-T (ITU-T X.3). It supports data rates of 100 Mbps and provides a high-speed alternative to Ethernet and Token Ring. When FDDI was designed, speeds of 100 Mbps required fiber-optic cable. Today, however, comparable speeds are available using copper cable. The copper version of FDDI is known as CDDI.

Access Method: **Token Passing**

In FDDI, access is limited by time. A station may send as many frames as it can within its allotted access period, with the proviso that real-time data be sent first.

To implement this access mechanism, FDDI differentiates between two types of data frames: synchronous and asynchronous. Synchronous here refers to information that is real-time, while asynchronous refers to information that is not. These frames are usually called S-frames and A-frames.

Each station that captures the token is required to send 'S-frames first. In fact, it must send its S-frames whether or not its time allotment has run out . Any remaining time may then be used to send A-frames. To understand how this mechanism ensures fair and timely link access, it is necessary to understand the FDDI time registers and timers.

Time Registers
FDDI defines three time registers to control circulation of the token and distribute link-access opportunities among the nodes equitably. Values are set when the ring is initialized and do

not vary in the course of operation. The registers are called synchronous allocation (SA), target token rotation time (TTRT), and absolute maximum time (AMT).

Synchronous Allocation (SA) The SA register indicates the length of time allowed each station for sending synchronous data. This value is different for each station and is negotiated during initialization of the ring.

Target Token Rotation Time (TTRT) The TTRT register indicates the average time required for a token to circulate around the ring exactly once (the elapsed time between a token's arrival at a given station and its next arrival at the same station). Because it is an average, the actual time of any rotation may be greater or less than this value.

Absolute Maximum Time (AMT) The AMT register holds a value equal to twice the TTRT. A token may not take longer than this time to make one rotation of the ring. If it does, some station or stations are monopolizing the network and the ring must be reinitialized.

Timers

Each station contains a set of timers that enable it to compare actual timings with the values contained in the registers. Timers can be set and reset, and their values decremented or incremented at a rate set by the system clock. The two timers used by FDDI are called the token rotation timer (TRT) and token holding timer (THT).

Token Rotation Timer (TRT) The TRT runs continuously and measures the actual time taken by the token to complete a cycle. In our implementation, we use an incrementing TRT for simplicity, although some implementations may use a decrementing timer.

Token Holding Timer (THT) The THT begins running as soon as the token is received. Its function is to show how much time remains for sending asynchronous frames once the synchronous frames have been sent. In our implementation, we use a decrementing THT for simplicity, although some implementations may use an incrementing one. In addition, we allow the value of THT to become negative although a real timer may stay at zero.

Station Procedure

When a token arrives, each station follows this procedure:

1. THT is set to the difference between TTRT and TRT (THT = TTRT — TRT).
2. TRT is reset to zero (TRT = 0).
3. The station sends its synchronous data.
4. The station sends asynchronous data as long as the value of THT is positive.

Electrical Specification

Signaling (Physical Layer)

FDDI uses-a special encoding mechanism called four bits/five bits (4B/5B). In this system, each four-bit segment of data is replaced by a five-bit code before being encoded in NRZ. The NRZ-I used here inverts on the 1.

The reason for this extra encoding step is that, although NRZ-I provides adequate synchronization under average circumstances, sender and receiver may go out of synchronization anytime the data includes a long sequence of 0s. 4B/5B encoding transforms each four-bit data segment into a five-bit unit that contains no more than two consecutive 0s.

Data Rate

FDDI supports data rates up to 100 Mbps.

Frame Format

The FDDI standard divides transmission functions into four protocols: physical medium dependent (PMD), physical (PHY), media access control (MAC), and logical link control (LLC). These protocols correspond to the physical and data link layers of the OSI model. In addition, the standard specifies a fifth protocol (used for station management), details of which are beyond the scope of this book.

Logical Link Control

The LLC layer is similar to that defined in the IEEE 802.2 protocols.

Media Access Control

The FDDI MAC layer is almost identical to that defined for Token Ring and the functions are similar,

Each frame is preceded by 16 idle symbols (1111), for a total of 64 bits, to initialize clock synchronization with the receiver.

Frame Fields

There are eight fields in the FDDI frame:

- **Start delimiter** (SD). The first byte of the field is the frame's starting flag. As in Token Ring, these bits are replaced in the physical layer by the control codes.
- **Frame control** (FC). The second byte of the frame identifies the frame type.
- **Addresses.** The next two fields are the destination and source addresses. Each address consists of two to six bytes.
- **Data**. Each data frame can carry up to 4500 bytes of data.
- **CRC**. FDDI uses the standard IEEE four-byte cyclic redundancy check.
- **End delimiter (ED).** This field consists of half a byte in the data frame or a full byte in the token frame.
- **Frame status (FS).** The FDDI FS field is similar to that of Token Ring. It is included only in the data/command frame and consists of 1.5 bytes.

Implementation: Physical Medium Dependent (PMD) Layer

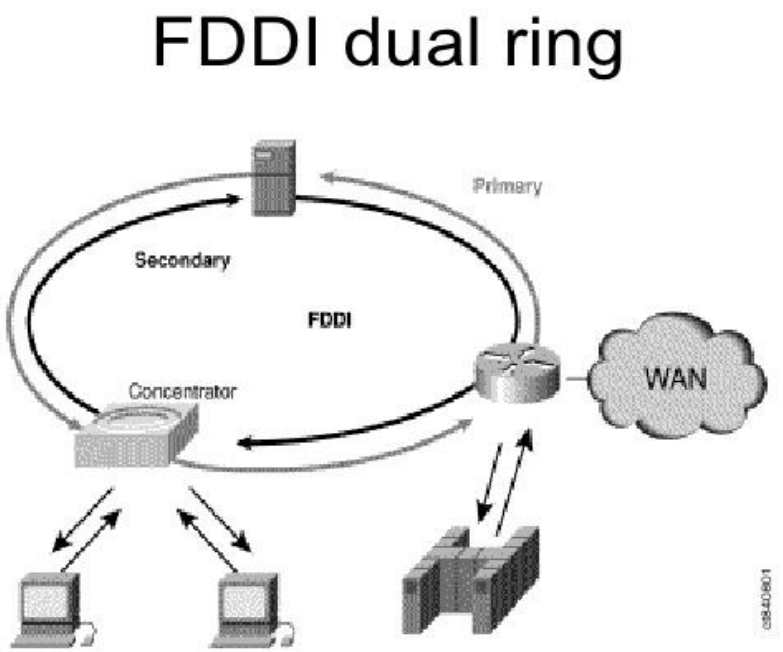
The physical medium dependent (PMD) layer defines the required connections and electronic components. Specifications for this layer depend on whether the transmission medium used is fiber-optic or copper cable.

Dual Ring

FDDI is implemented as a dual ring (see Figure 12.31). In most cases, data trans-mission is confined to the primary ring. The secondary ring is provided in case the primary fails.

The secondary ring makes FDDI self-healing. Whenever a problem occurs on the primary ring, the secondary can be activated to complete data circuits and maintain service Figure 11.16.

Figure 11.16 FDDI ring



Nodes connect to one or both rings using a media interface connector (MIC) that can be either male or female depending on the requirements of the station.

Nodes
FDDI defines three types of nodes: dual attachment station (DAS), single attachment station (SAS), and dual attachment concentrator (DAC).

DAS A dual attachment station (DAS) has two MICs (called MIC A and MIC B) and connects to both rings. To do so requires an expensive NIC with two inputs and two outputs. The connection to both rings gives it improved reliability and throughput. These improvements, however, are predicated on the stations remaining on. Faults are bypassed by a station's making a wrap connection from the primary ring to the secondary to switch signals from one input to another output. However, for DAS stations to make this switch, they must be active (turned on).

SAS Most workstations, servers, and minicomputers attach to the ring in single attachment station (SAS) mode. An SAS has only one MIC (called MIC S) and therefore can connect only to one ring. Robustness i, achieved by connecting SASs to intermediate nodes, called dual attachment concentrators (DACs), rather than to the FDDI ring directly. This configuration allows each workstation to operate through a simple NIC with only one input and one output. The concentrator (DAC) provides the connection to the dual ring. Faulty stations can be turned off and bypassed to keep the ring, alive.

DAC As mentioned above, a dual attachment concentrator (DAC) connects an SAS to the dual ring. It provides wrapping (diverting traffic from one ring to the other to bypass a failure) as well as control functions. It uses MIC M to connect to an SAS.

.

UNIT V

12. TCP/IP PROTOCOL SUITE PART I

The Transmission Control Protocol / Internet working Protocol (TCP/IP) is a set of protocols, or a protocol suite, that defines how all transmissions are exchanged across the Internet.

12.1 OVERVIEW OF TCP/IP

In 1969, a project was funded by the **Advanced Research Project Agency (ARPA),** an arm of the U.S. Department of Defense. ARPA established a packet-switching network of computers linked by point-to-point leased lines called **Advanced Research Project Agency Network (ARPANET)** that provided a basis for early research into networking. The conventions developed by ARPA to specify how individual computers could communicate across that network became TCP/IP.

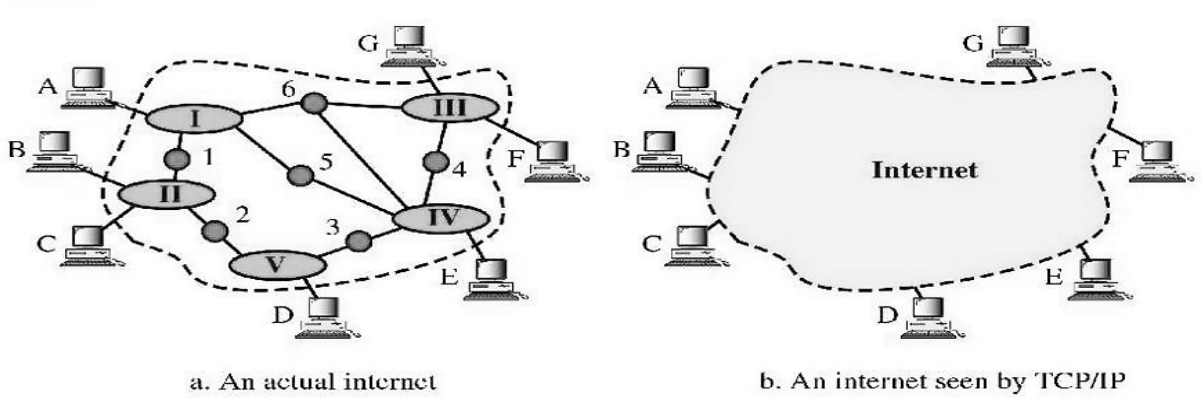
As networking possibilities grew to include other types of links and devices, ARPA adapted TCP/IP to the demands of the new technology. As involvement in TCP/IP grew, the scope of ARPANET expanded until it became the backbone of an internet-work today referred to as the Internet..

TCP/IP and the Internet

TCP/IP and the concept of internetworking developed together, each shaping the growth of the other. Before moving more deeply into the protocols, however, we need to understand how TCP/IP relates to the physical entity of any internet it serves.

An internet under TCP/IP operates like a single network connecting many computers of any size and type. Internally, an internet is an inter-connection of independent physical networks (such as LANs) linked together by internet-working devices. Figure 12.1 shows the topology of a possible internet. In this example, the letters A, B, C, and so on represent hosts. A **host** in TCP/IP is a computer. The solid circles in the figure, numbered 1, 2, 3, and so on, are routers or gateways. The larger ovals containing roman numerals (I, IL Ill, etc.) represent separate physical networks.

Figure 12.1 An internet according to TCP/IP



To TCP/IP, the same internet appears quite differently (Figure 12.1). TCP/IP considers all interconnected physical networks to be one huge network. It considers all of the hosts to be connected to this larger logical network rather than to their individual physical networks.

TCP/IP and OSI

Transmission Control Protocol (TCP) was developed before the OSI model. Therefore, the layers in the TCP/IP protocol do not match exactly with those in the OSI model. The TCP/IP protocol is made of five layers: physical, data link, network, trans-port, and application. The application layer in TCP/IP can be equated with the combination of session, presentation, and application layers of the OSI model.

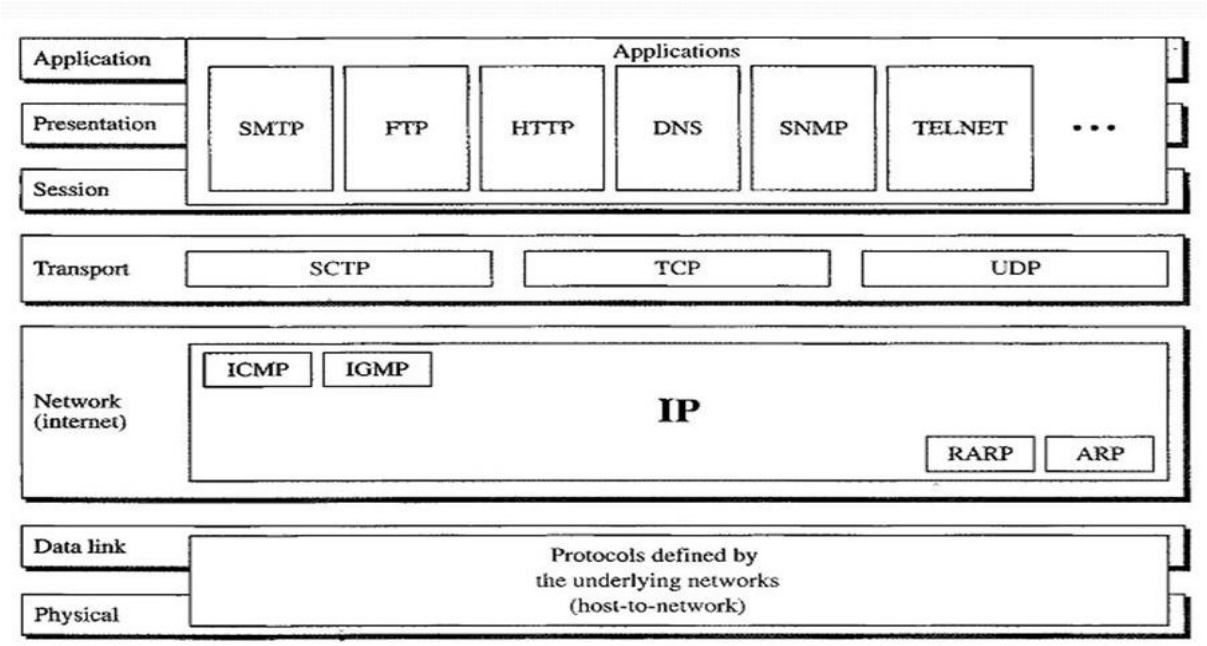
At the transport layer, TCP/IP defines two protocols: TCP and User **Datagram Protocol (UDP).** At the network layer, the main protocol defined by TCP/IP is Internetworking Protocol (IP), although there are some other protocols that support data movement in this layer.

At the physical and data link layers, TCP/IP does not define any specific protocol. A network in a TCP/IP internetwork can be a local area network (LAN), a metropolitan area network (MAN), or a wide area network (WAN).

Encapsulation

Figure 12.2 shows the encapsulation of data units at different layers of the TCP/IP protocol suite, The data unit created at the application layer is called a message. TCP or UDP creates a data unit that is called either a segment or a user datagram. The IP layer in turn will create a data unit called a datagram. The movement of the datagram across the Internet is the responsibility of the TCP/IP protocol. However, to be able to move physically from one network to another, the datagram must be encapsulated in a frame in the data link layer the underlying network and finally transmitted as signals along the transmission media.

Figure 12.2 TCP/IP and OSI model



12.2 NETWORK LAYER

At the network layer (or, more accurately, the internetwork layer), TCP/IP supports the internetwork protocol.(IP). IP, in turn, contains four supporting protocols: ARP, RARP, ICMP, and IGMP.

Internetwork Protocol (IP)

IP is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless datagram protocol—a best-effort delivery service. The term best-effort means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. Noise can cause bit errors during transmission across a medium; a congested router may discard a datagram if it is unable to relay it before a time limit runs out; routing quirks can end in looping and the ultimate destruction of a datagram; and disabled links may leave no usable path to the destination.

If reliability is important, IP must be paired with a reliable protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage. An example of a situation similar to pairing. IP with a protocol that contains reliability functions is a selfaddressed, stamped postcard included in a letter mailed through the post office. When the letter is delivered, the receiver mails the postcard back to the sender to indicate success. If the sender never receives the postcard, he or she assumes the letter was lost and sends out another copy.

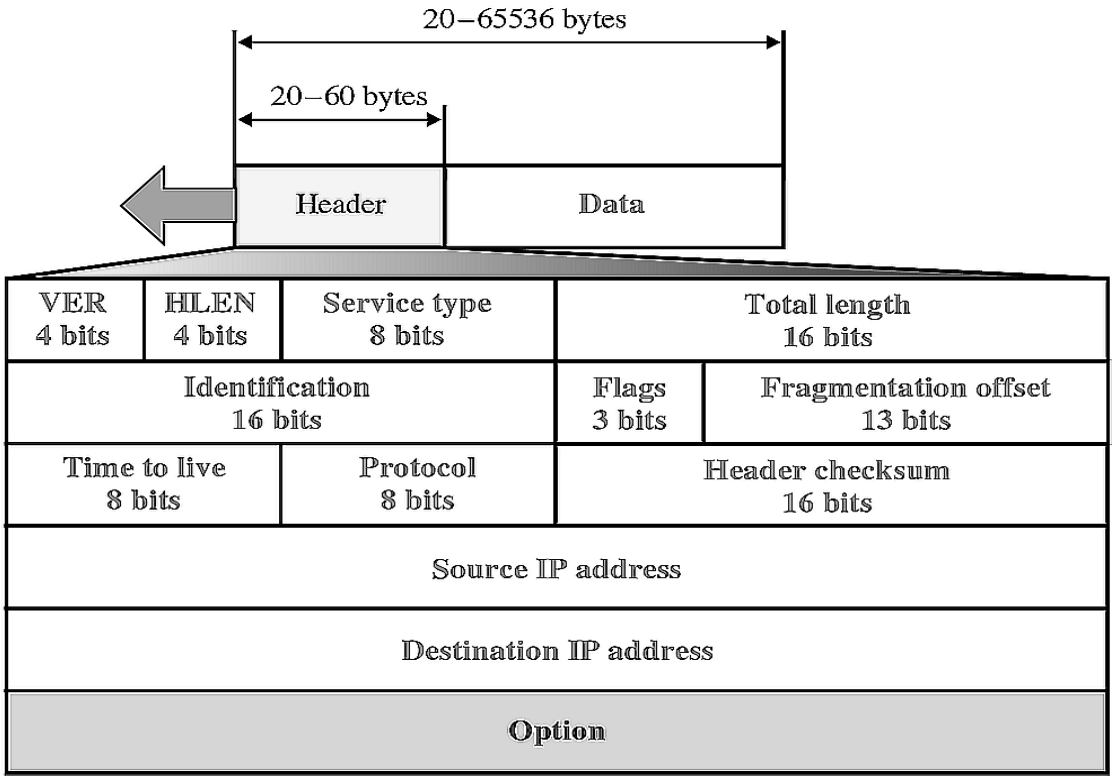
IP transports data in packets called datagram's, each of which is transported separately. Datagram's may travel along different routes and may arrive out of sequence or duplicated. IP does not keep track of the routes and has no facility for reordering datagram once they arrive. Because it is a connectionless service, IP does not create virtual circuits for delivery. There is no call setup to alert the receiver to an incoming transmission. The limited functionality of IP should not be considered a weakness, however. IP provides bare-bones transmission functions that free the user to add only those facilities necessary for a given application and thereby allows for maximum efficiency.

Datagram

Packets in the IP layer are called datagrams. Figure 12.3 shows the **IP datagram** format. A datagram is a variable-length packet (up to 65,536 bytes) consisting of two parts: header and data. The header can be from 20 to 60 bytes and contains information essential to routing and delivery. It is customary in TCP/IP to show the header in four-byte sections, A brief description of each field is in order.

- **Version.** The first field defines the version number of the Ix The current version is 4 (IPv4), with a binary value of 0100.
- **Header length (HLEN).** The HLEN field defines the length of the header in multiples of four bytes. The four bits can represent a number between 0 and 15, which, when multiplied by 4, gives a maximum of 60 bytes.
- **Service type.** The service type field defines how the datagram should be handled. It includes bits that define the priority of the datagram. It also contains bits that specify

Figure 12.3 IP datagram



the type of service the sender desires such as the level of throughput, reliability, and delay.

- **Total length.** The total length field defines the total length of the IP datagram. It is a two-byte field (16 bits) and can define up to 65,535 bytes.
- **Identification.** The identification field is used in **fragmentation.** A datagram, when passing through different networks, may be divided into fragments to match the network frame size. When this happens, each fragment is identified with a sequence number in this field.
- **Flags.** The bits in the flags field deal with fragmentation.
- **Fragmentation offset.** The fragmentation offset is a pointer that shows the offset of the data in the original datagram.
- **Time to live**. The time-to-live field defines the number of hops a datagram can travel before it is discarded. The source host, when it creates the datagram, sets this field to an initial value. Then, as the datagram travels through the Internet, router by router, each router decrements this value by 1, If this value becomes 0 before the datagram reaches its final destination, the datagram is discarded. This prevents a datagram from going back and forth forever between routers.
- **Protocol**. The protocol field defines which upper-layer protocol data are encapsulated in the datagram (TCP, UDP, ICMP, etc.).
- **Header checksum**. This is a 16-bit field used to check the integrity of the header, not the rest of the packet.

- **Source address.** The source address field is a four-byte (32-bit) Internet address. It identifies the original source of the datagram.
- **Destination address**. The destination address field is a four-byte (32-bit) Internet address. it identifies the final destination of the datagram.
- **Options.** The options field gives more functionality to the IP datagram. It can carry fields that control routing, timing, management, and alignment.

12.3 ADDRESSING

In addition to the physical addresses (contained on NICs) that identify individual devices, the Internet requires an additional addressing convention: an address that identifies the connection or a host to its network.

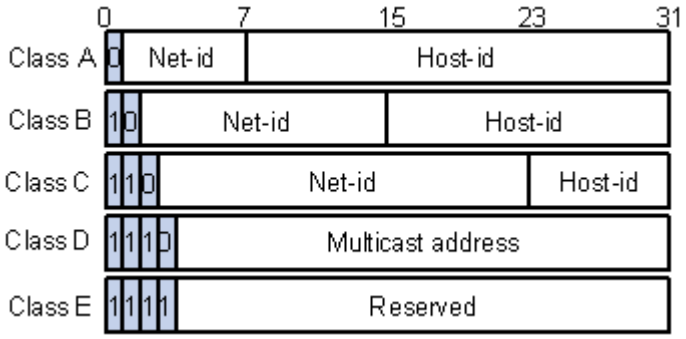
Each **Internet address** consists of four bytes (32 bits), defining three fields: class type, netid, and hostid.. These parts are of varying lengths, depending on the class of the address.

Classes

There are currently five different field-length patterns in use, each defining a **class of address**. The different classes are designed to cover the needs of different types of organizations. For example, class A addresses are numerically the lowest. They use only one byte to identify class type and netid, and leave three bytes available for hostid numbers. This division means that class A networks can accommodate far more hosts than can class B or class C networks, which provide two- and one-byte hostid fields, respectively. Currently both class A and class B are full. Addresses are available in class C only.

Class D is reserved for **multicast addresses. Multicasting** allows copies of a datagram to be passed to a select group of hosts rather than to an individual host. It is similar to broadcasting, hut, where broadcasting requires that a packet be passed to all possible destinations, multicasting allows transmission to a selected subset,. Class E addresses are reserved for future use. Figure 12.4 shows the structure of each **IP address class.**

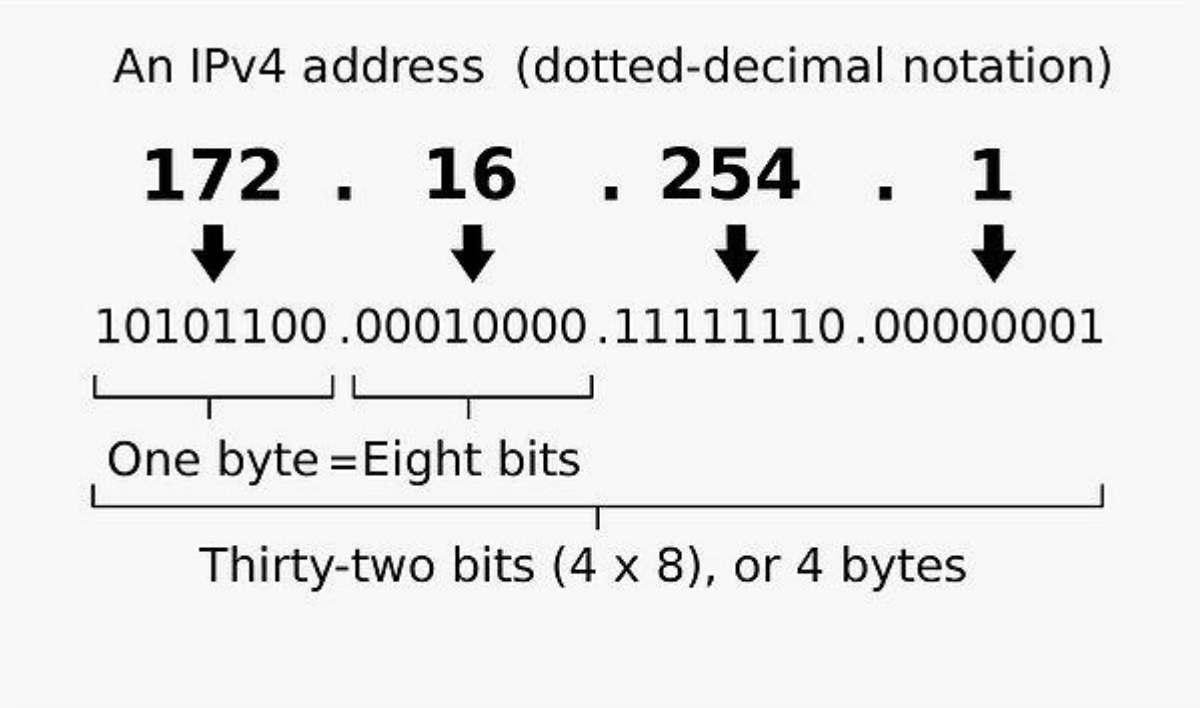
. Figure 12.4 Internet Classes



Dotted Decimal Notation

To make the 32-bit form shorter and easier to read, Internet addresses are usually written in decimal form with decimal points separating the bytes – dotted p decimal notation. Figure 12.5 shows the bit pattern and decimal format of a possible address.

Figure 12.5 IP addresses in decimal notation



Looking at the first byte of an address in decimal from allows us to determine at a glance to which class particular address belongs.

13. TCP/IP PROTOCOL SUITE: PART2 APPLICATION LAYER

13.1 FILE TRANSFER PROTOCOL (FTP)

File transfer protocol (FTP) is the standard mechanism provided by TCP/IP for copying a file from one host to another. Transferring files from one computer to another is one of the most common tasks expected from a networking or internetworking environment.

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

FTP differs from other client server applications in that it establishes two connections between the hosts. One connection is used for **data transfer**, the other for control information (commands and responses). Separation of commands and data transfer makes FTP more efficient. The control connection uses very simple rules of communication. We need to transfer only a line of command or a line of response at a time. The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.

Figure 13.1 FTP

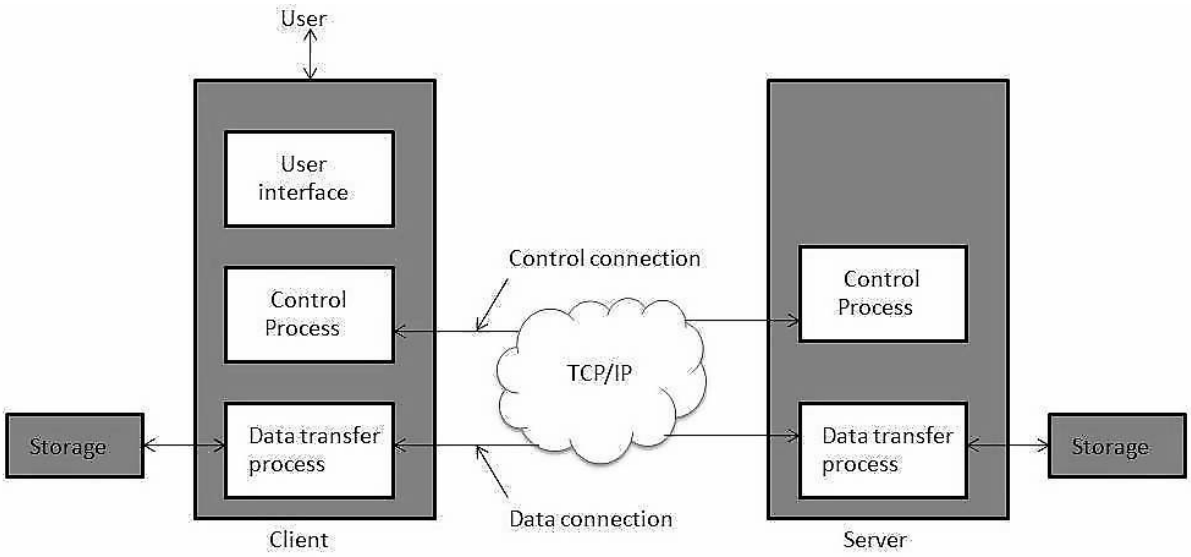


Figure 13.1 shows the basic model of FTP. The client has three components: the user interface, the client control process, and the client data transfer process. The server has two components: the server control process and the server data transfer process. The control connection is made between the control processes. The data connection is made between the data transfer processes.

The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transferred. It opens each time commands that involve transferring files are used, and it closes when the file is transferred. The t FTP connections, control and data, use different strategies and different port numbers

13.2 TELNET

The main task of the Internet and its TCP/IP protocol suite is to provide services for users. For, example, users want to be able to run different application programs at a remote site and create results that can be transferred to their local site. One way to satisfy these demands is to create different client server application programs for each desired service. Programs such as file transfer programs (FTP andTFTP), e-mail(SMTP) and so on are already available. But it would be impossible to write a specific client sever program for each demand.

The better solution is a general purpose client server program that lets a user access any application program on remote computer in other words, allow the user to log on to a remote computer. After logging on a user can use the services available on the remote computer and transfer the results back to the local computer.

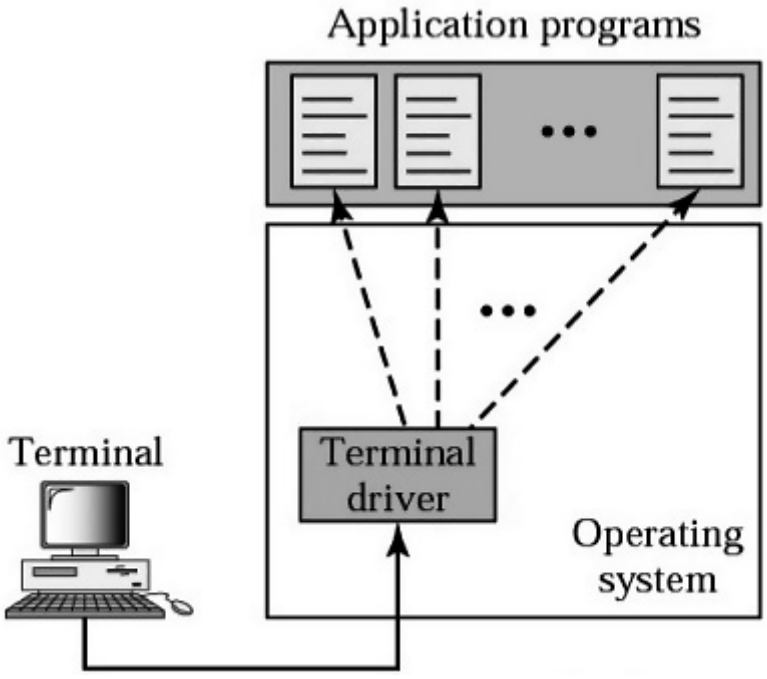
TELNET is a popular client server application program. TELENET is an abbreviation for TErminal NETwork. TELENT enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

TELNET is a general-purpose client server application program.

Local Login

When a user logs into a local time-sharing system, it .is called local login. As a user types at a terminal or at a workstation running a terminal emulator, the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system. The operating system, in turn, interprets the combination of characters and invokes the desired application program or utility (Figure 13.2).

Figure 13.2 Local Login



The mechanism, however, is not as simple as it seems because the operating sys-tem may assign special meanings to special characters. For example in UNIX some combinations of characters have special meanings, such as the combination of the control character with the character "z", which means suspend; the combination of the control character with the character "c", which means abort; and so on. Whereas these special situations do not create any problem in local login because the terminal emulator and the terminal driver know the exact meaning of each character or combination of characters, they may create problems in remote login.

Remote Login

When a user wants to access an application program or utility located on a remote machine, he or she performs remote login. Here the TELNET client and server programs come into use. The user sends the keystrokes to the terminal driver where the local operating system accepts the characters but does not interpret them. The characters are sent to the TELNET client, which transforms the characters to a universal character set called network virtual terminal characters and delivers them to the local TCP/IP stack (Figure 13.3).

Figure 13.3 Remote Login

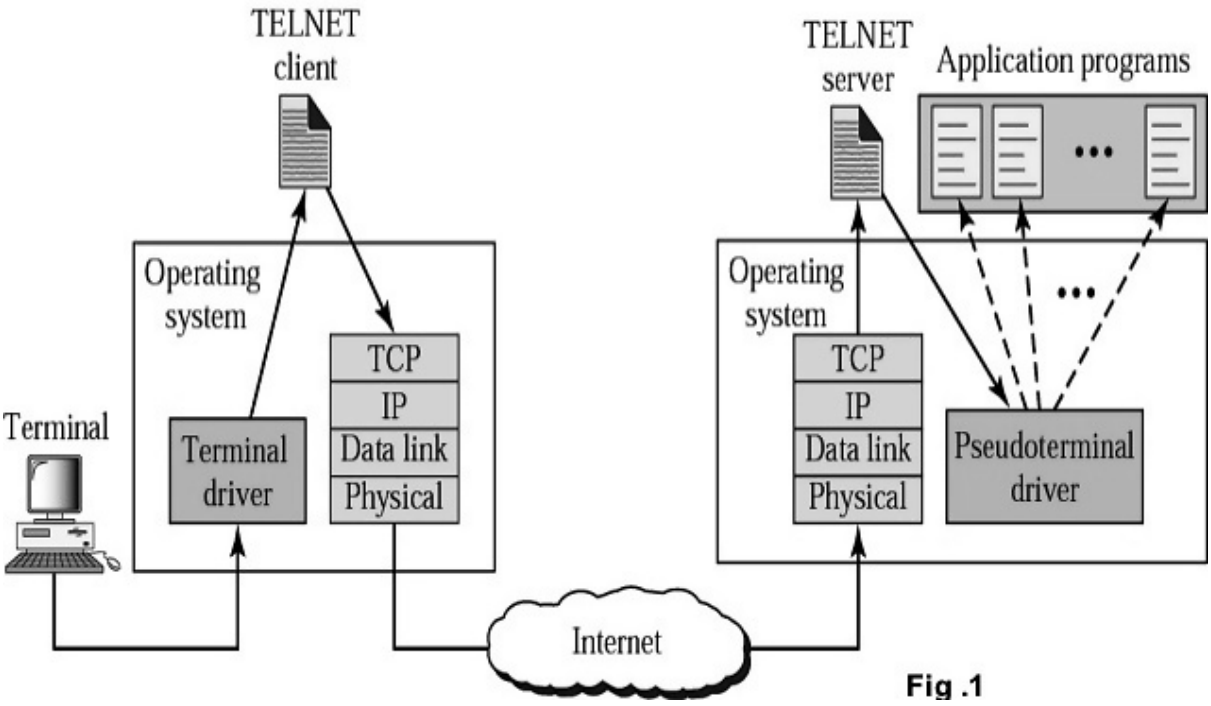


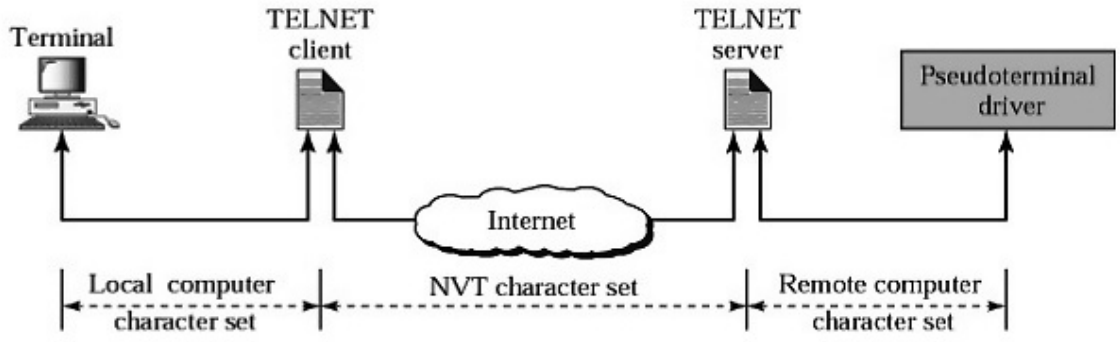
Fig .1

The commands or text, in NVT form, travel through the Internet and arrive at the TCP/IP stack at the remote machine. Here the characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer. However, the characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server: it is designed to receive characters from a terminal driver. The solution is to add a piece of software called a pseudoterminal driver, which pretends that the characters are coming from a terminal. The operating system then passes the characters to the appropriate application program

The mechanism to access a remote computer is complex. This is because every computer and its operating system accepts a special combination of characters as tokens. For example, the end-of-file token in a computer running the DOS operating system is Ctrl+z, while the UNIX operating system recognizes Ctrl+d.

We are dealing with heterogeneous systems. If we want to access any remote computer in the world, we first must know to what type of computer we will be connected, and we also must install the specific terminal emulator used by that computer. TELNET solves this problem by defining a universal interface called the network virtual terminal (NVT) character set. Via this interface, the client TELNET translates characters (data or commands) that come from the local terminal into NVT form and delivers them to the network. The server TELNET, on the other hand, translates data and commands from NVT form into the form acceptable by the remote computer.(Figure 13.4)

Figure 13.4 Concept of NVT



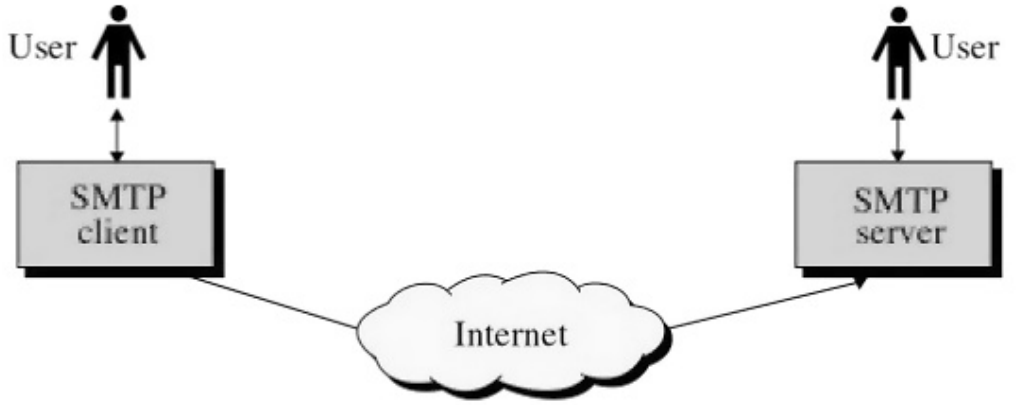
13.3 SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

One of the most popular network services is electronic mail(email). The TCP/IP protocol that supports electronic mail on the Internet is called **simple Mail Transfer Protocol** (SMTP). It is a system for sending messages to to other computer users based on email addresses. **SMPT** provides for mail exchange between users on the same or different computers and supports.

- Sending a single message to one or more recipients.
- Sending messages that include text, voice, video, or graphics.
- Sending messages to users on networks outside the Internet.

Figure 13.5 gives basic idea

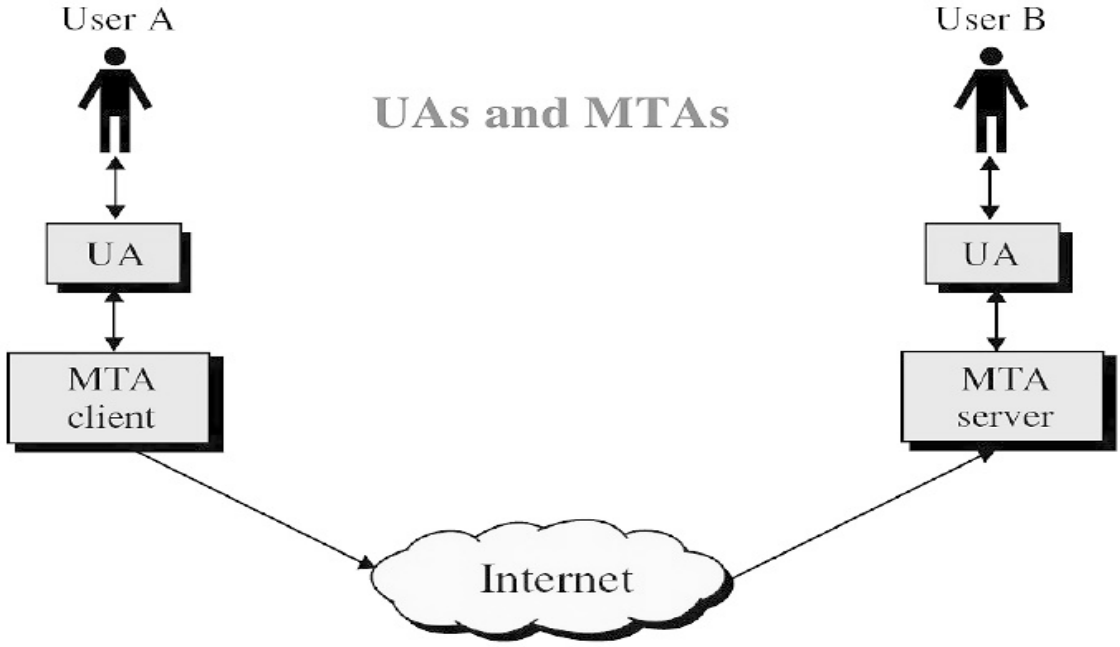
Figure 13.5 SMTP concept



Starting with this simple figure, we will examine the components of the SMTP system. gradually adding complexity. Let us begin by breaking down both the SMTP client and server into two components user agent (UA) and mail transfer agent (MTA).

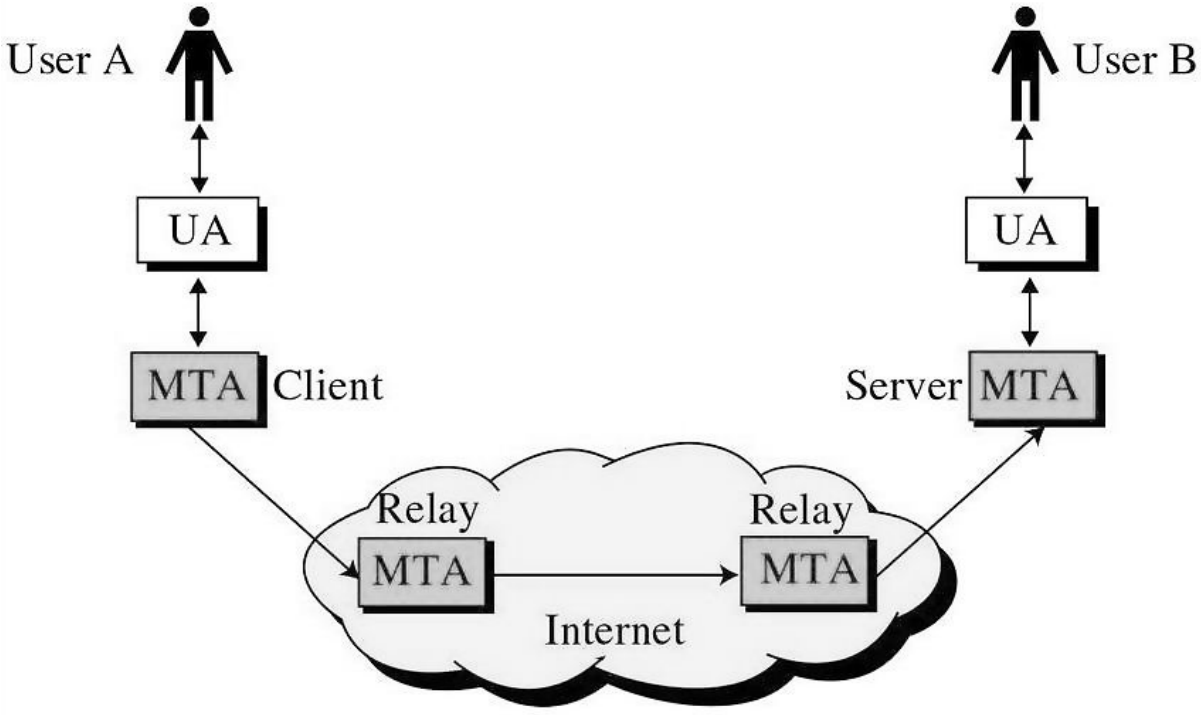
The UA prepares the message, creates the envelope, and puts the messages in the envelope. The MTA transfers the mail across the Internet. Show the previous figure with the addition of these two components. (Figure 13.6)

Figure 13.6 UAs and MTAs



SMTP protocol allows a more complex system than the one shown. Relaying could be involved. Instead of just one MTA at the sender site and one at the receiving site, other MTA acting either as client or server can relay the mail.(figure 13.7)

Figure 13.7 Relay MTAs



The relaying system allows sites that do not use the TCP/IP protocol suite to send email to users on other sites that may or may not use the TCP/IP protocol suite. This is accomplished through the use of a **mail gateway**. which is a relay MTA that can receive mail prepared by protocol other than SMTP and transform it to SMTP format before sending it. It can also receive mail in SMTP format and change it to another format before sending it.

User Agent (UA)

A user agent is defined in SMTP, but the implementation details are not. The UA is normally a program used to send and receive mail. Popular user agent programs are MH, Berkeley Mail, Elm, Zmail, and Mush.

Some user agents have an extra user interface that allows window-type interactions with the system.

Addresses

To deliver mail, a mail handling system must use a unique addressing system. The addressing system used by SMTP consists of two parts: a local part and a domain name, separated by an @ sign.

Local Part

The local part defines the name of a special file, called the user mailbox, where all of the mail received for a user is stored for retrieval by the user agent.

Domain Name

The second part of the address is the domain name. An organization usually selects one or more hosts to receive and send e-mail; they are sometimes called mail exchangers. The domain name assigned to each mail exchanger either comes from the DNS data-base or is a logical name (for example, the name of the organization).

Mail Transfer Agent (MTA)
The actual mail transfer is done through mail transfer agents (MTAs). To send .mail, a system must have a client MTA, and to receive mail, a system must have a server MTA. Although SMTP does not define a specific MTA, Send mail is a commonly used UNIX system MTA.

SMTP simply defines how commands and responses must be sent back and forth. Each network is free to choose a software package for implementation. . The user interface is a component that creates a user-friendly environment.

Multipurpose Internet Mail Extensions (MIME)
SMTP is a simple mail transfer protocol. Its simplicity, however, comes with a price. SMTP can send messages only in NVT seven-bit ASCII format. In other words, it has some limitations. For example, it cannot be 'used for languages that are not supported by seven-bit ASCII characters (such as French, German, Hebrew, Russian, Chinese, and Japanese). Also, it cannot be used to send binary files or to send video or audio data.

Multipurpose Internet Mail Extension (MIME) is a supplementary protocol that allows non-ASCII data to be sent through SMTP. MIME is not a mail protocol and can-not replace SMTP; it is only an extension to SMTP. MIME transforms non-ASCII data the sender site to NVT ASCII data and delivers it to the client SMTP to be sent Pugh the Internet. The server SMTP at the receiving side receives the NVT ASCII data and delivers it to MIME to be transformed back to the original data.

Post Office Protocol (POP)
SMTP expects the destination host, the mail server receiving the mail, to be on-line all the time; otherwise, a TCP connection cannot be established. For this reason, it is not practical to establish an SMTP session with a desktop computer because desktop computers are usually powered down at the end of the day.

In many organizations, mail is received by an SMTP server that is always on-line. This SMTP server provides a mail-drop service. The server receives the mail on behalf of every host in the organization. Workstations interact with the SMTP host to retrieve messages by using a client-server protocol such as Post Office Protocol (POP), version 3 (POP3).

Although POP3 is used to download messages from the server, the SMTP client is still needed on the desktop to forward messages from the workstation user to its SMTP mail server.

13.4 HYPERTEXT TRANSFER PROTOCOL (HTTP)

The **Hypertext Transfer Protocol (HTTP)** is a protocol used mainly to access data on the World Wide Web.. The protocol transfers data in the form of plain text, hypertext, audio, video, and so on. However, it is called the hypertext transfer protocol because its efficiency allows its use in a hypertext environment where there are rapid jumps from one document to another.

HTTP functions like a combination of FTP and SMTP. It is similar to FTP because it transfers files and uses the services of TCP. However, it is much simpler than FTP because it uses only one TCP connection. There is not a separate control connection; only data are transferred between the client and the server.

HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages. In addition, the format of the messages is controlled by MIME-like headers. However, HTTP differs from SMTP in the way the messages are sent from the client to the server and from the server to the client. Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser). SMTP messages are stored and forwarded, but HTTP messages are delivered immediately.

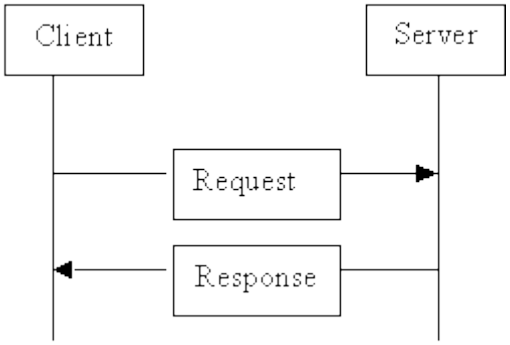
The idea of HTTP is very simple. A client sends a request, which looks like mail, to the server. The server sends the response, which looks like a mail reply, to the client. The request and response messages carry data in the form of a letter with MIME-like format.

The commands from the client to the server are embedded in a letter like request message. The contents of the requested file or other information are embedded in a letter like response message.

HTTP Transaction

Figure13.8 illustrates the HTTP transaction between the client and server. The client initializes the transaction by sending a request message. The server replies by sending a response.

Figure13.8 HTTP transaction



Messages

There are two general types of HTTP message request and response. Both message types follow almost the same format.

Request Messages

A request message consists of a request line, headers, and sometimes a body.

Response Message

A response message consists of a status line, headers, and sometimes a body.

Uniform Resource Locator (URL)

A client that wants to access a document needs an address. To facilitate the access of documents distributed throughout the world, HTTP uses the concept of locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: method, host computer, port, and path..

The method is the protocol used to retrieve the document, for example HTTP. The host is the computer where the information is located, although the name of the computer can be an alias. Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters "www." This is not mandatory, however, as the host can be any name given to the computer that hosts the web page.

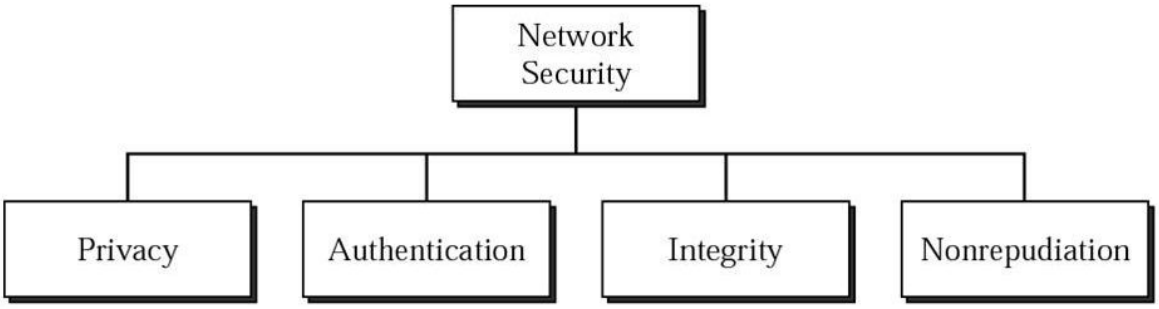
The URL optionally can contain the port number of the server. If the port is included, it should be inserted between the host and the path, and it should be separated from the host by a colon.

14. NETWORK SECURITY

14.1 FOUR ASPECTS of SECURITY

Based on the above expectations, we can say that security involves four aspects: privacy (confidentiality), message authentication, message integrity, and non-repudiation (Figure 14.1).

Figure 14.1 Aspects of Security



Privacy
Privacy means that the sender and the receiver expect confidentiality. The transmitted message should make sense to only the intended receiver. To all others, the message should be unintelligible.

Authentication
Authentication means that the receiver is sure of the sender's identity and that an imposter has not sent the message.

Integrity
Data integrity means that the data must arrive at the receiver exactly as it was sent. There must be no changes during the transmission, either accidental or malicious. As more and more monetary exchanges occur over the Internet, integrity is crucial. For example, it would be disastrous if a request for transferring $100 changes to a request for $10,000 or $100,000. The integrity .of the message must be preserved in a secure communication.

Non-Repudiation
Non-repudiation means that a receiver must be able to prove that a received message came from a specific sender. The sender must not be able to deny sending a message that he, in fact, did send. The burden of proof falls on the receiver.. For example, when a customer sends a message to transfer money from one account to another, the bank must have proof that the customer actually requested this transaction.

14.2 DIGITAL SIGNATURE

We said that security has four aspects: privacy, authentication, integrity, and non-repudiation. We have already discussed privacy. The other three can be achieved using what is called digital signature.

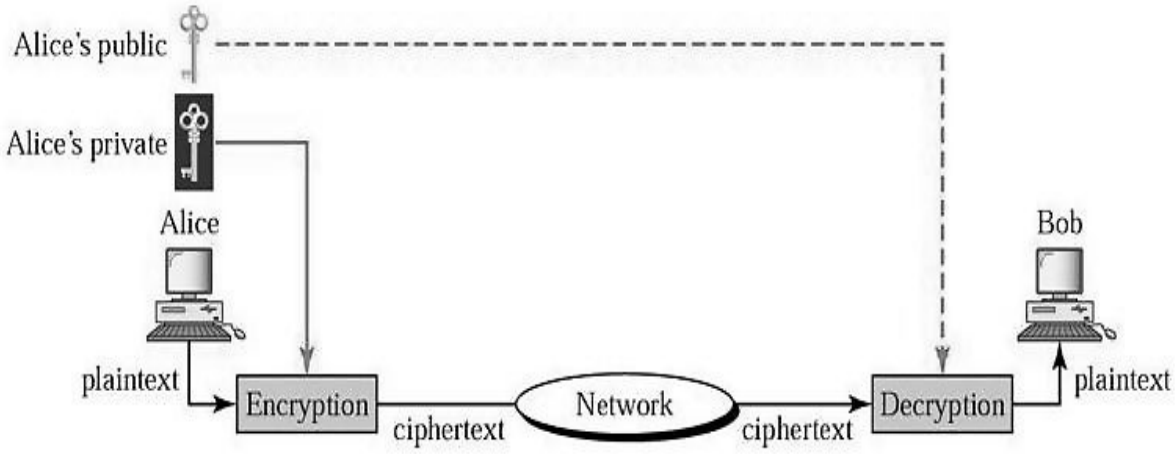
The idea is similar to the signing of a document. When we send a document electronically, we can also sign it. We have two choices: we can sign the entire document or we can sign a digest (condensed version) of the document.

Signing the Whole Document

Public key encryption can be used to sign a document. However, the roles of the public and private key are different here. The sender uses her private key to encrypt (sign) the message just as a person uses her signature (which is private in the sense that it is difficult to forge) to sign a paper document. The receiver, on the other hand, uses the public key of the sender to decrypt the message just as a person verifies from memory another person's. signature

. In digital signature the private key is used for encryption and the public key for decryption. This is possible because the encryption and decryption algorithms used today. such as RSA, are mathematical formulas and their structures are similar. Figure 14.2 shows how this is done.

Figure 14.2 Signing the whole document



Digital signature cannot be achieved using secret key encryption.
Digital signature can provide integrity, authentication, and non-repudiation.

Integrity

The integrity of a message is preserved because, if an intruder intercepts the message and partially or totally changes it, the decrypted message would be (with a high probability) unreadable.

Authentication

We can use the following reasoning to show how a message can be authenticated. If an intruder (user X) sends a message pretending that it is coming from someone else (user G), she must use her own private key (private X) for encryption. The message is then decrypted with the public key of user G and will therefore be non-readable. Encryption with X's private key and decryption with G's public key results in garbage.

Non-Repudiation

Digital signature also provides for non-repudiation. If the sender denies sending the message, her private key corresponding to her public key can be tested on the original plaintext. If the result of decryption matches the original message then we know the sender sent the message.

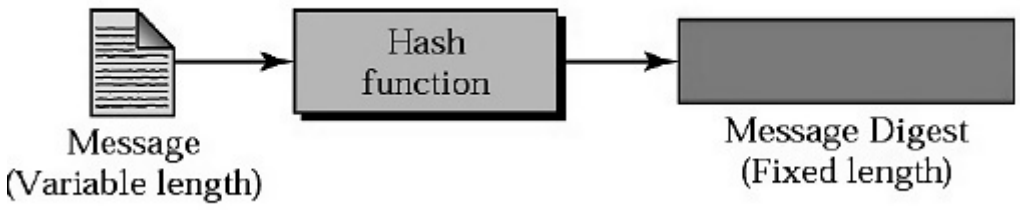
Digital signature does not provide privacy. If there is a need for privacy, another layer of encryption/decryption must be applied.

Signing the Digest

The public key encryption is efficient if the message is short. Using a public key to sign the entire message is very inefficient if the message is very long. The solution is to let the sender sign a digest of the document instead of the whole document. The sender creates a miniature version of the document and signs it; the receiver then checks the signature on the miniature.

To create a digest of the message, we use a hash function. The hash function creates a fixed-size digest from a variable-length message as shown in Figure 14.3

Figure 14.3 Signing the digest

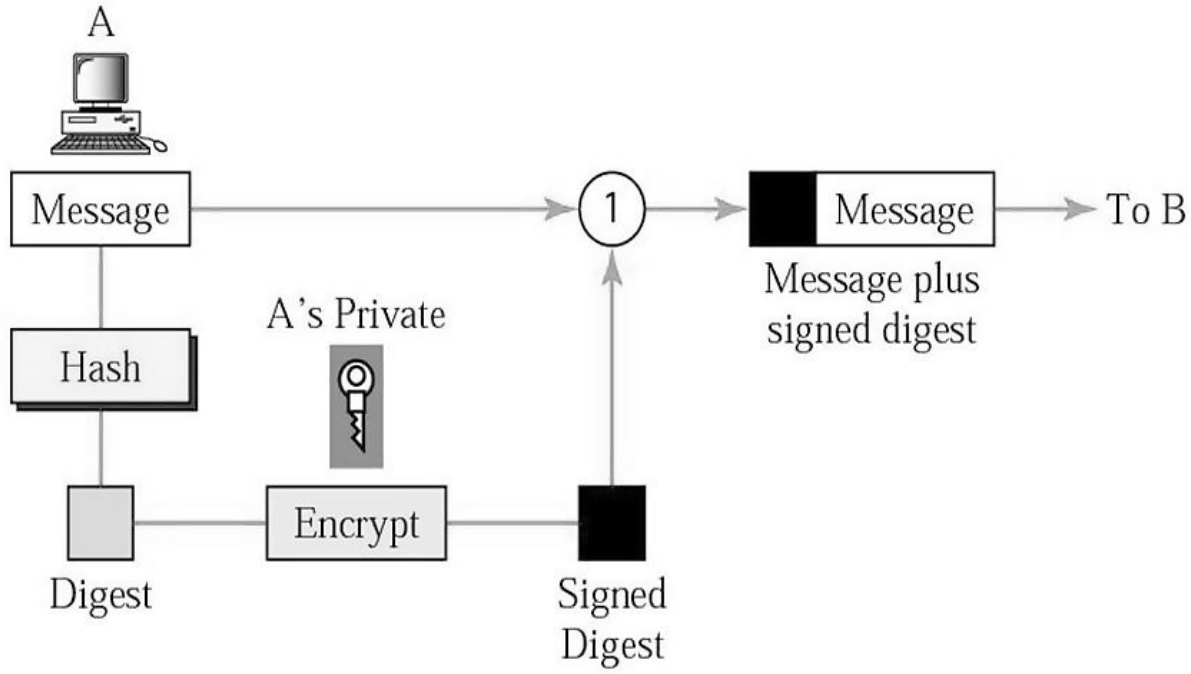


The two most common hash functions are called MD5 (Message Digest 5) and SHA-1 (Secure Hash Algorithm 1). The first one produces a 120-bit digest. The second produces a 160-bit digest.

Note that a hash function must have two properties to guarantee its success. First, hashing is one-way; the digest can only be created from the message, but not vice versa. Second, hashing is a one-to-one function; there is little probability that two messages will create the same digest. We will see the reason for this condition shortly.

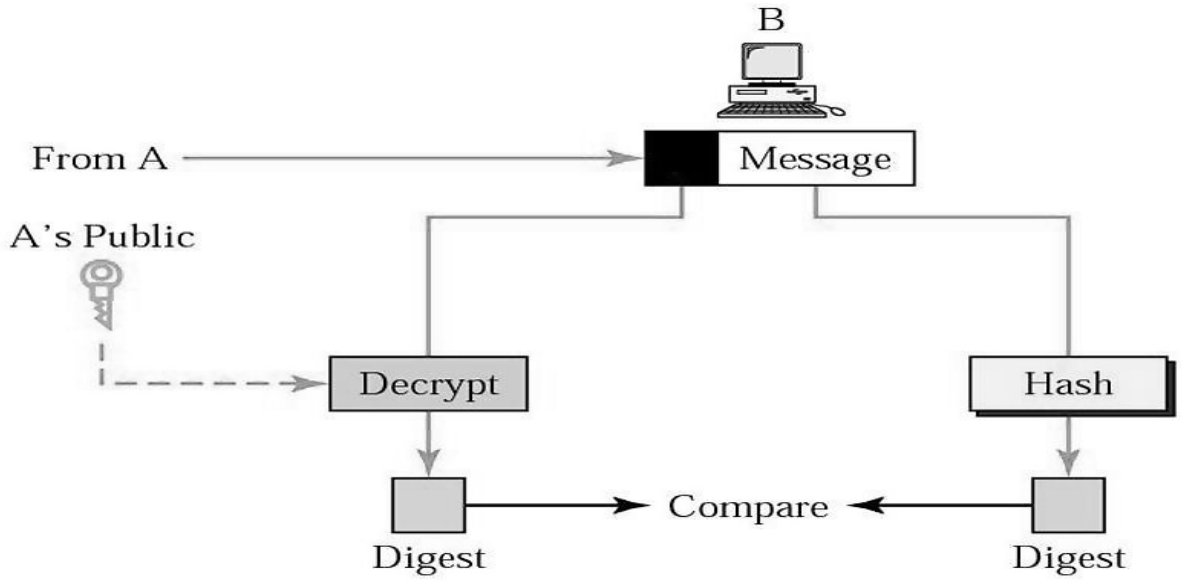
After the digest has been created, it is encrypted (signed) using the sender's private key. The encrypted digest is attached to the original message and sent to the receiver. Figure 14.4 shows the sender site.

Figure 14.4 Sender site



The receiver the original message and the encrypted digest and separates the two. The receiver applies the same hash function to the message to create a second digest. The receiver also decrypts the received digest using the public key of the sender. If the two digests are the same, all three aspects of security are preserved. Figure 14.5 shows the receiver site.

Figure 14.5 Receiver site



According to the previous section, we know that the digest is secure in terms a integrity, authentication, and non-repudiation; but what about the message itself? The following reasoning shows that these aspects are indeed provided for the message too:

1. The digest has not been changed (integrity) and the digest is a replica of the message. So the message has not been changed (remember no two messages concrete the same digest). Integrity has been provided.

2. The digest comes from the true sender, so the message also comes from the transgender. If an intruder had initiated the message, the message would not have created the same digest (no two messages can create the same digest).

3. The sender cannot deny the message since it is not possible to deny the digest; the only message that can create that digest is the received message.

14.3 PGP

For the good secure system let us discuss a very common security scheme ailed **Pretty Good Privacy (PGP)**, invented by Phil Zimmermann. PGP was designed provide all four aspects of security (privacy, integrity, authentication, and non-repudiation in the sending of e-mail.

PGP uses digital signature (a combination of hashing and public key encryption) t1 provide integrity, authentication, and non-repudiation. It uses a combination of secret key and public key encryption to provide privacy. In other words, it uses one hash function, one secret key, and two private—public key pairs.

Figure 14.5 shows how PGP creates secure e-mail at the sender site. The e-mail message is hashed to create a digest. The digest is encrypted (signed) using A's private key - The message and the digest are encrypted using the one-time secret key created by A. The secret key is encrypted using B's public key and sent together with the encrypted combination of message and digest.

Figure 14.5 PGP at the sender site

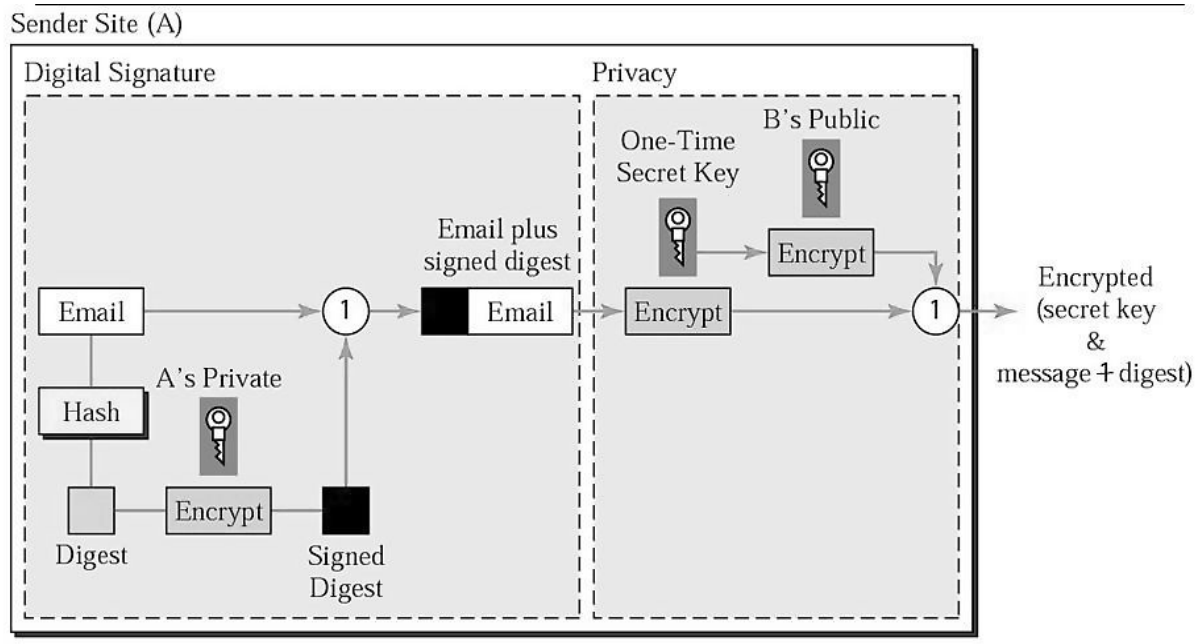
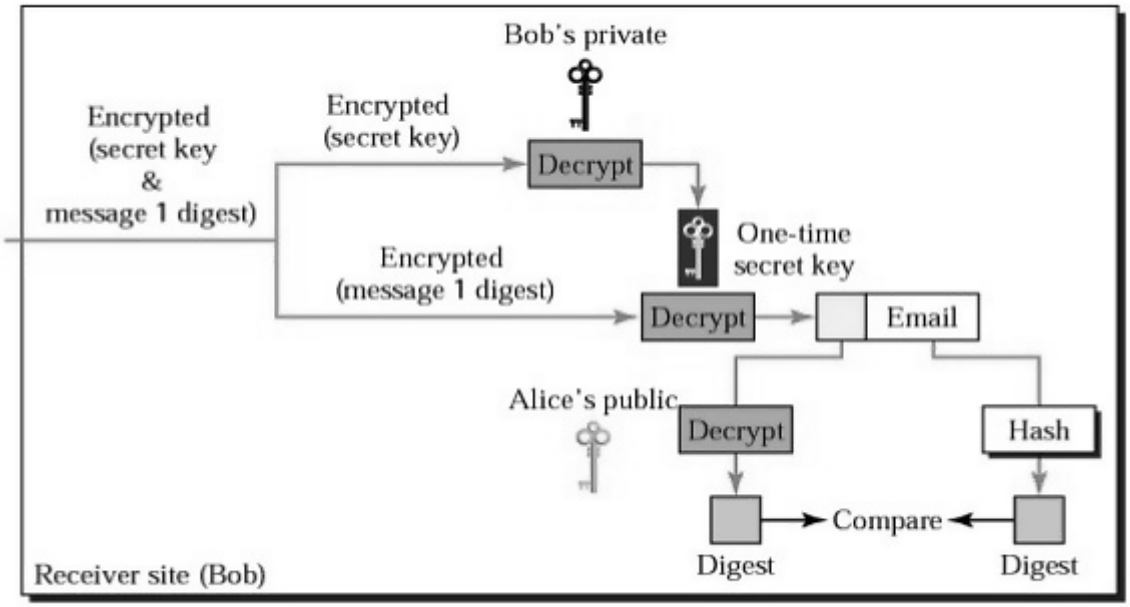


Figure 14.6 shows how PGP uses hashing and a combination of three keys to extract the original message at the receiver site.

Figure 14.6 PGP at the receiver site



The combination of encrypted secret key and message plus digest is received. The encrypted secret key first is decrypted (using B's private key) to get the one-time secret key

created by the sender. The secret key is then used to decrypt the combination of the message plus digest. The rest is the same as Figure 27.8 in the previous section.

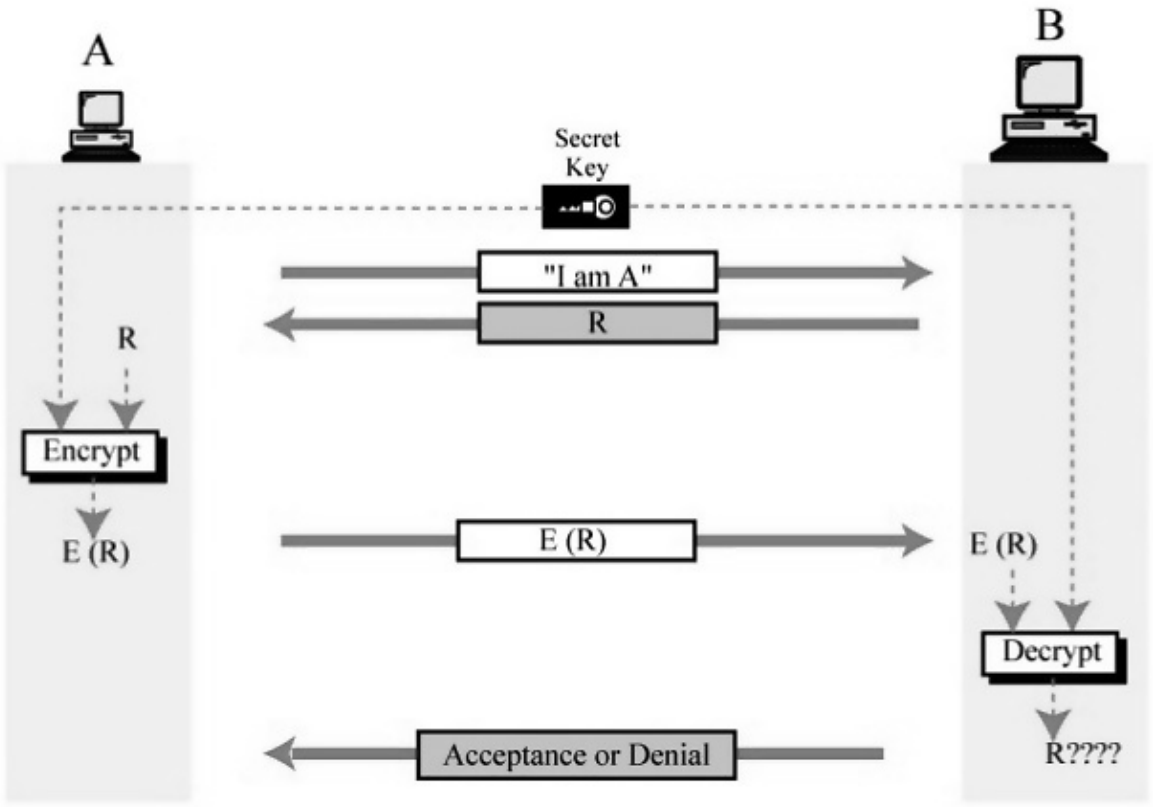
14.4 ACCESS AUTHORIZATION

Access authorization, another security measure, is often confused with authentication. In authentication, we have received a message and we need to be sure that the message was sent by the true sender: In access (or user) authorization, we first check the identity of the sender before access to the system is allowed.

User Authorization with Secret Key Encryption

User authorization with secret key encryption uses a procedure similar to those that date from the pre-computer era. Basically, the person who wants access authorization must know a secret key. Figure 14.7 shows how a secret key can be used for authorization.

. Figure 14.7 Access authorization with secret key encryption



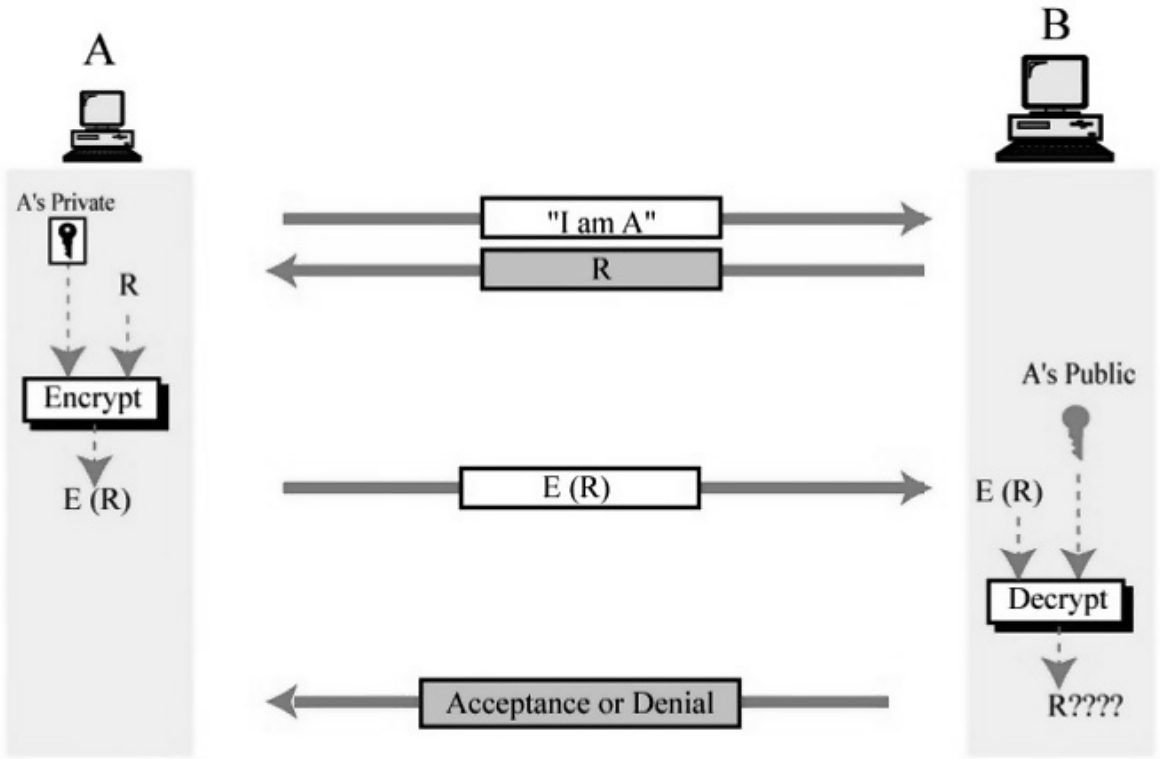
In the figure, user A wants to be authorized by user B (the system server). She sends a message to user B introducing herself. User B sends a number R, called the nonce (number once) to challenge user A. User A encrypts the number using the secret key and sends the encrypted number E(R) to user B. User B decrypts the message using the same secret key. If user B gets the original number, she authorizes the connection; otherwise, she denies it. In this type of authorization, a very important point is the selection of the number R (nonce). The number should be used just this once. If it needs to be used again, it should be in the

distant future. The reason is to prevent a playback attack. In such a situation, user A sends E(R) to user B. An intruder can listen to the line and make a copy of the packet containing E(R). If user B in the next challenge uses the same number, the intruder can send a copy of the packet containing E(R) and fool B into believing that the packet is sent by A.

User Authorization with Public Key Encryption

User authorization can also b w achieved using public key encryption as shown in Figure 14.8. In this case the public and private keys of the requesting party are used in the process.

Figure 14.8 Access authorization with public key encryption



In the figure, user A wants to be authorized by user B. She sends a message to user B introducing herself. User A encrypts a nonce with A's private key and sends it to B. User B decrypts the number using A's public key. If this decrypted number is the same as the nonce, user A is authorized; otherwise, access is denied.

MODEL QUESTION PAPER
COMPUTER NETWORKS

Max.Marks: 75
Time : 3 Hrs

PART A - (10X2=20 marks)
(Answer ALL the questions)

1. Write the components of data communication.
2. What is point to point and multipoint configuration?
3. Define Analog signal.
4. What is Unipolar encoding?
5. What is burst error?
6. Define flow control.
7. What are the two classes of synchronous protocols.
8. Write the three methods of switching.
9. Explain encapsulation in networking.
10. Write the aspects of security.

PART B – (5x5=25 marks)
(Answer ALL the questions)

11.a. Explain Distributed processing.

or

b. Write short notes on categories of networks.

12.a. Describe Analog and Digital signals.

or

b. Explain Digital to Analog conversion.

13.a. Briefly write about CRC.

or

b. Describe error control in the data link layer.

14. a. Write short notes on character oriented protocols.

or

b. Briefly write about token ring.

15.a. Describe addressing in TCP/IP protocol.

Or

b. What is TELNET? Explain.

PART C – (3x10=30 marks)
(Answer any **THREE** of the following)

16. Describe topologies of networking.
17. Explain OSI model and the functions of the layers.
18. Explain about Unguided media in detail.
19. Describe Circuit Switching.
20. Describe Network security with digital signature.