# VEER SURENDRA SAI UNIVERSITY OF TECHNOLOGY BURLA – 768018

## LECTURE NOTES

## ON

## COMPUTER NETWORKS

## CODE: MCA -210

By

Asst. Prof. Mrs. Etuari Oram

# DISCLAIMER

THIS DOCUMENT DOES NOT CLAIM ANY ORIGINALITY AND CANNOT BE USED AS A SUBSTITUTE FOR PRESCRIBED TEXTBOOKS. THE INFORMATION PRESENTED HERE IS MERELY A COLLECTION BY THE COMMITTEE MEMBERS FOR THEIR RESPECTIVE TEACHING ASSIGNMENTS. VARIOUS TEXT BOOKS AS WELL AS FREELY AVAILABLE MATERIAL FROM INTERNET WERE CONSULTED FOR PREPARING THIS DOCUMENT. THE OWNERSHIP OF THE INFORMATION LIES WITH THE RESPECTIVE AUTHORS OR INSTITUTIONS.

# SYLLABUS

## Module I:

Overview of Data Communications and Networking
Physical Layer : Analog and Digital, Analog Signals, Digital Signals, Analog versus Digital, Data Rate Limits, Transmission Impairment, More about signals.
Digital Transmission: Line coding, Block coding, Sampling, Transmission mode.
Analog Transmission: Modulation of Digital Data; Telephone modems, modulation of Analog signals.
Multiplexing: FDM 150, WDM 155, TDM 157
Transmission Media: Guided Media, Unguided media (wireless)
Circuit switching and Telephone Network: Circuit switching, Telephone network.

## Module II:

Data Link Layer: Error Detection and correction: Type of Errors, Detection, Error Correction. Data Link control and protocols: Flow and error Control, Stop-and-wait ARQ, Go-Back N ARQ, Selective Repeat ARQ, HDLC.
Point-to-Point Access: PPP
Point-to Point Protocol, PPP Stack
Multiple Accesses: Random Access, Controlled Access, Channelization.
Local area Network: Ethernet
Traditional Ethernet, Fast Ethernet, Gigabit Ethernet
Wireless LANs: IEEE 802. 11, Bluetooth virtual circuit: Frame Relay and ATM

## Module III:

Network Layer: Host to Host Delivery: Internetworking, addressing and Routing
Network Layer Protocols: ARP, IPVA, ICMP, IPV6 ad ICMPR6
Transport Layer; Process to process Delivery: UDP; TCP congestion control and Quality of service.

## Module IV:

Application Layer:
Client Server Model, Socket Interface Domain Name System (DNS):
Electronic Mail (SMTP) and file transfer (FTP) HTTP and WWW
Security: Cryptography, Message security, User Authentication

## Text Books:


1. Data Communication and Networking – B.A. Forouzan, TMH.
2. Computer Networks: Third Edition, A systems Approach, Larry L/Peterson & Bruce S. Davie ELSEVIER.
3. Computer Networks, A.S. Tannenbaum PHI.
4. Data and Computer Communications, William Stallings, 5th Edition, PHI.
5. Data Communications and Computer Networks – by P.C.Gupta, PHI.

# CONTENTS

Lecture 17: Ethernet: Frame length, Addressing, Access method

Lecture 18: Ethernet: 10 Base2-Thin Ethernet, 10 Base T- Twisted pair Ethernet, 10 base-F- Fiber Ethernet, Fast Ethernet, MAC Sub-layer

Lecture 19: Encoding, Giga bit Ethernet

Lecture 20: Giga bit Ethernet implementation, Wireless Communication

Lecture 21: Bluetooth

Lecture22: Introduction to Frame relay & ATM, Description of Frame relay Layers.

Lecture 23: ATM: Introduction, Benefits of ATM, ATM Devices & Network Environment, ATM Network Interfaces.

**Module 3:**

Lecture 24: Network layers: Objective, logical addressing, IPv4 addresses, IPv6 Addresses

Lecture 25: Internetworking

Lecture 26: Routing, Classification of Routing Algorithms, Delta Routing, and Multipath routing

Lecture 27: Hierarchical Routing, Routing Algorithm, Source Routing, Policy Based routing, Shortest path routing Dijkstr's Algorithm, The Floyd Warshal Algorithm: Description, Principles, Services

Lecture 28: Addressing, Mapping, Error Reporting and Multicasting

Lecture 29: Bootstrap Protocol Loose service route, IPv6: Packet format, Base Header, Next header, Destination

Lecture 30: ICMP: Redirection, Query messages, Encapsulation of ICMP query Messages

Lecture 31: IPv6: Priority, Congestion Controlled Traffic, Control Traffic, Flow label, Extension headers, Hop-by-Hop Option, Fragmentation, Authentication, Encrypted, Security payload

Lecture 32: Transport Layer, Process To Process Delivery Introduction, Addressing, Multiplexing & De- Multiplexing User data gram Protocol (UDP): User Datagram Source port number, Destination Port no.

Lecture 33: UDP operation, Flow& error Control, Queuing, Use of UDP, and TCP: Full Duplex Communication

Lecture 34: TCP: Flow control, error control, Congestion Control, Segment

Lecture 35: TCP Connection

Lecture 36: TCP Congestion Control, Quality of Service, Scheduling, FIFO Queuing, Priority queuing, Weighted fair queuing

**Module 4:**

Lecture 37: Internet Application

Lecture 38: FTP (file transfer protocol)

Lecture 39: WWW and HTTP

Lecture 40: Encryption, Cryptography, Message Security, User Authentication

***MODULE-I***

***LECTURE NOTE: 1***

# OVERVIEW DATA COMMUNICATIONS AND NETWORKING

## DATA COMMUNICATIONS

- Data communications are the exchange of data between two devices via some form of transmission medium such as a wired or wireless.
- For data communications to occur, the communicating devices must be part of a communication system made up of combination of hardware and software.
- The effectiveness of a data communications system depends on four fundamental characteristics: **delivery, accuracy and timeliness**.

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

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

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

## COMPONENTS

A data communications system has five components as follows;

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

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

3. **Receiver**. The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
4. **Transmission medium**. The transmission medium is the physical path/air by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.
5. **Protocol**. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. It defines **how** it is communicated, **when** communicated and **what** communicated. Key elements of protocols are **timing, syntax** and **semantics.** Without a protocol, two devices may be connected but not communicating.

## DATA REPRESENTATION

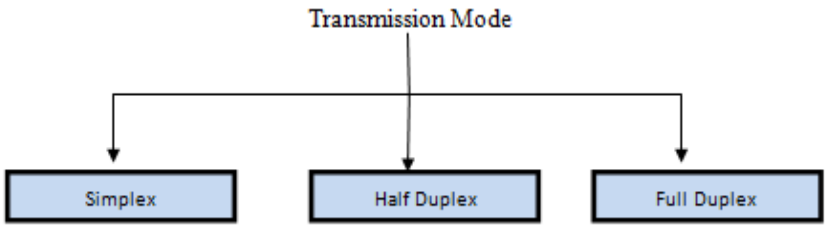
Information today comes in different forms such as **text, numbers, images, audio, and video.**

**1. Text:** In data communications, text is represented as a bit pattern, a sequence of bits (0s or 1s).
**2. Numbers:** Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent number; the number is directly converted to a binary number to simplify mathematical operations.
**3. Images:** Images are also represented by bit patterns. In its simplest form, an image is composed of matrix of pixels (picture elements), where each pixel is a small dot.
**4. Audio:** Audio refers to the recording or broadcasting of sound or music.
**5. Video:** Video refers to the recording or broadcasting of a picture or movie.

## MODE OF DATA FLOW OR TRANSMISSION MODE

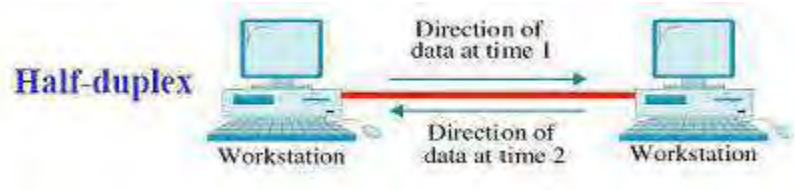
> The transmission mode is the direction of the exchanges where the number of bits sent simultaneously.

Communication between two devices can be a mode of **simplex**, **half-duplex**, or **full-duplex.**




- **Simplex:** In simplex mode, the communication is unidirectional, as on a one-way street. One of the simplest examples is monitor keyboard communication, where keyboard only introduce input and monitor can only display output, other examples are Television and remote, television broadcasting, radio broadcasting, loudspeaker etc.


- **Half-Duplex:** In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, other can only receive and vice versa. Half duplex mode is like one lane road with two direction traffic. In this mode entire capacity of channel is reserved or taken over by which the device transmitting at the time. Its examples are **walkie-talkies and citizen band radio** etc.




- **Full-Duplex**: In full-duplex mode both stations can transmit and receive simultaneously. The full-duplex mode is like a two way street with traffic flowing in both directions at the same time. Here signal sharing is either direction i.e. shares the capacity of link. Common example is the telephone network where both talk and listen possible at same time, same time channel is shared.


## NETWORK CRITERIA

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

- **Performance:** Performance can be measured in many ways, including transit time and response time.
- **Reliability:** In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.
- **Security**: Network security issues include protecting data from unauthorized access, protecting, from damage and development, and implementing policies and procedures for recovery from breaches and lost data.

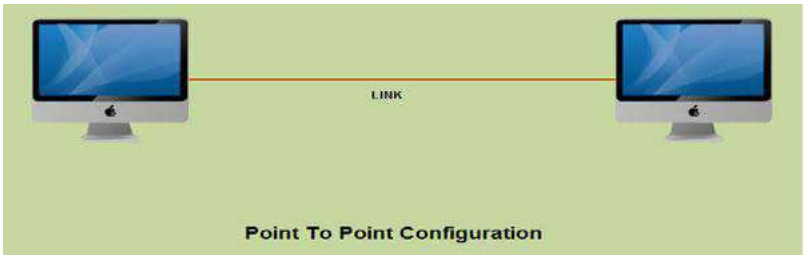
## PHYSICAL STRUCTURES

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

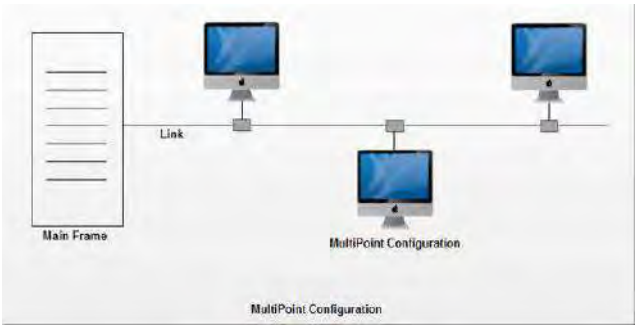
### Type of Connection

A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. There are two possible types of connections: **point-to-point and multipoint.**

1. **Point-to-Point:** A point-to-point connection provides a dedicated link between two devices.



2. **Multipoint:** A multipoint (also called multi-drop) connection is one in which more than two specific devices share a single link.

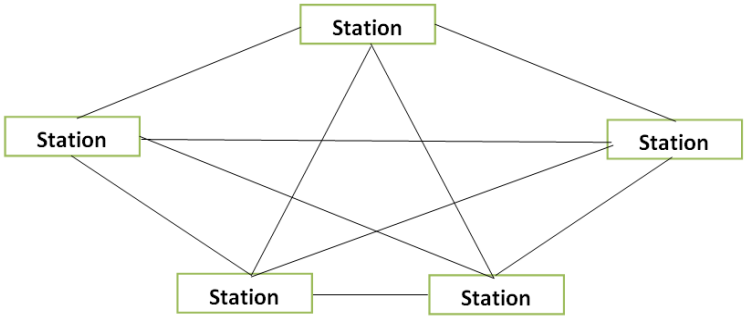



## Physical Topology

- The term physical topology refers to the way in which a network is laid out physically.
- The topology of a network is the geometric representation of the relationship of all the links and linking devices to one another.
- There are four basic topologies possible: **mesh, star, bus, and ring.**

## 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.
- To find the number of physical links in a fully connected mesh network with n nodes, we first consider that each node must be connected to every other node.
- However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2 i.e n(n-1)/2 physical channel to link n devices.
- To accommodate link every device in the network must have (n-1) input/ output ports.




A mesh offers several **advantages** over other network topologies.

- **Firs**t, the use of **dedicated links** guarantees that each connection can carry its own data load, thus eliminating the traffic problems that may occur when links are shared by multiple devices.
- **Second**, a mesh topology is **robust**. If one link becomes unusable, it does not incapacitate the entire system.
- **Third**, there is the advantage of **privacy or security**. Traffic can be routed to avoid links with suspected problems. This facility enables the network manager to discover the precise location of the fault and aids in finding its cause and solution.
- **Fourth,** point to point link make **fault identification** and **fault isolation** easy.

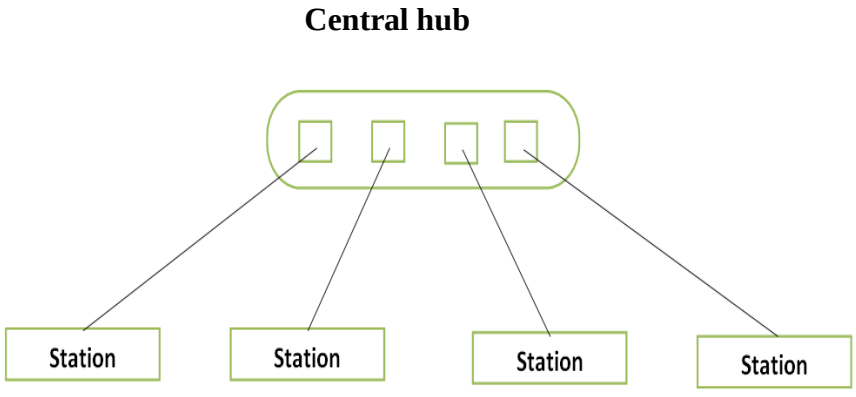
The main **disadvantages** of a mesh are the amount of cabling and the number of I/O ports required more in addition to that:

- Because every device must be connected to every other device, installation and reconnection are difficult.
- The sheer bulk of the wiring can be greater than the available space can accommodate.
- **Finally**, the hardware required to connect each link can be prohibitively expensive. For these reasons a mesh topology is usually implemented in a limited fashion.

# *LECTURE NOTE: 2*

## STAR TOPOLOGY

- In a star topology, each device has a dedicated point-to-point link only to a **central controller**, usually called a **hub**.
- The devices are not directly linked to one another.
- Unlike a mesh topology, a star topology does not allow direct traffic between devices.
- The controller acts as an **exchange**, If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device.
- 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 leads itself to easy **fault identification** and **fault isolation**. As long as the hub is working, it can be used to monitor link problems and bypass defective links.

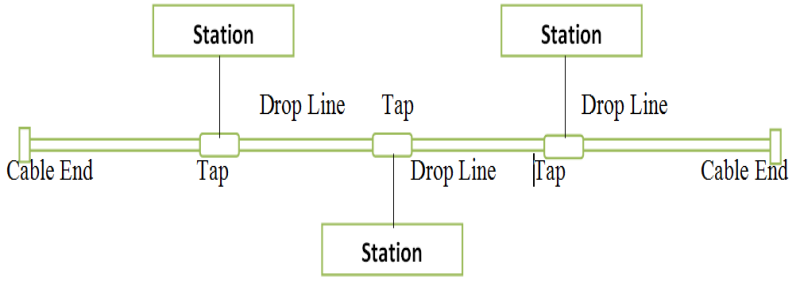



- One of the **big disadvantages** of a star topology is the dependency of the whole topology on one single point, the hub.

- If the **hub goes down**, the whole system is dead. Although a star requires far less cable than a mesh, each node must **be linked** to a central hub. For this reason, often more cabling is required in a star than in some other topologies.

## BUS TOPOLOGY

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




- **Advantages** of a bus topology include **ease of installation**.
- Bus uses less cabling than **mesh or star topologies**.
- In a star, for example, four network devices in the same room require four lengths of cable reaching all the way to the hub.
- In a bus, this redundancy is eliminated.
- Only the backbone cable stretches through the entire facility.
- Each drop line has to reach only as far as the nearest point on the backbone.

- **Its disadvantages** include difficult **reconnection** and **fault isolation**.

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

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

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

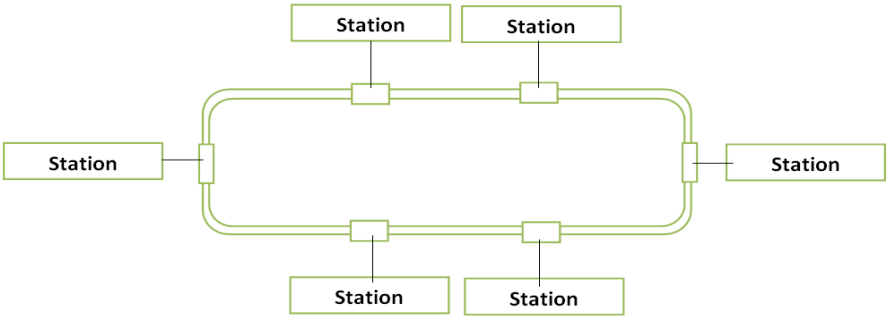
- In addition, **a fault or break in the bus cable stops all transmission**, even between devices on the same side of the problem.

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

**RING TOPOLOGY**

- In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it.
- A signal is passed along the ring in one direction, from device to device, until it reaches its destination.
- Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.
- A ring is relatively **easy to install and reconfigure**.
- Each device in ring is linked to only its immediate neighbors.
- To add or delete a device it requires changing only two connections.
- The only constraints here are media and traffic considerations. In addition, **fault isolation is simplified ring**. 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 issue used in ring 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 can disable the entire network**. This weakness can be solved by using a **dual ring or a switch** capable of closing off the break.

## COMPUTER NETWORKS

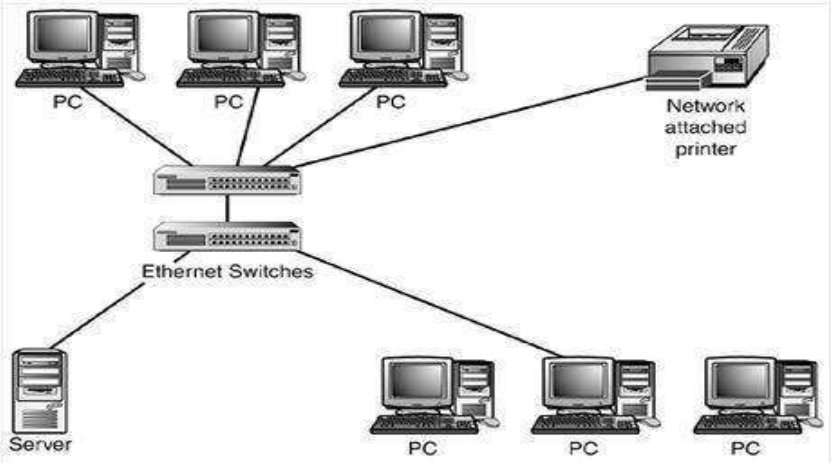
- A **network** is a set of devices/node connected by communication links.
- A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.
- Whereas **computer network** is a set of autonomous computer interconnected together through **communication medium** to facilitate communication between them.
- And through communication we can gain access resource sharing, file, program etc.
- Computer on network are called nodes. Connecting medium are either physical medium i.e. wire/cable or wireless medium (i.e. through radio waves).
- Best known computer network is Internet.

## CATEGORIES OF NETWORKS

- Today when we speak of networks, we are generally referring to two primary categories: local-area networks and wide-area networks.
- The categories into which a network falls is determined by its **size, scale, distances it cover, physical architecture, technology etc.**

## LOCAL AREA NETWORK (LAN)

- A LAN normally limited to few kilometer.
- In addition to operating in a limited space, LANs are also typically owned, controlled, and managed by a single person or organization.
- They also tend to use certain connectivity technologies, **Ethernet** and **Token** Ring.
- A local area network (LAN) is usually privately owned and links the devices within a single office, building, or campus.
- Depending on the needs of an organization and the type of technology used, a LAN can be a simple as two PCs and a printer in someone's home office, or it can extend throughout a company and include audio and video peripherals.
- LANs are designed to allow resources to be shared between personal computers or workstations. In addition to size, LANs are distinguished from other types of networks by their transmission media and topology.
- The most common LAN topologies are **bus, ring, and star.**

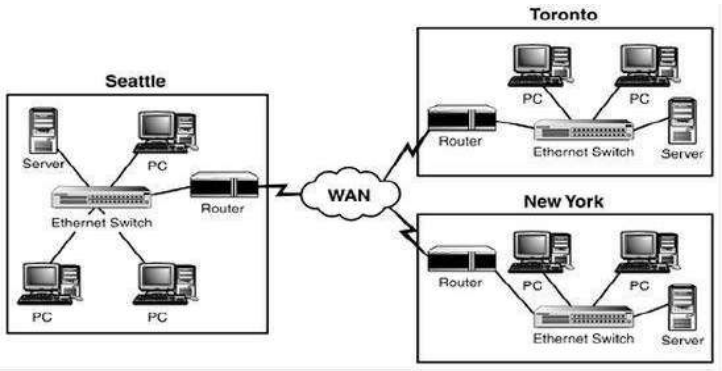


## Wireless Local Area Network


- It spans a large physical distance.
- The Internet is the largest WLAN, spanning the Earth.
- It is a geographically-dispersed collection of LANs.
- A network device called a router connects LANs to a WLAN.
- In IP networking, the router maintains both a LAN address and a WLAN address.
- A WLAN differs from a LAN in several important ways as, Most WANs (like the Internet) are not owned by any one organization but rather exist under collective or distributed ownership and management.
- WLANs tend to use technology like ATM, Frame Relay and X.25 for connectivity over the longer distances.

## WIDE AREA NETWORKS (WAN)

- A WAN can be worldwide.
- A WAN is a network that spans more than one geographical location often connecting separated LANs
- A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world.

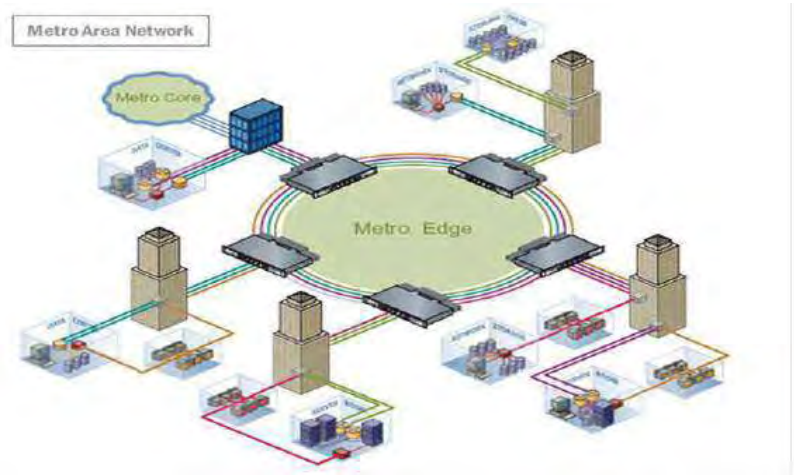



- A WAN can be as complex as the backbones that connect the Internet or as simple as a dial-up line that connects a home computer to the Internet.
- We normally refer to the first as a switched WAN and to the second as a point-to-point WAN.
- The point-to-point WAN is normally a line leased from a telephone or cable TV provider that connects a home computer or a small LAN to an Internet service provider.
- This type of WAN is often used to provide Internet access.
- WANs are slower than LANs and often require additional and costly hardware such as routers, dedicated leased lines, and complicated implementation procedures.

## METROPOLITAN AREA NETWORKS (MAN)

- A metropolitan area network (MAN) is a network with a size between a LAN and a WAN or it spans a physical area larger than a LAN but smaller than a WAN, such as a city.
- It normally covers the area inside a town or a city. A MAN is typically owned and operated by a single entity such as a government body or large corporation.
- It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city.

- A good example of a MAN is the part of the telephone company network that can provide a high-speed DSL line to the customer.




***LECTURE NOTE: 3***

# PROTOCOLS AND STANDARDS

The protocol is synonymous with rule and the standards which are agreed-upon rules.

## PROTOCOLS

- 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.

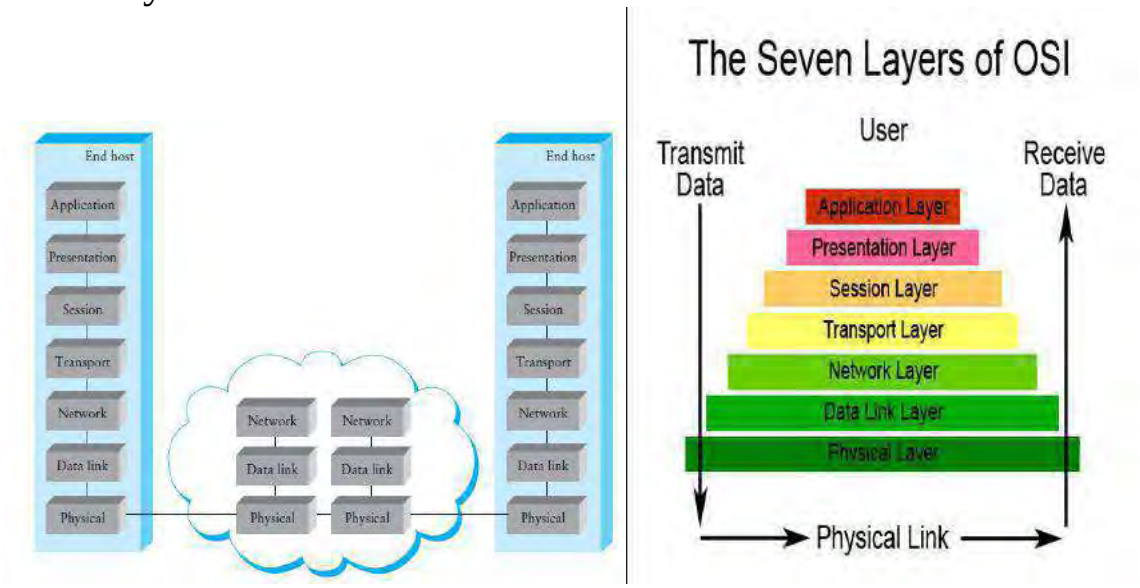
- Or it is a formal description of message formats and rules that computer must have to follow in order to exchange data
- The key elements of a protocol are **syntax, semantics, and timing.**

## 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.

## OSI MODEL

- The ISO Established in 1947 was one of the first organizations to formally define a common way to connect computers architecture, called the ***Open Systems Interconnection*** (OSI) architecture.
- The 7 layers OSI model was first introduced in the late 1970s.

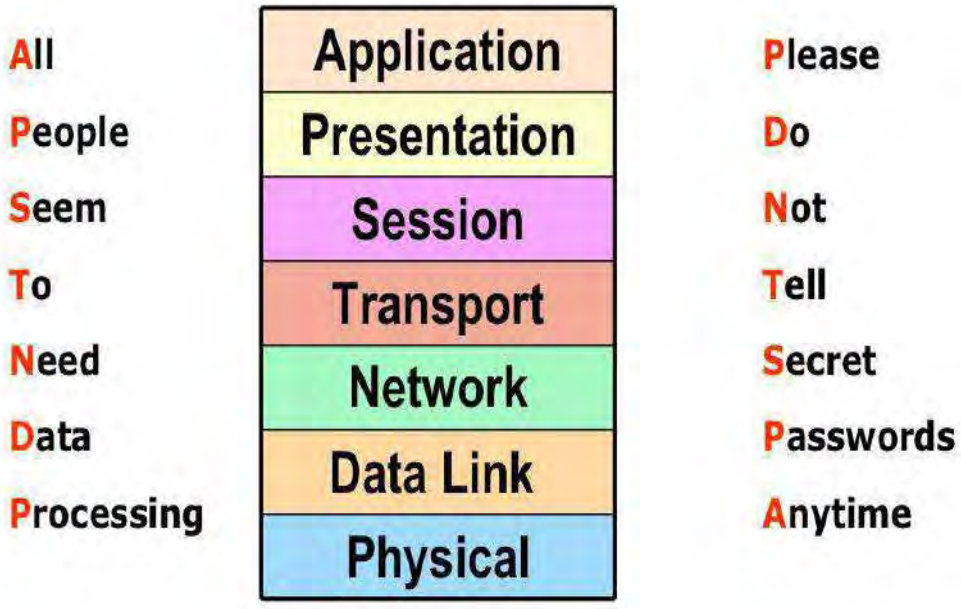



One or more node in the network

Figure: OSI network architecture

- This model defines a partitioning of network functionality into seven layers, where one or more protocols implement the functionality assigned to a given layer.
- It is a ***reference model*** for a protocol graph.

- The ISO, usually in conjunction with a second standards organization known as the International Telecommunications Union (ITU), it publishes a series of protocol specifications based on the OSI architecture.
- An ISO standard that covers all aspects of network communications is the **Open Systems Interconnection model**.
- An **open system** is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture.
- The purpose of the **OSI** model is to show how to facilitate communication between different systems and interoperable without requiring changes to the logic of the underlying hardware and software.
- The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is **flexible, robust, and interoperable**.
- The OSI model is a layered framework for the design of network systems that allows communication between all types of computer system.
- Some mnemonics of this model are:

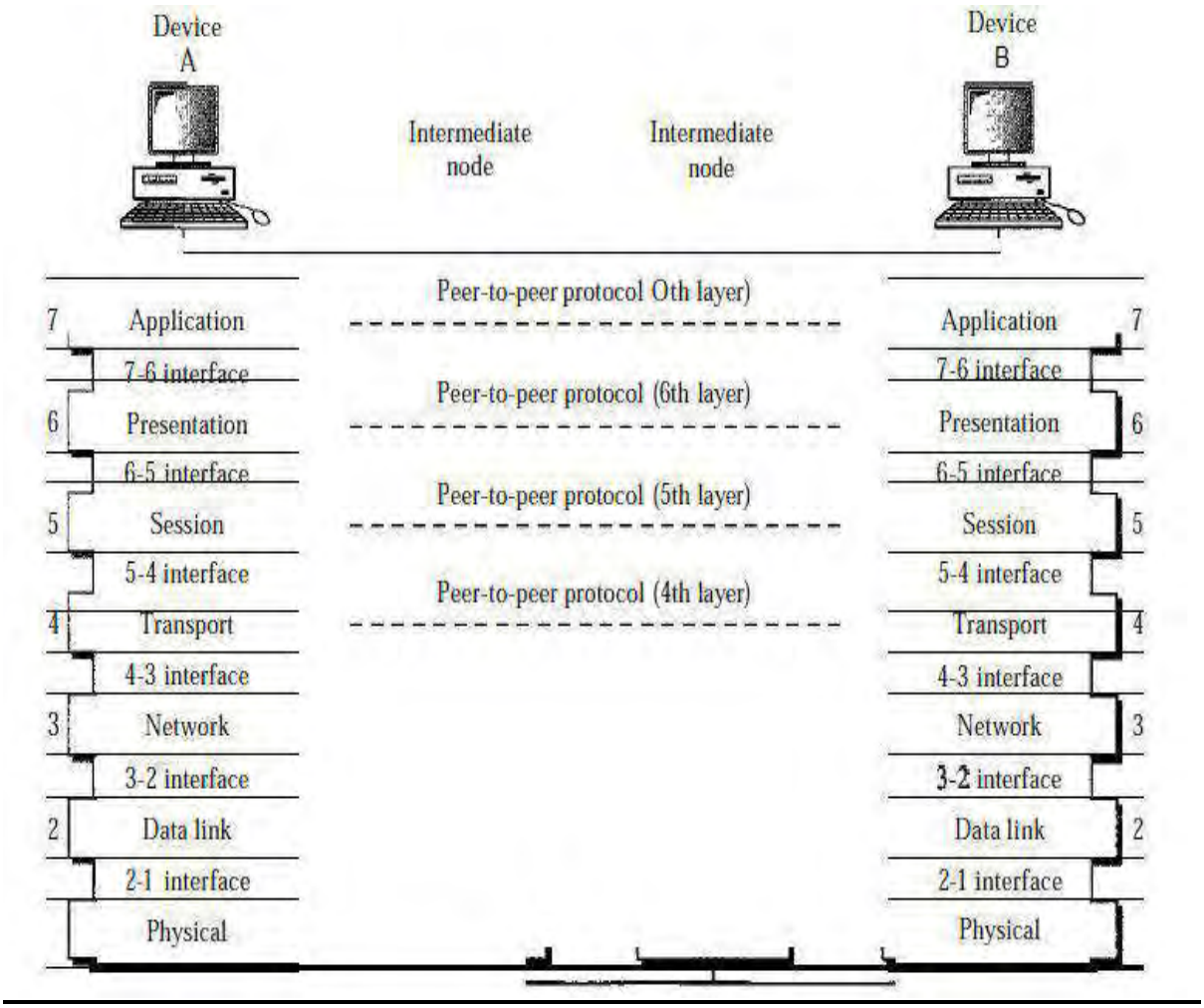



**DESCRIPTION**

A set of layers and protocols is called **network architecture.** It refers to the logical and physical design of a network.

- The **OSI model** is composed of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6) and application (layer 7).
- The above figure 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.
- 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 o*n one machine communicates with layer *x o*n another machine.
- This communication is governed by an agreed-upon series of **rules and conventions** called protocols.
- The processes on each machine that communicate at a given layer are called peer-to-peer processes. Communication between machines is therefore a peer-to-peer process using the protocols appropriate to a given layer.

**Interaction between the layers in OSI model**



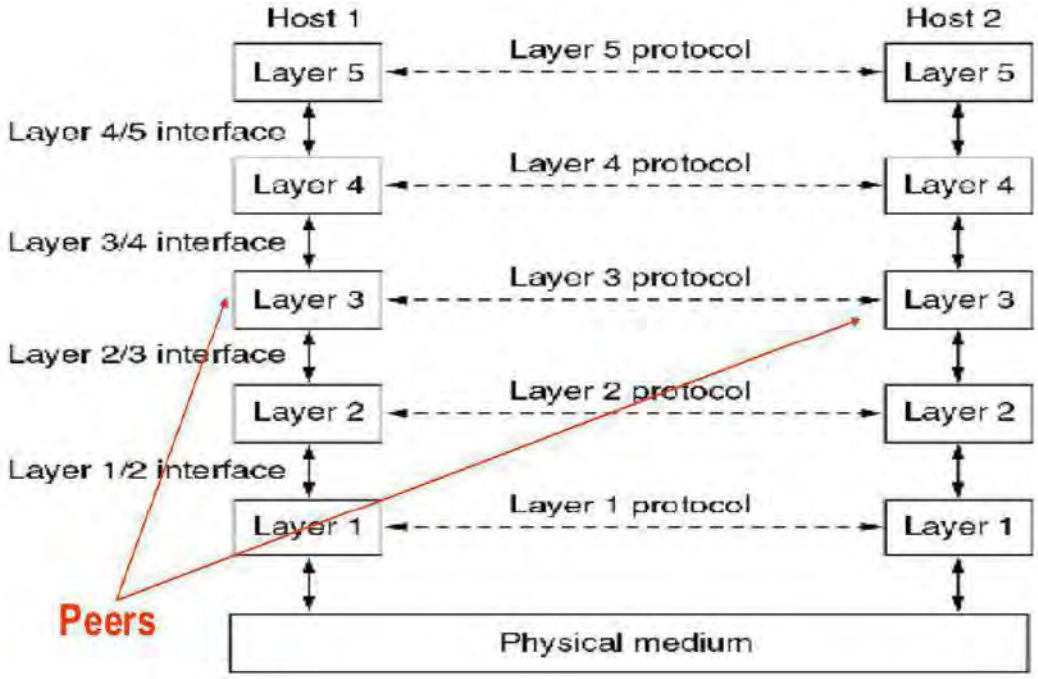

Physical communication

- At the **physical layer**, communication is direct, from the figure, device A sends a stream of bits to device B (through intermediate nodes).
- At the higher layers, however, communication must move down through the layers on device A, over to device B, and then back up through the layers.
- Each layer in the sending device 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.
- At layer 1 the entire package is converted to a form that can be transmitted to the receiving device.
- At the receiving machine, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it. For example, layer 2 removes the data meant for it, and then passes the rest to layer 3.

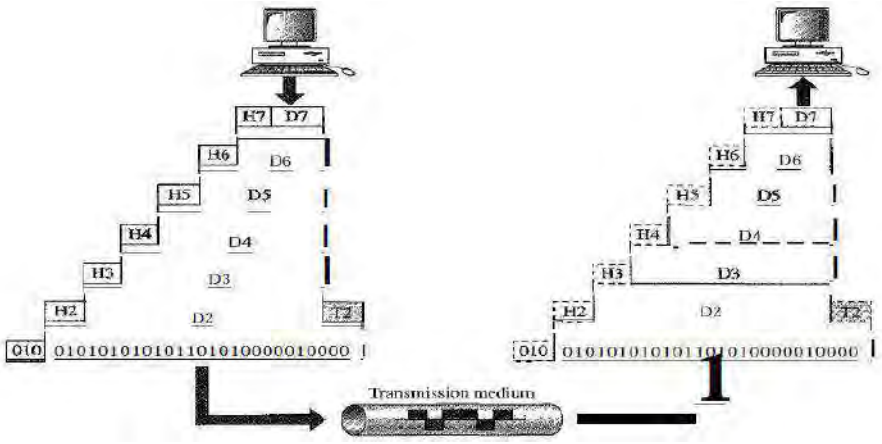
Layer 3 then removes the data meant for it and passes the rest to layer 4, and so on.

## INTERFACE BETWEEN THE LAYERS

An **interface** between each pair of adjacent layer made possible of passing data and information down through the layer of sending device and back through the layer of receiving device. **Interface** defines information as well as services (define what operation ) that must be provided by the layer, to the layer above it.



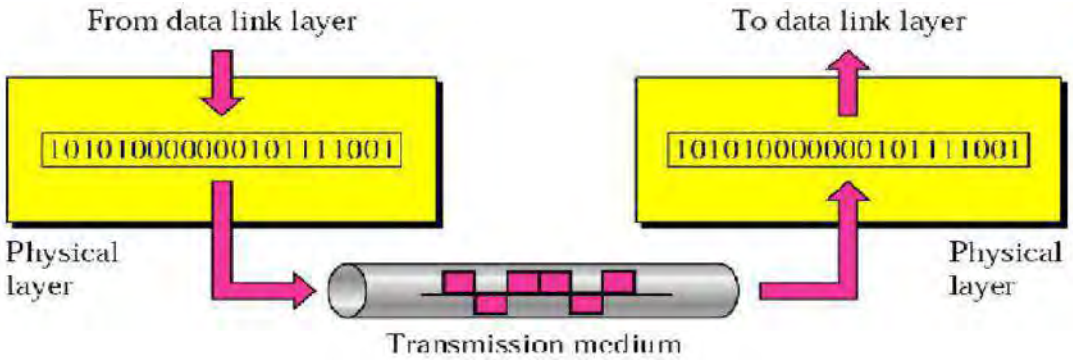

**Organization of the Layers**



- 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 etc)**.
- Layers 5, 6, and 7-session, presentation, and application-can be thought of as the **user support layers**; they allow interoperability among unrelated software systems.
- Layer 4, the transport layer, links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use.
- D7 means the data unit at layer 7, D6 means the data unit at layer 6, and so on.
- The process starts at layer 7 (the application layer), then moves from layer to layer in descending, sequential order.
- At each layer, a **header,** or possibly a **trailer,** can be added to the data unit. Commonly, the **trailer** is added only at layer 2.
- When the formatted data unit passes through the physical layer (layer 1), it is changed into an electromagnetic signal and transported along a physical link.
- Upon reaching its destination, the signal passes into layer 1 and is transformed back into digital form.

## FUNCTION OF EACH LAYER IN OSI MODEL

### PHYSICAL LAYER




- It deals with the mechanical and electrical specifications of the interface and transmission medium.

- Physical layer that, actually interacts with the transmission media and the physical part of the network that connects network components together.

- This layer is involved in carrying information from one node in the network to the next.
- One major task of this layer is to provide services for the data link layer.
- The data in the data link layer consists of 0s and 1s organized into frames that are ready to be sent across the transmission medium. This stream of 0s and 1s must first be converted into signals. **Major Responsibilities** of physical layers are:

---

The physical layer is responsible for movements of individual bits from one hop (node) to the next.

---

- **Physical characteristics of interfaces and medium:** The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.

- **Representation of bits:** The physical layer data consists of a stream of bits (sequence of 0s or 1s) with no interpretation. To be transmitted, bits must be encoded into signals i.e. electrical or optical. The physical layer defines the type of encoding (how 0s and 1s are changed to signals).

- **Data rate:** The transmission rate: the number of bits sent in each second-is also defined by the physical layer. Also the physical layer defines the duration of a bit, which is how long it lasts.

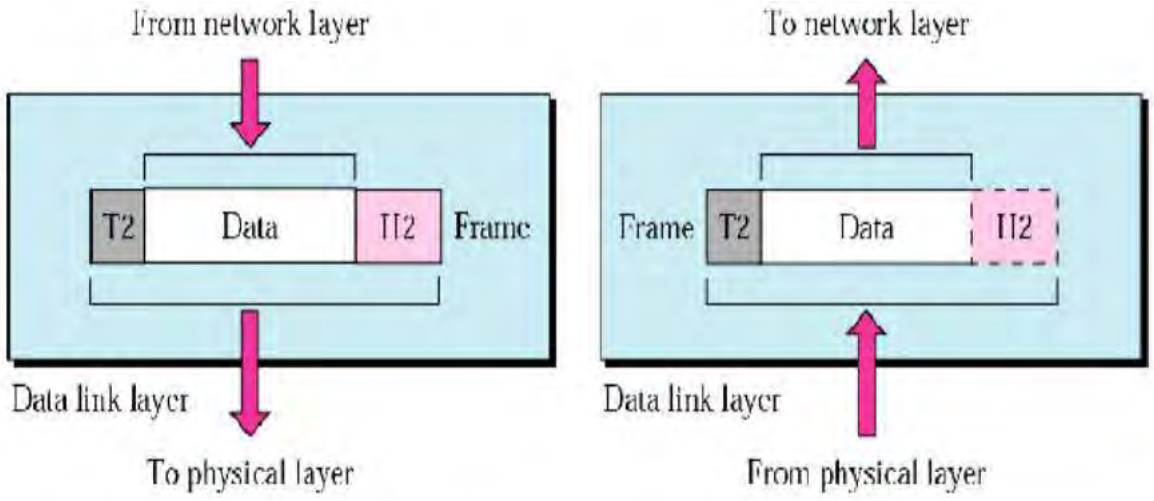
- **Synchronization of bits:** The sender and receiver must use the same bit rate also must be synchronized at the bit level. As well as, the sender and the receiver clocks must be synchronized.

- **Line configuration:** The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.

- **Physical topology:** The physical topology defines how devices are connected to make a network. It is the geographical representation of nodes, it can be generated by any of the topology (mesh, bus, star, ring) depends upon its application.

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

**DATA LINK LAYER**




- The *data link layer* is the second layer in the OSI (open systems interconnection) seven-layer reference model.
- It responds to service requests from the network layer above it and issues service requests to the physical layer below it.
- The data link layer is divided into two sub layers: the *media access control* (MAC) layer and the *logical link control* (LLC) layer.
- The former controls how computers on the network gain access to the data and obtain permission to transmit it.
- The data link layer is often implemented in software as a driver for a network interface card (NIC). Because the data link and physical layers are so closely related, many types of hardware are also associated with it.

Other major 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. Header defines the sender and/or receiver of the frame or physical address the network locally.

- **Flow control:** If the rate at which are absorbed by the receiver is less than the rate at which data are produced by the sender communication is unbalanced. The data link layer imposes a flow control mechanism to avoid **overwhelming** the receiver.

- **Error control:** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a **trailer** added to the end of the frame.

- **Access control:** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has preference to access and control over the link at any given time. So data link layer provide appropriate medium access control mechanism to the system.

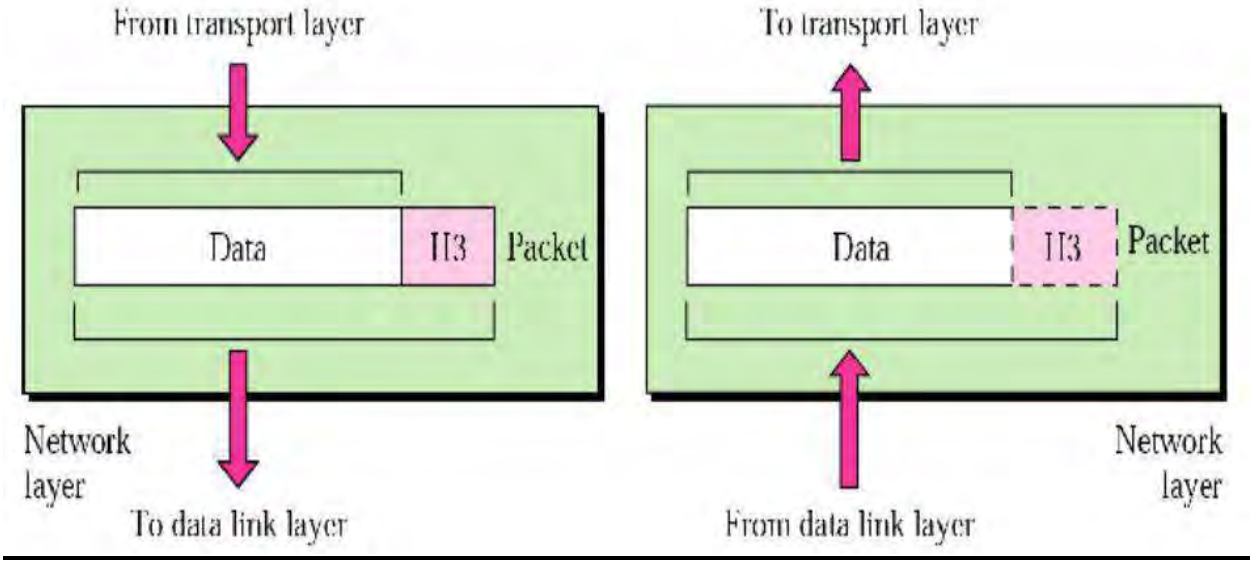
---

The data link layer is responsible for moving frames from one hop (node) to the next.

---

## NETWORK LAYER

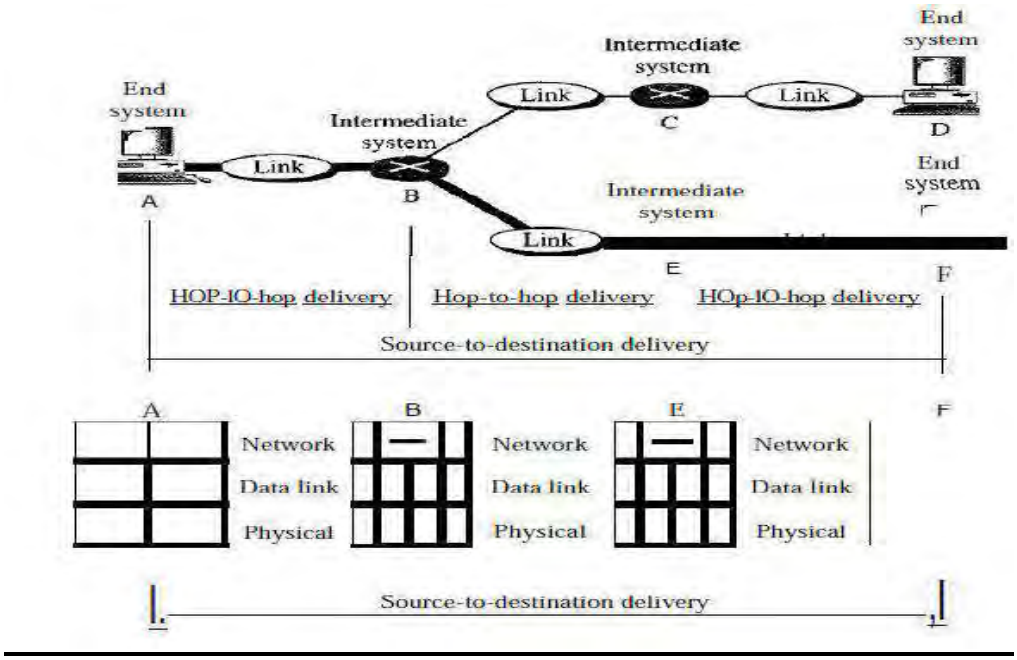
The network layer is responsible for the source-to-destination delivery of a packet across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links). The network layer ensures that each packet gets from its point of origin to its final destination.




Other responsibilities of the network layer include the following:

- **Logical addressing**: The physical addressing implemented by the data link layer handles the addressing problem **locally**. If a packet passes the network boundary or nodes located in other network then, we need logical address to distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that includes the logical addresses of the sender and receiver.

- **Routing.** Routing means finding suitable path to forward the message. When independent networks or links are connected to create *internetworks* (network of networks) or a large network, the connecting devices (called *routers* or *switches)* route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.



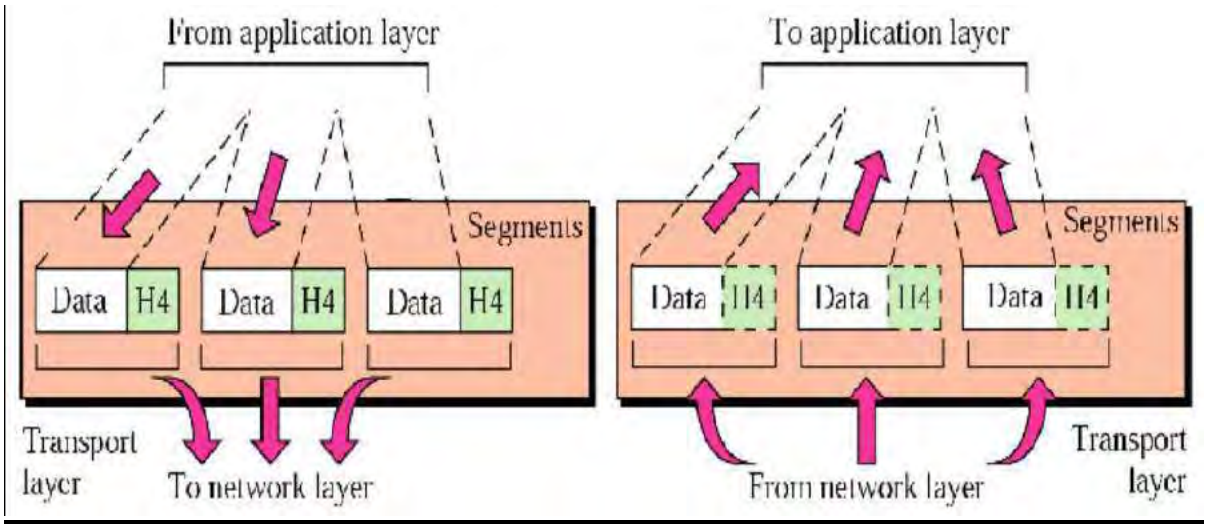

The above figure shows, a source-to-destination delivery. The network layer at A sends the packet to the network layer at B. When the packet arrives at router B, the router makes a decision based on the final destination (F) of the packet. Router B uses its routing table to find that the next hop is router E. The network layer at B, therefore, sends the packet to the network layer at E. The network layer at E, in turn, sends the packet to the network layer at F.

The network layer is responsible for the delivery of individual packets from the source host to the destination host.

## TRANSPORT LAYER

The transport layer is responsible for process-to-process delivery of the entire message.

A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message. The transport layer, on the other hand, ensures that the whole message arrives in order and each packet dependent to each other. It oversees both error control and flow control at the source-to-destination level. It is also responsible for end-to end communication.




Other responsibilities of the transport layer include the following:

- **Service-point addressing or port 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. For this purpose the transport layer header must include a type of address called a ***service-point address*** (or **port address**). The network layer gets each packet to the correct computer where the transport layer gets the entire message to the correct process on that computer.

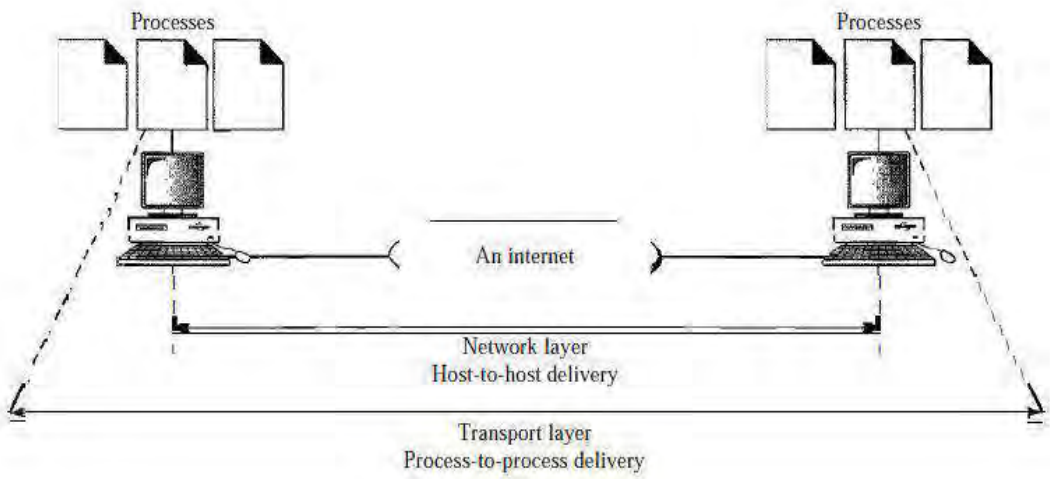
- **Segmentation and reassembly:** A message is divided into transmittable forms called segments. Each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost or damage in transmission.

- **Connection control:** The transport layer can be either connectionless or connection oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection oriented transport layer makes a connection with the transport layer at the destination machine first, before delivering the packets. After all the data are transferred, the connection is terminated.

- **Flow control:** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

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

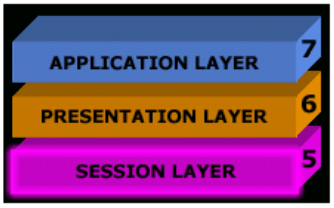
Figure below illustrates process-to-process delivery by the transport layer.




The transport layer is responsible for the delivery of a message from one process to another.

## SESSION LAYER

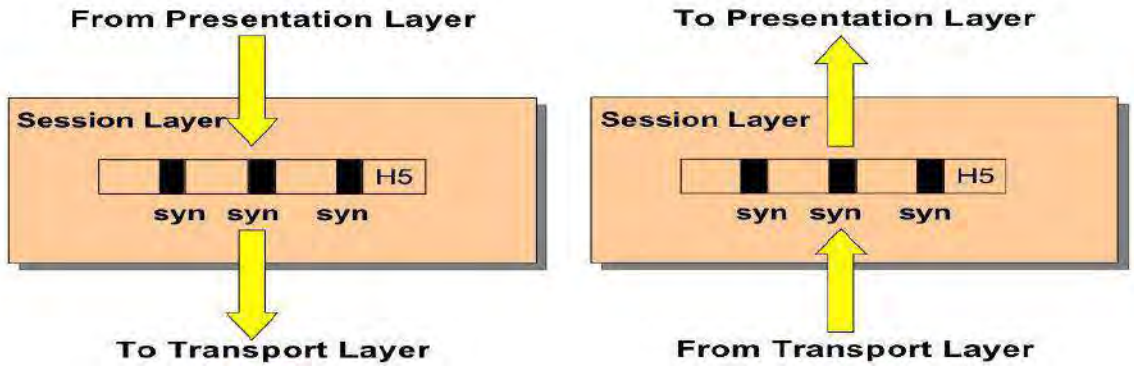
- The Session layer is responsible for setting up, managing and then tearing down sessions between Presentation layer entities.
- The session layer tracks connections, also called sessions.
- The Session layer also provides dialog control between devices, or nodes.
- It coordinates communication between systems and serves to organize their communication by offering three different modes: ***simplex, half-duplex and full-duplex.***
- The session layer basically keeps one application's data separate from other application's data.
- It establishes, maintains, and synchronizes the interaction among communicating systems.




The last 3 layers of the OSI model are reffered to the "Upper" layers. These layers are responsible for applications communicating between hosts. None of the upper layers know anything about networking or network addresses.

Some common protocols which work at the Session layer are: DNS, LDAP, NetBIOS.

The session layer is responsible for dialog control and synchronization.



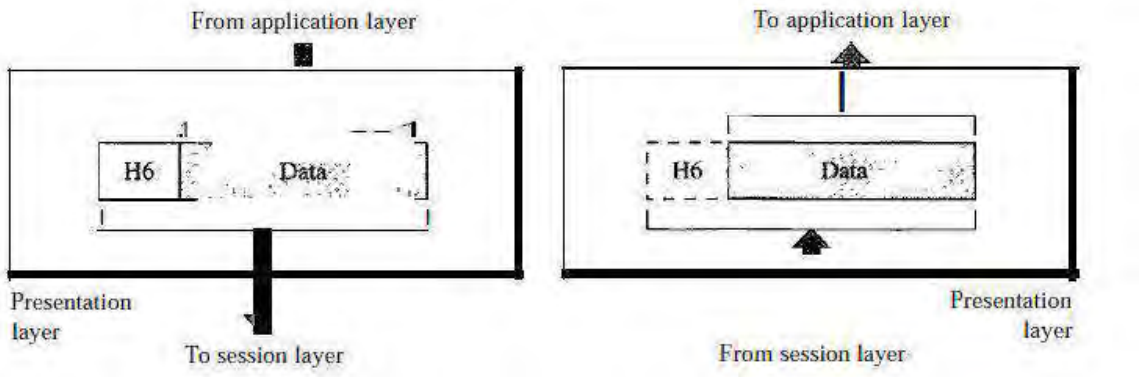

Specific responsibilities of the session layer include the following:

- **Dialog control**: The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half duplex (one way at a time) or full-duplex (two ways at a time) mode.

- **Synchronization**: The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Above figure illustrates the relationship of the session layer to the transport and presentation layers.

## PRESENTATION LAYER

- The presentation layer is primarily concerned with the syntax and semantics of the information exchanged between two systems.
- Presentation layer takes care that data sent in such a way the receiver will understand the data and will able to use.
- The presentation layer is the sixth layer of the OSI model.
- It responds to service requests from the application layer and issues service request to the session layer.
- Below figure shows the relationship between the presentation layer and the application and session layers.



Specific responsibilities of the presentation layer include the following:

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

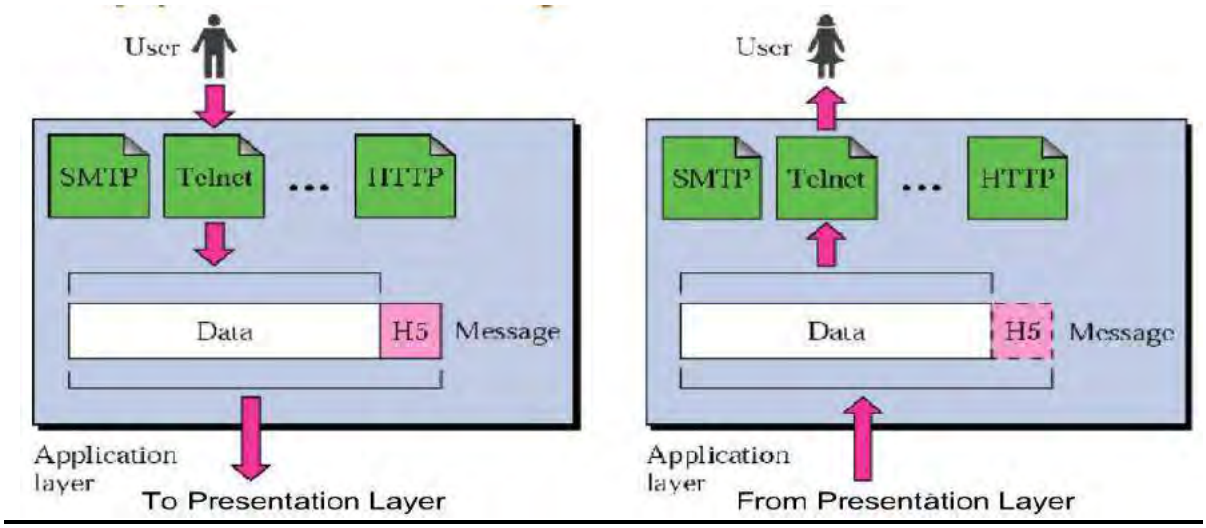
- **Encryption:** To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. **Decryption** reverses the original process to transform the message back to its original form.

- **Compression:** The goal of data compression is to represent an information source (e.g. a data file, a speech signal, an image, or a video signal) as accurately as possible using the fewest number of bits. Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video etc.

The presentation layer is responsible for translation, compression, and encryption.

## 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.
- The OSI application layer is responsible for displaying data and images to the user in a human-recognizable format and to interface with the ***presentation*** layer below it.




Specific services provided by the application layer include the following:

- **Network virtual terminal or remote log-in:** A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.

- **File transfer, access, and management:** This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.

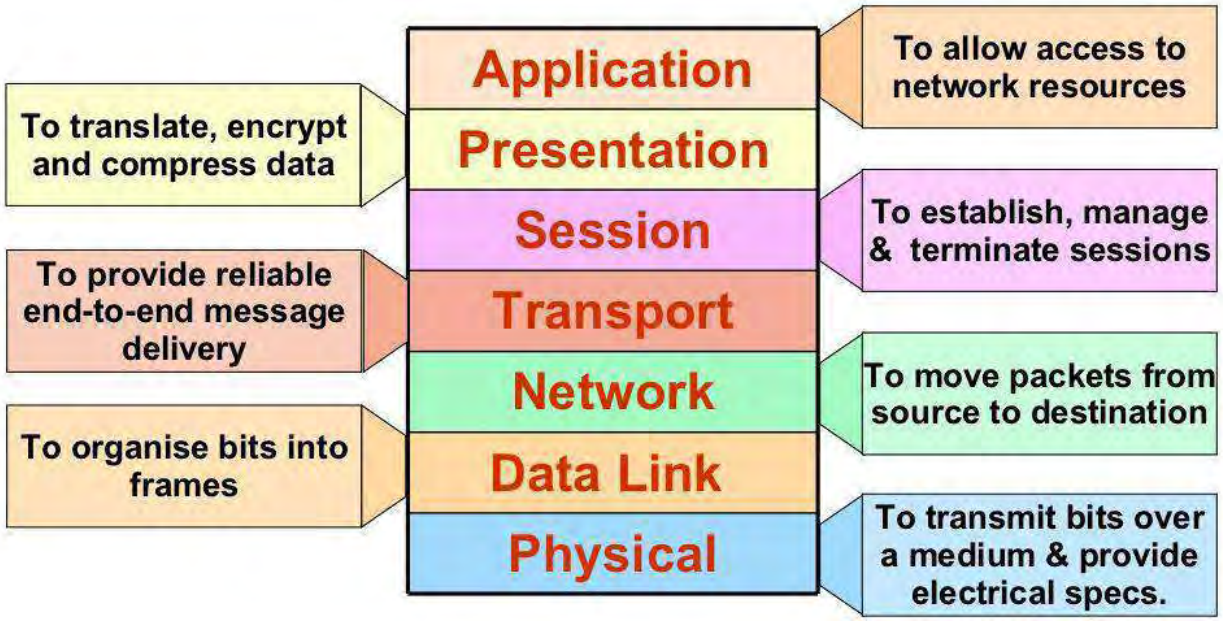
- **Mail services:** This application provides the basis for e-mail forwarding and storage.

- **Directory services or accessing the World Wide Web:** This application provides distributed database sources and access for global information about various objects and services. The most common application today is the access of the WWW.

The application layer is responsible for providing services to the user.

**Summary of Layers**

Figure below shows a summary of duties for each layer.



*LECTURE NOTE: 4*

# DETAILS OF PHYSICAL LAYER AND MEDIUM

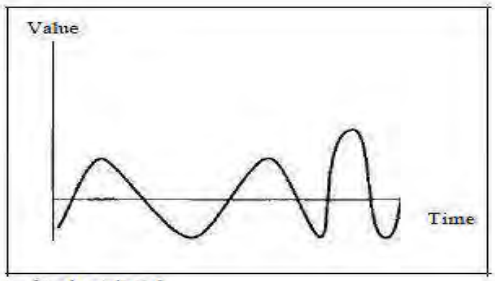
**Signals:** One of the major concerns of the physical layer lies in moving data in the form of electromagnetic signals across a transmission medium. For example to transmit a photograph through a medium you need to use encoder to convert stream of 1s and 0s instead of sending actual photograph. That must be converted into a form that transmission media can accept. Transmission media work by conducting energy along a physical path. So a data stream must be converted in to energy in the form of electromagnetic signals.

**Analog and Digital**

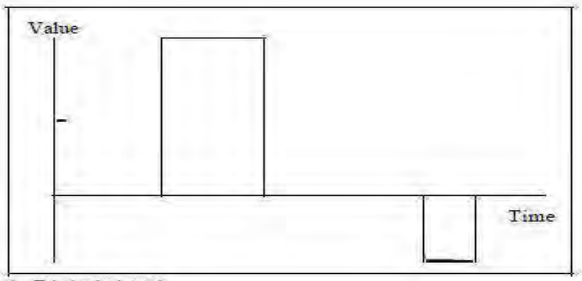
Both data and the signals that represent them can be either **analog or digital** form. Data can be analog or digital. The term **analog data** refers to information that is continuous where as digital **data** refers to information that has discrete states. We can discuss both the signal as, an **analog signal** is any continuous signal for which the time varying feature (variable) of the signal is a representation of some other time varying quantity, i.e., analogous to another time varying signal. A **digital signal** is a chemical signal that is a representation of a sequence of discrete values (a quantified discrete-time signal), for example of **arbitrary bit stream**, or **of a digitized** (sampled and analog-to-digital converted) **analog signal.**

A digital signal that is generated by means of a digital modulation method (digital pass band transmission), produced by a modem.

Analog data, such as the sounds made by a human voice, take on continuous values. When someone speaks, an analog wave is created in the air. This can be captured by a microphone and converted to an analog signal or sampled and converted to a digital signal. Digital data take on **discrete** values. For example, data are stored in computer memory in the form of 0s and 1s. They can be converted to a digital signal or modulated into an analog signal for transmission across a medium.



a. Analog signal



b. Digital signal

Data can be analog or digital. Analog data are continuous and take continuous values. Digital data have discrete states and take discrete values.

**Periodic and Non periodic Signals**

Both analog and digital signals can take one of two forms: **periodic or non periodic**. In data communications, we commonly use periodic analog signals

**Periodic Analog Signals**

- Periodic Signals are signals that repeat themselves after a certain amount of time.

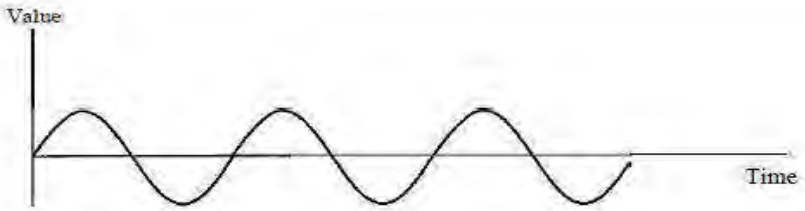
  a function *f(t)* is periodic if $f(t + T) = f(t)$ for some *T* and all *t*. The classic example of a periodic function is $sin(x)$ since $sin(x + 2\pi) = sin(x)$.

  A periodic signal completes a pattern within a measurable time frame, called a **period**, and repeats that pattern over subsequent identical periods.

- The completion of one full pattern is called a **cycle**.
- The sine wave is the most fundamental form of a periodic analog signal.
- Like the data they represent, signals can be either analog or digital. An analog signal has infinitely many levels of intensity over a period of time.

As the wave moves from value *A* to value *B*, it passes through and includes an infinite number of values along its path. A digital signal, on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and 0. The simplest way to show signals is by plotting them on a pair of perpendicular axes. The vertical axis represents the value or strength of a signal. The horizontal axis represents time.

*A sine wave*

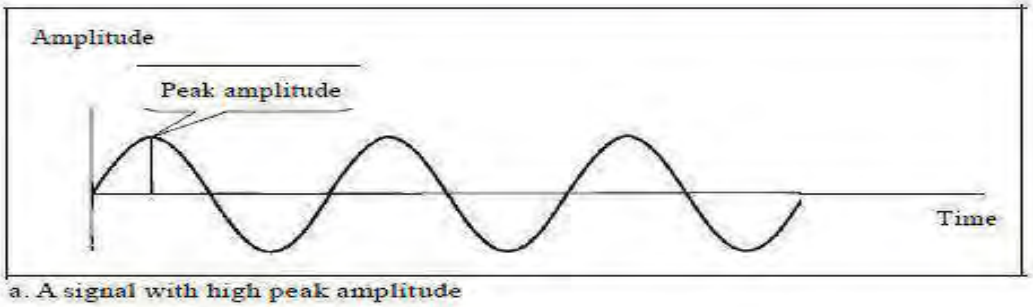



## PEAK AMPLITUDE

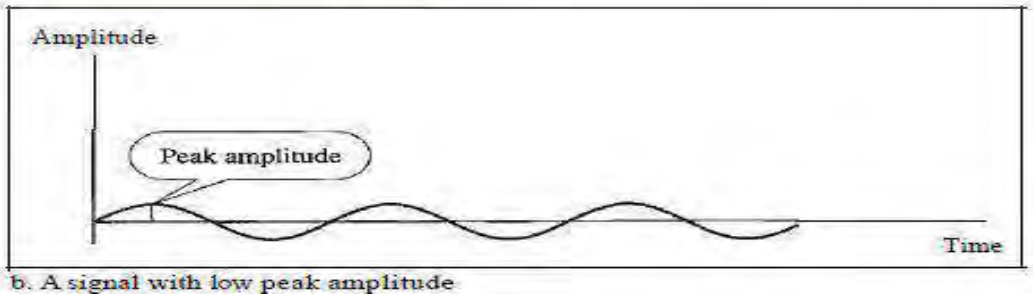
The peak amplitude of a signal is the absolute value of its highest intensity, proportional to the energy it carries. For electric signals, peak amplitude is normally measured in volts.

***In the below figure, two signals with the same phase and frequency, but different amplitudes***

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




a. A signal with high peak amplitude




b. A signal with low peak amplitude

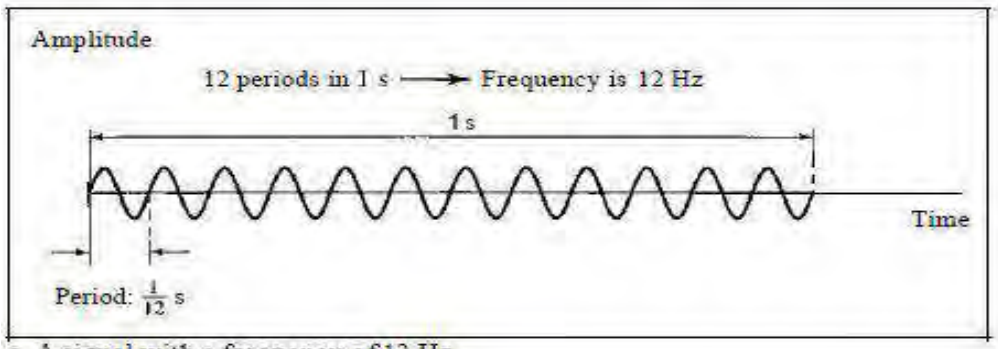
## PERIOD AND FREQUENCY

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

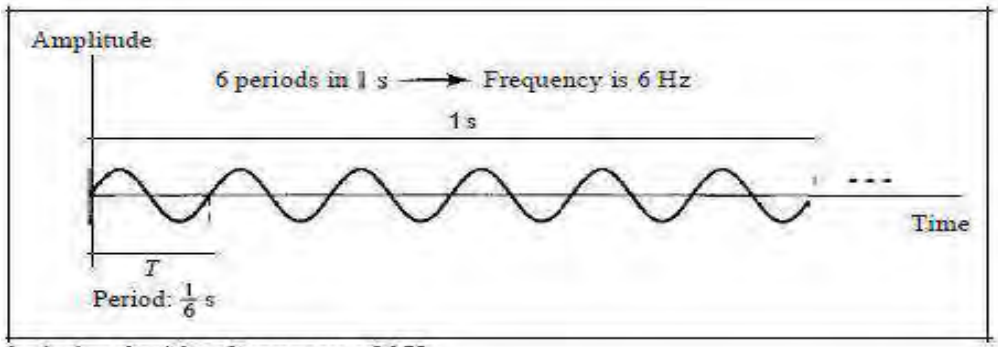
$$f = \left(\frac{1}{T}\right) \text{ And } T = \left(\frac{1}{f}\right)$$

**Frequency and period are the inverse of each other.**

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




a. A signal with a frequency of 12 Hz




b. A signal with a frequency of 6 Hz

Period is formally expressed in **seconds.** Frequency is formally expressed in **Hertz (Hz)**, which is cycle per second. Units of period and frequency are shown below:

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

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

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

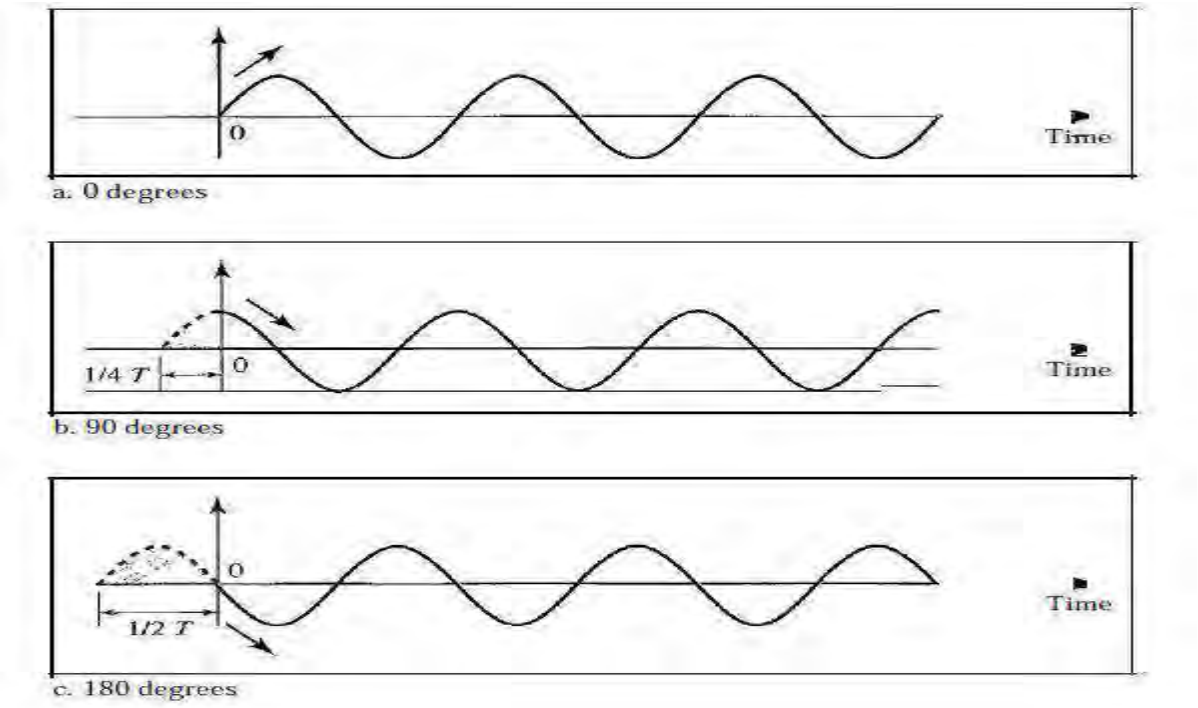
**If a signal does not change at all, its frequency is zero.**
**If a signal changes instantaneously, its frequency is infinite.**

## PHASE

- The term **phase** describes the position of the waveform relative to time 0.
- A complete cycle is defined as 360 degrees of phase.
- Phase can also be an expression of relative displacement between or among waves having the same frequency .
- If we think of the wave as something that can be shifted backward or forward along the time axis, phase describes the amount of that shift. It indicates the status of the first cycle.

Phase is measured in degrees or radians [360° is *2n* rad, 1° is *2n/360* rad, and 1 rad
is *360/(2n)]*. A phase shift of 360° corresponds to a shift of a complete period; a phase

shift of 180° corresponds to a shift of one-half of a period; and a phase shift of 90° corresponds to a shift of one-quarter of a period. The below figure shows three sine waves with the same amplitude and frequency, but different phases




Looking at above figure we can say that
1. A sine wave with a phase of 0° starts at time 0 with a zero amplitude. The amplitude is increasing.

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

3. A sine wave with a phase of 180° starts at time 0 with a zero amplitude. The amplitude is decreasing.

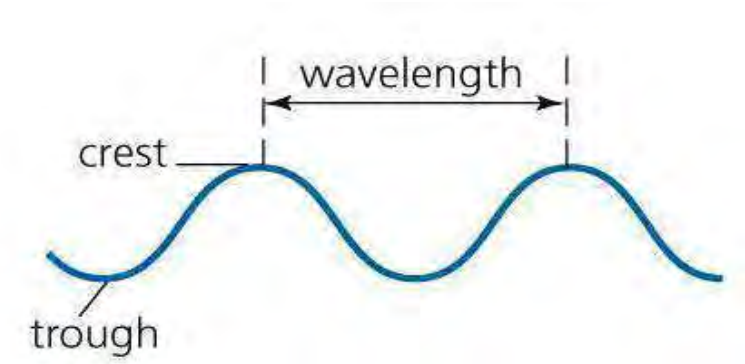
The time interval for one degree of phase is inversely proportional to the frequency. If the frequency of a signal (in hertz ) is given by $f$ , then the time $t_{deg}$ (in seconds) corresponding to one degree of phase is:

$t_{deg} = 1 / (360 f)$

The time $t_{rad}$ (in seconds) corresponding to one radian of phase is approximately:

$t_{rad} = 1 / (6.28\ f)$

**WAVELENGTH**




- Wavelength is the distance between identical points in the adjacent cycles of a waveform signal propogated in space or along a wire.
- Or it is the distance between one peak of a wave to the next corresponding peak, or between any two adjacent corresponding points. Wavelength is another characteristic of a signal travelling through a transmission medium.
- Wavelength binds the period or the frequency of a simple sine wave to the propagation speed of the medium.
- While the frequency of a signal is independent of the medium, the wavelength depends on both the frequency and the medium.
- Wavelength is a property of any type of signal.
- In data communications, we often use wavelength to describe the transmission of light in an optical fiber.
- The wavelength is the distance a simple signal can travel in one period.
- Wavelength can be calculated if one is given the propagation speed (the speed of light) and the period of the signal.
- However, since period and frequency are related to each other, if we represent wavelength by A, propagation speed by c (speed of light), and frequency by*1*, we get

**Wavelength =propagation speed x period=propagation speed/frequency**

The propagation speed of electromagnetic signals depends on the medium and on the frequency of the signal. For example, in a vacuum, light is propagated with a speed
of 3 x 108 m/s. That speed is lower in air and even lower in cable.

The wavelength is normally measured in micrometers (microns) instead of meters.

For example, the wavelength of red light (frequency =4 x 1014) in air is

A= c/f= $(3xl0^{8})/(4x10^{14})$ =0.75 x 10-6 m=0.75µm

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

## COMPOSITE SIGNALS

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

- If we had only one single sine wave to convey a conversation over the phone, it would make no sense and carry no information. We would just hear a buzz. As we will see in later chapters, we need to send a composite signal to communicate data.
- A **composite signal** is made of many simple sine waves.
- According to Fourier analysis, any composite signal is a combination of simple sine waves with different **frequencies**, **amplitudes**, and **phases**.
- A composite signal can be periodic or non periodic.
- If the composite signal is periodic, the decomposition gives a series of signals with discrete frequencies; if the composite signal is non periodic, the decomposition gives a combination of sine waves with continuous frequencies.

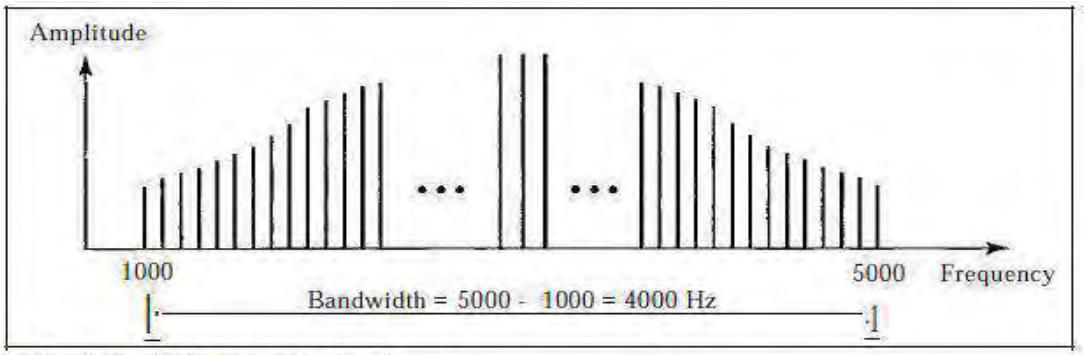
**Note, A single frequency sine wave is not useful in data communications, we need to send a composite signal, a signal made of many simple sine waves.**

## BANDWIDTH

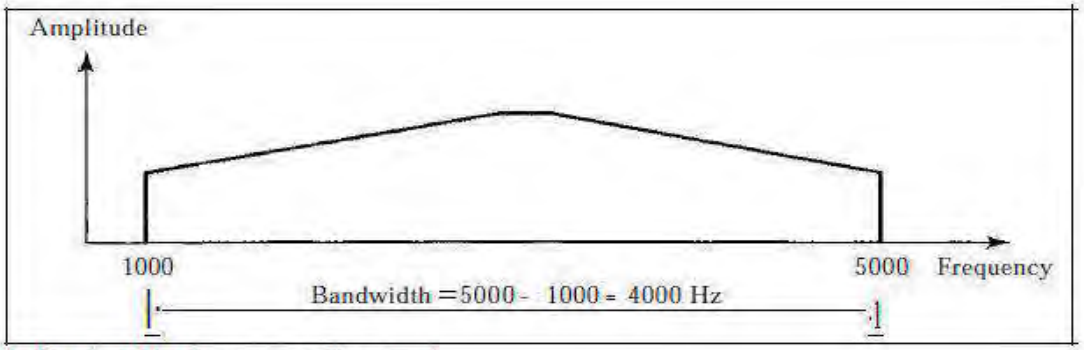
- The term Bandwidth defines the net bit rate , channel capacity, or the maximum throughput of a logical or physical communication path in a communication system.

- The range of frequencies contained in a composite signal is its bandwidth.
- The bandwidth in analog signal is normally a difference between two numbers i.e the difference between the highest and the lowest frequencies contained in that signal.
- For example, if a composite signal contains frequencies between 1000 and 5000, its bandwidth is 5000 - 1000, or 4000. Below figure shows the concept of bandwidth, where depicts two composite signals, one periodic and the other non periodic. The bandwidth of the periodic signal contains all integer frequencies between 1000 and 5000 (1000, 100 I, 1002, ...). The bandwidth of the non periodic signals has the same range, but the frequencies are continuous.

***The below figure showing the bandwidth of periodic and non periodic composite signals***




a. Bandwidth of a periodic signal




b. Bandwidth of a nonperiodic signal

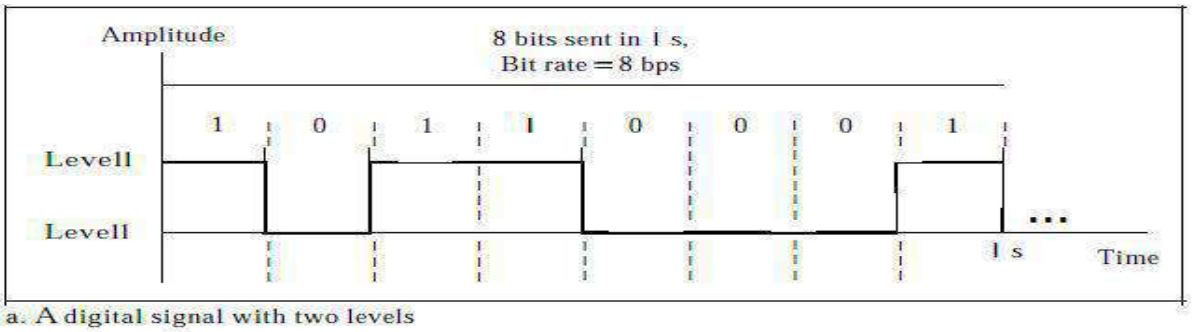
***LECTURE NOTE: 5***

# DIGITAL SIGNALS

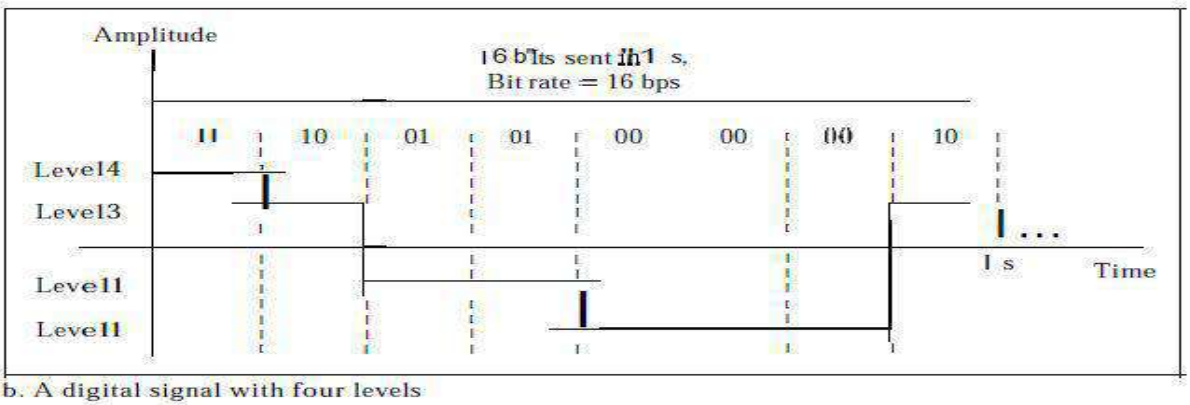
Information can also be represented by a digital signal. For example, a 1 can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels. We can send more than 1 bit for each level. Below figure a and b shows two signals, one with two levels and the other with four. In this case we send 1 bit per level in part of the figure and 2 bits per level in part b of the figure. In general, if a signal has *L* levels, each level needs $log_2 L$ bits.

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

Number of bits per level= $\log_2 8=3$
Each signal level is represented by 3 bits.



a. A digital signal with two levels



b. A digital signal with four levels

**BIT RATE**

Most digital signals are non periodic, and thus period and frequency are not appropriate characteristics. Another term-bit rate (instead frequency)-is used to describe digital signals so bandwidth of the digital signal is measured in bit rate. The **bit rate** is the number of bits sent in 1s, expressed in bits per second (bps).

**Example1**

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

the required bit rate of the channel?

**Solution**

A page is an average of 24 lines with 80 characters in each line. If we assume that one

Character requires 8 bits, the bit rate is

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

**Example 2**

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

**Solution**

The bit rate can be calculated as

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

**BIT LENGTH**

Wavelength for an analog signal is the distance one cycle occupies on the transmission medium. Similarly for a digital signal, the **bit length**: The bit length is the distance one bit occupies on the transmission medium. It is expressed as

**Bit length =propagation speed x bit duration**

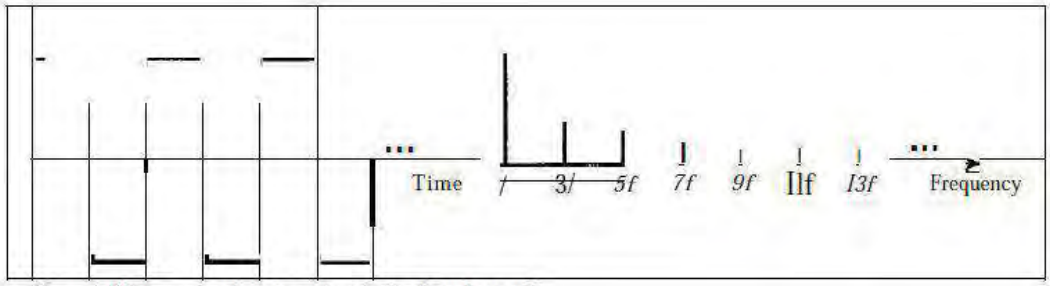
## Digital Signal as a Composite Analog Signal

- Based on Fourier analysis, a digital signal is a composite analog signal, Where the bandwidth is infinite.
- A digital signal, in the time domain, comprises connected vertical and horizontal line segments.
- A vertical line in the time domain means a frequency of infinity (sudden change in time), a horizontal line in the time domain means a frequency of zero (no change in time). Going from a frequency of zero to a frequency of infinity (and vice versa) implies all frequencies in between are part of the domain.

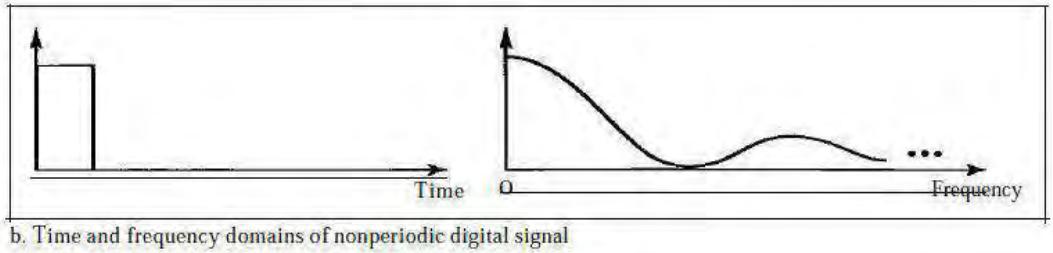
- Fourier analysis can be used to decompose a digital signal.

- If the digital signal is periodic, which is rare in data communications, the decomposed signal has a frequency domain representation with an infinite bandwidth and discrete frequencies.

- If the digital signal is non periodic, the decomposed signal still has an infinite bandwidth, but the frequencies are continuous. Below figure shows a periodic and a non periodic digital signal and their bandwidths.




a. Time and frequency domains of **periodic** digital signal




b. Time and frequency domains of nonperiodic digital signal

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

## ANALOG VS DIGITAL SIGNALS

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

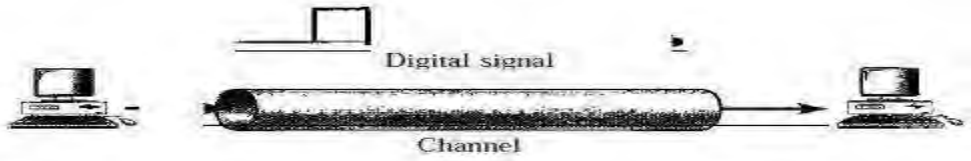
## TRANSMISSION OF DIGITAL SIGNALS

- The previous discussion asserts that a digital signal, periodic or non periodic, is a composite analog signal with frequencies between zero and infinity.
- Non periodic digital signal is used in data communication.
- We can transmit a digital signal by using one of the two different approaches: **baseband transmission or broadband transmission** (using modulation).

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

### *BASEBAND TRANSMISSION*

- Baseband transmission means sending a digital signal over a channel without changing the digital signal to an analog signal.
- Baseband transmission requires a low-pass channel, a channel with a band width that starts from zero. **Below Figure shows** baseband transmission.




- We can have two low-pass channels: one with a narrow bandwidth and the other with a wide bandwidth such as base band communication **of low pass channel with wide bandwidth and with limited bandwidth (narrow bandwidth).**
- In a low-pass channel with limited bandwidth, we approximate the digital signal with an analog signal. The level of approximation depends on the bandwidth available.
- We need to remember that a low-pass channel with infinite bandwidth is ideal, but we cannot have such a channel in real life.

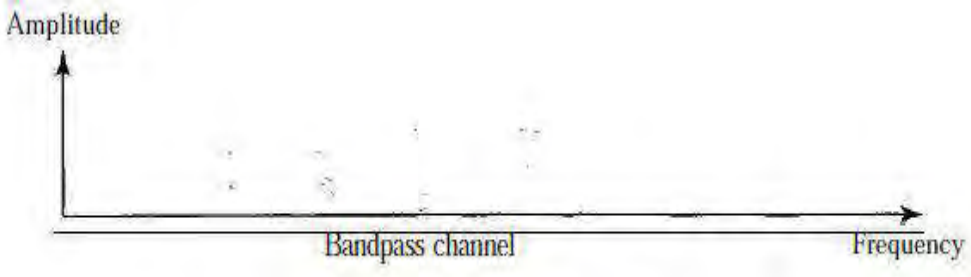
---

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

---

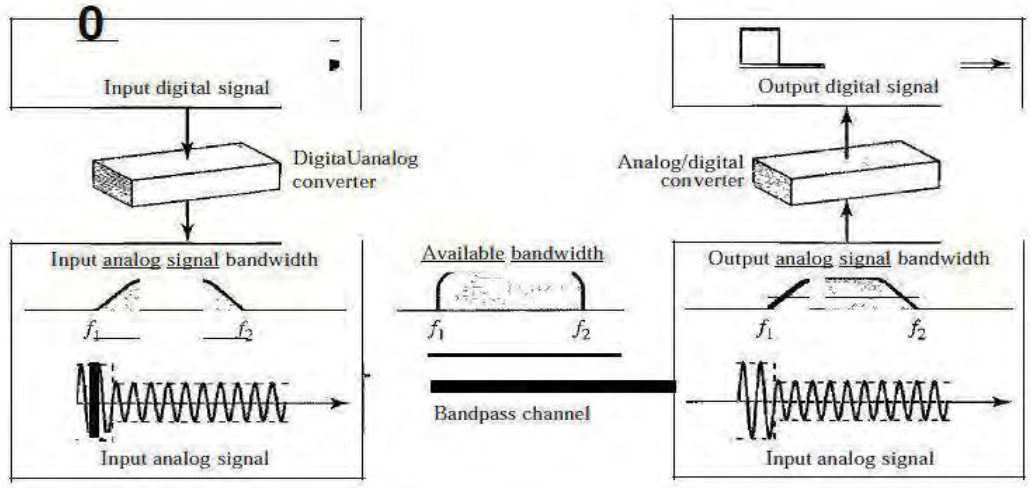
## *BROADBAND TRANSMISSION (USING MODULATION)*

- Broadband transmission or modulation means changing the digital signal to an analog
  signal for transmission.
- Modulation allows us to use a band pass channel, a channel with a band width that does not start from zero. This type of channel is more available than a low-pass channel.




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

Below figure shows the modulation of a digital signal. In the figure, a digital signal is converted to a composite analog signal. We have used a single-frequency analog signal called a carrier, the amplitude of the carrier has been changed to the digital signal. The result, however, is not a single-frequency signal, it is a composite signal. At the receiver, the received analog signal is converted to digital, and the result is a replica of what has been sent.

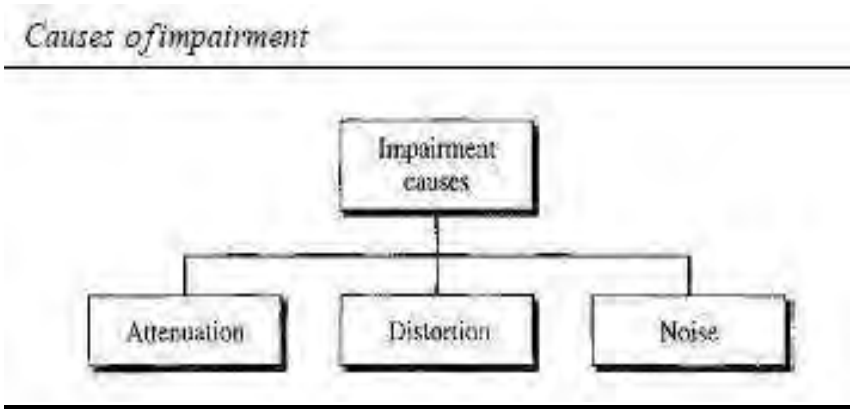



If the available channel is a bandpass channel, we cannot send the digital signal directly to the channel; we need to convert the digital signal to an analog signal before transmission.

## TRANSMISSION IMPAIRMENT

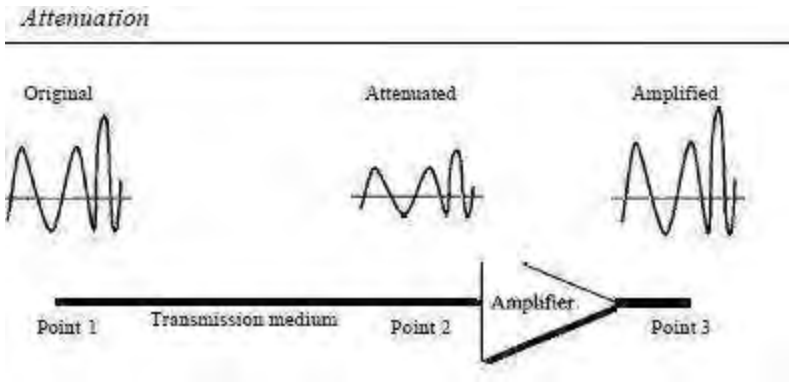
- Signals travel through transmission media, which are not perfect.
- The imperfection causes signal impairment.
- This means that the signal at the beginning of the medium is not the same as the signal at the end of the medium.
- What is sent is not what is received is because of impairment (unwanted interruption).

Three causes of impairment are **attenuation, distortion, and noise**




**Attenuation:**

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




**Decibel**

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

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

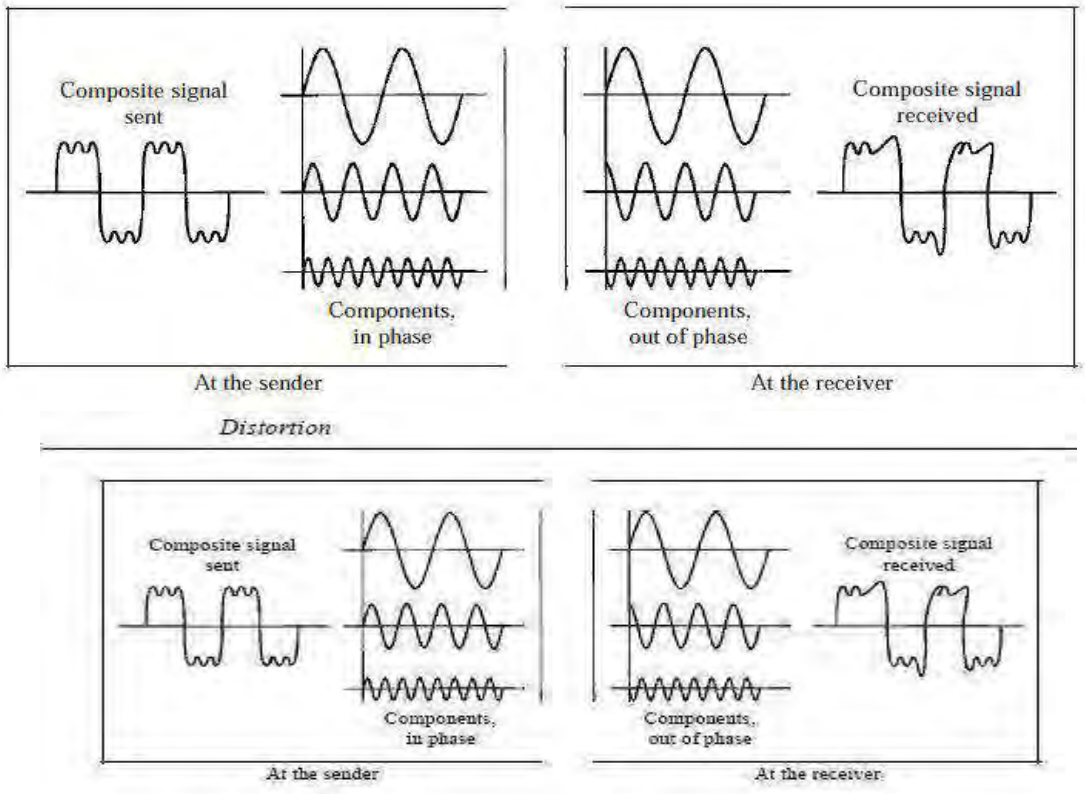
Variables *PI* and *P2* are the powers of a signal at points 1 and 2, respectively.

- **Note** that some engineering books define the decibel in terms of voltage instead of power. In this case, because power is proportional to the square of the voltage, the formula is **dB = 20 $\log_{10}$** ***(V2/V1)***.

  In this text, we express dB in terms of power.

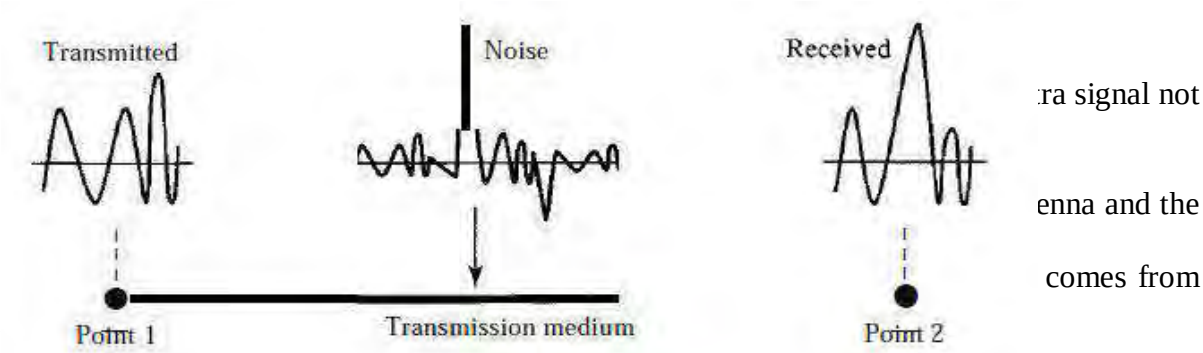
**Distortion**

- Distortion means that the signal changes its form or shape.
- Distortion can occur in a composite signal made of different frequencies.
- Each signal component has its own propagation speed through a medium and, therefore, its own delay in arriving at the final destination.
- Differences in delay may create a difference in phase if the delay is not exactly the same as the period duration.
- The shape of the composite signal is therefore not the same. The effect of distortion on a composite signal is showing below.



**Noise**

- Noise is another cause of impairment.
- Several types of noise, such as **thermal noise, induced noise, crosstalk, and impulse noise,** may corrupt the signal. Below figure shows the effect of noise on a signal.




ra signal not

enna and the

comes from

**SIGNAL-TO-NOISE RATIO (SNR)**

- As we will see later, to find the theoretical bit rate limit, we need to know the ratio of

the signal power to the noise power. The signal-to-noise ratio is defined as

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

- SNR is actually the ratio of what is wanted (signal) to what is not wanted (noise).
- A high SNR means the signal is less corrupted by noise; a low SNR means the signal is more corrupted by noise.
- Because SNR is the ratio of two powers, it is often described in decibel units, $SNR_{dB}$, defined as

$$\mathbf{SNR_{dB} = 10 \log_{10} SNR}$$

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

$$SNR = \frac{\text{signal power}}{0} = \infty$$

$$SNR_{dB} = 10 \log_{10} \infty = \infty$$

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

**DATA RATE LIMITS**

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

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

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

**Noiseless Channel: Nyquist Bit Rate**

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

$$\textbf{Bit Rate} = 2 \times \textbf{bandwidth} \times 10g_2 L$$

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

**Noisy Channel: Shannon Capacity**

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

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

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

- **Note** that in the Shannon formula there is no indication of the signal level, which means that no matter how many levels we have, we cannot achieve a data rate higher than the capacity of the channel.
- In other words, **the formula defines a characteristic of the channel, not the method of transmission.**

**Example**

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

$$C = B \log_2 (1 + SNR) = B\ 10g_2 (l + 0) = B \log_2 1 = 0$$

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

---

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

---

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

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

- There is an explicit relationship between the bandwidth in hertz and bandwidth in bits per seconds. Basically, an increase in bandwidth in hertz means an increase in bandwidth in bits per second.

- The relationship depends on whether we have baseband transmission or transmission with modulation.

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

*LECTURE NOTE: 6*

# DIGITAL TRANSMISSION

- A computer network is designed to send information from one point to another.
- This information needs to be converted to either a digital signal or an analog signal for transmission.
- In this chapter, conversion to digital signals is discussed as digital-to-digital conversion techniques, methods which convert digital data to digital signals. Second, we discuss analog to- digital conversion techniques, methods which change an analog signal to a digital signal finally, we discuss transmission modes.

## DIGITAL TO DIGITAL CONVERSION

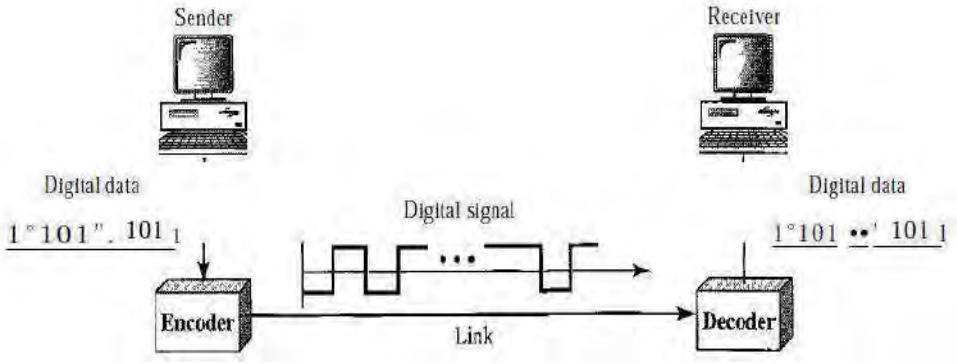
Data can be either digital or analog. We also said that signals that represent data can also be digital or analog. In this section, we can represent digital data by using digital signals. The conversion involves three techniques: **line coding**, **block coding**, and **scrambling**. Line coding is always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

## LINE CODING

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

- Line coding converts a sequence of bits to a digital signal.
- At the sender, digital data are encoded into a digital signal, at the receiver, the digital data are recreated by decoding the digital signal.

Figure showing **Line coding and decoding**

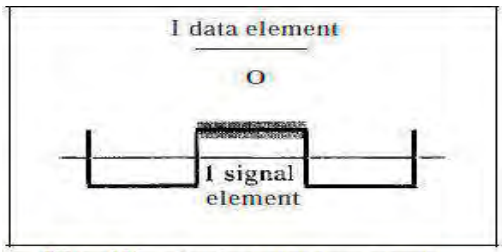



Before discuss about **Line coding** let us go through its characteristics as

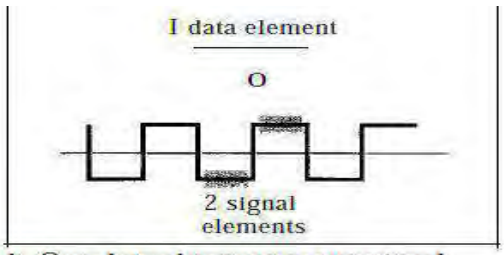
## SIGNAL ELEMENT VERSUS DATA ELEMENT

- A data element is the smallest entity that can represent a piece of information: this is the bit. In digital data communications, a signal element carries data elements.
- A signal element is the shortest unit (time wise) of a digital signal.
- In other words, **data elements are what we need to send, signal elements are what we can send.** Data elements are being carried signal elements are the carriers.

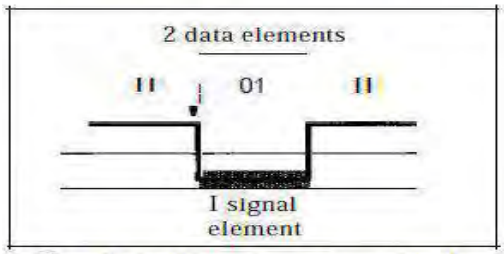
We define here a ratio *r* which is the number of data elements carried by each signal element. Figure shows several situations with different values of *r*.



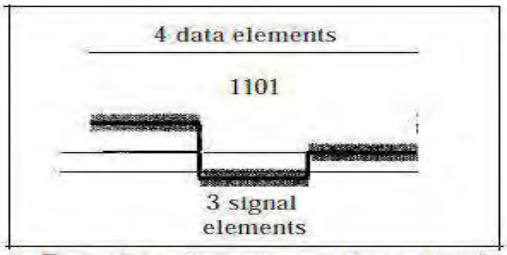

a. One data element per one signal element ($r = 1$)




b. One data element per two signal elements $\left(r = \frac{1}{2}\right)$




c. Two data elements per one signal element ($r = 2$)




d. Four data elements per three signal elements $\left(r = \frac{4}{3}\right)$

In part a of the figure, one data element is carried by one signal element ($r$ = 1). In part b of the figure, we need two signal elements (two transitions) to carry each data element ($r$ = 1/2 ), In part c of the figure, a signal element carries two data elements ($r$ = 2). Finally, in part d, a group of 4 bits is being carried by a group of three signal elements ($r$ = 4/3**)**

## DATA RATE VERSUS SIGNAL RATE

- The data rate defines the number of data elements (bits) sent in 1s. The unit is bits per second (bps).
- The signal rate is the number of signal elements sent in 1s. The unit is the baud. There are several common terminologies used in the literature.

  The data rate is sometimes called the **bit rate**, the signal rate is sometimes called the **pulse rate**, the **modulation rate**, or the **baud rate**.

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

S = cxNx 1/r baud

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

**Bandwidth:** the digital signal that carries information is non periodic. We also showed that the bandwidth of a non periodic signal is continuous with an infinite range. However, most digital signals we encounter in real life have a bandwidth with finite values. In other words, the bandwidth is theoretically infinite, but many of the components have such small amplitude that they can be ignored. The effective bandwidth is **finite**.

---

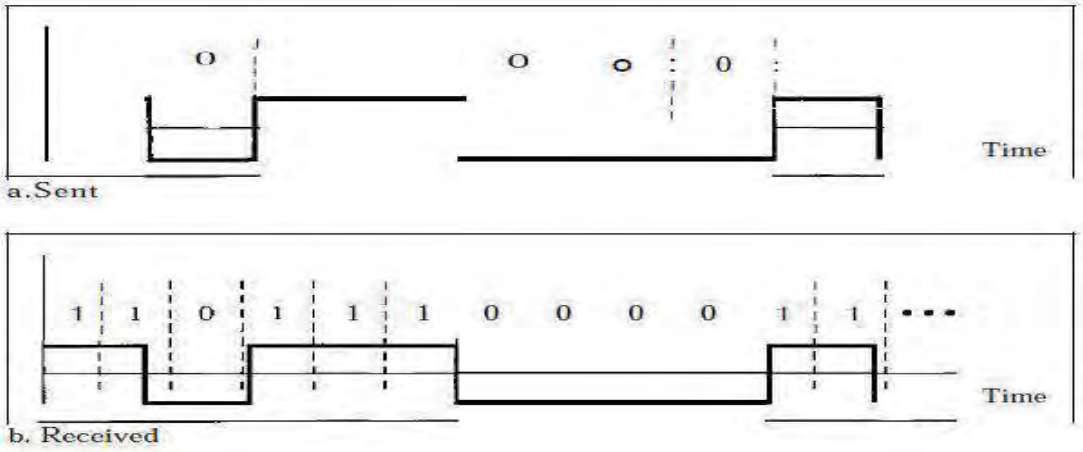
Although the actual bandwidth of a digital signal is infinite, the effective bandwidth is finite.

---

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

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

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



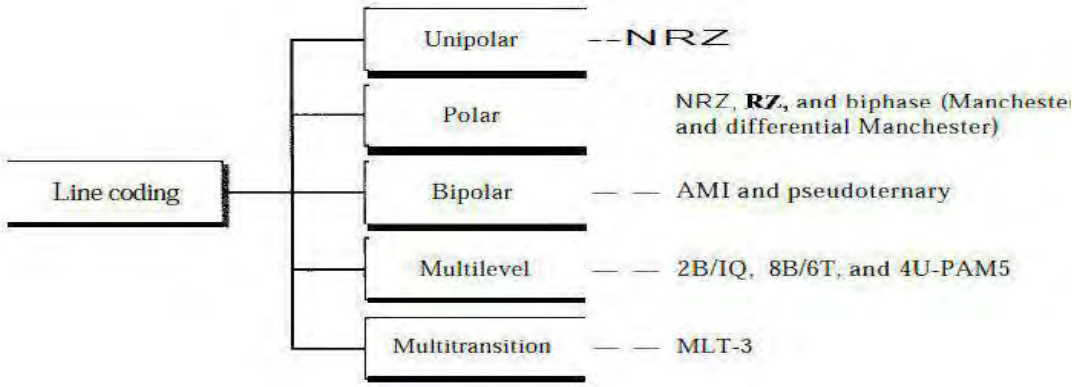

**Built-in Error Detection:** It is desirable to have a built-in error-detecting capability in the generated code to detect some of or all the errors that occurred during transmission. Some encoding schemes have this capability to some extent.

**Immunity to Noise and Interference:** Another desirable code characteristic is a code I that is immune to noise and other interferences. Some encoding schemes have this capability.

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

## LINE CODING SCHEMES

We can roughly divide line coding schemes into five broad categories,

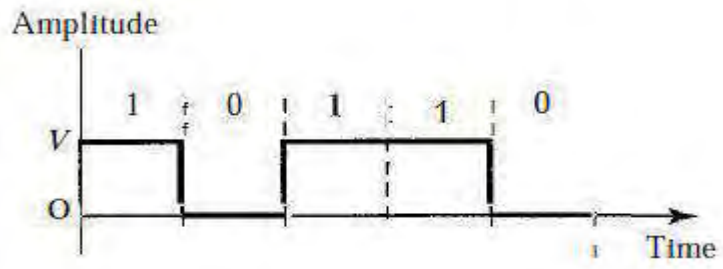


There are several schemes in each category as

## 1. UNIPOLAR

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

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




Compared with its polar counterpart (see the next section), this scheme is very costly.

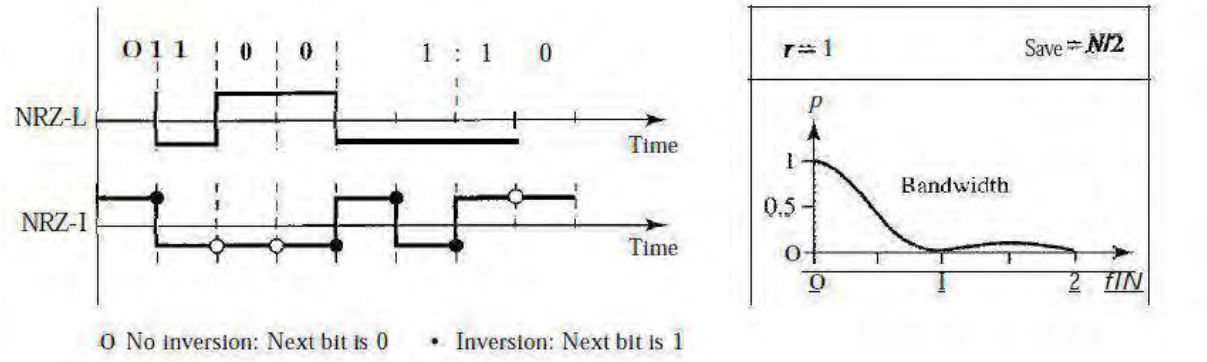
- The normalized power (power needed to send 1 bit per unit line resistance) is double that for polar NRZ.
- For this reason, this scheme is normally not used in data communications today.

## 2. POLAR

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

**(i) Non-Return-to-Zero (NRZ):** In polar NRZ encoding, we use two levels of voltage amplitude. We can have two versions of polar NRZ: NRZ-Land NRZ-I, as shown below. The figure also shows the value of *r*, the average baud rate, and the bandwidth.

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




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

When we compare these two schemes based on the criteria,

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

(2) Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. NRZ-I does not have this problem.

**Note** Both schemes have an average signal rate of *N/2* Bd, NRZ-L and NRZ-I both have a DC component problem.

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

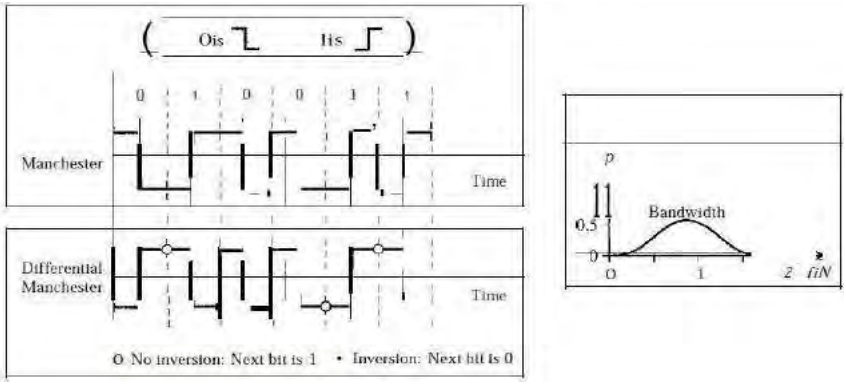
- The **main disadvantage** of RZ encoding is that it requires two signal changes to encode a bit and therefore occupies greater bandwidth, but there is no DC component problem.

- Another problem is the **complexity**: RZ uses three levels of voltage, which is more complex to create and discern.
- As a result of all these **deficiencies**, the scheme is not used today.
- Instead, it has been replaced by the **better-performing Manchester and differential Manchester schemes (discuss next).**

### (iii) Biphase (Manchester and Differential Manchester)

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

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




---

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

---

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

- First, there is no baseline wandering.

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

---

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

---

### 3. BIPOLAR

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

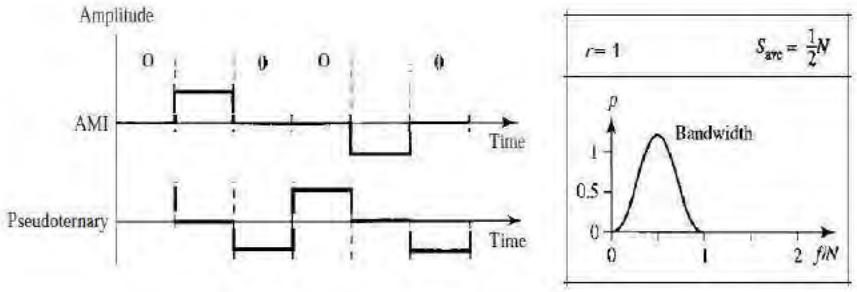
---

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

---

**AMI and Pseudo ternary**

- A common bipolar encoding scheme is called bipolar alternate mark inversion (AMI).
- In the term *alternate mark inversion,* the word *mark* comes from telegraphy and means 1.
- So **AMI means alternate 1 inversion**.
- A neutral zero voltage represents binary 0. Binary 1s are represented by alternating positive and negative voltages.
- **A variation of AMI encoding is called pseudo ternary** in which the 1 bit is encoded as a zero voltage and the 0 bit is encoded as alternating positive and negative voltages.




- The bipolar scheme was developed as an alternative to NRZ.
- The bipolar scheme has the same signal rate as NRZ, but there is no DC component.
- The NRZ scheme has most of its energy concentrated near zero frequency, which makes it unsuitable for transmission over channels.
- The concentration of the energy in bipolar encoding is around frequency *N/2*.

## 4. MULTILEVEL

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

---

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

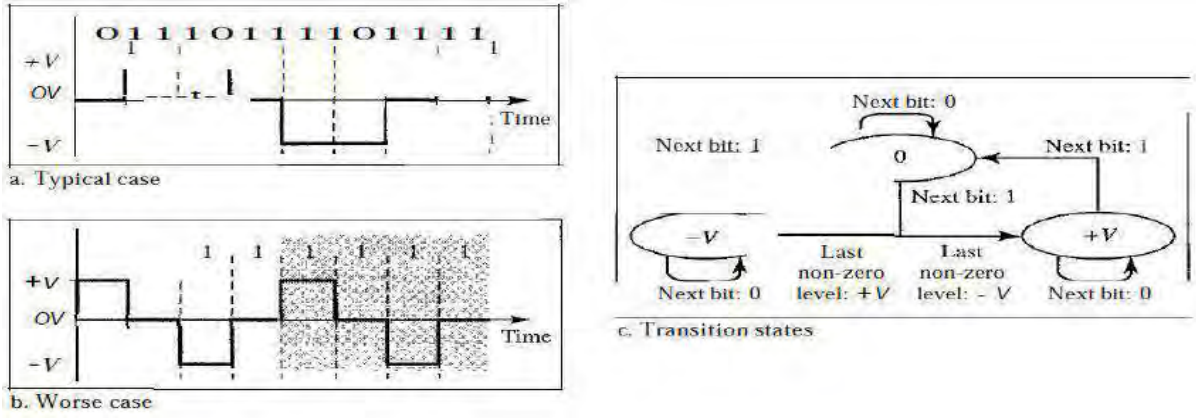
---

## 5. MULTITRANSITION

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

(1) IF THE NEXT BIT IS 0, THERE IS NO TRANSITION.
(2) IF THE NEXT BIT IS 1 AND THE CURRENT LEVEL IS NOT 0, THE NEXT LEVEL IS 0.
(3) If the next bit is 1 and the cut Tent level is 0, the next level is the opposite of the last non zero level.

***Multi transition: MLT-3 scheme***



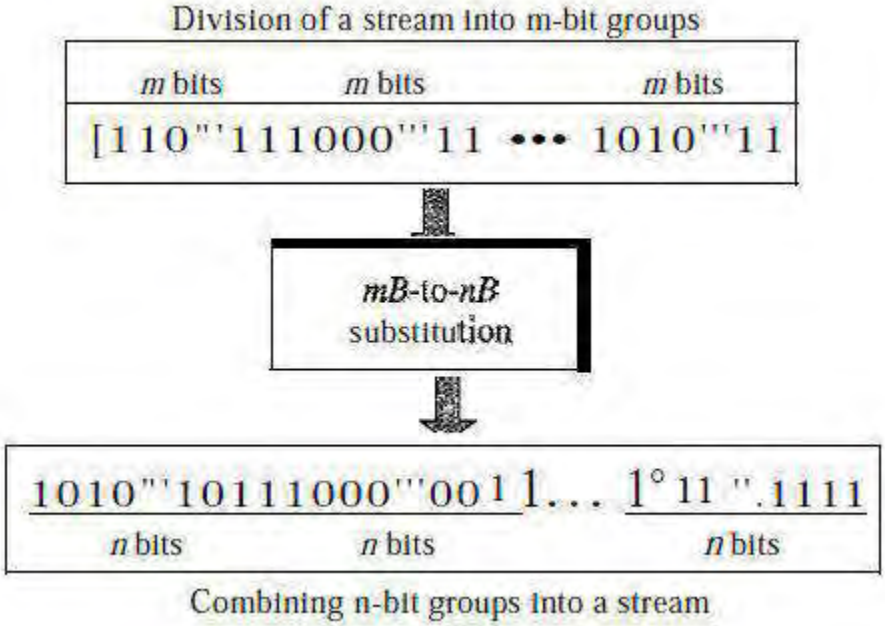

## BLOCK CODING

- Block coding can give us this redundancy and improve the performance of line coding.
- In general, block coding changes a block of *m* bits into a block of *n* bits, where *n* is larger than *m.* Block coding is referred to as an *mB/nB* encoding technique.

- The slash in block encoding (for example, 4B/5B) distinguishes block encoding from multilevel encoding (for example, 8B6T), which is written without a slash.
- Block coding normally involves three steps: **division, substitution, and combination**.

- In the division step, a sequence of bits is divided into groups of *m* bits. For example, in 4B/5B encoding, the original bit sequence is divided into 4-bit groups.

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




---

For details of 4B/5B refer text book behrouz A. forouzan in page no.116

## ANALOG TO DIGITAL CONVERSION:

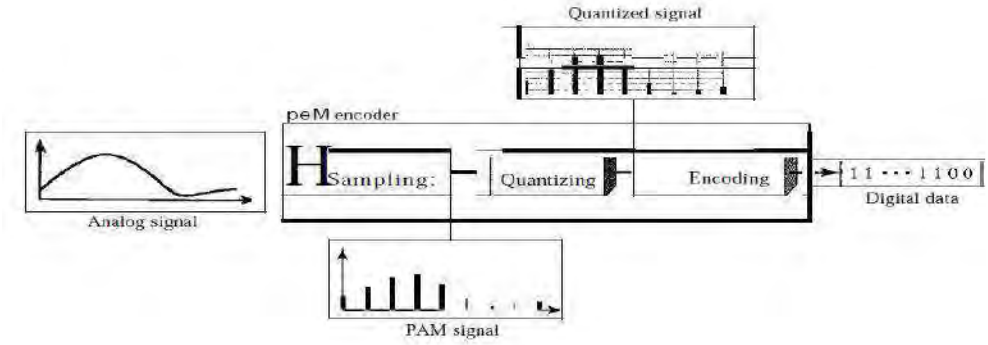
- Sometimes, we have an analog signal such as one created by a microphone or camera such as Microphones creates analog voice and camera creates analog videos, here in our case is treated is analog data.
- To transmit this analog data over digital signals we need an analog to digital conversion.
- Analog data is wave form continuous stream of data whereas digital data is discrete.
- For conversion two techniques are used, **pulse code modulation and delta modulation**.

- After the digital data are created (digitization), we can use one of the techniques described of line coding to convert the digital data to a digital signal.

## Pulse Code Modulation (PCM)

- To convert analog wave into digital data we use **Pulse Code Modulation**.
- Pulse Code Modulation is one of the most commonly used method to convert analog data into digital form.
- It involves three steps: **Sampling, Quantization and Encoding.**

***Components of PCM encoder***




*** Use PCM instead of peM on the figure**

**Steps involved are**

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

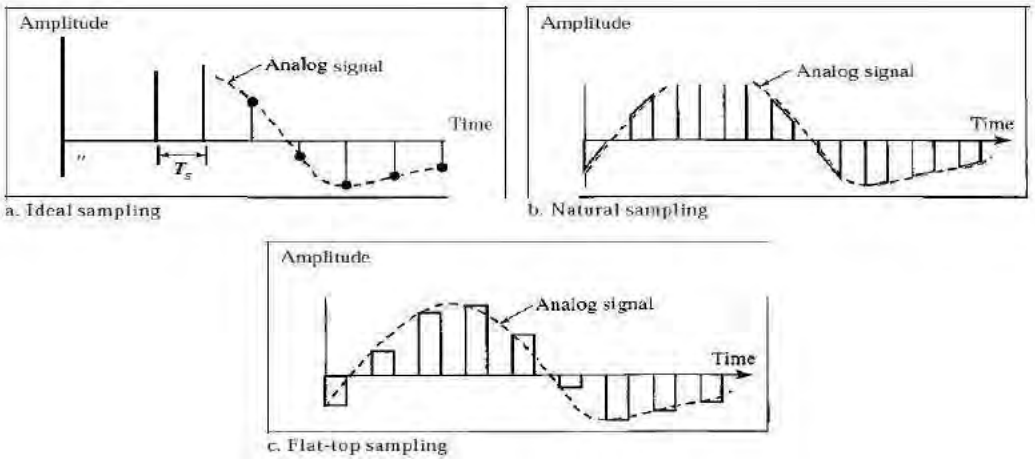
**SAMPLING**

The sampling process is sometimes referred to as pulse amplitude modulation (PAM). But, that result is still an analog signal with non integral values.

- The first step in PCM is sampling.

- The analog signal is sampled every *Ts* s, where *Ts* is the sample interval or period. The inverse of the sampling interval is called the sampling rate or sampling frequency.
- There are three sampling methods:
  - **Ideal**
  - **Natural**
  - **Flat-top**

- In **ideal sampling**, pulses from the analog signal are sampled.
- This is an ideal sampling method and cannot be easily implemented.
- In **natural sampling**, a high-speed switch is turned on for only the small period of time when the sampling occurs.
- The result is a sequence of samples that retains the shape of the analog signal.
- The most common sampling method, called **sample and hold,** however, creates **flat-top** samples by using a circuit. ***Three different sampling methods for PCM***

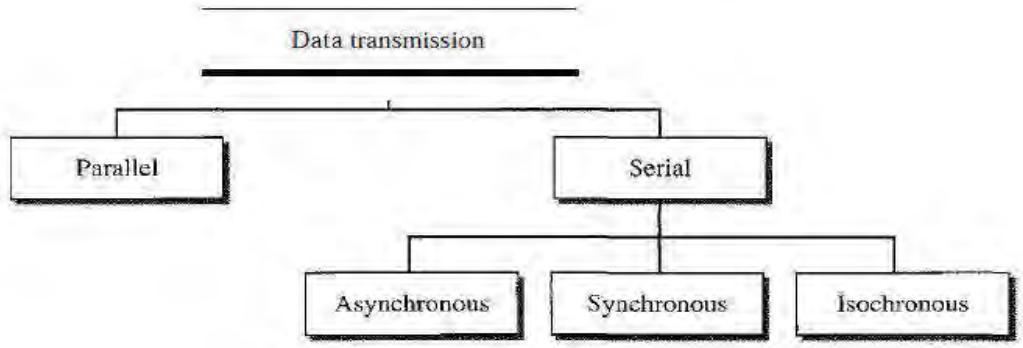



According to the Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal.

## TRANSMISSION MODE

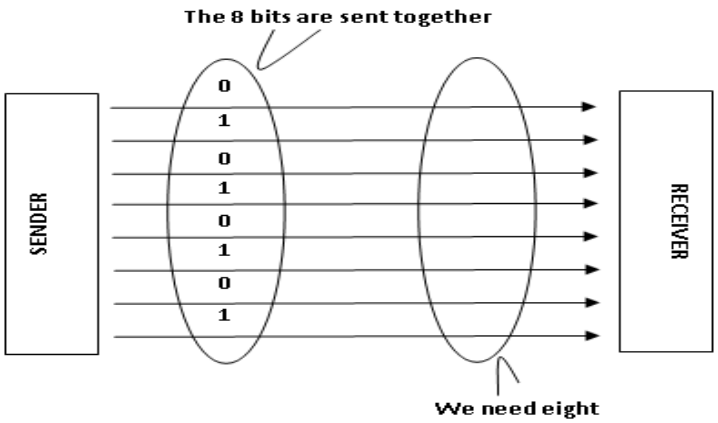
The transmission of data from one device to another is the wiring, and of primary concern when we are considering the wiring is the data stream. The transmission of binary data (0 and 1) across a link can be accomplished in either **parallel** or **serial** mode.

- In **parallel** mode, multiple bits are sent with each clock tick.
- In **serial** mode, 1 bit is sent with each clock tick. While there is only one way to send parallel data, there are three sub classes of serial transmission: **asynchronous, synchronous, and isochronous**.




## PARALLEL TRANSMISSION

Binary data, consisting of 1s and 0s, may be organized into groups of *n* bits each. Computers produce and consume data in groups of bits. By grouping, we can send data *n* bits at a time instead of 1. This is called parallel transmission.

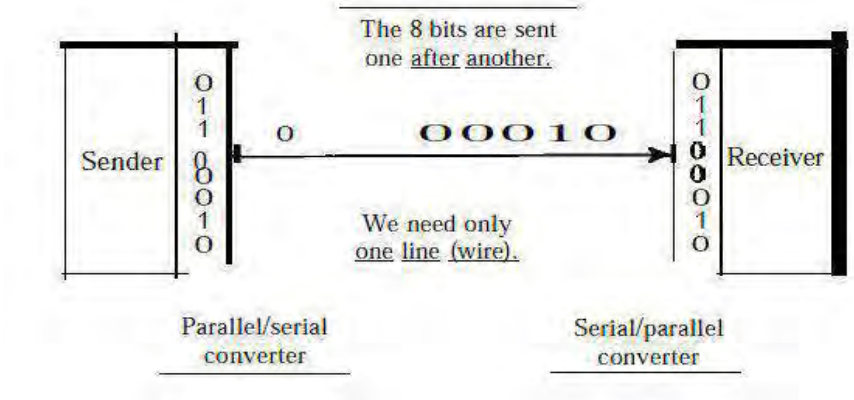


[Parallel Transmission]

- The mechanism for parallel transmission is a conceptually simple one: Use *n* wires to send *n* bits at one time.
- That way each bit has its own wire, and all *n* bits of one group can be transmitted with each clock tick from one device to another.
- The advantage of parallel transmission is **speed.**
- That is parallel transmission can increase the transfer speed by a factor of *n* over serial transmission.

- But there is a significant **disadvantage is cost:** Parallel transmission requires *n* communication lines (wires in the example) just to transmit the data stream. Because this is expensive, parallel transmission is usually limited to short distances.

## SERIAL TRANSMISSION



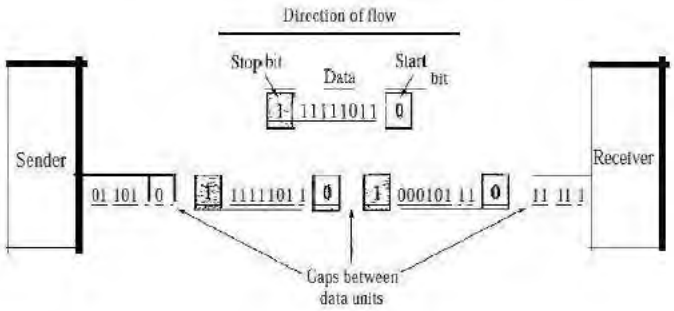
- In serial transmission one bit follows another, so we need only one communication channel rather than *n* to transmit data between two communicating devices.
- The **advantage** of **serial over parallel transmission** is that with only one communication channel, serial transmission reduces the cost of transmission over parallel by roughly a factor of *n.* Since communication within devices is parallel, conversion devices are required at the interface between the sender and the line (parallel-to-serial) and between the line and the receiver (serial-to-parallel).
  - Serial transmission occurs in one of three ways: **asynchronous, synchronous, and isochronous.**

## ASYNCHRONOUS

- Asynchronous transmission is so named because the timing of a signal is unimportant. Instead, information is received and translated by agreed upon patterns.
- Patterns are based on grouping the bit stream into bytes.
- Each group, usually 8 bits, is sent along the link as a unit.
- The sending system handles each group independently, relaying it to the link whenever ready, without regard to a timer. Without synchronization, the receiver cannot use timing to predict when the next group will arrive.
- To alert the receiver to the arrival of a new group, therefore, an extra bit is added to the beginning of each byte.
- This bit, usually a 0, is called the **start bit**. To let the receiver know that the byte is finished, 1 or more additional bits are appended to the end of the byte. These bits, usually 1s, are called **stop bits**.

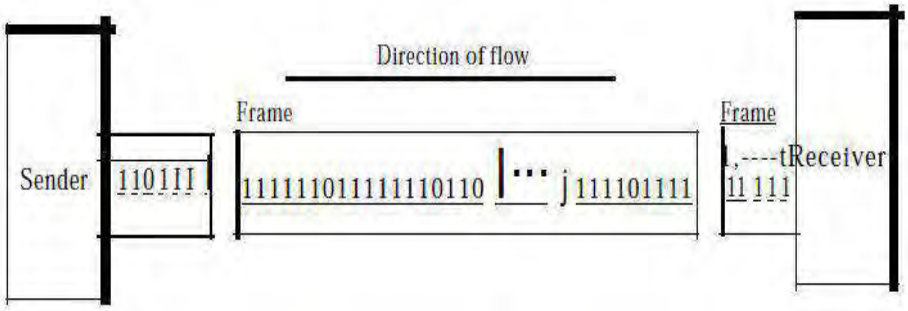
In asynchronous transmission, we send 1 start bit (0) at the beginning and 1 or more stop bits (Is) at the end of each byte. There may be a gap between each byte.

- The start and stop bits and the gap alert the receiver to the beginning and **end of each byte** allow it to synchronize with the data stream.
- This mechanism is called ***asynchronous*** because, at the byte level, the sender and receiver do not have to be synchronized.
- But within each byte, the receiver must still be synchronized with the incoming bit stream.




Asynchronous here means "asynchronous at the byte **level,"** but the bits are still synchronized; their durations are the same.

## SYNCHRONOUS




- In synchronous transmission, the bit stream is combined into longer "frames," which may contain multiple bytes.
- Each byte is introduced onto the transmission link without a gap between it and the next one.
- It is left to the receiver to separate the bit stream into bytes for decoding purposes.
- In **other words**, data are transmitted as an unbroken string of 1s and 0s, and the receiver separates that string into the bytes, or characters, it needs to reconstruct the information.
- The **advantage** of synchronous transmission is **speed**.
- With no extra bits or gaps to introduce at the sending end and remove at the receiving end, and, by extension, with fewer bits to move across the link.
- Synchronous transmission is faster than asynchronous transmission.
- For this reason, it is more useful for high-speed applications such as the transmission of data from one computer to another.

- Byte synchronization is accomplished in the data link layer.
- Although there is no gap between characters in synchronous serial transmission, there may be uneven gaps between frames.

In synchronous transmission, we send bits one after another without start or stop bits or gaps. It is the responsibility of the receiver to group the bits.

## ISOCHRONOUS

In real-time audio and video, in which uneven delays between frames are not acceptable synchronous transmission fails. For example, TV images are broadcast at the rate of 30 images per second; they must be viewed at the same rate. If each image is sent by using one or more frames, there should be no delays between frames. For this type of application, synchronization between characters is not enough; the entire stream of bits must be synchronized. The **isochronous** transmission guarantees that the data arrive at a fixed rate.

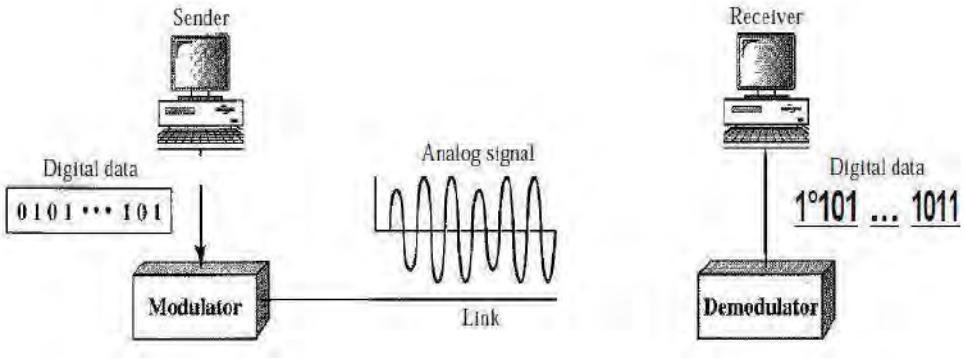
*LECTURE NOTE: 7*

# ANALOG TRANSMISSION

While digital transmission is very desirable, a low-pass channel is needed and analog transmission is the only choice if we have a band pass channel. Converting digital data to a band pass analog signal is traditionally called digital to- analog conversion. Converting a low-pass analog signal to a band pass analog signal is traditionally called analog-to-analog conversion.

## DIGITAL TO ANALOG CONVERSION

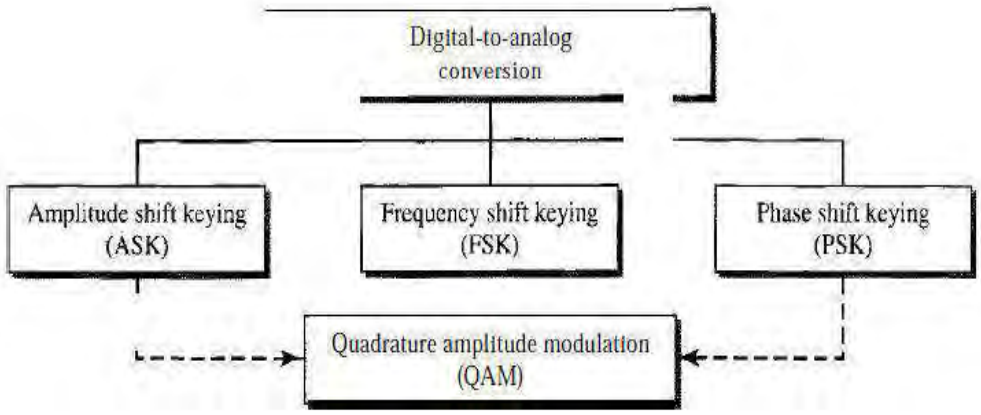
**Digital-to-analog conversion** is the process of changing one of the characteristics of an analog signal based on the information in digital data.




The above figure shows the relationship between the digital information, the digital-to-analog modulating process, and the resultant analog signal.

We have discussed that sine wave is defined by three characteristics: **amplitude, frequency, and phase.** When we vary anyone of these characteristics, we create a different version of that wave. So, by changing one characteristic of a simple electric signal, digital data is represented. Any of the three characteristics can be altered in least three mechanisms for modulating digital data into an analog signal: **amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK).** In addition, there is a fourth (and better) mechanism that combines changing both the amplitude and phase, called **quadrature amplitude** modulation (QAM). QAM is the most **efficient** of these options and is the mechanism commonly used today.

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




## ASPECTS OF DIGITAL-TO-ANALOG CONVERSION

Before we discuss specific methods of digital-to-analog modulation, basic issues must be reviewed along with the **bit** and **baud rates** and the **carrier signal these are:**

**i) Data Element vs Signal Element:** we have discussed the data element as the smallest piece of information to be exchanged, the bit and the signal element are also as the smallest unit of a signal that is constant. These terms are only little difference in digital to analog conversion.

**ii) Data Rate vs Signal Rate (Bit rate vs Baud rate):** Bit rate is the number of bits transmitted during 1 sec. Baud rate refer to the number of signal units per second that are required to represent those bits. Relationship between these two are:

**BAUD RATE= BIT RATE/NUMBER OF BITS PER SIGNAL UNIT**

Bit rate is the number of bits per second. Baud rate is the number of signal elements per second. In the analog transmission of digital data, the baud rate is less than or equal to the bit rate.

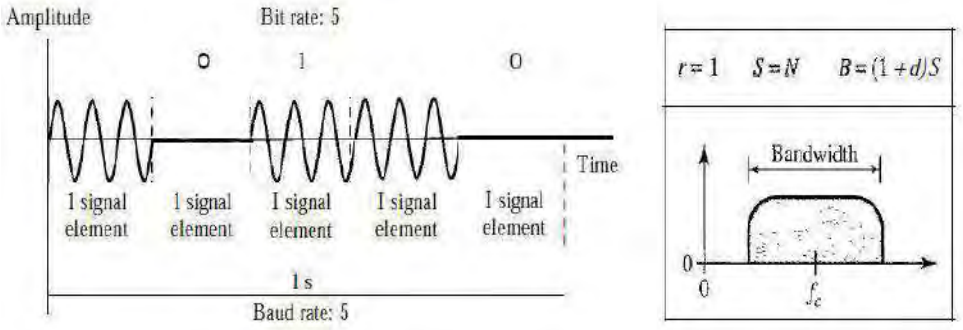
In transportation, a baud is analogous to a vehicle, and a bit is analogous to a passenger.

**iii) Bandwidth:** The required bandwidth for analog transmission of digital data is proportional to the signal rate except for FSK, in which the difference between the carrier signals needs to be added.

**iv) Carrier Signal*:*** In analog transmission, the sending device produces a high-frequency signal that acts as a **base** for the information signal. This base signal is called the **carrier signal or carrier frequency**. The receiving device is turned to the frequency of the carrier signal that it expects from the sender. Digital information then changes the carrier signal by modifying one or more of its characteristics (amplitude, frequency, or phase). This kind of modification is called **modulation (shift keying).**

## AMPLITUDE SHIFT KEYING

- In amplitude shift keying, the amplitude of the carrier signal is varied to create signal elements. Both frequency and phase remain constant while the amplitude changes.
- ASK is normally implemented using only two levels.
- This is referred to as binary amplitude shift keying or ***on-off keying*** (OOK). The peak amplitude of one signal level is 0, the other is the same as the amplitude of the carrier
  frequency.




**Bandwidth for ASK:** the bandwidth is proportional to the signal rate (baud rate). However, there is normally another factor involved, called *d*, which depends on the modulation and filtering process. The value of *d* is between 0 and 1.the relationship can be ex
Pressed as

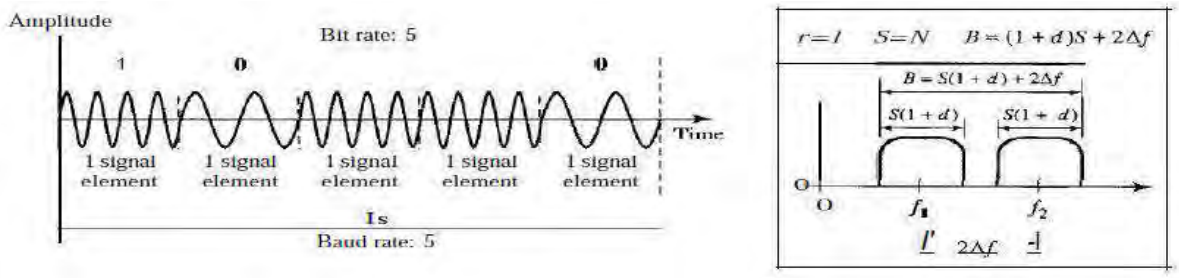
$$B = (1 + d) \times S(N_{baud})$$

*Where, B is the bandwidth, $S/N_{baud}$ is the baud rate and d is the factor related to the modulation process(with minimum value 0).*

## FREQUENCY SHIFT KEYING

In frequency shift keying, the frequency of the carrier signal is varied to represent data as 0 and 1.The frequency of the modulated signal(audio) is constant for the duration of one signal element and its depend upon bits(0 and 1), but changes for the next signal element if the data element changes. Both **peak amplitude** and **phase** remain constant for all signal elements.

***Binary FSK (BFSK)***




- Here binary FSK (or BFSK) is to consider two carrier frequencies. In the above figure we have selected two carrier frequencies, f1 and f2.
- It is assumed for first carrier the data element is 0 for second data element is 1.
- However, note that this is an unrealistic example used only for demonstration purposes.
- Normally the carrier frequencies are **very high**, and the difference between them is very **small**.

Here the middle of one bandwidth is f1 and the middle of the other is f2. Both f1 andf2 are Δf apart from the midpoint between the two bands. The difference between the two frequencies is 2Δf .

**BANDWIDTH FOR BFSK**

Carrier signals are only simple sine waves, but the modulation creates a non periodic composite signal with continuous frequencies. For FSK it can think as two ASK signals, each with its own carrier frequency f2. If the difference between the two frequencies is 2Δf, then the required bandwidth is

$$B=(l+d)xS+2\Delta f$$

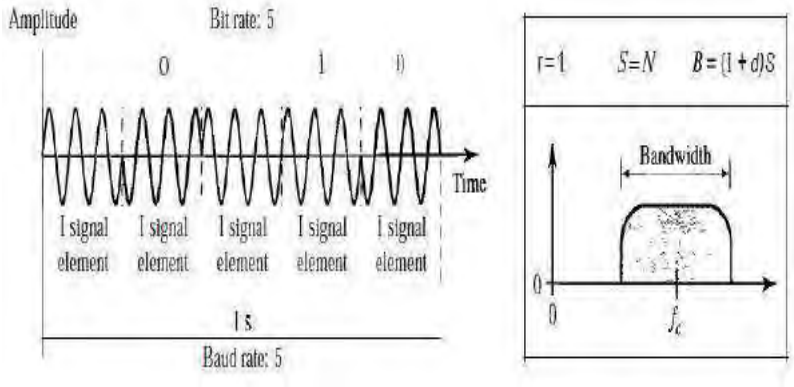
*Where, B is the bandwidth, S is the baud rate and d is the factor related to the modulation process (with minimum value 0) and* $2\Delta f$ is the frequency difference.

## PHASE SHIFT KEYING

- In phase shift keying, the phase of the carrier is varied to represent two or more different signal elements 0 and 1.
- Both peak amplitude and frequency remain constant as the phase changes.
- The phase of the signal during each bit duration is constant and its value depends on the bits (0 and 1).
- Today, PSK is more common than ASK or FSK. However QAM, which combines ASK and PSK, is the dominant method of digital-to- analog modulation.

### *BINARY PSK (BPSK OR 2-PSK)*

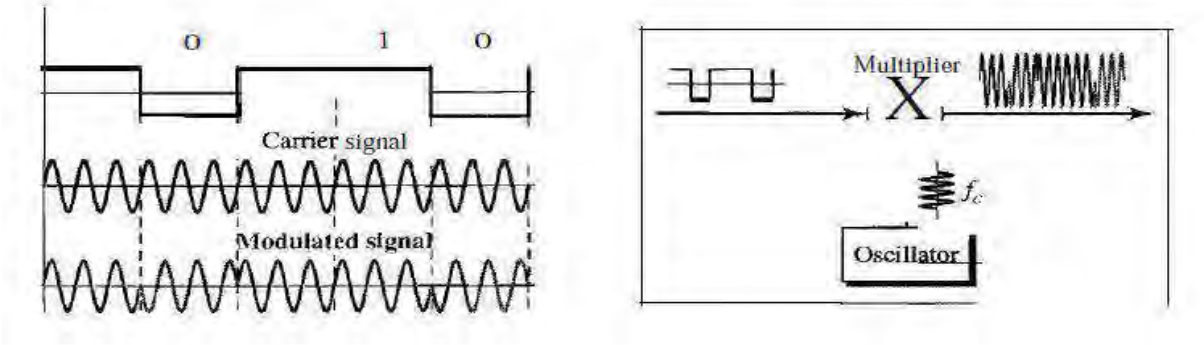
- The simplest PSK is binary PSK, in which we have used only two signal elements, one with a phase of 0°, and the other with a phase of 180°.
- Below figure shows a conceptual view of PSK and relationship of phase to bit value.
- Binary PSK is as simple as binary ASK with one big advantage-it is less susceptible to **noise**.



- In ASK, the criterion for bit detection is the **amplitude** of the signal, in PSK, it is the **phase**.
- **Noise** can change the amplitude easier than it can change the phase.
- In other words, PSK is **less susceptible to noise** than ASK.
- PSK is **superior** to FSK because we do not need two carrier signals.

## BANDWIDTH

The bandwidth for BFSK is the same as that for binary ASK, but less than that for BFSK. Here no bandwidth is **wasted** for separating two carrier signals.

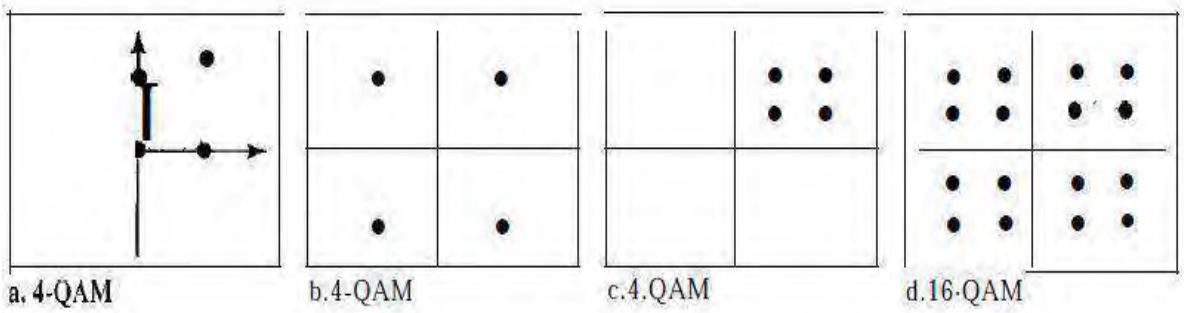



- The implementation of BPSK is as simple as that for ASK.
- The reason is that the signal element with phase 180° can be seen as the complement of the signal element with phase 0°.
- This gives us a clue on how to implement BPSK. Here it has been used same idea used for ASK but with a polar NRZ signal instead of a uni polar NRZ signal.
- The polar NRZ signal is multiplied by the carrier frequency, the 1 bit (positive voltage) is represented by a phase starting at 0°, the a bit (negative voltage) is represented by a phase starting at 180°.

## QUADRATURE AMPLITUDE MODULATION

Quadrature Amplitude Modulation, is the idea of using two carriers, one in-phase and the other quadrature, with different amplitude levels for each carrier.

***Constellation diagrams for some QAMs***




The possible variations of QAM are numerous. Below shows some of these schemes where 4-QAM scheme (four different signal element types) is a simplest one using a unipolar NRZ signal to modulate each carrier. This is the same mechanism used for ASK (OOK).

Similarly Part b shows another 4-QAM using polar NRZ, but this is exactly the same as QPSK. Part c shows another QAM-4 in which we used a signal with two positive levels to modulate each of the two carriers.

Finally, figure d shows a 16-QAM constellation of a signal with eight levels, four positive and four negative.

---

Quadrature amplitude modulation is a combination of ASK and PSK.

---

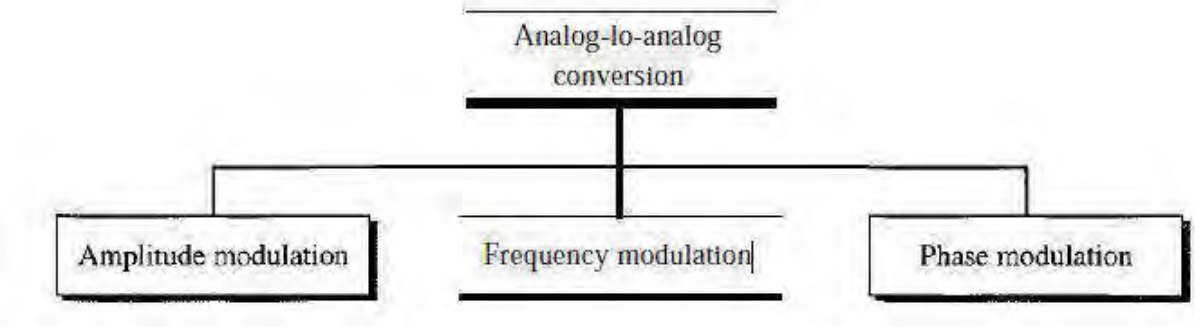
***Bandwidth for QAM***

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.

## ANALOG-TO-ANALOG CONVERSION

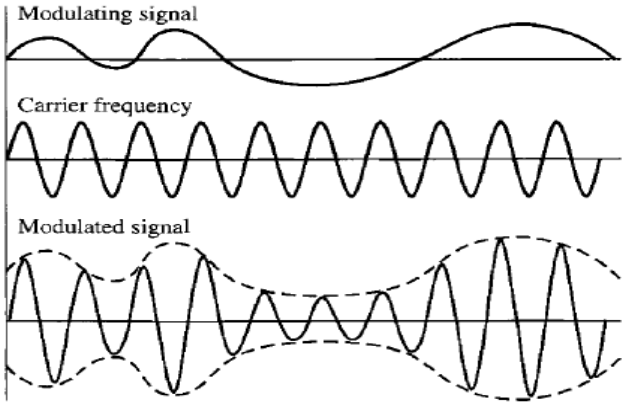
- Analog-to-analog conversion, or analog modulation, is the representation of analog information by an analog signal.
- This modulation is needed if the medium is **band pass** in nature or if only a **band pass channel** is available to us. An example is **radio**.
- The government assigns a narrow bandwidth to each radio station.
- The analog signal produced by each station is a **low-pass signal**, all in the same range.
- To be able to listen to different stations, the low-pass signals need to be **shifted**, each to a different range.

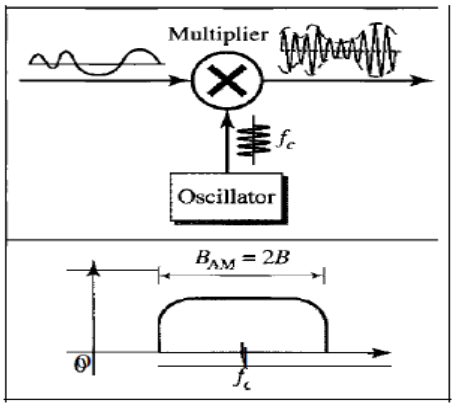
Analog-to-analog conversion can be accomplished in three ways: **amplitude modulation (AM)**, **frequency modulation (FM)**, and **phase modulation (PM).**




## AMPLITUDE MODULATION

- In **AM** transmission, the carrier signal is modulated so that its amplitude varies with the changing amplitudes of the modulating (audio) signal.
- The frequency and phase of the carrier remain the same, only the amplitude changes to follow variations in the information. Where the modulating signal is act as the envelope of the carrier.




**AM** is normally implemented by using a simple multiplier because the amplitude of the carrier signal needs to be changed according to the amplitude of the modulating signal (audio).

### *AM BANDWIDTH*

The modulation creates a bandwidth that is twice the bandwidth of the modulating signal and covers a range centered on the carrier frequency. However, the signal components above and below the carrier frequency carry exactly the same

information. For this reason, some implementations discard one-half of the signals and cut the bandwidth in half.

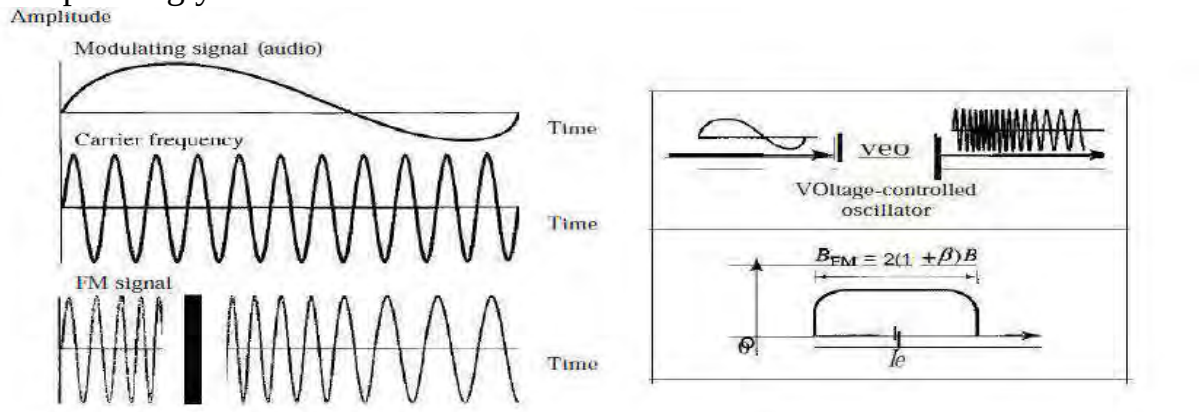
The total bandwidth required for **AM** can be determined from the bandwidth of the audio signal:

$$B_{AM} = 2B_m$$

Where, $B_{AM}$ ***bandwidth of AM signal and*** $B_m$ is bandwidth of modulating signal

## FREQUENCY MODULATION

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




The actual bandwidth is difficult to determine exactly, but it can be shown empirically that it is several times that of the analog signal or 2(1 + β)B where β is a factor depends on modulation technique with a common value of 4. In some books it is given that:

### *FM BANDWIDTH*

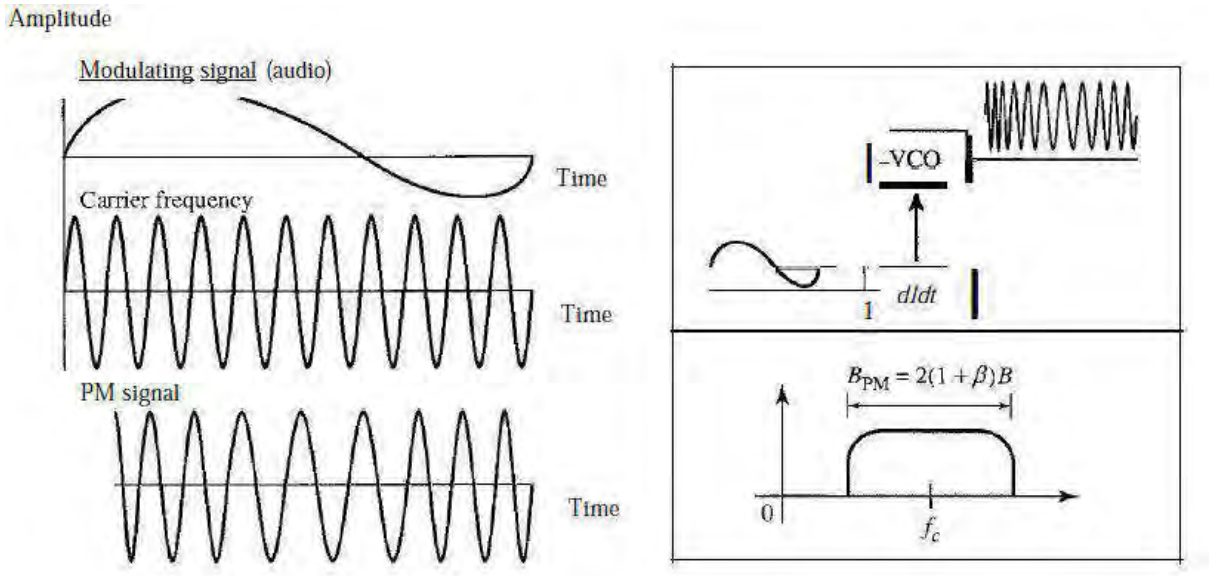
The total bandwidth required for FM can be determined from the bandwidth of the **audio** signal as : $B_{FM} = 10xB_m$

**Where**, $B_{FM}$ is bandwidth of FM signal and $B_m$ is the bandwidth of modulating signal.

## PHASE MODULATION

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

It has been proved that PM is the same as FM with one difference. In FM, the instantaneous change in the **carrier frequency** is proportional to the **amplitude** of the modulating signal where as in PM the instantaneous change in the carrier frequency is proportional to the derivative of the **amplitude** of the modulating signal.




As the above figure shows, PM is normally implemented by using a voltage-controlled oscillator along with a derivative. The frequency of the oscillator changes according to the derivative of the input voltage which is the amplitude of the modulating signal.

***PM BANDWIDTH***

- The actual bandwidth is difficult to determine exactly, but it can be shown empirically that it is several times that of the analog signal.
- Although, the formula shows the same bandwidth for FM and PM, the value of β is lower in the case of PM (around 1 for narrowband and 3 for wideband).

**The total bandwidth required for PM can be determined from the bandwidth and maximum amplitude of the modulating signal:**

$$B_{pm} = 2(1 + \beta)B.$$

***LECTURE NOTE: 8***

# MULTIPLEXING

The main challenge of electronic communication is wise use of given bandwidths of channel. However, *wise* may depend on the application. Here we explore two broad categories of bandwidth utilization: **multiplexing** and **spreading**. In multiplexing, our goal is **efficiency** where, we combine several channels with low bandwidth together into one with large bandwidth. In **spreading**, our goals are privacy and **anti jamming** where, we expand the bandwidth of a channel to insert redundancy, which is necessary to achieve these goals of accuracy.

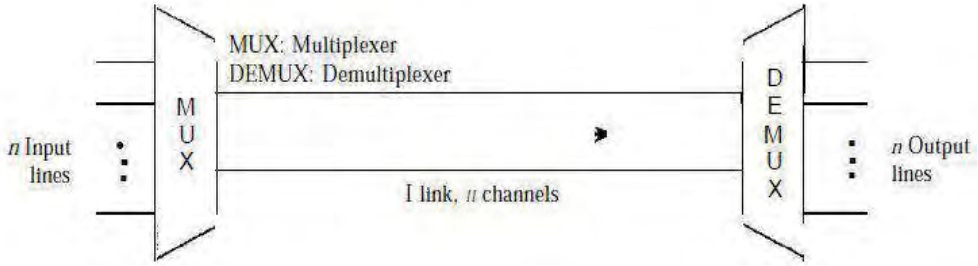
- **It** combines **multiple signals (analog or digital)** for transmission over a **single line or media**.
- A common type of multiplexing combines several **low-speed signals** for transmission over a **single high-speed connection**.
- Multiplexing is done by using a device called multiplexer (MUX) that combines n input lines to generate one output line i.e. (many to one). Therefore multiplexer (MUX) has several inputs and one output.

- At the receiving end, a device called demultiplexer (DEMUX) is used that separates signal into its component signals. So DEMUX has one input and several outputs.

- Multiplexing is a popular networking technique that integrates multiple analog and digital signals into a signal transmitted over a shared medium.

- Multiplexers and de-multiplexers are used to convert multiple signals into one signal.
- If the bandwidth of a link is greater than the bandwidth needs of the devices connected to it, the bandwidth is wasted. To make efficient use several channels are share. It is process of making the most effective use of the available channel capacity.

- The most common use of multiplexing is in long-haul communication using coaxial cable, microwave and optical fiber.

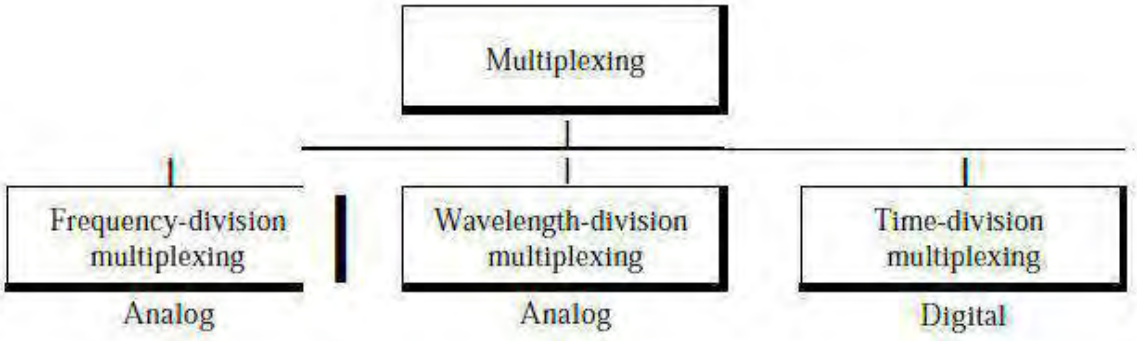



- In a multiplexed system, *n* lines share the bandwidth of one link where the lines on the left direct their transmission streams to a multiplexer (MUX), which combines them into a single stream **(many-to-one).**

- Similarly at the receiving end, that stream is fed into a de multiplexer (DEMUX), which separates the stream back into its component transmissions **(one-to-many)** and directs them to their corresponding lines.

- In the figure, the word **link** refers to the physical path. The word **channel** refers to the portion of a **link** that carries a transmission between a given pair of lines. One **link** can have **many *(n)*** channels.
- Phone calls are a good example of multiplexing in telecommunications. That is, more than one phone call is transmitted over a single medium.

- Multiplexing techniques can be categorized into the following three types:

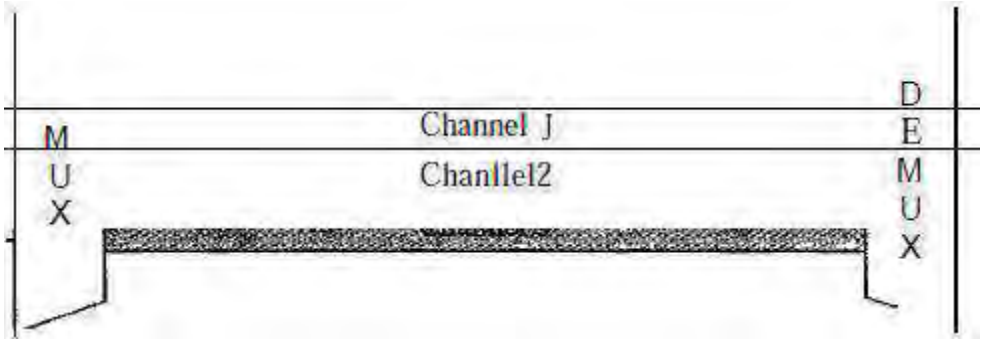


**Frequency-division multiplexing (FDM)**

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

- Signals generated by each sending device modulate different carrier frequencies. These modulated signals are then combined into a single composite signal that can be transported by the link.

- Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal Here the frequency spectrum is divided into several logical channels, giving each user exclusive possession of a particular frequency band.




- Channels can be separated by strips of unused bandwidth called **guard bands** to prevent signals from overlapping.

- In addition, carrier frequencies must not interfere with the original data frequencies.

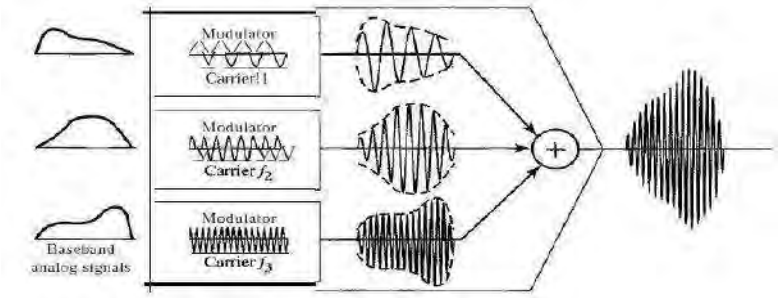
- Although FDM is consider to be an analog multiplexing technique, a digital signal can also be converted to an analog signal.

**Multiplexing Process**

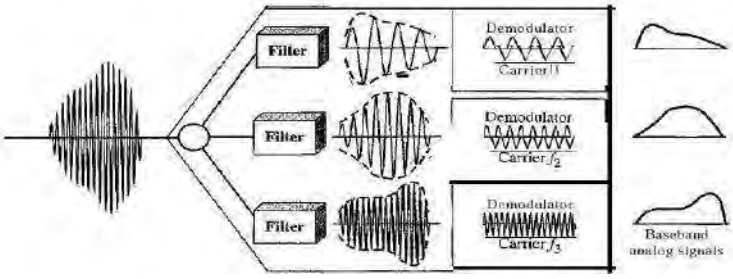
- Each source generates a signal of a similar frequency range.

- Inside the multiplexer, these similar signals modulates different carrier frequencies *(f1,f2* and f3*).*

- The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.




**DE MULTIPLEXING PROCESS**




- The demultiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals.

- The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines.

- FDM is used in many telephone companies, switched or leased lines To maximize the efficiency of their infrastructure.

- A very common application of FDM is AM and FM radio broadcasting. Radio uses the air as the transmission medium. A special band from 530 to 1700 kHz is assigned to AM radio.

- The first generation of cellular telephones (still in operation) also uses FDM. Each user is assigned two 30-kHz channels, one for sending voice and the

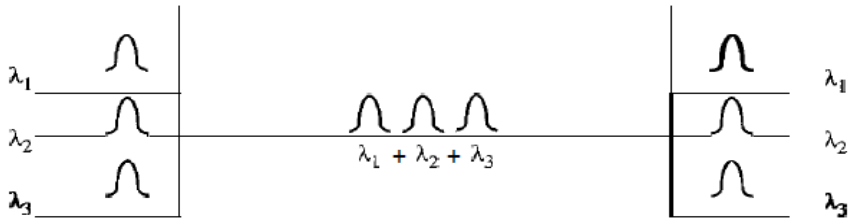
other for receiving. The voice signal, which has a bandwidth of 3 kHz (from 300 to 3300 Hz), is modulated by using FM.

**Note**: FM signal has a bandwidth 10 times that of the modulating signal

- FDM can be implemented very easily as in the radio and television broadcasting; there is no need for a physical multiplexer or de multiplexer.

**Wavelength-Division Multiplexing**

- Wavelength-division multiplexing (WDM) is designed to use the high-data-rate capability of fiber-optic cable.
- The optical fiber data rate is higher than the data rate of other metallic transmission cable.
- Using a fiber-optic cable for one single line wastes the available bandwidth.
- Multiplexing idea allows us to combine several lines into one and utilized properly.
- Concept of WDM is same as FDM, except that the multiplexing and de multiplexing, here involve optical signals transmitted through fiber-optic channels. And here also combining different signals of different frequencies, where frequencies are **very high**.

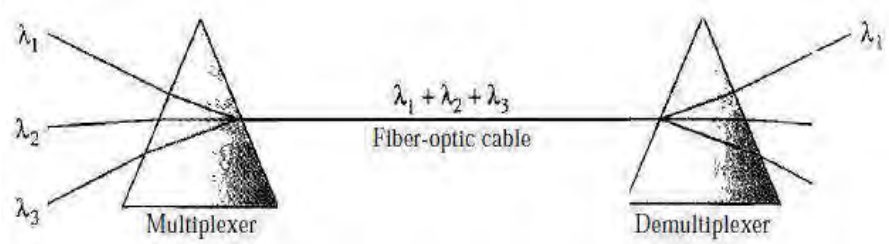



- Very narrow bands of light from different sources are combined to make a wider band
  of light.
- At the receiver, the signals are separated by the de multiplexer. The basic idea is that combine multiple light sources into one single light at the multiplexer and do the reverse at the de multiplexer.

- One of the common examples is the combining and splitting of light sources is easily handled by a prism.

- Using this technique, a multiplexer can be made to combine several input beams of light, each containing a narrow band of **frequencies**, into one output beam of a **wider** band of **frequencies**. A **demultiplexer** can also be made to reverse the process.

***Prisms in wavelength-division multiplexing and de multiplexing***




- One application of WDM is the **SONET** network in which multiple optical fiber lines are multiplexed and de multiplexed.

- A new method, called dense WDM (DWDM), can multiplex a very large number of channels by spacing channels very close to one another.

- It achieves even greater efficiency.

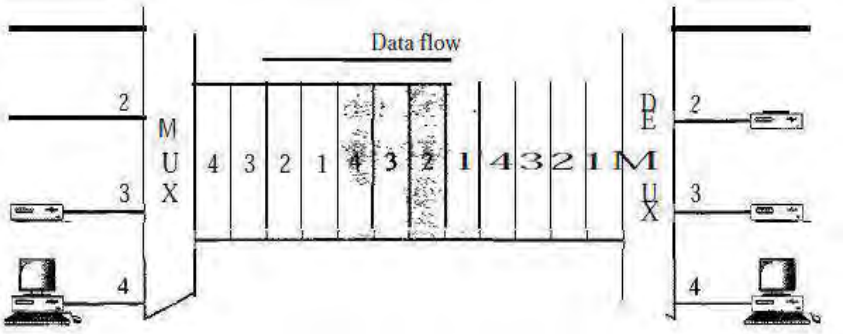
***TIME-DIVISION MULTIPLEXING (TDM)***

- It is also called synchronous TDM, which is commonly used for multiplexing digitized voice stream.

- The users take turns using the entire channel for short burst of time and that allows several connections to share the high bandwidth of a line Instead of sharing a portion of the bandwidth as in FDM, likewise time is shared.

- Each connection occupies a portion of time in the link.

- **Note** that the same link is used as in FDM; here, however, the link is shown sectioned by time rather than by frequency.



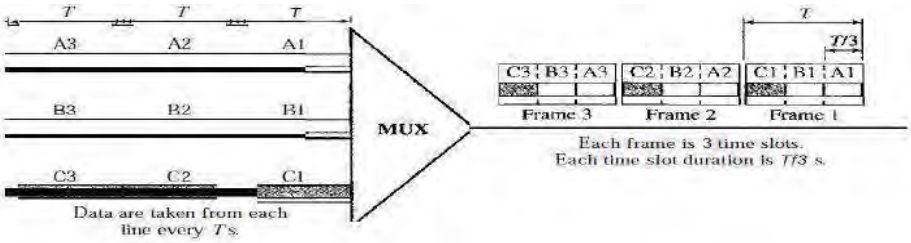

- In the figure, portions of signals 1, 2, 3, and 4 occupy the link sequentially.

- Here we concerned with only multiplexing, not switching. This means that all the data in a message from source 1 always go to one specific destination, be it 1, 2, 3, or 4 where, delivery is fixed and unvarying, unlike switching.

- Digital data from different sources are combined into one timeshared link. However, this does not mean that the sources cannot produce analog data; analog data can be sampled, changed to digital data, and then multiplexed by using TDM.

- We can divide TDM into two different schemes: **synchronous** and **statistical (refer book) TDM**.

- In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot.




- A unit can be 1 bit, one character, or one block of data. Each input unit becomes one output unit and occupies one output time slot. However, the duration of an output time slot is ***n* times shorter** than the duration of an **input time slot**. If an input time slot is *T* s, the output time slot is *T/n* s, where *n* is the number of connections.

- If we have *n* connections, a frame is divided into *n* time slots and one slot is allocated for each unit, one for each input line and the duration of each frame is *T*

- The data rate of the output link must be *n* times the data rate of a connection to guarantee the flow of data.

- Above figure shows an example of synchronous TDM where *n* is 3.

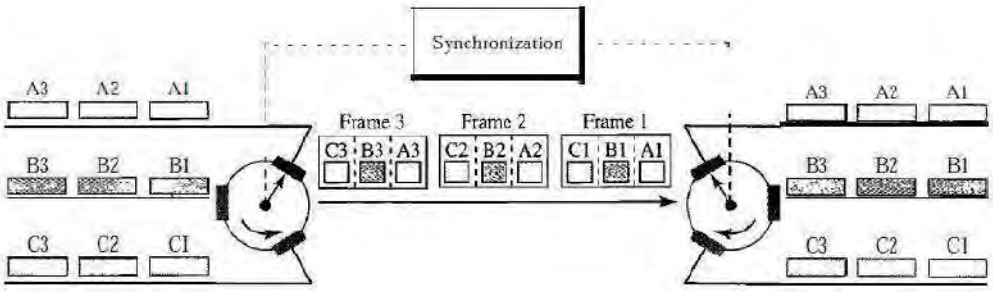
---

In synchronous TDM, the data rate of the link is *n* times faster, and the unit duration is *n* times shorter.

---

### *INTERLEAVING*

TDM can be visualized as two fast-rotating switches, one on the multiplexing side and the other on the de multiplexing side. The switches are synchronized and rotate at the same speed, but in opposite directions where on the multiplexing side, as the switch opens in front of a connection, send a unit onto the path. This process is called **interleaving.** On the de multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.




**Interleaving process** for the connection shown in Figure assume that no switching is involved and that the data from the first connection at the multiplexer site go to the first connection at the de multiplexer.

***LECTURE NOTE: 9***

# TRANSMISSION MEDIA

- **Transmission medium** can be broadly defined as anything that can carry information from a source to a destination.

- For example, the transmission medium for two people having a dinner conversation is the air. The air can also be used to convey the message in a smoke signal or semaphore.

- A **transmission medium** (**plural *transmission media***) is a material substance (**solid, liquid, gas, or plasma, such as free space (air, vacuum, and water), metallic cable, or fiber-optic cable**.) which can propagate energy waves or in other words transmission media is a path/ means through which data transmission possible from one place to another.

- For example, the transmission medium for sound received by the ears is usually air, but solids and liquids may also act as transmission media for sound.

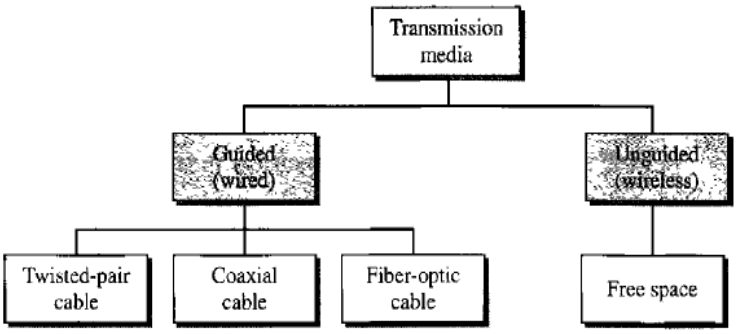
- Transmission media are the physical pathways that connect computers, other devices, and people on a network.

- Transmission media are actually located below the physical layer and are directly controlled by the physical layer.

- You could say that transmission media belong to layer zero.

- A transmission medium can be classified as:

- **GUIDED MEDIA**
- **UNGUIDED MEDIA**




## GUIDED MEDIA

- Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.
- A signal travelling along any of these media is directed and contained by the physical limits of the medium.

- Twisted-pair and coaxial cable use metallic (copper) conductors that accept and transport signals in the form of electric current.

- Optical fiber is a cable that accepts and transports signals in the form of light. Three common types of guided media are used described below as

- **Twisted pair cable**
- **Coaxial Cable.**
- **Fiber optics**

## TWISTED PAIR CABLE

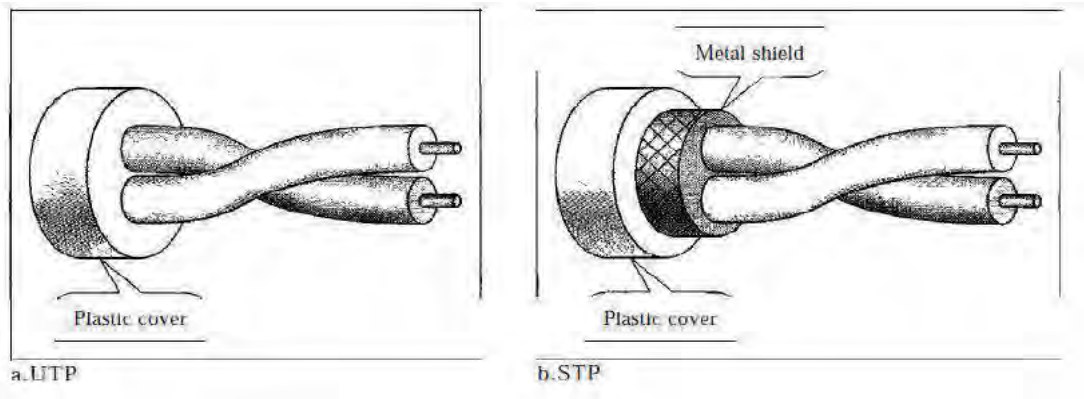
- The most popular network cabling is twisted pair cable.
- It is light weight, easy to install, inexpensive and support many different types of network.
- It also supports the speed of **100 mps.**
- Twisted pair cabling is made of pairs of solid or stranded copper twisted along each other.
- A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together, One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference.
- The receiver uses the difference between the two.


**Fig.: Twisted pair wire**

- If the two wires are parallel, the effect of these unwanted signals is not the same in both wires because they are at different locations relative to the noise or crosstalk sources (e,g.one is closer and the other is farther). This results in a difference at the receiver.
- By twisting the pairs, a balance is maintained.
- For example, suppose in one twist, one wire is closer to the noise source and the other is farther; in the next twist, the reverse is true
- Twisting makes it probable that both wires are equally affected by external influences (noise or crosstalk).
- This means that the receiver, which calculates the difference between the two, receives no unwanted signals which are canceled out.

## *UNSHIELDED VERSUS SHIELDED TWISTED-PAIR CABLE*

The most common twisted-pair cable used in communications is **unshielded twisted-pair (UTP).** It can be either voice grade or data grade depending on the condition. It is less expensive than **STP** and easily available. IBM has produced a version of twisted-pair cable for its use called **shielded twisted-pair** (STP). STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive, provide higher transmission rate as well as STP is seldom used outside of IBM. Figure below shows the difference between UTP and STP.

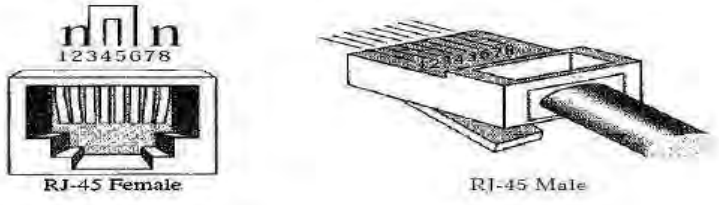



### *Categories*

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

### *CONNECTORS*

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



***Applications***

- Twisted-pair cables are used in telephone lines to provide voice and data channels.

- The DSL lines that are used by the telephone companies to provide high-data-rate connections also use the high-bandwidth capability of unshielded twisted-pair cables.

- Local-area networks, such as 10Base-T and 10Base-T, also use twisted-pair cables

## COAXIAL CABLE

Coaxial cable (or ***coax)*** carries signals of higher frequency ranges than those in twisted pair cable, because the two media are constructed quite differently. Coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two. The outer metallic wrapping, serves 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. Coaxial cable is very common & widely used commutation media. For example TV wire is usually coaxial. Coaxial cable gets its name because it contains two conductors that are parallel to each other.

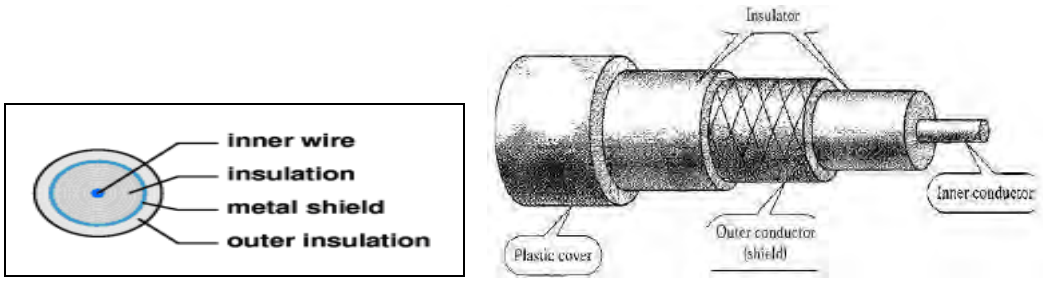



**Figure: Cross-section of a coaxial cable**

## *COAXIAL CABLE STANDARDS*

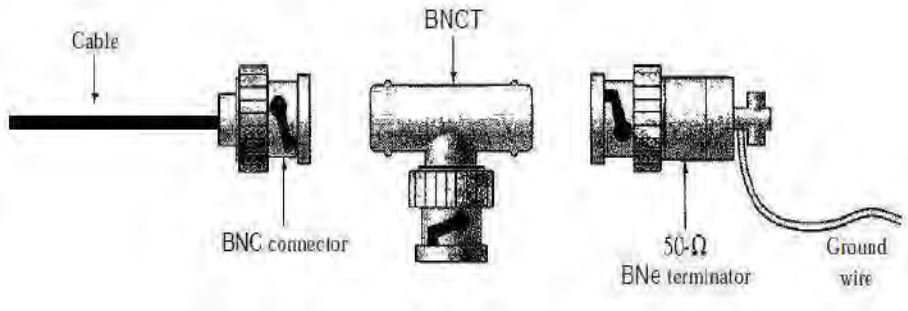
Coaxial cables are categorized by their **radio government (RG)** ratings. Each RG number denotes a unique set of physical specifications. Specification including the, wire gauge of the **inner conductor**, the **thickness** and **type of the inner insulator**, the construction of the **shield**, and the **size** and t**ype** of the outer casing.

Categories of coaxial cables

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

## *CONNECTORS*

The most common type of connector used today is the Bayone-Neill-Concelman (BNe), connector. Figure below shows three popular types of these connectors: the **BNC** connector, the **BNC T** connector and the **BNC** terminator.




The BNC connector is used to connect the end of the cable to a device, such as a TV set. The BNC T connector is used in Ethernet networks to branch out to a connection to a computer or other device. The BNC terminator is used at the end of the cable to prevent the reflection of the signal.

***Performance***

Attenuation is much higher in coaxial cables than in twisted-pair cable. In other words, although coaxial cable has a much higher bandwidth, the signal weakens rapidly and requires the frequent use of repeaters.

***Applications***

- Coaxial cable was widely used in analog telephone networks where a single coaxial network could carry 10,000 voice signals Later it was used in **digital telephone** networks where a it could carry digital data up to 600 Mbps.
- However, coaxial cable in telephone networks has largely been replaced today with fiber-optic cable.
- Cable TV networks also use coaxial cables.
- In the traditional cable TV network, the entire network used coaxial cable.
- Hybrid networks use coaxial cable only at the network boundaries, near the consumer premises.
- Another common application of coaxial cable is in traditional Ethernet LANs

**ADVANTAGES**

- Inexpensive
- Easy to wire
- Moderate level of EMI immunity

**DISADVANTAGE**

- Single cable failure can fail or take down an entire network.

## FIBER OPTIC CABLE (FIBER OPTICS)

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. 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.

Fiber optic cable uses electrical signals to transmit data. In fiber optic cable light only moves in one direction for two way communication to take place a second connection must be made between the two devices. It is actually two stands of cable.

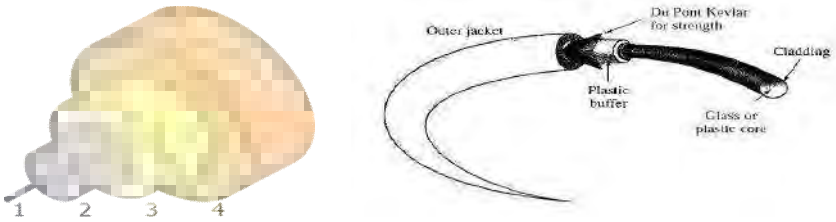
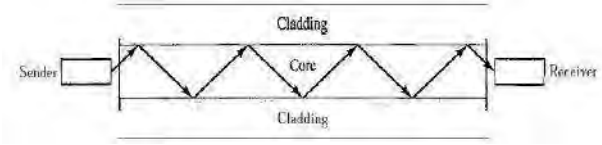


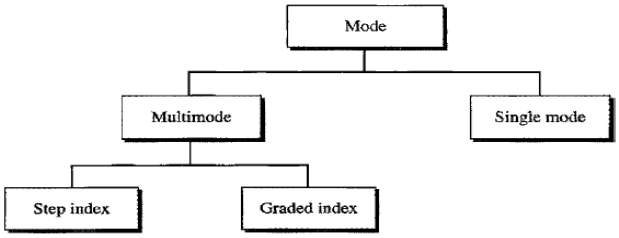

**Figure: Structure of fiber optic cable**

### *Propagation Modes*

Current technology supports two modes (**multimode and single mode**) for propagating light along optical channels. **Multimode** can be implemented in two forms: **step-index** or **graded-index.**




### Propagation mode



- In **Multimode** multiple beams from a light source move through the core in different paths. These beams move within the cable depends on the structure of the core.

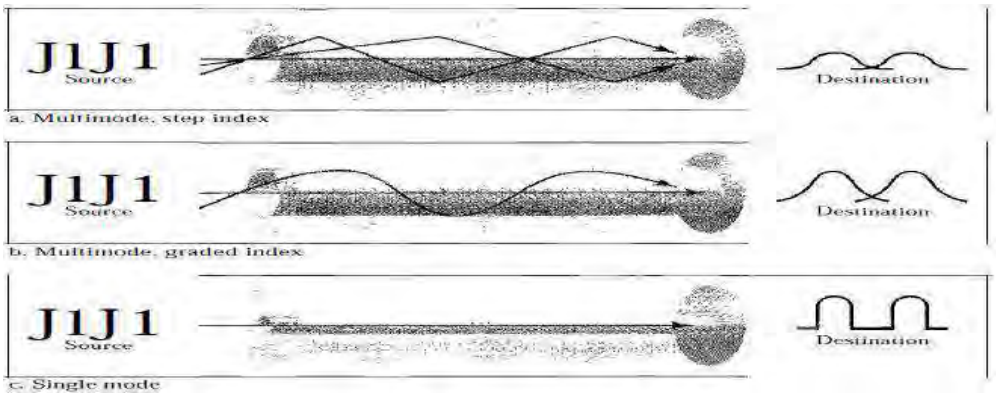
- In **multimode step-index** fiber, the density of the core remains constant from the center to the edges. A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change due to a lower density; this alters the angle of the beam's motion. The term ***step index*** refers to the **suddenness** of this change.

- The **multimode graded-index** fiber decreases the distortion of the signal through the cable. The word ***index*** here refers to the index of refraction where, the 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.

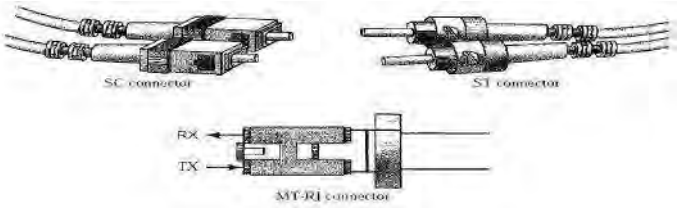
The **single mode** fiber is manufactured with a much smaller diameter than that of multimode fiber, and with substantially lower density (index of refraction). Single-mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles.

The decrease in density results in a critical angle that is close to 90° to make the propagation of beams almost horizontal. In this case, propagation of different beams is almost identical, and delays are negligible. All the beams arrive at the destination "**together**" and can be recombined with little distortion to the signal.




## *CONNECTORS*

There are three types of connectors for fiber-optic cables, as shown in Figure




The **subscriber channel** (SC) **connector** is used for cable TV. It uses a push/pull locking system. The **straight-tip** (ST) **connector** is used for connecting cable to networking devices. It uses a bayonet locking system and is more reliable than SC.

**MT-RJ** is a connector that is the same size as RJ45.

### *Performance*

In fiber-optic cable attenuation is flatter than in the case of twisted-pair cable and coaxial cable. The performance is such that we need fewer (actually 10 times less) **repeaters** when we use fiber-optic cable.

### *Applications*

- Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective.

- Some cable TV companies use a combination of optical fiber and coaxial cable, thus creating a hybrid network.

***Advantages***

- It supports **higher bandwidth** than either twisted-pair or coaxial cable.
- **Less signal attenuation** since Fiber-optic transmission distance is significantly greater and hence signal can run 50 km without requiring regeneration.
- **Immunity to electromagnetic interference:** Electromagnetic noise cannot affect fiber-optic cables.
- **Resistance** to corrosive materials.
- **Light weight:** Fiber-optic cables are much lighter than copper cables.
- **Greater immunity to tapping**: Fiber-optic cables are more immune to tapping than copper cables.

**Disadvantages**

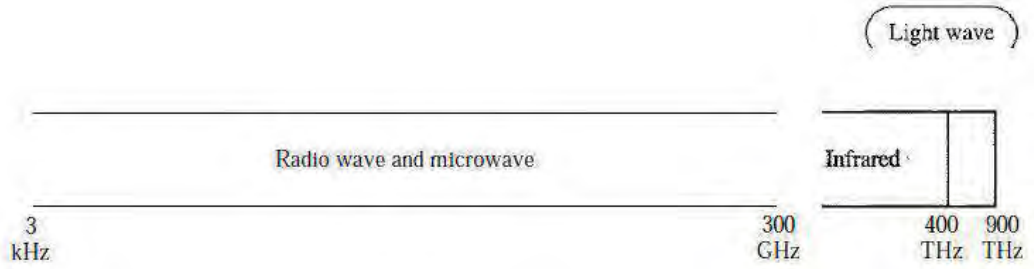
- **Installation and maintenance:** Fiber-optic cable is a relatively new technology that's why its installation and maintenance require expertise that is not yet available everywhere.
- **Unidirectional light propagation** here Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.
- **Cost:** The cable and the interfaces are relatively more expensive than those of other guided media.

***LECTURE NOTE: 10***

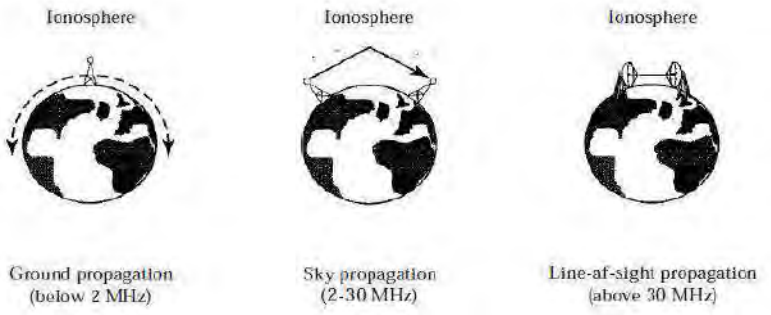
# UNGUIDED MEDIA

- **Unguided transmission media** are methods that allow the transmission of data without the use of physical means/path or conductor to transmit data.
- This type of communication is often referred as wireless communication where Signals are normally broadcast through free space.
- Unguided media provide a means for transmitting data in the form of electromagnetic waves but do not guide them.
- Examples are propagation through air, vacuum and sea water.
- For wireless communication the electromagnetic spectrum, ranging from 3 kHz to 900 THz.

*Figure:* ***Electromagnetic spectrum for wireless communication***




Unguided signals can travel from the source to destination in several ways: **ground propagation, sky propagation, and line-of-sight propagation.**




- In **ground propagation**, radio waves travel through the lowest portion of the atmosphere, with hugging the earth.

- These low-frequency signals emanate in all directions from the transmitting antenna and follow the curvature of the planet.
- Distance cover is depends on the amount of power in the signal as the **greater** the **powe**r, the **greate**r the **distance**.

- In **sky propagation**, higher-frequency radio waves radiate upward into the ionosphere (the layer of atmosphere where particles exist as ions) where they(waves) are reflected back to earth.

- This type of transmission allows for greater distances with lower output power.

- 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.

- The electromagnetic spectrum defined as radio waves and microwaves which are divided into eight ranges, called ***bands***, each are regulated by government authorities.

- These bands are rated from *very low frequency* (VLF) to ***extremely high frequency*** (EHF).

- Table listed these bands, their ranges, propagation methods, and some applications.

| *Band* | *Range* | *Propagation* | *Application* |
|---|---|---|---|
| VLF (very low frequency) | 3-30 kHz | Ground | Long-range radio navigation |
| LF (low frequency) | 30-300 kHz | Ground | Radio beacons and navigational locators |
| MF (middle frequency) | 300 kHz-3 MHz | Sky | AM radio |
| HF (high frequency) | 3-30 MHz | Sky | Citizens band (CB), ship/aircraft communication |
| VHF (very high frequency) | 30-300 MHz | Sky and line-of-sight | VHF TV, FM radio |
| UHF (ultrahigh frequency) | 300 MHz-3 GHz | Line-of-sight | UHF TV, cellular phones, paging, satellite |
| SHF (superhigh frequency) | 3-30 GHz | Line-of-sight | Satellite communication |
| EHF (extremely high frequency) | 30-300 GHz | Line-of-sight | Radar, satellite |

We divide wireless transmission into some category as:

- **microwave**
- **radio waves**
- **infrared waves**

**Microwave**

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

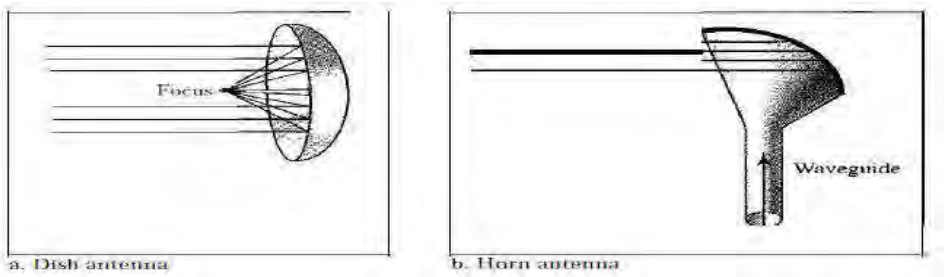
- **Microwave propagation is line-of-sight.**
    - To make mounted antennas to be in direct sight of each other towers that are far apart need to be very tall.
    - The curvature of the earth as well as other blocking obstacles do not allow two **short** towers to communicate by using

microwaves it means with the short tower it is not possible to transmit microwaves.

- Repeaters are often needed for long distance communication.

- **Very high-frequency** microwaves cannot penetrate walls if receiver is inside the building, this is the disadvantage.
- The **microwave band is relatively wide**, almost 299 GHz, therefore high data rate is possible with higher sub bands.
- Use of certain portions of the band requires permission from authorities.

***Unidirectional Antenna***




- Microwaves need unidirectional antennas that send out signals in one direction.

- Two types of antennas are used for microwave communications: the **parabolic dish** and the **horn**.

- A parabolic dish antenna is based on the geometry of a parabola in which every line parallel to the line of symmetry (line of sight) reflects off the curve at angles such that all the lines intersect in a common point called the **focus**.

- The parabolic dish works as a funnel, catching a wide range of waves and directing them to a common point (focus).
- Outgoing transmissions are broadcast through a **horn** aimed at the dish. The microwaves hit the dish and are deflected outward in a reversal of the receipt path.

- A **horn** antenna looks like a **gigantic scoop**. Outgoing transmissions are broadcast up a stem (resembling a handle) and deflected outward in a series of narrow parallel beams by the curved head.

- Received transmissions are collected by the scooped shape of the horn, in a similar manner to the parabolic dish, and are deflected down into the stem.

***Applications***

- Microwaves, due to their unidirectional properties, are very useful in uncast (One-to-one) communication.
- They are used in cellular phones, satellite networks and wireless LANs.

**Radio waves**

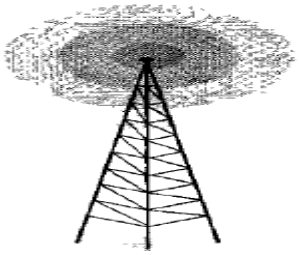
- Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz or frequency below to visible light are called radio waves.
- Flow of signals is omnidirectional when an antenna transmits radio waves, they are propagated in all directions where the sending and receiving antennas do not have to be aligned.
- The omnidirectional property has a disadvantage that the radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.
- Radio waves particularly propagate in the sky mode and can travel long distances. This makes radio waves a good candidate for long-distance broadcasting such as AM radio.

- Radio waves, particularly those of low and medium frequencies, can penetrate walls. This characteristic can be both an advantage and a disadvantage.

- It is an advantage because, for example, an AM radio can receive signals inside a building.
- It is a disadvantage because we cannot isolate a communication to just inside or outside a building.
- The radio wave band is relatively narrow, just under 1 GHz, compared to the microwave band.
- When this band is divided into sub bands, the sub bands are also narrow, leading to a low data rate for digital communications.

- Almost the entire band is regulated by authorities (e.g., the FCC in the United States) for using any part of the band requires permission from the authorities.

*Omnidirectional Antenna*



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

***Applications***

The omnidirectional characteristics of radio waves make its useful for multicasting, Where AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

---

Radio waves are used for multicast communications, such as radio and television, and paging systems.

---

**INFRARED**

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

- One common example is infrared remote control, when used; it does not interfere with the use of the remote by our neighbors. However, this same characteristic makes infrared signals useless for long-range communication.

- In addition to that, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

***Applications***

- The infrared band, almost 400 THz, due to this wide bandwidth it can be used to transmit digital data with a very high data rate.

- There is an association called ***Infrared Data Association*** **(IrDA)**, sponsoring the use of infrared waves and has established **standards** for using these signals for communication between devices such as keyboards, mice, PCs, and printers.

- For example, manufacturers provide a special port called the **IrDA** port that allows a wireless keyboard to communicate with a PC.

- The **IrDA** port on the keyboard needs to point to the PC for transmission to occur.

- The standard originally defined a data rate of 75 kbps for a distance up to 8 m, the recent standard defines a data rate of 4 Mbps. Infrared signals defined by IrDA transmit through **line of sight**.

**Satellites:** `When used for communications, a satellite acts as a repeater. Its height above the Earth means that signals can be transmitted over distances that are very much greater than the line of sight.

*LECTURE NOTE: 11*

# CIRCUIT SWITCHING

## SWITCH:

- Switch in a network is a connecting device that filters and forwards packets between LAN segments. Switches operate at the data link layer (layer 2) and sometimes the network layer (layer 3) of the OSI Reference Model and therefore support any packet protocol.

- Switches connect computers, printers and servers within a building or campus, serves as a controller, enabling networked devices to talk to each other efficiently.

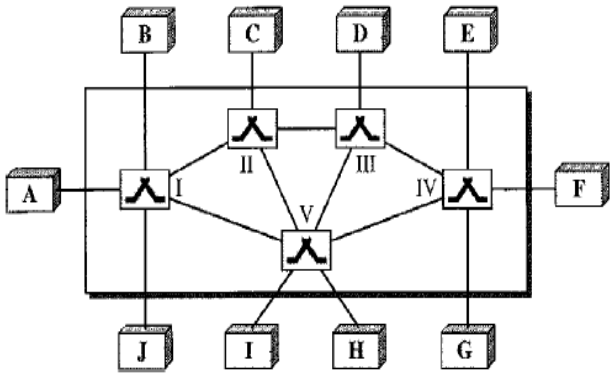
A network is a set of connected devices. Whenever we have multiple devices and have to connect it in one-to-one connection is the problem. One of the possible solutions is either follow the mesh topology or star topology where this connection is possible. Again for the large networks it is impractical to apply because large number of links, its length may leads too much infrastructure cost moreover, most of the time it may remain idle. If we go through multipoint connection of bus topology total number of devices and distance between devices may increase beyond capacity. So it's only one solution is to go through the switching concept.

## *SWITCHING:*

**Switching is defined as the process where connecting devices (Switches) interlinked are capable of creating temporary connections between two or more devices linked to the switch.** In switched network, some of these nodes are connected to the end systems computers or telephones others are used only for routing purpose.

It is the mechanism for moving information between different computer network and network segment in computer network.

For example: - whenever a telephone called is placed, there are numerous junctions in the communication path that perform this movement of data from one network onto another network.

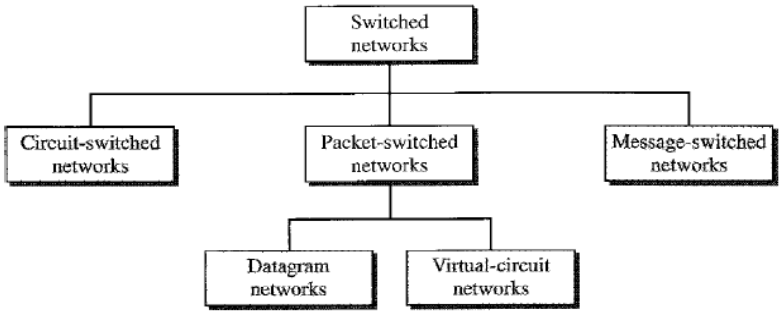



In the figure the end systems (communicating devices) are labeled A, B, C, D, and so on, and the switches are labeled I, II, III, IV, and V. Each switch is connected to multiple links.

Traditionally switching technology has been categorized into three parts:

- **circuit switching**
- **Packet switching:** The other common communications method is **packet switching**, which divides messages into packets and sends each packet individually. The **Internet** is based on a packet-switching protocol, **TCP/IP**.
- **Message switching:** In message switching, each switch stores the whole message and forwards it to the next switch. Although, we don't see message switching at lower layers, it is still used in some applications like electronic mail (e-mail).

***Its taxonomy is shown as***

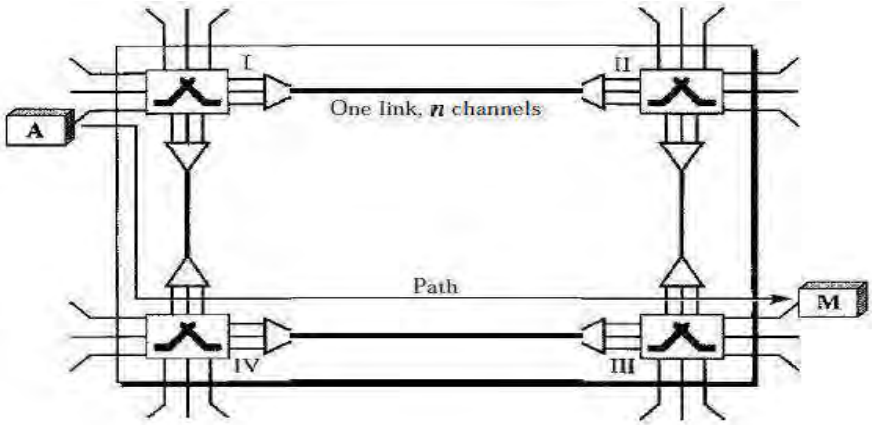



- **circuit switching**

A type of communications in which a **dedicated channel** (or *circuit*) is established for the duration of a transmission. The most ubiquitous circuit-switching network is the **telephone system**, which links together wire segments to create a single unbroken line for each telephone call. Circuit-switching

systems are ideal for communications that require data to be transmitted in **real-time**. Circuit-switching networks are sometimes called ***connection-oriented*** networks.

Circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Each link is normally divided into *n* channels by using FDM or TDM.

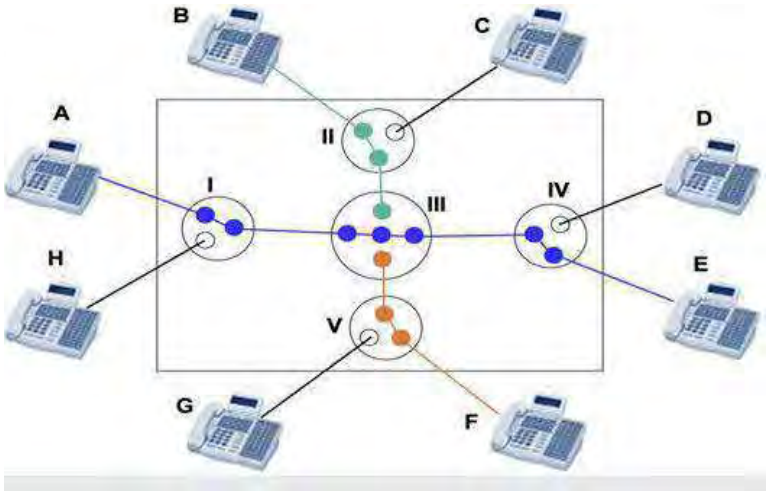



*Certain phases involved to make circuit switching connections and communication are:*

- ***Path or circuit establishment:*** the circuit must be established, when signals are transmitted from one network node to other node. This can send information through signals before that the receiving end and sending end must establish their circuit. **Note** that end-to-end addressing is required for creating a connection between the two end systems.

- ***Data transfer:*** after connection is established, data can be transmitted from one network node to other network node through network. The data may be analog signals or digital signals that depend on the nature of the network protocol.

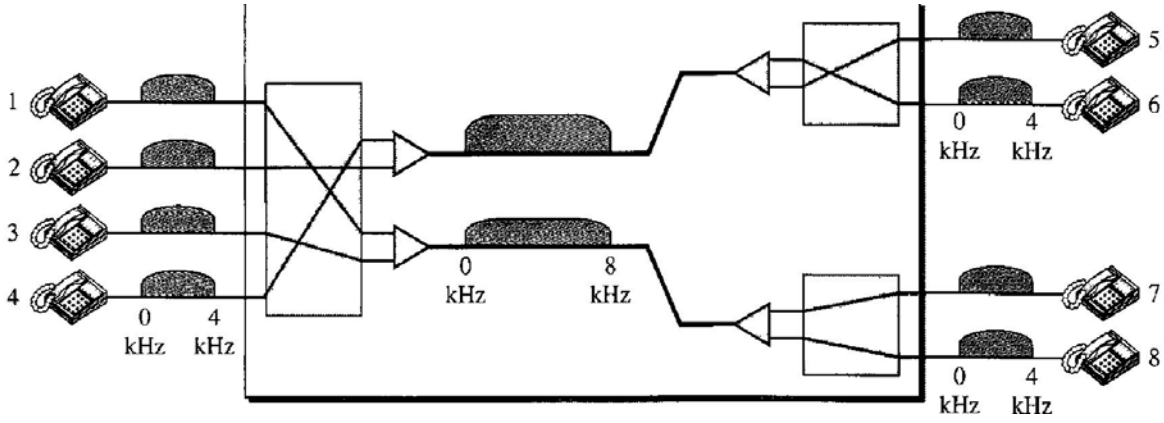
- ***Circuit disconnect or Teardown Phase:*** After some time of data transfer, the connection is terminated, generally by the action of one of the two network nodes.




The above figure shows that device A is connected to device E through the switches I, III & IV. Other devices can connect to each other's by moving the levers of the switches.

***For Example:***

Let us use a circuit-switched network to connect eight telephones in a small area. Communication is through 4-kHz voice channels. We assume that each link uses FDM to connect a maximum of two voice channels. The bandwidth of each link is then 8 kHz. This picture shows the situation. Telephone 1 is connected to telephone 7, 2 to 5, 3 to 8 and 4 to 6. Of course the situation may change when new connections are made. The switch controls the connections.



**Efficiency**

Circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to other connections. In a telephone network, people normally terminate the communication after finished the conversation. But in computer networks, a computer can be connected to another computer even if there is no activity for a long time. It means connected device remains idle.

**Delay**

Delay in this type of network is minimal. During data transfer the data are not delayed at each switch or small delay is negligible. Whatever total delay is due to the time needed to create the connection, transfer data, and disconnect the circuit.

**Structure of Circuit Switches**

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

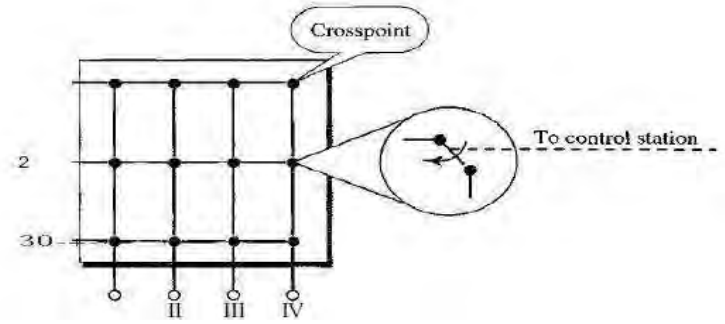
***Space-Division Switch***

- In this switch, the path from one device to another is spatially separate from other paths.
- This technology was originally designed for use in analog networks but is now a day's used in both analog and digital networks.
- In this technique paths in the circuit are separated from one another spatially.
- It uses Crossbar Switch that connects *n* inputs to *m* outputs in a grid, using electronic micro switches (transistors) at each cross point.

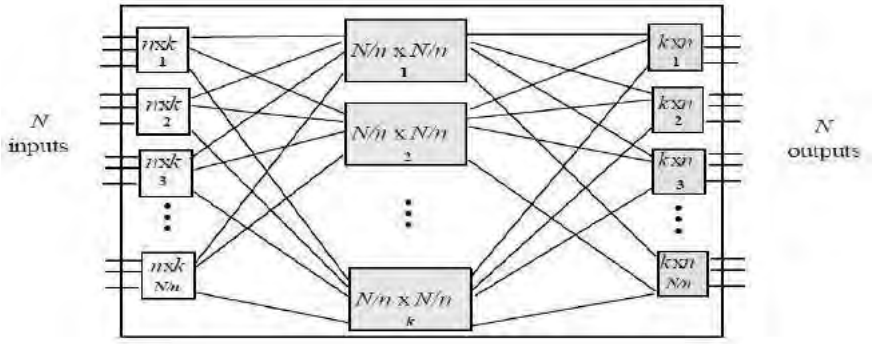
- The major limitation of this design is the number of cross points required.

- To connect *n* inputs to *m* outputs using a crossbar switch requires *n* x *m* cross points. Suppose, to connect 1000 inputs to 1000 outputs require a switch with 1,000,000 cross points. But a crossbar with this number of cross points is impractical and such a switch is also inefficient because statistics in practice that, fewer than 25 percent of the cross points are in use at any given time practically and rest are at idle condition.

Fig: ***Crossbar switch with three inputs and four outputs***



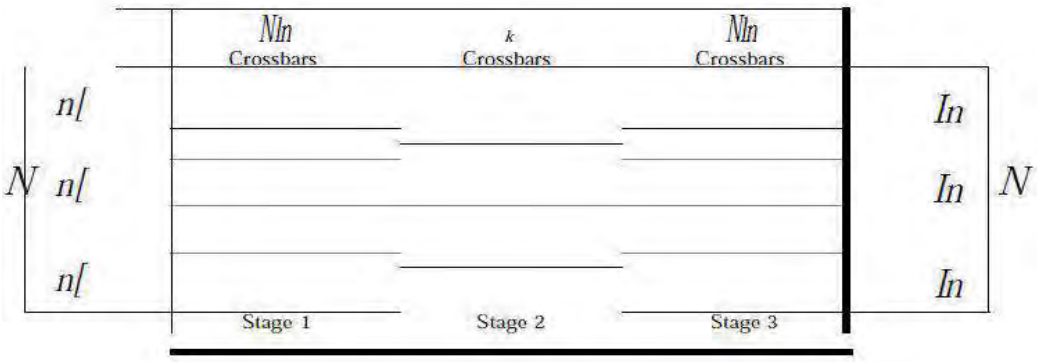

## MULTISTAGE SWITCH




The solution to the **limitations** of the crossbar switch is the **multistage switch**, which combines crossbar switches in several (normally three) stages. In a single crossbar switch, only one row or column (one path) is active for any connection. So for *N* x N cross points, we can allow multiple paths inside the switch(multipath switch), we can decrease the number of cross points. As cross points in the middle stage can be accessed by multiple cross points in the first or third stage.

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 but output merely busy.

In multistage switching, ***blocking*** issue occurred during heavy traffic. Where blocking refers to times when one input cannot be connected to an output because there is no path available between them—all the possible intermediate switches are occupied. One solution to blocking is to increase **the number of intermediate switches** based on the criteria.




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

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

Now we can calculate the total number of cross points as follows:

**N/n(nxk)+k(N/nxN/n)+N/n(kxn)**

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

---

In a three-stage switch, the total number of crosspoints is

$$2kN + k\left(\frac{N}{n}\right)^2$$

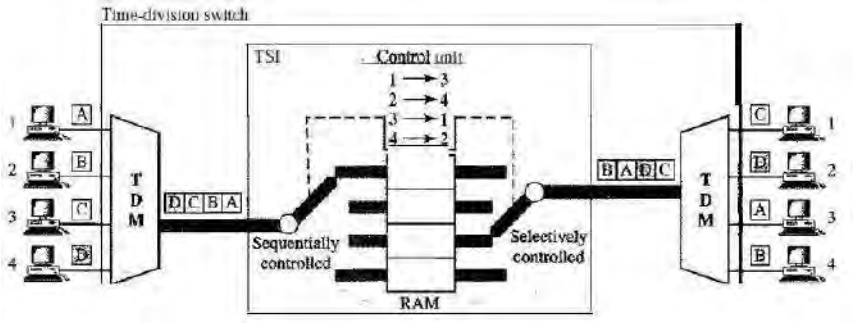
which is much smaller than the number of crosspoints in a single-stage switch ($N^2$).

---

So disadvantage space division technique is the number of cross points required to make space-division switching acceptable in terms of blocking. And its advantage is that, it is **instantaneous**.

***Time-Division Switch***

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




Time-Slot Interchange in the figure 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:

1 → 3 2 → 4 3 → 1 4 → 2

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

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.

### *Time- and Space-Division Switch Combinations*

When we combine space-division and time-division technologies we can optimized two results in switches are in 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)** switch. Other some possible are STP, SSTT, STST, TSTS, TSST etc.

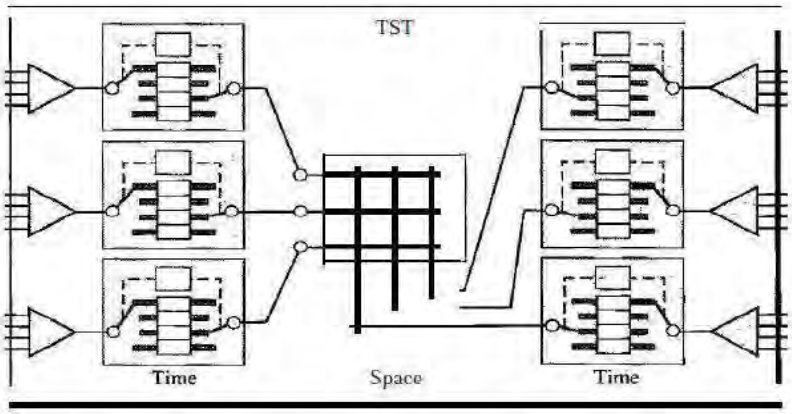



Figure shows a simple TST switch that consists of two time stages and one space stage and has 12 inputs and 12 outputs. Instead of one time-division switch, it divides the inputs into three groups (of four inputs each) and directs them to three timeslot interchanges. The result is that the average delay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

The last stage is a mirror image of the first stage. The middle stage is a space division switch (crossbar) that connects the TSI groups to allow connectivity between all possible input and output pairs.

## TELEPHONE NETWORK

Telephone networks were originally created to provide voice communication. Later on it started both digital as well as analog transmission. To transfer digital data it was needed **dial-up modem**. With the advancement of Internet came, it need for high-speed downloading and uploading for which this type of modem became just too slow. Then the telephone companies added a new technology, the ***digital subscriber line*** **(DSL).** Although dial-up modems still exist in many places all over the world, **DSL** provides much faster access to the Internet through the telephone network. Cable networks were originally created to provide access to TV programs for those subscribers who had no reception because of natural obstructions such as mountains.

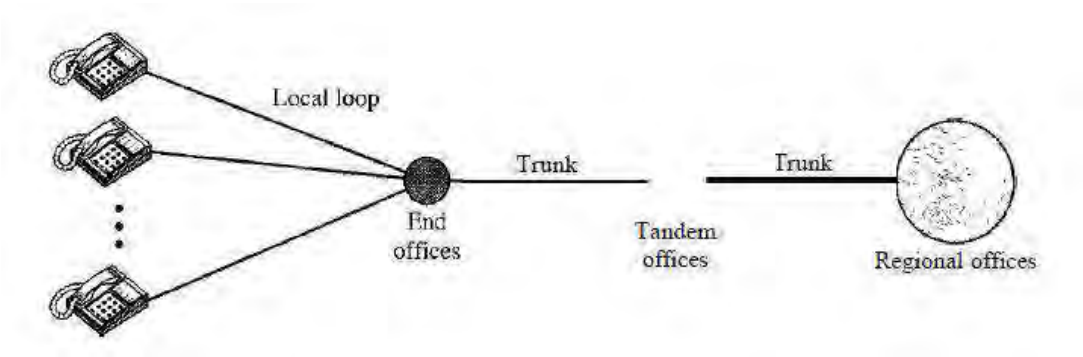
Later on it enabled access to remote broadcasting stations via microwave connections.

**Telephone networks** use circuit switching. The telephone network was started in the late 1800s. The entire network, which is referred to as the **plain old telephone system (POTS),** was originally an analog system using analog signals to transmit voice. Nearly1980s, began to carry data in addition to voice, and now digital as well as analog.

## MAJOR COMPONENTS OF TELEPHONE NETWORK

The telephone network made of three major components: **local loops, trunks**, and **switching** offices. The telephone network has several levels of switching offices such as **end offices, tandem offices,** and **regional offices**.

### *A TELEPHONE SYSTEM:*




### *LOCAL LOOPS*

- Local loop in a telephone network is a twisted-pair cable that connects the subscriber telephone to the nearest end office or local central office. Or a physical wiring cable that connect you to the telephone network.
- This line can be a voice line or data line. The local loop, used for voice, has a bandwidth of 4000 Hz (4 kHz). If we interestingly examine the telephone number associated with each local loop.
- The first three digits of a local telephone number define the office or area code or national destination code, and the next three digits are exchange and last four digits define the local loop number or subscriber number or phone number that represent small circuit bundles within same switch .

### TRUNKS

- It a communication path connecting two switching system in a network.
- Trunks are transmission media that handle the communication between offices.
- A trunk normally handles hundreds or thousands of connections through multiplexing.
- Transmission is usually through optical fibers or satellite links.

## SWITCHING OFFICES

- Switching office connects telephone caller.
- It can support many connections simultaneously.
- To avoid having a permanent physical link between any two subscribers or user, the telephone company has switches located in a switching office. It means several users or subscriber directed not connected each other instead of that switches are used to link all.

- Switches are located in switching office to manage every connection.
- A switch connects several local loops or trunks and allows a connection between different subscribers.
- It has several levels such as **end offices**: it handle service such as call forwarding and call waiting, **tandem offices:** it connect offices in the same network or between network but always deal with trunk rather than customer line, **regional offices**: it connects office regionally.

## LATAs

LATA (**local access and transport area**) is a term or an expression used in the U.S communication law. It is for a geographic and administrative area responsible for local exchange carriers (LECs) covered by one or more local telephone companies.

**Intra-LATA Services:** calls within a LATA or a connection between two local exchanges within the LATA are called **Intra LATA.**

**Inter-LATA Services:** calls between LATAs or a connection between a carrier in one LATA to a carrier in another LATA are called **Inter LATA.** Calls that cross LATA boundaries are handed off to an **IXC** (Inter-exchange Carrier)

## POINTS OF PRESENCE

### Services Provided by Telephone Networks

- ***Analog Services:*** telephone companies provided their subscribers with analog services.
- Analog Switched Services
- The wide-area telephone service (WATS)
- Analog Leased Service: an analog leased service offers customers the opportunity to lease a line, sometimes called a *dedicated line* that is permanently connected to another customer.
- *Digital Services*

## DIAL-UP MODEM

- The term modem is a composite word that refers to the two functional entities a signal modulator and a signal demodulator.
- A modulator creates a band pass analog signal from binary data.
- A demodulator recovers the binary data from the modulated signal.

Traditional telephone lines can carry frequencies between 300 and 3300 Hz, giving a bandwidth of 3000 Hz. These range used for voice transmission, where it deals with interference and distortion without any loss of integrity. But edge of this band is not used for data communication. The effective bandwidth of a telephone line being used for data transmission is 2400 Hz, covering the range from 600 to 3000 Hz. **Note** that today some telephone lines are capable of handling greater bandwidth than traditional lines.

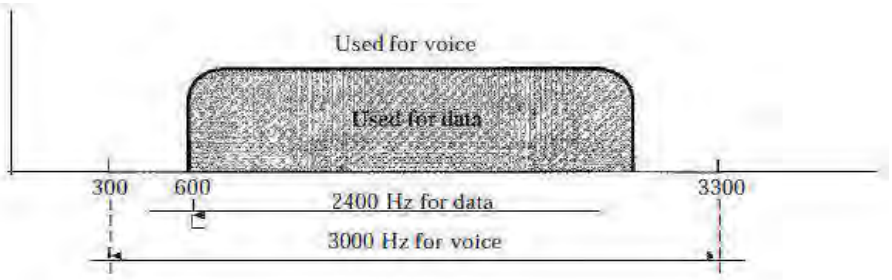
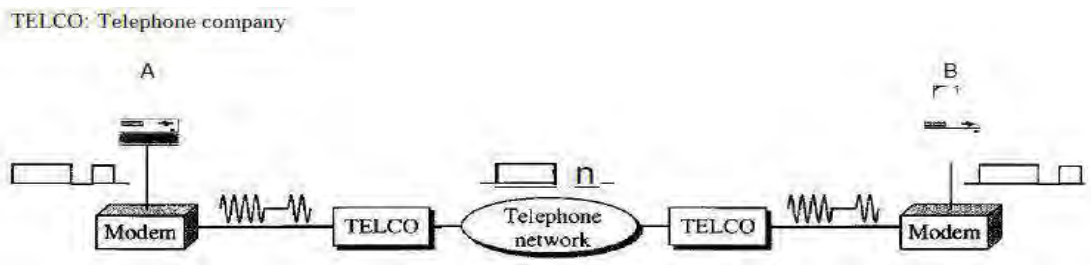



Figure shows the relationship of modems to a communications link. The computer on the left sends a digital signal to the modulator portion of the modem. The data are modulated and sent an analog signal on the telephone lines. The modem on the right receives the analog signal, demodulates it through its demodulator, and delivers data to the computer on the right in digital form. The communication can be bidirectional same as modulation/demodulation processes.

*Modem* stands for modulator/demodulator.

*Modulation/demodulation*



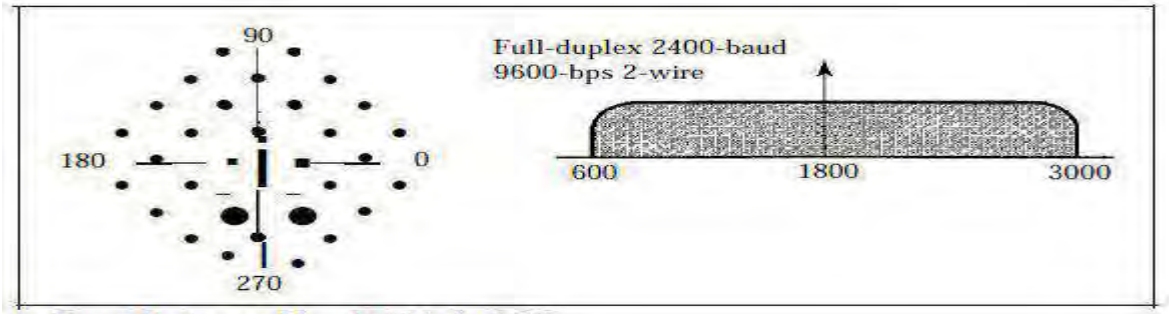
Modem Standards Various V-series based modems standards are published by the ITU-T such as:

***V.32 and V.32bis***

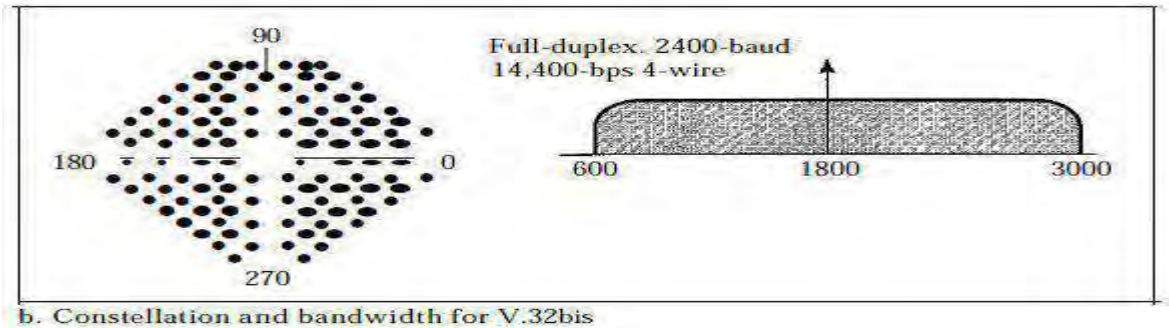
The V.32 modem uses a combined modulation and encoding technique called trellis coded modulation. Trellis is a QAM plus a redundant bit. The data stream is divided into 4-bit sections. Instead of a quad bit (4-bit pattern) a *penta bit* (5-bit pattern) is transmitted. The value of the extra bit is calculated from the values of the data bits. The extra bit is used for error detection. The V.32bis modem was the first of the ITU-T standards to support 14,400-bps transmission. The V.32bis uses 128-QAM transmission (7 bits/baud with 1 bit for error control) at a rate of 2400 baud (2400 x 6 = 14,400 bps). Y.32bis provide the inclusion of an automatic **fall-back** and **fall-forward** feature that enables the modem to adjust its speed upward or downward that depending on the quality of the line or signal.

***V.34bis***

The V.34bis modem provides a bit rate of 28,800 with a 960-point constellation and a bit rate of 33,600 bps with a 1664-point constellation.




a. Constellation and bandwidth for V.32




b. Constellation and bandwidth for V.32bis

***V.90***

V.90 modems used with a bit rate of 56,000 bps these are called 56K modems. These modems may be used only if one party is using digital signaling (such as through an Internet provider). They are asymmetric in that the downloading rate (flow of data from the Internet service provider to the PC) is a maximum of 56 kbps, while the uploading rate (flow of data from the PC to the Internet provider) can be a maximum of 33.6 kbps.

**V.92**

It is advance version of V.92. These modems can adjust their speed, and if the noise allows, they can upload data at the rate of 48 kbps. The downloading rate is still 56 kbps. It has one additional feature that the modem can interrupt the Internet connection when there is an incoming call, if the line has call-waiting service.

**TELEPHONE MODEMS**

- **DSL(digital subscriber line)**

(i) To provide higher-speed access to the Internet telephone companies developed another technology, DSL.

(ii) It supports high-speed digital communication over the existing local loops.

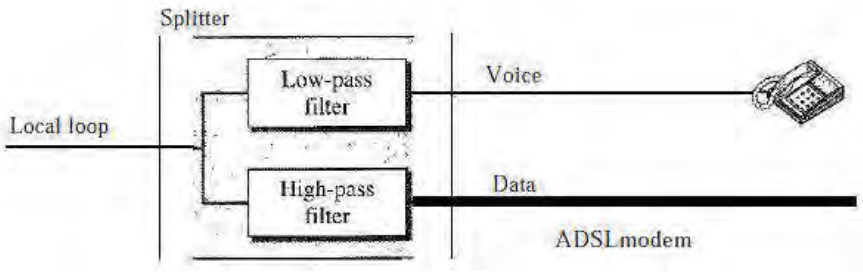
(iii) DSL technology is a set of technologies, each differing in the first letter

(ADSL, VDSL, HDSL, and SDSL).

(iv)The set is often referred to as xDSL, where *x* can be replaced by A, V, H, or S

- **ADSL** (**asymmetric digital subscriber line** )

- It is the first technology in DSL.
- Like a 56K modem, provides higher speed (bit rate) in the downstream direction (from the Internet to the resident) than in the upstream direction (from the resident to the Internet).
- That is the reason it is called asymmetric.




ADSL is an asymmetric communication technology designed for residential users; it is not suitable for businesses.

- **HDSL (high bit rate digital subscriber line**)

- Originally it was designed as an alternative to the T-1 line (1.544 Mbps).
- The T-1line uses alternate mark inversion (AMI) encoding, which is very susceptible to attenuation at high frequencies.
- This property limits the length of a T-l line to 3200 ft (1 km).
- For longer distances communication, a repeater is necessary, which means increased costs.
- Now, HDSL uses 2B1Q encoding which is less susceptible to attenuation.
- A data rate of 1.544 Mbps (sometimes up to 2 Mbps) can be achieved in transmission even without repeaters up to a distance of 12,000 ft (3.86 km).
- HDSL uses two twisted pairs (one pair for each direction) to achieve full-duplex transmission.

- **SDSL (symmetric digital subscriber line)**

- it is a one twisted-pair version of HDSL.
- It provides full-duplex symmetric communication supporting up to 768 kbps in each direction.
- SDSL provides symmetric communication that can be considered an alternative to ADSL.

- **VDSL (very high-bit-rate digital subscriber line)**

  - The very high-bit-rate digital subscriber line (VDSL)
  - It is an alternative approach to ADSL, uses coaxial, fiber-optic, or twisted-pair cable for short distances.
  - The modulating technique is DMT.

- It provides a range of bit rates (25 to 55 Mbps) for upstream communication at distances of 3000 to 10,000 ft.
- The downstream rate is normally 3.2 Mbps.

Summary of DSL technology

| *Technology* | *Downstream Rate* | *Upstream Rate* | *Distance (jt)* | *Twisted Pairs* | *Line Code* |
|---|---|---|---|---|---|
| ADSL | 1.5-6.1 Mbps | 16-640 kbps | 12,000 | 1 | DMT |
| ADSL Lite | 1.5 Mbps | 500 kbps | 18,000 | 1 | DMT |
| HDSL | 1.5-2.0 Mbps | 1.5-2.0 Mbps | 12,000 | 2 | 2B1Q |
| SDSL | 768 kbps | 768 kbps | 12,000 | 1 | 2B1Q |
| VDSL | 25-55 Mbps | 3.2 Mbps | 3000-10,000 | 1 | DMT |

*MODULE-II*

*LECTURE NOTE: 12*

# DATA LINK LAYER

The data link layer uses the services of the physical layer to send and receive bits over communication channels. It has a number of functions, including:

1. Providing a well-defined service interface to the network layer.
2. Dealing with transmission errors.
3. Regulating the flow of data so that slow receivers are not swamped by fast senders.
4. Dealing with the physical addressing and channel access mechanism.

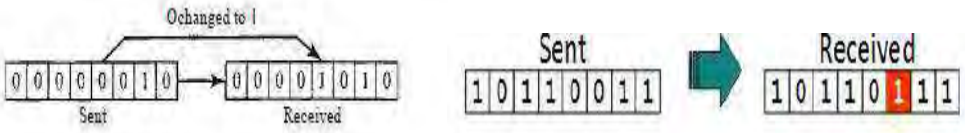
To accomplish these goals, the data link layer takes the packets that gets from the network layer and encapsulates them into frames for transmission. Each frame contains a frame header, a payload field for holding the packet, and a frame trailer

## TYPES OF ERRORS

Data in the form of bits transfer along the medium may not accurate due to many unpredictable errors or interference may occur and these types are discussing details below.

## SINGLE-BIT ERROR:

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




- ✓ Single-bit errors are the least likely type of error in serial data transmission
- ✓ For a single-bit error to occur the noise must have a duration of only 1 micro second which is very rare; noise normally lasts much longer than this.

**MULTIPLE BITS ERROR**


Frame is received with more than one bit in different positions in corrupted state.

**BURST ERROR***:*

- ✓ The term *burst error* means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.
- ✓ 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.
- ✓ **Note** that a burst error does not necessarily mean that the errors occur in consecutive bits.
- ✓ A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits.

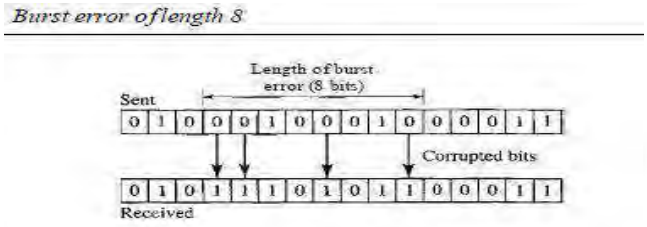



fig:a

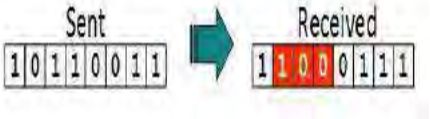



fig:b

Here in the figure three bits has been chenged from 0 bits to 1s and 1s to 0s. Here legnth of burst is 8 in figure a and 3 in figure b.

- ✓ Error control mechanism may involve two possible ways
  Error detection
  Error correction

**REDUNDANCY**

It is some extra bits or redundancy bits sends with our data to detect or correct errors. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.

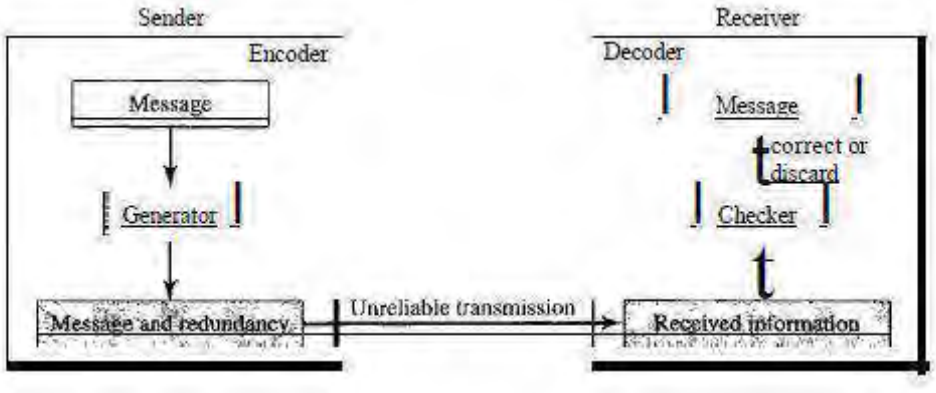
## ERROR DETECTION

Errors in the received frames are detected by means of **Parity Check**, **Cyclic Redundancy Check (CRC), checksum** etc. In all cases, few extra bits or redundancy bits are sent along with actual data to confirm that bits received at other end are same as they were sent. If the counter-check at receiver' end fails, the bits are considered corrupted.

If the following two conditions are met, the receiver can detect a change in the original code word.

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

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




**Example**

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

A code for error detection

| Data words | Code words |
|---|---|
| 00 | 000 |
| 01 | 011 |
| 10 | 101 |
| 11 | 111 |

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

1. The receiver receives 011. It is a valid codeword. The receiver extracts the data word 01 from it.
2. The codeword is corrupted during transmission, and 111 is received (the leftmost bit is corrupted). This is not a valid codeword and is discarded.
3. The codeword is corrupted during transmission, and 000 is received (the right two bits are corrupted). This is a valid codeword. The receiver incorrectly extracts the data word 00. Two corrupted bits have made the error undetectable.

| An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected. |
|---|

## PARITY CHECK

The most common and least expensive mechanism for error detection is the parity checking. Parity checking can be **simple parity check** or **multi dimensional parity check**.

### Simple parity check

- ✓ It is simple mechanism.
- ✓ In this mechanism redundant bit is called parity bit.
- ✓ Simple parity bit checking can be either even or odd parity.
- ✓ One extra bit or parity bit is sent along with the original data unit to make number of 1s either even in case of even parity, or odd in case of odd parity.

The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value 0 is added. This way number of 1s remains even. If the number of 1s is odd, to make it even a bit with value 1 is added. One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.

The receiver simply counts the number of 1s in a frame on received. If the count of 1s is even and even parity is used, the frame is considered to be not-corrupted and is accepted and if number of 1s odd then discarded. Similarly for odd parity, If the count of 1s is odd and odd parity is used, the frame is still not corrupted otherwise discarded.

**Performance**

- ✓ Simple parity check can detect all single bit error.
- ✓ It can detect error as long as the total number of bits changed is odd (1, 3, 5…).
- ✓ But when more than one bit are erroneous and total number of bit changed is even, then it is very hard for the receiver to detect the error.

## TWO DIMENSIONAL PARITY CHECKS

Two dimensional parity checks is better approach than simple parity check. In this method, a block of bits is organized in a table (rows and columns). First we calculate the parity bit for each data unit, and then organized into table as in rows and columns. For example original data of 4 bytes and lets its data units are of 7 bits each. For even parity check, first row wise 1 is added at right side, similarly column wise bits are added.

five 7-bit character packet, even parity

| | |
|---|---|
| 0110100 | 1 |
| 1011010 | 0 |
| 0010110 | 1 |
| 1110101 | 1 |
| 1001011 | 0 |
| 1000110 | 1 |

Now data and parity bits represented as:

**01101001 10110100 00101101 11101011 10010110 10001101(parity bit)**

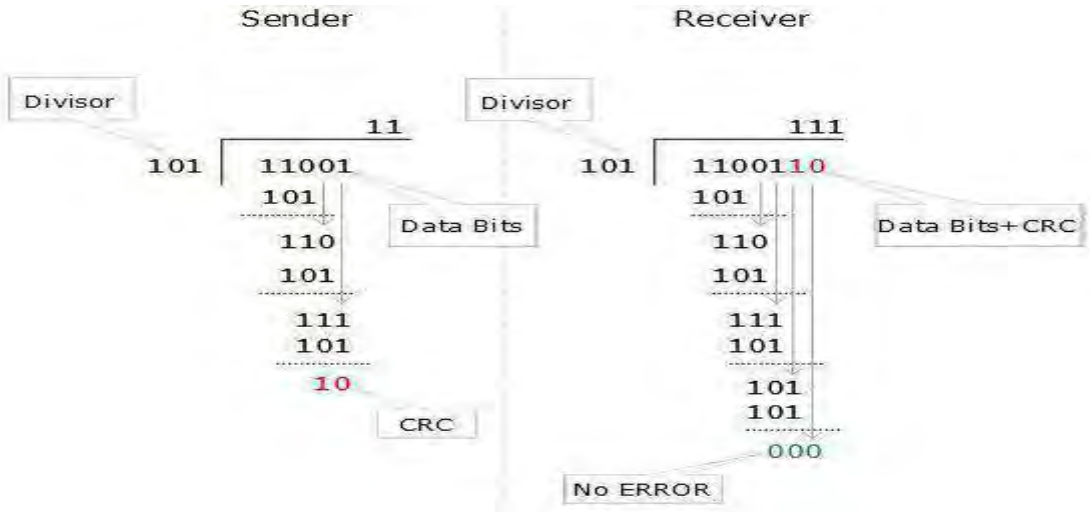
We then attach the 8 parity bits to the original data and send them to the reciever.

**performance**

- ✓ This mechanism increase the likelyhood of detecting burst error.
- ✓ Redundance of n bits can easily detect a burst error of n bits.
- ✓ A burst error of more than n bits is also detected by this method with high probability but one exclusive is if two data units damaged exactly in same position then checker will fail to detect.

## CYCLIC REDUNDANCY CHECK (CRC)

CRC is a different approach and most powerful mechanism of redundancy check to detect if the received frame contains invalid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits this remainder bits is called CRC or CRC remainder. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as code words.




At the other end, the receiver performs division operation on code words using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

The redundancy bits used by CRC are derived by dividing the data unit by a predetermined divisor i.e the remainder is CRC. To validate CRC, must have two qualities: it must have exactly one less bit than the divisor and appending it to the

end of the original data string must make the resulting bit sequence exactly divisible by the divisor.

Here CRC generator uses modulo-2 division to generate data similarly the CRC checker functions exactly as generator does. After receiving it does modulo-2 division, if the remainders are all zeros the CRC is dropped and data are accepted otherwise discarded and data are resent.

## POLYNOMIALS

The divisor in the CRC generator is most often represented in string of 1s and 0s but as an algebraic polynomial. It is useful for two reasons that are, it is short and it can be used to prove the concept mathematically (which is beyond the scope of the book). The relationship between the polynomials to its corresponding binary representation is:

$$x^7+x^6+x^4+x^2+1 = 11010101$$

A polynomial should be selected based on the following properties:

- ✓ It should not divisible by x.
- ✓ it should be divisible by x+1

### Performance

- ✓ CRC is very effective error detection mechanism.
- ✓ It can detect all burst error that effect an odd number of bits.
- ✓ It can detect all burst error of length less than or equal to the degree of the polynomial.
- ✓ It can detect all burst error with high probability even burst error of length greater than the degree of polynomial.

## DETECTION VERSUS CORRECTION

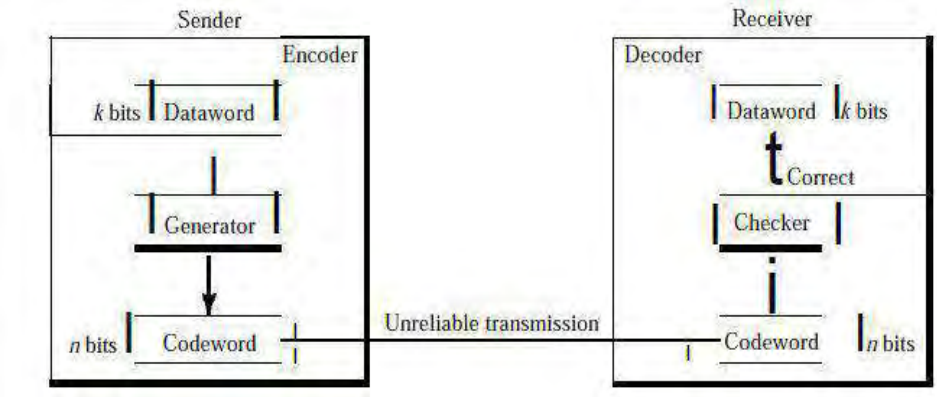
- ✓ The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a burst error.
- ✓ In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location i.e error location in the message. The number of the errors and the size of the message are important factors.

**Forward Error Correction versus Retransmission (backward Error correction)**

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

**ERROR CORRECTION**

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




Above figure shows the role of block coding in error correction. The idea is same as error detection but the checker functions are much more complex here.

For m data bits, r redundant bits are used, r bits can provide 2r combinations of information. In m+r bit codeword, there is possibility that the r bits themselves may get corrupted. So the number of r bits used must inform about m+r bit locations plus no-error information, i.e. m+r+1.

$$2^r >= m+r+1$$

**Example**

Assume the data word is 01. The sender consults the table (or uses an algorithm) to create the Code word 01011. The codeword is corrupted during transmission, and 01001 is received (error in the second bit from the right). First, the receiver finds that the received codeword is not in the table. This means an error has occurred. (Detection must come before correction.) The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct data word.

A code for error correction

| Data words | Code words |
|---|---|
| 00 | 00000 |
| 01 | 01011 |
| 10 | 10101 |
| 11 | 11110 |

1. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.
2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.
3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the data word 01

*LECTURE NOTE: 13*

# DATA LINK CONTROL

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

**Data link control** functions include framing, flow and error control, and software implemented protocols that provide smooth and reliable transmission of frames between nodes. To implement data link control, we need protocols. Each protocol is a set of rules that need to be implemented in software and run by the two nodes involved in data exchange at the data link layer. Five protocols are there to implement data link control**: two for noiseless (ideal)** channels and three for **noisy (real)** channels. Those in the first category are **not actually** implemented, but provide a foundation for understanding the protocols in the second category.The responsibility of data link layer is point to point flow and error control.

**FRAMING**

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

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

Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt. Although the whole message could be packed in one frame, that is not normally done. Because a frame can be very large, making flow and error control very

inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

**FIXED-SIZE FRAMING**

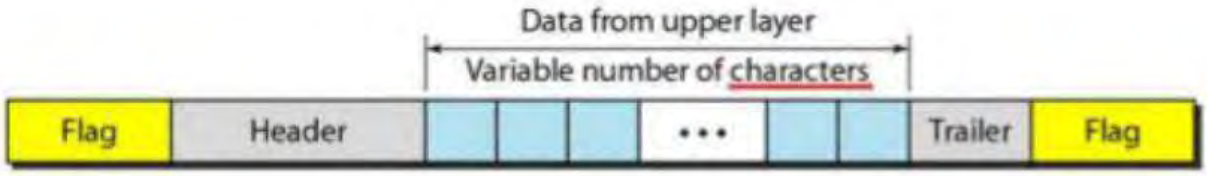
In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.

**VARIABLE-SIZE FRAMING**

In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose: **a character-oriented approach** and **a bit-oriented approach**.

***CHARACTER-ORIENTED PROTOCOLS:*** In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII. The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.

Figure: *A frame in a character-oriented protocol*



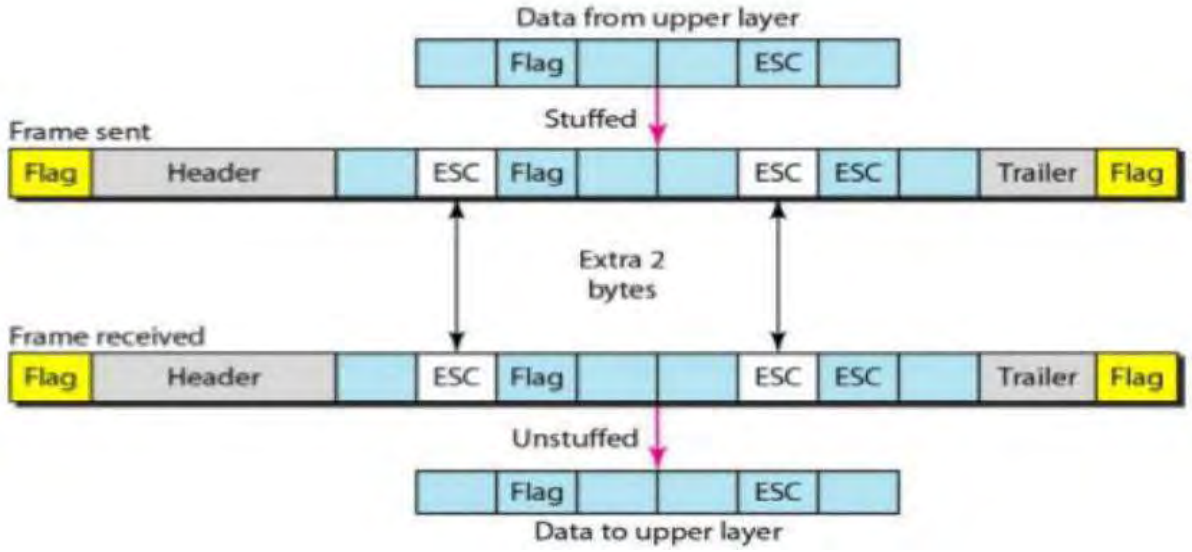

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte. This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag. Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it **creates**

**another problem**. if the text contains one or more escape characters followed by a flag, the receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame.

To solve this problem, if the escape character is part of the text, an extra one is added to show that the second one is part of the text. Character-oriented protocols present another problem in data communications. The universal coding systems in use today, such as Unicode, have 16-bit and 32-bit characters that conflict with 8-bit characters. We can say that in general, the tendency is moving toward the bit-oriented protocols.

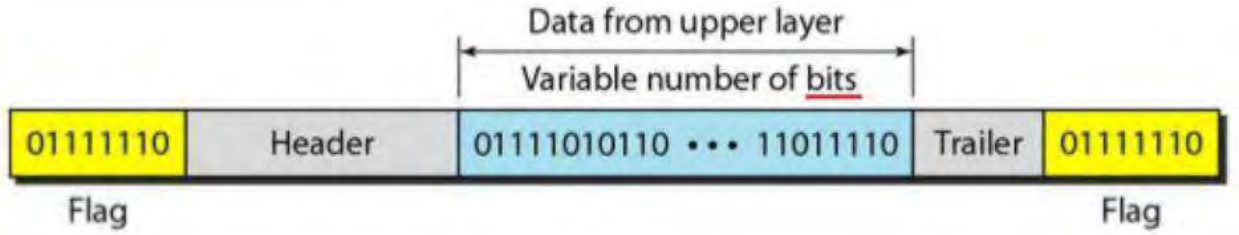
Figure: *Byte stuffing and un stuffing*




***BIT-ORIENTED PROTOCOLS****:* In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on. However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame. This flag can create the same type of problem as in the byte-oriented protocols.

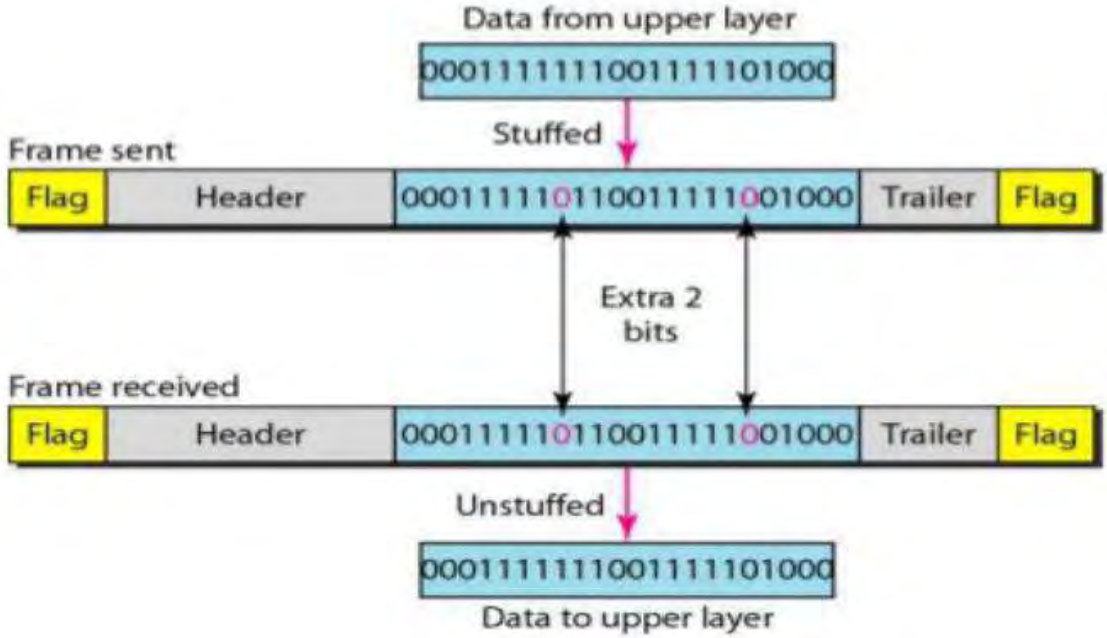
That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called **bit stuffing**.

Figure: *A frame in a bit-oriented protocol*




In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver. **Note** that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.

Figure: *Bit stuffing and un stuffing*




The above figure shows bit stuffing at the sender and bit removal at the receiver. **Note** that even if we have a 0 after five 1s, we still stuff a 0. The 0 will be removed by the receiver. This means that if the flag like pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver.

Bit stuffing is the process of adding one extra 0 whenever five consecutive 18 follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

## FLOW AND ERROR CONTROL

### FLOW CONTROL

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

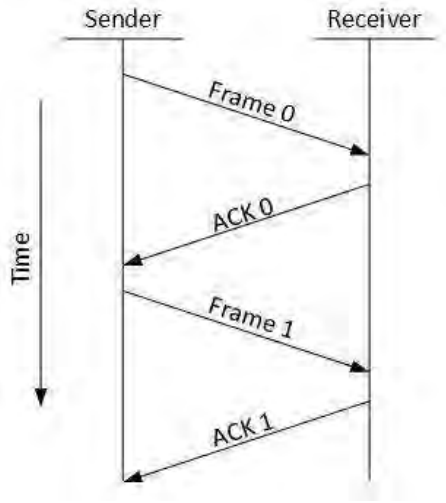
Flow control coordinates the amount of data that can be sent before receiving an acknowledgment. In most protocols, **flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver**. The flow of data must not be allowed 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 if it can process the incoming data slowly.

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 that the sender can send before waiting for acknowledgment.

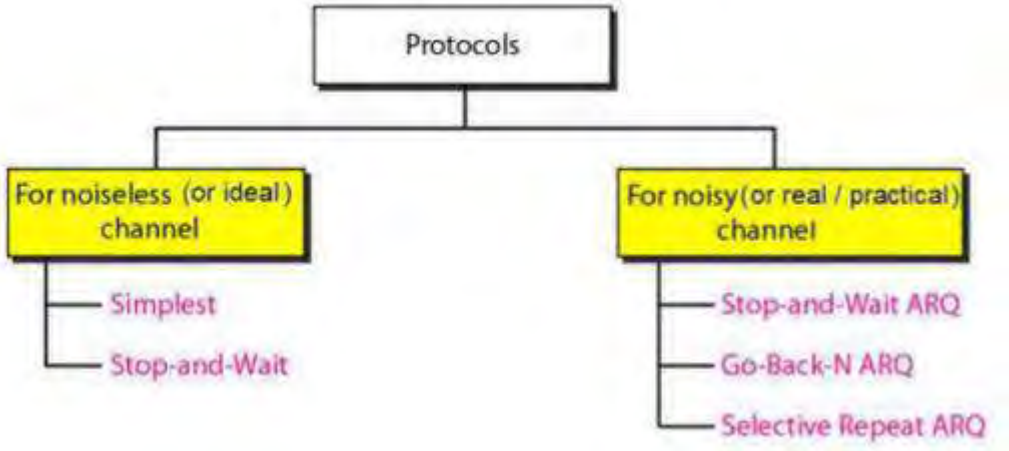
Two types of mechanisms can be deployed to control the flow and error:

**Stop and wait:** the data frames travel from one node, called the sender, to another node, called the receiver. The special frames, called acknowledgment (ACK) and negative acknowledgment (NAK) can flow in the opposite direction for flow and error control purposes, data flow in only one direction. In this mechanism, sender has to wait for acknowledgment after sending frame. Its name itself implies the meaning of the mechanism (i.e stop sending and wait for some time).




## SLIDING WINDOW

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. The stop and wait flow control mechanism wastes resources, but this protocol tries to make use of underlying resources as much as possible by using sliding window.



Protocols are divided into those that can be used for noiseless (error-free) channels and those that can be used for noisy (error-creating) channels. The protocols in the first category cannot be used in real life, but they serve as a basis for understanding the protocols of noisy channels.

**ERROR CONTROL**

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

Requirements for error control mechanism:

- **Error detection** - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
- **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.
- **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame or out of ordered frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
- **Retransmission:** The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or its acknowledgement is lost in transit.

In noisy channel, there are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

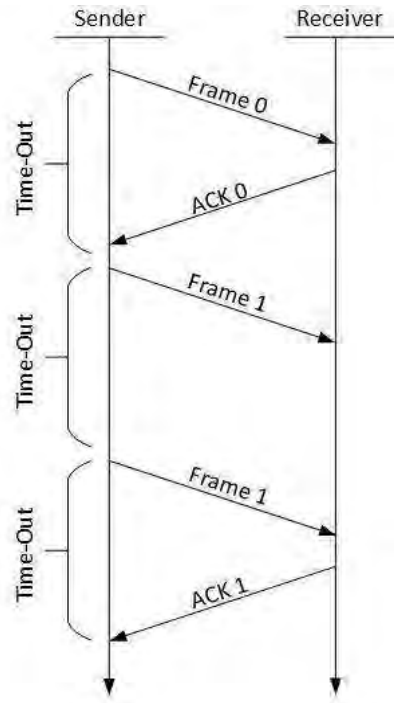
**Stop-and-Wait Automatic Repeat Request**

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.

The following transition may occur in Stop-and-Wait ARQ:

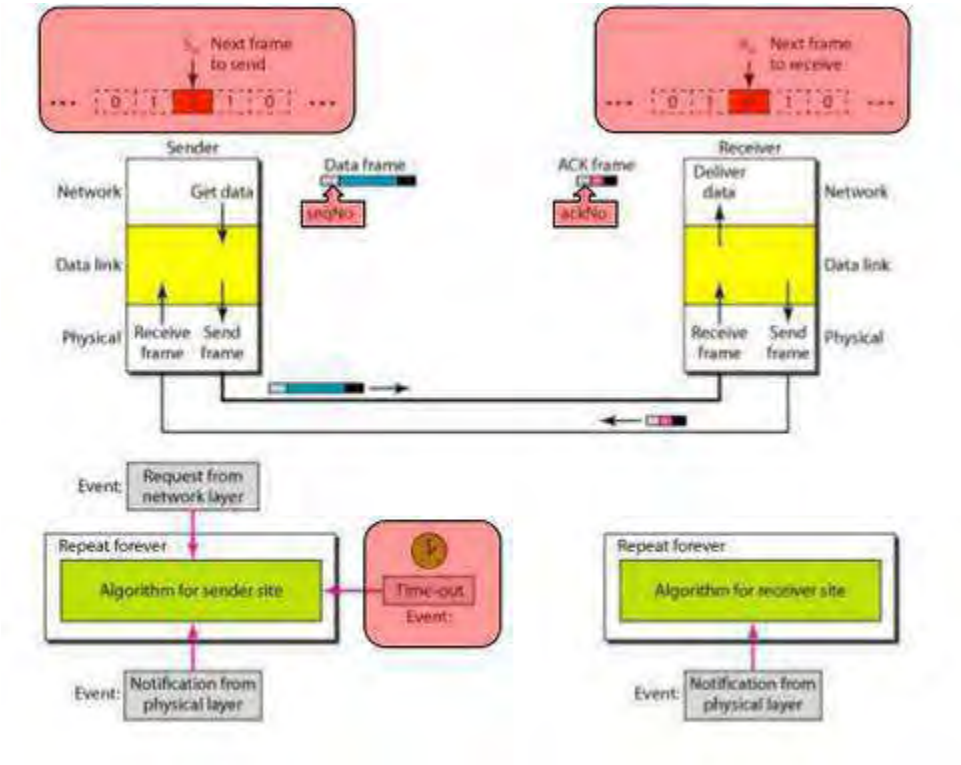
- The sender maintains a timeout counter.

- To detect and correct corrupted frames, **redundancy bits** are added to our data frame.
- To identify a received frame is correct one, or a duplicate, or a frame out of order numbered the frames sequentially.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.




When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver. Lost frames are more difficult to handle than corrupted ones. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.

Design of stop and wait ARQ Protocol




The completed and lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, the sender keeps a copy of the sent frame to resent it. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK even though several copies of the same frame can be in the network. Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number. The ACK frame for this protocol has a sequence number field. In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.

***Design***

In the design of the Stop-and-Wait ARQ Protocol, the sending device keeps a copy of the last frame transmitted until it receives an acknowledgment for that frame. A data frames uses a seqNo (sequence number) similarly an ACK frame uses an ackNo (acknowledgment number). The sender has a control variable *Sn* (sender, next frame to send), that holds the sequence number for the next frame to be sent (0 or 1) and the receiver has a control variable *Rn* (receiver, next frame expected), that holds the number of the next frame expected. When a frame is sent, the value

of *Sn* is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. When a frame is received, the value of *Rn* is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. Variable *Sn* points to the slot that matches the sequence number of the frame that has been sent, but not acknowledged, *Rn* points to the slot that matches the sequence number of the expected frame.

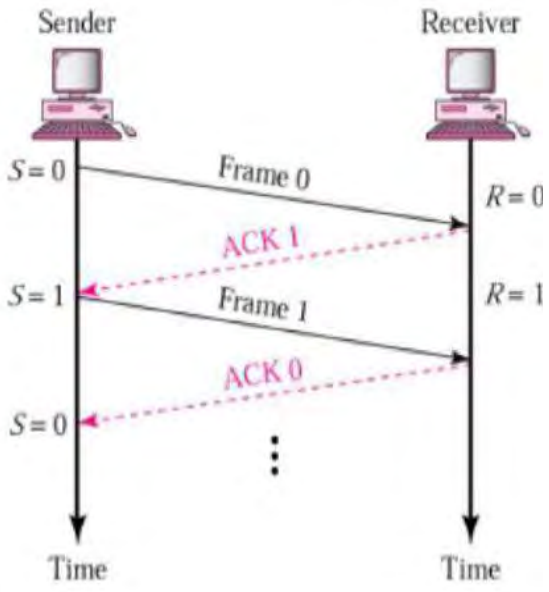
***Analysis***

The processes in the first event (Send Frame, Store Frame, and Purge Frame) use an *Sn* defining the frame sent out. This buffer is to hold this frame until we are sure that it is received safe and sound. If the frame is not corrupted and the ackNo of the ACK frame matches the sequence number of the next frame to send, we stop the timer and purge the copy of the data frame we saved. Otherwise, we just ignore this event and wait for the next event to happen. After each frame is sent, a timer is started. When the timer expires, the frame is resent and the timer is restarted.

On the receiving side all arrived data frames that are corrupted are ignored. If the seqNo of the frame is the one that is expected *(Rn )*, the frame is accepted, the data are delivered to the network layer, and the value of *Rn* is incremented. However, there is one subtle point here. Even if the sequence number of the data frame does not match the next frame expected, an ACK is sent to the sender. This ACK, however, just reconfirms the previous ACK instead of confirming the frame received. This is done because the receiver assumes that the previous ACK might have been lost, the resent ACK may solve the problem before the time-out does it.

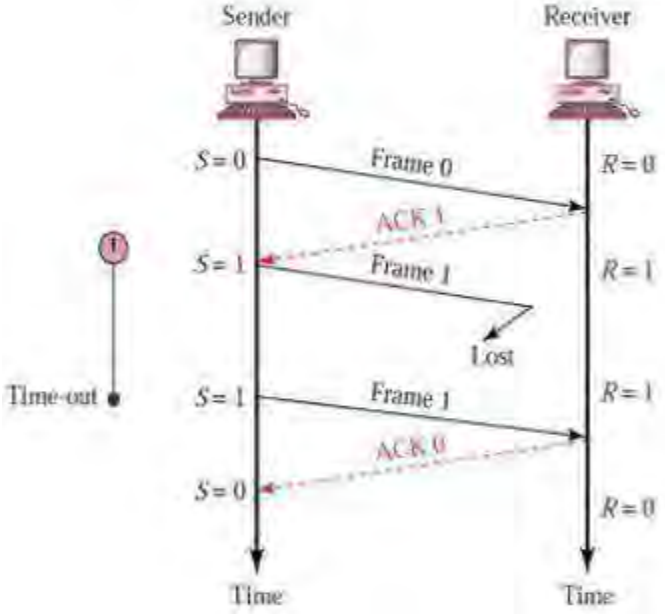
***Efficiency***

The Stop-and-Wait ARQ is very inefficient, if the channel is *thick* and *long.* By *thick*, means that channel has a large bandwidth, *long*, means the round-trip delay is long. The product of these two is called the band width delay product. The bandwidth-delay product is a measure of the
number of bits we can send out of our system while waiting for news from the receiver.



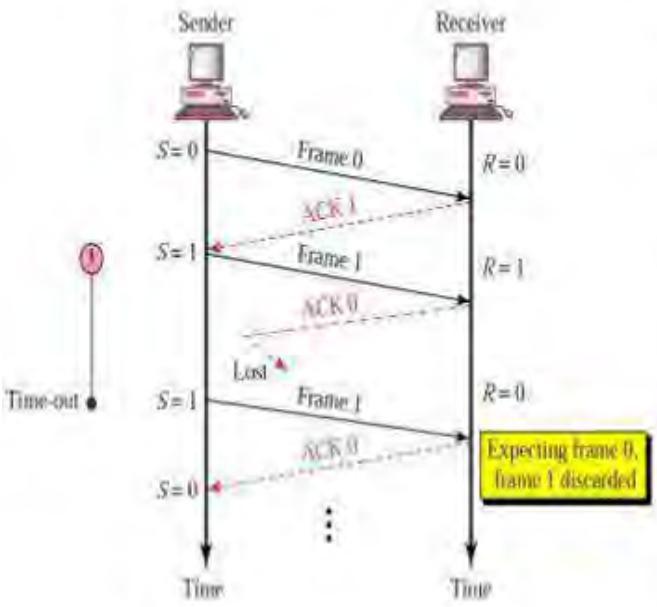

- Sender keeps a copy of the last frame until it receives an acknowledgement.
- For identification, both data frames and acknowledgements (ACK) frames are numbered alternatively 0 and 1.
- Sender has a control variable (S) that holds the number of the recently sent frame. (0 or 1)
- Receiver has a control variable ® that holds the number of the next frame expected (0 or 1).
- Sender starts a timer when it sends a frame. If an ACK is not received within a allocated time period, the sender assumes that the frame was lost or damaged and resends it
- Receiver send only positive ACK if the frame is intact.
- ACK number always defines the number of the next expected frame

# Stop-and-Wait ARQ, lost ACK frame



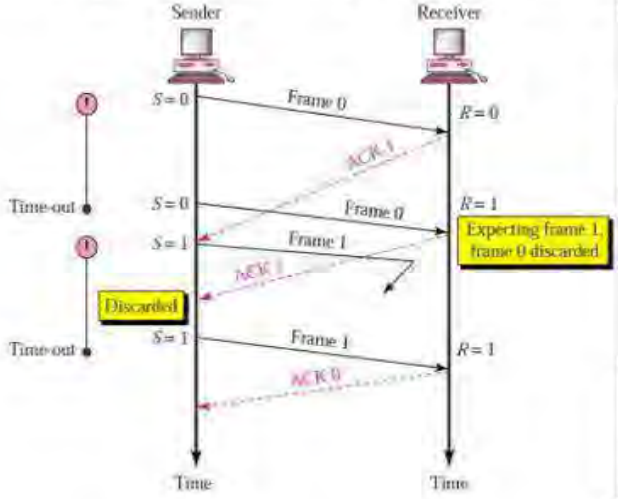

- When a receiver receives a damaged frame, it discards it and keeps its value of R.
- After the timer at the sender expires, another copy of frame 1 is sent.

# Stop-and-Wait, lost ACK frame

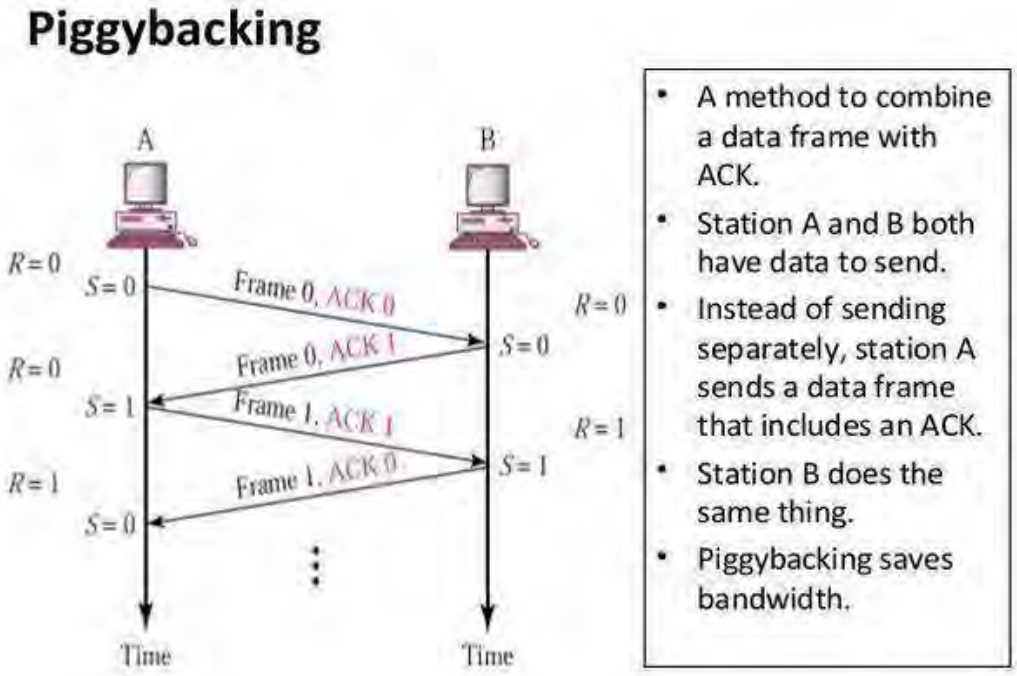



- If the sender receives a damaged ACK, it discards it.
- When the timer of the sender expires, the sender retransmits frame 1.
- Receiver has already received frame 1 and expecting to receive frame 0 (R=0). Therefore it discards the second copy of frame 1.

# Stop-and-Wait, delayed ACK frame




- The ACK can be delayed at the receiver or due to some problem
- It is received after the timer for frame 0 has expired.
- Sender retransmitted a copy of frame 0. However, R =1 means receiver expects to see frame 1. Receiver discards the duplicate frame 0.
- Sender receives 2 ACKs, it discards the second ACK.




**GO-BACK-N AUTOMATIC REPEAT REQUEST**

To improve the efficiency of transmission, multiple frames must be in transition while waiting for acknowledgment. In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment.

The first is called Go-Back-N Automatic Repeat Request. In this protocol we can send several frames before receiving acknowledgments. We keep a copy of these frames until the acknowledgments arrive.

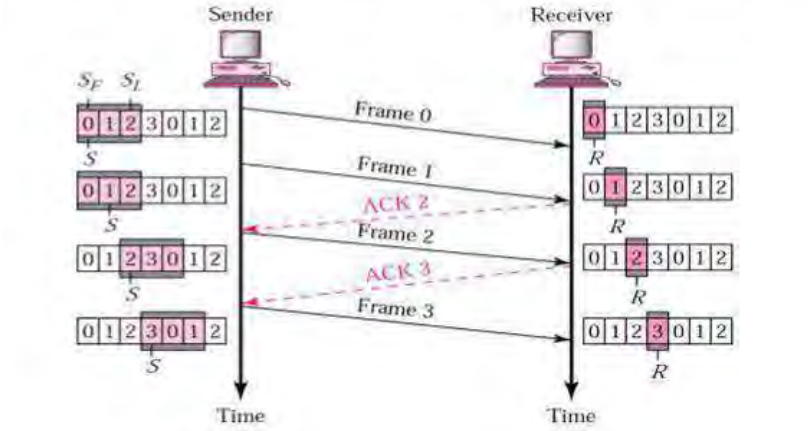
In Go-Back-N ARQ method, both sender and receiver maintain a window. The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. This is the advantage here that, burst of data can send at a time which is not happened in previous mechanism. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has

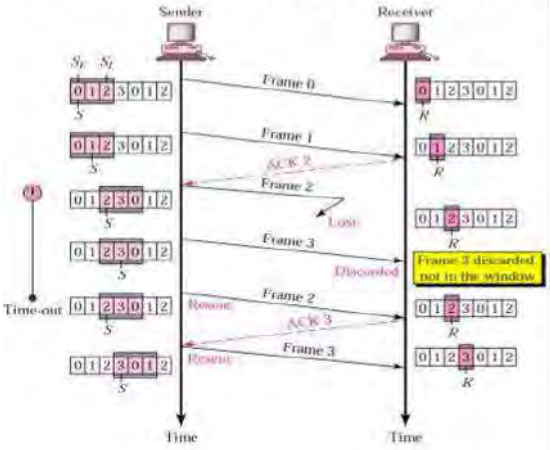
received NACK or has not received any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

## Go-Back-N ARQ, normal operation

- The sender keeps track of the outstanding frames and updates the variables and windows as the ACKs arrive.



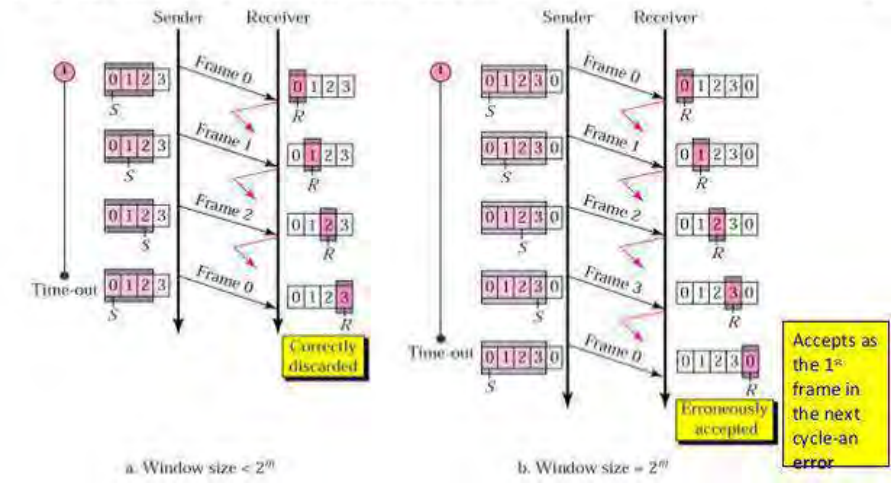

## Go-Back-N ARQ, lost frame




- Frame 2 is lost
- When the receiver receives frame 3, it discards frame 3 as it is expecting frame 2 (according to window).
- After the timer for frame 2 expires at the sender site, the sender sends frame 2 and 3. (go back to 2)

## Go-Back-N ARQ, sender window size

- Size of the sender window must be less than $2^m$. Size of the receiver is always 1. If m = 2, window size = $2^m - 1 = 3$.
- Fig compares a window size of 3 and 4.




**Sequence Numbers**

Sequence numbers are provided to the data frame to hold the sequence number of each frame. The range of the sequence numbers is chosen as small as to minimize the frame size so that it provides unambiguous communication. The sequence numbers of course can wrap around. For example, the field size is *m* bits long, the sequence numbers start from 0, go to $\boldsymbol{2^m - 1}$, and then are repeated. In other words, the sequence numbers are modulo-$2^m$.

Let us assume field size m=3, sequence numbers are:0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7…..and so on.

Let us reason out the range of sequence numbers we need.

Let have used *x* as a sequence number; we only need to use *x* + 1 after that, no need for x+2. Reason for this is, assume that the sender has sent the frame numbered *x*. possible things occurred are 1. The frame arrives safe and sound at the receiver site on receiving the receiver sends an acknowledgment. The acknowledgment arrives at the sender site, requesting the sender to send the next frame numbered *x* + 1.

2. Second time the frame arrives correctly, receiver sends an acknowledgment, but the acknowledgment is corrupted or lost somewhere else and time has expired. The sender resends the frame (numbered *x)* after the time-out. **Note** that the frame here is a duplicate. The receiver can recognize this fact because it expects frame *x* + 1 but frame *x* was received and simply discard.

3. if the frame is corrupted or never arrives at the receiver site; the sender resends the frame (numbered *x)* after the time-out. We can see that there is a need for sequence numbers $x$ and $x + 1$ because the receiver needs to distinguish between case 1 and case 2. But there is no need for a frame to be numbered $x + 2$. In case 1, the frame can be numbered $x$ again because frames $x$ and $x + 1$ are acknowledged and there is no ambiguity at either site. In cases 2 and 3, the new frame is $x + 1$, not $x + 2$. If only $x$ and $x + 1$ are needed, we can let $x = 0$ and $x + 1 == 1$. This means that the sequence is 0, 1, 0, 1, 0, and so on.

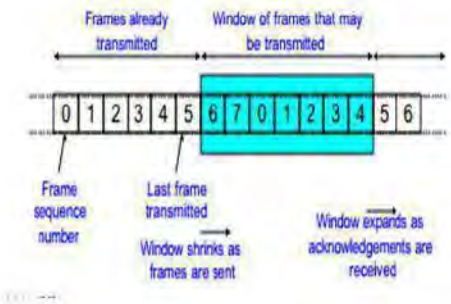
***Acknowledgment Numbers***

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

***Sliding Window***

The sliding window is an abstract concept that defines the range of sequence numbers that concern the sender and receiver. The sender and receiver deal with only part of the possible sequence numbers. Window concern of the sender is called the send sliding window; the window that is the concern of the receiver is called the receive sliding window. The send window is an imaginary box covering the sequence numbers of the data frames which can be in transit. In each window position, some sequence numbers of the frames define that have been sent; others define those that can be sent. The maximum size of the window is $2^m - 1$ for reasons that we discuss later. The window divides the possible sequence numbers into four regions. The first region defines the sequence numbers belonging to frames that are already acknowledged. The second region in the above figure defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We .The third defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer. Finally, the fourth region defines sequence numbers that cannot be used until the window slides, as we see next.

Figure: **Sending Window**




**Sender Sliding Window**

- At the sending site, to hold the outstanding frames until they are acknowledged, we use the concept of a window.
- The size of the window is at most $2^m$ -1 where m is the number of bits for the sequence number.
- Size of the window can be variable, e.g. TCP.
- The window slides to include new unsent frames when the correct

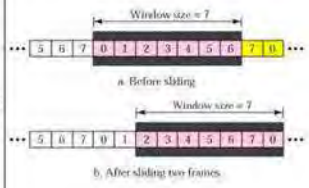
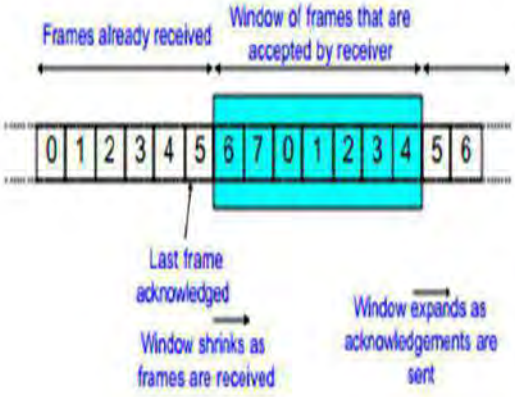
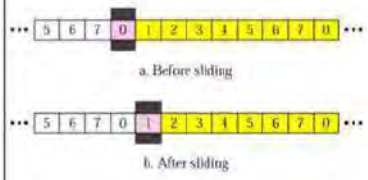



Figure: **Receiving Window**



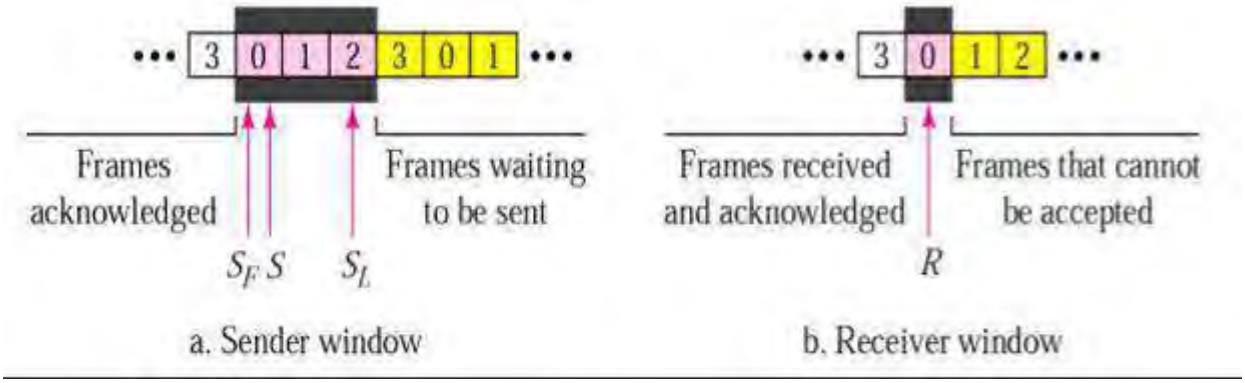

**Receiver Sliding Window**

- Size of the window at the receiving site is always 1 in this protocol.
- Receiver is always looking for a specific frame to arrive in a specific order.
- Any frame arriving out of order is discarded and needs to be resent.
- Receiver window slides as shown in fig. Receiver is waiting for frame 0 in part a.




## Control Variable

- Sender has 3 variables: S, $S_F$, $S_L$.
- S holds the sequence number of recently sent frame.
- $S_F$ holds the sequence number of first frame.
- $S_L$ holds the sequence number of the last frame.
- Receiver only has the one variable, R, that holds the sequence number of the frame it expects to receive. If the sequence number is the same as the value of R, the frame is accepted, otherwise rejected.




***Acknowledgment***

The receiver sends a positive acknowledgment if a frame has arrived safe and sound and in order. If a frame is damaged or is received out of order, the receiver is silent and will discard all subsequent frames until it receives the one it is expecting. The silence of the receiver causes the timer of the unacknowledged frame at the sender site to expire. This, in turn, causes the sender to go back and resend all frames, beginning with the **one with the expired timer**. The receiver does not have to acknowledge each frame received. It can send one cumulative acknowledgment for several frames.

***Resending a Frame***

When the timer expires, the sender resends all outstanding frames. For example, suppose the sender has already sent frame 6, but the timer for frame 3 expires. This means that frame 3 has not been acknowledged; the sender goes back and sends frames 3, 4,5, and 6 again. That is why the protocol is called *Go-Back-N* ARQ.

## SELECTIVE REPEAT AUTOMATIC REPEAT REQUEST

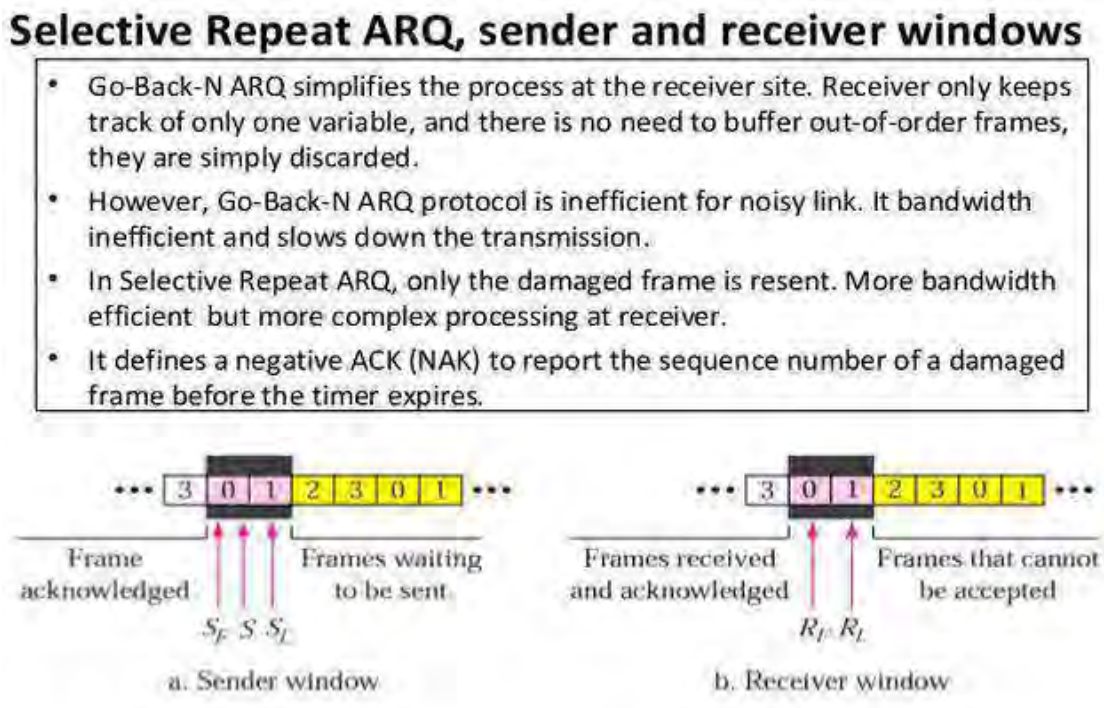
This protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not resend *N* frames when just one frame is damaged, only the damaged frame is resent. This mechanism is called **Selective Repeat ARQ**. It is more efficient for noisy links, but the processing at the receiver is more complex.

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.

In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK (negative ACK) for only frame which is missing or damaged. The sender in this case, sends only packet for which NACK is received.

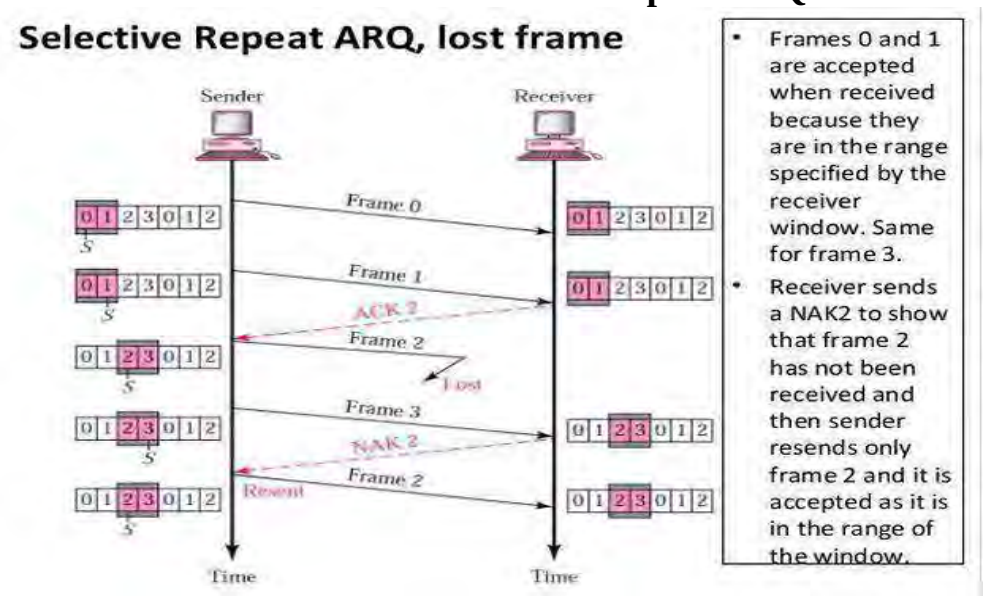
***Windows***

The Selective Repeat Protocol also uses two windows: a send window and a receive window. There are differences between the windows in this protocol and the ones in Go-Back-N. First, the size of the send window is much smaller; it is half of the $2^m$. Second, the receive window is the same size as the send window. For example, if $m$ = 4, the sequence numbers go from 0 to 15, but the size of the window is just 8 (it is 15 in the *Go-Back-N* Protocol). The smaller window size means less efficiency in filling the pipe, but the fact that there are fewer duplicate frames can compensate for this. The protocol uses the same variables as we discussed for Go-Back-N.



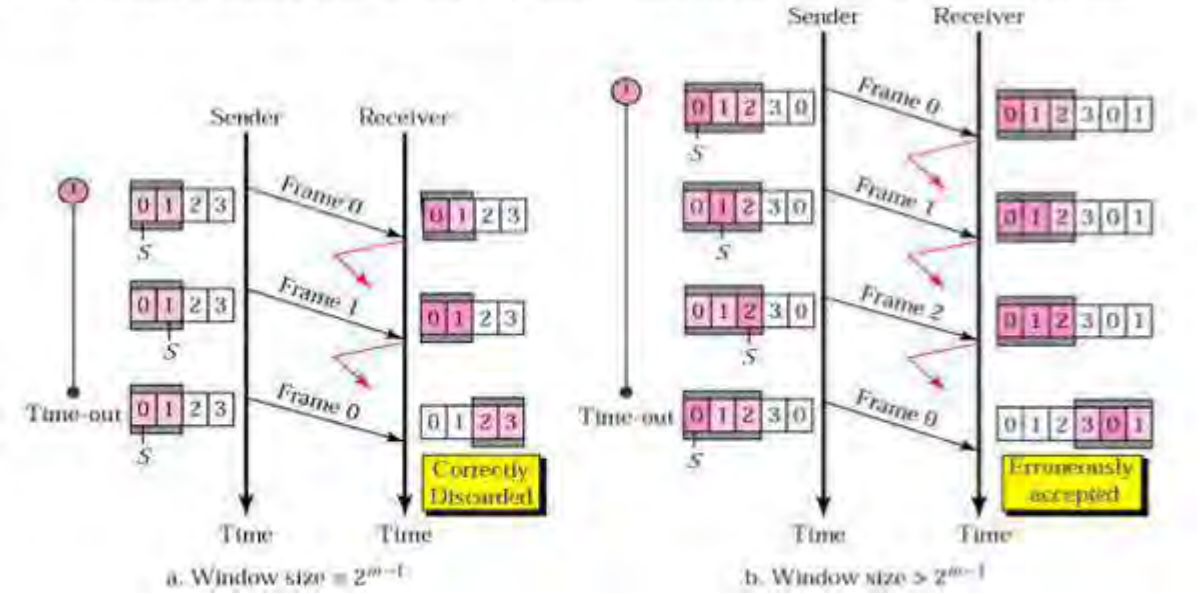

The Selective Repeat Protocol allows as many frames as the size of the receive window. Frame may arrive out of order and be kept store until there is a set of in-order frames then finally delivered to the network layer. Those slots inside the window that are colored in the above figure define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

**Receive window for selective repeat ARQ**




## Selective Repeat ARQ, sender window size

- Size of the sender and receiver windows must be at most one-half of $2^m$. If m = 2, window size should be $2^m/2 = 2$. Fig compares a window size of 2 with a window size of 3. Window size is 3 and all ACKs are lost, sender sends duplicate of frame 0, window of the receiver expect to receive frame 0 (part of the window), so accepts frame 0, as the 1st frame of the next cycle – an **error**.



**HDLC**

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms Configurations and Transfer Modes.

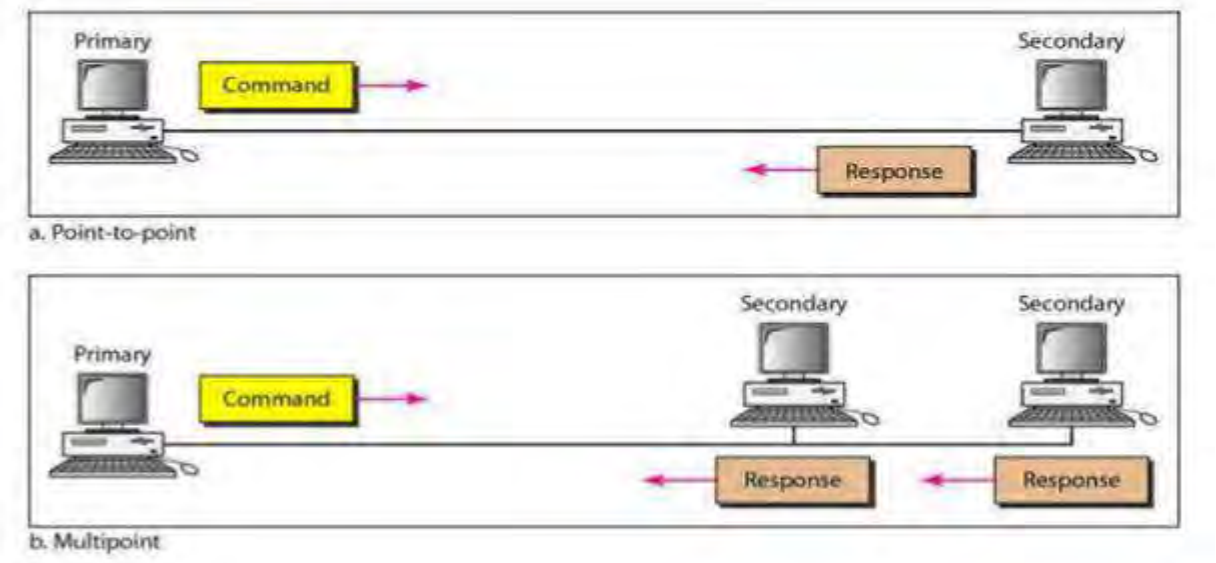
**Configurations and Transfer Modes**

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

**Normal Response Mode**

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

**Figure:Normal Response Mode (NRM)**




**Asynchronous Balanced Mode**

- This is the common mode today.
- In asynchronous balanced mode (ABM), the configuration is balanced.
- It is used for point-to-point link, and each station can function as a primary and a secondary (acting as peers).

## Asynchronous Balance Mode (ABM)

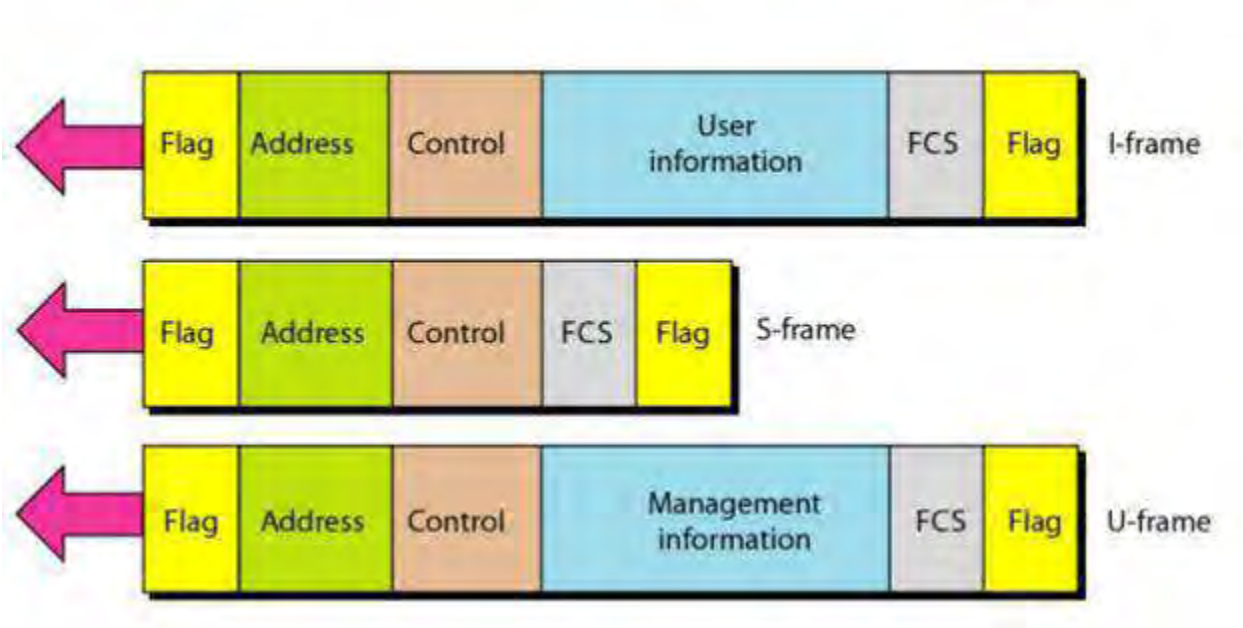

## Frames type of HDLC

HDLC defines three types of frames: **information frames (I-frames)**, **supervisory frames (S-frames)**, and **unnumbered frames (U-frames)**. Each type of frame serves as an envelope for the transmission of a different type of message. **I-frames** are used to transport user data and control information relating to user data (piggybacking). **S-frames** are used only to transport control information. **U-frames** are reserved for system management. Information carried by u-frames is intended for managing the link itself.

### *Frame Format*

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

Figure: ***HDLC frames***




***Fields***

Let us now discuss the fields and their use in different frame types.

**Flag field**: The flag field of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame and serves as a synchronization pattern for the receiver.

**Address field:** it contains the address of the secondary station. If a primary station created the frame, it contains a "*to* address". If a secondary creates the frame, it contains *a "from"* address. An address field can be 1 byte or several bytes long, depending on the needs of the network. One byte can identify up to 128 stations (l bit is used for another purpose). Larger networks require multiple-byte address fields. If the address field is only 1 byte, the last bit is always a 1. If the address is more than 1 byte, all bytes but the last one will end with 0; only the last will end with 1 for example for 3 byte:10110110 10101010 11101101.

**Control field:** The control field is a 1- or 2-byte segment of the frame used for flow and error control. The interpretation of bits in this field depends on the frame type.

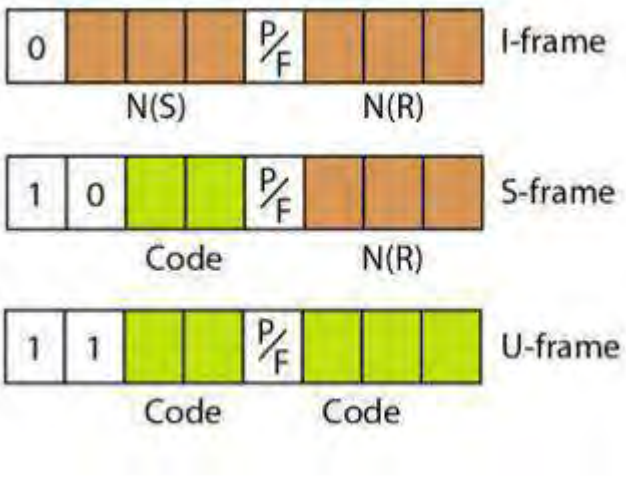
**Information field:** The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.

**FCS field:** The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte ITU-T CRC.

**Control Field**

HDLC defines three types of frame. Each of the control field determines the type of frame and defines its functionality.

**Figure:** ***Control field format for the different frame types***




***Control Field for I-Frames***

I-frames are designed to carry user data from the network layer. They can also include flow and error control information (piggybacking). Subfields of this control field are defined its function as: The first bit defines the type. If its first bit in the control field is 0, the frame is an I-frame.

The next 3 bits, called *N(S),* define the sequence number of the frame. Note that with 3 bits, we can define a sequence number between 0 and 7. This value of this field corresponds to the value of the control variable s. The single bit between *N(S)* and *N(R)* is called the *PIF* bit. The *PIF* 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).

The last 3 bits, called *N(R),* correspond to the acknowledgment number when piggybacking is used.

***Control Field for S-Frames***

Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate (e.g., 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. If the first 2 bits of the control field is **10**, this means the frame is an S-frame. The next 2 bits called **code** is used to define the type of S-frame itself. With 2 bits, we can have four types of S-frames, as described below:

- **Receive ready (RR)**: If the value of the code subfield is 00, it is an RR S-frame.

This kind of frame acknowledges the receipt of a safe and sound frame or group of frames. . In this case, the value *N(R)* field defines the acknowledgment number.

- **Receive not ready (RNR):** If the value of the code subfield is 10, it is an RNRS-frame. This kind of frame is an RR frame with additional functions. It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames. It acts as a kind of **congestion control mechanism** by asking the sender to slow down. The value of *N(R)* is the acknowledgment number.

- **Reject (REJ):** If the value of the code subfield is **01**, it is a REJ S-frame. This is a NAK frame, but not like the one used for Selective Repeat ARQ. It is a NAK that can be used in *Go-Back-N* ARQ to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged. The value of *N(R)* is the negative acknowledgment number.
- **Selective reject (SREJ):** If the value of the code subfield is **11**, it is an SREJ S-frame.

This is a NAK frame used in Selective Repeat ARQ. Note that the HDLC Protocol uses the term ***selective reject*** instead of ***selective repeat.*** The value of *N(R)* is the negative acknowledgment number.

***CONTROL FIELD FOR 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 that contain for system management information, not user data.

U-frame codes are divided into two sections: a 2-bit prefix before the P/F bit and a 3-bit suffix after the P/F bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames. Some of the more common types are shown below:

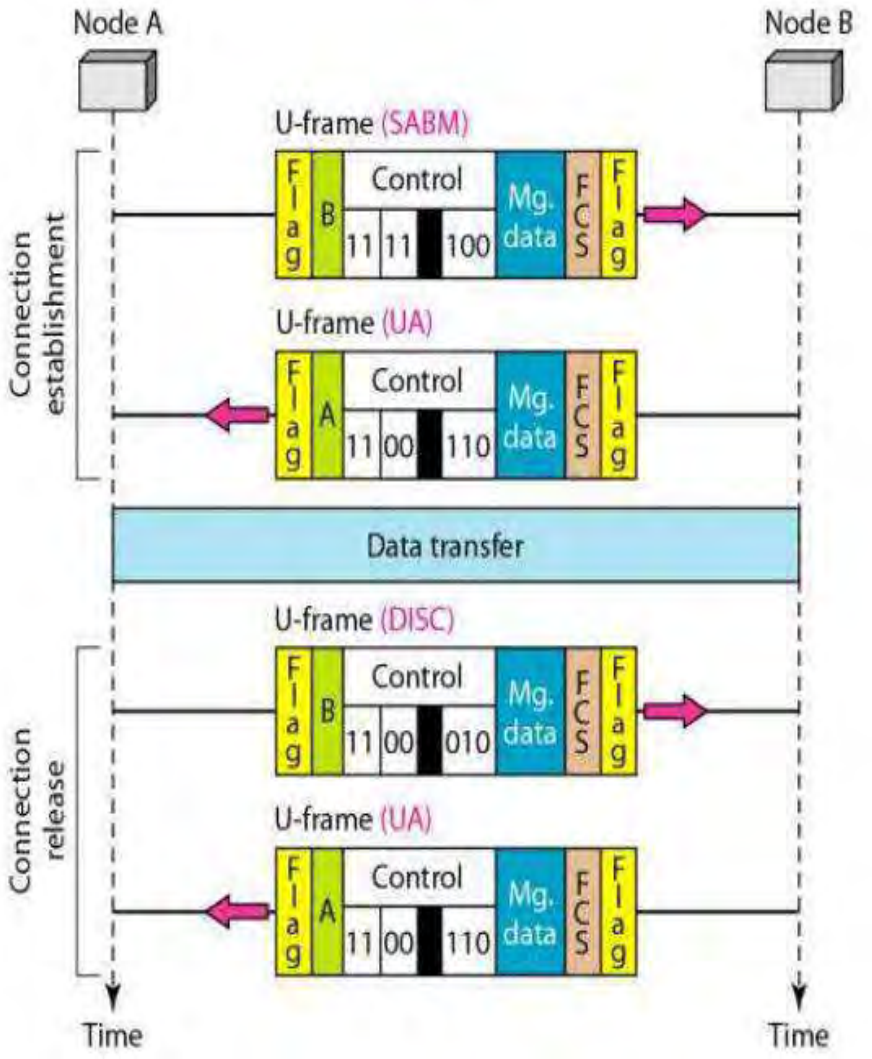
Table: **U- frame *control command and response***

| *Code* | *Command* | *Response* | *Meaning* |
|---|---|---|---|
| **00 001** | SNRM | | Set normal response mode |
| **11 011** | SNRME | | Set normal response mode, extended |
| **11 100** | SABM | **DM** | Set asynchronous balanced mode or **disconnect mode** |
| **11 110** | SABME | | Set asynchronous balanced mode, extended |
| **00 000** | UI | **UI** | Unnumbered information |
| **00 110** | | **UA** | **Unnumbered acknowledgment** |
| **00 010** | DISC | **RD** | Disconnect or **request disconnect** |
| **10 000** | SIM | **RIM** | Set initialization mode or **request information mode** |
| **00 100** | UP | | Unnumbered poll |
| **11 001** | RSET | | Reset |
| **11 101** | XID | **XID** | Exchange ID |
| **10 001** | FRMR | **FRMR** | Frame reject |

*Example:* ***Connection/Disconnection***

U-frames can be used for connection establishment and connection release. Node A asks for a connection with a set asynchronous balanced mode (SABM) frame. node B gives a positive response with an unnumbered acknowledgment (UA) frame. After these two exchanges, data can be transferred between the two nodes (not shown in the figure). After data transfer, node A sends a DISC (disconnect) frame to release the connection; it is confirmed by node B responding with a UA (unnumbered acknowledgment).

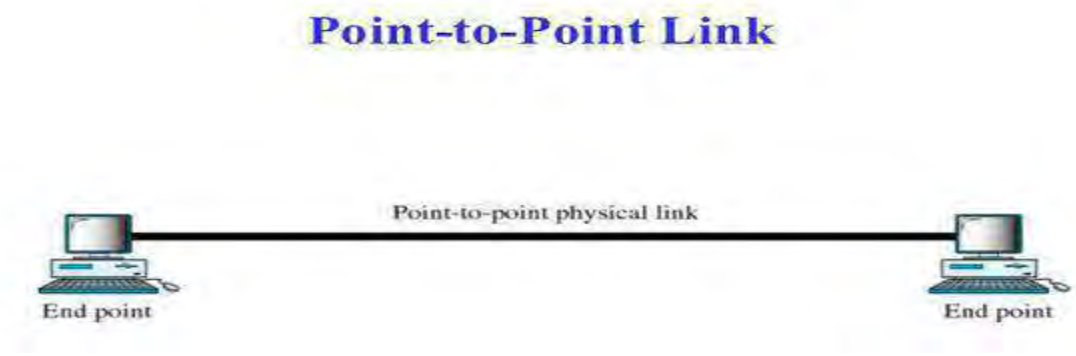
Figure: **Connection establishment**

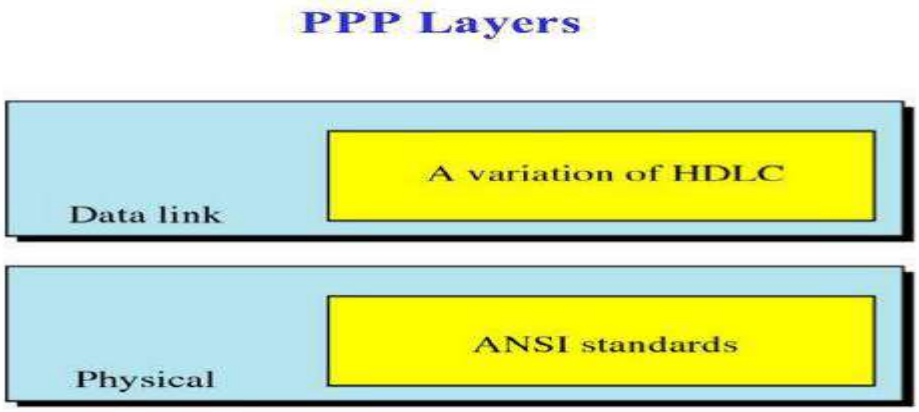



***LECTURE NOTE: 14***

# POINT-TO-POINT PROTOCOL

The most common protocols for point-to-point access is the Point-to-Point Protocol (PPP). Most of the Internet users who need to connect their home computers to the server of an Internet service provider use PPP. It control and manage the transfer of data at the data link layer




**PPP provides several services:**

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

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

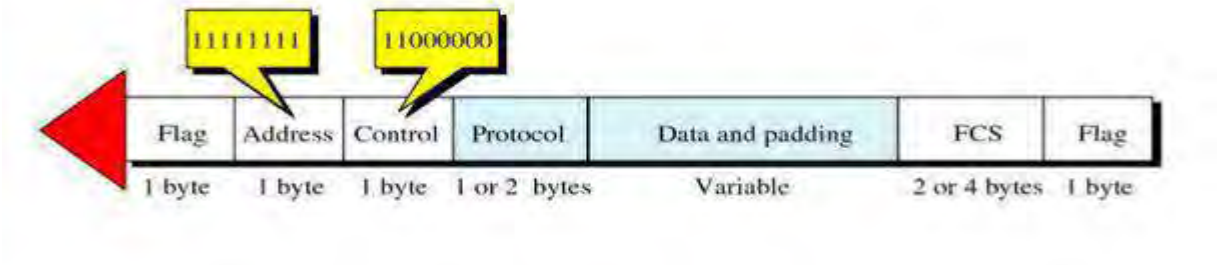
1. PPP does not provide flow control. A sender can send several frames one after another with no concern about overwhelming the receiver.
2. PPP has a very simple mechanism for error control. A CRC field is used to detect errors. If the frame is corrupted, it is silently discarded; the upper-layer protocol needs to take care of the problem.
3. PPP does not provide a sophisticated addressing mechanism to handle frames in a multipoint configuration.

**FRAMING**

PPP is a byte-oriented protocol.

***FRAME FORMAT***

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




**Flag:** it Indicates frames beginning or end. It is of a 1-byte start with the bit pattern 01111110. This pattern is the same as that used in HDLC, but difference is that PPP is a byte-oriented protocol where as HDLC is a bit-oriented protocol.

**Address:** The address field in this protocol is a constant value and set to 11111111 (broadcast address). Or it is used for broadcast address (destination address). During negotiation, the two parties may agree to omit this byte.

**Control**: This field is set to the constant value 1100000. As PPP does not provide any flow control and Error control for error detection this field is not needed at all, so two parties can agree, during negotiation, to omit this byte.

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

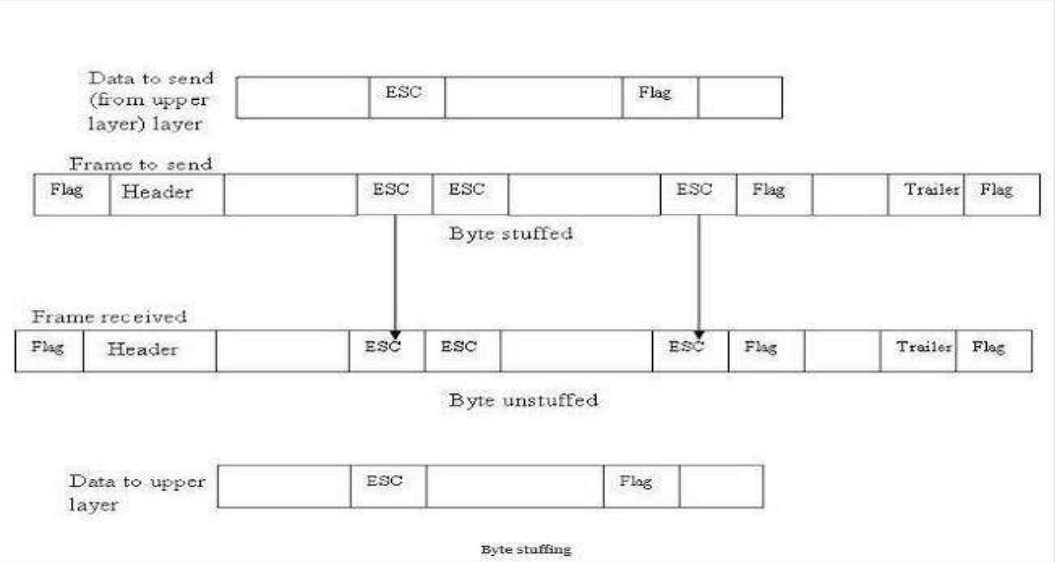
**Payload field:** This field carries either the user data or other. The data field is a sequence of bytes with the default of a maximum of 1500 bytes, but this can be changed during negotiation. The data field is byte stuffed if the flag byte pattern appears in this field. Because there is no field defining the size of the data field, padding is needed if the size is less than the maximum default value or the maximum negotiated value.

**FCS:** The frame check sequence (FCS) is simply a 2-byte or 4-byte standard CRC. Or in other words it is 2 or more bytes: - error check sum

***Byte Stuffing and bit stuffing***

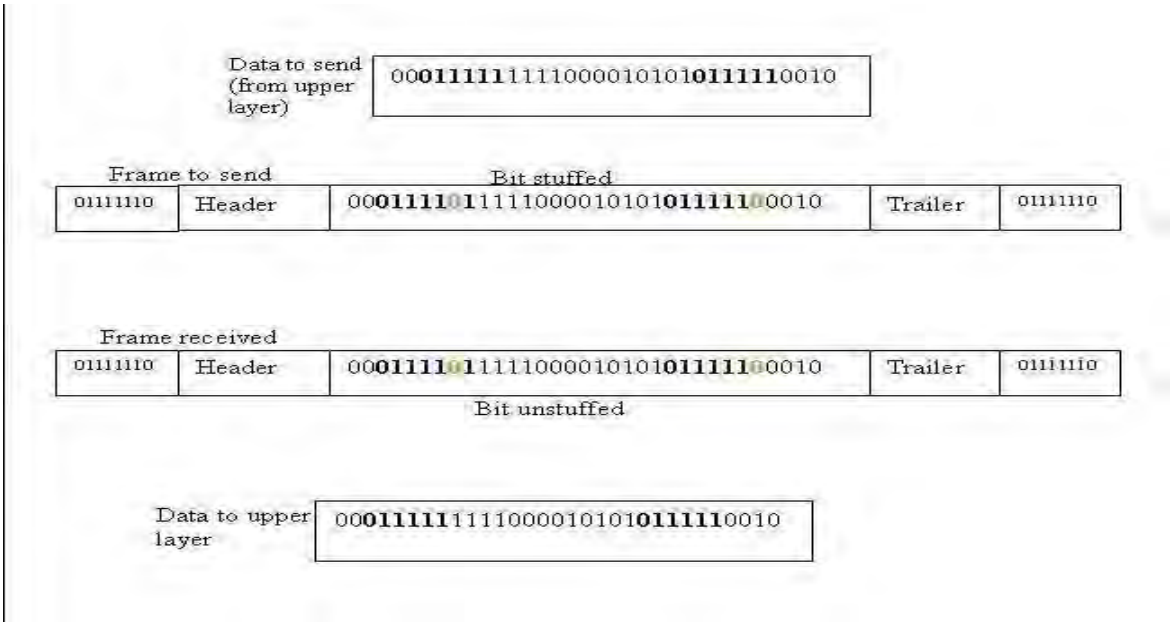
A **byte stuffing** (**also known as character stuffing**) approach was included to character-oriented protocol. The escape byte is 01111101, used as an extra bit that stuffed to tell the receiver that the next byte is not a flag. In byte stuffing a special byte is add to the data part, this is known as **escape character** (ESC). The escape characters have a predefined pattern. The receiver removes the escape character and keeps the data part. if the text contains escape characters as part of data, it causes another problem. To deal with this, an escape character is prefixed with another escape character.

Note: **Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data. Byte stuffing is the method of adding 1 extra byte if there is a flag or escape character in the text.**



## BIT-ORIENTED PROTOCOLS

In a bit-oriented protocol, the data to send is a series of bits. In order to distinguish frames, most protocols use a bit pattern of 8-bit length (01111110) as flag at the beginning and end of each frame. Here also cause the problem of appearance of flag in the data part to deal with this an extra bit added. This method is called **bit stuffing.**
In bit stuffing, if a 0 and five successive 1 bits are encountered, an extra 0 is added. The receiver node removes the extra-added zero. This process is illustrated below:




### Transition Phases

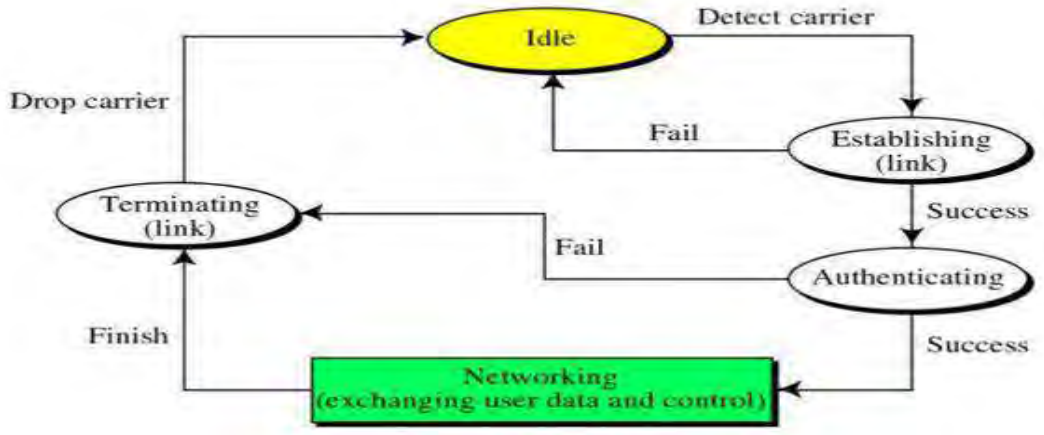
A PPP connection goes through several phases which can be shown in a transition phase diagram

**Dead:** In the dead phase the link is not being used. There is no active carrier (at the physical layer) and the line is quiet.

**Establish**: When one of the nodes starts the communication, the connection goes into this phase. In this phase, options are negotiated between the two parties. If the negotiation is successful, the system goes to the authentication phase (if authentication is required) or directly to the networking phase. Several packets may be exchanged here. LCP packets are used for this purpose.

**Authenticate:** The authentication phase is optional. The two nodes may decide, during the establishment phase, not to skip this phase. To proceed with authentication, they send several authentication packets. If the result is successful, the connection goes to the networking phase; otherwise, it goes to the termination phase.

**Network:** In the network phase, negotiation for the network layer protocols takes place. PPP specifies that two nodes establish a network layer agreement before data at the network layer can be exchanged. The reason is that PPP supports multiple protocols at the network layer. If a node is running multiple protocols simultaneously at the network layer, the receiving node needs to know which protocol will receive the data.




**Open:** In the open phase, data transfer takes place. When a connection reaches this phase, the exchange of data packets can be started. The connection remains in this phase until one of the endpoints wants to terminate the connection.

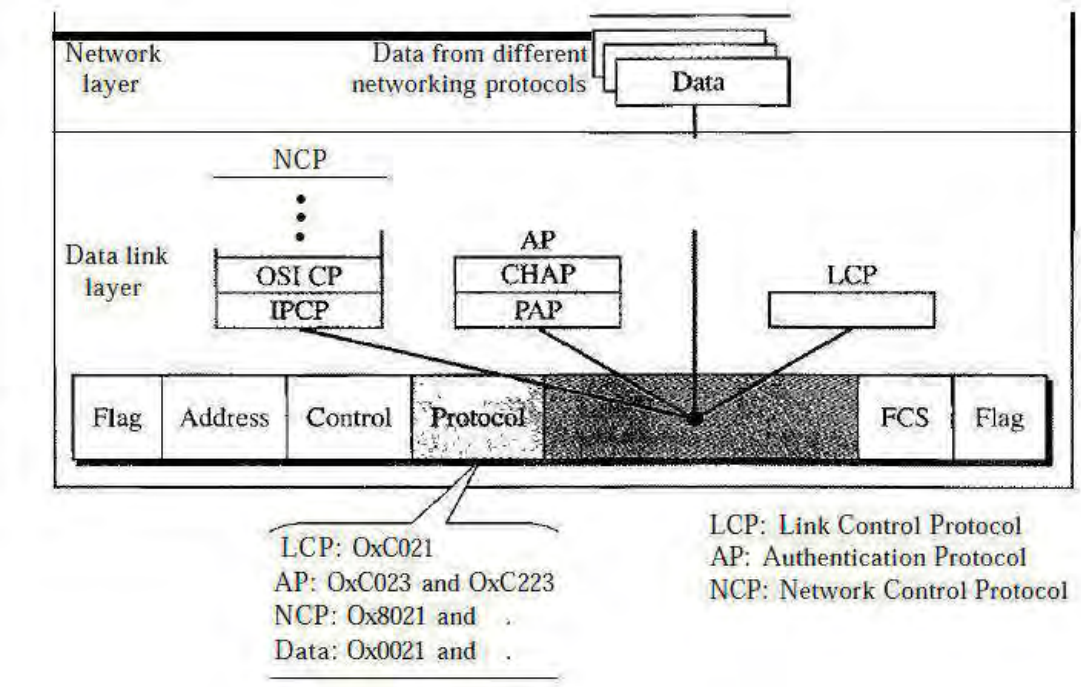
**Terminate:** In the termination phase the connection is terminated. Several packets are exchanged between the two ends and closing the link. After data transfer connection is terminated.

**Multiplexing**

PPP uses another set of other protocols to establish the link, authenticate the parties involved, and carry the network layer data. Three sets of protocols are defined to make PPP powerful are: the **Link Control Protocol (LCP)**, two **Authentication**

**Protocols (APs)**, and several **Network Control Protocols (NCPs)**. At any moment, a PPP packet can carry data from one of these protocols in its data field, as shown in Figure below. Note that there is one LCP, two APs, and several NCPs. Data may also come from several different network layers.

***Figure: Multiplexing in PPP***




PPP uses several other protocols to establish link, authenticate users and to carry the network layer data.

The various protocols used are:

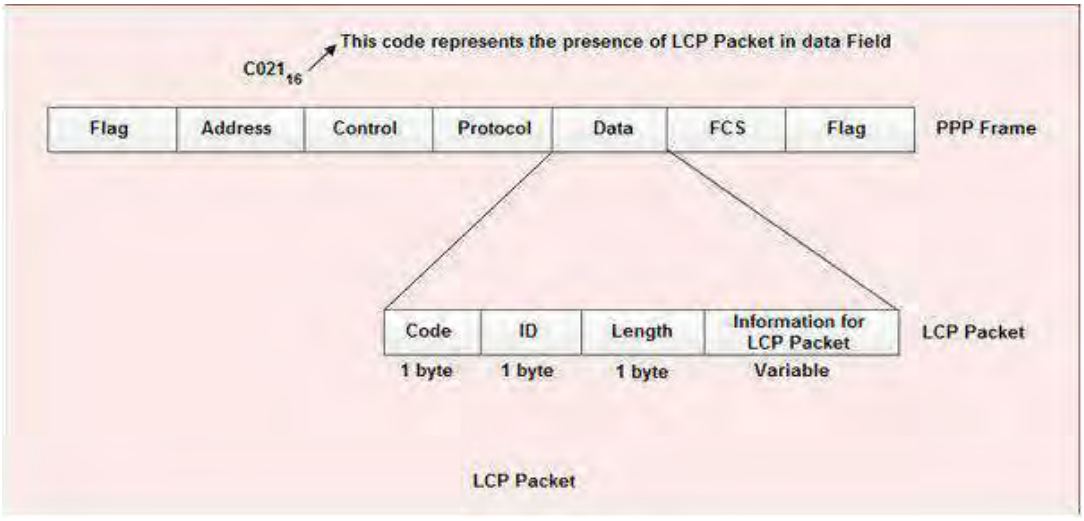
**1. Link Control Protocol**

**2. Authentication Protocol**

**3. Network Control Protocol**

***LINK CONTROL PROTOCOL***

- The **Link Control Protocol** (LCP) is responsible for establishing, maintaining, configuring, and terminating links.
- It provides negotiation mechanisms to set options between the two endpoints. Both endpoints of the link must reach an agreement about the options before the link can be established.
- All LCP packets are carried in the data field of the PPP frame.
- The presence of a value $C021_{16}$ in the protocol field of PPP frame indicates that LCP packet is present in the data field.
- The various fields present in LCP packet are:




1. **Code**: 1 byte-specifies the type of LCP packet.

2. I**D**: 1 byte-holds a value used to match a request with the reply.

3. **Length**: 2 byte-specifies the length of entire LCP packet.

4. **Information**: Contains extra information required for some LCP packet.

There are 11 different types of LCP packets as shown in Table. These are categorized in three groups:

1. **Configuration packet**: These are used to negotiate options between the two ends. For example: configure-request, configure-ack, configure-nack, configure-reject are some configuration packets.

2. **Link termination packets**: These are used to disconnect the link between two end points. For example: terminate-request, terminate-ack, are some link termination packets.

3. **Link monitoring and debugging packets**: These are used to monitor and debug the links. For example: code-reject, protocol-reject, echo-request, echo-reply and discard-request are some link monitoring and debugging packets.

From the below table, First four packet types, is used for link configuration during the establish phase. The packet types 5 and 6, is used for link termination during the termination phase. The last five packets are used for link monitoring and debugging.

| *Code* | *Packet Type* | *Description* |
|---|---|---|
| OxOl | Coofigure-request | Contains the list of proposed options and their values |
| Ox02 | Configure-ack | Accepts all options proposed |
| Ox03 | Configure-nak | Announces that some options are oot acceptable |
| Ox04 | Configure-reject | Announces that some options are not recognized |
| Ox05 | Terminate-request | Request to shut down the line |
| Ox06 | Terminate-ack | Accept the shutdown request |
| Ox07 | Code-reject | Announces an unknown code |
| Ox08 | Protocol-reject | Announces an unknown protocol |
| Ox09 | Echo-request | A type of hello message to check if the other end is alive |
| OxOA | Echo-reply | The response to the echo-request message |
| OxOB | Discard-request | A request to discard the packet |

There are many options that can be negotiated between the two endpoints. Options are inserted in the information field of the configuration packets. the information field is divided into three fields: option type, option length, and option data. Some of the most common options in Table are:

Table: ***Common options***

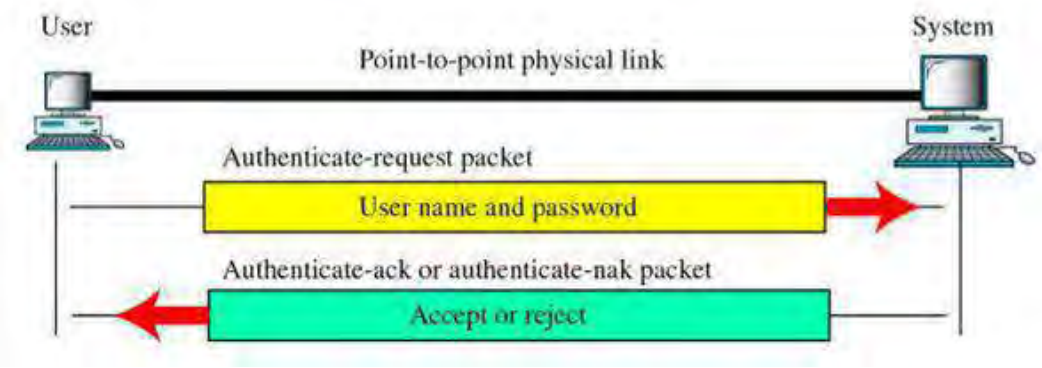
| *Option* | *Default* |
|---|---|
| Maximum receive unit (payload field size) | 1500 |
| Authentication protocol | None |
| Protocol field compression | Off |
| Address and control field compression | Off |

## 2. AUTHENTICATION PROTOCOL

PPP is designed for use over dial-up links where verification of user identity is necessary. **Authentication** means validating the identity of a user who needs to access the resources.

PPP has created two protocols for authentication: **Password Authentication Protocol** and **Challenge Handshake Authentication Protocol**. **Note** that these protocols are used during the authentication phase.

Figure: password authentication protocol




### i) PAP (Password Authentication Protocol)

it is a simple protocol provides two step authentication procedures as:

**STEP 1**. The user who wants to access a system sends authentication identification usually the **user name** and a **password**.
**STEP 2**. The system checks the validity of the user name and password and either accepts
or denies connection.

• PAP packets are also carried in the data field of PPP frames.

• The presence of PAP packet is identified by the value **$C023_{16}$** in the protocol field of PPP frame.

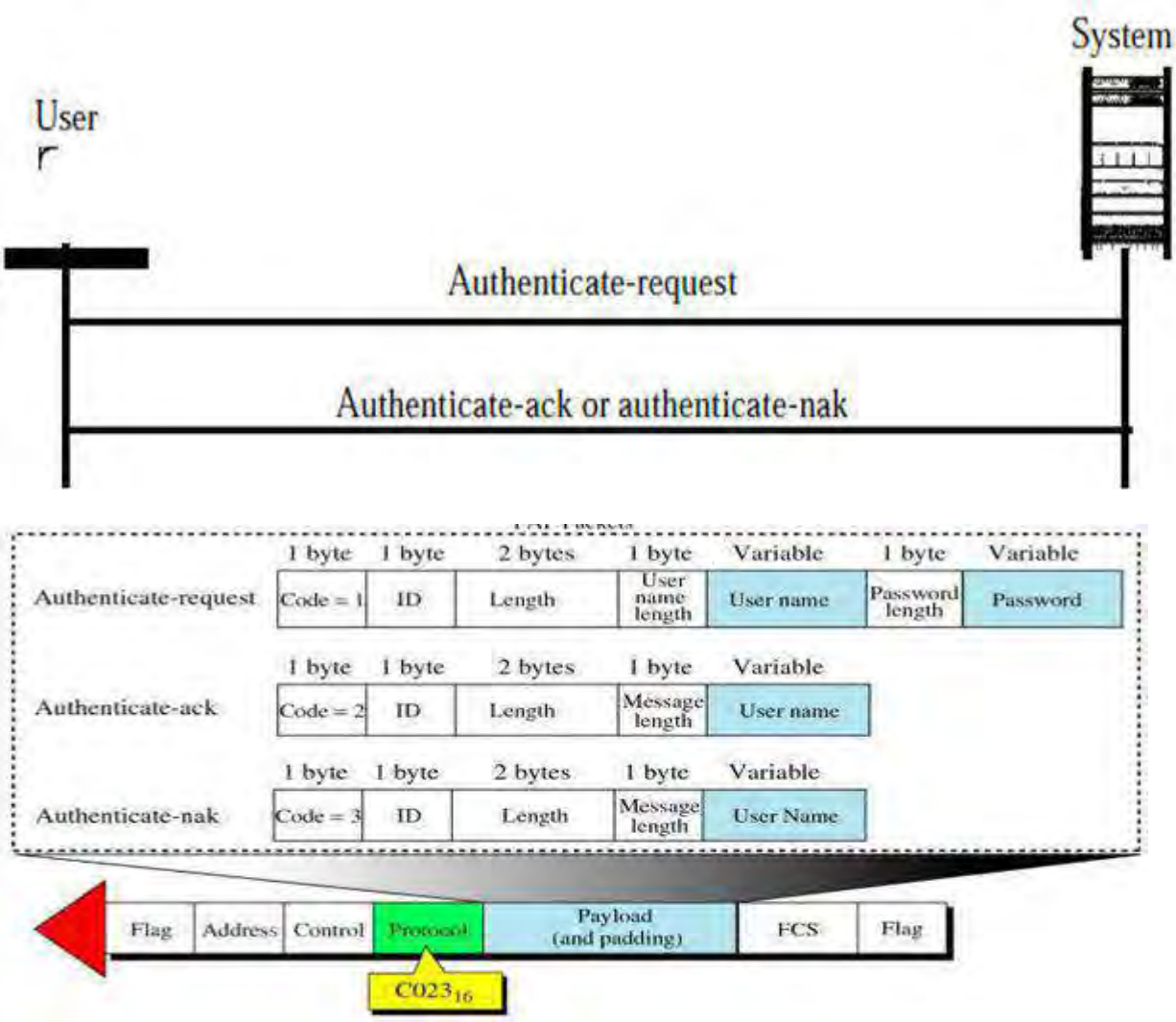
Three types of packets used by PAP and how they are actually exchanged are:

i) **Authenticate-request**: used to send user name & password.

ii) **Authenticate-ack**: used by system to allow the access.

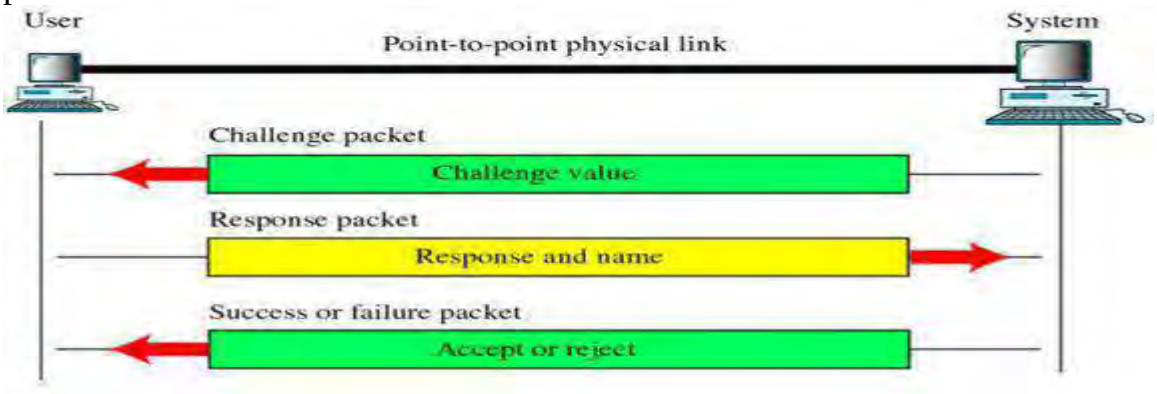
iii) **Authenticate-nak**: used by system to deny the access.

***PAP packets encapsulated in a PPP frame***



## 2. CHAP (CHALLENGE HANDSHAKE AUTHENTICATION PROTOCOL)

It provides greater security than PAP. **In** this method, the password is kept secret, it is never sent online. It is a three-way hand-shaking authentication protocol.



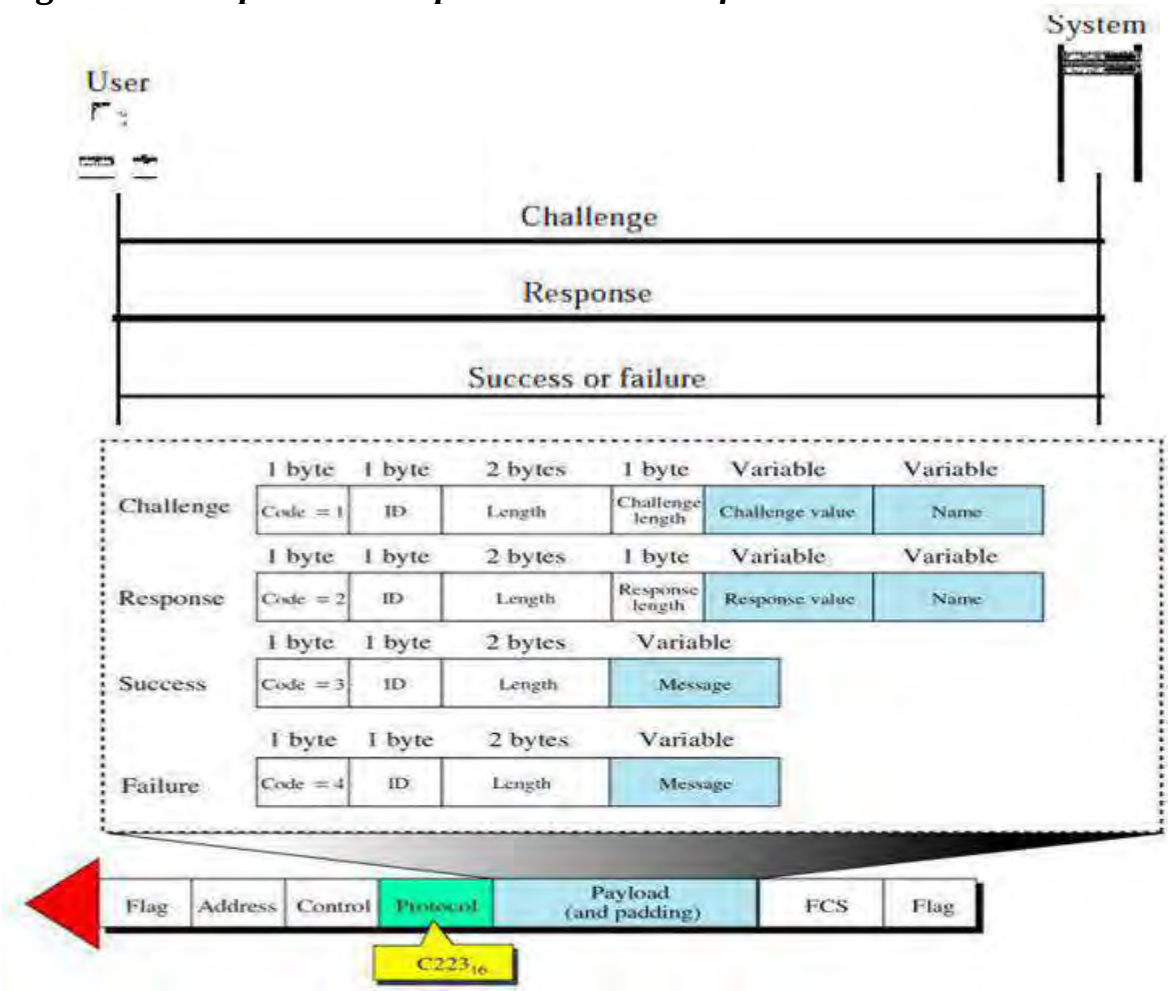

1. System sends a challenge packet containing a value, usually a few bytes to the user.

2. Using a predefined function to take challenge value with the user password that create result and sends the resultant packet back to the system.

3. System then applies the same function to the password of the user (known to the system) and challenge value and creates a result. If result created is same as the result sent in the response packet, access is granted, otherwise, it is denied.

CHAP is more secure than PAP, especially if the system continuously changes the challenge value. Even if the intruder learns the challenge value and the result, the password is still secret.

**Figure:** ***CHAP packets encapsulated in a PPP frame***




CHAP packets are encapsulated in the PPP frame with the protocol value C223 in hexadecimal. **There are 4 types of CHAP packets: challenge, response, success, and failure**.

1. Challenge-used by system to send challenge value.
2. Response-used by the user to return the result of the calculation.
3. Success-used by system to allow access to the system.
4. Failure-used by the system to deny access to the system.

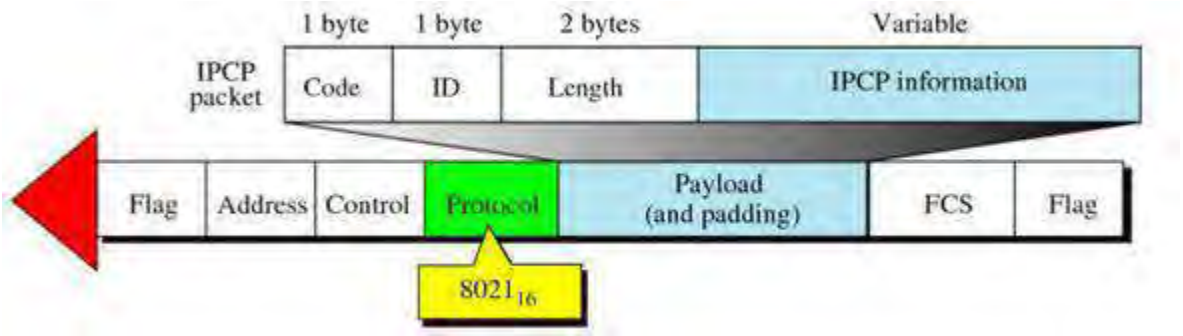
## 3. NETWORK CONTROL PROTOCOL (NCP)

PPP is a multiple-network layer protocol. It can carry a network layer data packet from protocols defined by the Internet, OSI, Xerox, DEC net, AppleTalk, Novel, and so on. To carry these packets PPP has defined a specific Network Control Protocol for each network protocol.

For example, IPCP (Internet Protocol Control Protocol) configures the link for carrying IP data packets. Xerox CP does the same for the Xerox protocol data packets, and so on.

**Note that** none of the NCP packets carry network layer data, they just configure the link at the network layer for the incoming data.

**IPCP Internet Protocol Control Protocol (IPCP)**: This protocol configures the link used to carry IP packets in the Internet. The format of an IPCP packet is shown below. **Note** that the value of the protocol field in hexadecimal is 8021.




IPCP defines seven packets, distinguished by their code values, as shown below. There are other NCP protocols for other network layer protocols. The OSI Network Layer Control Protocol has a protocol field value of 8023, the Xerox NS IDP Control Protocol has a protocol field value of 8025, and so on. The value of the code and the format of the packets for these other protocols are the same as shown in the table.

***Code value for IPCP packets***

| *Code* | *IPCP Packet* |
|---|---|
| OxO! | Configure-request |
| Ox02 | Configure-ack |
| Ox03 | Configure-nak |
| Ox04 | Configure-reject |
| Ox05 | Terminate-request |
| Ox06 | Terminate-ack |
| Ox07 | Code-reject |

• After establishing the link and authenticating the user, PPP connects to the network layer. This connection is established by NCP.

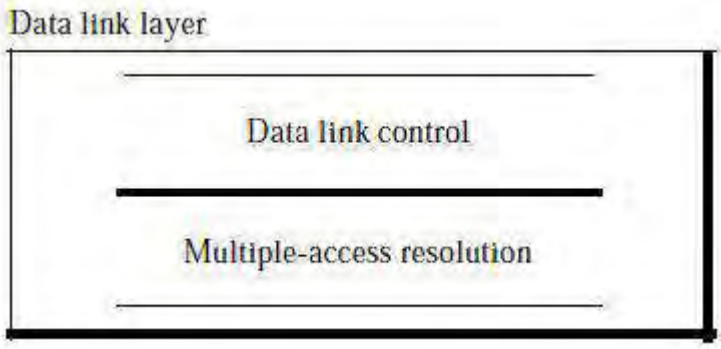
• Therefore NCP is a set of control protocols that allow the encapsulation of the data coming from network layer.

• After the network layer configuration is done by one of the NCP protocols, the users can exchange data from the network layer.

• PPP can carry a network layer data packet from protocols defined by the Internet, DECNET, Apple Talk, Novell, OSI, Xerox and so on.

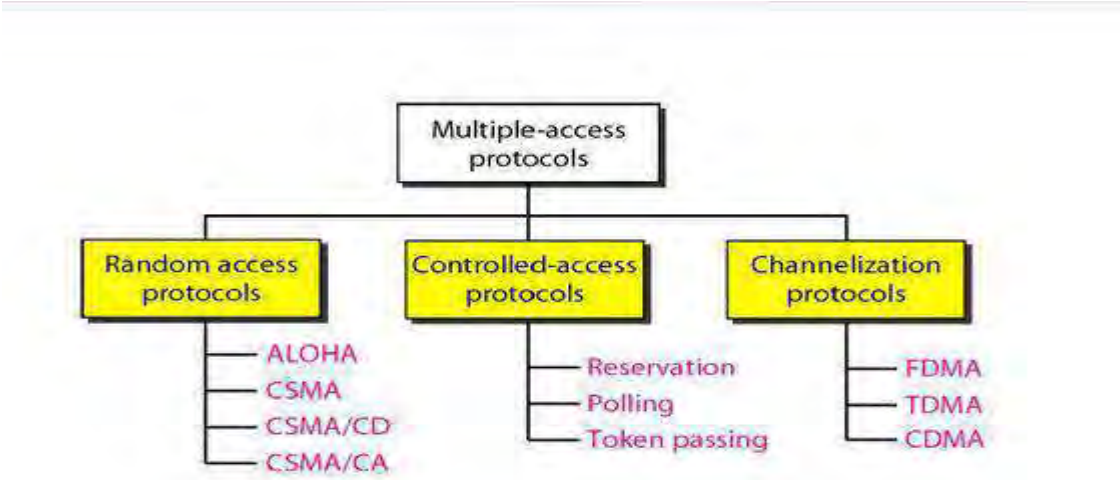
**MULTIPLE ACCESSES**

The data link layer is considered as two sub layers. The upper sub layer is responsible for data link control, and the lower sub layer is responsible for resolving access to the shared media. If the channel is dedicated, we do not need the lower sub layer.




The upper sub layer that is responsible **for flow and error control** is called the **logical link control (LLC)** layer; the lower sub layer that is mostly responsible for **multiple access resolution** is called the **media access control (MAC)** layer. When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.

Many formal protocols have been devised to handle access to a shared link. These protocols are categorized into three groups. Protocols belonging to each group are shown in Figure below




## RANDOM ACCESS

Its name itself implies that channel is accessed randomly. In a random access method, each station has the right to access the medium randomly without being controlled by any other station.

Two features give this method its name. **First**, it is **called *random access method because;*** there is no scheduled time for a station to transmit each station transmits randomly among them. **Second**, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called ***contention* methods.**

In a random access method, each station has the right to access the medium without being controlled by any other station. However, if more than one station tries to send frame or access the channel, there is an **access conflict-collision-**and the frames will be either **destroyed** or **modified.** To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:

- When can the station access the medium?
- What can the station do if the medium is busy?
- How can the station determine the success or failure of the transmission?
- What can the station do if there is an access conflict?

The random access methods have evolved from a protocol known as **ALOHA**, which used a very simple procedure called multiple accesses **(MA)**. The method was improved with the additional procedure that forces the station to sense the medium before transmitting. This was called **carrier sense multiple access**. This

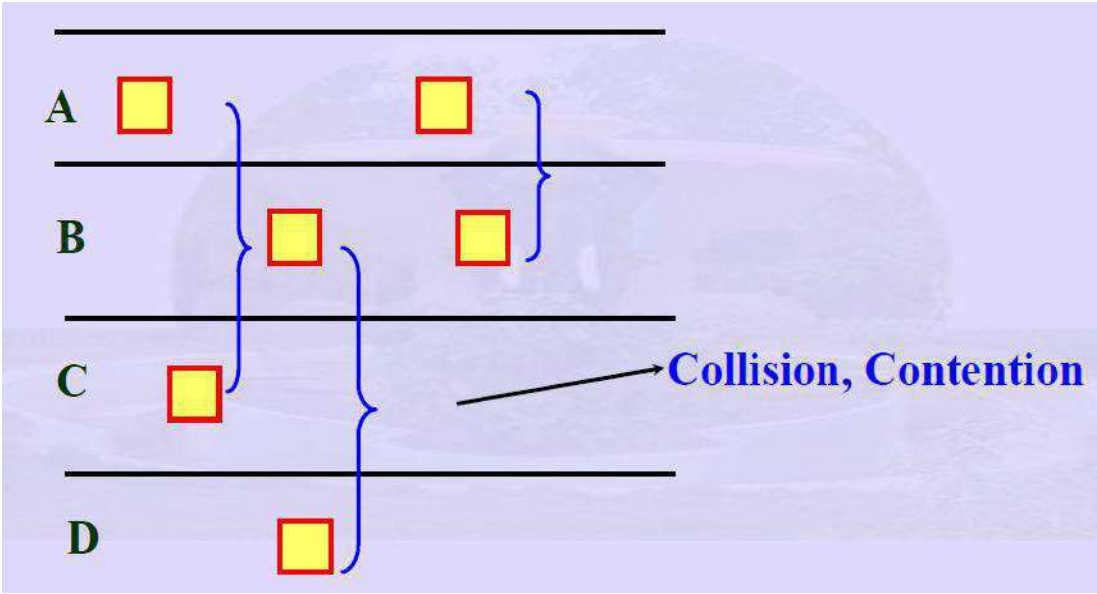
method again later evolved into two parallel methods: **carrier senses multiple access with collision detection (CSMA/CD)** and **carrier sense multiple accesses with collision avoidance *(CSMA/CA)***. *CSMA/CD* tells the station what to do when a collision is detected. *CSMA/CA* tries to avoid the collision.

**ALOHA**

**ALOHA** refers to a simple communications scheme in which each source (transmitter) in a network sends data whenever there is a **frame** to send. If the frame successfully reaches the destination (receiver), the next frame is sent. If the frame fails to be received at the destination, it is sent again. This is the earliest random access method or protocol was originally developed at the University of Hawaii in early 1970 for use with **satellite** communication systems in the Pacific. ALOHA served as the basis for the development of Ethernet and Wi-Fi networking.
This term is also known as ALOHA net.

It was designed for a radio (wireless) LAN, but it can be used on any shared medium. It is obvious that there are potential collisions in this arrangement; because the medium is shared between the stations. When a station sends data, another station may attempt to do so at the same time. The data from the two stations **collide** and become **garbled**. In a **wireless** broadcast system or a half-duplex two-way link, **Aloha** works perfectly. But as networks become more complex, for example in an **Ethernet** system involving multiple sources and destinations that share a common data path, trouble occurs because data frames **collide (conflict).** The heavier the communication is degradation in system efficiency.

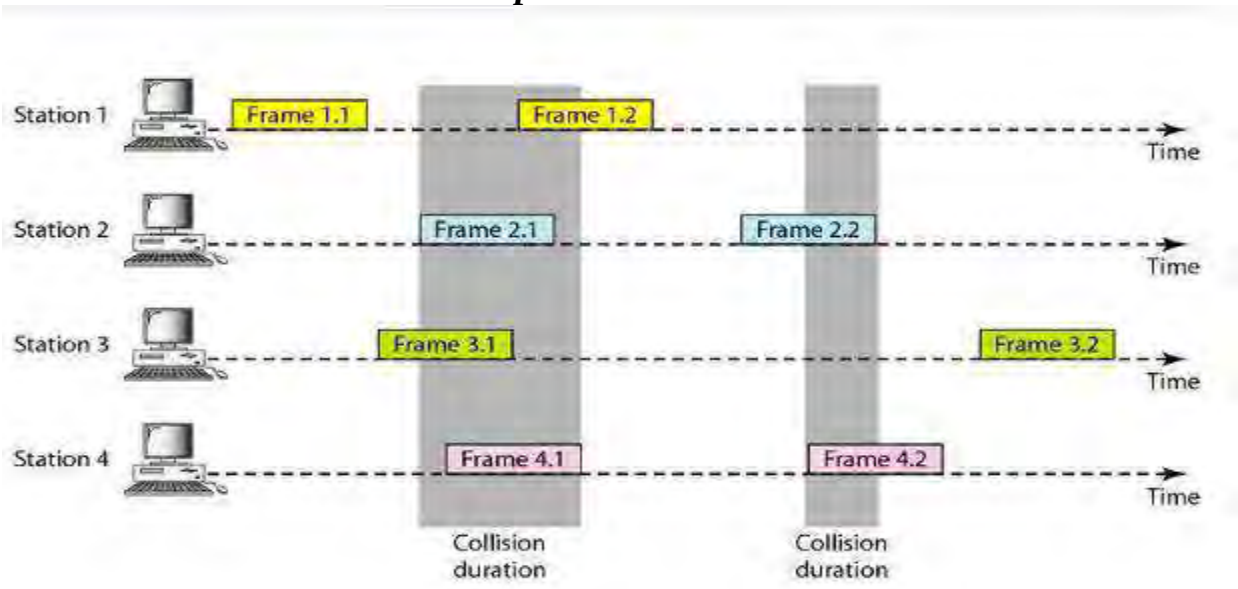
**ALOHA**




**Pure ALOHA**

It is the original ALOHA protocol. The idea of this protocol is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations.

***Frames in a pure ALOHA network***



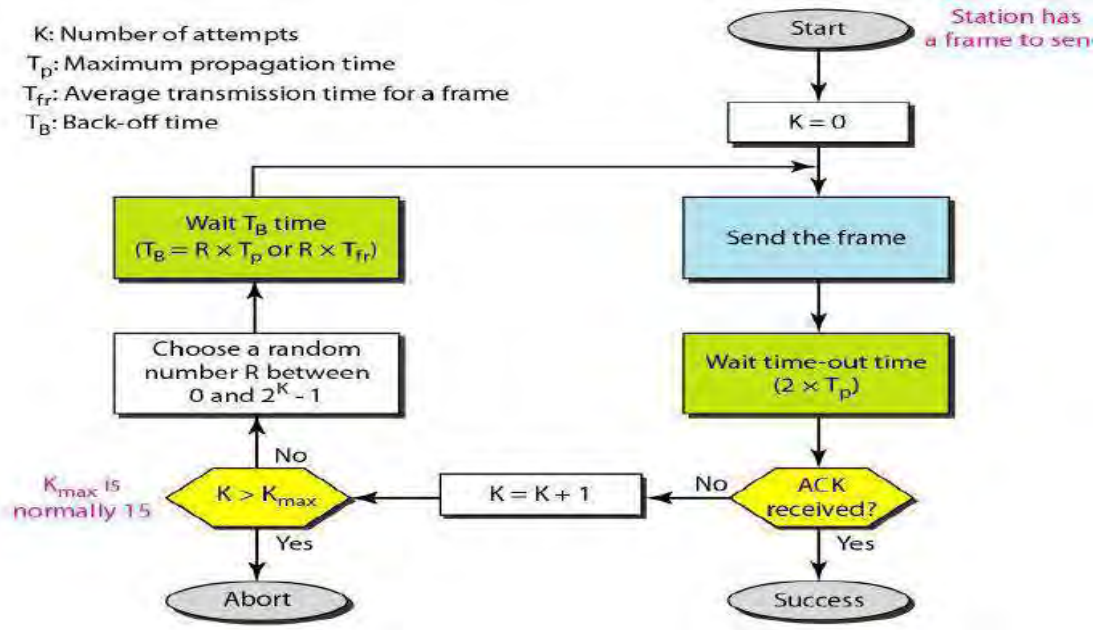
There are four stations (unrealistic assumption) that contend with one another for access to the shared channel. Each station sends two frames; there are a total of eight frames on the shared medium. Some of these frames collide because multiple frames are in contention for the shared channel.

In the above figure shows that only two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. We need to mention that even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.

The pure ALOHA protocol relies on acknowledgments from the receiver and it need to resend the frames that have been destroyed during transmission. When a station sends a frame and waits for certain time period for an acknowledgment. If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.

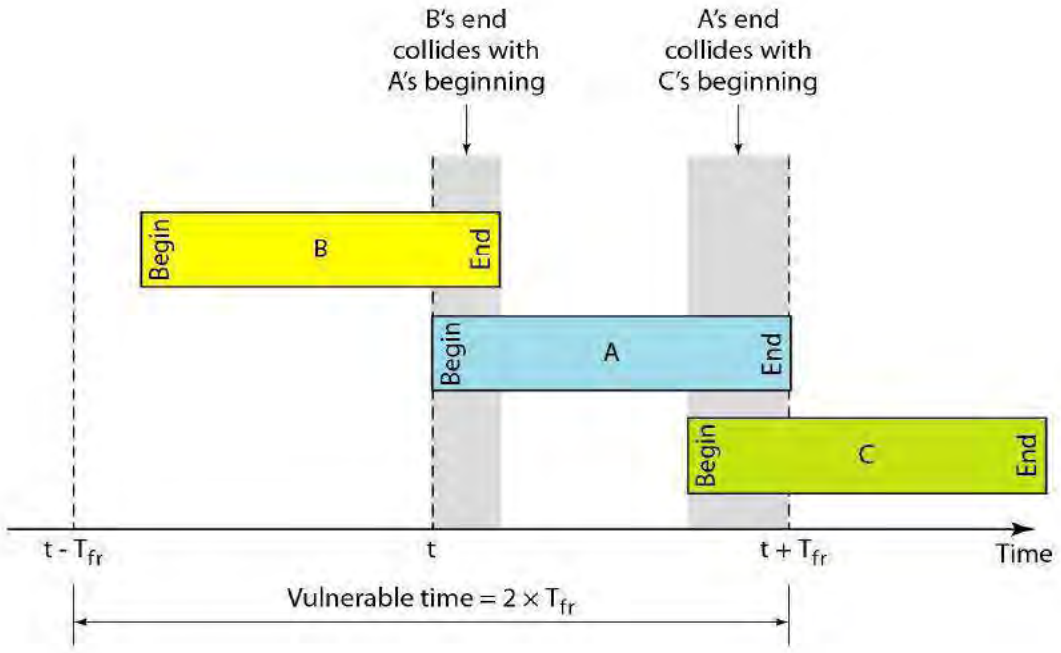
If all these stations try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each station waits a **random amount** of time before resending its frame. The randomness will help **avoid more collisions**. We call this time the **back-off time** $\boldsymbol{T_B}$**.**

Pure ALOHA has a second method to prevent congesting the channel with retransmitted frames. After a maximum number of retransmission attempts $K_{max}$' a station must give up and try later. Figure shows the procedure for pure ALOHA based on the above strategy.




The time-out period is equal to the maximum possible round-trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations **(2 x $T_p$)'** The **back-off time $T_B$** is a random value that normally depends on *K* (the number of attempted unsuccessful transmissions). The **$T_B$ (back-off time)** depends on the implementation. In one common formula **binary exponential back-off** for each retransmission, a multiplier in the range 0 to $2^K$ - 1 is randomly chosen and multiplied by **$T_p$ (maximum propagation time)** or **$T_{rr}$ (the average time required to send out a frame)** to find **$T_B$.** Note that in this procedure, the range of the random numbers increases after each collision. The value of **$K_{max}$** is usually chosen as 15.

***Vulnerable time for pure ALOHA protocol***




Pure ALOHA has a vulnerable time of 2 x $T_{fr}$. This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of $T_{fr}$ s and force the station to send only at the beginning of the time slot.

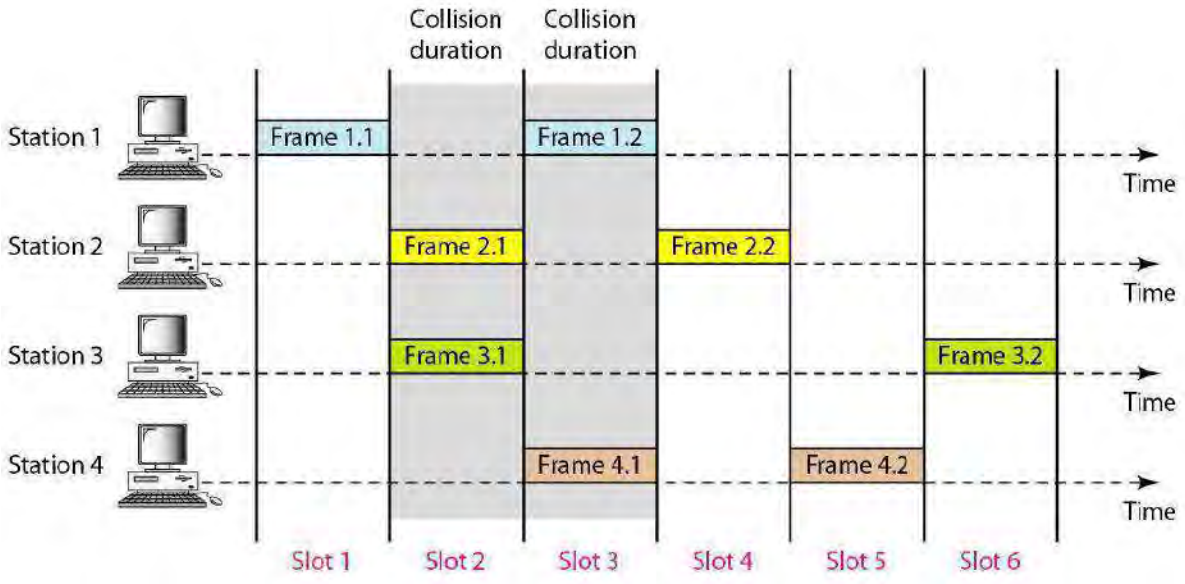
**Slotted ALOHA**

To minimize the number of collisions, there by optimizing network efficiency and increasing the number of subscribers that can use a given network; a scheme called ***slotted Aloha*** was developed. This system employs signals called beacons that are sent at precise intervals and tell each source when the channel is clear to send a frame. Further improvement can be realized by a more sophisticated protocol called Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

Pure ALOHA has a vulnerable time of 2 x $T_{fr}$ . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of $T_{fr}$ s and force the station to send only at the beginning of the time slot. Figure 12.6 shows an example of frame collisions in slotted ALOHA.

***Frames in a slotted ALOHA network***

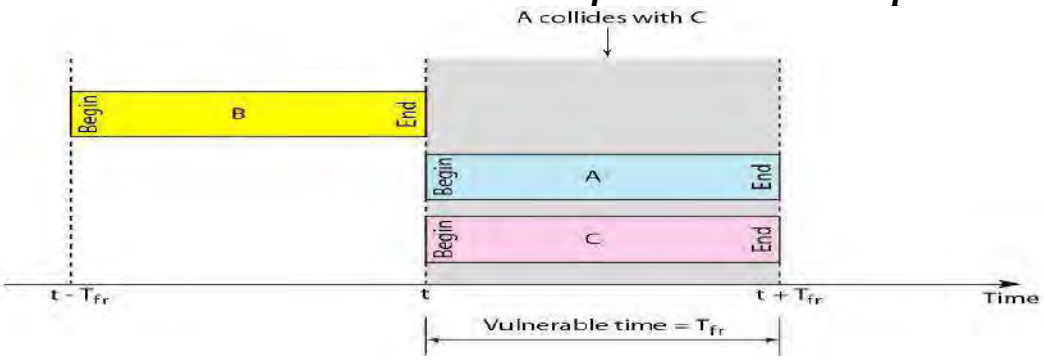



Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to $T_{fr}$ Figure 12.7 shows the situation. Figure 12.7 shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA.

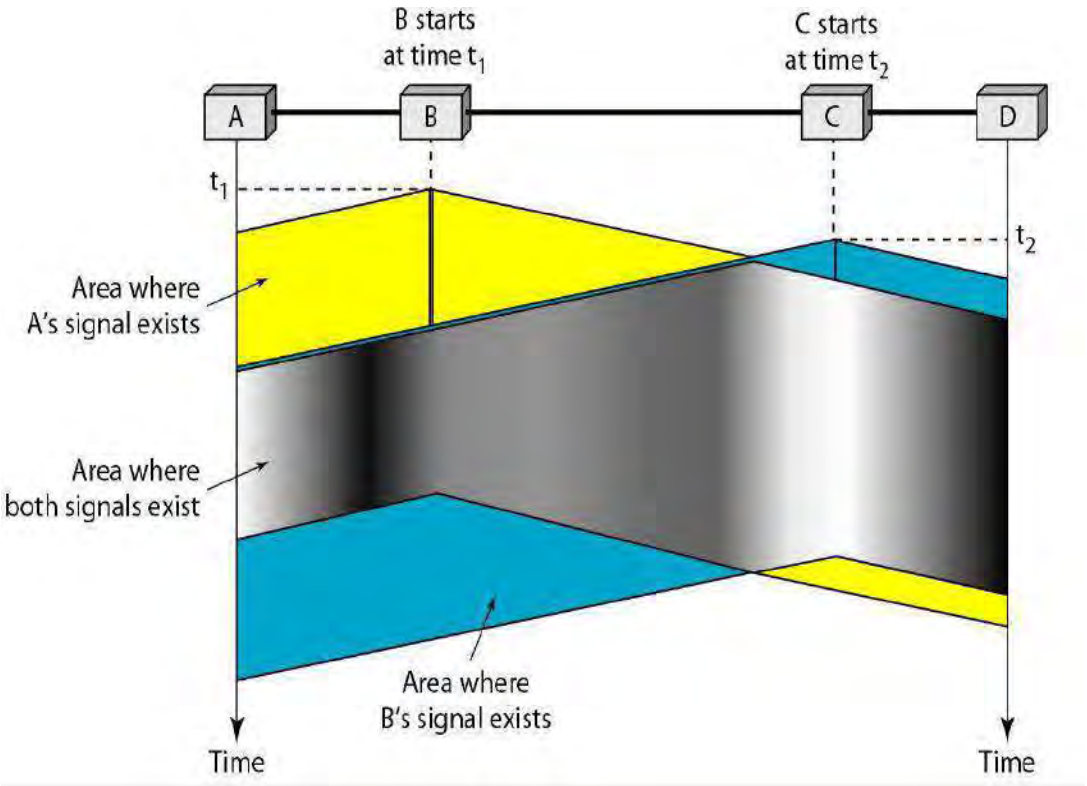
Slotted ALOHA vulnerable time = $T_{fr}$

Throughput It can be proved that the average number of successful transmissions for slotted ALOHA is $S = G \times e^{-G}$. The maximum throughput Smax is 0.368, when G = 1. In other words, if a frame is generated during one frame transmission time, then 36.8 percent of these frames reach their destination successfully. This result can be expected because the vulnerable time is equal to the frame transmission time. Therefore, if a station generates only one frame in this vulnerable time (and no other station generates a frame during this time), the frame will reach its destination successfully.

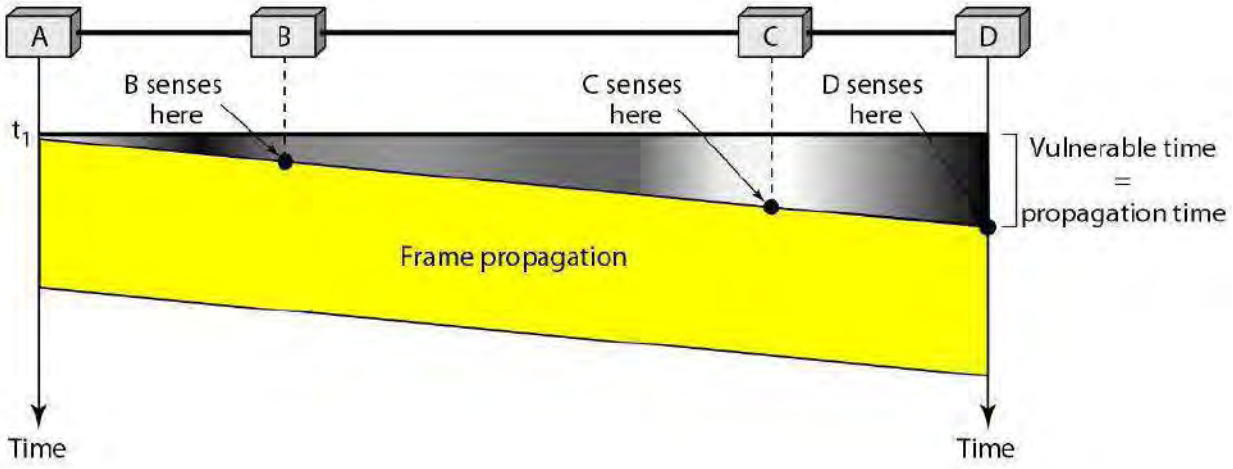
***Vulnerable time for slotted ALOHA protocol***



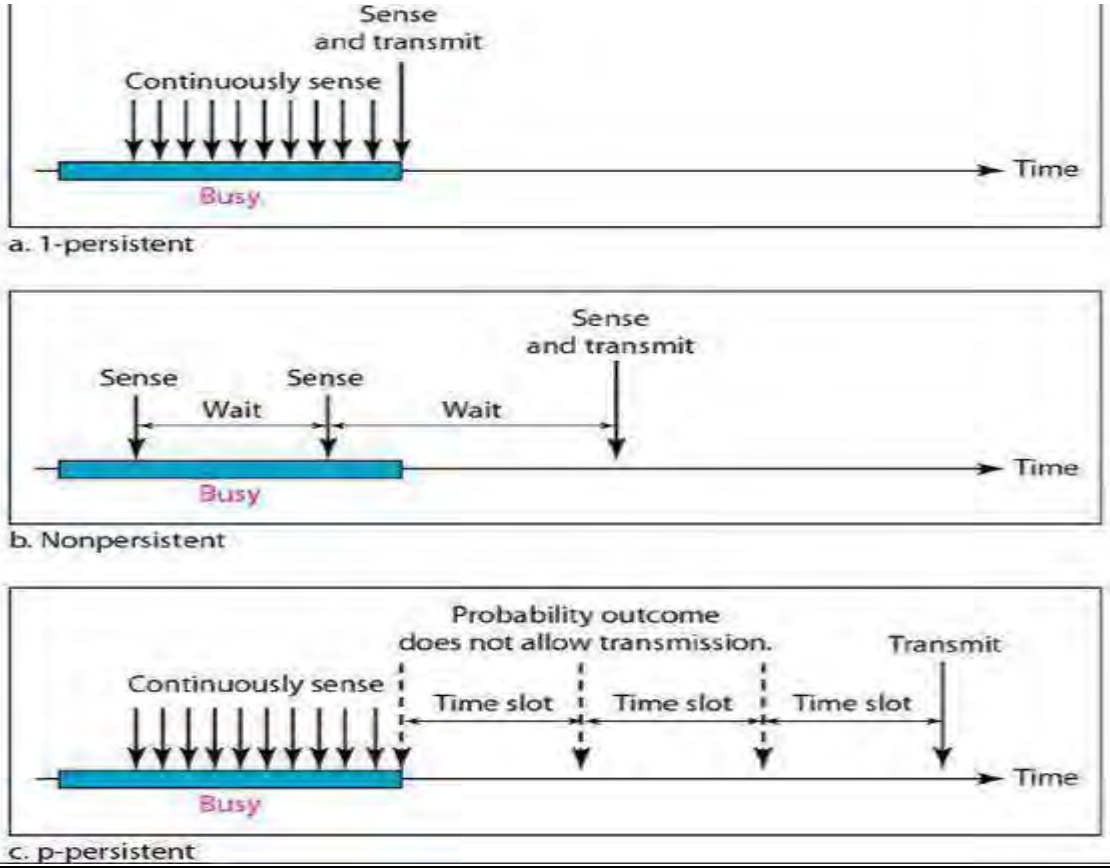

***Space/time model of the collision in CSMA***



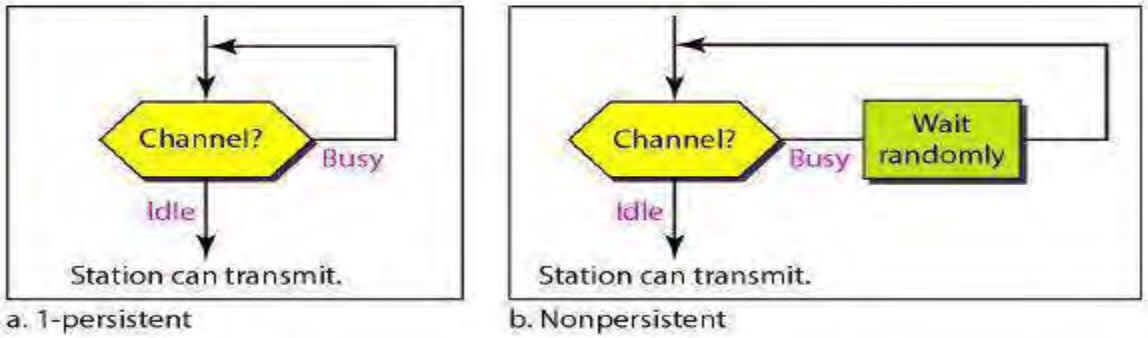
***Vulnerable time in CSMA***

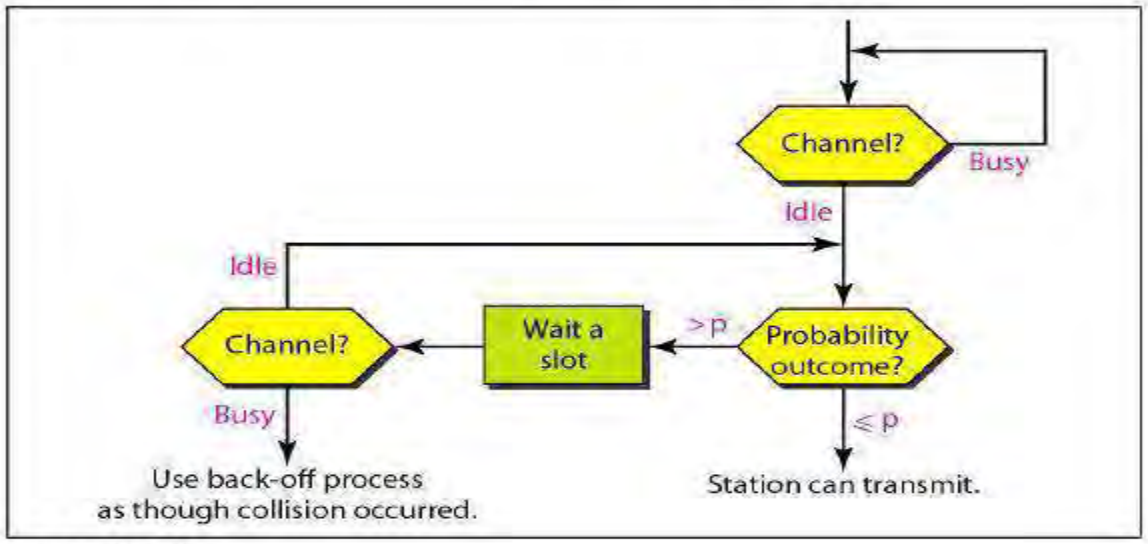



***Behaviour of three persistence methods***

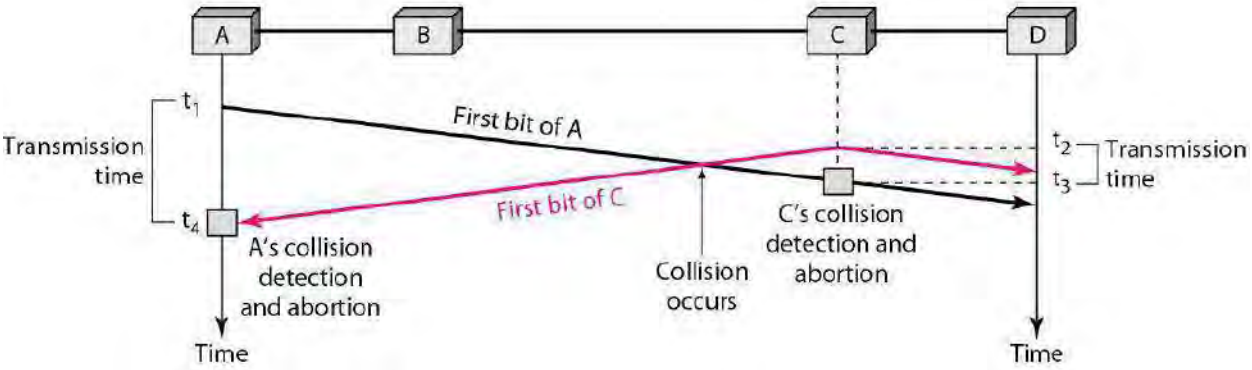


***Flow diagram for three persistence methods***

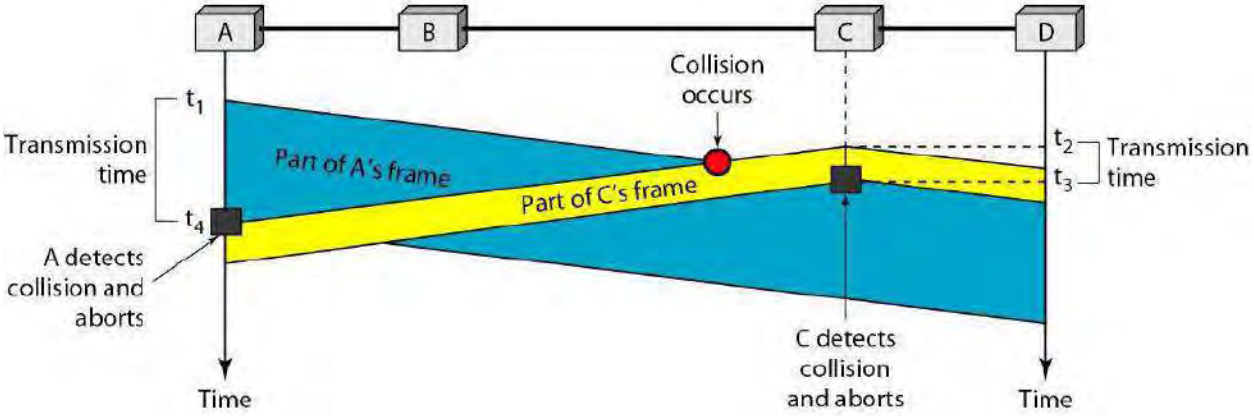


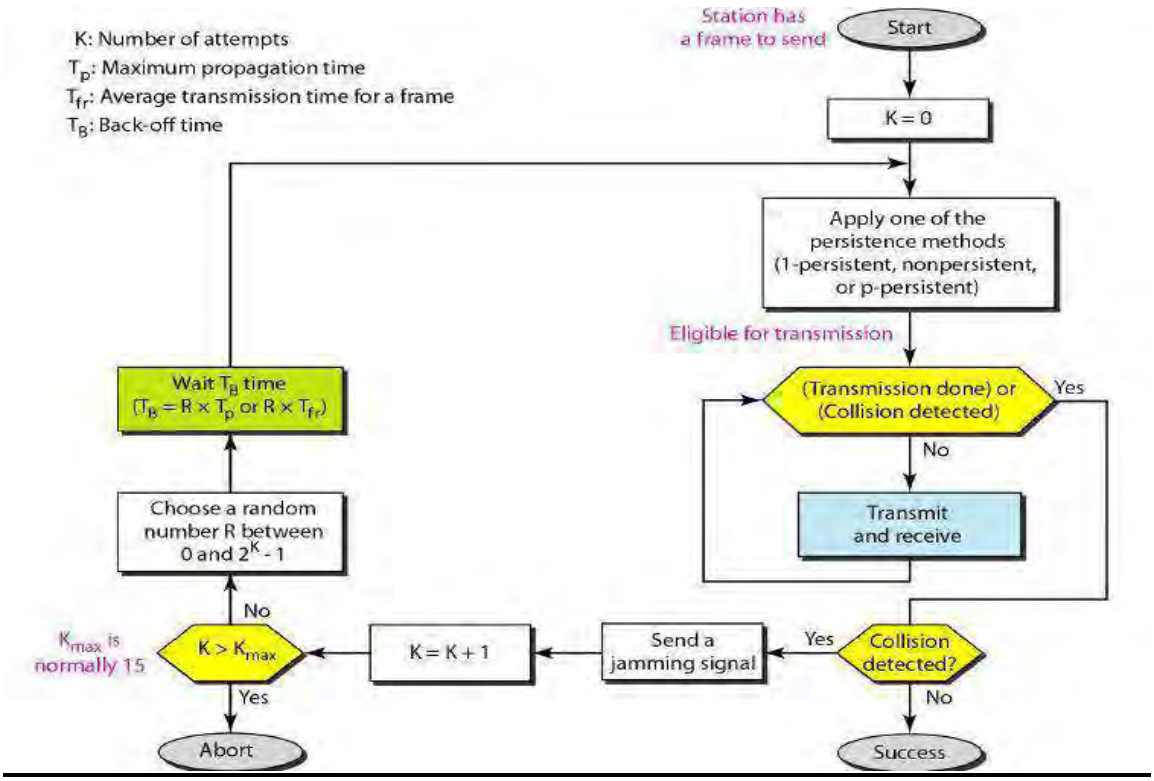

***Collision of the first bit in CSMA/CD***



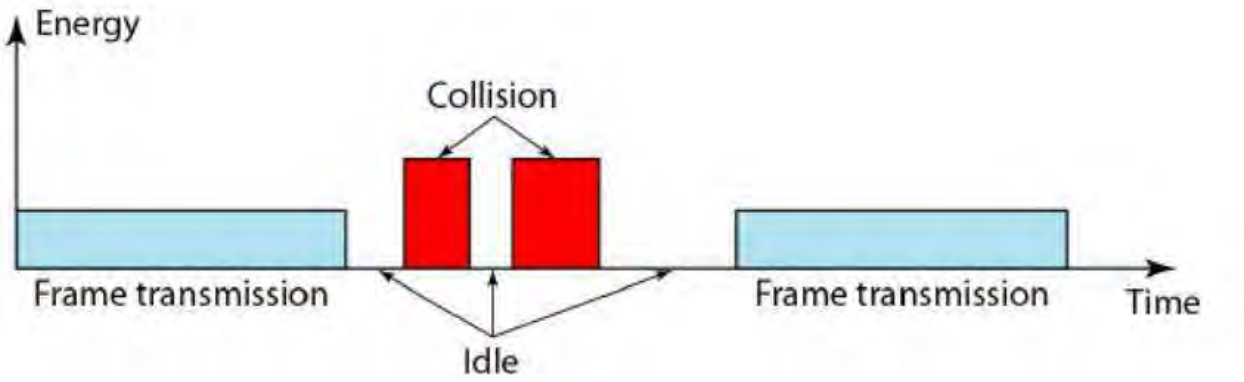
## *Collision and abortion in CSMA/CD*



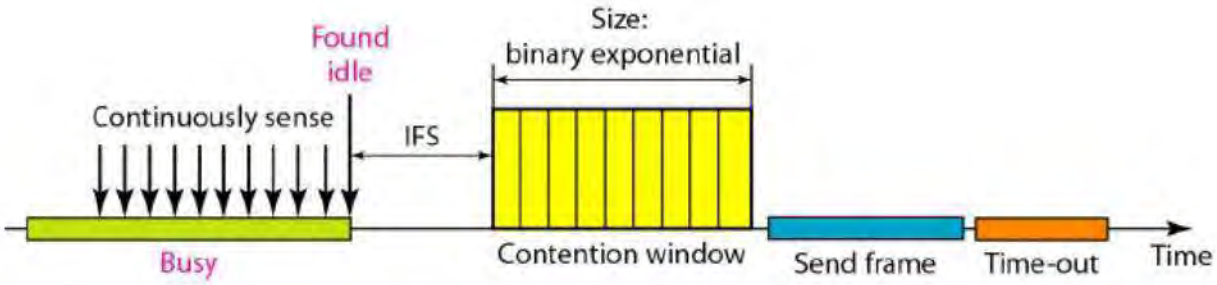

## *Flow diagram for the CSM/CD*



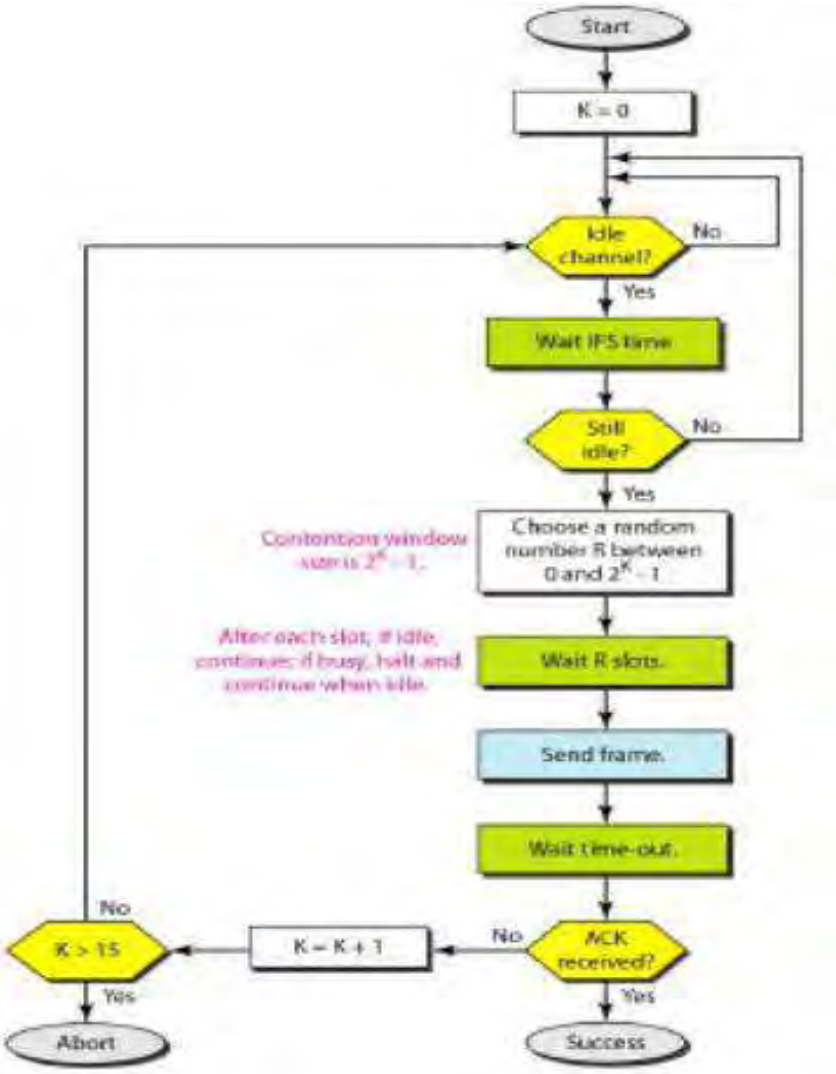
***Energy level during transmission, idleness, or collision***




***Timing in CSMA/CA***



***Flow diagram for CSM/CA***



*LECTURE NOTE: 15*

# CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We discuss three popular controlled-access methods.

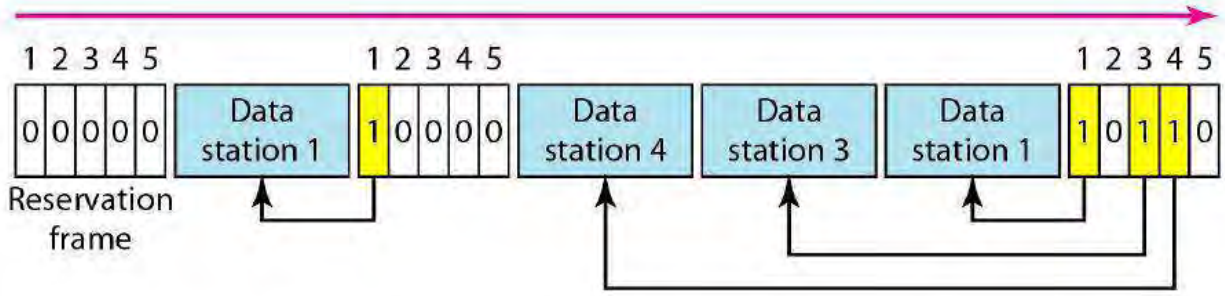
## RESERVATION

In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

If there are *N* stations in the system, there are exactly *N* reservation mini slots in the reservation frame. Each mini slot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own mini slot. The stations that have made reservations can send their data frames after the reservation frame.

Figure 12.18 shows a situation with five stations and a five-minislot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.

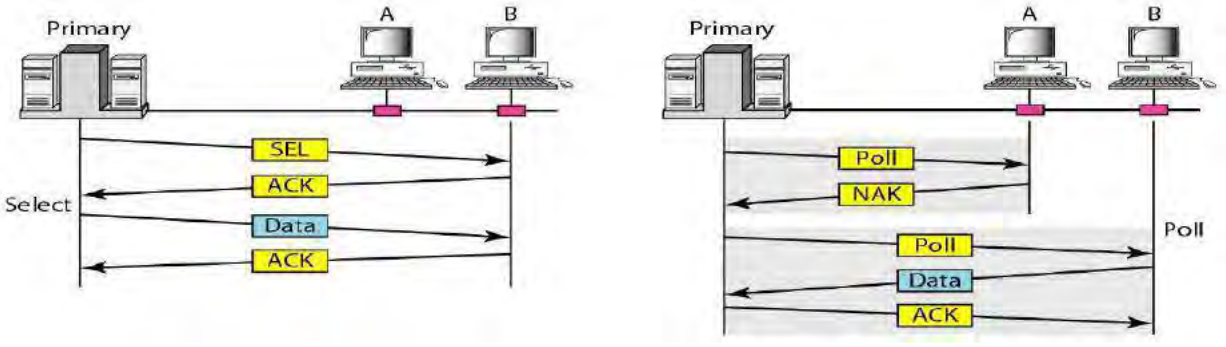
***Reservation access method***




## POLLING

Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data 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 device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session.

***Select and poll functions in polling access method***




If the primary wants to receive data, it asks the secondary if they have anything to send; this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

**SELECT**

The *select* function is used whenever the primary device has something to send. Remember that the primary controls the link. If the primary is neither sending nor receiving data, it knows the link is available. If it has something to send, the primary device 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.

**POLL**

The *poll* function is used by the primary device to solicit transmissions from the secondary devices. 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), then the primary polls the next secondary in the same manner 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.

**TOKEN PASSING**

In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a *predecessor* and a *successor.* The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current

station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send.

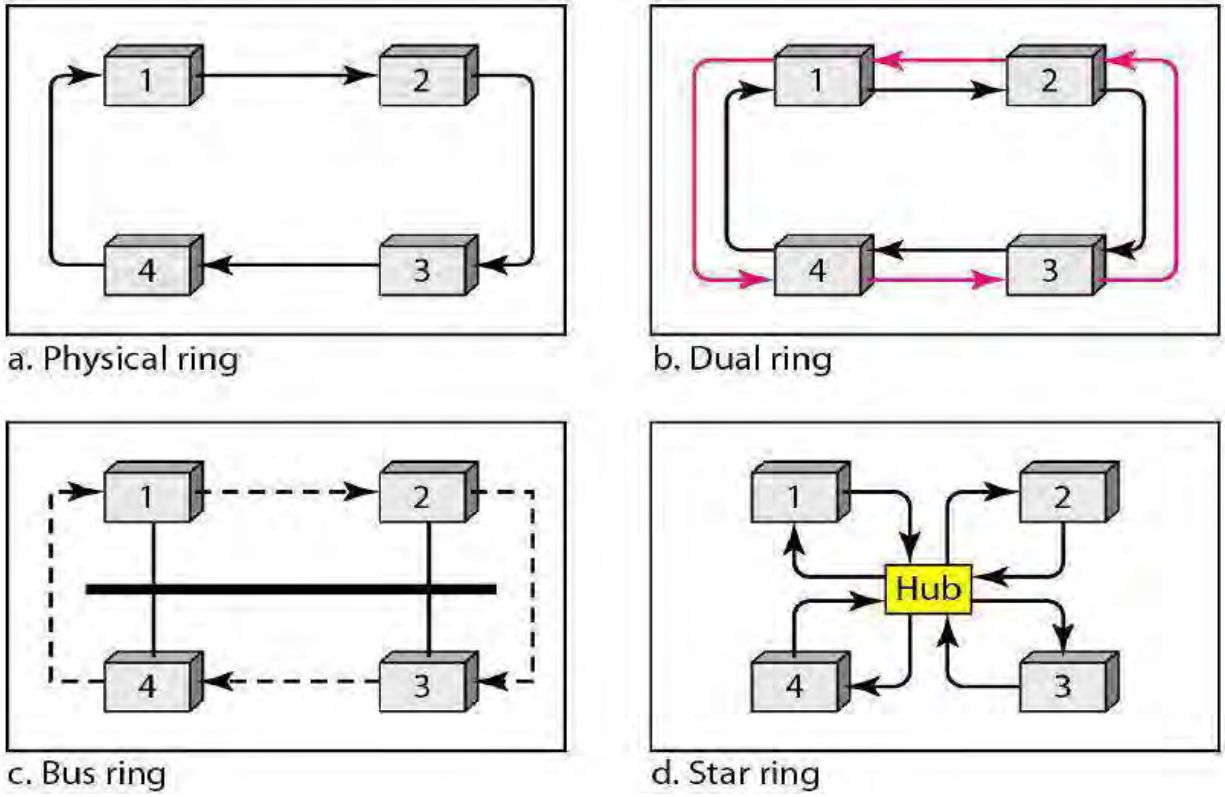
But how is the right to access the channel passed from one station to another? In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data. When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data. When the station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round. In this process, when a station receives the token and has no data to send, it just passes the data to the next station.

Token management is needed for this access method. Stations must be limited in the time they can have possession of the token. The token must be monitored to ensure it has not been lost or destroyed. For example, if a station that is holding the token fails, the token will disappear from the network. Another function of token management is to assign priorities to the stations and to the types of data being transmitted. And finally, token management is needed to make low-priority stations release the token to high priority stations.

**LOGICAL RING**

In a token-passing network, stations do not have to be physically connected in a ring; the ring can be a logical one. In the physical ring topology, when a station sends the token to its successor, the token cannot be seen by other stations; the successor is the next one in line. This means that the token does not have to have the address of the next successor. The problem with this topology is that if one of the links-the medium between two adjacent stations fails, the whole system fails.

***Logical ring and physical topology in token-passing access method***




The dual ring topology uses a second (auxiliary) ring which operates in the reverse direction compared with the main ring. The second ring is for emergencies only (such as a spare tire for a car). If one of the links in the main ring fails, the system automatically combines the two rings to form a temporary ring. After the failed link is restored, the auxiliary ring becomes idle again. Note that for this topology to work, each station needs to have two transmitter ports and two receiver ports. The high-speed Token Ring networks called FDDI (Fiber Distributed Data Interface) and CDDI (Copper Distributed Data Interface) use this topology.

In a star-ring topology, the physical topology is a star. There is a hub, however, that acts as the connector. The wiring inside the hub makes the ring; the stations are connected to this ring through the two wire connections. This topology makes the network less prone to failure because if a link goes down, it will be bypassed by the hub and the rest of the stations can operate. Also adding and removing stations from the ring is easier. This topology is still used in the Token Ring LAN designed by IBM.

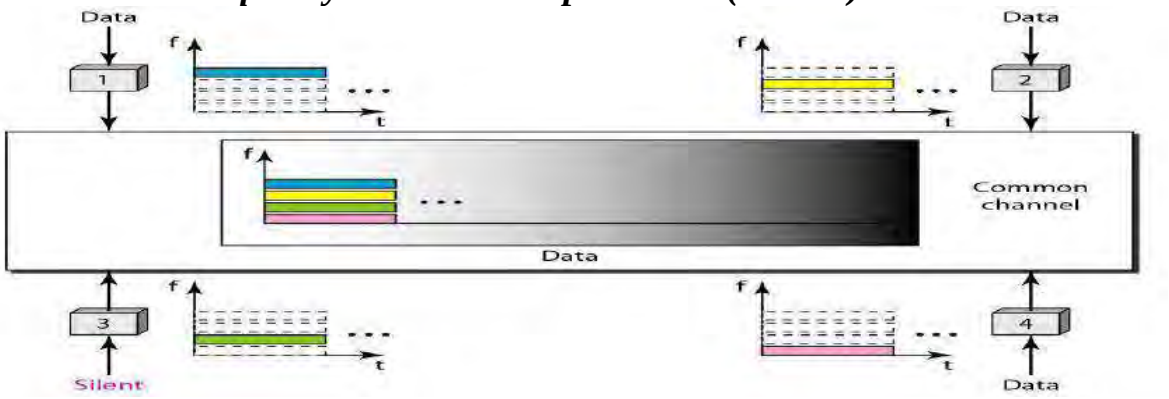
## CHANNELIZATION

Channelization is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. In this section, we discuss three channelization protocols: FDMA, TDMA, and CDMA.

## FREQUENCY-DIVISION MULTIPLE ACCESS (FDMA)

In frequency-division multiple access (FDMA), the available bandwidth is divided into frequency bands. Each station is allocated a band to send its data. In other words, each band is reserved for a specific station, and it belongs to the station all the time. Each station also uses a band pass filter to confine the transmitter frequencies. To prevent station interferences, the allocated bands are separated from one another by small *guard bands.* Figure shows the idea of FDMA.

***Frequency-division multiple access (FDMA)***



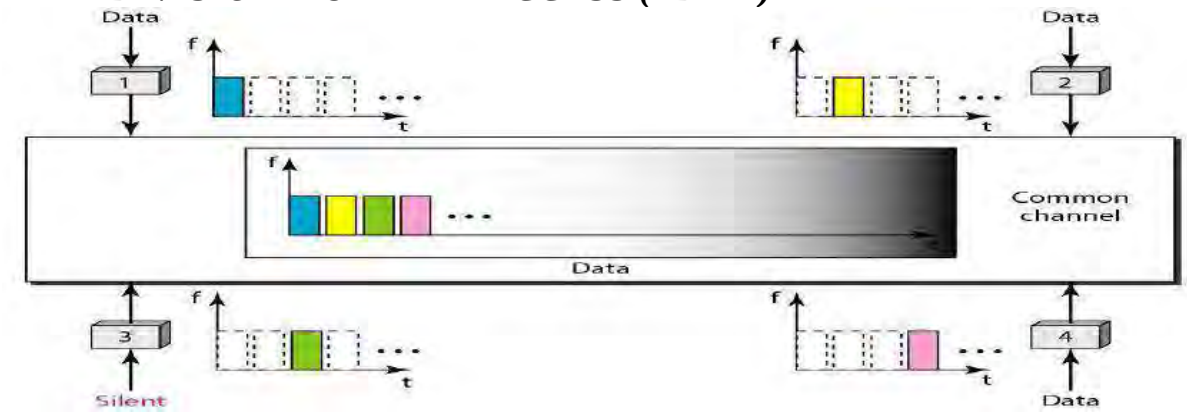

FDMA specifies a predetermined frequency band for the entire period of communication. This means that stream data (a continuous flow of data that may not be packetized) can easily be used with FDMA.

FDMA, is an access method in the data link layer. The data link layer in each station tells its physical layer to make a band pass signal from the data passed to it. The signal must be created in the allocated band. There is no physical multiplexer at the physical layer. The signals created at each station are automatically band pass-filtered. They are mixed when they are sent to the common channel.

## TIME-DIVISION MULTIPLE ACCESS (TDMA)

In time-division multiple access (TDMA), the stations share the bandwidth of the channel in time. Each station is allocated a time slot during which it can send data. Each station transmits its data in is assigned time slot. Below figure shows the idea behind TDMA.

***TIME-DIVISION MULTIPLE ACCESS (TDMA)***




The main problem with TDMA lies in achieving synchronization between the different stations. Each station needs to know the beginning of its slot and the location of its slot. This may be difficult because of propagation delays introduced in the system if the stations are spread over a large area. To compensate for the delays, we can insert ***guard times***. Synchronization is normally accomplished by having some synchronization bits (preamble bits) at the beginning of each slot.

TDMA, is an access method in the data link layer. The data link layer in each station tells its physical layer to use the allocated time slot. There is no physical multiplexer at the physical layer.

## CODE-DIVISION MULTIPLE ACCESS (CDMA)

Code-division multiple access (CDMA) was conceived several decades ago. Recent advances in electronic technology have finally made its implementation possible. CDMA differs from FDMA because only one channel occupies the entire

bandwidth of the link. It differs from TDMA because all stations can send data simultaneously; there is no timesharing.

***ANALOGY***

Let us first give an analogy. CDMA simply means communication with different codes. For example, in a large room with many people, two people can talk in English if nobody else understands English. Another two people can talk in Chinese if they are the only ones who understand Chinese, and so on. In other words, the common channel, the space of the room in this case, can easily allow communication between several couples, but in different languages (codes).

***IDEA***

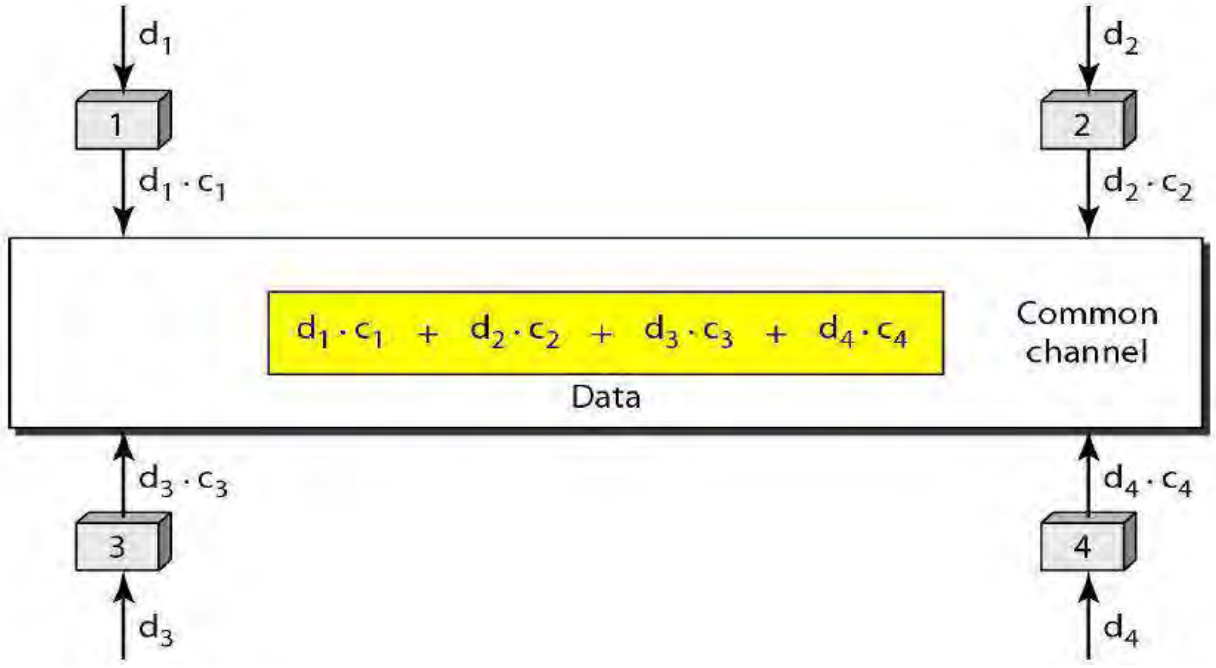
Let us assume we have four stations 1, 2, 3, and 4 connected to the same channel. The data from station 1 are $d_1$ , from station 2 are *d2*, and so on. The code assigned to the first station is c1, to the second is *c2*, and so on. We assume that the assigned codes have two properties.

1. If we multiply each code by another, we get 0.
2. If we multiply each code by itself, we get 4 (the number of stations).

With these two properties in mind, let us see how the above four stations can send data using the same common channel, as shown in the below figure.

Station 1 multiplies (a special kind of multiplication, as we will see) its data by its code to get dl.cl' Station 2 multiplies its data by its code to get *d2 . c2* and so on.

***Simple idea of communication with code***



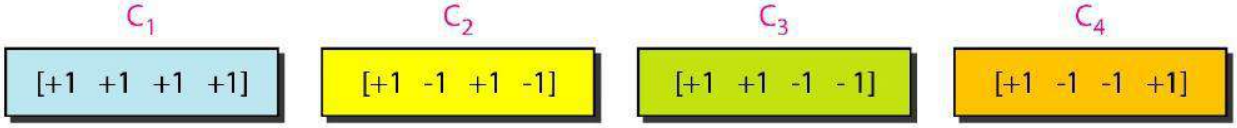
The data that go on the channel are the sum of all these terms, as shown in the box. Any station that wants to receive data from one of the other three multiplies the data on the channel by the code of the sender. For example, suppose stations 1 and 2 are talking to each other. Station 2 wants to hear what station I is saying. It multiplies the data on the channel by cl' the code of station 1.
Because (cl . cl) is 4, but *(c2* . c1), *(c3* . c1), and *(c4* . cl) are all 0s, station 2 divides the result by 4 to get the data from station 1.

data =*(d1* . c1 + dz . cz +*d3* . *c3* + *d4* . *c4)* . *c1*
=*d 1*. c1. c1 + d2. c2 . cl + *d3* . *c3* . cl + *d4* . *c4*.c1 =4.*d1*

***CHIPS***

CDMA is based on coding theory. Each station is assigned a code, which is a sequence of numbers called chips, as shown in figure below, the codes are for the previous example.

***Chip sequences***




The sequences were not choosing randomly; they were carefully selected. They are called orthogonal sequences and have the following properties:

1. Each sequence is made of *N* elements, where *N* is the number of stations.
2. If we multiply a sequence by a number, every element in the sequence is multiplied by that element. This is called multiplication of a sequence by a scalar. For example,

2. [+1+1-1-1]= [+2+2-2-2]

3. If we multiply two equal sequences, element by element, and add the results, we get *N*, where *N* is the number of elements in the each sequence. This is called the inner product of two equal sequences. For example, [+1 +1-1-1]. [+1 +1 -1 -1] = 1 + 1 + 1 + 1 = 4

4. If we multiply two different sequences, element by element, and add the results, we get 0. This is called inner product of two different sequences. For example,

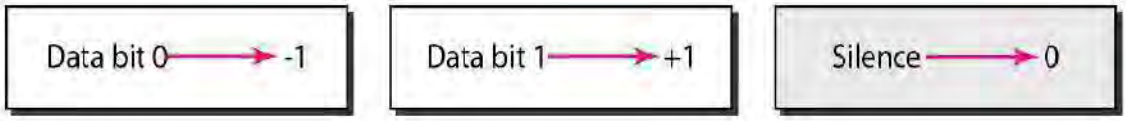
[+1 +1 -1 -1] . [+1 +1 +1 +1] = 1 + 1 - 1 - 1 = 0

5. Adding two sequences means adding the corresponding elements. The result is another sequence. For example, [+1+1-1-1]+[+1+1+1+1]=[+2+2 0 0]

### *DATA REPRESENTATION*

We follow these rules for encoding: If a station needs to send a 0 bit, it encodes it as -1; if it needs to send a 1 bit, it encodes it as +1. When a station is idle, it sends no signal, which is interpreted as a 0. These are shown in Figure

***Data representation in CDMA***



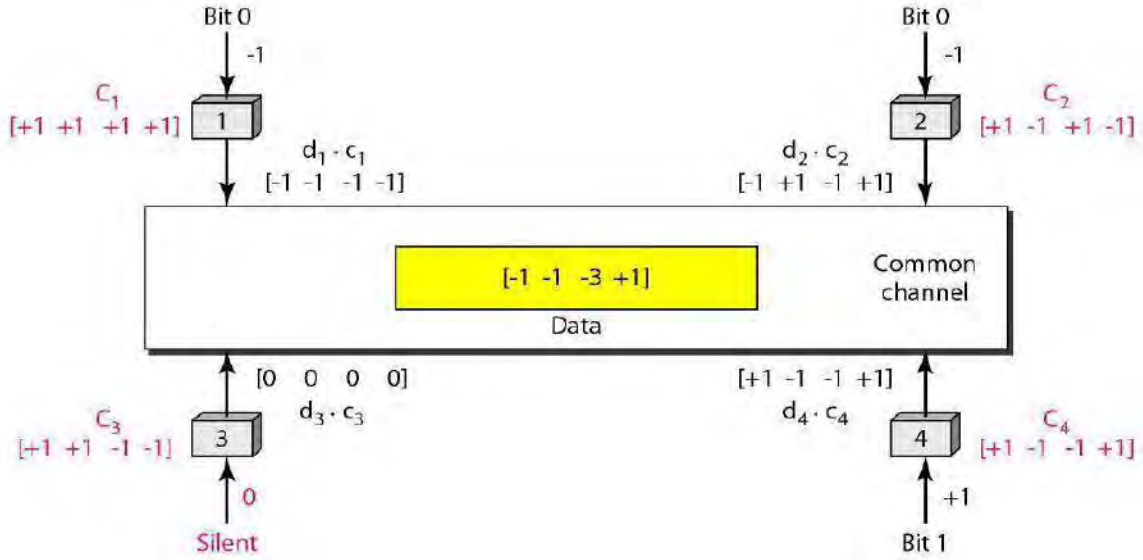

## *ENCODING AND DECODING*

As a simple example, we show how four stations share the link during a 1-bit interval. The procedure can easily be repeated for additional intervals. We assume that stations 1 and 2 are sending a 0 bit and channel 4 is sending a 1 bit. Station 3 is silent. The data at the sender site are translated to -1, -1, 0, and +1. Each station multiplies the corresponding number by its chip (its orthogonal sequence), which is unique for each station. The result is a new sequence which is sent to the channel. For simplicity, we assume that all stations send the resulting sequences at the same time. The sequence on the channel is the sum of all four sequences as defined before. Figure below shows the situation.

Now imagine station 3, which we said is silent, is listening to station 2. Station 3 multiplies the total data on the channel by the code for station 2, which is [+1 -1 +1-1], to get

[-1-1-3 +1]· [+1-1 +1-1] =-4/4 =-1 ..... bit 1

***Sharing channel in CDMA***




## *SIGNAL LEVEL*

The process can be better understood if we show the digital signal produced by each station and the data recovered at the destination (see Figure below). The figure shows the corresponding signals for each station (using NRZ-L for simplicity) and the signal that is on the common channel.

***Digital signal created by four stations in CDMA***

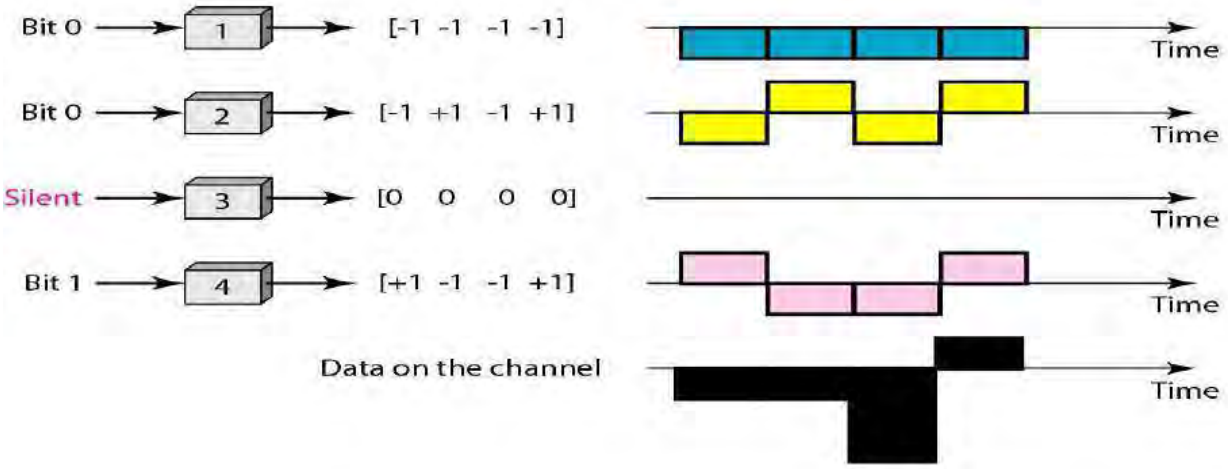
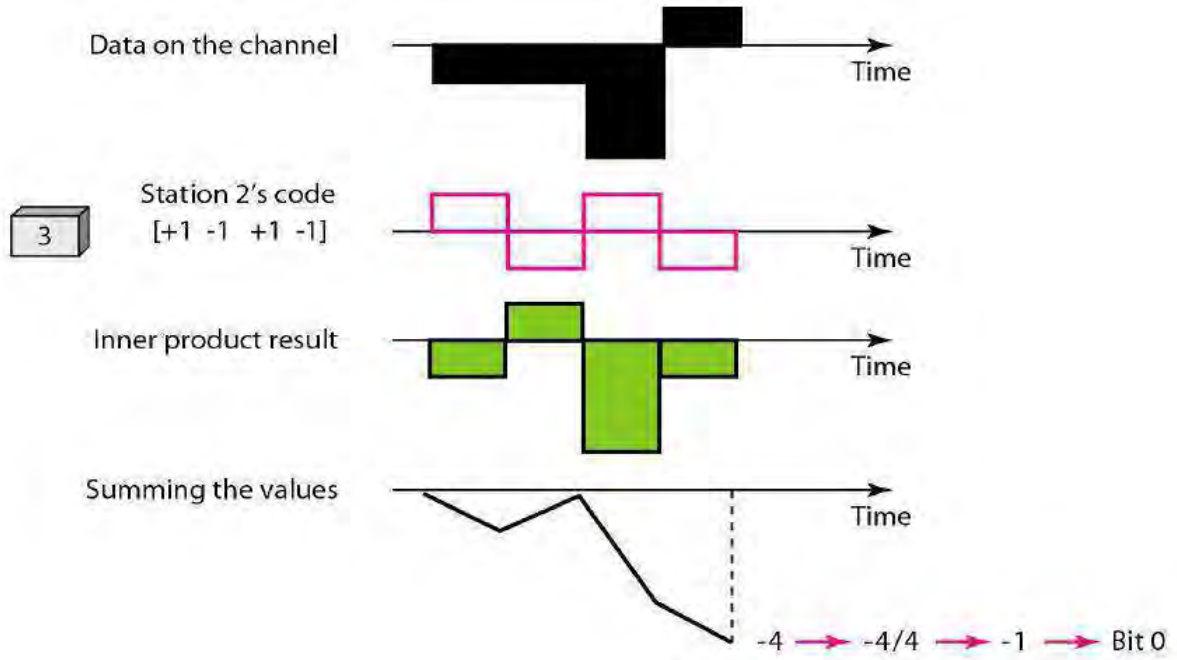


Figure below shows how station 3 can detect the data sent by station 2 by using the code for station 2. The total data on the channel are multiplied (inner product operation) by the signal representing station 2 chip code to get a new signal. The station then integrates and adds the area under the signal, to get the value -4, which is divided by 4 and interpreted as bit O.

***Figure: Decoding of the composite signal for one in CDMA***

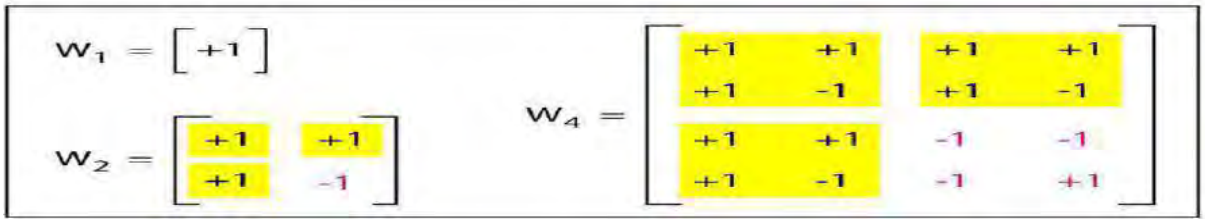



***SEQUENCE GENERATION***

To generate chip sequences, we use a **Walsh table,** which is a two-dimensional table with an equal number of rows and columns, as shown in Figure as

Figure: ***General rule and examples of creating Walsh tables***


a. Two basic rules



b. Generation of w1,w2 and w4

In the Walsh table, each row is a sequence of chips. *W1* for a one-chip sequence has one row and one column. We can choose –l or +1 for the chip for this trivial table (we chose +1). According to Walsh, if we know the table for *N* sequences *WN'* we can create the table for *2N* sequences $W_{2N}$, as shown in Figure(decoding of signal in CDMA) previous figure. The $W_N$ with the over bar $W_N$ stands for the complement of $W_{N'}$ where each +1 is changed to -1 and vice versa.

Figure walsh table also shows how we can create *W2* and *W4* from *W1'* After we select *W1*, *W2* can be made from four *Wj* 's, with the last one the complement of *W1'* After *W2* is generated, *W4* can be made of four *W2's*, with the last one the complement of *W2*. Of course, Ws is composed of four *W4's*, and so on. **Note** that after $W_N$ is made, each station is assigned a chip corresponding to a row. Something we need to emphasize is that the number of sequences *N* needs to be a power of 2. In other words, we need to have $N = 2^m$.

---

The number of sequences in a Walsh table needs to be$N = 2^m$.

---

**Wired LANs: Ethernet**

The LAN market has seen several technologies such as Ethernet, Token Ring, Token Bus, FDDI, and ATM LAN. Some of these technologies survived for a while, but Ethernet is by far the dominant technology. The IEEE Standard Project 802, designed to regulate the manufacturing and interconnectivity between different LANs.

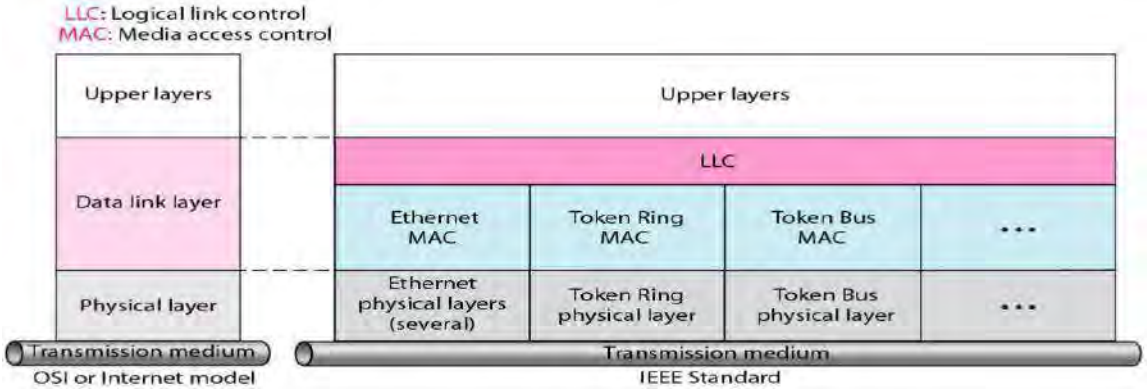
**IEEE STANDARDS**

In 1985, the Computer Society of the IEEE started a project, called Project 802, to set standards to enable intercommunication among equipment from a variety of manufacturers. Project 802 is a way of specifying functions of the physical layer and the data link layer of major LAN protocols.

The relationship of the 802 Standard to the traditional OSI model is shown in the below **Figure** .The IEEE has subdivided the data link layer into two sub layers:

logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.

IEEE standard for LAN



***LECTURE NOTE: 16***

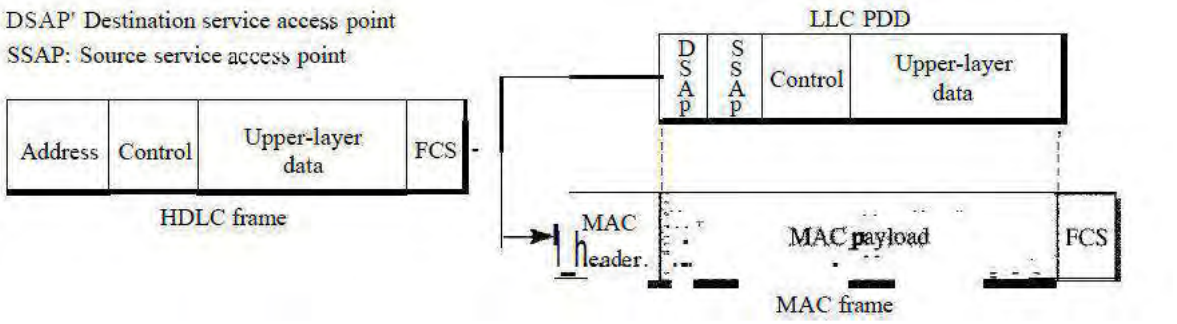
# DATA LINK LAYER

The data link layer in the IEEE standard is divided into two sub layers: **LLC and MAC**.

## LOGICAL LINK CONTROL (LLC)

The data link control handles framing, flow control, and error control. In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sub layer called the logical link control. Framing is handled in both the LLC sub layer and the MAC sub layer.

The LLC provides one single data link control protocol for all IEEE LANs. The LLC is different from the media access control sub layer, which provides different protocols for different LANs. A single LLC protocol can provide interconnectivity between different LANs because it makes the MAC sub layer transparent.




## HDLC FRAME COMPARED WITH LLC AND MAC FRAMES

The purpose of the LLC is to provide flow and error control for the upper-layer protocols that actually demand these services. For example, if a LAN or several LANs are used in an isolated system, LLC may be needed to provide flow and error control for the application layer protocols. However, most upper-layer protocols such as IP, do not use the services of LLC.

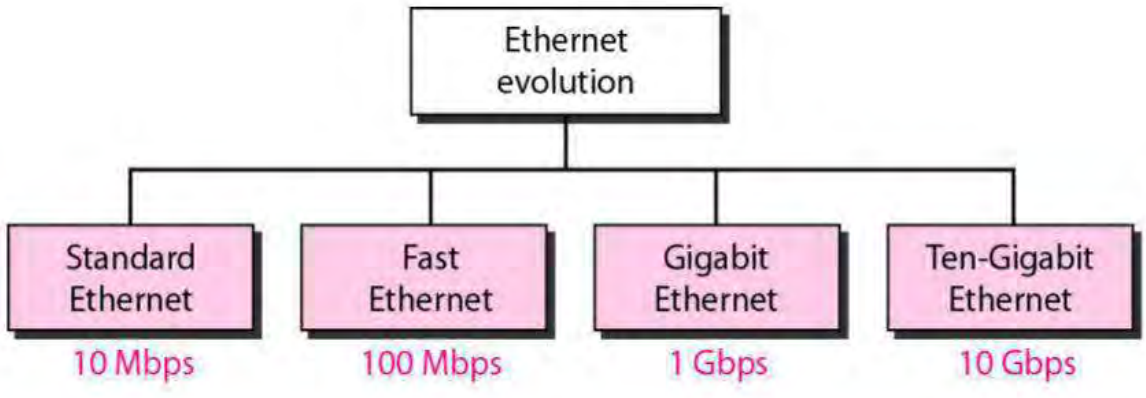
## MEDIA ACCESS CONTROL (MAC)

It defines the specific access method for each LAN. For example, it defines *CSMA/CD* as the media access method for Ethernet LANs and the token passing method for Token Ring and Token Bus LANs. Part of the framing function is also handled by the MAC layer. In contrast to the LLC sub-layer, the MAC sub-layer contains a number of distinct modules; each defines the access method and the framing format specific to the corresponding LAN protocol.

## PHYSICAL LAYER

The physical layer is dependent on the implementation and type of physical media used. IEEE defines detailed specifications for each LAN implementation. For example, although there is only one MAC sub layer for Standard Ethernet, there is a different physical layer specification for each Ethernet implementations.

The original Ethernet was created in 1976 at Xerox's Palo Alto Research Centre (PARC).
Since then, it has gone through four generations: traditional or Standard Ethernet (10 Mbps), Fast
Ethernet (100 Mbps), Gigabit Ethernet (l Gbps), and Ten-Gigabit Ethernet (l0 Gbps),
as shown in **Figure**:



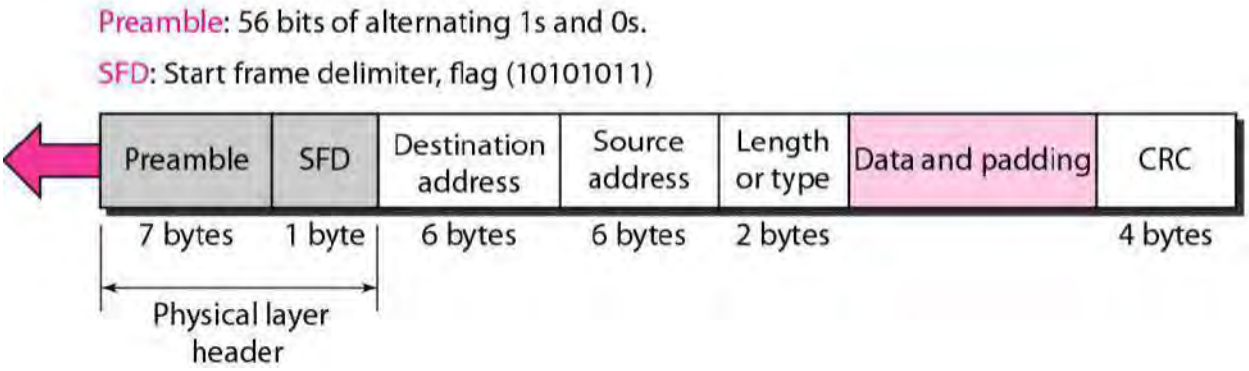

(*Ethernet evolution through four generations*)

## TRADITIONAL (STANDARD) ETHERNET

It is the first generation Ethernet of 10 Mbps. In Standard Ethernet, the MAC sub layer governs the operation of the access method. It also frames data received from the upper layer and passes them to the physical layer.

### Frame Format

The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of protocol data unit (PDU), upper-layer data, 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 is shown in below




(802.3 Media Access Control Frames)

**Preamble:** it is the first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to miss some bits at the beginning of the frame. The preamble is actually added at the physical layer and is not (formally) part of the frame.

**Start frame delimiter (SFD):** the second field (l byte: 10101011) signals the beginning of the frame. The SFD warns the station or stations that this is the last chance for synchronization. The last 2 bits is 11 and alerts the receiver that the next field is the destination address.

**Destination address (DA):** the DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet.

**Source address (SA):** the SA field is also 6 bytes and contains the physical address of the sender of the packet.

**Length or type:** this field is defined as a type field or length field. The original Ethernet used this field as the type field to define the upper-layer protocol using the MAC frame. The IEEE standard used it as the length field to define the number of bytes in the data field.

**Data:** This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes, as we will see later.

**CRC:** The last field contains error detection information, in this case a CRC-32.

***LECTURE NOTE: 17***

# FRAME LENGTH

Ethernet has imposed restrictions on both the minimum and maximum lengths of a frame, as shown in below

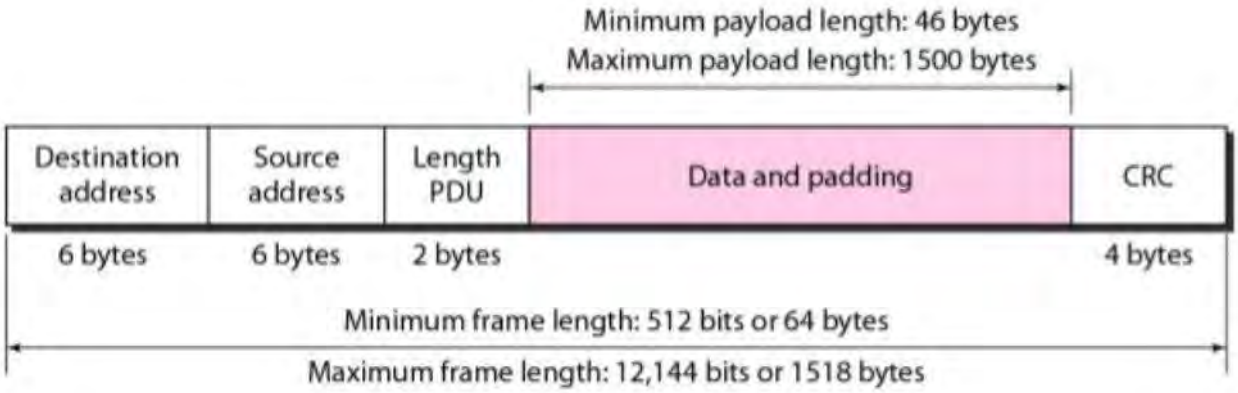



figure: *Minimum and maximum lengths*

The minimum length restriction is required for the correct operation of *CSMA/CD*. An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. If we count 18 bytes of header and trailer (6 bytes of source address, 6 bytes of destination address, 2 bytes of length or type, and 4 bytes of CRC), then the minimum length of data from the upper layer is 64 - 18 = 46 bytes. If the upper-layer packet is less than 46 bytes, padding is added to make up the difference. The standard defines the maximum length of a frame as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes. The maximum length restriction has two reasons. First, memory was very expensive when Ethernet was designed: a maximum length restriction helped to reduce the size of the buffer. Second, the maximum length restriction prevents one station from monopolizing the shared medium, blocking other stations that have data to send.

Note:

**Frame length:**
**Minimum: 64 bytes (512 bits)**
**Maximum: 1518 bytes (12,144 bits)**

## ADDRESSING

Each station on an Ethernet network (such as a PC, workstation, or printer) has its own network interface card (NIC). The NIC fits inside the station provides the station with a 6-byte physical address. The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.

**Example of an Ethernet address in hexadecimal notation:**


**Unicast, Multicast, and Broadcast Addresses**: A source address is always a unicast address-the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast. A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one. A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many. The broadcast address is a special case of the multicast address; the recipients are all the stations on the LAN. A broadcast destination address is forty-eight (11111111 11111111 11111111 11111111 11111111 11111111) 1s.

**Figure: Unicast and multicast addresses**


**Note: the least significant bit of the first byte defines the type of address. if the bit is 0, the address is unicast, otherwise, it is multicast. The broadcast address is a special case of the multicast address in which all bits are 1s.**

Example: define the type of the following destination addresses:

a. 4A:30:10:21:1O:1A
b. 47:20:1B:2E:08:EE
c. FF:FF:FF:FF:FF:FF

**Solution**

To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast. Therefore, we have the following:
a. This is a unicast address because A in binary is 1010 (even).
b. This is a multicast address because 7 in binary is 0111 (odd).
c. This is a broadcast address because all digits are F's.
The way the addresses are sent out on line is different from the way they are written
in hexadecimal notation. The transmission is left-to-right, byte by byte; however, for each byte, the least significant bit is sent first and the most significant bit is sent last.
This means that the bit that defines an address as unicast or multicast arrives first at the receiver.

**Access Method: CSMA/CD**

Standard Ethernet uses 1-persistent CSMA/CD Slot Time. In an Ethernet network, the round-trip time required for a frame to travel from one end of a maximum-length network to the other plus the time needed to send the jam sequence is called the slot time.

**Slot time = round-trip time + time required to send the jam sequence**

The slot time in Ethernet is defined in bits. It is the time required for a station to send 512 bits. This means that the actual slot time depends on the data rate; for traditional 10-Mbps Ethernet it is 51.2 μs.

**Slot Time and Collision:** The choice of a 512-bit slot time was not accidental. It was
chosen to allow the proper functioning of CSMA/CD. To understand the situation, let us consider two cases.

In the first case, we assume that the sender sends a minimum-size packet of 512 bits.

Before the sender can send the entire packet out, the signal travels through the network and reaches the end of the network. If there is another signal at the end of the network (worst case), a collision occurs. The sender has the opportunity to abort the sending of the frame and to send a jam sequence to inform other stations of the collision. The round-trip time plus the time required to send the jam sequence should be less than the time needed for the sender to send the minimum frame, 512 bits. The sender needs to be aware of the collision before it is too late, that is, before it has sent the entire frame.

In the second case, the sender sends a frame larger than the minimum size (between 512 and 1518 bits). In this case, if the station has sent out the first 512 bits and has not heard a collision, it is guaranteed that collision will never occur during the transmission of this frame. The reason is that the signal will reach the end of the network in less than one-half the slot time. If all stations follow the *CSMA/CD* protocol, they have already sensed the existence of the signal (carrier) on the line and have refrained from sending. If they sent a signal on the line before one-half of the slot time expired, a collision has occurred and the sender has sensed the collision. In other words, collision can only occur during the first half of the slot time, and if it does, it can be sensed by the sender during the slot time. This means that after the sender sends the first 512 bits, it is guaranteed that collision will not occur during the transmission of this frame. The medium belongs to the sender, and no other station will use it. In other words, the sender needs to listen for a collision only during the time the first 512 bits are sent. Of course, all these assumptions are invalid if a station does not follow the *CSMA/CD* protocol. In this case, we do not have a collision, we have a corrupted station. Slot Time and Maximum Network Length There is a relationship between the slot time and the maximum length of the network (collision domain). It is dependent on the propagation speed of the signal in the particular medium. In most transmission media, the signal propagates at 2 x 108 m/s (two-thirds of the rate for propagation in air). For traditional Ethernet, we calculate

Max Length = Propagation Speed x Slot Time/2

Max Length= $(2x\ 10^{8})$ X$(512\ X10^{-6}\ (2)$= 5120m

Of course, we need to consider the delay times in repeaters and interfaces, and the time required to send the jam sequence. These reduce the maximum-length of a traditional Ethernet network to 2500 m, just 48 percent of the theoretical calculation.

Max Length=2500 m

**Physical Layer**

The Standard Ethernet defines several physical layer implementations; four of the most common, are shown in **figure**:

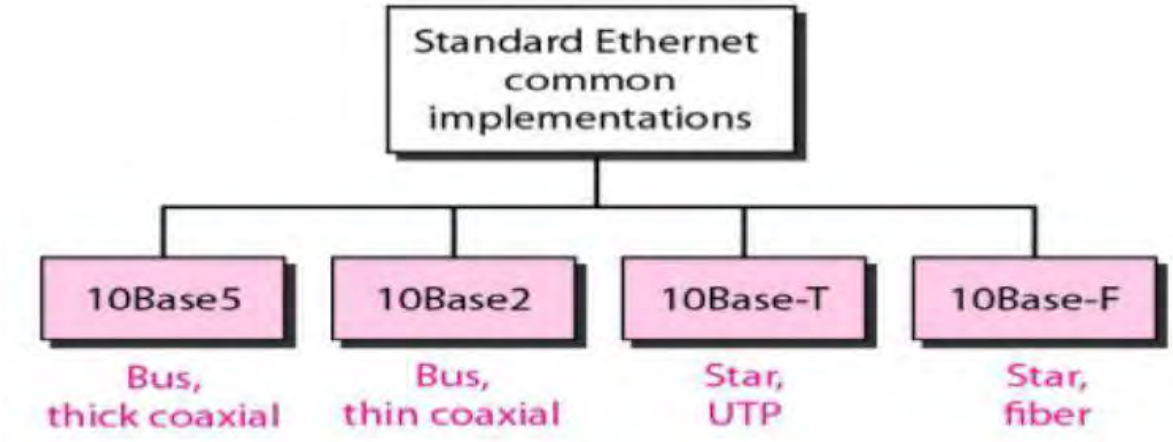
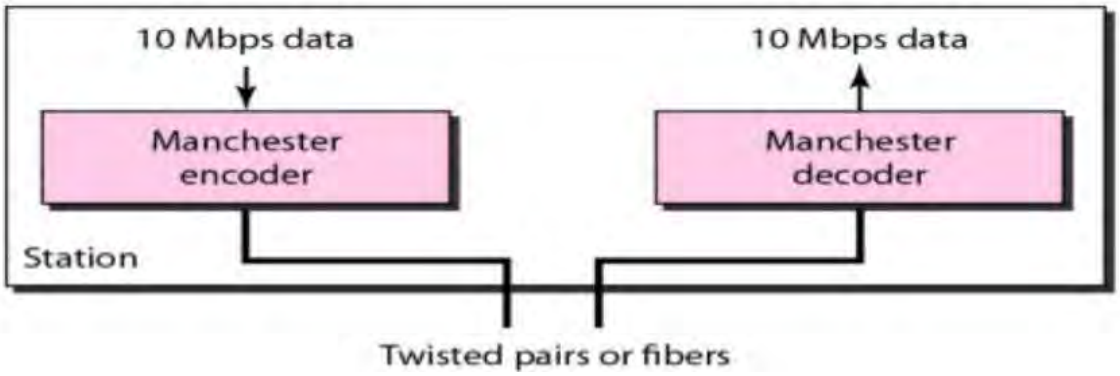



Figure: (*Categories of Standard Ethernet*)

**Encoding and Decoding**

All standard implementations use digital signaling (baseband) at 10 Mbps. At the sender, data are converted to a digital signal using the Manchester scheme, at the receiver; the received signal is interpreted as Manchester and decoded into data. The Manchester encoding is self-synchronous, providing a transition at each bit interval.

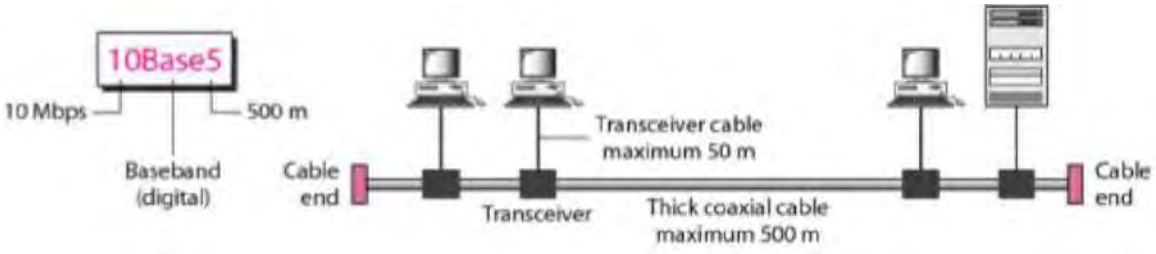
Figure: Encoding in a Standard Ethernet implementation




**10Base5: Thick Ethernet**

The first implementation is called **10Base5, thick Ethernet, or Thick net.** The nickname derives from the size of the cable, which is roughly the size of a garden hose and too stiff to bend with your hands. 10Base5 was the first Ethernet specification to use a bus topology with an external **transceiver** (transmitter/receiver) connected via a tap to a thick coaxial cable. **Figure** below shows a schematic diagram of a 10Base5 implementation.

Figure: 10*Base5 implementation*




The transceiver is responsible for transmitting, receiving, and detecting collisions. The transceiver is connected to the station via a transceiver cable that provides separate paths for sending and receiving.

The maximum length of the coaxial cable must not exceed *500* m, otherwise, there is excessive degradation of the signal. If a length of more than *500*m is needed, up to five segments, each a maximum of 500-meter, can be connected using repeaters.

***LECTURE NOTE: 18***

# 10Base2: Thin Ethernet

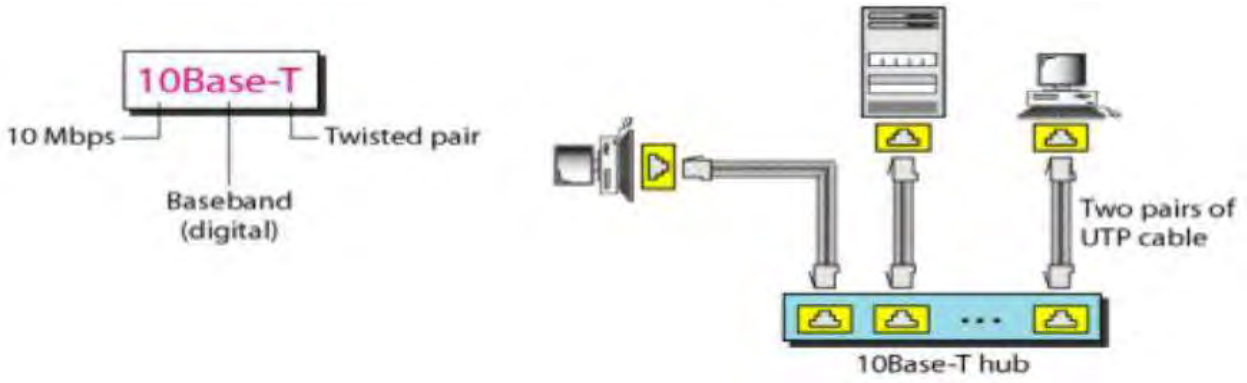
The second implementation is called 10Base2, **thin** Ethernet, or **Cheaper n**et. 10Base2 also uses a bus topology, but the cable is much thinner and more flexible. The cable can be bent to pass very close to the stations. In this case, the transceiver is normally part of the network interface card (NIC), which is installed inside the station.

Figure: *10Base2 implementation*


## 10BASE-T: TWISTED-PAIR ETHERNET

The third implementation is called **10Base-T** or **twisted-pair Ethernet**. 10Base-T uses a physical star topology. The stations are connected to a hub via two pairs of twisted cable, as shown below

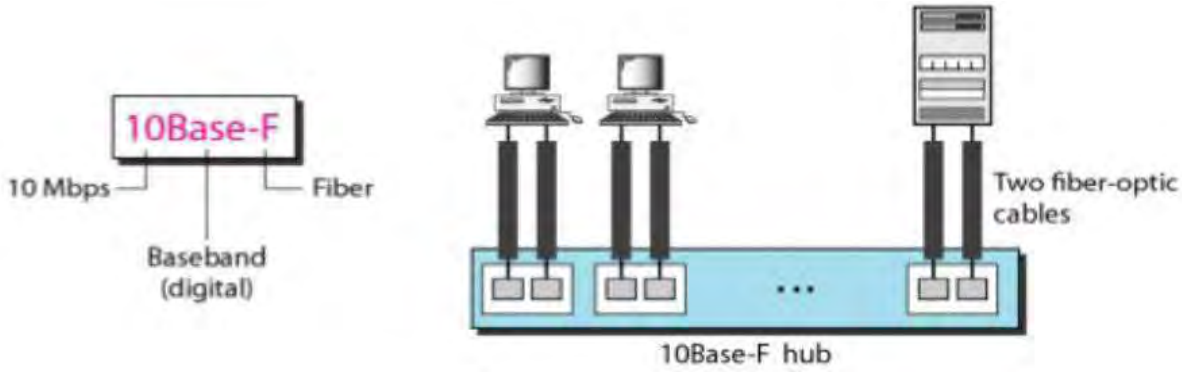
Figure: *10Base-T implementation*




## 10BASE-F: FIBER ETHERNET

Among the several types of optical fiber 10-Mbps Ethernet, the most common is called **10Base-F**. 10Base-F uses a star topology to connect stations to a hub. The stations are connected to the hub using two fiber-optic cables, as shown below

Figure: 10Base-F implementation




**Summary of Standard Ethernet implementations in a tabular form**

| *Characteristics* | *lOBase5* | *lOBase2* | *lOBase-T* | *IOBase-F* |
|---|---|---|---|---|
| Media | Thick coaxial cable | Thin coaxial cable | 2UTP | 2 Fiber |
| Maximum length | 500m | 185 m | 100m | 2000m |
| Line encoding | Manchester | Manchester | Manchester | Manchester |

## CHANGES IN THE STANDARD

The 10-Mbps Standard Ethernet has gone through several changes before moving to the higher data rates. These changes actually opened the road to the evolution of the Ethernet to become compatible with other high-data-rate LANs. Some of these changes are:

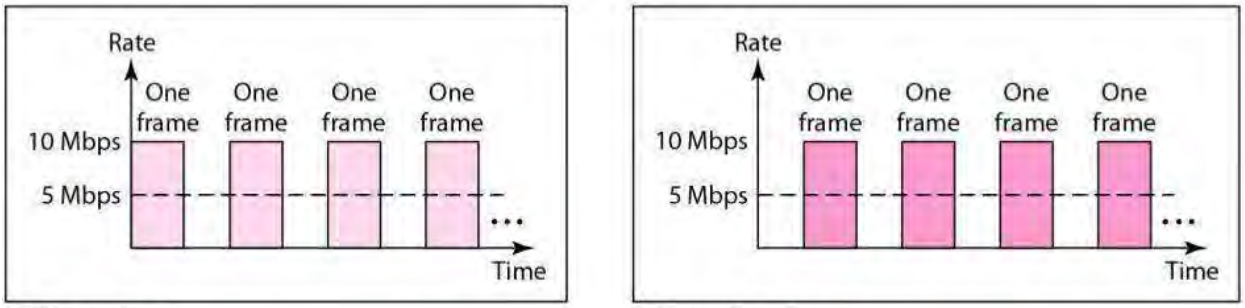
## BRIDGED ETHERNET

The first step in the Ethernet evolution was the division of a LAN by bridges. Bridges have two effects on an Ethernet LAN: They raise the bandwidth and they separate collision domains.

## RAISING THE BANDWIDTH

In an un bridged Ethernet network, the total capacity (10 Mbps) is shared among all stations with a frame to send; the stations share the bandwidth of the network. If only one station has frames to send, it benefits from the total capacity (10 Mbps). But if more than one station needs to use the network, the capacity is shared. For example, if two stations have a lot of frames to send, they probably alternate in usage. When one station is sending, the other one refrains from sending. We can

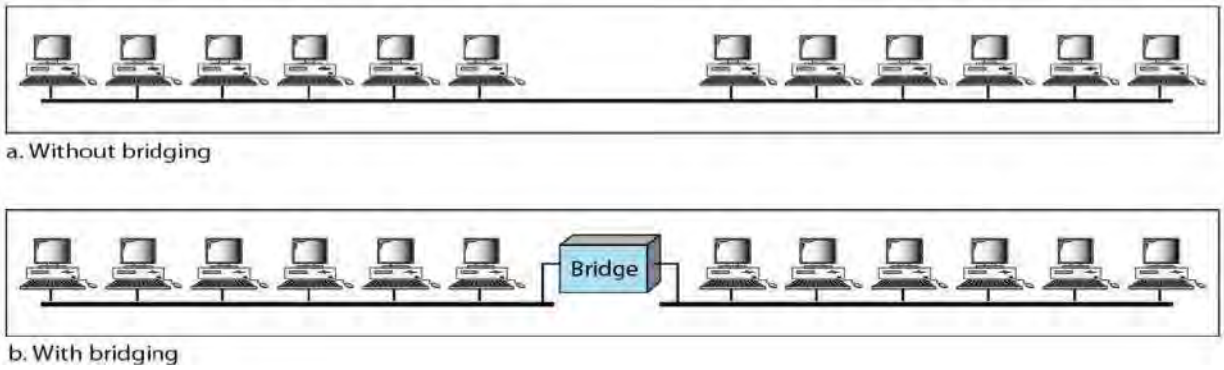
say that, in this case, each station on average, sends at a rate of 5 Mbps. **Figure** shows the situation.




a. First station

b. Second station

A bridge divides the network into two or more networks. Bandwidth-wise, each network is independent. For example, in **Figure** below, a network with 12 stations is divided into two networks, each with 6 stations. Now each network has a capacity of 10 Mbps. The 10-Mbps capacity in each segment is now shared between 6 stations (actually 7 because the bridge acts as a station in each segment), not 12 stations. In a network with a heavy load, each station theoretically is offered 10/6 Mbps instead of 10/12 Mbps, assuming that the traffic is not going through the bridge. It is obvious that if we further divide the network, we can gain more bandwidth for each segment. For example, if we use a four-port bridge, each station is now offered 10/3 Mbps, which is 4 times more than an un bridged network.



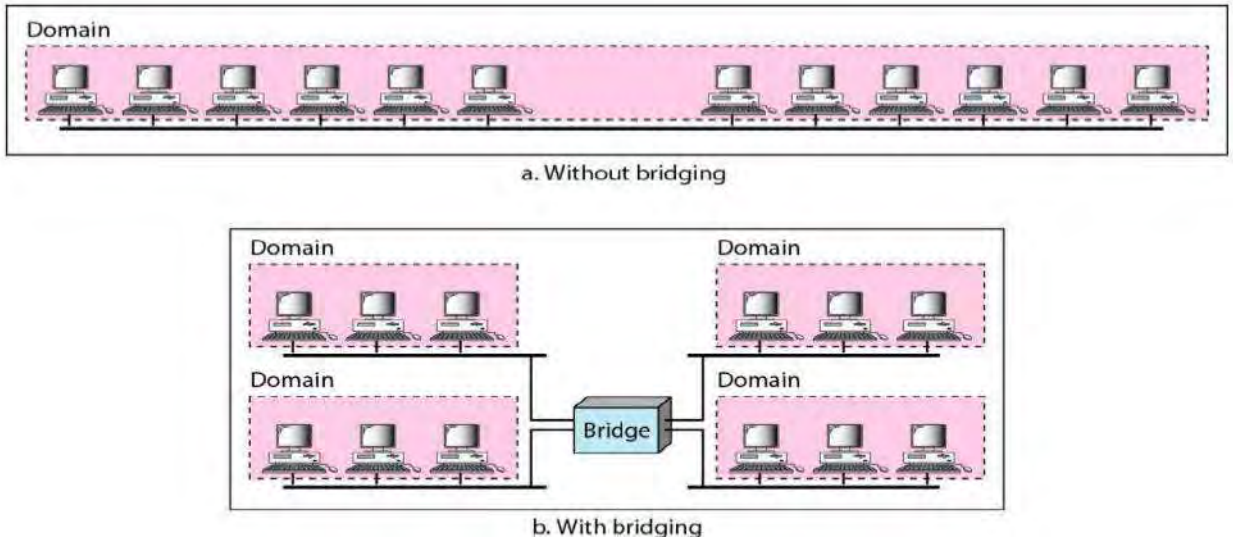

a. Without bridging

b. With bridging

## SEPARATING COLLISION DOMAINS

Another advantage of a bridge is the separation of the collision domain. **Figure** below shows the collision domains for an un bridged and a bridged network. You can see that the collision domain becomes much smaller and the probability of collision is reduced tremendously. Without bridging, 12 stations contend for access to the medium; with bridging only 3 stations contend for access to the medium.

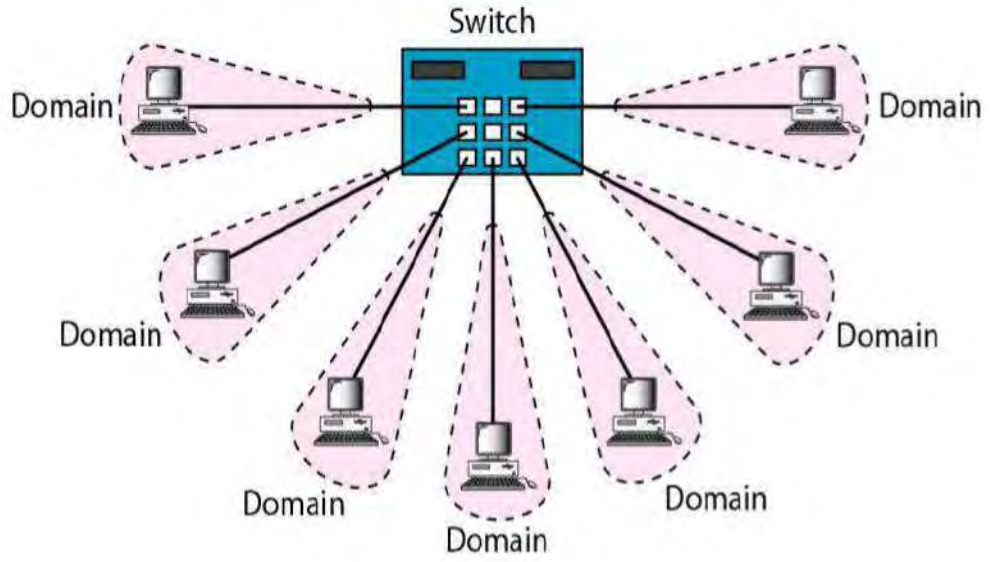
**Figure:** *Collision domains in an un bridged network and a bridged network*




## SWITCHED ETHERNET

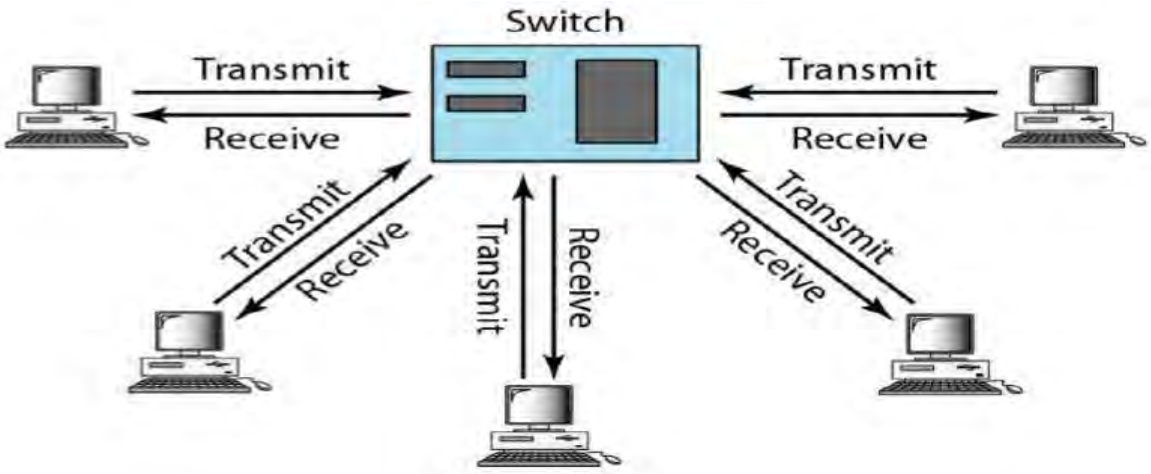
The idea of a bridged LAN can be extended to a switched LAN. Instead of having two to four networks, why not have *N* networks, where *N* is the number of stations on the LAN? In other words, if we can have a multiple-port bridge, why not have an *N-port* switch? In this way, the bandwidth is shared only between the station and the switch (5 Mbps each). In addition, the collision domain is divided into *N* domains.

A layer 2 switch is an *N-port* bridge with additional sophistication that allows faster handling of the packets. Evolution from a bridged Ethernet to a switched Ethernet was a big step that opened the way to an even faster Ethernet, as we will see. **Figure** below shows a switched LAN.




**FULL-DUPLEX ETHERNET**

One of the limitations of 10Base5 and 10Base2 is that communication is half-duplex (10Base-T is always full-duplex); a station can either send or receive, but may not do both at the same time. The next step in the evolution was to move from switched Ethernet to full-duplex switched Ethernet. The full-duplex mode increases the capacity of each domain from 10 to 20 Mbps. **Figure** below shows a switched Ethernet in full-duplex mode. **Note** that instead of using one link between the station and the switch, the configuration uses two links: one to transmit and one to receive.




**NO NEED FOR CSMA/CD**

In full-duplex switched Ethernet, there is no need for the *CSMA/CD* method. In a full duplex switched Ethernet, each station is connected to the switch via two separate links. Each station or switch can send and receive independently without worrying about collision. Each link is a point-to-point dedicated path between the

station and the switch. There is no longer a need for carrier sensing; there is no longer a need for collision detection. The job of the MAC layer becomes much easier. The carrier sensing and collision detection functionalities of the MAC sub layer can be turned off.

## MAC CONTROL LAYER

Standard Ethernet was designed as a connectionless protocol at the MAC sub layer. There is no explicit flow control or error control to inform the sender that the frame has arrived at the destination without error. When the receiver receives the frame, it does not send any positive or negative acknowledgment. To provide for flow and error control in full-duplex switched Ethernet, a new sub layer, called the MAC control, is added between the LLC sub layer and the MAC sub layer.

## FAST ETHERNET

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fibber Channel. IEEE created Fast Ethernet under the name 802.3u. Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps. The goals of Fast Ethernet can be summarized as follows:

1. Upgrade the data rate to 100 Mbps.
2. Make it compatible with Standard Ethernet.
3. Keep the same 48-bit address.
4. Keep the same frame format.
5. Keep the same minimum and maximum frame lengths.

## MAC SUB LAYER

It uses star topology to connect. For the star topology, there are two choices: half duplex and full duplex. The access method is the same *(CSMA/CD)* for the half-duplex approach, for full duplex Fast Ethernet, there is no need for *CSMA/CD.* A new feature added to Fast Ethernet is called **auto negotiation**. It allows a station or a hub a range of capabilities. Auto negotiation allows two devices to negotiate the mode or data rate of operation. It was designed particularly for the following purposes:

To allow incompatible devices to connect to one another. For example, a device with a maximum capacity of 10 Mbps can communicate with a device with

a 100 Mbps capacity to allow one device to have multiple capabilities and station to check a hub's capabilities.

***AUTO NEGOTIATING***

A new feature added to Fast Ethernet is called auto negotiation. Auto negotiation allows two devices to negotiate the mode or data rate of operation. It was designed particularly for the following purposes:

- To allow incompatible devices to connect to one another. For example, a device with

a maximum capacity of 10 Mbps can communicate with a device with a 100 Mbps capacity

(but can work at a lower rate).

- To allow one device to have multiple capabilities.
- To allow a station to check a hub's capabilities.

***PHYSICAL LAYER***

The physical layer in Fast Ethernet is more complicated than the one in Standard Ethernet.

**TOPOLOGY**

Fast Ethernet is designed to connect two or more stations together. If there are only two stations, they can be connected point-to-point. If there are more stations need to be connected in a star topology with a hub or a switch at the centre.

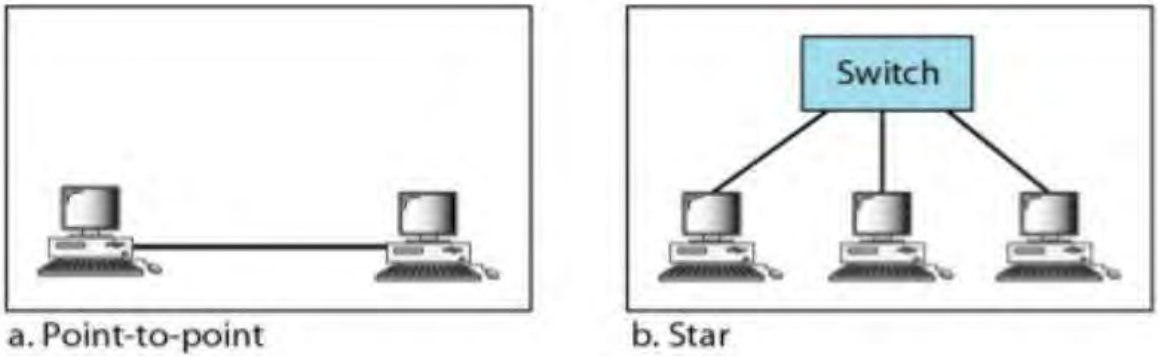
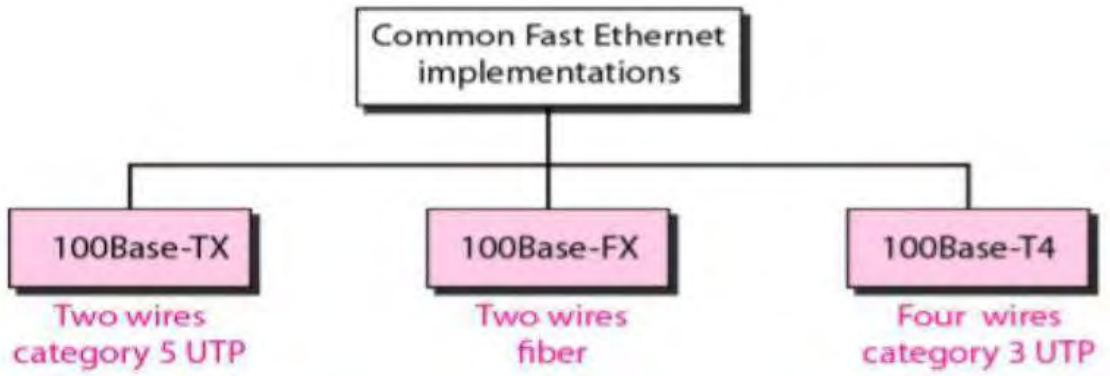



Figure: *Fast Ethernet topology*

**IMPLEMENTATION**

Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire. The two-wire implementation can be either category 5 UTP (100Base-TX) or fiber-optic cable (100Base-FX). The four-wire implementation is designed only for category 3 UTP (l00Base-T4). Shown below

Figure: *Fast Ethernet implementations*



# ENCODING

Manchester encoding needs a 200-Mbaud bandwidth for a data rate of 100 Mbps, which makes it unsuitable for twisted-pair cable medium. For this reason, the Fast Ethernet designers sought some alternative encoding/decoding scheme.

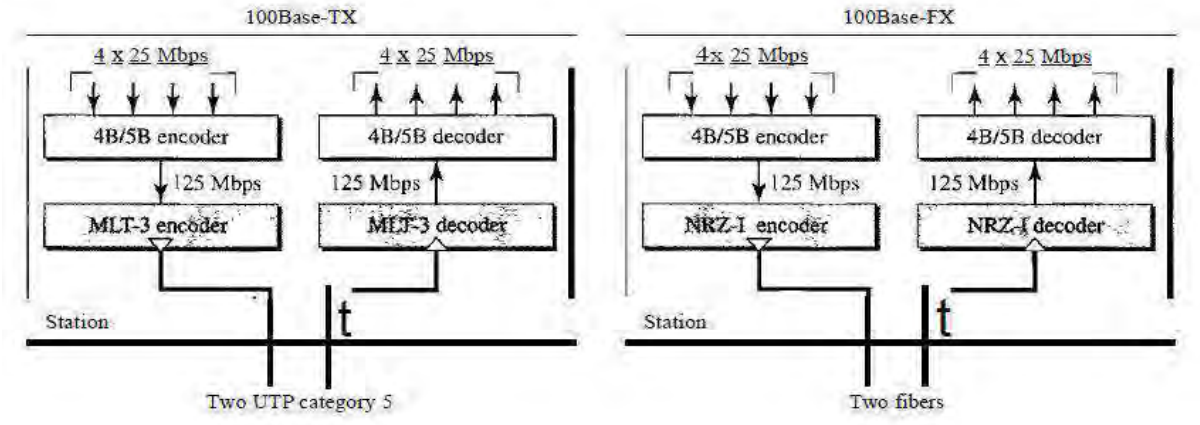



Figure: *Encoding for Fast Ethernet implementation*

**100Base-TX:** it uses two pairs of twisted-pair cable (either category 5 UTP or STP). For this implementation, the MLT-3 scheme was selected since it has good bandwidth performance. Since MLT-3 is not a self-synchronous line coding scheme, 4B/5B block coding is used to provide bit synchronization by preventing the occurrence of a long sequence of 0s and 1s . This creates a data rate of 125 Mbps, which is fed into MLT-3 for encoding.

**100Base-FX:** it uses two pairs of fiber-optic cables. Optical fiber can easily handle high bandwidth requirements by using simple encoding schemes. The designers of 100Base-FX selected the NRZ-I encoding scheme for this implementation. To overcome this problem, the designers used 4B/5B block encoding as we described for 100Base-TX. The block encoding increases the bit rate from 100 to 125 Mbps, which can easily be handled by fiber-optic cable.

**Summary of Fast Ethernet implementations**

| *Characteristics* | *lOOBase-TX* | *lOOBase-FX* | *100Base-T4* |
|---|---|---|---|
| Media | Cat 5 UTP or STP | Fiber | Cat 4 UTP |
| Number of wires | 2 | 2 | 4 |
| Maximum length | 100m | 100m | 100m |
| Block encoding | 4B/5B | 4B/5B | |
| Line encoding | MLT-3 | NRZ-I | 8B/6T |

## GIGABIT ETHERNET

The higher data rate resulted in the design of the Gigabit Ethernet protocol (1000 Mbps). The IEEE committee calls it the Standard 802.3z. The goals of the Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 1 Gbps.
2. Make it compatible with Standard or Fast Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format; Keep the same minimum and maximum frame lengths.

### MAC Sub layer

Gigabit Ethernet has two distinctive approaches for medium access: half-duplex and full-duplex. Almost all implementations of Gigabit Ethernet follow the full-duplex approach.

### Full-Duplex Mode:

In full-duplex mode, there is a central switch connected to all computers or other switches. In this mode, each switch has buffers for each input port in which data are stored until they are transmitted. There is no collision in this mode. This means that *CSMA/CD* is not used.

### Half-Duplex Mode:

Gigabit Ethernet can also be used in half-duplex mode, although it is rare. In this case, a switch can be replaced by a hub, which acts as the common cable in which a collision might occur. The half-duplex approach uses *CSMA/CD*. The maximum length of the network in this approach is totally dependent on the minimum frame size.

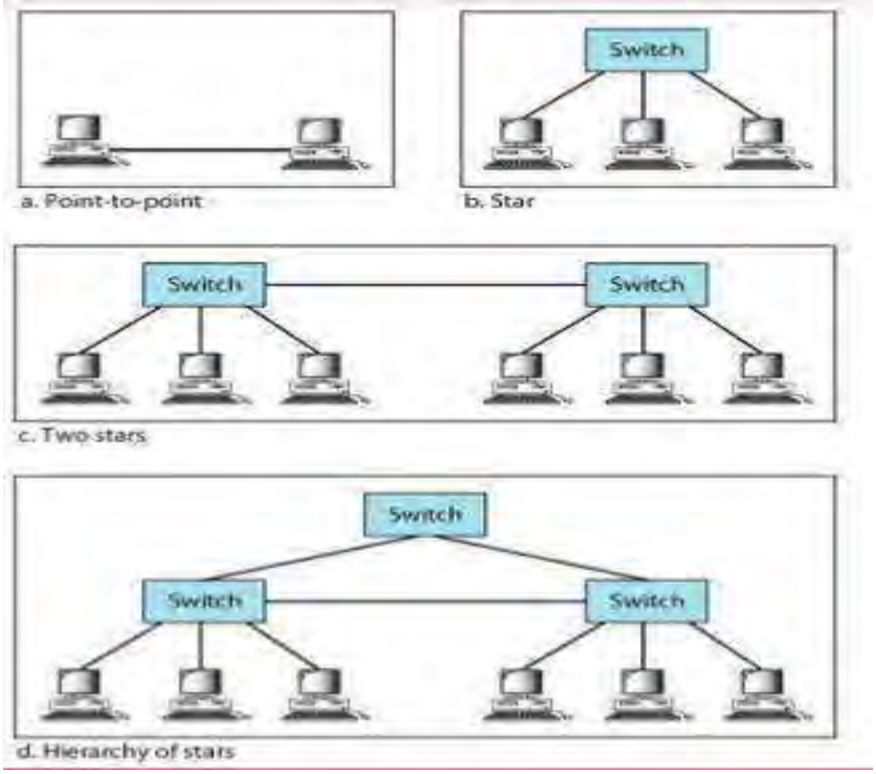
**Physical Layer**

The physical layer in Gigabit Ethernet is more complicated than that in Standard or Fast Ethernet.

**Topology**

Gigabit Ethernet is designed to connect two or more stations. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected
in a star topology with a hub or a switch at the center.

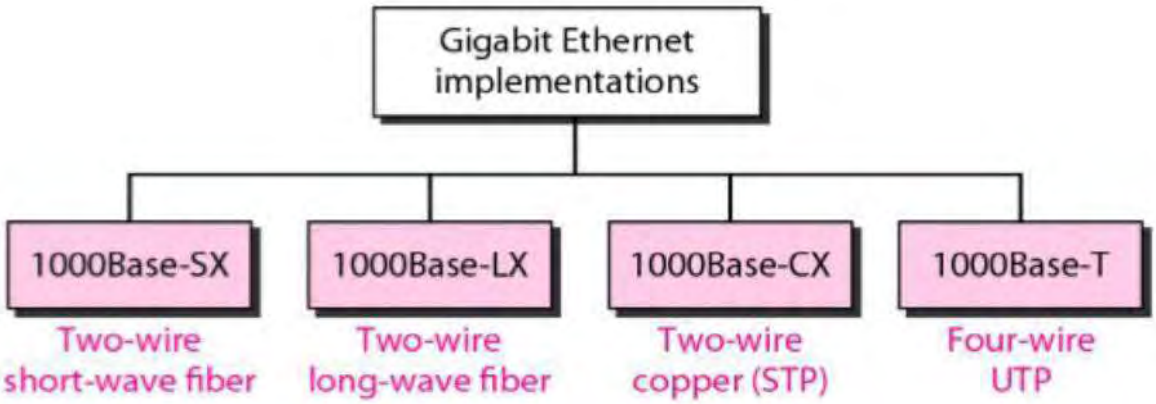
Topologies of Gigabit Ethernet




**Implementation**

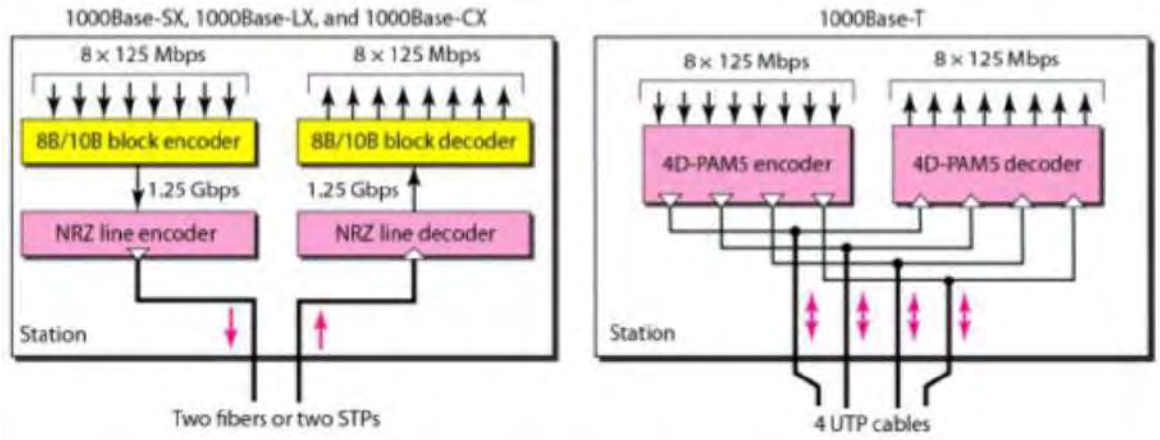
Gigabit Ethernet can be categorized as either a two-wire or a four-wire implementation.

The two-wire implementations use fiber-optic cable (1000Base-SX, short-wave, or 1000Base-LX, long-wave), or STP (1000Base-CX). The four-wire version uses category 5 twisted-pair cable (1000Base-T). 1000Base-T was designed in response to those users who had already installed this wiring for other purposes such as Fast Ethernet or telephone services.


Gigabit Ethernet implementations
1000Base-SX
1000Base-LX
1000Base-CX
1000Base-T
Two-wire short-wave fiber
Two-wire long-wave fiber
Two-wire copper (STP)
Four-wire UTP

# ENCODING

Figure: *Encoding in Gigabit Ethernet implementations*




Gigabit Ethernet cannot use the Manchester encoding scheme because it involves a very high bandwidth (2 GBaud). The two-wire implementations use an NRZ scheme, but NRZ does not self-synchronize properly. To synchronize bits, particularly at this high data rate, 8B/10B block encodings used. In the four-wire implementation it is not possible to have 2 wires for input and 2 for output, because each wire would need to carry 500 Mbps, which exceeds the capacity for category 5 UTP. As a solution, 4D-PAM5 encoding is used to reduce the bandwidth. Thus, all four wires are involved in both input and output, each wire carries 250 Mbps, which is in the range for category 5 UTP cable.

***Summary of Gigabit Ethernet implementations***

| *Characteristics* | *lOOOBase-SX* | *lOOOBase-LX* | *lOOOBase-CX* | *lOOOBase-T* |
|---|---|---|---|---|
| Media | Fiber short-wave | Fiber long-wave | STP | Cat 5 UTP |
| Number of wires | 2 | 2 | 2 | 4 |
| Maximum length | 550m | 5000m | 25m | 100m |
| Block encoding | 8B/lOB | *8B/lOB* | 8B/lOB | |
| Line encoding | NRZ | NRZ | NRZ | 4D-PAM5 |

## WIRELESS LANS

Wireless communication is one of the fastest-growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas. There are two wireless technologies for LAN: **IEEE 802.11** wireless LANs, sometimes called wireless **Ethernet**, and **Bluetooth**, a technology for small wireless LANs.

### IEEE 802.11

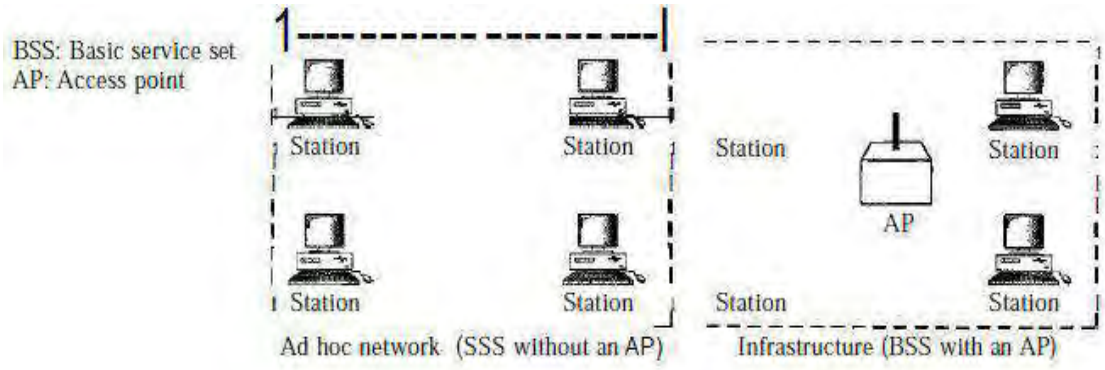
IEEE has defined the specifications for a wireless LAN, called IEEE 802.11, which covers the physical and data link layers.

### Architecture

The standard defines two kinds of services: the **basic service set (BSS)** and the **extended service set (ESS)**.

### *BASIC SERVICE SET*

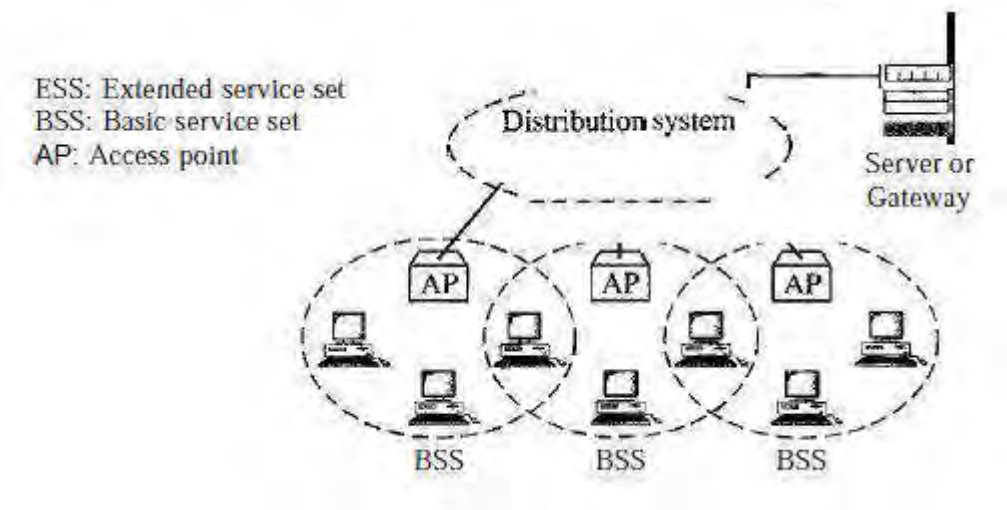
A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the **access point (AP).** This architecture standard shows two sets as: The BSS without an AP is a **stand-alone network or an *ad hoc architecture***. It cannot send data to other BSSs. In this architecture, stations can form an other network without the need of an AP, they can locate one another and agree to be part of a BSS. A BSS with an AP is referred to as an ***infrastructure*** network. In this architecture stations are get connected through the access point.




A BSS without an AP is called an ad hoc network;
a BSS with an AP is called an infrastructure network.

***EXTENDED SERVICE SET***

An extended service set (ESS) is made up of two or more BSSs with APs (access points). In this case, the BSSs are connected through a ***distribution system***, which is usually a wired LAN. The distribution system connects the APs in the BSSs. IEEE 802.11 does not restrict the distribution system, it can be any **IEEE LAN** such as an **Ethernet**. **Note** that the extended service set uses two types of stations: **mobile** and **stationary**. The mobile stations are normal stations inside a BSS. The stationary stations are AP stations that are part of a wired LAN.



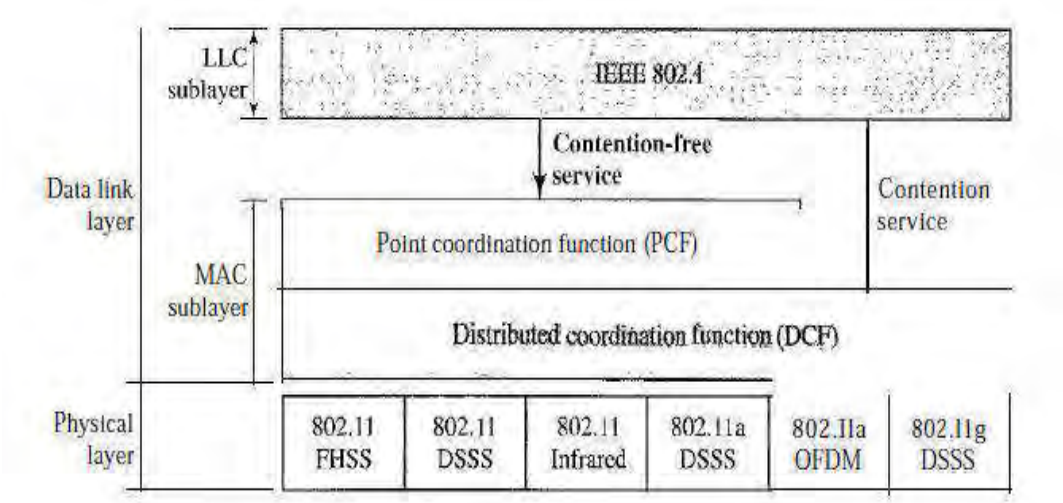

When BSSs are connected, the stations within reach of one another can communicate without the use of an AP. However, communication between two stations in two different BSSs usually occurs via two APs. The idea is similar to communication in a cellular network if we consider each BSS to be a cell and each AP to be a base station**. Note** that a mobile station can belong to more than one BSS at the same time.

***STATION TYPES***

IEEE 802.11 defines three types of stations based on their mobility in a wireless LAN: **no-transition, BSS-transition, and ESS-transition mobility**. A station with no-transition mobility is either stationary (not moving) or moving only inside a BSS. A station with BSS-transition mobility can move from one BSS to another, but the movement is confined inside one ESS. A station with ESS-transition mobility can move from one ESS to another. However, IEEE 802.11 does not guarantee that communication is continuous during the move.

**MAC Sub layer**

IEEE 802.11 defines two MAC sub layers: the distributed **coordination function (DCF)** and **point coordination function (PCF). Figure** below shows the relationship between the
two MAC sub layers, the LLC sub layer, and the physical layer.




***DISTRIBUTED COORDINATION FUNCTION***

One of the two protocols defined by IEEE at the MAC sub layer is called the distributed coordination function (DCF). DCF uses *CSMA/CA* as the access method. Wireless LANs cannot implement *CSMA/CD* for three reasons:

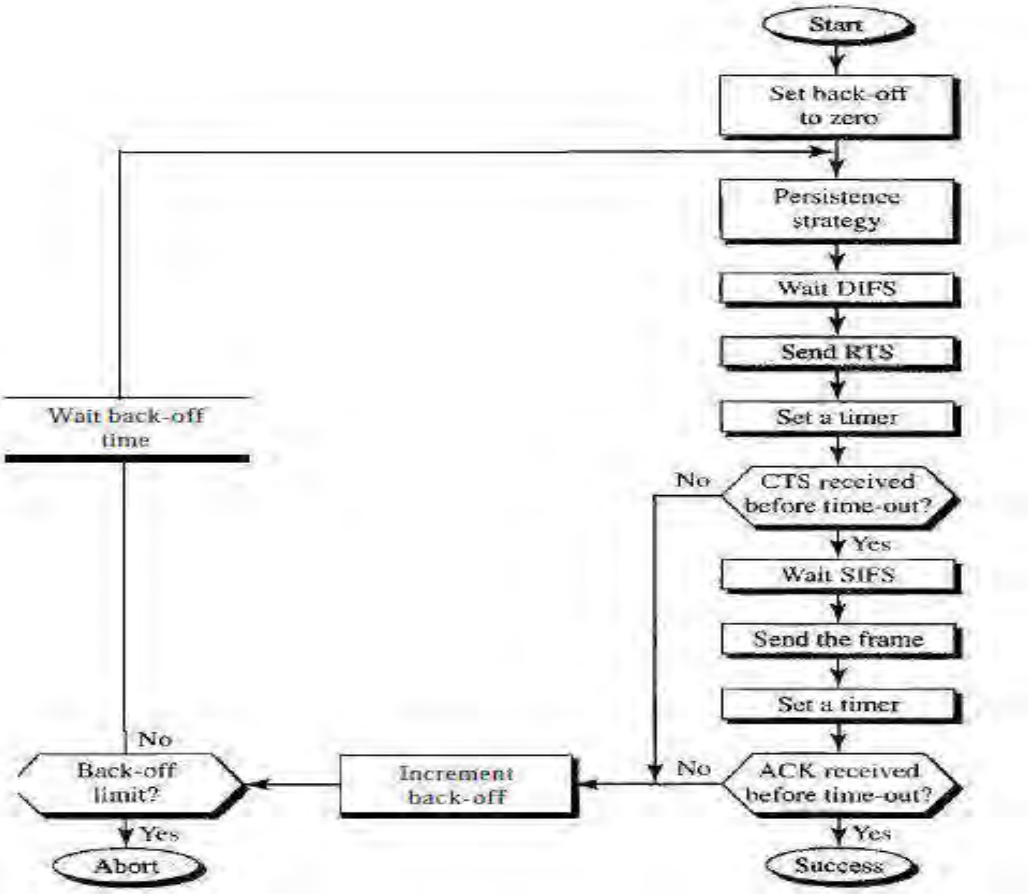
1. For collision detection a station must be able to send data and receive collision signals at the same time. This can mean costly stations and increased bandwidth requirements.

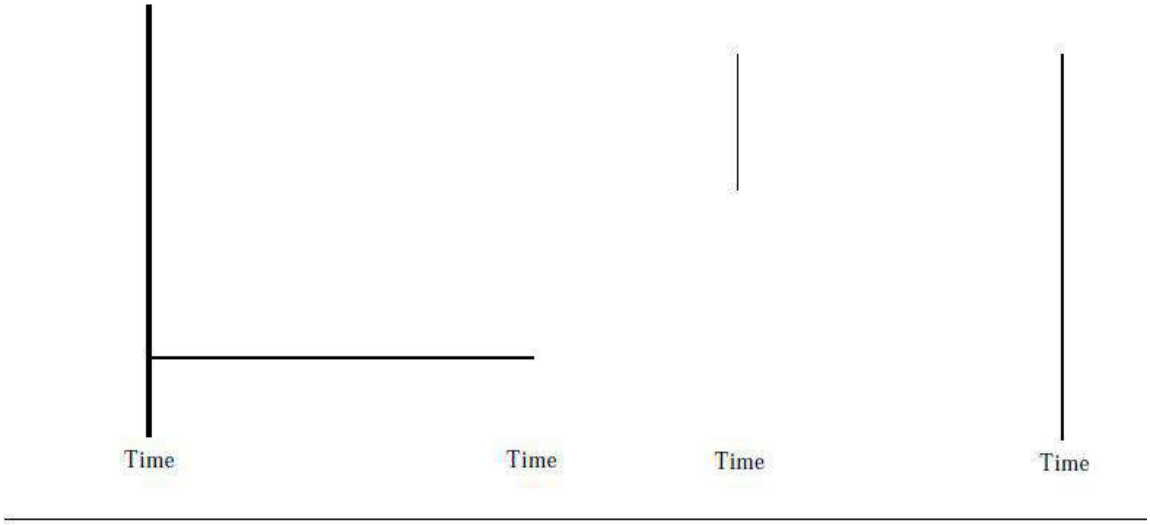
2. Collision may not be detected because of the hidden station problem.

3. The distance between stations can be great. Signal fading could prevent a station at one end from hearing a collision at the other end.

Process Flowchart in the **Figure** shows the process flowchart for *CSMA/CA* as used in wireless LANs. Frame Exchange Time Line **Figure** below flow chart shows the exchange of data and control frames in time.

Figure: Flow chart for *CSMAICA* as used in wireless LANs



1. Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.

a. The channel uses a persistence strategy with back-off until the channel is idle.

b. After the station is found to be idle, the station waits for a period of time called the

Distributed inter frame space (DIFS), and then the station sends a control frame called the request to send (RTS).

2. After receiving the RTS and waiting a period of time called the short inter frame space (SIFS), the destination station sends a control frame, called the clear to send (CTS), to the source station. This control frame indicates that the destination station is ready to receive data.

3. The source station sends data after waiting an amount of time equal to SIFS.

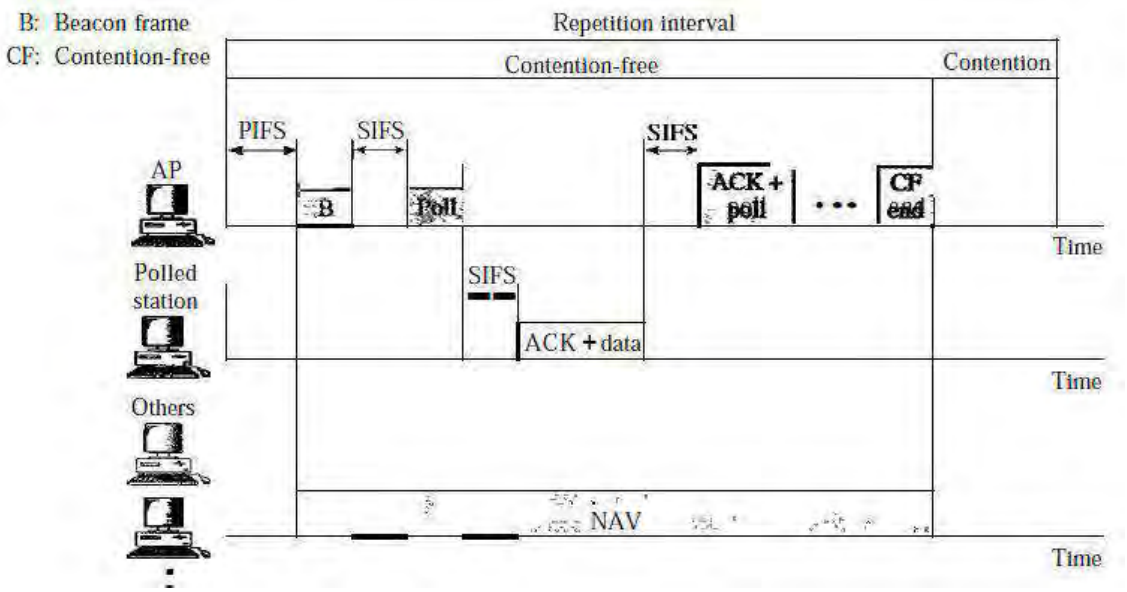
4. The destination station, after waiting an amount of time equal to SIFS, sends an acknowledgment to show that the frame has been received. Acknowledgment is needed in this protocol because the station does not have any means to check for the successful arrival of its data at the destination. On the other hand, the lack of collision in *CSMA/CD* is a kind of indication to the source that data have arrived.

**Network Allocation Vector:** *collision avoidance* aspect of this protocol accomplished by the key is a feature called NAV (network allocation vector). When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel. The stations that are affected by this transmission create a timer called a **network allocation vector (NAV**) that shows how much time must pass before these stations are allowed to check the channel for idleness. Each time a station accesses the system and sends an RTS frame, other stations start their NAV. In other words, each station, before sensing the physical medium to see if it is idle, first checks its NAV to see if it has expired.

**Collision during Handshaking**: if there is collision during the time when RTS or CTS control frames are in transition, often called the handshaking period? Two or more stations may try to send RTS frames at the same time. Two or more stations may try to send RTS frames at the same time. These control frames may collide.The back-off strategy is employed, and the sender tries again.

## Point Coordination Function (PCP)

The point coordination function (PCF) is an optional access method that can be implemented in an infrastructure network (not in an ad hoc network). It is implemented on top of the DCF and is used mostly for time-sensitive transmission. PCF has a centralized, contention-free polling access method. The AP (access point) performs polling for stations that are capable of being polled. The stations are polled one after another, sending any data they have to the AP. To give priority to PCF over DCF, another set of inter frame spaces has been defined: PIFS and SIFS. The SIFS is the same as that in DCF, but the PIFS (PCF IFS) is shorter than the DIFS. This means that if, at the same time, a station wants to use only DCF and an AP wants to use PCF, the AP has priority. Due to the priority of PCF over DCF, stations that only use DCF may not gain access to the medium. To prevent this, a repetition interval has been designed to cover both **contention-free** (PCF) and **contention-based** (DCF) traffic. The repetition interval, which is repeated continuously, starts with a control frame, called a **beacon frame**. When the stations hear the beacon frame, they start their NAV for the duration of the contention-free period of the repetition interval. Below figure shows an example of a repetition interval.



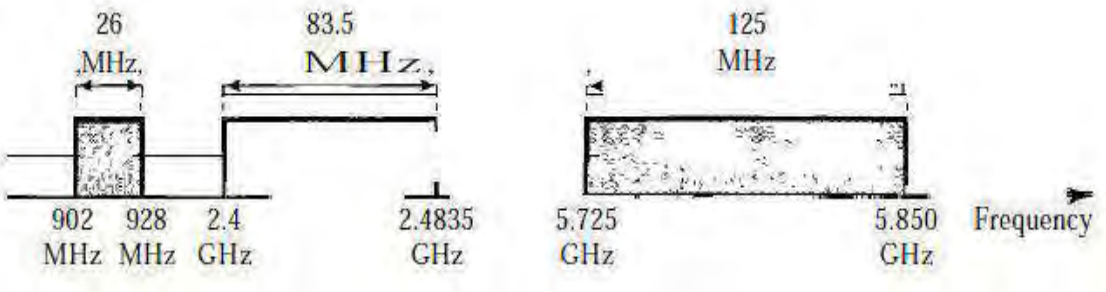

During the repetition interval, the PC (point controller) can send a poll frame, receive data, send an ACK, receive an ACK, or do any combination of these (802.11 uses piggybacking). At the end of the contention-free period, the PC sends a CF end (contention-free end) frame to allow the contention-based stations to use the medium.

## PHYSICAL LAYER

We discuss six specifications, as shown in Table

| *IEEE* | *Technique* | *Band* | *Modulation* | *Rate (Mbps)* |
|---|---|---|---|---|
| 802.11 | FHSS | 2.4 GHz | FSK | 1 and 2 |
| | DSSS | 2.4 GHz | PSK | 1 and 2 |
| | | Infrared | PPM | 1 and 2 |
| 802.11a | OFDM | 5.725 GHz | PSKorQAM | 6 to 54 |
| 802.11b | DSSS | 2.4 GHz | PSK | 5.5 and 11 |
| 802.11g | OFDM | 2.4 GHz | Different | 22 and 54 |

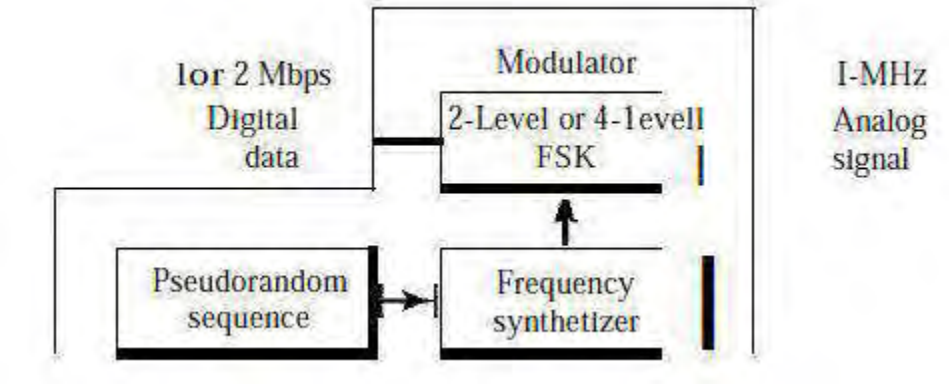
All implementations, except the infrared, operate in the *industrial, scientific,* and *medical (ISM)* band, which defines three unlicensed bands in the three ranges 902-928 MHz, 2.400-4.835 GHz, and 5.725-5.850 GHz, as shown in Figure:




### *IEEE 802.11 FHSS*

IEEE 802.11 FHSS uses the frequency-hopping spread spectrum (FHSS) method. FHSS uses the 2.4-GHz ISM band. The band is divided into 79 sub bands of 1 MHz (and some guard bands). A pseudo random number generator selects the hopping sequence. The modulation technique in this specification is either two-level FSK or four-level FSK with 1 or 2 bits baud, which results in a data rate of 1 or 2 Mbps.

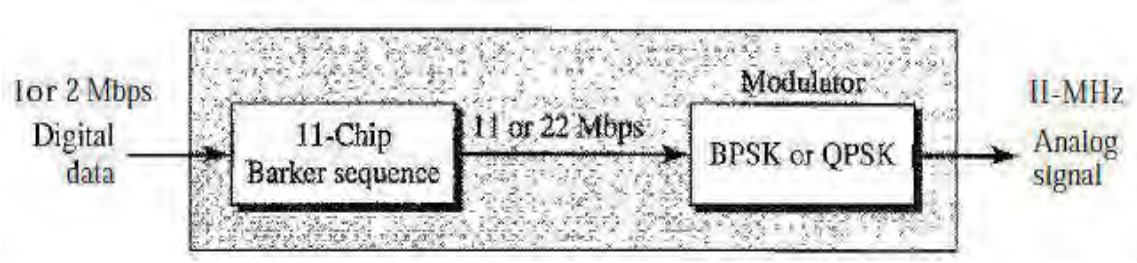
Figure: *Physical layer of IEEE 802.11 FHSS*




***IEEE 802.11 DSSS***

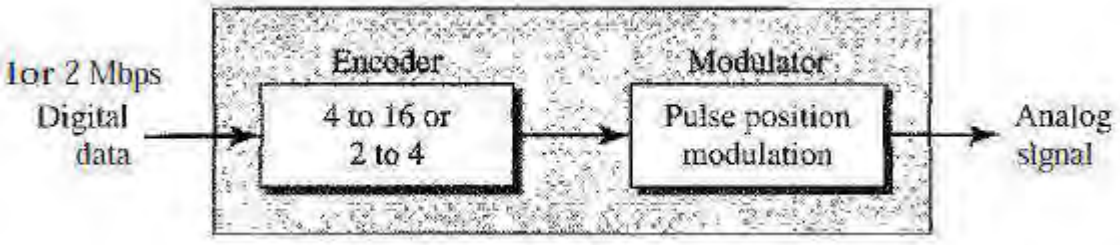
IEEE 802.11 DSSS uses the direct sequence spread spectrum (DSSS) method. DSSS uses the 2.4-GHz ISM band. The modulation technique in this specification is PSK at 1 Mbaud/s. The system allows 1 or 2 bits baud (BPSK or QPSK), which results in a data rate of 1 or 2 Mbps.

*Figure: Physical layer of IEEE 802.11 DSSS*




***IEEE 802.11 Infrared***

IEEE 802.11 infrared uses infrared light in the range of 800 to 950 nm. The modulation technique is called pulse position modulation (PPM).




For a 1-Mbps data rate, a 4-bit sequence is first mapped into a 16-bit sequence in which only one bit is set to 1 and the rest are set to 0. For a 2-Mbps data rate, a 2-bit sequence is first mapped into a 4-bit sequence in which only one bit is set to 1

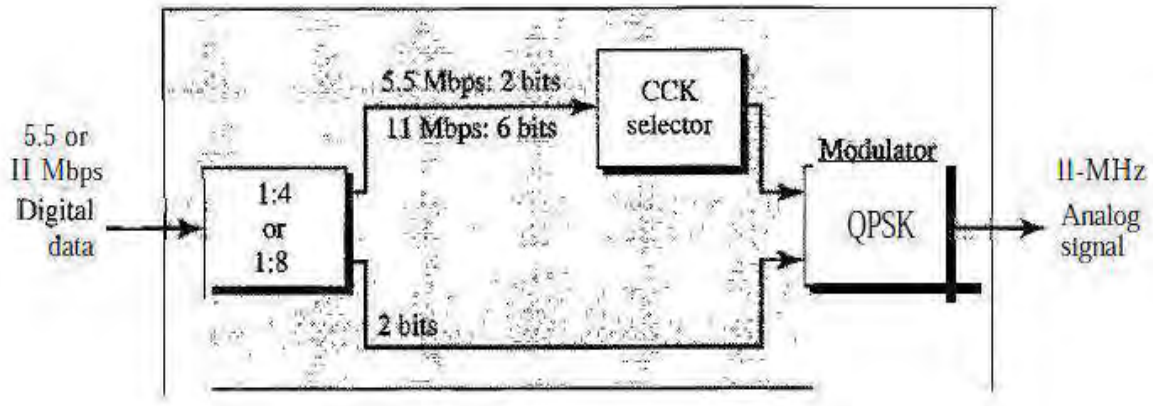
and the rest are set to 0. The mapped sequences are then converted to optical signals; the presence of light specifies 1, the absence of light specifies 0.

### *IEEE 802.11a OFDM*

IEEE 802.Ila OFDM describes the orthogonal frequency-division multiplexing (OFDM) method for signal generation in a 5-GHz ISM band. OFDM is similar to FDM with one major difference: All the sub bands are used by one source at a given time. Sources contend with one another at the data link layer for access. The band is divided into 52 sub bands, with 48 sub bands for sending 48 groups of bits at a time and 4 sub bands for control information. Dividing the band into sub bands diminishes the effects of interference. If the sub bands are used randomly, security can also be increased. OFDM uses PSK and QAM for modulation. The common data rates are 18 Mbps (PSK) and 54 Mbps (QAM).

### *IEEE 802.11b DSSS*

IEEE 802.11 b DSSS describes the high-rate direct sequence spread spectrum (HRDSSS) method for signal generation in the 2.4-GHz ISM band. HR-DSSS is similar to DSSS except for the encoding method, which is called complementary code keying (CCK). CCK encodes 4 or 8 bits to one CCK symbol. To be backward compatible with DSSS, HR-DSSS defines four data rates: 1,2, 5.5, and 11 Mbps. The first two use the same modulation techniques as DSSS. The 5.5-Mbps version uses BPSK and transmits at 1.375 Mbaudls with 4-bit CCK encoding. The 11-Mbps version uses QPSK and transmits at 1.375 Mbps with 8-bit CCK encoding. **Figure shows** the modulation technique for this standard.




### *IEEE 802.11g*

This new specification defines forward error correction and OFDM using the 2.4-GHz ISM band. The modulation technique achieves a 22- or 54-Mbps data rate. It is backward compatible with 802.11b, but the modulation technique is OFDM.

*LECTURE NOTE: 21*

# BLUETOOTH

Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers (desktop and laptop), cameras, printers, coffee makers, and so on. A Bluetooth LAN is an **ad hoc** network, which means that the network is formed spontaneously. The devices connected to this network are called **gadgets** and the network is called a **piconet.** A Bluetooth LAN can even be connected to the Internet if one of the gadgets(devices) has this capability. A Bluetooth LAN, by nature, cannot be large. If there are many gadgets that try to connect, there is chaos. Bluetooth technology has several applications. Peripheral devices such as a wireless mouse or keyboard can communicate with the computer through this technology. Monitoring devices can communicate with sensor devices in a small health care centre, home security devices to connect different sensors to the main security controller etc. Bluetooth was originally started as a project by the **Ericsson Company**. It is named for Harald Blaatand, the king of Denmark (940-981) who united Denmark and Norway. *Blaatand* translates to ***Bluetooth*** in English.

Bluetooth wireless technology is a ***short-range radio technology***, which is developed for Personal Area Network (PAN). It is an ad hoc type network operable over a small area such as a room or hall. Its technology for implementation of protocols is standardized by IEEE 802.15 standard. Bluetooth is a standard for a **small, cheap radio chip to be plugged into computers, printers, mobile phones, etc**. A Bluetooth chip is designed to replace cables by taking the information normally carried by the cable, and transmitting it at a special frequency to a receiver Bluetooth chip.

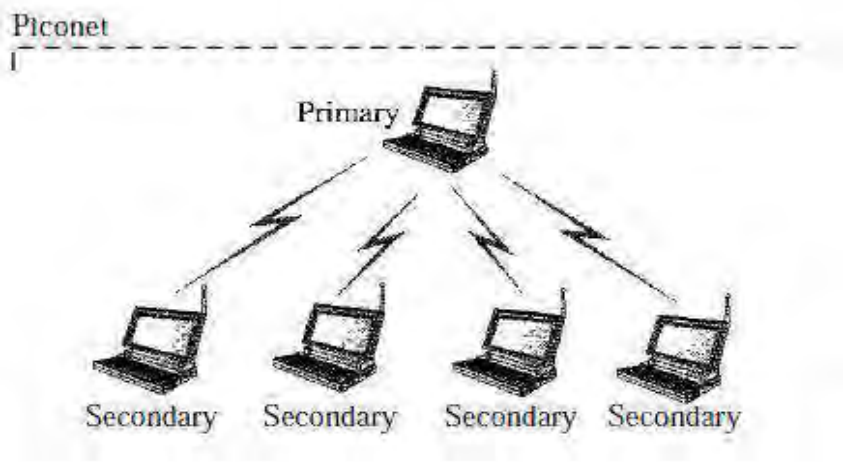
Today, Bluetooth technology is the implementation of a protocol defined by the IEEE 802.15 standard. The standard defines a wireless personal-area network (PAN) operable in an area the size of a room or a hall.

## BLUETOOTH ARCHITECTURE

Bluetooth defines two types of networks architecture: **piconet** and **scatternet.**

***PICONETS***

It a small net that can have up to eight stations, one of which is called the primary, the rest are called secondaries. All the secondary stations synchronize their clocks and hopping sequence with the primary station. **Note** that a piconet can have only one primary station. The communication between the primary and the secondary can be one-to-one or one-to-many. Figure below shows a piconet.

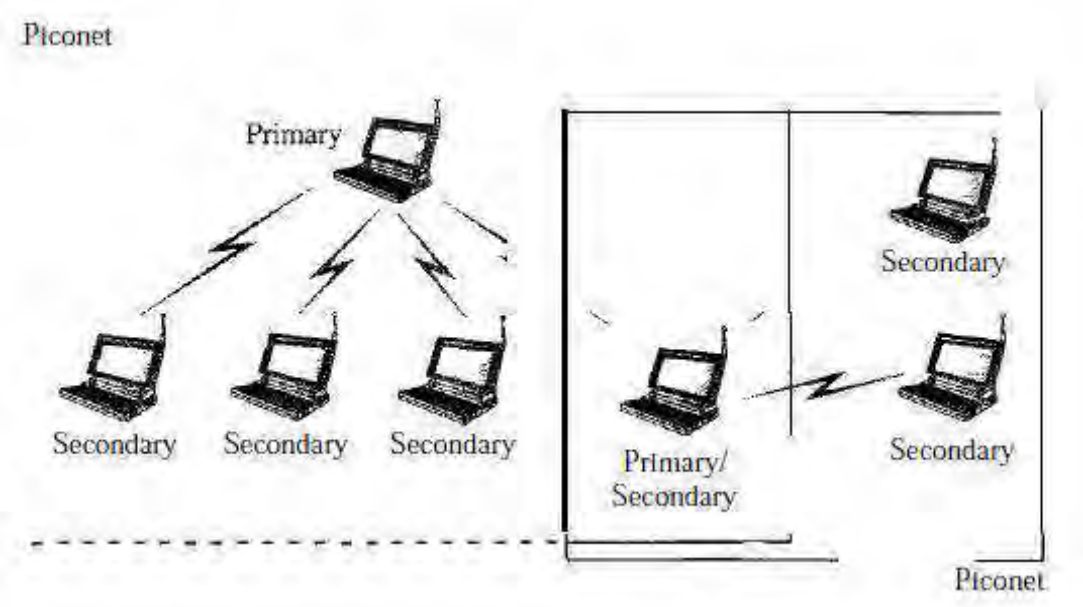



Except a maximum of seven secondaries, an additional eight secondaries that can have, will be in the ***parked state***. A secondary in a parked state is synchronized with the primary, but cannot take part in communication until it is moved from the parked state. Because only eight stations can be active in a piconet, activating a station from the parked state means that an active station must go to the parked state.

**SCATTERNET**

Scatternet is the combination of piconets. A secondary station in one piconet can be the primary in another piconet. This secondary station can receive messages from the primary in the first piconet and, acting as a primary for other piconet. A station can be a member of two piconets. Figure below illustrates a scatternet.

Figure: *Scatternet*

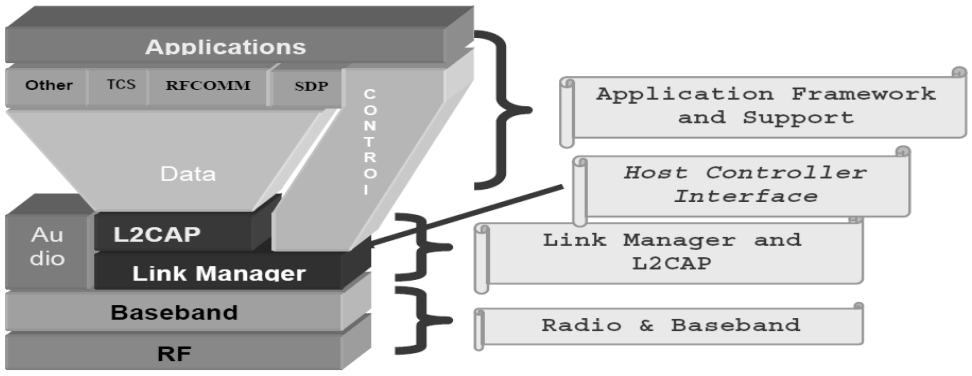



***BLUETOOTH DEVICES***

A Bluetooth device has a built-in short-range radio transmitter. The current data rate of this technology is 1 Mbps with a 2.4-GHz bandwidth. This means that there is a possibility of interference between the IEEE 802.11b wireless LANs and Bluetooth LANs.

**BLUETOOTH LAYERS**

Bluetooth system has several layers that do not exactly match those of the Internet model. All the layers function one after the other. One layer comes into play only after the data has been through the previous layer.



**RADIO:** The radio layer is equivalent to the physical layer of the Internet model. Bluetooth devices are low-power and have a range of 10 m. It is the lowest layer in the Bluetooth protocol stack. Bluetooth uses a technique called **frequency hopping**.
**TRANSMITTER CHARACTERISTICS:** Each device is classified into 3 power classes, Power Class **1**, **2** and **3**.

**Power Class 1**: is designed for long range (~100m) devices, with a max output power of 20 dBm (Decibel-milliwatt). **Power Class 2:** for ordinary range devices (~10m) devices, with a max output power of 4 dBm**. Power Class 3:** for short range devices (~10cm) devices, with a max output power of 0 dBm.

**BASEBAND LAYER:** The baseband layer is roughly equivalent to the MAC sub layer in LANs. The baseband is the digital engine of a Bluetooth system. It is responsible for constructing and decoding packets, encoding and managing error correction, encrypting and decrypting for secure communications, calculating radio transmission frequency patterns, maintaining synchronization, controlling the radio, and all of the other low level details necessary to realize Bluetooth communications. Bluetooth uses a 2.4-GHz ISM band divided into 79 channels of 1 MHz each. Its access method to access channel is TDMA. Channel is accessed by using time slot, and during the time one frequency is used. This layer describes the specification of the Bluetooth Link Controller (LC), which carries out the baseband protocols and other low-level link routines. It specifies Piconet/Channel definition, where primary and secondary communicate with each other. Note that the communication is only between the primary and a secondary, secondary cannot communicate directly with one another.

**LMP:** The **Link Manager Protocol** (LMP) is used by the Link Managers (on either side) for link set-up and control. Link Manager is responsible for managing the physical details for Bluetooth connections. It is responsible for creating the links, monitoring their health, and terminating them gracefully upon command or failure. The link manager is implemented in a **mix of hardware and software**. The Link Manager carries out link setup, authentication, link configuration and other protocols

**HCI:** The **Host Controller Interface** (HCI) provides a command interface to the Baseband Link Controller and Link Manager, and access to hardware status and control registers. This is the layer of the stack that contains the firmware i.e. the software that actually controls all the activities happening in the Baseband and Radio layers. It provides a common interface between the Bluetooth host and a

Bluetooth module. It manages the hardware links with the scatternets. It also contains the drivers for the hardware devices used in the connection. Basically the BIOS is loaded in the HCI Layer.

**L2CAP: Logical Link Control and Adaptation Protocol** (L2CAP) supports higher level protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service information. It is roughly equivalent to the LLC sub layer in LANs. The L2CAP has specific duties**: multiplexing, segmentation and reassembly, quality of service (QOS):** it will do its best under the circumstances, **and group management:** it manage group of devices as part of multicast group to receive data**.** Another functionality of L2CAP is to allow devices to create a type of logical addressing between themselves.

**RFCOMM**: The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10.

**SDP:** The Service Discovery Protocol (SDP) provides a means for applications to discover, which services are provided by a Bluetooth device. It also allows applications to determine the characteristics of those available services.

*LECTURE NOTE: 22*

# FRAME RELAY AND ATM

Frame relay is an affordable way to transmit switched-packet data within LANs and WANs.

Frame Relay is a high-speed protocol that can provide some services not available in other WAN technologies such as DSL, cable TV, and T lines. It is a packet-switching telecommunication service designed for cost-efficient data transmission for intermittent traffic between local area networks **(LANs**) and between endpoints in wide area networks (**WANs**). It puts data in a variable-size unit called a ***frame*** and leaves any necessary error correction (retransmission of data) up to the endpoints, which speeds up overall data transmission. For most services, this network provides a permanent virtual circuit (**PVC**), which means that there is a continuous, dedicated connection without having to pay for a full-time leased line, while the service provider figures out the route of frame travel to its destination and can charge based on usage. Switched virtual circuits (SVC), by contrast, are temporary connections that are destroyed after a specific data transfer is completed. Frame relay transmits packets at the **data link layer** of the Open Systems Interconnection (**OSI**) model rather than at the **network layer.**

ATM technology originally designed as a WAN technology now can also use in LAN technology.

Prior to Frame relay organizations used other protocol as X.25 as a virtual-circuit switching network. But X.25 has problems of low data rate, high amount of flow and error control, high overhead, costly, which is not necessary in networks using a reliable transmission medium.

Frame relay has the following properties to overcome the above drawbacks as follows:

1. Frame relay operates at a higher speed (1.544 Mbps and recently 44.736 Mbps). This means that it can easily be used instead of a mesh of T1 or T3 lines.
2. Frame relay operates in the physical and data link layers. Thus it can be easily used in backbone networks.
3. Frame relay allows bursty data.

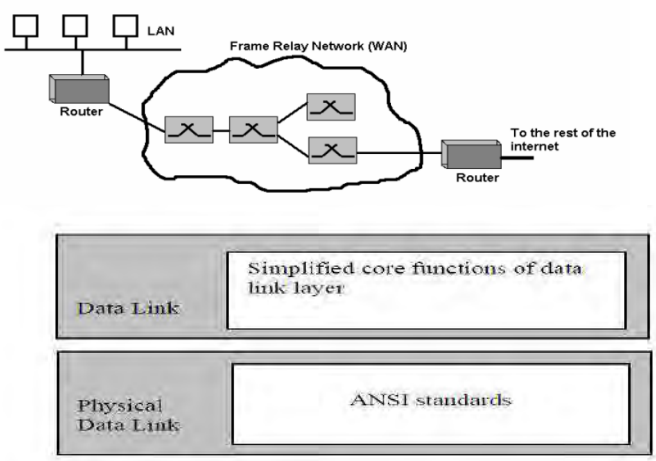
4. Frame relay allows a frame size of 9000 bytes, which can accommodate all local area network frame sizes.
5. Frame relay is less expensive than other traditional WANs.
6. Frame relay has error detection at the data link layer only. It does not have any flow control or retransmission policy, if a frame is damaged, it is silently dropped.

One of the nice features of Frame Relay is that it provides congestion control and quality of service (QoS).

## ARCHITECTURE

Frame Relay provides permanent virtual circuits and switched virtual circuits. **Figure** below shows an example of a Frame Relay network connected to the Internet. The routers are to connect LANs and WANs in the Internet.

A **virtual circuit** in a Frame Relay is identified by a number called a **data link connection identifier (DLCI)**. The frame relay network consists of switches each of which has a table to route frames. The table matches an incoming **port-DLCI** combination with an **outgoing-DLCI** combination.




VCIs in Frame Relay are called DLCIs.

***PERMANENT VIRTUAL CIRCUITS***

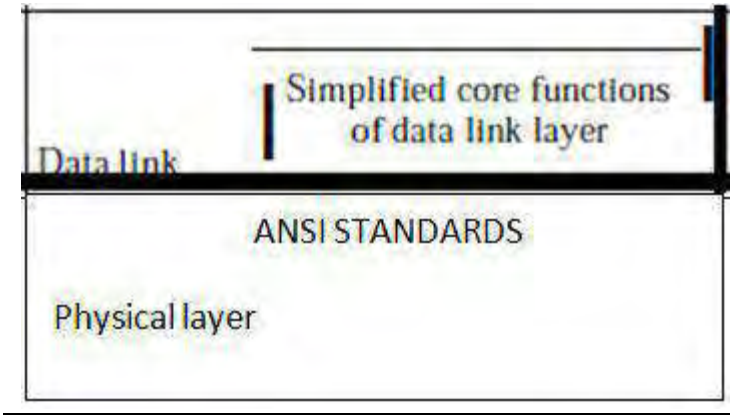
A source and a destination may choose to have a permanent virtual circuit (PVC). In this case, corresponding table entry is recorded for all switches by the administrator (remotely and electronically). An outgoing DLCI is given to the source, and an incoming DLCI is given to the destination. But PVC connections have two drawbacks. First, they are costly because two parties pay for the connection all the time even when it is not in use. Second, a connection is created from one source to one single destination. For one source and multiple destinations it needs PVC for each connection. An alternate approach is the switched virtual circuit (SVC). The SVC creates a temporary, short connection that exists only when data are being transferred between source and destination.

***SWITCHES***

Switches in a Frame Relay network are used to route frames. The table it uses, matches an incoming port-DLCI combination with an outgoing port-DLCI.

**FRAME RELAY LAYERS**

Frame Relay has only physical and data link layers.

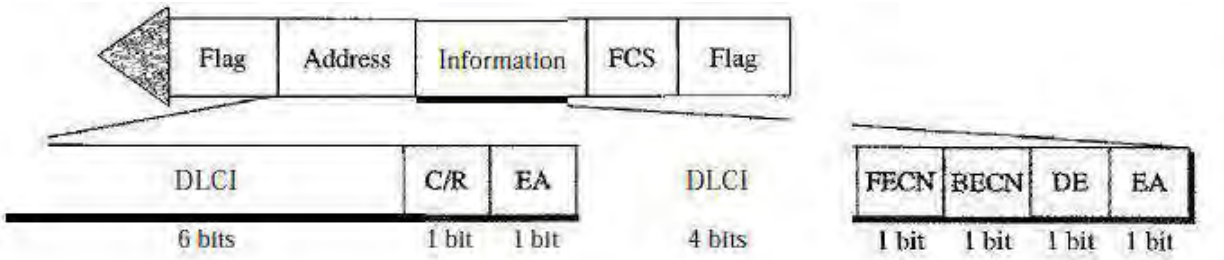



**PHYSICAL LAYER**

No specific protocol is defined for the physical layer in frame relay. Frame Relay supports any of the protocols recognized by ANSI.

**DATA LINK LAYER**

Frame relay at data link layer uses a simple protocol like HDLC that does not support flow or error control. It only has an error detection mechanism. Its frame is similar to the HDLC frame, but the control field is missing.




**Address (DLCI) field**. It is of 6 bits. The second part of the DLCI uses the first 4 bits of the second byte. These bits are part of the 10-bit data link connection identifier defined by the standard. The address field defines the DLCI as well as some bits used to control congestion and traffic. This value (10 bits) represents the virtual connection between DTE devices and the switch.

**COMMAND/RESPONSE (C/R):** The C/R is the bit that follows the most significant DLCI byte in the address field. The C/R bit is provided to identify a frame as either a command or response frame. It is not used by the Frame Relay protocol.

**EXTENDED ADDRESS (EA)**

To increase the range of DLCIs, the Frame Relay address has been extended from the original

2-byte address to 3- or 4-byte addresses. The EA is used to indicate whether the byte in which the EA value is 1 is the last addressing field. **Note** that the EA field defines the number of bytes i.e it is 1 in the last byte of the address, and it is 0 in the other bytes.In the 3- and 4- byte address formats the bit before the last bit is set to 0. This bit (3 bit) indicates whether the current byte is the final byte or octets of the address. An EA of 0 means that another address is to follow. The EA of 1 means that the current byte is the final one or last DLCI octet.

| DLCI | | | C/R | EA = 0 |
|---|---|---|---|---|
| DLCI | FECN | BECN | DE | EA = 1 |

a. Two-byte address (lO-bit DLCI)

| DLCI | | C/R | EA=O |
|---|---|---|---|
| OLCI IFECNIBECN | | DE | EA=O |
| DLCI | | 0 | EA= I |

b. Three-byte address (16-bit DLCI)

| OLCI ICIR | EA=O |
|---|---|
| OLCI IPECN IBECN I OE | EA=O |
| DLCI | EA=O |
| DLCI I 0 | EA= 1 |

c. Four-byte address (23-bit DLCI)

**FORWARD EXPLICIT CONGESTION NOTIFICATION (FECN):** this field is of 3 bit used for congestion notification mechanism**.** This bit can be set by any switch to indicate that traffic is congested in the direction that the frame is traveling. This bit informs the destination that congestion has occurred.

**DISCARD ELIGIBILITY (DE)**

The DE bit (3 bit) indicates the priority level of the frame. In emergency situations, switches may have to discard frames to relieve the bottlenecks. The network may discard a frame whose DE bit is set. This bit may be set either by the sender of the frames or by any switch in the network.

Frame Relay does not provide flow or error control; they must be provided by the upper-layer protocols.

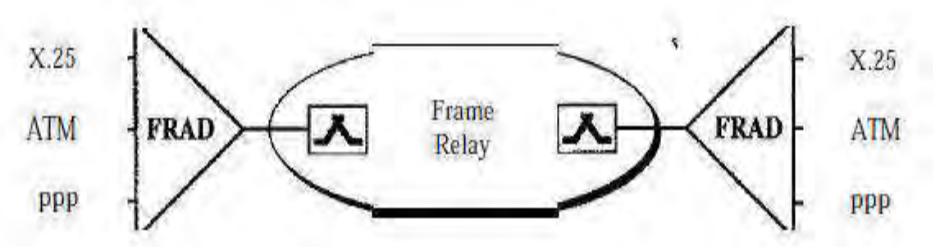
**FLAGS:** Delimits the beginning and end of the frame. The value of this field is always the same and is represented either as the hexadecimal number 7E or as the binary number 01111110.

**DATA**: It contains encapsulated upper layer data. Each frame in this variable length field includes a user data or payload field that varies in length up to 16,000 octets.

**FRAME CHECK SEQUENCE:** It ensures the integrity of transmitted data. This value is computed by the source device and verified by receiver to ensure integrity of data in the transmission.

**FRAD (**Frame Relay assembler/disassemble**)**

It is a device used to handle frames arriving from other protocols Frame**.** FRAD can be implemented as a separate device or as a part of a switch



**VOFR**

Frame Relay networks offer a **Voice over Frame Relay** to sends voice through the network. Here voice is digitized using PCM, compressed and sent as data frames over the network. But the quality of voice is not as good as voice over a circuit-switched network such as the telephone network.

**LMI**

**Local Management Information (LMI)** is a protocol added to the Frame Relay protocol to provide management functions. It is also designed to provide PVC (permanent virtual circuit) connections. LMI can provide its feature as:

- A keep-alive mechanism to check if data are flowing.
- A multicast mechanism to allow a local end system to send data frame to more than one remote end system.
- A mechanism to allow an end system to check the status of a switch (e.g., to see if
  the switch is congested).

---

Frame Relay operates only at the physical and data link layers.

---

*LECTURE NOTE: 23*

# ATM

**ATM** is a high-speed networking standard (i.e International Telecommunication Union-Telecommunications Standards Section (ITU-T)) designed to support both voice and data communications in small fixed size cells. It is a network technology based on transferring data in ***cells*** or ***packets*** of a fixed size. It is normally utilized by Internet service providers on their private long-distance networks. It is operates at the data link layer (Layer 2 in the OSI model) over either fiber or twisted-pair cable. ATM networks are connection-oriented.

The cell used with ATM is relatively smaller than other older technologies. The small, constant cell size allows ATM equipment to transmit video, audio, and computer data over the same network, and assure that no single type of data hogs the line. ATM technology is designed to improve utilization and quality of service (QoS) on high-traffic networks. So without routing and with fixed-size cells, networks can easily manage bandwidth under ATM than under Ethernet.

## BENEFITS OF ATM

- **Dynamic bandwidth for bursty traffic:** mean**s** application needs and delivering high utilization of networking resources. Most applications are or can be viewed as inherently bursty, for example voice is bursty, as both parties are neither speaking at once nor all the time; video is bursty, as the amount of motion and required resolution varies over time.
- **Smaller header** with respect to the data to make the efficient use of bandwidth.
- **Can handle Mixed network traffic very efficiently:** Variety of packet sizes makes traffic unpredictable. All network equipments should incorporate elaborate software systems to manage the various sizes of packets. ATM handles these problems efficiently with the fixed size cell.
- **Cell network:** All data is loaded into identical cells that can be transmitted with complete predictability and uniformity.
- **Class-of-service support** for multimedia traffic allowing applications with varying throughput and latency requirements to be met on a single network.
- **Scalability in speed** and network size supporting link speeds of T1/E1 to OC–12 (622 Mbps). **Common LAN/WAN architecture** allowing ATM to

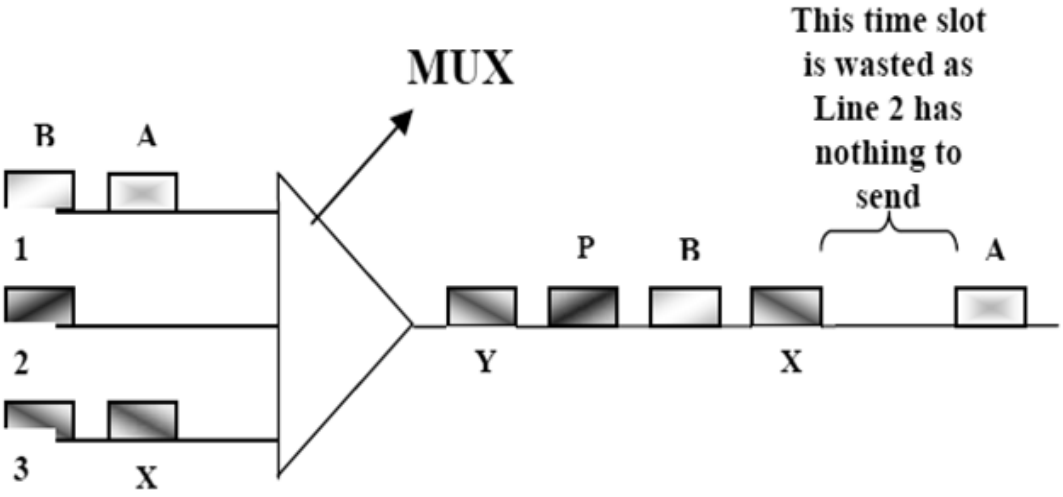
be used consistently from one desktop to another; traditionally, LAN and WAN technologies have been very different, with implications for performance and interoperability. But ATM technology can be used either as a LAN technology or a WAN technology.

- **International standards compliance** in central-office and customer-premises environments allowing for multivendor operation.

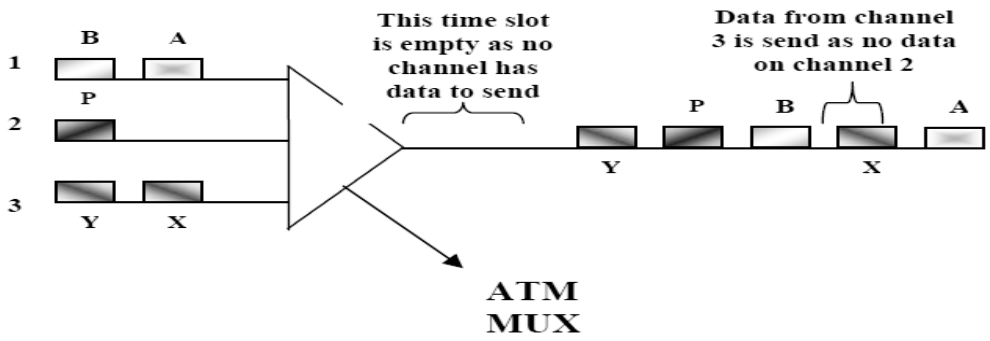
**ATM DEVICES AND THE NETWORK ENVIRONMENT**

ATM is a cell-switching and multiplexing technology that combines the benefits of circuit switching (guaranteed capacity and constant transmission delay) with those of packet switching (flexibility and efficiency for intermittent traffic). It provides scalable bandwidth from a few megabits per second (Mbps) to many gigabits per second (Gbps). Because of its asynchronous nature, ATM is more efficient than synchronous technologies, such as time-division multiplexing (TDM).

With TDM, each user is assigned to a time slot, and no other station can send in that time slot as shown in Fig below. If a station has much data to send, it can send only when its slot turn comes up, even if all other time slots are empty. However, if a station has nothing to transmit when its time slot comes up, the time slot is sent empty and is **wasted**.




Because ATM is asynchronous, time slots are available on demand with information identifying the source that contained in ATM cell header. **Figure below** shows how cells from 3 inputs have been multiplexed. At the first clock tick input 2 has no data to send, so multiplexer fills the slot with the cell from third input. When all cells from input channel are multiplexed then output slot are empty.




**ATM Devices**

An *ATM network* is made up of an *ATM* ***switch*** and *ATM* ***endpoints***. An ATM switch is responsible for cell transit through an ATM network. The job of an ATM switch is well defined. It accepts the incoming cell from an ATM endpoint or another ATM switch. It then reads, updates the cell header information and quickly switches the cell to an output interface towards its destination. An ATM endpoint (or end system) contains an ATM network interface adapter. Examples of ATM endpoints are workstations, routers, digital service units (DSUs), LAN switches, and video coder-decoders (Codec's).

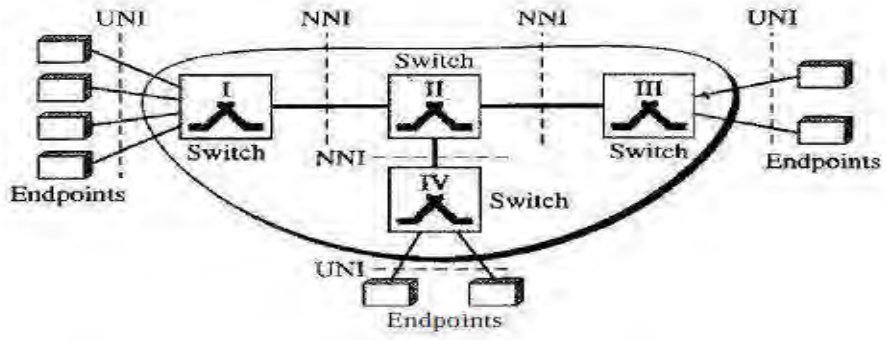
**ATM Network Interfaces**

An ATM network consists of a set of ATM switches interconnected by point-to-point ATM links or interfaces. ATM switches support two primary types of interfaces: UNI and NNI. The **UNI (User-Network Interface)** connects ATM end systems (such as hosts and routers) to an ATM switch. UNI can be public and private. The public UNI is for connection of end-user equipment to a public ATM network. The private UNI is for use within a single organization's premises or for a private network. The **NNI (Network-Network Interface)** connects two **ATM switches**. UNI and NNI can be further subdivided into public and private UNIs and NNIs. A private UNI connects an ATM endpoint and a private ATM switch. Its public counterpart connects an ATM endpoint or private switch to a public switch. A private NNI connects two ATM switches within the same private organization. A public one connects two ATM switches within the same public organization.

## ARCHITECTURE

The basic data unit in an ATM network is called a **cell**. A cell is only 53 bytes long with 5 bytes allocated to the header and 48 bytes carrying the payload (user data may be less than 48 bytes)
Like Frame Relay, ATM uses two types of connections: PVC and SVC.
In **PVC** its name itself imply that a **permanent virtual-circuit connection** is established between two endpoints by the network provider.

**SVC (switched virtual circuit connection):** here each time an endpoint wants to make a connection with another endpoint, a new virtual circuit must be established. ATM
needs the network layer addresses and the services of another protocol (such as IP) to perform this job.




## SWITCHING

ATM uses switches to route the cell from a source endpoint to the destination endpoint. A switch routes the cell using both the VPls and the VCls.

## ATM LAYERS

The ATM standard defines three layers. They are, from top to bottom, the **application**
**Adaptation layer(AAL), the ATM layer, and the physical layer**. For more details please refer text book.

### *APPLICATION ADAPTATION LAYER*

The application adaptation layer (AAL) was designed to enable two ATM concepts.

## PHYSICAL LAYER

ATM cells can be carried by any physical layer carrier.

### *ATM Layer*

The ATM layer provides routing, traffic management, switching, and multiplexing services. It processes outgoing traffic by accepting 48-byte segments from the AAL sub layers and transforming them into 53-byte cells by the addition of a 5-byte header

ATM uses two formats of header, one for user-to-network interface (UNI) cells and another for network-to-network interface (NNI) cells.

*MODULE-III*

*LECTURE NOTE: 24*

# NETWORK LAYER

The data link layer oversees the delivery of the packet between two systems on the same network (links) whereas, the network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links) as well as this layer ensures that each packet gets from its point of origin to its final destination. The network layer adds a header that includes the logical addresses (IP addressing) of the sender and receiver to the packet corning from the upper layer, which help to travel a packet through Internet: where Internet is the most common example of internetworks(i.e network of networks). One of the important functions of the network layer is to provide a routing mechanism.

The network layer is responsible for the delivery of individual packets from the source to the destination host.

## LOGICAL ADDRESSING

Communication at the network layer is host-to-host (computer-to-computer); where a computer is somewhere in the world needs to communicate with another computer somewhere else in the world or this communication is through the Internet. The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer. For this type of communication a global addressing is required called logical addressing. Today we use **IP address (Internet protocol address) is also called Internet address** to mean a logical address in the network layer. The most common protocols in this layer are IPX(Internetwork packet Exchange) and IP(Internet protocol). IPX was predominantly used on novell networks, but is mostly deprecated. Now a days IP is mostly used and its two versions are:

1. IPv4 (32-bits)
2. IPv6 (128-bits)

Where IP was developed by the department of Defence (DoD) during 1970's. It was included in group of protocols known as TCP/IP suite. The most commonly and being currently used among twos is IPv4,

which contain 32 bits internet address length, this may divided into 4 octets, each of 8 bits. It can gives maximum $2^{32}$ addresses. The other concern about IP layer is IPv6 which is the motivated new design; contain 128-bits addresses that gives much greater flexibility in a address allocation. But IPv6 addresses, which may become dominant in the future.

## IPv4 ADDRESSES

Things related to the IP address are:

HOST ID
NETWORK ID
NETWORK ADDRESS
SUBNETTING/SUPERNETTING
BROADCAST ID

- An **IPv4** address of 32-bits long address is ***uniquely (unique throughout the world)*** and ***universally*** defines the connection of a device (for example, a computer or a router) to the Internet. IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet and addressing system must be accepted by any host that wants to be connected to the Internet. Two devices on the Internet can never have the same address at the same time. But if devices have *m* connections to the Internet, it needs to have *m* addresses. Other way by using some strategies, an address may be assigned to a device for a time period and then taken away and assigned to another device.

The IPv4 addresses are unique and universal.

### Address Space

An address space is the total number of addresses used by the protocol. IPv4 protocol that defines addresses has an address space. If a protocol uses *N* bits to define an address, the address space is $2^N$ because each bit can have two different values (0 or 1) and *N* bits can have $2^N$ values. The IPv4 uses 32-bit addresses, which means that the address space is $2^{32}$ or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there

were no restrictions, more than 4 billion devices could be connected to the Internet.

---

The address space of IPv4 is $2^{32}$ or 4,294,967,296.

**NOTATIONS**

There are two prevalent notations to show an IPv4 address such as:

1) Binary notation
2) Dotted decimal notation.

**BINARY NOTATION**

- In binary notation, the IPv4 address is displayed as 32 bits.
- Each octet is often referred to as a byte. So it is common to referred IPv4 address as a 32-bit address or a 4-byte address.
- The following is an example of an IPv4 address in binary notation:
  01110101 10010101 00011101 00000010

**DOTTED-DECIMAL NOTATION**

- To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes.

  The following is the dotted-decimal notation of the address in binary form:

*Dotted-decimal notation and binary notation for an IPv4 address*


***Example1***

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111

**Solution**

We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

a. 129.11.11.239
b. 193.131.27.255

*Example2*
Change the following IPv4 addresses from dotted-decimal notation to binary notation.
a. 111.56.45.78
b. 221.34.7.82

Solution
We replace each decimal number with its binary equivalent.
a. 01101111 00111000 00101101 01001110
b. 11011101 00100010 00000111 01010010

To represent the address in decimal form user must be noted that
a. There must be no leading zero (045).
b. There can be no more than four numbers in an IPv4 address.
c. Each number needs to be less than or equal to 255 (301 is outside this range).
d. A mixture of binary notation and dotted-decimal notation is not allowed.

For example: a) 111.56.045.78
b) 221.34.7.8.20 (more than four numbers)
c) 75.45.301.14
d) 11100010.23.14.67 (mixed)

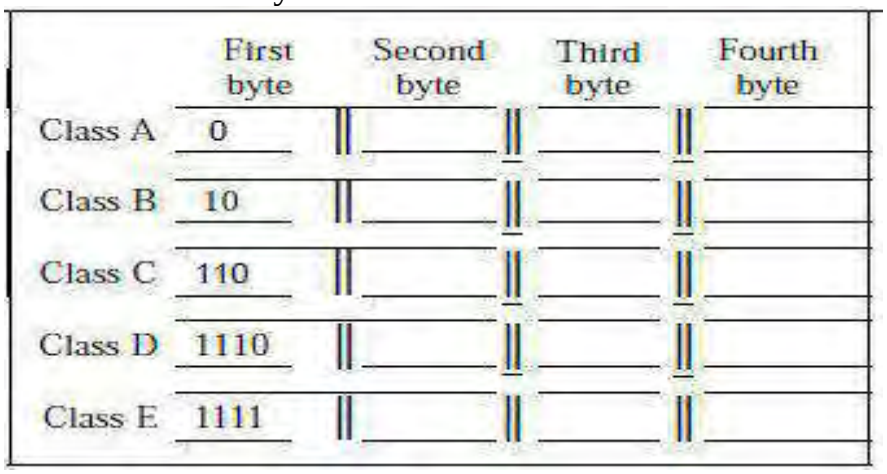
## CLASSFUL ADDRESSING

In the classful addressing system all the IP addresses or address spaces that are available are divided into the five classes A,B,C,D and E, in which class A,B and C address are frequently. Class D is for Multicast and is rarely used and class E is reserved and is not currently used. Each class occupies some part of the address space. Each of the IP address belongs to a particular class that's why they are **classful addresses**. Previously this addressing system did not have any name, but when classless addressing system came into existence then it is named as Classful addressing system. The main **disadvantage** of classful addressing is that it limited the flexibility, number of addresses that can be assigned to any device and it does not send subnet information but it will send the complete network address.

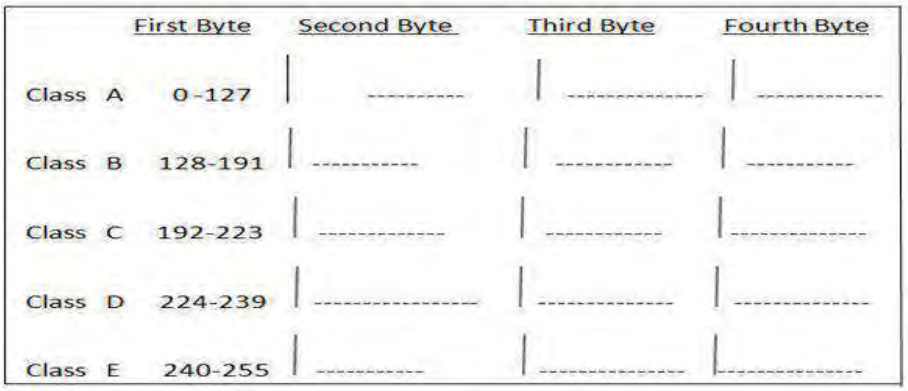
The router will supply its own subnet mask based on its locally configured subnets. But as long as we have the same subnet mask and the network is contiguous, we can use subnets of a classful network address.

IPv4 addressing used the concept of classes. We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class of the address. If the address is given in decimal-dotted notation, the first byte defines the class. Both the methods are shown below:

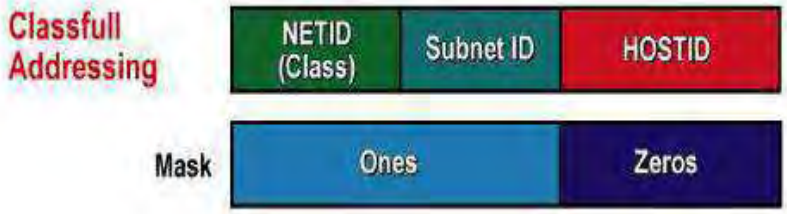
Binary notation



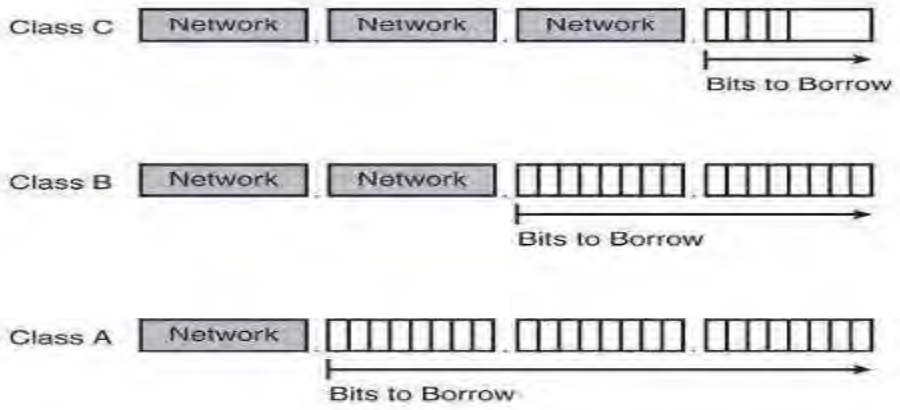

**Dotted decimal notation**




But from the above range **0** is reserved and represents all IP addresses, **127** is a reserved address and is used for loop back testing, **255 is a reserved address** and is used for broadcasting purposes.




Examples of classful addressing:




The **network number** uniquely identifies a segment in the network and a **host number** uniquely identifies a device on a segment. The combination of these two numbers must be unique throughout the entire network.

***CLASSES AND BLOCKS***

One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size as shown in the below Table.

Number of blocks and block size of each class

| *Class* | *Number ofBlocks* | *Block Size* | *Application* |
|---|---|---|---|
| A | 128 | 16,777,216 | Unicast |
| B | 16,384 | 65,536 | Unicast |
| C | 2,097,152 | 256 | Unicast |
| D | 1 | 268,435,456 | Multicast |
| E | 1 | 268,435,456 | Reserved |

Previously, when an organization requested a block of addresses, it was granted one in class A, B, or C. Class A addresses were designed for large organizations with a large number of attached hosts or routers. Class B addresses were designed

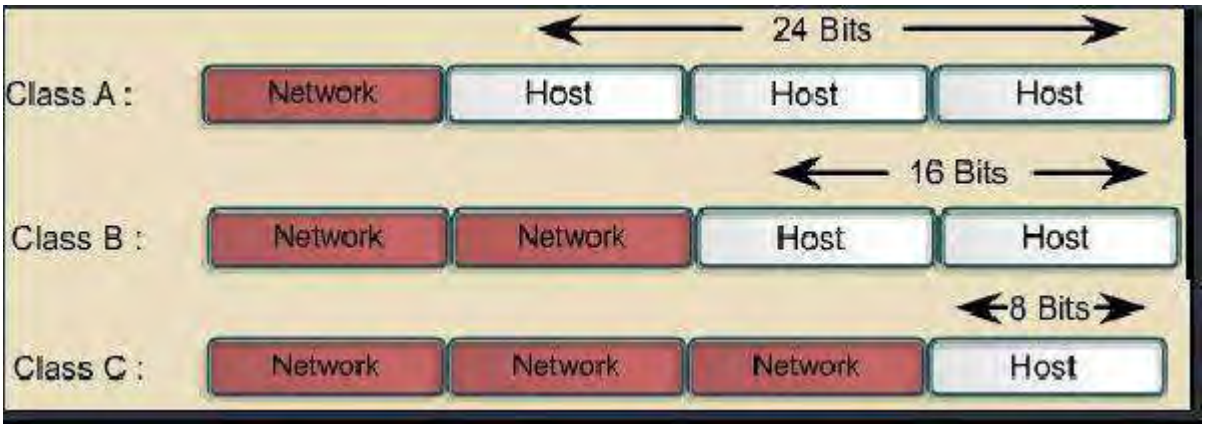
for midsize organizations with tens of thousands of attached hosts or routers. Class C addresses were designed for small organizations with a small number of attached hosts or routers.

From the table we can see the flaw in this design. A block in class A address is too large for almost any organization. This means most of the addresses in class A were wasted and were not used. A block in class B is also very large, probably too large for many of the organizations that received a class B block. A block in class C is probably too small for many organizations. Class D addresses were designed for multicasting. Each address in this class is used to define one group of hosts on the Internet. The Internet authorities wrongly predicted a need for 268,435,456 groups. But this is never happened and many addresses were wasted here too. And lastly, the class E addresses were reserved for future use; only a few were used, resulting in another waste of addresses.

In classfnl addressing, a large part of the available addresses were wasted.

**NETID AND HOSTID**

All IP addresses have a network and host portion. Classful addressing divides an IP address into the Network and Host portions along octet boundaries. These are represented as: net id or network id and host id or network number and host number.




In classful addressing, an IP address in class A, B, or C is divided into netid and hostid are of varying lengths, depending on the class of the address. Figure shows some netid and hostid bytes. **Note** that the concept does not apply to classes D and E.

In class A, one byte defines the netid and three bytes define the hostid. In class B, two bytes define the netid and two bytes define the hostid. In class C, three bytes define the netid and one byte defines the hostid.

**MASK**

Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous 1s and followed by contiguous as. The masks for classes A, B, and C are shown in the below table and this concept does not apply to classes D and E.

*Default masks for classful addressing*

| class | Binary | | | Dotted-decimal | CIDR |
|---|---|---|---|---|---|
| A | 11111111 00000000 | 00000000 | 00000000 | 255.0.0.0 | /8 |
| B | 11111111 00000000 | 11111111 | 00000000 | 255.255.0.0 | /16 |
| C | 11111111 00000000 | 11111111 | 11111111 | 255.255.255.0 | /24 |

The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid. Similarly for class B and class C The last column of Table above shows the mask in the form " */n*" where *n* can be 8, 16, or 24 in classful addressing. This notation is also called **slash notation** or **Classless Interdomain Routing (CIDR) notation**. The notation is used in classless addressing can also be applied to classful addressing, Where classful addressing is a special case of classless addressing.

**SUBNETTING**

Subnetting is the strategy used to partition a single physical network into more than one smaller logical sub-network (subnets). During the era of classful addressing, subnetting was introduced. Subnets are designed by accepting bits from the IP address's host part and using these bits to assign a number of smaller sub-networks inside the original network. Subnetting allows an organization to add sub-networks without the need to acquire a new network number via the Internet

service provider (ISP). Subnets were initially designed for solving the shortage of IP addresses over the Internet. Each IP address consists of a subnet mask. All the class types, such as Class A, Class B and Class C include the subnet mask known as the **default subnet mask**. The **subnet mask** determines the type and number of IP addresses required for a given local network. The firewall or router is called the default gateway. The default subnet mask is as follows as in the above table:

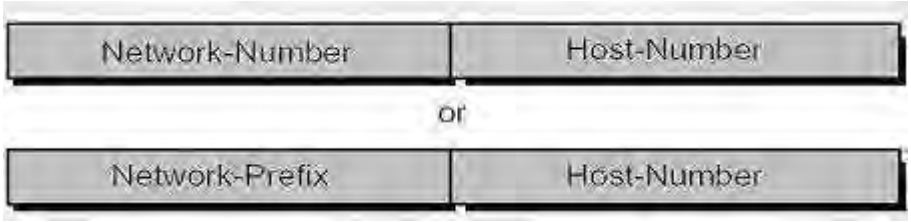
- Class A: 255.0.0.0
- Class B: 255.255.0.0
- Class C: 255.255.255.0

The subnetting process allows the administrator to divide a single Class A, Class B, or Class C network number into smaller portions. The subnets can be subnetted again into sub-subnets.

Dividing the network into a number of subnets provides the following benefits:

- Reduces the network traffic by reducing the volume of broadcasts
- Helps to surpass the constraints in a local area network (LAN), for example, the maximum number of permitted hosts.
- Enables users to access a work network from their homes; there is no need to open the complete network.

If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbors. Subnetting increases the number of 1s in the mask.




This level is now known as Prefix and Host where all hosts (individual computers) in the network share the prefix (subnet mask).

**SUBNET MASK**

A subnet mask is a 32-bit number used to differentiate the network component of an IP address by dividing the IP address into a network address and host address. Like the IP address, a subnet mask is written using the "dotted-decimal" notation.

Subnet masks are used to design subnetworks, or subnets, that connect local networks. It determines both the number and size of subnets where the size of a subnet is the number of hosts that can be addressed. we can create a subnet mask by taking the 32-bit value of an existing IP address, choosing how many subnets you want to create or alternatively, how many nodes you need on each subnet, and then setting all subsequent network bits to "1" and host bits to "0". The resulting 32-bit value is our subnet mask.

In any given network, two host addresses are always reserved for special purposes. The "0" address becomes the network address or network identification and the "255" address is assigned as a broadcast address. **Note** that these cannot be assigned to a host.

**For example:** The subnet mask is the network address plus the bits reserved for identifying the subnetwork. Given

**Full address is:** 10010110.11010111.00010001.00001001

The Class B network part is: 10010110.11010111

The host address is: 00010001.00001001

If this network is divided into 14 subnets, then the first 4 bits of the host address (0001) are reserved for identifying the subnet.

The bits for the network address are all set to 1. In this case, therefore, the subnet mask would be 11111111.11111111.11110000.00000000. It's called a *mask* because it can be used to identify the subnet to which an IP address belongs. On performing bitwise AND operation on the mask and IP resulted sub network address i.e. 150.215.16.0

Subnet Mask: 255.255.240.0 11111111.11111111.11110000.00000000

IP address: 150.215.17.9 10010110.11010111.00010001.00001001

Subnet address: 150.215.16.0 10010110.11010111.00010000.00000000

## SUPERNETTING

Supernetting Combine several IP network addresses into one IP address and thus reduces the number of entries in a routing table. It is done in CIDR addressing as well as internal networks.

The time came when most of the class A and class B addresses were depleted. There ia a huge demand for midsize blocks. The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations. Even a midsize organization needed more addresses. then supernetting is the solution in an organization that combine several class C blocks to create a larger range of addresses. In other words, **several networks are combined to create a supernetwork or a supernet**. An organization can apply for a set of class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks.

The organization can then use these addresses to create one supernetwork. Supernetting decreases the number of 1s in the mask. For example, if an organization is given four class C addresses, the mask changes from /24 to /22. We will see that classless addressing eliminated the need for supernetting.

While supernetting, data bits are borrowed from the network ID and allocated to the host ID. A larger and more complicated network can block other routers from making topological changes, so a supernet improves convergence speed and enables a better and more stable environment. Supernetting requires the use of routing protocols that help to support CIDR. The other protocols - Interior Gateway Routing Protocol

## ADDRESS DEPLETION

The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses. We have run out of class A and B addresses, and a class C block is too small for most midsize organizations.

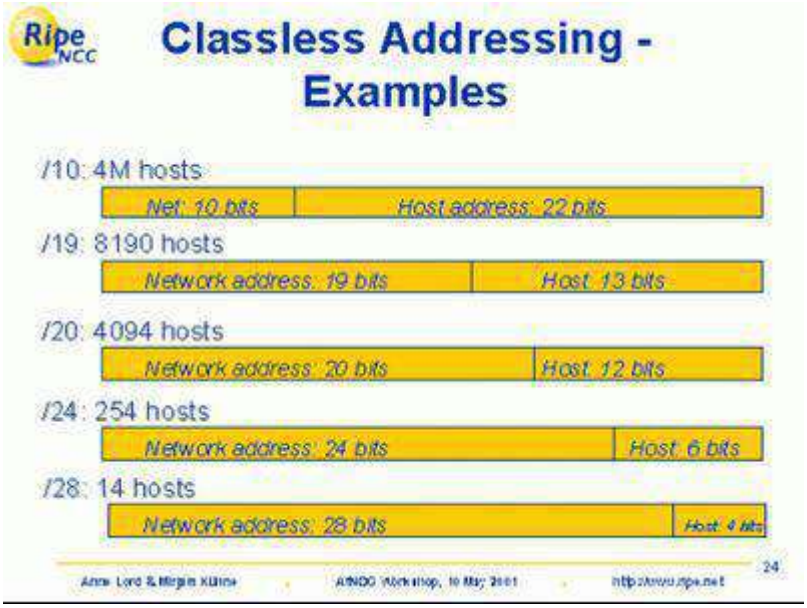
One solution that has alleviated the problem is the idea of classless addressing. Classful addressing, which is almost obsolete, is replaced with classless addressing.

## CLASSLESS ADDRESSING

Classless addressing can set the network boundary practically anywhere, thus breaking the classful limitations. Classless addressing treats the IP address as a 32 bit stream of ones and zeroes, where the boundary between network and host portions can fall anywhere between bit 0 and bit 31.

Classless addressing system is also known as **CIDR(Classless Inter-Domain Routing).** Classless addressing is a way to allocate and specify the Internet addresses used in inter-domain routing more flexibly than with the original system of Internet Protocol (IP) address classes. In classful addressing if any company needs more than 254 host machines but far fewer than the 65,533 host addresses then the only option for the company is to take the class B address. Now suppose company needs only 1000 IP addresses for its host computers then in this (65533-1000=64533) IP addresses get wasted. For this reason, the Internet was, running out of address space much more quickly than necessary before CIDR. CIDR effectively solved the problem by providing a new and more flexible way to specify network addresses in routers. A CIDR network address looks like this: **192.30.250.00/15**

Classless addressing is the preferred method for assigning IP addresses, mainly because it saves address space




To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

**Address Blocks**

In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity. For example, a household may be given only two addresses; a large organization may be given thousands of addresses.

An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

Restriction to simplify the handling of addresses, the Internet authorities imposes three restrictions on classless address blocks:

1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ... ).
3. The first address must be evenly divisible by the number of addresses.

**MASK**

A better way to define a block of addresses is to select any address in the block and the mask. As we discussed before, a mask is a 32-bit number in which the *n* leftmost bits are 1s and the 32 - *n* rightmost bits are 0s. However, in classless addressing the mask for a block can take any value from 0 to 32. It is very convenient to give just the value of *n* preceded by a slash (CIDR notation).

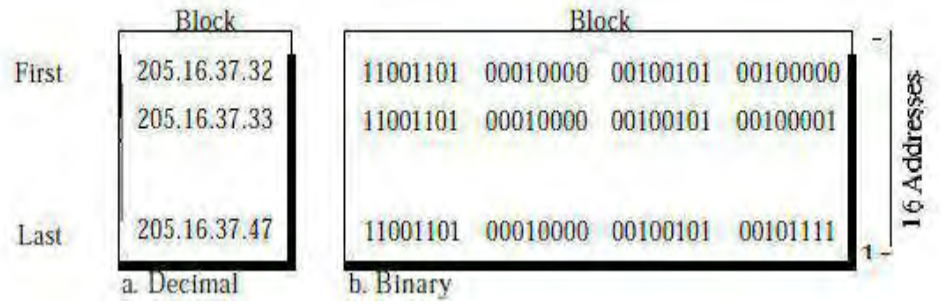
In 1Pv4 addressing, a block of addresses can be defined as x.y.z.t/n in which x.y.z.t defines one of the addresses and the */n* defines the mask. The address and the */n* notation completely define the whole block (the first address, the last address, and the number of addresses). The first address in the block can be found by setting the 32 - *n* rightmost bits in the binary notation of the address to 0s. And the last address in the block can be found by setting the rightmost 32 - *n* bits in the binary notation to 1s.

***For Example***
A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. What is the first address in the block?

**Solution**
The binary representation of the given address is 11001101 00010000 00100101 00100111. If we set 32 - 28 rightmost bits to 0, we get 11001101 00010000 00100101 00100000 or 205.16.37.32.

**Example 2**
Find the last address for the block in 205.16.37.39/28

**Solution**
The binary representation of the given address is 11001101 00010000 00100101 00100111. If we
Set 32 - 28 rightmost bits to 1, we get 11001101 00010000 00100101 0010 1111 or 205.16.37.47.
This is actually the block shown in the Figure below




**Number of Addresses**: The number of addresses in the block is the difference between the last and first address. **It can easily be found using the formula $2^{32-n}$.**

**PUBLIC ADDRESSES**

Public addresses are Class A, B, and C addresses that can be used to access devices in other public networks, such as the Internet.

**PRIVATE ADDRESSES**

Within the range of addresses for Class A, B, and C addresses are some reserved addresses, commonly called **private addresses**. Anyone can use private addresses; however, this creates a problem if you want to access the Internet. Each device in

the network (in this case, this includes the Internet) must have a unique IP address. If two networks are using the same private addresses, it would run into reach ability issues. To access the Internet, our source IP addresses must have a unique Internet public address. This can be accomplished through address translation. Here is a list of private addresses.

Class A: 10.0.0.0 - 10.255.255.255 (1 Class A network)
Class B: 172.16.0.0 - 172.31.255.255 (16 Class B networks)
Class C: 192.168.0.0 - 192.168.255.255 (256 Class C networks)

## STATIC /DYNAMIC

Each device in an IP network is either assigned a permanent address (static) by the network administrator or is assigned a temporary address (dynamic) via DHCP software. Routers, firewalls and proxy servers use static addresses where as Client machines may use static or dynamic IP addresses. In routers and operating systems, the default configuration for clients is dynamic IP

## DHCP

DHCP stands for Dynamic Host Configuration Protocol. Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway. This protocol assigns network IP addresses to clients on the network at start up. With DHCP, each client workstation does not need to be set up with a static IP address. DHCP is recommended on large networks to set up dynamically. It would be very time consuming to manually assign a static IP address to every workstation on your network. With static IP addressing, the IP address that you assign to a device never changes. A DHCP server contains a pool of IP addresses that it can draw from to assign to devices that are connecting to the network.

## NETWORK ADDRESS TRANSLATION (NAT)

NAT is the process where a network device, usually a firewall, assigns a public address to a computer (or group of computers) inside a private network. The main use of NAT is to limit the number of public IP addresses an organization or company must use, for both economy and security purposes. The most common form of network translation involves a large private network using addresses in a

private range **(10.0.0.0 to 10.255.255.255(total $2^{24}$), 172.16.0.0 to 172.31.255.255(total $2^{20}$), or 192.168.0 0 to 192.168.255.255(total $2^{16}$)).** The private addressing scheme works well for computers that only have to access resources inside the network, like workstations needing access to file servers and printers. Routers inside the private network can route traffic between private addresses with no trouble. However, to access resources outside the network, like the Internet, these computers have to have a public address in order for responses to their requests to return to them. This is where NAT comes into play. NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.

To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved as private address.

***LECTURE NOTE: 25***

# INTERNETWORKING

Internetworking is the process or technique of connecting different networks by using intermediary devices such as routers or gateway devices. The network layer or internetwork layer is responsible for host-to-host delivery and for routing the packets through the routers or switches. **Internetworking** is implemented in Layer 3 (Network Layer) of this model. The most notable example of internetworking is the Internet (capitalized). There are three variants of internetwork or **Internetworking**, depending on who administers and who participates in them:

**Extranet**
**Intranet**
**Internet**

Intranets and extranets may or may not have connections to the Internet. If connected to the Internet, the intranet or extranet is normally protected from being accessed from the Internet without proper authorization. The Internet is not

considered to be a part of the intranet or extranet, although it may serve as a portal for access to portions of an extranet.

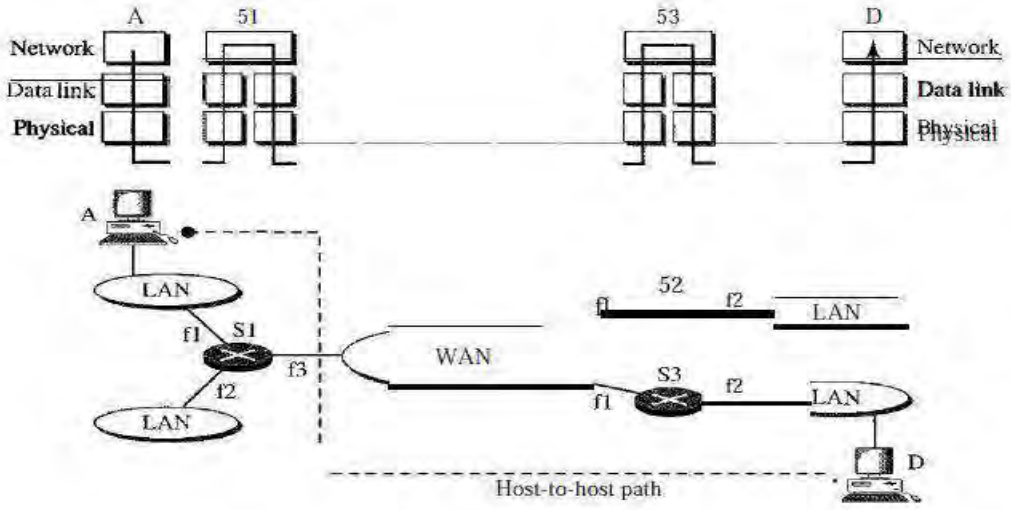



Figure above shows the general idea of the functionality of the network layer at a source, at a router, and at the destination. The network layer at the source is responsiblefor creating a packet from the data coming from another protocol (such as a transport layer protocol or a routing protocol). The header of the packet contains information, the logical addresses of the source and destination. The network layer is responsible for checking its routing table to find the routing information (such as the outgoing interface of the packet or the physical address of the next node). If the packet is too large, the packet is fragmented.

The switch or router is responsible for routing the packet. When a packet arrives, the router or switch consults its routing table and finds the interface from which the packet must be sent. The packet, after some changes in the header, with the routing information is passed to the data link layer again. The network layer at the destination is responsible for address verification. It makes sure that the destination address on the packet is the same as the address of the host. If the packet is a fragment, the network layer waits until all fragments have arrived, and then reassembles them and delivers the reassembled packet to the transport layer.

**Internet** as a Datagram Network

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

Switching at the network layer in the Internet uses the datagram approach to packet switching.

## INTERNET AS A CONNECTIONLESS NETWORK

Delivery of a packet can be accomplished by using either a connection-oriented or a connectionless network service. In a connection-oriented service, the source first makes a connection with the destination before sending a packet. When the connection is established, a sequence of packets from the same source to the same destination can be sent one after another. In this case, there is a relationship between packets. They are sent on the same path in sequential order. A packet is logically connected to the packet travelling before it and to the packet travelling after it. When all packets of a message have been delivered, the connection is terminated.

In a **connection-oriented** protocol, the decision about the route of a sequence of packets with the same source and destination addresses can be made only once, when the connection is established. Switches do not recalculate the route for each individual packet. This type of service is used in a virtual-circuit approach to packet switching such as in Frame Relay and ATM.

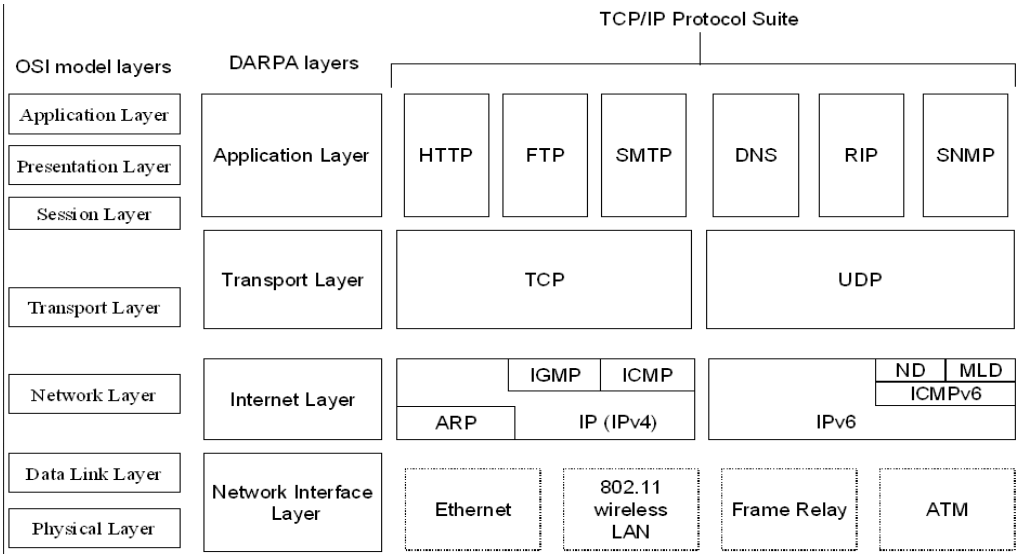
In **connectionless service**, the network layer protocol treats each packet independently, with each packet having no relationship to any other packet. The packets in a message may or may not travel the same path to their destination. This type of service is used in the datagram **approach to packet switching**. The Internet has chosen this type of service at the network layer. Because Internet is made of so many heterogeneous networks that it is almost impossible to create a connection from the source to the destination without knowing the nature of the networks in advance. So when we summarise the main functions performed by the network layer are as follows:

- Routing
- Congestion Control
- Internetworking

## IPv4 (Internet Protocol version 4)

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

**Figure** below shows the position of IPv4 in the suite.



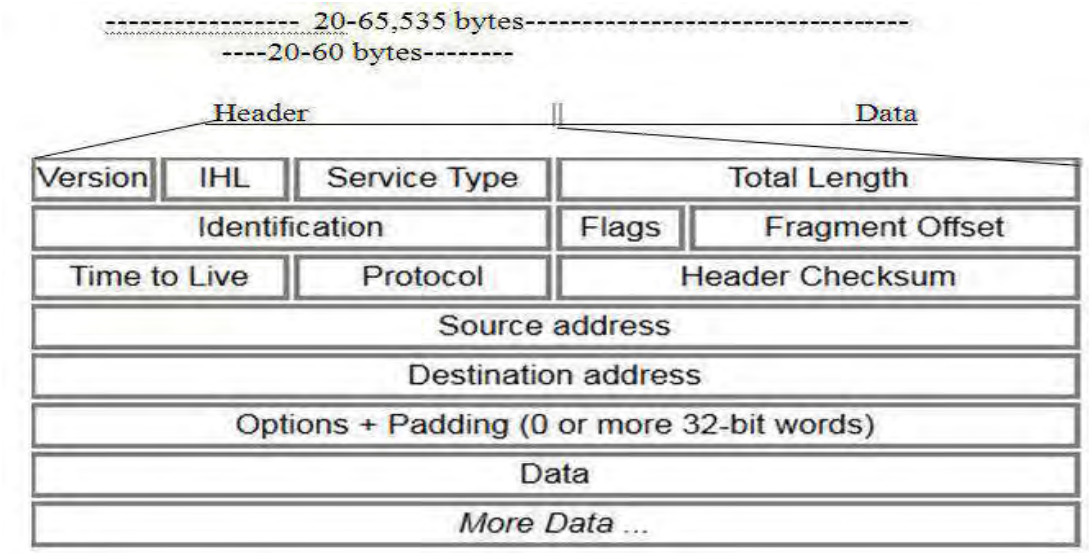

IPv4 is an unreliable and connectionless datagram protocol, a best-effort delivery service. The term ***best-effort*** means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. 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.

If reliability is important, IPv4 must be paired with a reliable protocol such as TCP may provides reliable to this protocol. IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach. This means that each datagram is handled independently, and each datagram can follow a different route to the destination. This implies that datagram's sent by the same source to the same destination could arrive out of order. Also, some could be lost or corrupted during transmission. Again, IPv4 relies on a higher-level protocol to take care of all these problems.

## DATAGRAM

Packets in the IPv4 layer are called datagram's. Figure below shows the IPv4 datagram format. A datagram is a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery. Internet Protocol being a layer-3 protocol (OSI)

takes data Segments from layer-4 (Transport) and divides it into packets. IP packet encapsulates data unit received from above layer and add to its own header information.




**Version:** Version (4 bytes) of the IP protocol which determines how to interpret the header. Currently the only permitted values are 4 (0100) or 6 (0110). This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4. The header format shown here is valid for IPv4 only.

**HL (header length):** Length of header as a number of 32-bit words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is
at its maximum size, the value of this field is 15 (15 x 4 = 60).

**Type of service:** service is now called differentiated services. This field is is often ignored by current routers but is meant to allow traffic to be prioritized (among other things).

**Total Length:** The length of the entire datagram including header and data: maximum permitted it 65,535 bytes or 64K. This is a in-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

Length of data = total length - header length

Since the field length is 16 bits, the total length of the IPv4 datagram is limited to 65,535 ($2^{16 - 1}$) bytes, of which 20 to 60 bytes are the header and the rest is data from the upper layer. The size of the IPv4 datagram may increase in the near future as the underlying technologies allow even more throughput (greater bandwidth).

**IDENTIFICATION, FLAGS AND FRAGMENT OFFSET**

These values allow datagram's to be fragmented for transmission and reassembled at the destination.

**Time to live:** An integer which is decremented at each router "hop"; supposed to be interpreted as a number of seconds but more often treated as a "hop count". If the value reaches zero the datagram is discarded and an ICMP message is sent to the source host. This field is used mostly to control the maximum number of hops (routers) visited by the datagram. When a source host sends the datagram, it stores a number in this field. This value is approximately 2 times the maximum number of routes between any two hosts. Each router that processes the datagram decrements this number by 1. If this value, after being decremented, is zero, the router discards the datagram. This field is needed because routing tables in the Internet can become corrupted.
A datagram may travel between two or more routers for a long time without ever getting delivered to the destination host. **This field limits the lifetime of a datagram**. Another use of this field is **to intentionally limit the journey of the packet**. For example, if the source wants to confine the packet to the local network, it can store 1 in this field. When the packet arrives at the first router, this value is decremented to 0, and the datagram is discarded.

**PROTOCOL:** This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. It identifies the transport-layer protocol which will interpret the *Data* section. This will typically be TCP or UDP but other values are possible. Protocols are identified by a unique number. This field specifies the final destination protocol to which the IPv4 datagram is delivered. In other words, since the IPv4 protocol carries data from different other protocols, the value of this field helps the receiving network layer know to which protocol the data belong.

**HEADER CHECKSUM**

This is used to verify the header, and is recomputed at each router hop. This field is left out of IPv6 which relies on the transport layer for verification.

## ADDRESSES AND OPTIONS

**Source address**: This 32-bit field defines the IPv4 address of the source. This field Must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

**Destination address:** This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

These are 32-bit IP addresses which identify the network and host address. Note that IP does not have to specify addresses of any intermediate nodes, this can be left to the router. Routing requirements can also be specified in the Options field. Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging. Although options are not a required part of the IPv4 header, option processing is required of the IPv4 software. This means that all implementations must be able to handle options if they are present in the header.

## IPV6 (INTERNET PROTOCOL VERSION 6 )

IPv6 (Internetworking Protocol, version 6), also known as IPng (Internetworking Protocol, next generation), was proposed and is now a standard. An IPv6 address consists of 16 bytes (octets); it is 128 bits long. An IPv6 address is 128 bits long. IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet are:

- In IPv4 subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet.
- Minimum delay strategies and reservation of resources not provided in the IPv4 design.
- No encryption or authentication is provided by IPv4.

So to overcome these deficiencies IPv6 has proposed. The format and the length of the IP address were changed along with the packet format. Related protocols, such as ICMP, were also
modified. Other protocols in the network layer, such as ARP, RARP, and IGMP, were also modified.

## ADVANTAGES

The next-generation IP, or IPv6, has some advantages over IPv4 that can be summarized

as follows:

- Larger address space.
- Better header format.
- New options. IPv6 has new options to allow for additional functionalities.
- Allowance for extension:IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- Support for resource allocation. Support for more security.

## IPV6 DATAGRAM PACKET HEADER AND FIELDS

**Version:** The size of the Version field is 4 bits. The Version field shows the version of IP and is set to 6.

• **Traffic Class:** The size of Traffic Class field is 8 bits. Traffic Class field is similar to the IPv4 Type of Service (ToS) field. The Traffic Class field indicates the IPv6 packet's class or priority.

• **Flow Label:** The size of Flow Label field is 20 bits. The Flow Label field provide additional support for real-time datagram delivery and quality of service features. The purpose of Flow Label field is to indicate that this packet belongs to a specific sequence of packets between a source and destination and can be used to prioritized delivery of packets for services like voice. The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.

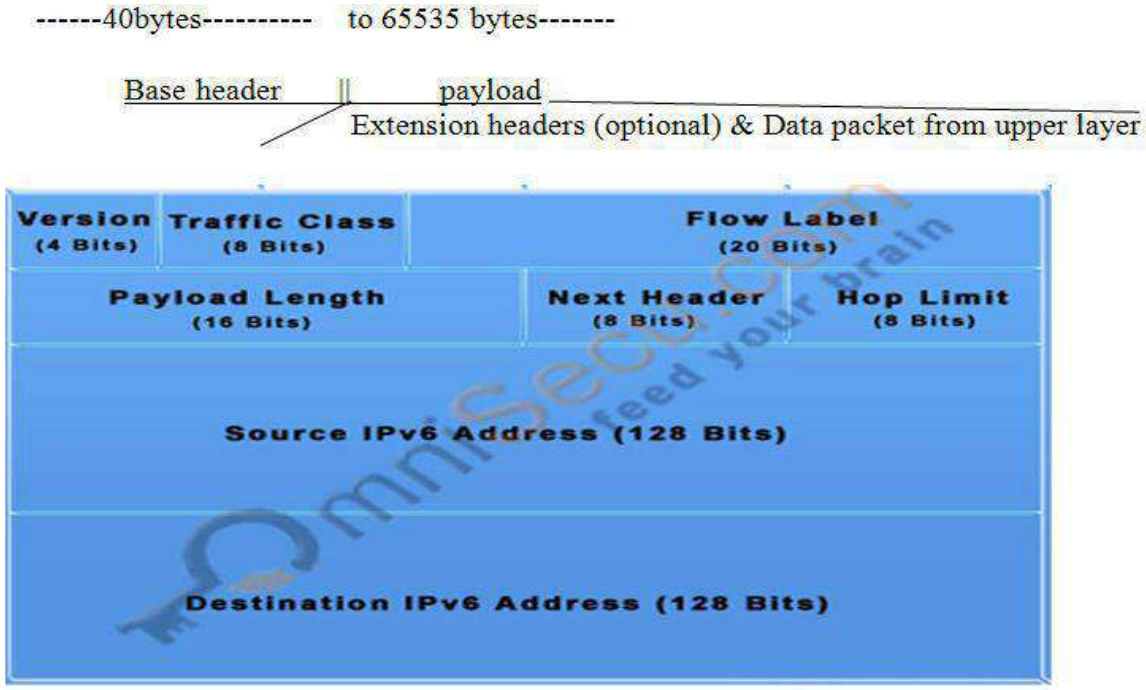
• **Payload Length:** The size of the Payload Length field is 16 bits. The Payload Length field shows the length of the IPv6 payload, including the extension headers and the upper layer protocol data.

• **Next Header:** The size of the Next Header field is 8 bits. The Next Header field shows either the type of the first extension (if any extension header is available) or the protocol in the upper layer such as TCP, UDP, or ICMPv6.

• **Hop Limit:** The size of the Hop Limit field is 8 bits The Hop Limit field shows the maximum number of routers the IPv6 packet can travel. This Hop Limit field is similar to IPv4 Time to Live (TTL) field. This field is typically used by distance vector routing protocols, like Routing Information Protocol (RIP) to prevent layer 3 loops (routing loops).

• **Source Address:** The size of the Source Address field is 128 bits. The Source Address field shows the IPv6 address of the source of the packet.

• **Destination Address:** The size of the Destination Address field is 128 bits. The Destination Address field shows the IPv6 address of the destination of the packet.

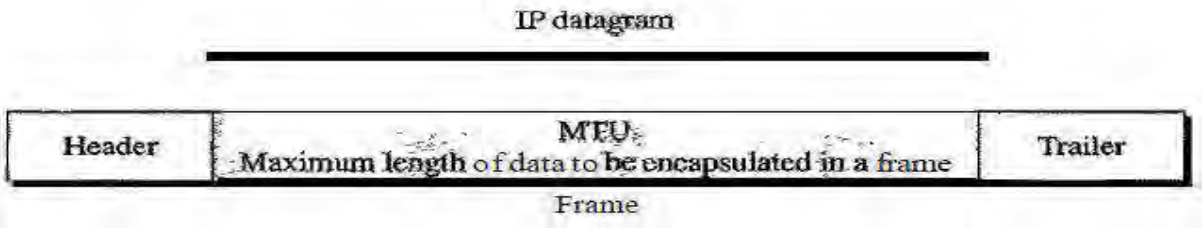



**FRAGMENTATION**

A datagram can travel through different networks. Each router decapsulates the IPv4 datagram from the frame it receives, processes it, and then encapsulates it in another frame. The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled. The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel. For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

**MAXIMUM TRANSFER UNIT (MTU)**

When a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network . The value of the MTU depends on the physical network protocol.




To make the IPv4 protocol independent of the physical network, the designers decided to make the maximum length of the IPv4 datagram equal to 65,535 bytes. This makes transmission more efficient if we use a protocol with an MTU of this size. When we divide the datagram to make it possible to pass through these networks. This is called **fragmentation**.

The source usually does not fragment the IPv4 packet. The transport layer will instead segment the data into a size that can be accommodated by IPv4 and the data link layer in use.

When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but with some changed. A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. In other words, a datagram can be fragmented several times before it reaches the final destination. In IPv4, a datagram can be fragmented by the source host or any router in the path although there is a tendency to limit fragmentation only at the source. The reassembly of the datagram, is done only by the destination host because each fragment becomes an independent datagram.

Whereas the fragmented datagram can travel through different routes, and we can never control or guarantee which route a fragmented datagram may take, all the fragments belonging to the same datagram should finally arrive at the destination host. So it is logical to do the reassembly at the final destination.

When a datagram is fragmented, required parts of the header must be copied by all fragments. The host or router that fragments a datagram must change the values of three **fields: flags, fragmentation offset, and total length**. The rest of the fields must be copied. Of course, the value of the checksum must be recalculated regardless of fragmentation.

### *FIELDS RELATED TO FRAGMENTATION*

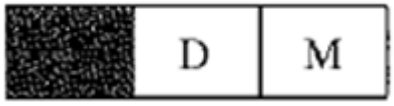
The fields that are related to fragmentation and reassembly of an IPv4 datagram are the identification, flags, and fragmentation offset fields.

**Identification**: This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. To

guarantee uniqueness, the IPv4 protocol uses a counter to label the datagram. The counter is initialized to a positive number. When the IPv4 protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by' ~ 1. As long as the counter is kept in the main memory, uniqueness is guaranteed. When a datagram is fragmented, the value in the identification field is copied to all fragments.

In other words, all fragments have the same identification number, the same as the original datagram. The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value must be assembled into one datagram.

**Flags**: This is a 3-bit field. The first bit is reserved. The second bit is called the ***do not fragment*** bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary. The third bit is called the *more fragment* bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.




M- for more fragment
D-for not fragment

If a fragment itself is fragmented. In this case the value of the offset field is always relative to the original datagram. It is obvious that even if each fragment follows a different path and arrives out of order, the final destination host can reassemble the original datagram from the fragments received by using the following strategy:

1. The first fragment has an offset field value of zero.
2. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.
3. Divide the total length of the first and second fragments by 8. The third fragment has an offset value equal to that result.
4. Continue the process. The last fragment has a *more* bit value of 0.

**FRAGMENTATION**

The concept of fragmentation in IPv6 is the same as that in IPv4. In IPv4, the source or a router is required to fragment if the size of the datagram is larger than the MTU of the network over which the datagram travels. In IPv6, only the original source can fragment. A source must use a path MTU discovery technique to find the smallest MTU supported by any network on the path. The source then fragments using this knowledge.

**AUTHENTICATION**

The authentication extension header has a dual purpose: it validates the message sender and ensures the integrity of data.

**ENCRYPTED SECURITY PAYLOAD**

The encrypted security payload (ESP) is an extension that provides confidentiality and guards against eavesdropping. Destination Option The destination option is used when the source needs to pass information to the destination only. Intermediate routers are not permitted access to this information.

*LECTURE NOTE: 26*

# ROUTING

Routing is the process of forwarding of a packet in a network so that it reaches its intended destination. The main goals of routing are

1. **Correctness:** The routing should be done properly and correctly so that the packets may reach their proper destination.
2. **Simplicity:** The routing should be done in a simple manner so that the overhead is as low as possible. With increasing complexity of the routing algorithms the overhead also increases.
3. **Robustness:** The algorithms designed for routing should be robust enough to handle hardware and software failures and should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted. And the network rebooted every time when some router goes down.
4. **Stability:** The routing algorithms should be stable under all possible circumstances.
5. **Fairness:** Every node connected to the network should get a fair chance of transmitting their packets. This is generally done on a first come first serve basis.
6. **Optimality:** The routing algorithms should be optimal in terms of throughput and minimizing mean packet delays.

## CLASSIFICATION OF ROUTING ALGORITHMS

The routing algorithms may be classified as follows:

1. **Adaptive Routing Algorithm:** These algorithms change their routing decisions to reflect changes in the topology and in traffic as well. These

get their routing information from adjacent routers or from all routers. The optimization parameters are the **distance, number of hops and estimated transit time**. This can be further classified as follows:

i. **Centralized:** In this type some central node in the network gets entire information about the network topology, about the traffic and about other nodes. This then transmits this information to the respective routers. The **advantage** of this is that only one node is required to keep the information. The **disadvantage** is that if the central node goes down the entire network is down, i.e. single point of failure.

ii. **Isolated:** In this method the node decides the routing without seeking information from other nodes. The sending node does not know about the status of a particular link. The **disadvantage** is that the packet may be send through a congested route resulting in a delay. Some examples of this type of algorithm for routing are:

- **Hot Potato:** When a packet comes to a node, it tries to get rid of it as fast as it can, by putting it on the shortest output queue without regard to where that link leads.

- **Backward Learning:** In this method the routing tables at each node gets modified by information from the incoming packets. One way to implement backward learning is to include the identity of the source node in each packet, together with a hop counter that is incremented on each hop. When a node receives a packet in a particular line, it notes down the number of hops it has taken to reach it from the source node. If the previous value of hop count stored in the node is better than the current one then nothing is done but if the current value is better then the value is updated for future use. The **problem** with this is that when the best route goes down then it cannot recall the second best route to a particular node. Hence all the nodes have to forget the stored information's periodically and start all over again.

iii. **Distributed:** In this the node receives information from its neighboring nodes and then takes the decision about which way

to send the packet. The **disadvantage** is that if in between the interval it receives information and sends the packet something changes then the packet may be delayed.

2. **Non-Adaptive Routing Algorithm:** These algorithms do not base their routing decisions on measurements and estimates of the current traffic and topology. Instead the route to be taken in going from one node to the other is computed in advance, off-line, and downloaded to the routers when the network is booted. This is also known as static routing. This can be further classified as:

   i. **Flooding:** Flooding adapts the technique in which every incoming packet is sent on every outgoing line except the one on which it arrived. One problem with this method is that packets may go in a loop. As a result of this a node may receive several copies of a particular packet which is undesirable. Some techniques adapted to overcome these problems are as follows:
   - **Sequence Numbers:** Every packet is given a sequence number. When a node receives the packet it sees its source address and sequence number. If the node finds that it has sent the same packet earlier then it will not transmit the packet and will just discard it.
   - **Hop Count:** Every packet has a hop count associated with it. This is decremented(or incremented) by one by each node which sees it. When the hop count becomes zero(or a maximum possible value) the packet is dropped.
   - **Spanning Tree:** The packet is sent only on those links that lead to the destination by constructing a spanning tree routed at the source. This avoids loops in transmission but is possible only when all the intermediate nodes have knowledge of the network topology.

     Flooding is not practical for general kinds of applications. But in cases where high degree of robustness is desired such as in military applications, flooding is of great help.

   ii. **Random Walk:** In this method a packet is sent by the node to one of its neighbours randomly. This algorithm is highly robust. When the network is highly interconnected, this algorithm has the property of making an alternative routes. It

is usually implemented by sending the packet onto the least queued link.

**DELTA ROUTING**

Delta routing is a hybrid of the centralized and isolated routing algorithms. Here each node computes the cost of each line (i.e some functions of the delay, queue length, utilization, bandwidth etc) and periodically sends a packet to the central node giving it these values which then computes the **k** best paths from node **i** to node **j**. Let **Cij1** be the cost of the best **i-j** path, **Cij2** the cost of the next best path and so on. If **Cijn - Cij1 < delta**, (**Cijn** - cost of **n'th** best **i-j** path, **delta** is some constant) then path **n** is regarded equivalent to the best **i-j** path since their cost differ by so little. When **delta -> 0** this algorithm becomes centralized routing and when **delta -> infinity** all the paths become equivalent.

**MULTIPATH ROUTING**

In networks if there are several paths between pairs of nodes to forward a packet this technique is called multipath routing bifurcated routing. Sometimes in order to improve the performance multiple paths between single pair of nodes are used. In this case each node maintains a table with one row for each possible destination node. A row gives the best, second best, third best, etc outgoing line for that destination, together with a relative weight. Before forwarding a packet, the node generates a random number and then chooses among the alternatives, using the weights as probabilities. The tables are worked out manually and loaded into the nodes before the network is brought up and not changed thereafter.

*LECTURE NOTE: 27*

# HIERARCHICAL ROUTING

In this method of routing the nodes are divided into regions based on hierarchy. A particular node can communicate with nodes at the same hierarchial level or the nodes at a lower level and directly under it. Here, the path from any source to a destination is fixed and is exactly one if the heirarchy is a tree.

This is essentially a 'Divide and Conquer' strategy. The network is divided into different regions and a router for a particular region knows only about its own domain and other routers. Thus, the network is viewed at two levels:

1. The Sub-network level, where each node in a region has information about its peers in the same region and about the region's interface with other regions. Different regions may have different 'local' routing algorithms. Each local algorithm handles the traffic between nodes of the same region and also directs the outgoing packets to the appropriate interface.

2. The Network Level, where each region is considered as a single node connected to its interface nodes. The routing algorithm's at this level handle the routing of packets between two interface nodes, and is isolated from intra-regional transfer.

**Networks can be organized in hierarchies of many levels; e.g. local networks of a city at one level, the cities of a country at a level above it, and finally the network of all nations**.

In Hierarchical routing, the interfaces need to store information about:

- All nodes in its region which are at one level below it.
- Its peer interfaces.
- At least one interface at a level above it, for outgoing packages.

Advantages of Hierarchical Routing :

- Smaller sizes of routing tables.
- Substantially lesser calculations and updates of routing tables.

**Disadvantage:**

- Once the hierarchy is imposed on the network, it is followed and possibility of direct paths is ignored. This may lead to sub optimal routing.

## NON-HIERARCHICAL ROUTING

In this type of routing, interconnected networks are viewed as a single network, where bridges, routers and gateways are just additional nodes.

- Every node keeps information about every other node in the network
- In case of adaptive routing, the routing calculations are done and updated for all the nodes.

The above two are also the **disadvantages of non-hierarchical routing**, since the table sizes and the routing calculations become too large as the networks get bigger. So this type of routing is feasible only for small networks.

## SOURCE ROUTING

Source routing is similar in concept to virtual circuit routing. It is implemented as under:

- Initially, a path between nodes wishing to communicate is found out, either by flooding or by any other suitable method.
- This route is then specified in the header of each packet routed between these two nodes. A route may also be specified partially, or in terms of some intermediate hops.

**Advantages**:

- Bridges do not need to look up their routing tables since the path is already specified in the packet itself.

- The throughput of the bridges is higher, and this may lead to better utilization of bandwidth, once a route is established.

**Disadvantages:**

- Establishing the route at first needs an expensive search method like flooding.
- To cope up with dynamic relocation of nodes in a network, frequent updates of tables are required, else all packets would be sent in wrong direction. This too is expensive.

## POLICY BASED ROUTING

In this type of routing, certain restrictions are put on the type of packets accepted and sent. e.g.. The IIT- KGP router may decide to handle traffic pertaining to its departments only, and reject packets from other routes. This kind of routing is used for links with very low capacity or for security purposes.

## SHORTEST PATH ROUTING

Here, the central question dealt with is 'How to determine the optimal path for routing ?' Various algorithms are used to determine the optimal routes with respect to some predetermined criteria. A network is represented as a graph, with its terminals as nodes and the links as edges. A 'length' is associated with each edge, which represents the cost of using the link for transmission. Lower the cost, more suitable is the link. The cost is determined depending upon the criteria to be optimized. Some of the important ways of determining the cost are:

- **Minimum number of hops:** If each link is given a unit cost, the shortest path is the one with minimum number of hops. Such a route is easily obtained by a breadth first search method. This is easy to implement but ignores load, link capacity etc.

- **Transmission and Propagation Delays:** If the cost is fixed as a function of transmission and propagation delays, it will reflect the link capacities and the geographical distances. However these costs are essentially static and do not consider the varying load conditions.

- **Queuing Delays:** If the cost of a link is determined through its queuing delays, it takes care of the varying load conditions, but not of the propagation delays.

## *LECTURE NOTE: 28*

# *NETWORK LAYER: ADDRESS MAPPING, ERROR REPORTING, AND MULTICASTING:*

We need protocols to create a mapping between physical and logical addresses. IP packets use logical (host-to-host) addresses. These packets, however, need to be encapsulated
in a frame, which needs physical addresses (node-to-node). When booting a diskless network or leasing an IP address to a host three protocols are designed for this purpose: RARP, BOOTP, and DHCP are used to map an address.

**ADDRESS MAPPING**

An internet is made of a combination of physical networks connected by internetworking devices such as routers. A packet starting from a source host may pass through several different physical networks before finally reaching the destination host. The hosts and routers are recognized at the network level by their logical (IP) addresses. However, packets pass through physical networks to reach these hosts and routers.

At the physical level, the hosts and routers are recognized by their physical addresses (local address). Its jurisdiction is a local network which is unique locally, but is not necessarily unique universally. It is called a *physical* address because it is usually (but not always) implemented in hardware. An example of a physical address is the 48-bit MAC address in the Ethernet protocol, which is imprinted on the **NIC**(network interface card) installed in the host or router.

Delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to map a logical address to its corresponding physical address and vice versa. These can be done by using either static or dynamic mapping. In **Static mapping** a table is created that associates a logical address with a physical address. This table is stored in each machine on the

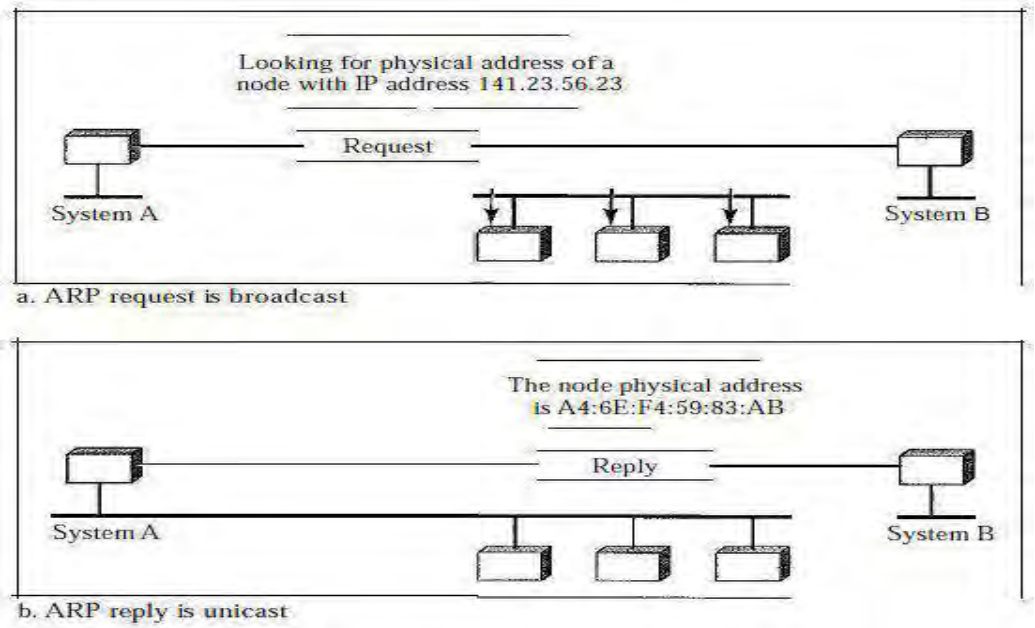
network. Each machine that knows, the IP address of another machine but not its **physical address** and can look it up in the table.

Because of some limitations because physical addresses may change in the following ways:

- A machine could change its NIC, resulting in a new physical address.
- In some LANs, such as Local Talk, the physical address changes every time the computer is turned on.
- A mobile computer can move from one physical network to another, resulting in a change in its physical address.

This overhead could affect network performance. To implement these changes, a static mapping table must be updated periodically. One of its solution is the **dynamic mapping**: in dynamic mapping each time a machine knows one of the two addresses (logical or physical), by using a protocol to find the other one. These protocols are: **ARP, RARP** etc.

**Mapping Logical to Physical Address: ARP**




The logical (IP) address is obtained from the DNS, if the sender is the host or it is found in a routing table, if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address
of the receiver. **The host or the router sends an ARP query packet**. The packet includes the physical and IP addresses of the sender and the IP address of the

receiver. Because the sender does not know the **physical address** of the receiver, the query is broadcast over the network.

Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's **IP and physical addresses**. The packet is **unicast** directly to the inquirer (or sender) by using the physical address received in the query packet.

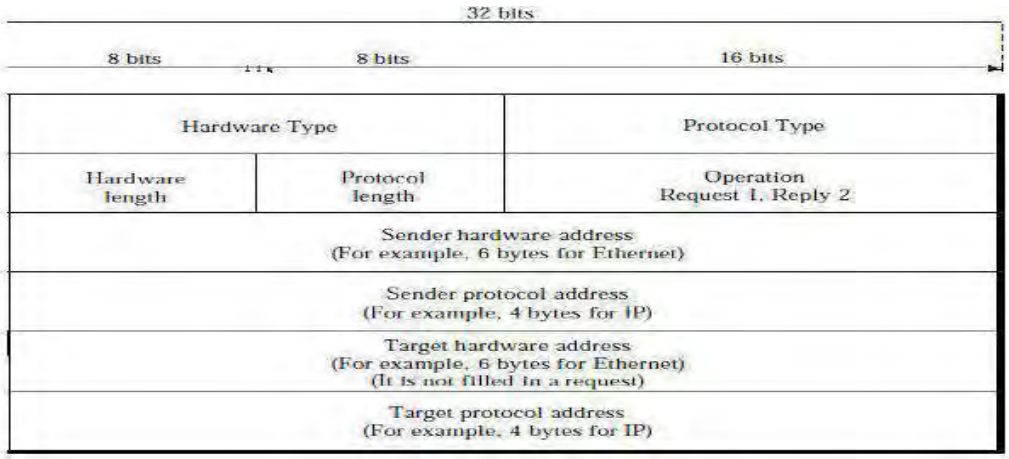
In the above figure the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of 141.23.56.23. This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination by using the **physical address** it received.

### *CACHE MEMORY*

A cache memory is used for the purpose of store and forwards a packet for some times.

ARP can be useful if the ARP reply is cached (kept in cache memory for a while) because a system normally sends several packets to the same destination. A system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes unless the space in the cache is exhausted. Before sending an ARP request, the system first checks its cache to see if it can find the mapping.

### *PACKET FORMAT OF ARP*




**Hardware type:** This is a 16-bit field defining the type of the network on which ARP is running.

**Protocol type:** This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016, ARP can be used with any higher-level protocol.

**Hardware length:** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.

**Protocol length:** This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.

**Operation:** This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) and ARP reply (2).

**Sender hardware address:** This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.

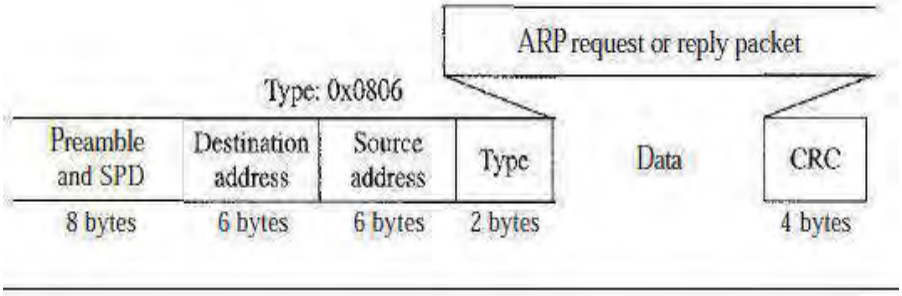
**Sender protocol address:** This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.

**Target hardware address:** This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all 0s because the sender does not know the physical address of the target.

**Target protocol address**: This is a variable-length field defining the logical (for example, IP address) of the target. For the IPv4 protocol, this field is 4 bytes long.

***ENCAPSULATION***

An ARP packet is encapsulated directly into a data link frame. As in the Figure below an ARP packet is encapsulated in an Ethernet frame. Note that the type field indicates that the data carried by the frame are an ARP packet.




***OPERATION***

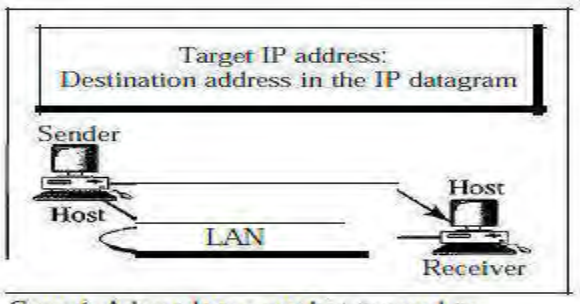
The steps involved in an ARP process are:

1. The sender knows the IP address of the target.
2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with 0s.
3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.
4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes its IP address.
5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast to ender directly.
6. The sender receives the reply message. It now knows the physical address of the target machine.
7. The IP datagram, which carries data for the target machine, is now encapsulated in frame and is unicast to the destination.

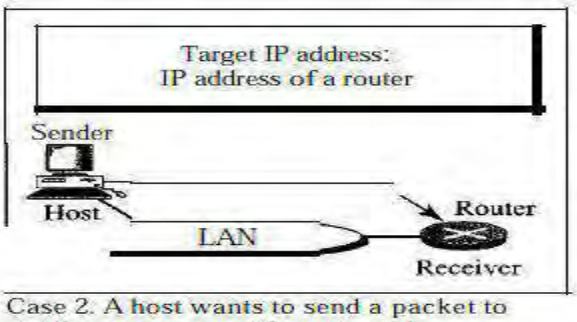
***FOUR DIFFERENT CASES***

The following are four different cases in which the services of ARP can be used:

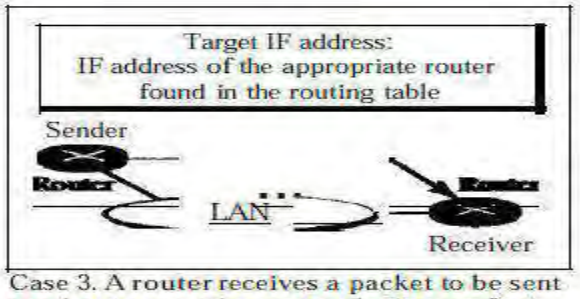
1. The sender is a host and wants to send a packet to another host on the same network. then logical address that must be mapped to a physical address is the destination IP address in the datagram header.



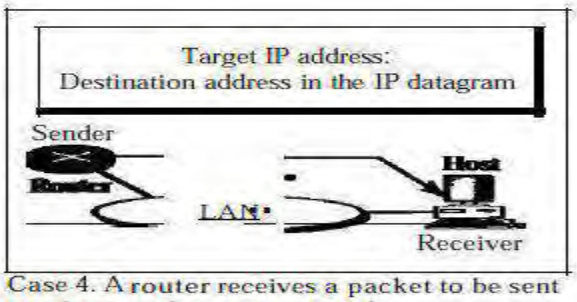

Case 1. A host has a packet to send to another host on the same network.




Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.




Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.




Case 4. A router receives a packet to be sent to a host on the same network.

2. The sender is a host and wants to send a packet to another host on another network. In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table, it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.

3. The sender is a router that has received a datagram destined for a host on another network. It checks its routing table and finds the IP address of the next router. The IP address of the next router becomes the logical address and that must be mapped to a physical address.

4. The sender is a router that has received a datagram destined for a host on the same network. The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.

---

An ARP request is broadcast; an ARP reply is unicast.

---

## Mapping Physical to Logical Address: RARP, BOOTP, and DHCP

There are some situations when a host knows its physical address, but needs to know its logical address such situations are as:

1. A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.
2. An organization does not have enough IP addresses to assign to each station, it needs to assign IP addresses on demand. The station can send its physical address and ask for a short time lease.

### *RARP*

Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address. To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file. However, a diskless machine is usually booted from ROM, which has minimum booting information. The ROM is installed by the manufacturer. It cannot include the IP address because the IP addresses on a network are assigned by the network administrator. The machine can get its physical address (by reading its NIC, for example), which is **unique locally**. It can then use the physical address to get the logical address by using the RARP protocol.

A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program and the responding machine must be running a RARP server program.

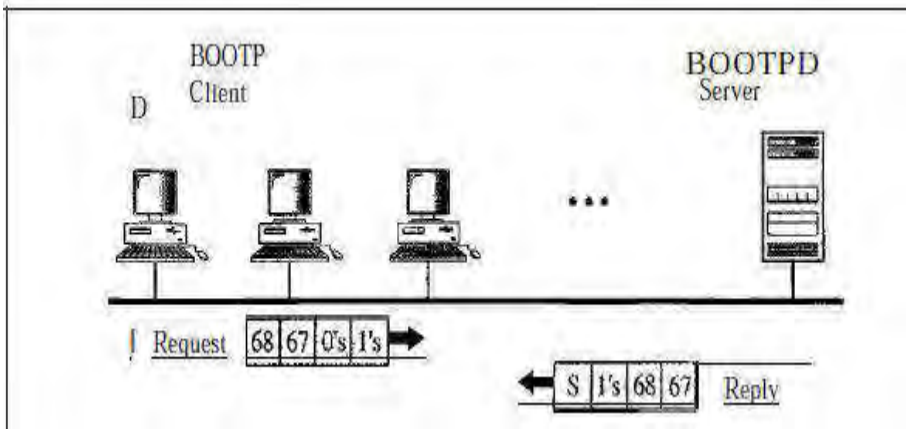
But the problem with RARP is: Broadcasting is done at the data link layer. The physical broadcast address, all is in the case of Ethernet, does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet. This is the reason that RARP is almost obsolete. two protocols, **BOOTP** and **DHCP**, are replacing RARP.

## *LECTURE NOTE: 29*

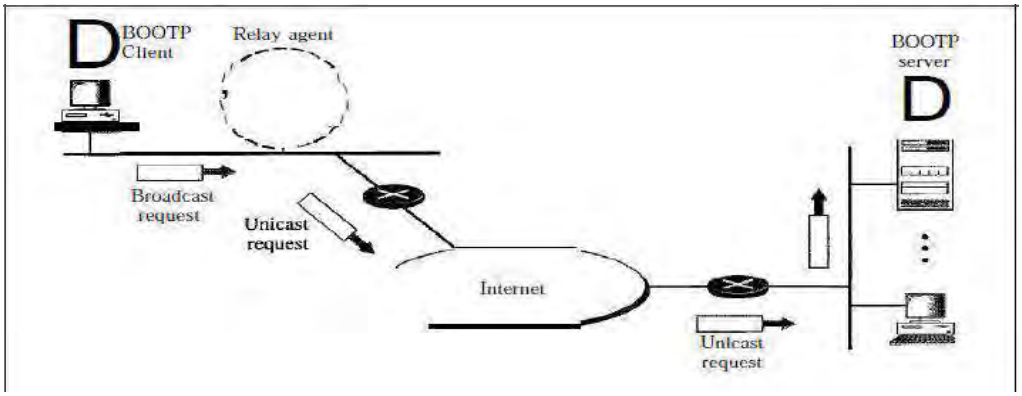
# *BOOTP*

The Bootstrap Protocol (BOOTP) is a client/server protocol. It is designed to provide physical address to logical address mapping. BOOTP is an application layer protocol. The administrator may put the client and the server on the same network or on different networks, as shown in Figure below. BOOTP messages are encapsulated in a UDP packet, and the UDP packet itself is encapsulated in an IP packet.

a. Client and server on the same network




b. Client and server on different networks




One may ask how a client can send an IP datagram when it knows neither its own IP address (the source address) nor the server's IP address (the destination address).

Here the client simply uses all 0s as the source address and all 1s as the destination address.

One of the advantages of BOOTP over RARP is that the client and server are application-layer processes. The BOOTP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router. One of the hosts (or a router that can be configured to operate at the application layer) can be used as a relay called a **relay agent**. The relay agent knows the unicast address of a BOOTP server.

When relay agent receive packet, it encapsulate the message in unicast datagram and sends the request to the BOOTP server. The packet carrying a unicast destination address, is routed by any router and reaches the BOOTP server. The BOOTP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent. The relay agent, after receiving the reply, sends it to the BOOTP client.

***DHCP***

BOOTP is not a dynamic configuration protocol, it is a static configuration protocol. Where the binding between the physical address and the IP address of the client already exists or the binding is predetermined. But question arise what if a host moves from one physical network to another? What if a host wants a temporary IP address? BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator. The Dynamic Host Configuration Protocol (DHCP) has been proposed to provide static and dynamic address allocation that can be **manual or automatic**.

**Static Address Allocation:** a host running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.

**Dynamic Address Allocation:** DHCP provides temporary IP addresses for a limited time. DHCP has a second database with a pool of available IP addresses. This database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.

When a DHCP client sends a request to a DHCP server for IP address, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned (or assigned). If the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and

adds the entry to the dynamic database. This dynamic aspect of DHCP is needed when a host moves from network to other network or is connected and disconnected from a network. The addresses assigned from the pool are temporary addresses. The DHCP server issues a lease for a specific time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the option to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

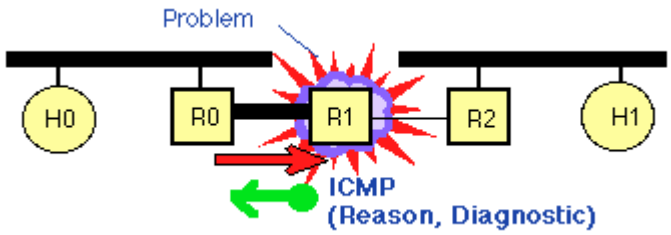
## MANUAL AND AUTOMATIC CONFIGURATION

One problem with the BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured. This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes. Where DHCP allows both manual and automatic configurations. Static addresses are created manually and dynamic addresses are created automatically.

## ICMP

The Internet Control Message Protocol (ICMP) protocol is classic example of a client server application. The ICMP server executes on all IP end system computers and all IP intermediate systems (i.e routers). The protocol is used to report problems with delivery of IP datagrams within an IP network. It can be sued to show when a **particular End System (ES) is not responding**, when an IP network is **not reachable**, when a **node is overloaded**, when an **error occurs** in the IP header information, etc. The protocol is also frequently used by Internet managers to verify correct operations of End Systems (ES) and to check that routers are correctly routing packets to the specified destination address.

Although, IP provide unreliable and connectionless datagram and best effort delivery. It has two deficiencies that: it has **no error-reporting or error-correcting mechanism secondly it lacks a mechanism for host and management queries.** The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protoco1. It report about the error or any mishappening occurred to the datagram.



In the above figure ICMP messages generated by router R1, in response to message sent by H0 to H1 and forwarded by R0. This message could, instance be generated if the MTU of the link between R0 and R1 was smaller than size of the IP packet, and the packet had the Don't Fragment (DF) bit set in the IP packet header. The ICMP message is returned to H0, since this is the source address specified in the IP packet that suffered the problem. A modern version of Path MTU Discovery provides a mechanism to verify the Path MTU.

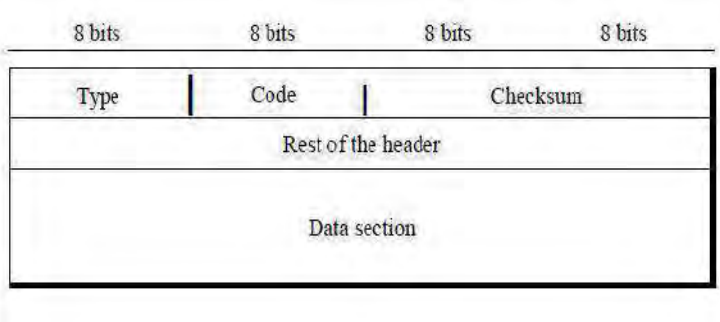
**Types of Message:** ICMP messages are divided into two broad categories:

1. **Error-reporting messages:** The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

**Query messages**:- The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbours also, hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

**Message Format**

An ICMP message has an 8-byte header and a variable-size data section. The general format of the header is different for each message type, where the first 4 bytes are common to all.



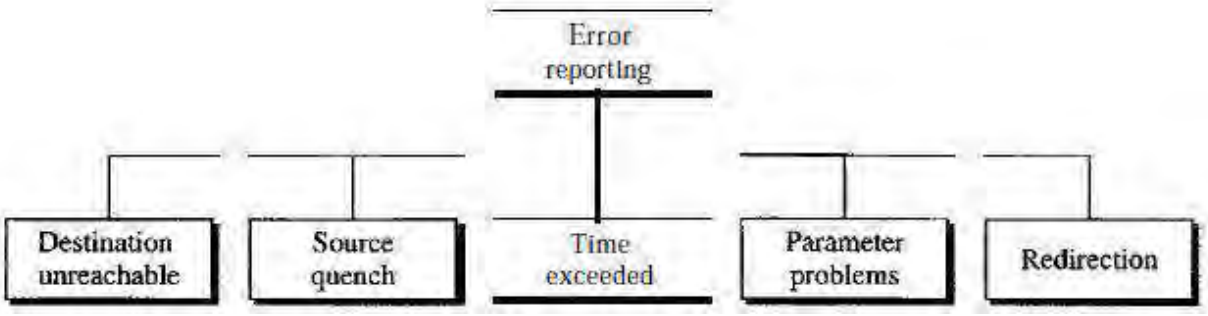

The first field ICMP type, defines the type of the message. The code field specifies the reason for the particular message type. The rest of the header is specific for each message type. The data section in error messages carries information for finding the original packet that had the error. In query messages, the data section carries extra information based on the type of the query. The **CHECKSUM** provides a method for determining the integrity of the message.

**Error Reporting**

- One of the main responsibilities of ICMP is to report errors. error checking and error control mechanism are provided by ICMP However, ICMP does not **correct errors** it simply reports them. Error correction is left to the higher-level protocols. ICMP uses the source IP address to send the error message to the source (originator) of the datagram.

ICMP always reports error messages to the original source.

- Five types of errors are handled: **destination unreachable, source quench, time exceeded, parameter problems, and redirection**

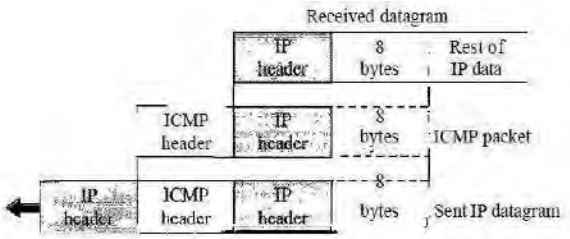



- The following are important points about ICMP error messages:

1) No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
2) No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
3) No ICMP error message will be generated for a datagram having a multicast address.
4) No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

- All error messages contain a data section that includes the IP header of the original datagram plus the first 8 bytes of data.
- The original datagram header contain original source, which receives the error message, information about the datagram itself.
- UDP and TCP protocols, the first 8 bytes provide information about the port numbers (UDP and TCP) and sequence number (TCP).

- This information is needed so the source can inform the protocols (TCP or UDP) about the error.
- ICMP forms an error packet, which is then encapsulated in an IP datagram




[Contents of data field for the error messages]

**DESTINATION UNREACHABLE**

- When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.
- Destination-unreachable messages can be created by either a router or the destination host.
- When a packet is undeliverable, a Destination Unreachable, ICMP error is generated. this ICMPs can have a Code value of 0 to 15:

**SOURCE QUENCH**

- An ICMP Source Quench message has a Type field of 4 and Code 0. Source Quench messages are sent when the destination is unable to process traffic as fast as the source is sending it.
- The Source Quench ICMP tells the source to cut back the rate at which it is sending data.
- The destination will continue to generate Source Quench ICMPs until the source is sending at an acceptable speed.
- The source-quench message in ICMP was designed to add a kind of flow control to the IP.
- When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram.
- This message has two purposes. **First**, it informs the source that the datagram has been discarded. **Second**, it warns the source that there is

congestion somewhere in the path and that the source should slow down (quench) the sending process.

**TIME EXCEEDED**

If a router or host discards a packet due to a time-out, it will generate a Time Exceeded Type 11 ICMP. The Time Exceeded ICMP will have a Code value of either 0 or 1. A Code 0 is generated when the hop count of a datagram is exceeded and the packet is discarded. A Code 1 is generated when the reassemble of a fragmented packet exceeds the time-out value.

- The time-exceeded message is generated in two cases: routers use routing tables to find the next hop (next router) that must receive the packet. If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly.
- Each datagram contains a field called ***time to live*** that controls this situation.
- When a datagram visits a router, the value of this field is decremented by 1. When the time-to-live value reaches 0, after decrementing, the router discards the datagram. However, when the datagram is discarded, a time-exceeded message must be sent by the router to the original source.
- Second, a time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.

**PARAMETER PROBLEM**

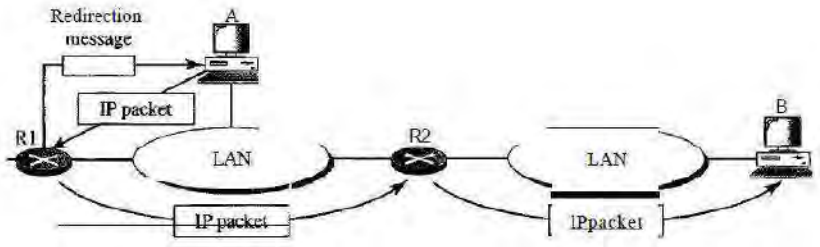
- Any ambiguity in the header part of a datagram can create serious problems as the datagram travels through the Internet.
- If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.

*LECTURE NOTE: 30*

# REDIRECTION

An intermediary device will generate an ICMP Redirect Message when it determines that a route being requested can be reached either locally or through a better path. Redirect Message ICMPs are Type 5.

- When a router needs to send a packet destined for another network, it must know the IP address of the next router. The same is true if the sender is a host. Both routers and hosts, then, must have a routing table to find the address of the next router.
- Updating the routing tables of hosts dynamically produces unacceptable traffic.
- The hosts usually use static routing.
- When a host comes up, its routing table has a limited number of entries. It usually knows the IP address of only one router, the default router.
- For this reason, the host may send a datagram, which is destined for another network, to the **wrong router**. In this case, the router that receives the datagram will forward the datagram to the **correct router**.
- However, to update the routing table of the host, it sends a **redirection message** to the host.

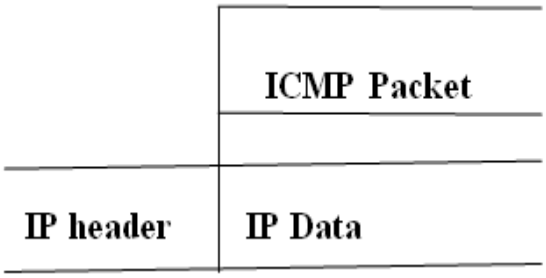



- Host A wants to send a datagram to host B. Router R2 is obviously the most efficient routing choice, but host A did not choose router R2.

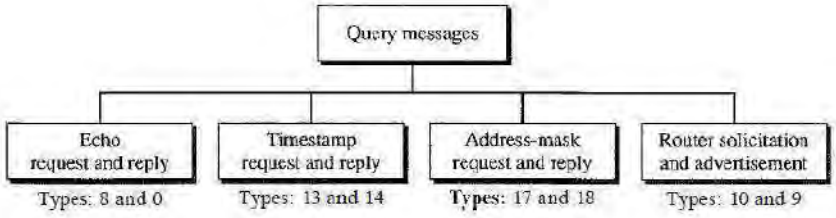
- The datagram goes to R1(wrong route) instead. Router R1, after consulting its table, finds that the packet should have gone to R2. It sends the packet to R2 and, at the same time, sends a redirection message to host A.

**Query message**

**Encapsulation of ICMP query messages**




- In addition to error reporting ICMP can diagnose some network problems through query message.
- A query message is encapsulated in an IP packet (no bytes added to original message) which is encapsulated in data link layer.
- These query messages are of four different pairs of messages.
- In this type of ICMP message, a node sends a message is in a specific format by the destination node.




[Query messages]

**1) Echo Request and Reply**

It is used to test IP connectivity commonly known as PING. The Echo Request ICMP will have a Type field of 8 and a Code field of 0. Echo Replies have a Type field of 0 and a Code field of 0.

- The echo-request and echo-reply messages are designed for diagnostic purposes. Network managers and users utilize this pair of messages to identify network problems.
- The combination of echo-request and echo-reply messages determines whether two systems(hosts or routers) can communicate with each other.
- The echo-request and echo-reply messages can be used to determine if there is communication at the IP level.

**2) Timestamp Request and Reply**

It is a is a rudimentary method for synchronizing the time maintained on different devices. The Request has a Type field of 13 and the Reply is Type 14. This method for time synchronization is crude and unreliable. Therefore, it is not heavily used.

- Two machines (hosts or routers) can use the timestamp request and timestamp reply messages to determine the round-trip time needed for an IP datagram to travel between them.
- It can also be used to synchronize the clocks in two machines.

**3) Address-Mask Request and Reply**

Address mask request is used by a booting computer to determine the subnet mask in use on the local network, its field Type 17. An intermediary device or computer acting as an intermediary device will reply with a Type 18 ICMP Address Mask Reply ICMP.

- A host may know its IP address, but it may not know the corresponding mask.
- To obtain its mask, a host sends an address-mask-request message to a router on the LAN.
- If the host knows the address of the router, it sends the request directly to the router. If it does not know, it broadcasts the message.
- The router receiving the address-mask-request message responds with an address-mask-reply message, providing the necessary mask for the host.
- This can be applied to its full IP address to get its subnet address.

**4) Router Solicitation and Advertisement**

- When a host wants to send data to a host on another network needs to know the address of routers connected to its own network and also to

know the router information such as routers are alive and functioning or not, the router-solicitation and router-advertisement messages can help. So this message designed to allow a booting host to discover an IP address.

- A host can broadcast (or multicast) a router-solicitation message.
- The router or routers that receive the solicitation message broadcast their routing information using the router-advertisement message. A router can also periodically send router-advertisement messages even if no host has solicited.
- When a router sends out an advertisement, it announces not only its own presence but also the presence of all routers on the network of which it is aware.

***Checksum***

In ICMP the checksum is calculated over the entire message (header and data).

**DEBUGGING TOOLS**

- Debugging tools are used in the Internet for debugging, to determine the viability of a host or router and to trace the route of a packet.
- The ICMP used two tools for debugging are: ***ping*** **and** ***trace route***.

**PING**

- A ping program is used to find if a host is alive and responding. Its working is described as:
- The source host sends ICMP echo-request messages (type: 8, code: 0); the destination, if alive, responds with ICMP echo-reply messages.
- The *ping* program sets the identifier field in the echo-request and echo-reply message and stans the sequence number from 0; this number is incremented by 1 each time a new message is sent.
- *Ping* can calculate the round-trip time. It inserts the sending time in the data section of the message.
- When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT).

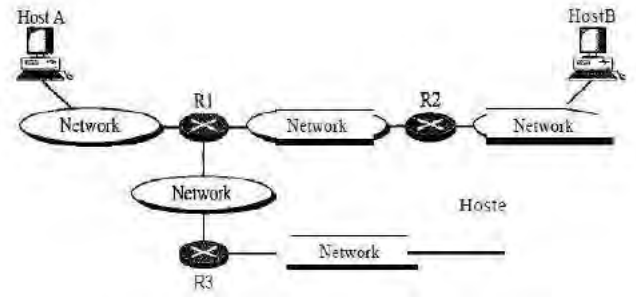
**a) Trace route**

- The *trace route* program in UNIX or in Windows can be used to trace the route of

a packet from the source to the destination.

- This program uses two ICMP messages: **time exceeded and destination unreachable,** to find the route of a packet. This is a program at the application level that uses the services of UDP.

*The trace route program operation*




In the figure a packet from host A to host B travels through routers Rl and R2. However, most of the time, we are not aware of this topology. There could be several routes from A to B. The *trace route* program uses the ICMP messages and the TTL (time to live) field in the IP packet to find the route.
The *trace route* program uses several steps to find the address of the router Rl and the round-trip time between host A and router Rl are:

Step1: The *trace route* application at host A sends a packet to destination B using UDP; the message is encapsulated in an IP packet with a TTL value of 1. The program notes the time the packet is sent.

Step2: Router Rl receives the packet and decrements the value of TTL to 0. It then discards the packet (because TTL is 0). The router, however, sends a time-exceeded ICMP message (type: 11, code: 0) to show that the TTL value is 0 and the packet was discarded.

Step3: The *trace route* program receives the ICMP messages, it note the packet time arrival and uses the destination address to find the IP address of router Rl. The difference between this time and the time at step a is the round-trip time.

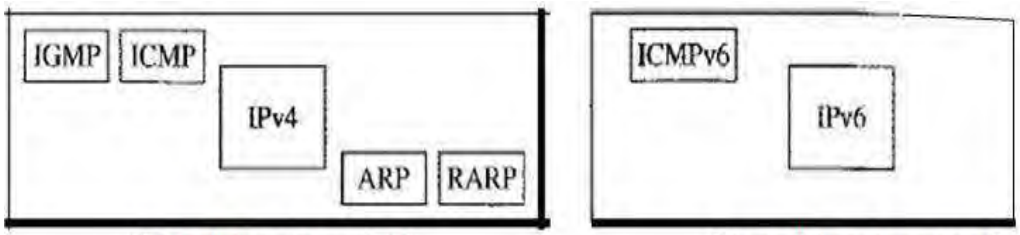
It repeats steps a to c three times to get a better average round-trip time. The first trip time may be much longer than the second or third because it takes time for the ARP program to find the physical address of router RI. For the second and third trips, ARP has the address in its cache.

***LECTURE NOTE: 31***

# ICMPv6

ICMP protocol is the modified version 6 of the TCP/IP protocol suite that is ICMPv6. This new version follows the same strategy and purposes of version 4. ICMPv4 has been modified to make it more suitable for IPv6. In addition, some protocols that were independent in version 4 are now part of Internetworking Control Message Protocol (ICMPv6).

The ARP and IGMP protocols in version 4 are combined in ICMPv6. The RARP Protocol is dropped from the suite because it was rarely used and BOOTP has the same functionality.




Just as in ICMPv4 it divides its message into two parts:

**Error Reporting**

One of the main responsibilities of ICMP is to report errors. Five types of errors are handled: destination unreachable, packet too big, time exceeded, parameter problems, and redirection. ICMPv6 forms an error packet, encapsulated in an IP datagram then delivered to the original source of the failed datagram. When compared version 4 with version 6, the source-quench message is eliminated in version 6 because the priority and the flow label fields allow the router to control congestion and discard the least important messages. The packet-too-big message is added because fragmentation is the responsibility of the sender in IPv6 so no need to inform the sender to slow down.

The concept of ***Destination Unreachable, Time Exceeded, Parameter Problem, Redirection*** is exactly same as ICMPv4.

***Packet Too Big***

This is a new type of message added to version 6. If a router receives a datagram that is larger than the maximum transmission unit (MTU) size of the network through which the datagram should pass, **two things happen. First**, the router discards the datagram and then an ICMP error- packet a **packet-too-big** message- is sent to the source.

**Query**

Error reporting and diagnosing some network problems are accomplished by the ICMP through the query messages. Four different groups of messages have been defined: **echo request and reply, router solicitation and advertisement, neighbor solicitation and advertisement, and group membership.** When both are compared **here t**wo sets of query messages are eliminated from ICMPv6**:**

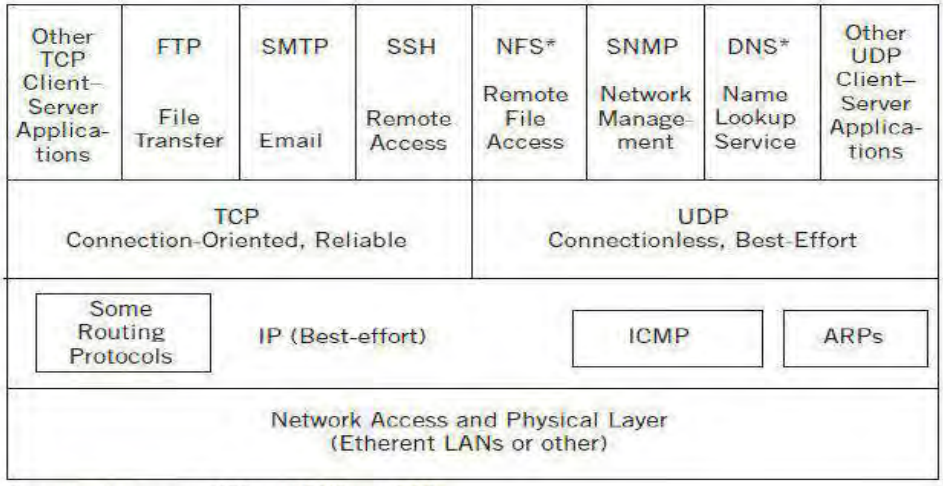
**i) time-stamp request and reply**

**ii) address-mask request and reply**.

First one eliminated because it is implemented in other protocol such as TCP and rarely used in past. Second option is eliminated because, subnet section of an address allows the subscriber to use up to $2^{32 - 1}$ subnets. Therefore, subnet masking, as defined in IPv4, is not needed here. *Echo Request and Reply*, *Router Solicitation and Advertisement*, *Neighbor Solicitation and Advertisement* are similar to version 4. In the version 4 independent protocol IGMP is mentioned, here it is eliminated and its duties are included in ICMPv6.

***LECTURE NOTE: 32***

# TRANSPORT LAYER




*In some instances, NFS and DNS use TCP.

The transport layer is responsible for process-to-process delivery of the entire message where a process is an application program running on a host. The transport layer, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level.

> The transport layer is responsible for the delivery of a message from one process to another.

The transport layer is responsible for the delivery of a message from one process to another. Source to-destination delivery means delivery not only from one computer to the next but also from a specific process on one computer to a specific process on the other. The transport layer header must

therefore include a type of address called a ***service-point address*** in the OSI model and **port number** or **port addresses** in the Internet and TCP/IP protocol suite. A transport layer protocol 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 first before delivering the packets. After all the data is transferred, the connection is terminated.

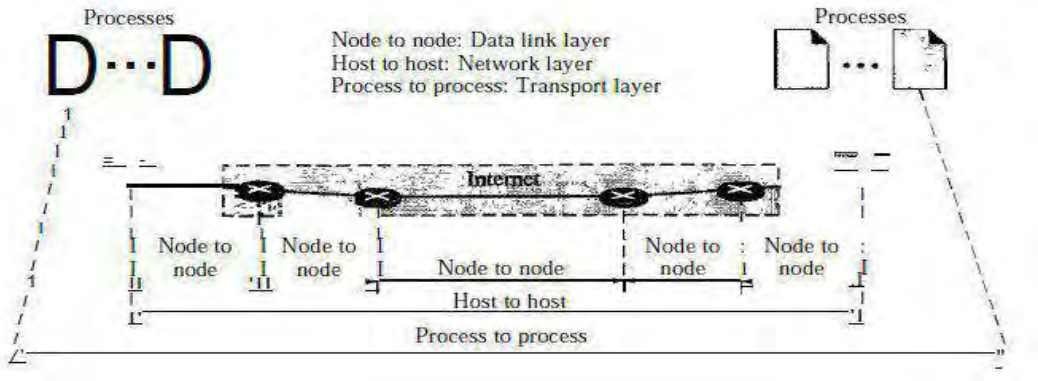
In the transport layer, if a message size is large divided into transmittable segments. A connectionless protocol, such as UDP, treats each segment separately. A connection oriented protocol, such as TCP and SCTP, creates a relationship between the segments using sequence numbers.

This layer is responsible for flow and error control at end to end rather than across a single link.

**Process to process delivery**

The data link layer is responsible for delivery of frames between two neighboring nodesover a link is called ***node-to-node delivery***. This layer responsible for delivery of datagram's between two hosts is called ***host-to-host delivery***. *Actual* Communication on Internet is not defined by exchange of data but takes place between two processes (an application program), for that we need process to process delivery. Two processes communicate in a client/server relationship.

Types of data delivery




**Client/Server Paradigm**

The most common one client/server paradigm is the ways to achieve process-to-process communication. A process on the local host, called a client, needs services from a process on the remote host, called a server.

Both processes (client and server) have the same name i.e Daytime. To get the day and time from a remote machine, we need a Daytime client process running on the local host and a Daytime server process running on a remote machine. A remote computer can run several server programs at the same time, just as local computers can run one or more client programs at the same time. For communication, we must define the following:

1. Local host
2. Local process
3. Remote host
4. Remote process

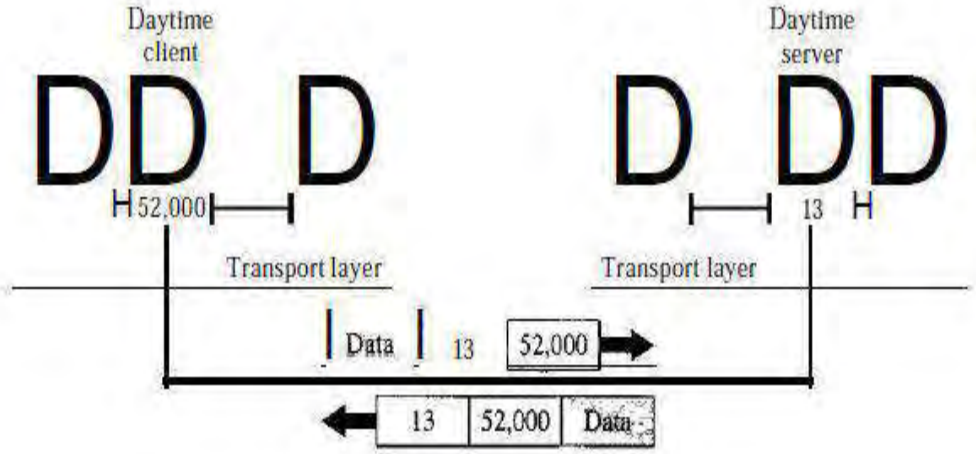
**ADDRESSING:**

At a data link layer, whenever we need to deliver something to one specific destination among many, we need a destination MAC address to choose one node among several nodes if the connection is not point-to-point and source MAC address for reply. At network layer we need source and destination IP address similarly at the transport layer, we need a transport layer address, called a port number (destination port number), to choose among multiple processes running on the destination host and source port number to reply.

In the Internet model, the port numbers are 16-bit integers between 0 and 65,535.

The client program defines itself with a port number, chosen randomly by the transport layer software running on the client host called **ephemeral port number**.

The server process must also define itself with a port number, however, cannot be chosen randomly. If a server process assigns a random number as the port number, the process at the client site that wants to access that server will not know the port number then it can send a special packet and request the port number of a specific server, but this requires more overhead. Therefore Internet has decided to use **universal port numbers** for servers called **well-known port** numbers. To this rule exception are; there are clients that are assigned well-known port numbers. Every client process knows the well-known port number of the corresponding server process. For example, while the Daytime client process can use an ephemeral (temporary) port number 52,000 to identify itself, the Daytime server process must use the well-known (permanent) port number 13.




The IP addresses and port numbers play different roles in selecting the final destination of data. The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this particular host

**lANA Ranges**

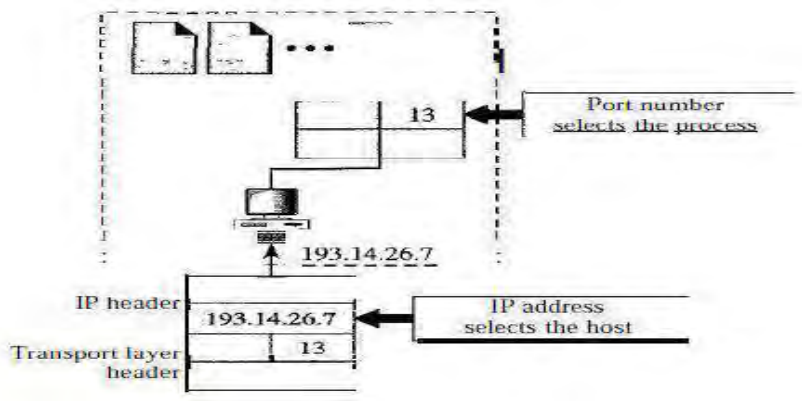
The lANA (Internet Assigned Number Authority) has divided the port numbers into three ranges: well **known, registered, and dynamic or private.**

* **Well-known ports**: The ports ranging from 0 to 1023 are assigned and controlled by lANA.

* **Registered ports**: The ports ranging from 1024 to 49,151 are not assigned or controlled by lANA. They can only be registered with lANA to prevent duplication.

* **Dynamic ports**: The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used by any process. These are the **ephemeral ports.**

***IP addresses versus port numbers***




**Socket Addresses**

Process-to-process delivery needs two identifiers, **IP address** and the port **number**, at each end to make a connection. The combination of an IP address and a port number is called a **socket address**. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely. A transport layer protocol needs a pair of socket addresses: the client socket address (IP address and port number) and the server socket address (IP address and port number). These four pieces of information are part of the IP header and the transport layer protocol header. The below figure showing socket address:

| value | In X client | In X server |
|---|---|---|
| IP address | 10.10.12.166 | 10.10.11.66 |
| Port | 32825 | 55555 |
| Socket address | 10.10.12.166:32825 | 10.10.11.66:55555 |

**Multiplexing and Demultiplexing:**

The addressing mechanism allows multiplexing and demultiplexing by the transport layer.

**Multiplexing:**

At the sender site, there may be several processes that need to send packets. However, there is only one transport layer protocol at any time. This is a **many-to-one** relationship and requires multiplexing. The protocol accepts messages from different processes, differentiated by their assigned

**port numbers**. After adding the header, the transport layer passes the packet to the network layer.

**Demultiplexing:**

At the receiver site, the relationship is one-to-many and requires demultiplexing. The transport layer receives datagram's from the network layer. After error checking and dropping of the header, the transport layer delivers each message to the appropriate process based on the port number.

**Connectionless Versus Connection-Oriented Service**

A transport layer protocol can either be connectionless or connection-oriented.

***Connectionless Service***

In a connectionless service, the packets are sent from one party to another with no need for connection establishment or connection release. The packets are not numbered therefore they may be delayed or lost or may arrive out of sequence. There is no acknowledgment either. UDP is an example of connectionless protocol.

***Connection Oriented Service***

In a connection-oriented service, a connection is first established between the sender and the receiver. Data are transferred. At the end, the connection is released. TCP and SCTP are example of connection-oriented protocols.

**Reliable Versus Unreliable**

- The transport layer service can be **reliable** or **unreliable**.
- If the application layer program needs reliability, we use a reliable transport layer protocol by implementing flow and error control at the transport layer. This may leads slower and more complex service.
- On the other hand, if the application program does not need reliability because it uses its own flow and error control mechanism or it needs fast service and does not demand flow and error control (real-time applications), then an unreliable protocol can be used.
- In the Internet, there are three common different transport layer protocols used, **UDP is connectionless and unreliable, TCP and SCTP** are **connection oriented** and **reliable**. These three can respond to the demands of the application layer programs or taking care of the application layer.

If the data link layer is reliable and has flow and error control, do we need this at the transport layer, too? The answer is **yes**. Reliability at the data link layer is between two nodes. We need reliability between two ends. Because the network layer in the Internet is unreliable (best-effort delivery), we need to implement reliability at the transport layer. The error control at the data link layer does not guarantee error control at the transport layer.

flow and error control in TCP is implemented by the sliding window protocol. The window, however, is character-oriented, instead of frame-oriented.

## USER DATAGRAM PROTOCOL (UDP)

- The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol.
- It only provide process to process communication to IP, also it performs very limited error checking.
- UDP is a very simple protocol using a minimum of overhead.
- If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP.

Although UDP is so powerless, why would a process want to use it? Because UDP is a very simple protocol use a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP.

### Well-Known Ports for UDP

| *Port* | *Protocol* | *Description* |
|---|---|---|
| **7** | Echo | Echoes a received datagram back to the sender |
| 9 | Discard | Discards any datagram that is received |
| 11 | users | Active users |
| 13 | **Daytime** | Returns the date and the time |
| 17 | Quote | Returns a quote of the day |
| 19 | Char gen | Returns a string of characters |

| | | |
|---|---|---|
| 53 | Name server | Domain Name Service |
| 67 | BOOTPs | Server port to download bootstrap information |
| 68 | BOOTPc | Client port to download bootstrap information |
| 69 | TFTP | Trivial File Transfer Protocol |
| 111 | RPC | Remote Procedure Call |
| 123 | NTP | Network Time Protocol |
| 61 | SNMP | Simple Network Management Protocol |
| 62 | SNMP | Simple Network Management Protocol (trap) |

*Example*
In UNIX, the well-known ports are stored in a file called fetcfservices. Each line in this file gives the name of the server and the well-known port number. We can use the *grep* utility to extract theline corresponding to the desired application. The following shows the port for FTP. Note that FTP can use port 21 with either UDP or TCP.

```
$grep ftp  fetclservices
ftp      21ftcp
fip      21iudp
```

SNMP uses two port numbers (161 and 162), each for a different purpose.

```
$grep          snmp      fetclservices
Snmp           161ftcp   #Simple Net Mgmt Proto
Snmp           161iudp   #Simple Net Mgmt Proto
Snmptrap       162tudp   #Traps for SNMP
```

**User Datagram**

UDP packets, called user datagram, have a fixed-size header of 8 bytes. The fields are as follows:

**Source port number:** This port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535. If the source host is the client (a client sending a request), then port number used is an **ephemeral port** number requested by the process and chosen by the UDP software running on the source host. If the source host is the server (a server sending a response), the port number, used is a **well-known port** number.

**Destination port number**: This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases, is a **well-known port number**. If the destination host is the client (a server sending a response), the port number, in most cases, is an **ephemeral port number**. In this case, the server copies the ephemeral port number it has received in the request packet.

**Length:** This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be much less because a UDP user datagram is stored in an IP datagram with a total length of 65,535 bytes.

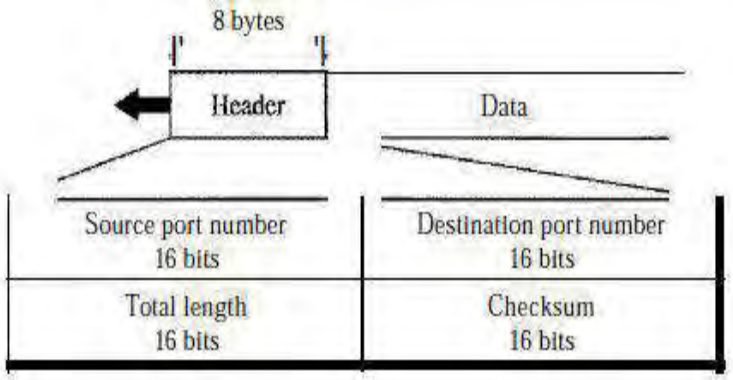
The length field in a UDP is actually not necessary, because a user datagram is encapsulated in an IP datagram. There is a field in the IP datagram that defines the total length. There is another field in the IP datagram that defines the length of the header. So if we subtract the value of the second field from the first, to get the UDP that is encapsulated in an IP datagram i.e:

**UDP length = IP length - IP header's length**

**Note that** that when the IP software delivers the UDP user datagram to the UDP layer, it has already dropped the IP header.

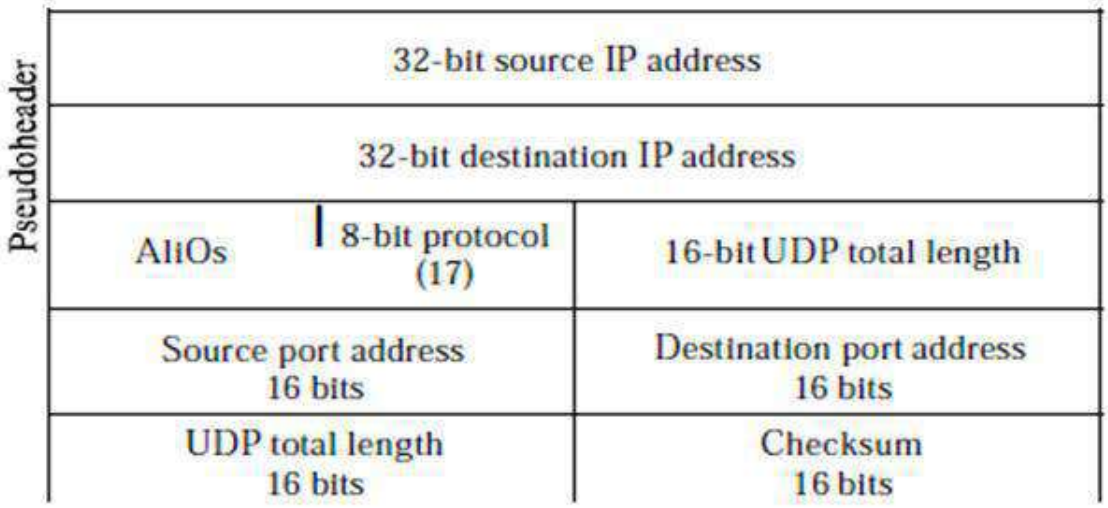
**Checksum:** This field is used to detect errors over the entire user datagram (header plus data).

***User datagram format***




**UDP checksum calculation**

The UDP checksum calculation is different from the one for IP and ICMP. The UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections: a **pseudo header, the UDP header, and the data** coming from the application layer. The pseudo header is the part of the header of the IP packet in which the user datagram is to be encapsulated with some fields filled with 0s.




If the checksum does not include the pseudo header, a user datagram may arrive safe and sound. However, if the IP header is corrupted, it may be delivered to the wrong host. The protocol field is added to ensure that the packet belongs to UDP,

and not to other transport-layer protocols. The value of the protocol field for UDP is 17. If this value is changed during transmission, the checksum calculation at the receiver will detect it and UDP drops the packet. It is not delivered to the wrong protocol. **Note** the similarities between the pseudo header fields and the last 12 bytes of the
IP header.

***LECTURE NOTE: 33***

# UDP Operation

UDP uses concepts common to the transport layer.

**Connectionless Services**

UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagram's even if they are coming from the same source process and going to the same destination program. The user datagram's are not numbered. There is no connection establishment and no connection termination, as is the case for TCP. It means each user datagram can travel on a different path.

**Flow and Error Control:**

UDP is a very simple, unreliable transport protocol. There is no flow control mechanism and hence no window mechanism. The receiver may overflow with incoming messages. There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a

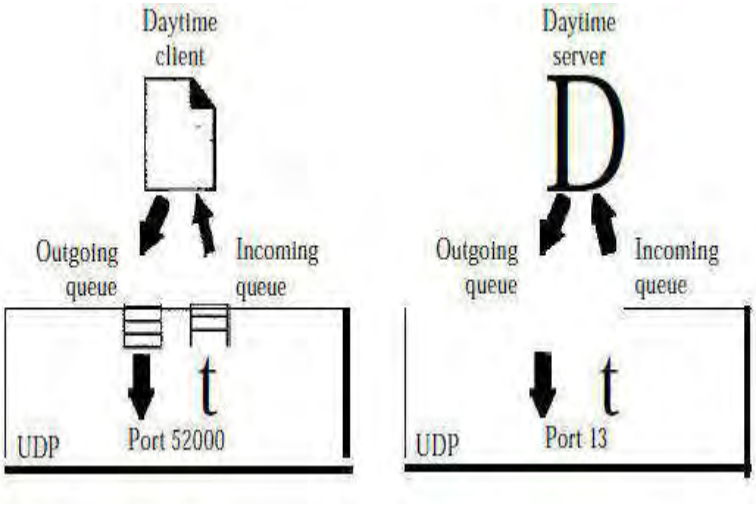
message has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded.

The lack of flow control and error control means that the process using UDP should provide these mechanisms.

**Encapsulation and Decapsulation**

To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

**Queuing**




In UDP, queues are associated with ports.

At the client site, when a process starts, it requests a port number from the operating system. Some implementations create both an incoming and an outgoing queue associated with each process. Other implementations create only an incoming queue associated with each process.

**Note** that even if a process wants to communicate with multiple processes, it obtains only one port number and eventually one outgoing and one incoming queue. The queues opened by the client are, in most cases, identified by **ephemeral port numbers**. The queues function as long as the process is running. When the process terminates, the queues are destroyed.

The client process send messages to the outgoing queue by using the source port number specified in the request. UDP removes the messages one by one and, after adding the UDP header, delivers them to IP. When an outgoing queue overflows operating system can ask the client process to wait before sending any more messages.

When a message arrives for a client, UDP checks if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a ***port unreachable*** message to the server. All the incoming messages for one particular client program, whether coming from the same or a different server, are sent to the same queue. If an incoming queue overflows UDP drops the user datagram and asks for a **port unreachable** message to be sent to the server.

- At the server site, the server asks for incoming and outgoing queues, using its **well-known port,** when it starts running. The queues remain open as long as the server is running.

- When a message arrives for a server, UDP checks to see if an incoming queue has been created for the port number specified in the destination port number field of the user datagram. If there is such a queue, UDP sends the received user datagram to the end of the queue. If there is no such queue, UDP discards the user datagram and asks the ICMP protocol to send a **port unreachable** message to the client. All the incoming messages for one particular server, whether coming from the same or a different client, are sent to the same queue. If an incoming queue overflows UDP drops the user datagram and asks for a **port unreachable** message to be sent to the client.

When a server wants to respond to a client, it sends messages to the outgoing queue, using the source port number specified in the request. UDP removes the messages one by one and, after adding the UDP header, delivers them to IP. If an outgoing queue overflows the operating system asks the server to wait before sending any more messages.

**Use of UDP**

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control.
- It is not usually used for a process such as FTP that needs to send bulk data.
- UDP is suitable for a process with internal flow and error control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control. It can easily use UDP.

- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for management processes such as SNMP.
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP).

**TCP**

- It is a transport layer protocol called Transmission Control Protocol (TCP).
- TCP, like UDP, is a process-to-process (program-to-program) protocol therefore it uses port numbers.
- It is called a ***connection-oriented, reliable*** transport protocol: it creates a virtual connection between two TCPs to send data.
- It adds Connection-oriented and reliability features to the services of IP.
- TCP uses flow and error control mechanisms at the transport level.
- It adds connection-oriented and reliability features to the services of IP.

**TCP Services**

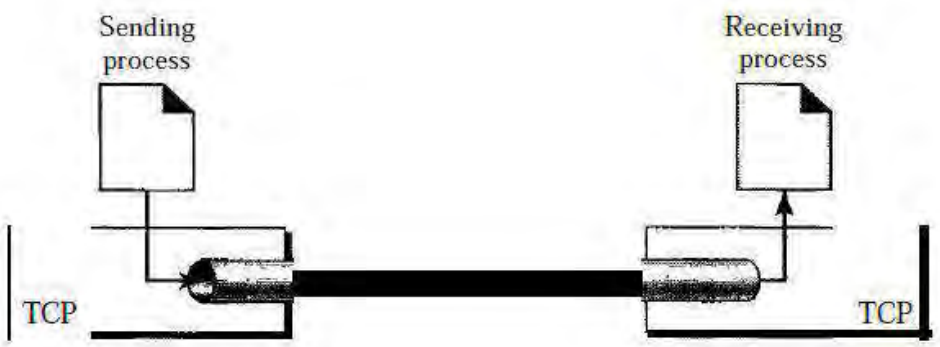
The services offered by TCP to the processes at the application layer are described below.

**Stream Delivery Service**

TCP, unlike UDP, is a stream-oriented protocol. TCP allows the sending process to deliver data as a **stream of bytes** and allows the receiving process to obtain data as a **stream of bytes**. TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their data across the Internet. The sending process produces (writes to) the stream of bytes, and the receiving process consumes (reads from) them.

Sending and Receiving Buffers: because the sending and the receiving processes may not write or read data at the same speed, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction.

***Stream delivery***




For simplicity, it has shown two buffers of 20 bytes each and is not always the case. But normally the buffers are hundreds or thousands of bytes, depending on the implementation.

***Sending and receiving buffers***

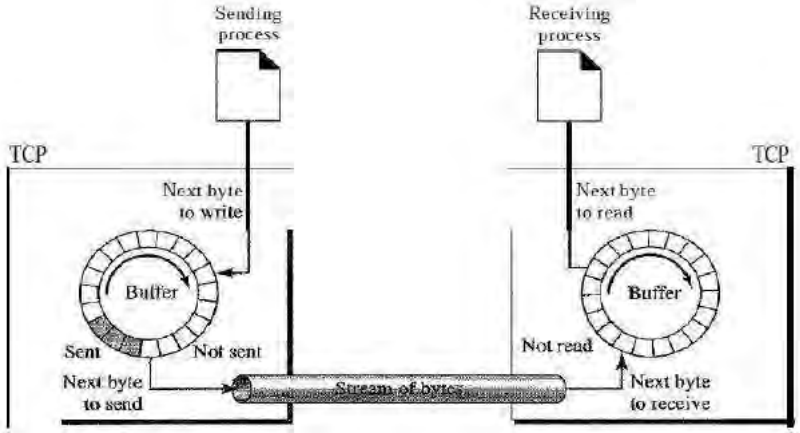



Figure above shows the movement of the data in one direction. At the sending site, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The gray area holds bytes that have been sent but not yet acknowledged. TCP keeps these bytes in the buffer until it receives an acknowledgment. The coloured area contains bytes to be sent. However, TCP may be able to send only part of this coloured section due to the slowness of the receiving process or perhaps to congestion in the network. **Note** that after the bytes in the gray chambers are acknowledged, the chambers are recycled and available for use by the sending process. This is why we show a circular buffer.

The operation of the buffer at the receiver site is simpler. The buffer is divided into two areas (shown as white and coloured). The white area contains empty chambers to be filled by bytes received from the network. The coloured sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

**Segments:** At the transport layer, TCP groups a number of bytes together into a packet called a segment. TCP adds a header to each segment (for control purposes) and delivers the segment to the IP layer for transmission. The segments are encapsulated in IP datagram's and transmitted. This entire operation is transparent to the receiving process.

***Reliable Service***

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.

***Process-to-Process Communication***

Like UDP, TCP provides process-to-process communication using port numbers. list of some well-known port numbers used by TCP are given below:

| *Port* | *Protocol* | *Description* |
|---|---|---|
| 7 | Echo | Echoes a received datagram back to the sender |
| 9 | Discard | Discards any datagram that is received |
| 11 | Users | Active users |
| 13 | Daytime | Returns the date and the time |
| 17 | Quote | Returns a quote of the day |
| 19 | Chargen | Returns a string of characters |
| 20 | FIP, Data | File Transfer Protocol (data connection) |
| 21 | FIP, Control | File Transfer Protocol (control connection) |
| 23 | TELNET | Tenninal Network |
| 25 | SMTP | Simple Mail Transfer Protocol |
| 53 | DNS | Domain Name Server |
| 67 | BOOTP | Bootstrap Protocol |
| 79 | Finger | Finger |
| 80 | HTTP | Hypertext Transfer Protocol |
| 111 | RPC | Remote Procedure Call |

**Full-Duplex Communication:**

TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions.

**Connection-Oriented Service**

TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two TCPs establish a connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

Note that this is a virtual connection, not a physical connection. The TCP segment is encapsulated in an IP datagram and can be sent out of order, or lost, or corrupted, and then resent. Each may use a different path to reach the destination. There is no physical connection. TCP creates a stream-oriented environment in which it accepts the responsibility of delivering the bytes in order to the other site. The situation is similar to creating a bridge that spans multiple islands and passing all the bytes from one island to another in one single connection.

***LECTURE NOTE: 34***

# TCP Features

TCP has several features that are briefly summarized as:

**Numbering System**

There are two fields called the sequence number and the acknowledgment number. These two fields refer to the byte number.

**Byte Number:** TCP numbers all data bytes that are transmitted in a connection. Numbering is independent in each direction. When TCP receives bytes of data from a process, it stores them in the sending buffer and numbers them. The numbering does not necessarily start from 0. Instead, TCP generates a random number between 0 and $2^{32 - 1}$ for the number of the first byte. For example, if the random number happens to be 1057 and the total data to be sent are 6000 bytes, the bytes are numbered from 1057 to 7056. We will see that byte numbering is used for flow and error control.

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

**Sequence Number:** After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment. The sequence number in each direction shows the number of the first byte carried by the segment.

---

The value in the sequence number field of a segment defines the number of the first data byte contained in that segment.

---

**Acknowledgment Number:** acknowledgment number defines the number of the next byte that the party expects to receive. In addition, the acknowledgment number is cumulative, which means that the party takes the number of the last byte that it has received, safe and sound, adds 1 to it, and announces this sum as the knowledgement number. The term ***cumulative*** here means that if a party uses 5643 as an acknowledgment number, it has received all bytes from the beginning up to 5642. **Note** that this does not mean that the party has received 5642 bytes because the first byte number does not have to start from 0.

---

The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

---

**Flow Control**

TCP, unlike UDP, provides *flow control.* The receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

**Error Control**

To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented.

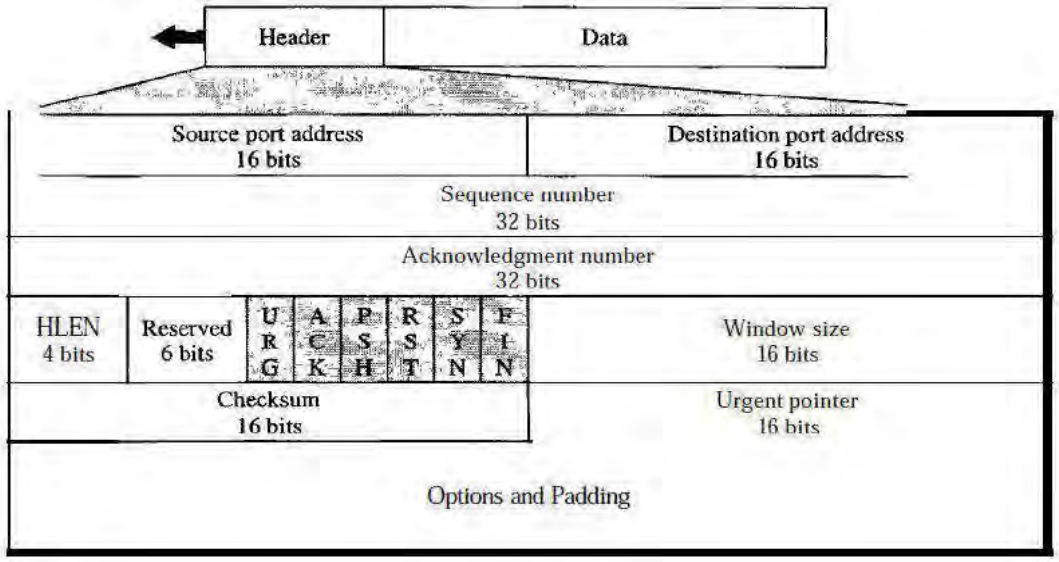
**Congestion Control**

TCP, unlike UDP, takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

**Segment:**

A packet in TCP is called a segment. The segment consists of a 20- to 60-byte header, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options. We will discuss some of the header fields in this section. The meaning and purpose of these will become clearer as we proceed through the chapter.

***TCP segment format***




**Source port address:** This is a 16-bit field that defines the port number of the application program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP header.

**Destination port address**: This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. This serves the same purpose as the destination port address in the UDP header.

**Sequence number:** This 32-bit field defines the number assigned to the first byte of data contained in this segment. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment. During connection establishment, each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.

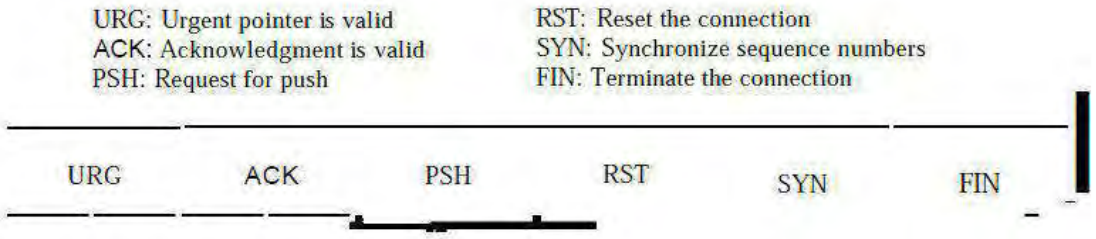
**Acknowledgment number:** This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number *x* from the other party, it defines *x* + 1 as the acknowledgment number. Acknowledgment and data can be piggybacked together.

**Header length**: This 4-bit field indicates the number of 4-byte words in the TCP header.
The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 (5 x 4 =20) and 15 (15 x 4 =60).

**Reserved**: This is a 6-bit field reserved for future use.

**Control:** This field defines 6 different control bits or flags as shown below in Figure one or more of these bits can be set at a time.

***Control field***




These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.

| *Flag* | *Description* |
|---|---|
| URG | The value of the urgent pointer field is valid. |
| ACK | The value of the acknowledgment field is valid. |
| PSH | Push the data. |
| RST | Reset the connection. |
| SYN | Synchronize sequence numbers during connection. |
| FIN | Terminate the connection. |

**Window size:** This field defines the size of the window, in bytes. **Note** that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

**Checksum:** This 16-bit field contains the checksum TCP follows the same procedure as the one described for UDP. Inclusion of the checksum for TCP is mandatory where in UDP optional. Pseudo header, serving the same purpose, here for TCP the value for the protocol field is 6.

**Urgent pointer:** This l6-bit field, which is valid only if the urgent flag is set, and is used when the segment contains urgent data.

**Options:** There can be up to 40 bytes of optional information in the TCP header.

***LECTURE NOTE: 35***

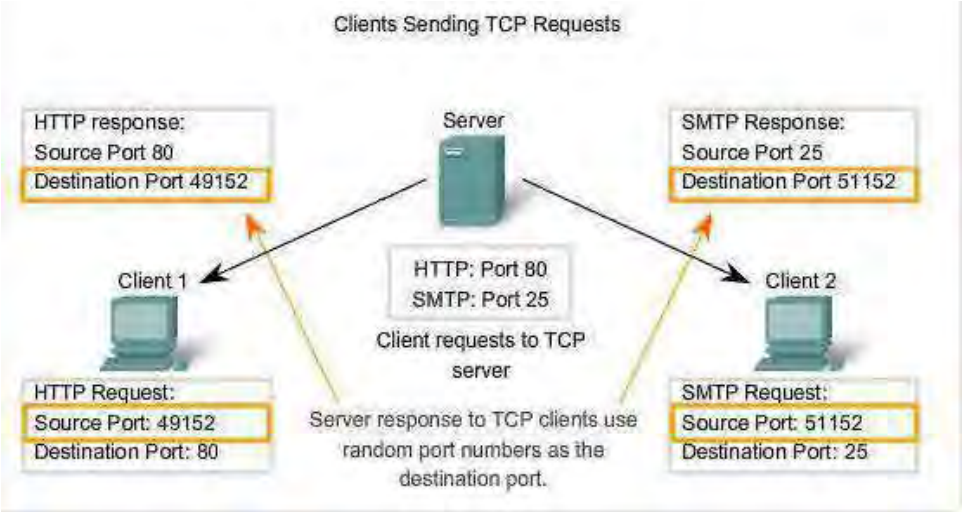
# TCP Connection

***TCP Server Processes:***

An application processes run on servers. These processes wait until a client initiates communication with a request for information or other services. Each application process running on the server is configured to use a port number, either by default or manually by a system administrator. **An individual server cannot have two services assigned to the same port number within the same Transport layer services.** The host running a web server application and a file transfer application cannot have both

configured to use the same port (for example, TCP port 8080). When an active server application is assigned to a specific port, that port is considered to be "open" on the server. Any incoming client request addressed to the correct socket is accepted and the data is passed to the server application. There can be many simultaneous ports open on a server, one for each active server application. It is common for a server to provide more than one service, such as a web server and an FTP server, at the same time.

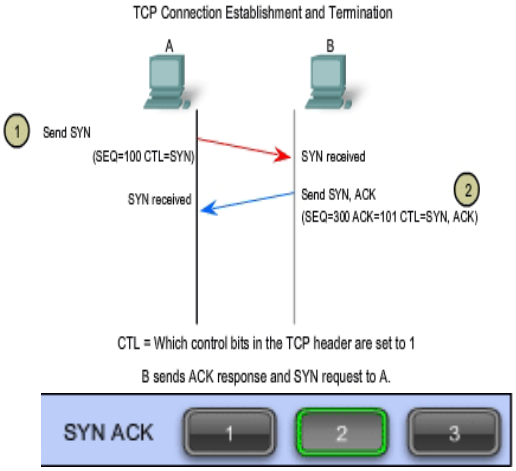
In order to improve security on a server, it is to restrict server access to only those ports associated with the services and applications should be accessible to authorized requestors. The following figure showing the typical allocation of source and destination ports in TCP client/server operations.

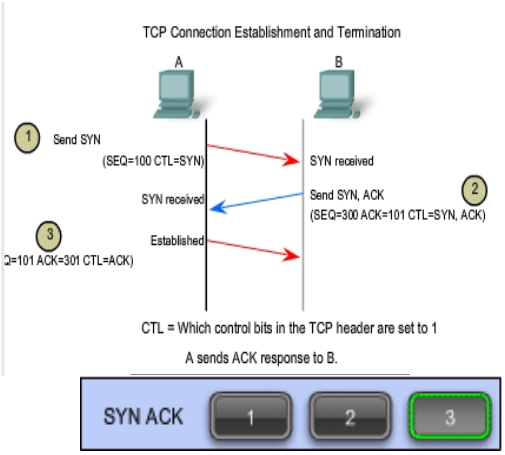


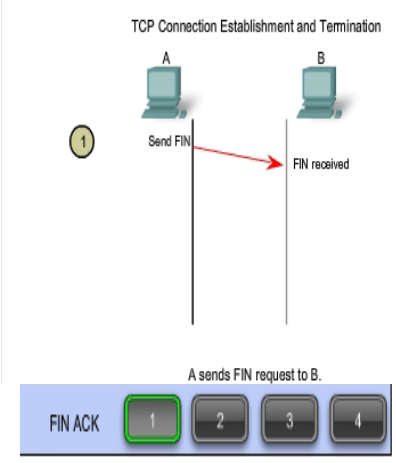

TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All the segments belonging to a message are then sent over this virtual path. Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

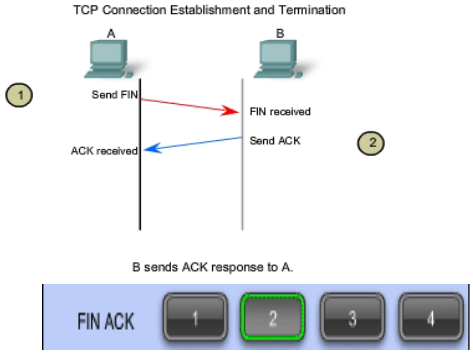
In TCP, connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

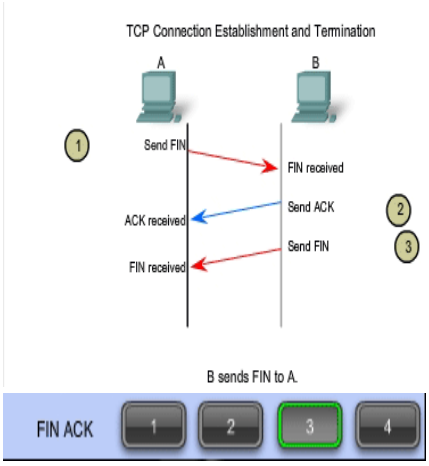
**Connection Establishment:**

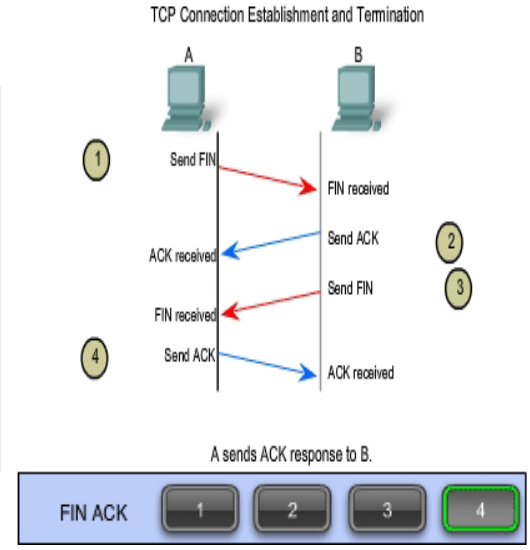

TCP Connection Establishment and Termination
A
B
1
Send SYN
(SEQ=100 CTL=SYN)
SYN received
SYN received
Send SYN, ACK
(SEQ=300 ACK=101 CTL=SYN, ACK)
2
CTL = Which control bits in the TCP header are set to 1
B sends ACK response and SYN request to A.
SYN ACK
1
2
3

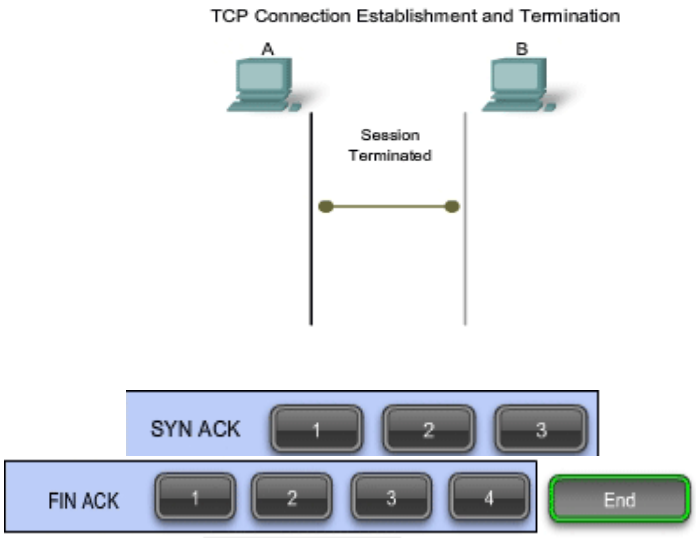

TCP Connection Establishment and Termination
A
B
1
Send SYN
(SEQ=100 CTL=SYN)
SYN received
SYN received
Send SYN, ACK
(SEQ=300 ACK=101 CTL=SYN, ACK)
2
3
Established
Q=101 ACK=301 CTL=ACK)
CTL = Which control bits in the TCP header are set to 1
A sends ACK response to B.
SYN ACK
1
2
3


TCP Connection Establishment and Termination
A
B
1
Send FIN
FIN received
A sends FIN request to B.
FIN ACK
1
2
3
4


TCP Connection Establishment and Termination
A
B
1
Send FIN
FIN received
ACK received
Send ACK
2
B sends ACK response to A.
FIN ACK
1
2
3
4


TCP Connection Establishment and Termination
A
B
1
Send FIN
FIN received
ACK received
Send ACK
2
Send FIN
3
FIN received
B sends FIN to A.
FIN ACK
1
2
3
4


TCP Connection Establishment and Termination
A
B
1
Send FIN
FIN received
ACK received
Send ACK
2
Send FIN
3
FIN received
4
Send ACK
ACK received
A sends ACK response to B.
FIN ACK
1
2
3
4



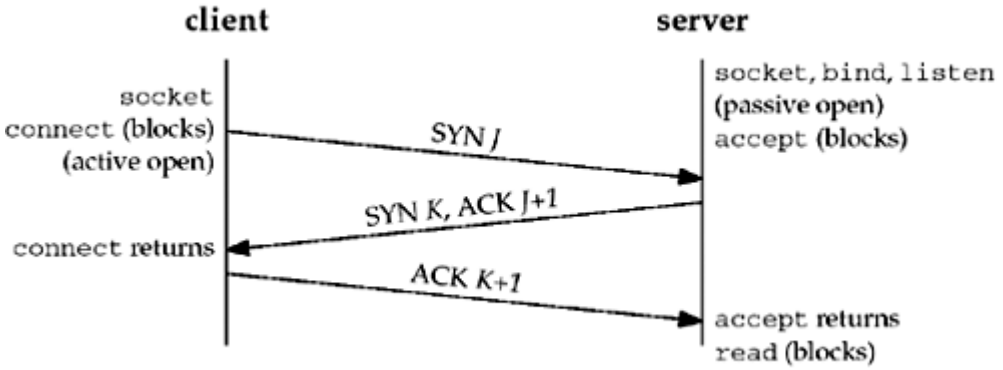

In TCP host communication, a connection is established before data can be exchanged. After the communication is completed, the sessions are closed and the connection is terminated. The connection and session mechanisms enable TCP's reliability function.

By using the information in the TCP header each host tracks each data segment within a session and exchanges information about what data is received by each host. Each connection represents **two one-way (one from the client to the server, and the other from the server to the client) communication streams, or sessions**. To establish the connection, the hosts perform a three-way handshake. Control bits in the TCP header indicate the progress and status of the connection.

The **three-way** handshake:

- Establishes that the destination device is present on the network
- Verifies that the destination device has an active service and is accepting requests on the destination port number that client intends to use for session
- Informs the destination device that the source client intends to establish a communication session on that port number

In TCP connections, client initiates the session to the server. Steps are:




1. The initiating client sends a segment containing an initial sequence value, which serves as a request to the server to begin a communications session.

2. The server responds with a segment containing an acknowledgement value equal to the received sequence value plus 1, plus its own synchronizing sequence value. The value is one greater than the sequence number because the ACK is always the next expected Byte or Octet.

3. Initiating client responds with an acknowledgement value equal to the sequence value it received plus one. This completes the process of establishing the connection.

Within the TCP segment header, there are six 1-bit fields that contain control information used to manage the TCP processes. Those fields are:

**URG** - Urgent pointer field significant

**ACK** - Acknowledgement field significant

**PSH** - Push function

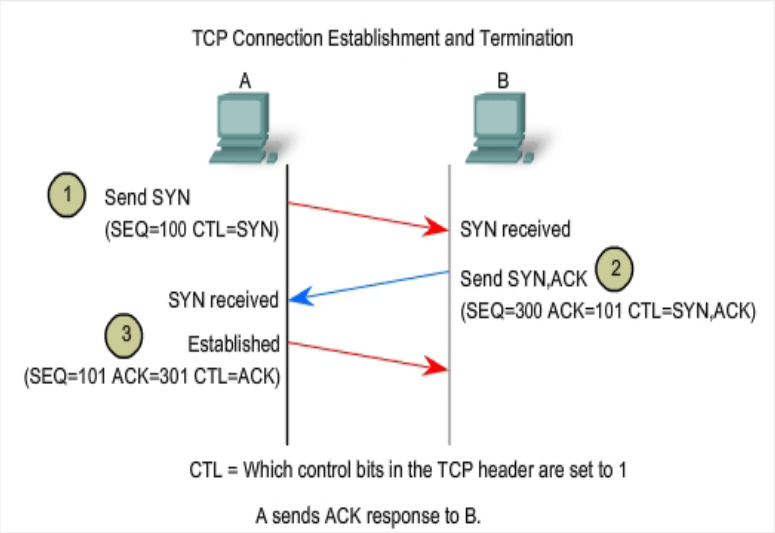
**RST** - Reset the connection

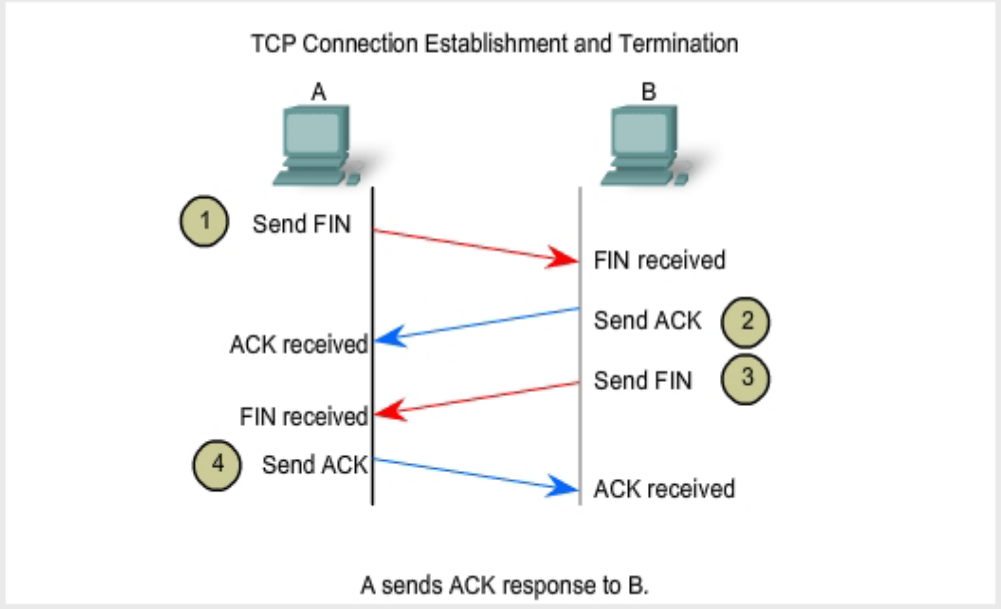
**SYN** - Synchronize sequence numbers

**FIN** - No more data from sender

These fields are referred to as **flags**, because the value of one of these fields is only 1 bit and, therefore, has only two values: **1 or 0**. When a bit value is set to 1, it indicates what control information is contained in the segment.

Using a four-step process, flags are exchanged to terminate a TCP connection as:




**Step 1**

A TCP client begins by sending a segment with the SYN (Synchronize Sequence Number) control flag set, indicating an initial value in the sequence number field in

the header. This initial value for the sequence number, known as the Initial Sequence Number (ISN), which is randomly chosen and it is used to begin tracking the flow of data from the client to the server for this session. The ISN is increased by one for each byte of data sent from the client to the server as the data conversation continues.

**Step 2**

The TCP server acknowledged the receipt of the SYN segment from the client to establish the session from the **client to the server**. For that the server sends a segment back to the client with the **ACK flag set** indicating that the Acknowledgment number is significant. With this flag set in the segment, the client recognizes this as an acknowledgement that the server received the SYN from the TCP client. The value of the acknowledgment number field is equal to the client initial sequence number plus 1. This establishes a session from the client to the server. This establishes a session from the client to the server. To start this session, the server uses the SYN flag in the same way that the client did. It sets the SYN control flag in the header to establish a session from the server to the client. The **SYN flag** indicates that the initial value of the sequence number field is in the header. This value used to track the flow of data in this session from the server back to the client.

**Step 3**

Finally, the TCP client responds with a segment containing an **ACK** that is the response to the TCP SYN sent by the server. There is no user data in this segment. The value in the acknowledgment number field contains one more than the initial sequence number received from the server. Once both sessions are established between client and server, segments exchanged in this communication takes place and will have additional ACK flag set.

> **Data Transfer:**
> After connection is established, bidirectional data transfer can take place. The client and server can both send data and acknowledgments. Data traveling in the same direction as an acknowledgment are carried on the same segment. The acknowledgment is piggybacked with the data.

Security can be added to the data network by:

- Denying the establishment of TCP sessions
- Only allowing sessions to be established for specific services

- Only allowing traffic or channel as a part of already established sessions

  This security can be implemented for all TCP sessions or only for selected sessions.

***TCP Session Termination***

To close a connection, the FIN (Finish) control flag in the segment header must be set. To end each one-way TCP session, a two-way handshake, consisting of a FIN segment and an ACK segment is used. Therefore, to terminate a single conversation supported by TCP, four exchanges are needed to end both sessions. **Note**: the client and server are used in this description as a reference, but the termination process can be initiated by any two hosts that complete the session:

1. When the client has no more data to send in the stream, it sends a segment with the FIN flag set.

2. The server sends an ACK to acknowledge the receipt of the FIN to terminate the session from client to server.

3. The server sends a FIN to the client, to terminate the server to client session.

4. The client responds with an ACK to acknowledge the FIN from the server.

When the client end of the session has no more data to transfer, it sets the FIN flag in the header of a segment. Next, at the end of the connection the server will send a normal segment containing data with the **ACK flag** set using the acknowledgment number, it confirms that all the bytes of data have been received. When all segments have been acknowledged, the session is closed. The session in the other direction is closed using the same process. The receiver indicates that there is no more data to send by setting the FIN flag in the header of a segment sent to the source. A return acknowledgement confirms that all bytes of data have been received and that session is, in turn, closed. As shown in the above figure, the FIN and ACK control flags are set in the segment header, thereby closing a HTTP session.

It is also possible to terminate the connection by a three-way handshake. When the client has no more data to send, it sends a FIN to the server. If the server also has

no more data to send, it can reply with both the FIN and ACK flags set, combining two steps into one. The client replies with an ACK.

TCP transmits data in full-duplex mode. When two TCPs in two machines are connected, they are able to send segments to each other simultaneously. This implies that each party must initialize communication and get approval from the other party before any data are transferred. **Three-Way Handshaking:** The connection establishment in TCP is called three way handshaking. Here an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol. The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This is called a request for a ***passive open***. Although the server TCP is ready to accept any connection from any machine in the world, it cannot make the connection itself. The client program issues a request for an ***active open***. A client that wishes to connect to an open server tells its TCP that it needs to be connected to that particular server. TCP can now start the three-way handshaking process that is shown above.

**Connection Termination:**

Any of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client. Most implementations today allow two options for connection termination: three-way handshaking and four-way with a half-close option.

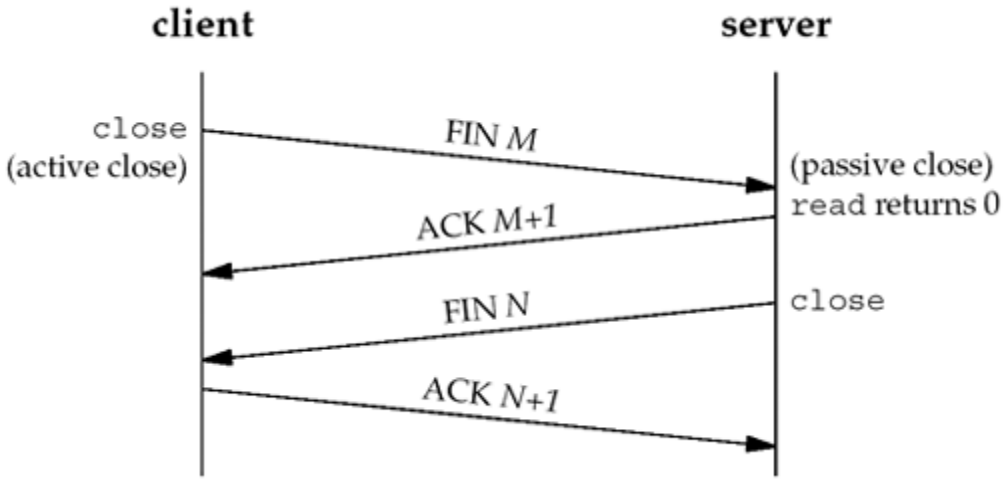
Three-Way Handshaking for connection termination as shown in figure:

1. Client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment (chunk of data sent by the client) in which the FIN flag is set.

---

The FIN segment consumes one sequence number ifit does not carry data.

---




2. The server TCP, after receiving the FIN segment, informs its process and sends the second segment, a FIN +ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction. This segment also contains the last chunk of data from the server. If it does not carry data, it consumes only one sequence number.

The FIN + ACK segment consumes one sequence number if it does not carry data.

2. The client TCP sends the last segment that is an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is 1 plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

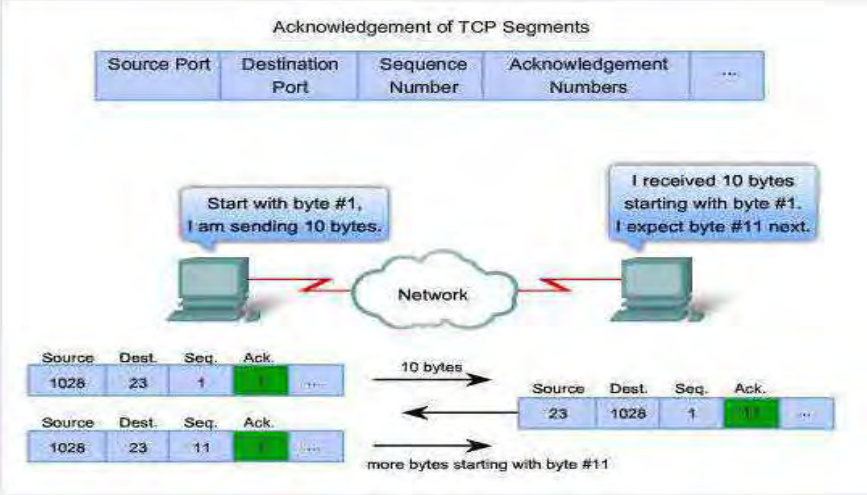
**Half-Close:** In TCP, one end can stop sending data while still receiving data. This is called a half-close. Although either end can issue a half-close, it is normally initiated by the client. It can occur when the server needs all the data before processing can begin. A good example is **sorting**.
When the client sends data to the server to be sorted, the server needs to receive all the data before sorting can start. This means the client, after sending all the data, can close the connection in the outbound direction. However, the inbound direction must remain open to receive the sorted data. The server, after receiving the data, still needs time for sorting; its outbound direction must remain open.

**Managing TCP Sessions**

***TCP Segment Reassembly:*** Sequence numbers are assigned in the header of each packet to achieve this goal of reorder and reassembling at the destination. During session setup, an initial sequence number (ISN) is set. This initial sequence number represents the starting value for the bytes for this session that will be transmitted to the receiving application. As data is transmitted during the session, the sequence number is incremented by the number of bytes that have been transmitted. This tracking of data byte enables each segment to be uniquely identified and acknowledged. Missing segments can be identified. Segment sequence numbers **enable reliability** by indicating how to reassemble and reorder received segments.

***TCP Acknowledgement with Windowing:*** **Confirming Receipt of Segments**

The segment header sequence number and acknowledgement number are used together to confirm receipt of the bytes of data contained in the segments.




The sequence number is the relative number of bytes that have been transmitted in this session plus 1 (which is the number of the first data byte in the current segment). TCP uses the acknowledgement number in segments sent back to the source to indicate the next byte in this session that the receiver expects to receive. This is called ***expectation acknowledgement.***

***TCP Retransmission:*** **Handling Segment Loss**

TCP provides methods of managing the segment losses. It provides a mechanism to retransmit segments with unacknowledged data. A destination host service using TCP usually only acknowledges data for contiguous sequence bytes. If one or more segments are missing, only the data in the segments that complete the stream are acknowledged. For example, if segments with sequence numbers 1500 to 3000 and 3400 to 3500 were received, the acknowledgement number would be 3001.

This is because there are segments with the sequence numbers 3001 to 3399 that have not been received. When TCP at the source host has not received an acknowledgement time interval (predetermined time) it will go back to the last acknowledgement number that it received and retransmit data from that point forward.

***TCP Congestion Control - Minimizing Segment Loss : flow control***

TCP provides mechanisms for flow control. Flow control assists the reliability of TCP transmission by adjusting the effective rate of data flow between the two services in the session. When the source is informed that the specified amount of data in the segments is received, it can continue sending more data for this session. This Window Size field in the TCP header specifies the amount of data that can be transmitted before an acknowledgement must be received. The initial window size is determined during the session start up via the three-way handshake. TCP feedback mechanism adjusts the effective rate of data transmission to the maximum flow that the network and destination device can support without loss. TCP attempts to manage the rate of transmission so that all data will be received and retransmissions will be minimized.

**Error Control**

TCP is a reliable transport layer protocol. This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.

TCP provides reliability using error control. Error control includes mechanisms for detecting corrupted segments, lost segments, out-of-order segments, and duplicated segments. Error control also includes a mechanism for correcting errors after they are detected. Error detection and correction in TCP is achieved through the use of three simple tools: checksum, acknowledgment, and time-out.

**Checksum**

Each segment includes a checksum field which is used to check for a corrupted segment. If the segment is corrupted, it is discarded by the destination TCP and is considered as lost. TCP uses a 16-bit checksum that is mandatory in every segment. In that the 16-bit checksum is considered

inadequate for the new transport layer, SCTP. However, it cannot be changed for TCP because this would involve reconfiguration of the entire header format.

***Acknowledgment***

TCP uses acknowledgments to confirm the receipt of data segments. Control segments that carry no data but consume a sequence number are also acknowledged. ACK segments are never acknowledged.

***LECTURE NOTE: 36***

# CONGESTION CONTROL AND QUALITY OF SERVICE

Congestion control and quality of service are two issues related to the three layers: the data link layer, the network layer, and the transport layer.

**CONGESTION: it is an** important issue in a packet-switched network. Congestion in a network may occur if the **load** on the network-the number of packets sent to the network-is greater than the *capacity* of the network (the number of packets a network can handle).

**Congestion control** refers to the mechanisms and techniques to control the congestion and keep the load below the capacity. Congestion happens in any system that involves waiting. For example, congestion happens on a freeway because any abnormality in the flow, such as an accident during rush hour, creates blockage. Congestion in a network (internetwork) occurs because routers and switches have queues-buffers that hold the packets before and after processing.

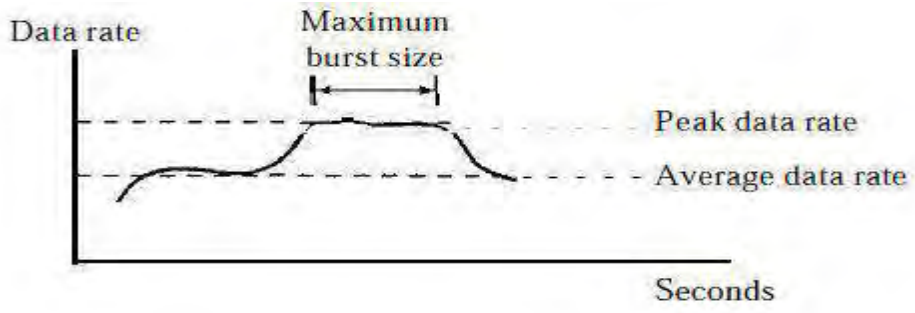
## DATA TRAFFIC

The main focus of **congestion control** and **quality of service** is **data traffic**. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic. So, before discussing about congestion control and quality of service, we describe data traffic as given below:

### Traffic Descriptor

Traffic descriptors are qualitative values that represent a data flow. Below Figure shows a traffic flow with some of these values.

Fig: ***Traffic descriptors***




***Average Data Rate***

The average data rate is the number of bits sent during a period of time, divided by the number of seconds in that period. We use the following equation:

Average data rate = amount of data / Time

The average data rate is a very useful characteristic of traffic because, it indicates the average bandwidth needed by the traffic.

***Peak Data Rate***

The peak data rate defines the maximum data rate of the traffic. In the figure of **traffic descriptor (above),** it is the maximum *y* axis value. The peak data rate is a very important measurement because it indicates the peak bandwidth that the network needs for traffic to pass through without changing its data flow.

***MAXIMUM BURST SIZE***

**The maximum burst size normally refers to the maximum length of time the traffic is generated at the peak rate.** Although the peak data rate is a critical value for the network, it can be ignored if the duration of the peak value is very

short. For example, if data are flowing steadily at the rate of 1 Mbps with a sudden peak data rate of 2 Mbps for just 1 ms, the network probably can handle the situation. However, if the peak data rate lasts 60 ms, there may be a problem for the network.

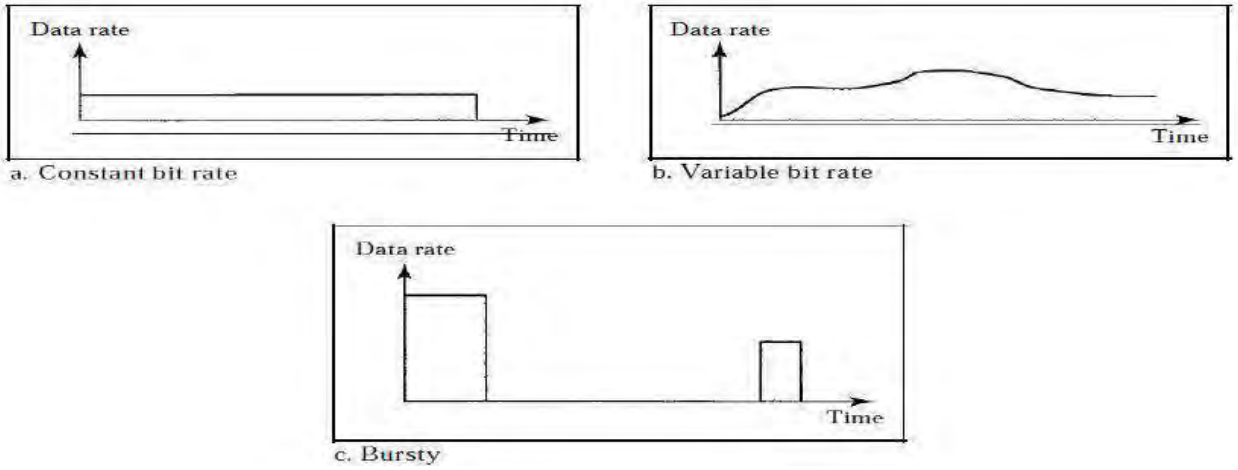
***Effective Bandwidth***

The effective bandwidth is the bandwidth that the network needs to allocate for the flow of traffic. The effective bandwidth is a function of three values: **average data rate, peak data rate, and maximum burst size**. The calculation of this value is very complex.

**Traffic Profiles**

The data flow can have one of the following traffic profiles: **constant bitrate**, **variable bit rate**, or **bursty** as shown below:

**Fig:*Three traffic profiles***




**i) *Constant Bit Rate***

A constant-bit-rate (CBR), or a fixed-rate, traffic model has a data rate that does not change. In this type of flow, the average data rate and the peak data rate are the same. The maximum burst size is not applicable. This type of traffic is very easy for a network to handle since it is predictable. The network knows in advance how much bandwidth to allocate for this type of flow.

ii) ***Variable Bit Rate***

In the variable-bit-rate (VBR) category, the rate of the data flow changes in time, with the smooth changes instead of sudden and sharp. In this type of flow, the average data rate and the peak data rate are different. The maximum burst size is usually a small value. This type of traffic is more difficult to handle than constant-bit-rate traffic, but it normally does not need to be reshaped.

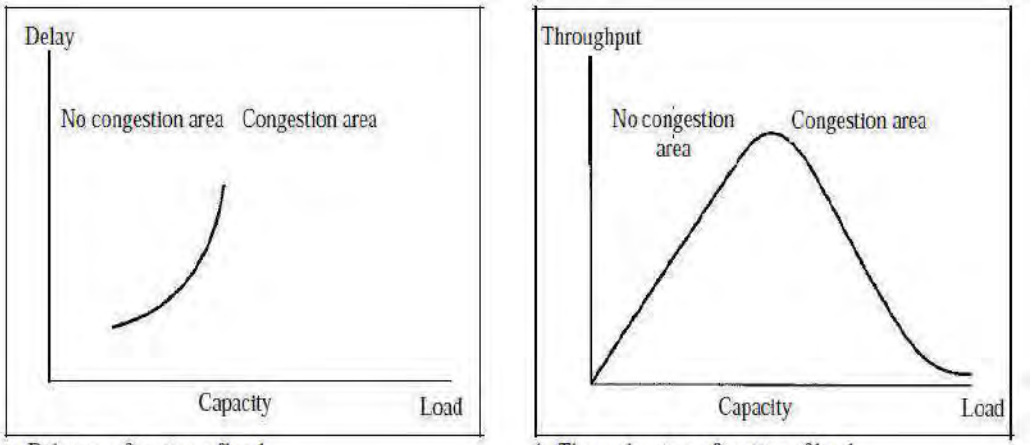
***Bursty***

In this category, the data rate changes suddenly in a very short time. It may jump from zero, for example, to 1 Mbps in a few microseconds and vice versa. The average bit rate and the peak bit rate are very different values in this type of flow. The maximum burst size is significant. This is the most difficult type of traffic for a network to handle because the profile is very unpredictable. To handle burst traffic, the network normally needs to reshape it, using reshaping techniques. Bursty traffic is one of the main causes of **congestion** in a network.

**Network Performance**

Congestion control involves two factors that measure the performance of a network: ***delay***
and ***throughput.***

***Packet delay and throughput as functions of load***




a. Delay as a function ofload

b. Throughput as a function of load

***Delay Versus Load***

Note that when the load is much less than the capacity of the network, the delay is at a minimum. This minimum delay is composed of propagation delay and processing delay, both of which are negligible. When the load reaches network capacity, the delay increases sharply because, waiting time in the queues (for all routers in the path) is added to the total delay. **Note** that the delay becomes infinite when the load is greater than the capacity.

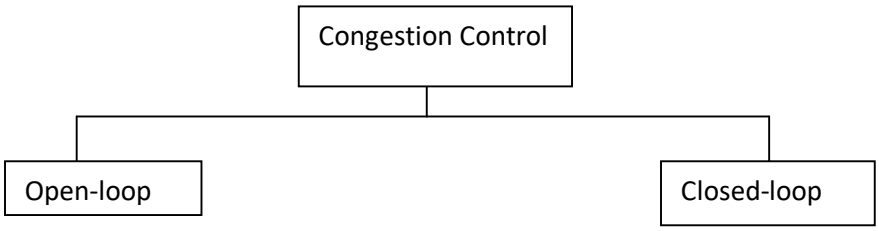
***Throughput Versus Load***

Throughput is the number of bits passing through a point in a second. We can define throughput in a network as the number of packets passing through the network in a unit of time. when the load is below the capacity of the network, the throughput increases proportionally with the *load.* When the load exceeds the

capacity, the queues become full and the routers have to discard some packets. Discarding packet, does not reduce the number of packets in the network because the sources retransmit the packets, using time-out mechanisms, when the packets has not reach the destinations.

## CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal).




1. *Retransmission Policy*
2. *Window Policy*
3. *Acknowledgment Policy*
4. *Discarding Policy*
5. *Admission Policy*

1. *Backpressure*
2. *Choke Packet*
3. *Implicit Signaling*
4. *Explicit Signaling*

### Open-Loop Congestion Control

In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination. List of policies that can prevent congestion are as given below:

1. **Retransmission Policy**

Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet is retransmitted. Retransmission

in may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion. For example, the retransmission policy used by TCP is designed to prevent or alleviate congestion.

2. **Window Policy**

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the *Go-Back-N* window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, tries to send only the specific packets that have been lost or corrupted.

**3. Acknowledgment Policy**

The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion. Several approaches are used in this case are: A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A receiver may decide to acknowledge only *N* packets at a time. Sending fewer acknowledgments means imposing fewer loads on the network.

**4. Discarding Policy**

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

**5. Admission Policy**

An admission policy is a quality-of-service mechanism. It can also prevent congestion in virtual-circuit networks. Switches in a flow, first check the resource requirement of a flow before admitting it to the network. A

router can deny establishing a virtual circuit connection, if there is congestion in the network or if there is a possibility of future congestion.

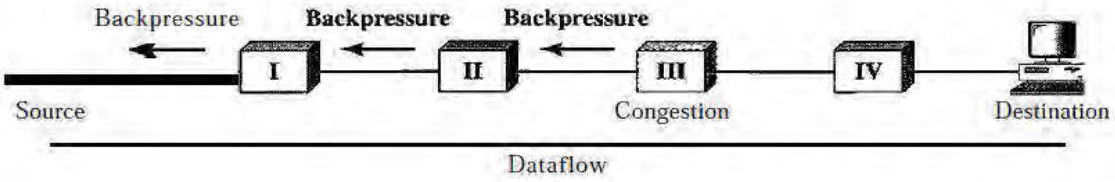
**Close Loop Congestion Control**

Closed-Loop Congestion Control: Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols. We describe a few of them here.

1. **Backpressure**

The technique of ***backpressure*** refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream nodes or nodes. And so on. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to **virtual circuit networks**, in which each node knows the upstream node from which a flow of data is coming.

***Backpressure method for alleviating congestion***




Node III in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node II to slow down the forwarding. Node II, in turn, may be congested because it is slowing down the output flow of data. If node II is congested, it informs node I to slow down, which in turn may create congestion. If so, node I inform the source of data to slow down. In this way alleviates the congestion. Note that the ***pressure*** on node III is moved backward to the source to remove the congestion.

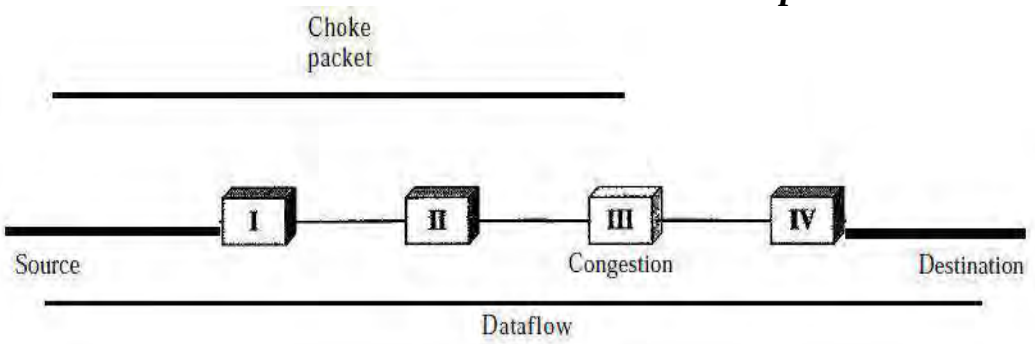
None of the virtual-circuit networks use backpressure. It is only implemented in the first virtual-circuit network, X.25. This technique cannot be implemented in a

datagram network because in this type of network, a node (router) does not have the slightest knowledge of the upstream router.

**2. Choke Packet:**

A choke packet is a packet sent by a node to the source to inform it of congestion. **Note** the difference between the backpressure and choke packet methods. In backpressure, the warning is from one node to its upstream node, the warning may eventually reach the source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned. We have seen example of this type of control in ICMP. It informs the source host, using a source quench ICMP message. The warning message goes directly to the source station, the intermediate routers, and does not take any action.

**Idea of a *Choke packet***




**3. Implicit Signaling**

In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network, the source should slow down.

**4. Explicit Signaling**

The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose. In the explicit signaling method, the signal is included in the packets that carry data**. Explicit signaling**, as in **Frame Relay congestion control**, can occur in either the forward or the backward direction.

**Backward Signaling:** A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down, to avoid the discarding of packets.

**Forward Signaling:** A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

## QUALITY OF SERVICE

We can informally define quality of service as **something a flow seeks to attain**.

### Flow Characteristics

Traditionally, four types of characteristics are attributed to a flow:

1. **reliability**
2. **delay**
3. **jitter**
4. **Bandwidth**

#### 1. Reliability

Reliability means safe and sound packet. Reliability is a characteristic that a flow needs. The Lack of reliability means, losing a packet or acknowledgment, which entails retransmission. However, the sensitivity of application programs to reliability is not the same. For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

**2. Delay**

When packet is not transmitted with real time then delay factor is arises. Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees. In this case, telephony, audio conferencing, video conferencing, and remote log-in need minimum delay, while delay in file transfer or e-mail is less important.

**3. Jitter**

Jitter is the variation in delay for packets belonging to the same flow. Jitter is defined as the variation in the packet delay. High jitter means the difference between delays is large, low jitter means the variation is small.For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time.

On the other hand, if the above four packets arrive at 21, 23, 21, and 28, they will have different delays: 21, 22, 19, and 24.

For audio and video applications, the **first** case is completely acceptable, the **second** case is not. For these applications, it does not matter if the packets arrive with a short or long delay as long as the delay is the same for all packets. The multimedia communication deals with jitter. If the jitter is high, some action is needed in order to use the received data.

**4. Bandwidth**

Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.

5. **Flow Classes**

Based on the flow characteristics, we can classify flows into groups, with each group having similar levels of characteristics. This categorization is not formal or universal; some protocols such as ATM have defined classes.

**Techniques to Improve QOS**

Some techniques that can be used to improve the quality of service and common four

Methods are:

- **scheduling**
- **traffic shaping**
- **admission control**
- **Resource reservation.**

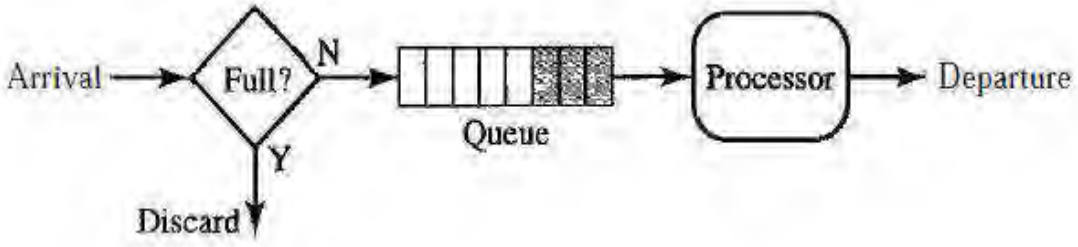
**Scheduling**:

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Several scheduling techniques are designed to improve the quality of service among that common three are: **FIFO queuing**, **priority queuing**, and **weighted fair queuing**.

***i) FIFO Queuing***

In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them. If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded. A FIFO queue is familiar to those who have had to wait for a bus at a bus stop.

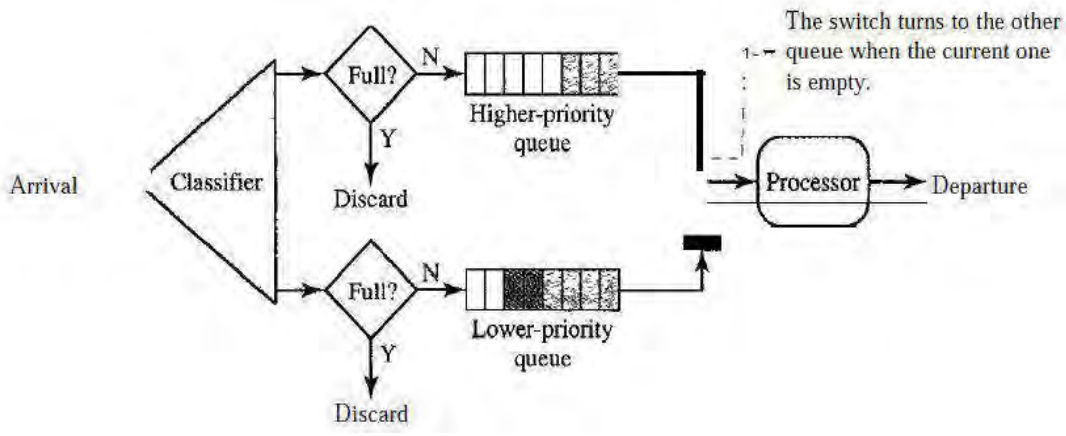
**Conceptual view of a FIFO queue**




***ii) Priority Queuing***

In priority queuing, packets are first assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last. **Note** that the system does not stop serving a queue until it is empty.

**Priority queuing with two priority levels**

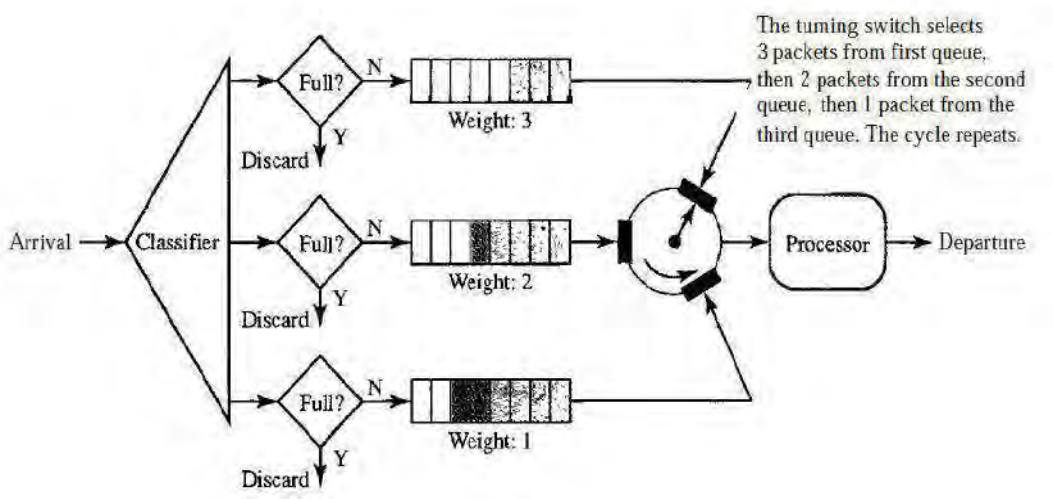



A priority queue can provide better QoS than the FIFO queue because higher priority traffic, such as multimedia, can reach the destination with less delay. However, there is a potential **drawback**. If there is a continuous flow in a high-priority queue, the packets in the
Lower priority queues will never have a chance to be processed. This condition is called **starvation.**

**iii) Weighted Fair Queuing**

It is a better scheduling method. In this technique, the packets are assigned to different classes and admitted to different queues. The queues, however, are weighted based on the priority of the queues. Higher priority means a higher weight. The system processes packets in each queue in a **round-robin fashion** with the number of packets selected from each queue based on the corresponding weight. For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue. If the system does not impose priority on the classes, all weights can be equal. In this way, we have fair queuing with priority.

**Figure: The technique with three classes**




## 2. Traffic Shaping

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: **leaky bucket and token bucket.**

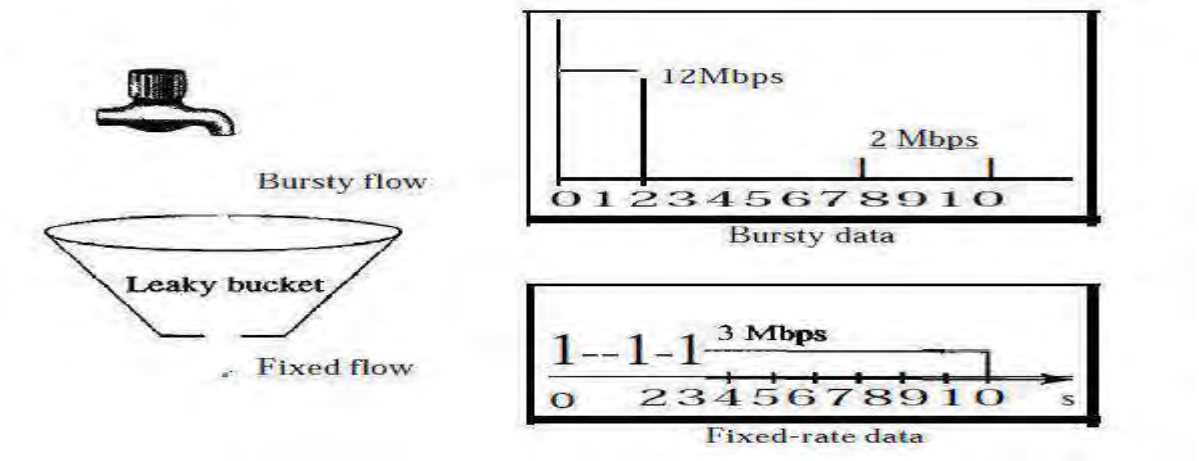
### i) *Leaky Bucket*

If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant Rate as long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty.

The input rate can **vary**, but the **output rate remains constant**. Similarly, in networking,a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.

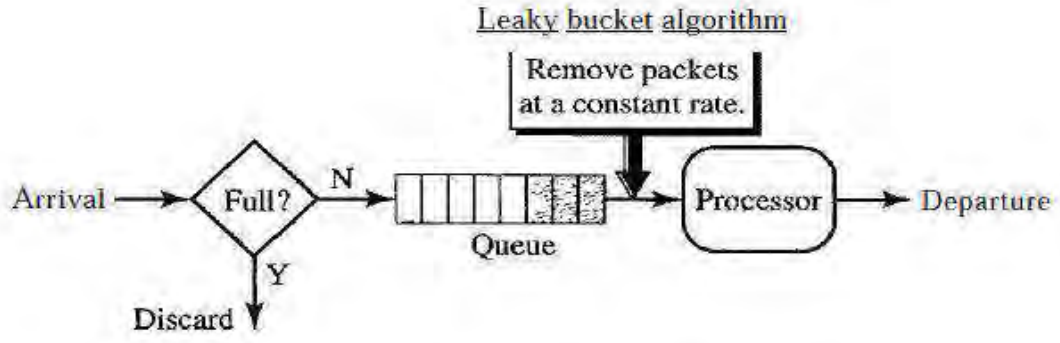
In the below figure it is assumed that the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucket shapes the input traffic to make it conform to this commitment.

Here the host sends a burst of data, at a rate of 12 Mbps for 2 s, for a total of 24(12x2) Mbits of data. The host is silent for 5s and then sends data at a rate of 2 Mbps for 3s, for a total of 6(2x3) Mbits of data. In all, the host has sent 30(24+6) Mbits of data in 10s. The leaky bucket smooth's the traffic by sending out data at a rate of 3 Mbps during the same 10s. Without the **leaky bucket**, the beginning burst may have hurt the network by consuming more bandwidth than is set aside for this host. Leaky bucket may also prevent congestion.




A simple leaky bucket implementation is shown in the below Figure. A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock.

***Leaky bucket implementation***




If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or bits.

The following is an algorithm for variable-length packets:
1. Initialize a counter to *n* at the tick of the clock.
2. If *n* is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until *n* is smaller than the packet size.
3. Reset the counter and go to step 1.

A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

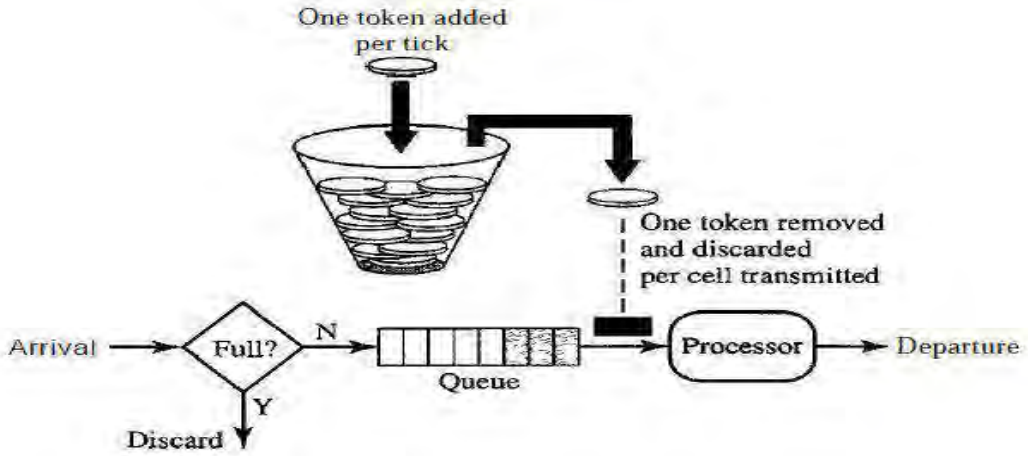
***ii) Token Bucket***

The leaky bucket is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty. Now if the host has bursty data, the leaky bucket allows only an **average rate**. The time when the host was **idle** is not taken into account. On the other hand, the **token bucket** algorithm allows idle hosts to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends *n* tokens to the bucket. The system removes one token for every cell (or byte) of data sent. For example, if *n* is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens. Now the host can consume all these tokens in one tick with 10,000 cells, or the host takes 1000 ticks with 10 cells per tick. In other words, the host can send bursty data as long as the bucket is not empty.

The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1. When the counter is zero, the host cannot send data.

The token bucket allows bursty traffic at a regulated maximum rate.

**Fig: Concept of token bucket**




***Combining Token Bucket and Leaky Bucket***

The two techniques can be combined to credit an **idle host** and at the same time regulate the traffic. The leaky bucket is applied after the token bucket: the rate of the leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

**Resource Reservation**

A flow of data needs resources such as a **buffer**, **bandwidth**, **CPU time**, and so on. The quality of service is improved if these resources are reserved beforehand. One of the QoS model called **Integrated Services**, which depends heavily on resource reservation to improve the quality of service.

**Admission Control**

Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, **buffer size**, **CPU speed**, etc.) and its previous commitments to other flows can handle the new flow.

## INTEGRATED SERVICES

Two models have been designed to provide quality of service in the Internet: **Integrated Services** and **Differentiated Services**. Both models emphasize the use of quality of service at the network layer (IP), although the model can also be used in other layers such as the data link.

**Integrated Services**, sometimes called **IntServe**, is *a flow-based* QoS model, which means that a user needs to create a flow, a kind of virtual circuit, from the source to the destination and inform all routers of the resource requirement.

Integrated Services is a *flow-based* QoS model designed for IP.

**Signaling**

IP is a connectionless, datagram, packet-switching protocol. To implement a flow-based model over this connectionless protocol, signalling mechanism is used for making a reservation. For this purpose protocols run over the IP called Resource Reservation Protocol (RSVP).

**Flow Specification**

When a source makes a reservation, it needs to define a flow specification. A flow specification has two parts: **Rspec (resource specification)** and Tspec **(traffic specification)**. Rspec defines the resource ((**buffer, bandwidth, etc**.) that the flow needs to reserve. Tspec defines the traffic characterization of the flow.

**Admission**

After a router receives the flow specification from an application, it decides to admit or deny the service. The decision is based on the previous commitments of the router and the current availability of the resource.

**Service Classes**

Two classes of services have been defined for Integrated Services: **guaranteed service**
and **controlled-load service**.

***Guaranteed Service Class***

Guaranteed services are quantitative services, in which the amount of end-to-end delay and the data rate must be defined by the application. This type of service is designed for real-time traffic that needs a guaranteed minimum end-to-end delay. The end-to-end delay is the sum of the delays in the routers, the propagation delay in the media, and the setup mechanism. Only the first, the sum of the delays in the

routers, can be guaranteed by the router. This type of service guarantees that the packets will arrive within a certain delivery time and are not discarded if flow traffic stays within the boundary of time specification.

***Controlled-Load Service Class***

The controlled load service is a qualitative type of service in that the application requests the possibility of low-loss or no-loss packets. This type of service is designed for applications that can accept some delays, but are sensitive to an overloaded network and to the danger of losing packets. **Examples** of these types of applications are **file transfer, e-mail**, and **Internet access**.

**RSVP**

The Resource Reservation Protocol (RSVP) is a signalling protocol to help IP create a flow and consequently make a resource reservation. In the Integrated Services model, an application program needs resource reservation. The IntServ model for the resource reservation is for a ***flow***.

This means that if we want to use IntServ at the IP level, we need to create a flow, a kind of virtual-circuit network, out of the IP, which was originally designed as a datagram packet-switched network. A virtual-circuit network needs a signalling system to set up the virtual circuit before data traffic can start. The Resource Reservation Protocol (RSVP) is a signalling protocol to help IP create a flow and consequently make a resource reservation. It is an independent protocol separate from the **Integrated Services** model. It may be used in other models in the future.

# INTERNET APPLICATION

The primary errand of the Internet is to give administrations to clients. And the main responsible of the application layer is to enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, file access and transfer, access to system resources, Telnet, electronic mail, chat, www and network management etc are most popular.

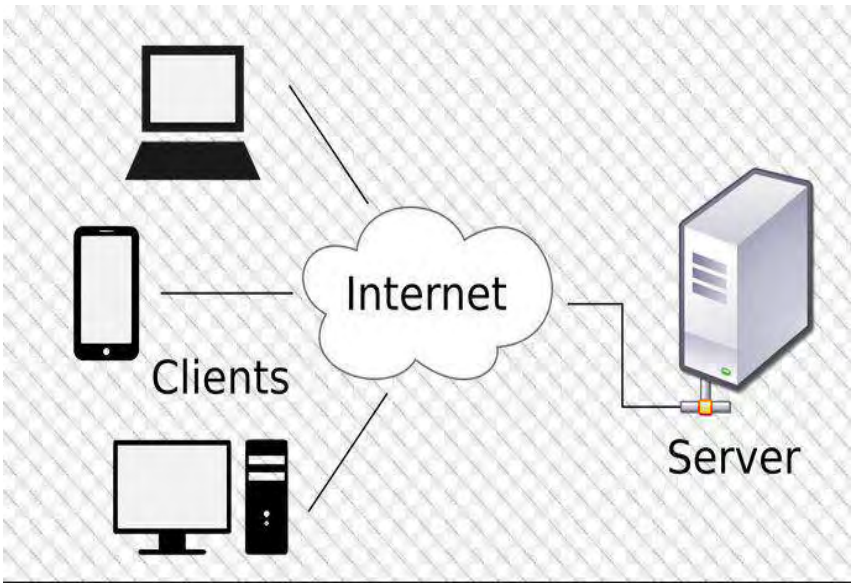
## CLIENT SERVER MODEL

One of the most common ways that a computer can ask service for another computer is the client server model. The purpose of network or internetwork is to provide service to users. A user at local site wants to receive a service from a computer at a remote site. That means two computers are connected by an internet must each run a program, one to request services other to provide service. This communication is possible because of application programs running at two end computers. But this approach has many questions as:

1. Should both the applications be able to request service and provide a service? Application program called client is running on the local machine request a service from another application program called server running on the remote machine.

2. Should an application program provide service only to one specific application program installed somewhere in an internet? Server provides service for any client, not a particular client.
3. When should application program be running? All the time. Generally a client program, which request a service should run only when it is needed and server program should run all the time because, it does not know when services will come.

4. Should there be only one universal program that provides any type of service to client or user, or there should one application program for each service? In the internet service needed by many users have specific client server application program. For example we have separate client server application programs that allow users to access files and send mails etc. For more customized service there should be a generic program that allow to access remotely.




**Client:** a client is a program running on the local machine requesting service from a server. It is started by user and terminates when service is complete.

***LECTURE NOTE-38***

# FTP

## 1. Introduction:

- **File Transfer Protocol** (**FTP**)is a standard network protocol used to transfer files from one host to another host over a TCP-based network, such as the Internet and used in Application layer of TCP/IP suite.

- The main objective of this protocol is:
    1. To transfer data reliably and efficiently
    2. To promote sharing of files (compute programs and /or data)
    3. To transfer files between FTP client and FTP servers( file download, upload)

- While transferring data over the network, four data representations can be used

    1. ASCII mode
    2. Image mode (commonly called Binary mode)
    3. EBCDIC mode

4. Local mode

**Data transfer can be done in any of three modes:**

- Stream mode: Data is sent as a continuous stream,
- Block mode: FTP breaks the data into several blocks
- Compressed mode:

## 2. History:

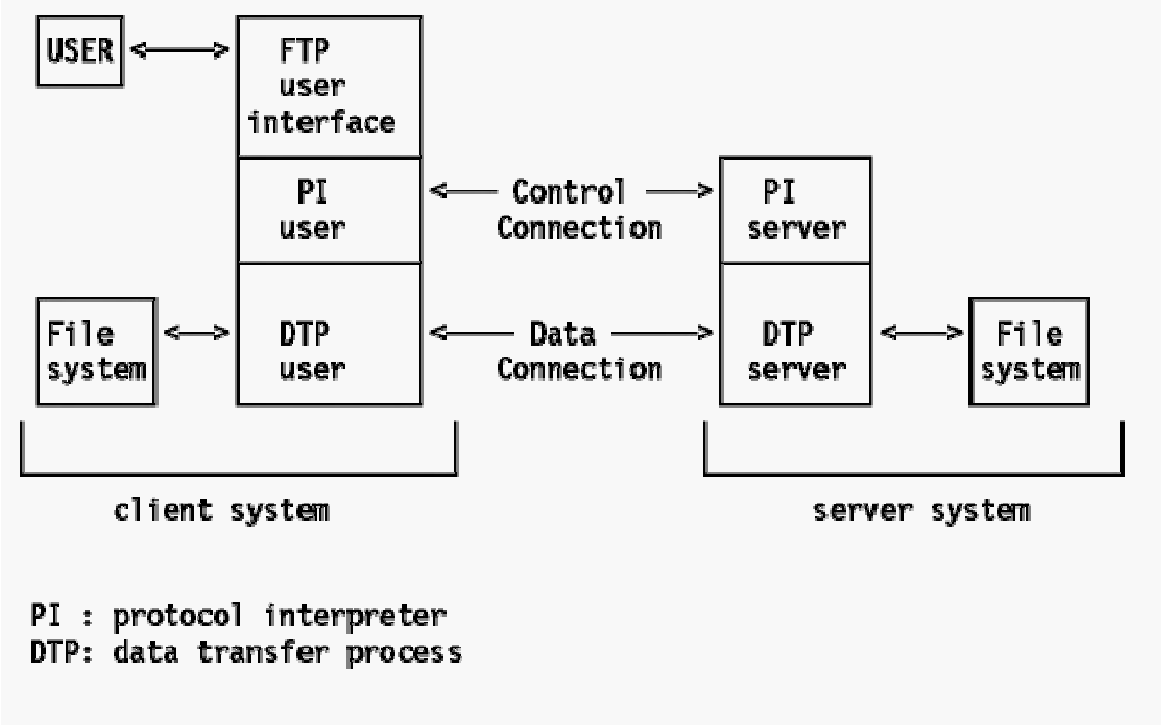
- The original specification for the File Transfer Protocol was written by **AbhayBhushan** and published as RFC 114 on 16 April 1971 and later replaced by RFC 765 (June 1980) and RFC 959 (October 1985), the current specification. Several proposed standards amend RFC 959, for example RFC 2228 (June 1997) proposes security extensions and RFC 2428 (September 1998) adds support for IPv6 and defines a new type of passive mode.
- **A Request for Comments** (**RFC**) is a publication of the Internet Engineering Task Force (**IETF**) and the Internet Society, the principal technical development and standards-setting bodies for the Internet.

## 3. The FTP Model

- FTP uses TCP as transport protocol to provide reliable end-to-end connections.

- Two connections are used: the first is the control connection and the second is the data connection that is managing the data transfer.

- On both sides of the link the FTP application is built with a protocol interpreter (PI) and a data transfer process (DTP). On the client side of the link there exists also a user interface.




- The user interface communicates with the protocol interpreter, which is in charge of the control connection.

- The protocol interpreter, besides its function of responding to the control protocol, has also to manage the data connection. During the file transfer, the data management is performed by the DTPs.

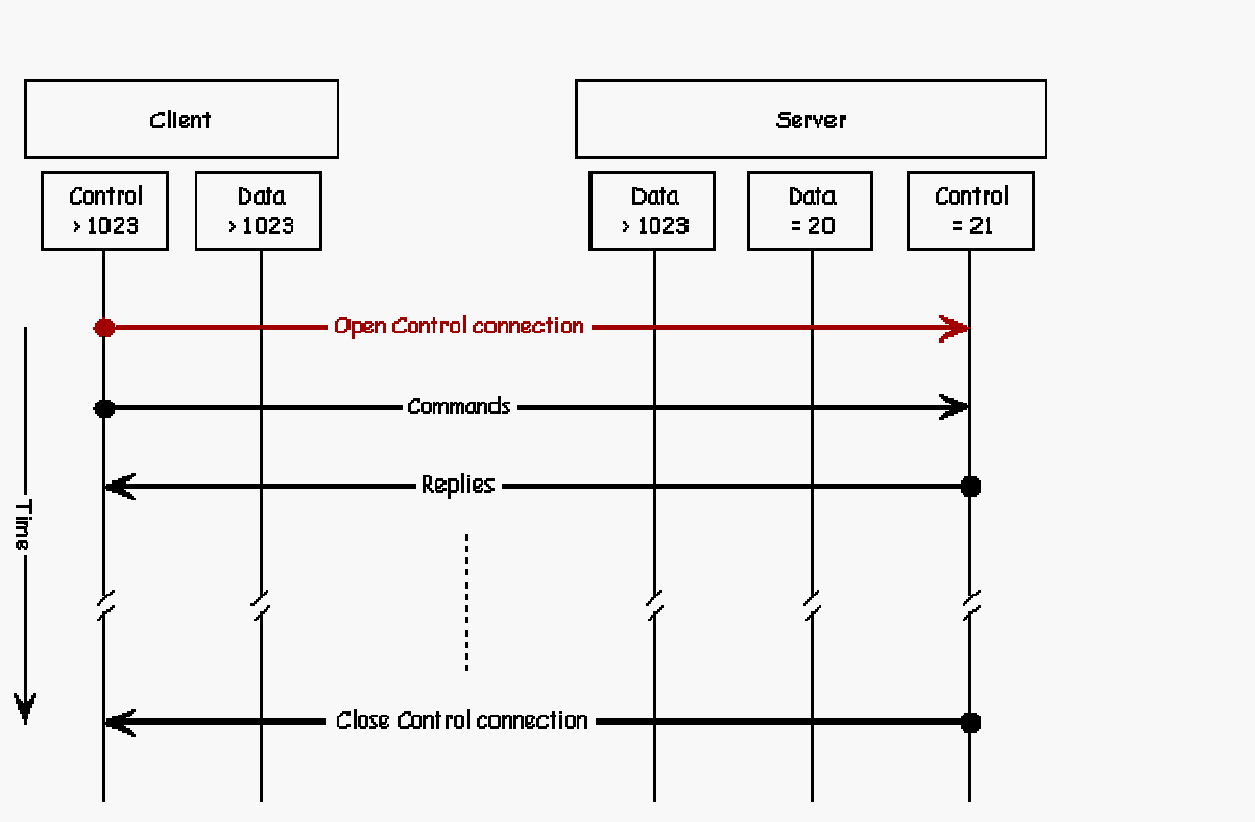
## 4. Protocol Overview:

The FTP protocol uses a control connection (the primary connection) and a data connection (the secondary connection).

### 4.1 The Control connection:

- The control connection is the communication path between the USER-PI and SERVER-PI for the exchange of commands and replies. This connection follows the Telnet Protocol.
- When an FTP client wants to exchange files with an FTP server, the FTP client must first set up the control connection. The client makes a TCP

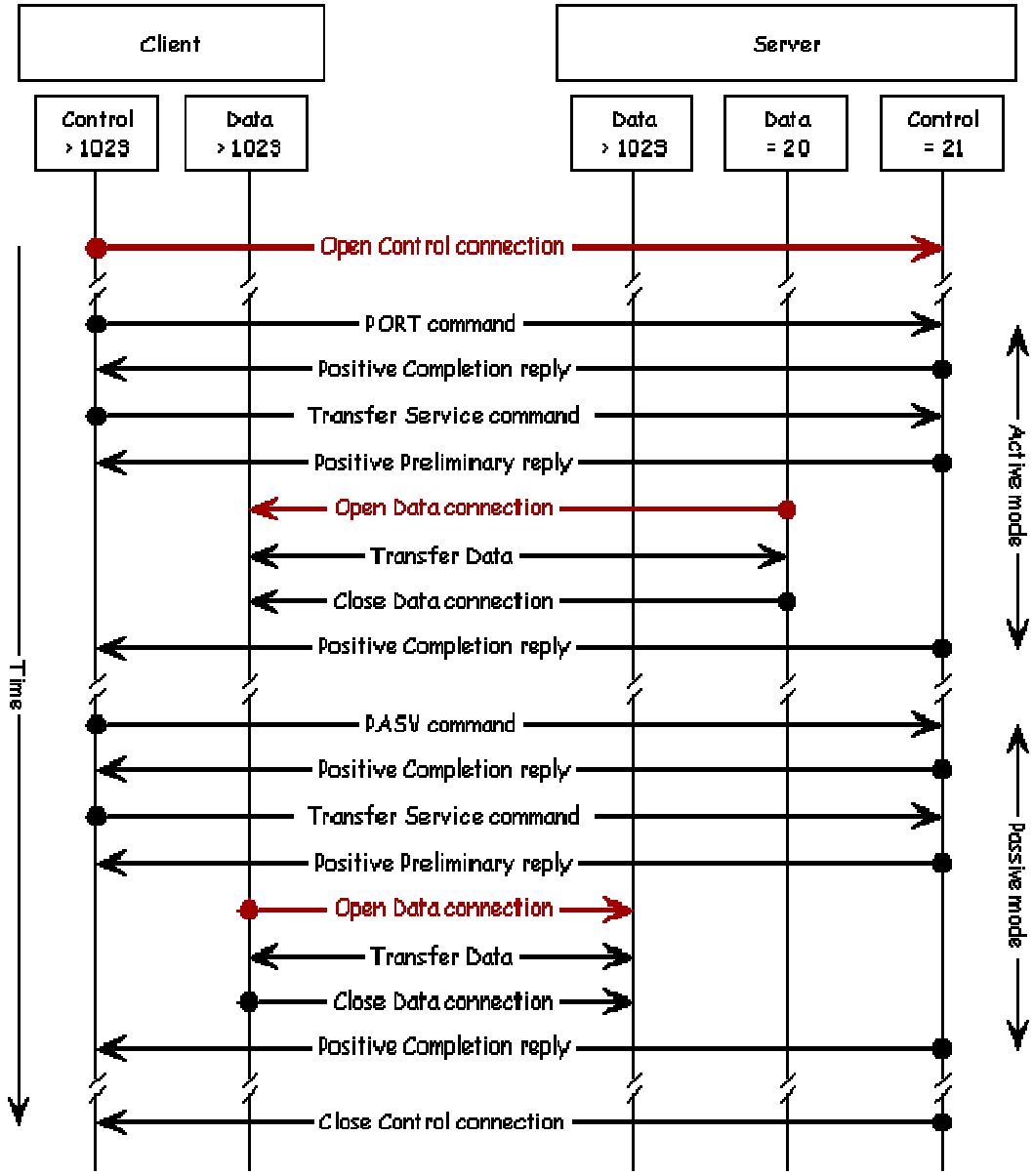
connection from a random unprivileged port N (N > 1023) to the FTP server's well known command port 21 (the IANA assigned port number).




- The protocol requests the control connection to remain open while the data transfer is in progress.
- A data connection cannot exist without an open control connection.
- The data connection doesn't need to exist all of the time and there can be many data connections during the lifetime of a control connection.
- It is the responsibility of the user to request the closing of the control connection when finished using the FTP service. However, it is the server who takes the action to close the control connection.

## 4.2 The Data connection:

- The data connection is the communication path between the USER-DTP and SERVER-DTP for the exchange of the real data, being directory lists and files. Depending on the chosen FTP mode, the data connection is initiated from the server (active mode) or the client (passive mode).




## 5. Overview: FTP Basics Operations:

**Goal:**

Setup control and data connections, transfer data, closed connections.

**Topology**: A client H1 is connected to a FTP server S1 via Internet.

**Steps:**

1. H1 requests for a control connection with S1.
2. S1 requests for a data connection with H1.
3. S1 transfers data to H1.
4. When data transfer is done, S1 requests to close data connection and control connection.

**H1: Control connection request**

At H1, user types: ftp 1.1.2.1. It triggers H1 sending a Control Connection Request packet to S1.

When S1 receives this request, it sends an Ack back to H1.Upon receiving Ack, H1 prints "20 FTP Server ready" to indicate that control connection is up.

**H1: Get foo, PORT**

- User types "get foo" at H1 to ask S1 to send a file foo.

- Then H1 sends a PORT command. Click PORT to see H1's port information: (IP: 1.1.1.1, port: 54705).

**Note:** Here FTP runs in active mode. It is server that initiates data connection. But server needs to know client's port number first. This is why H1 sends an unsolicited PORT command to S1.

**S1: Data connection request**

- Upon receiving PORT, S1 sends data_Conn to H1 (source port 20, destination port 54705)

- H1 responds with an Ack_data_Conn. Now data connection is up.

- S1 receives the Ack and sends a message to H1 (not shown in animation)

- H1 receives the message and prints “150 Opening BINARY....” to indicate that data transfer are starting.

**S1 transfers foo to H1**

- With data connection established, S1 starts to transmit foo data one packet (ftp_Data) at a time.

- When H1 receives a data packet, it responds an Ack_Data.

- When S1 receives Ack, it sends the next data packet.

**S1: close data connection**

- After S1 has transmitted all data packets, it sends a message to H1

- When H1 receives this message, it print's"226 Transfer complete" to indicate the file transfer is done.

- S1 closes the data connection and sends Close Data request to H1.

- H1 receives this request and sends an Ack to grant it. This closes FTP data connection.

**S1: close control connection**

- User has no other FTP tasks to do and types "quit." It triggers a message to S1

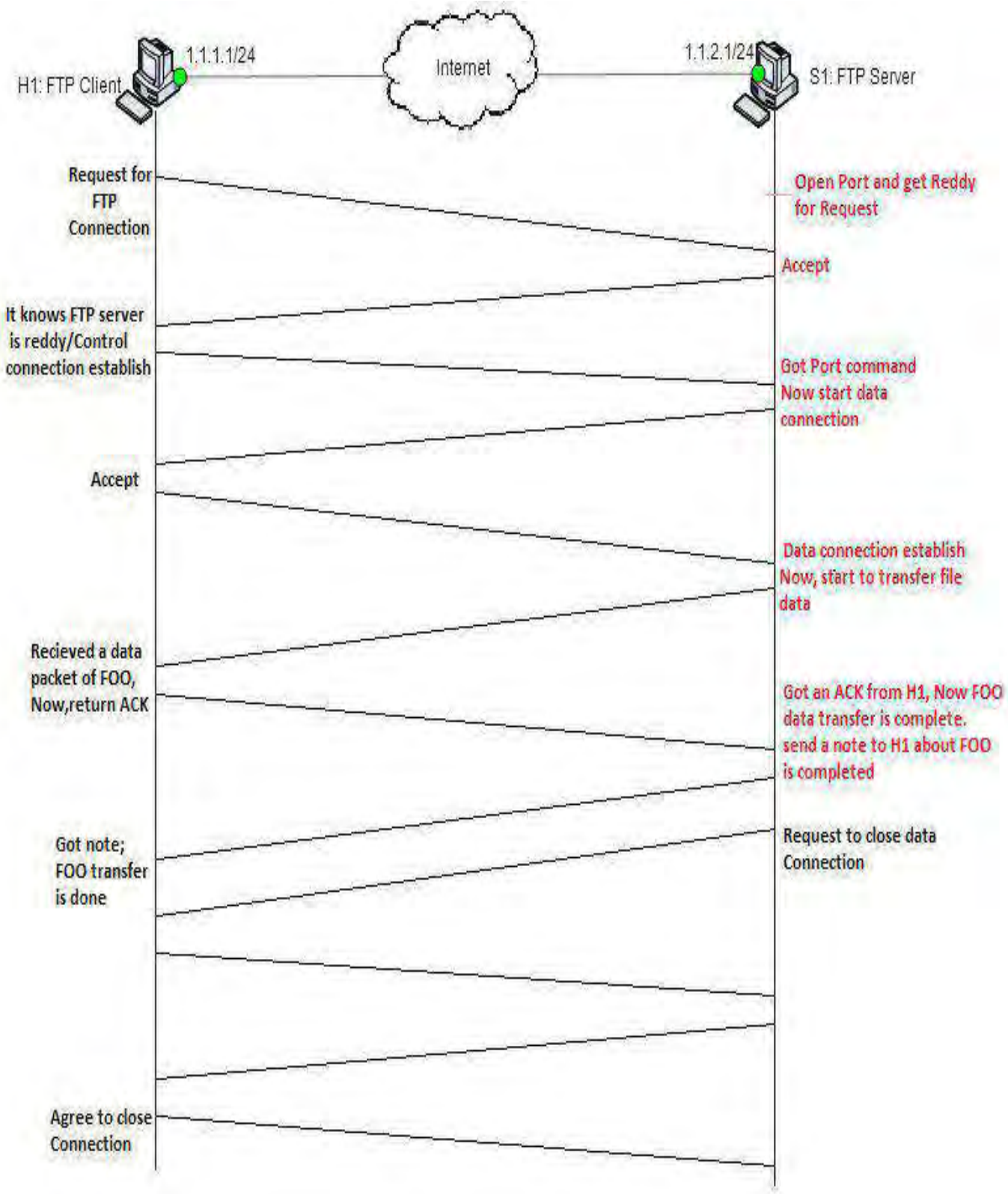
- When S1 receives the quit message, it sends a goodbye message to H1

- H1 receives this message and prints "221 Goodbye" to tell user that FTP is exited.

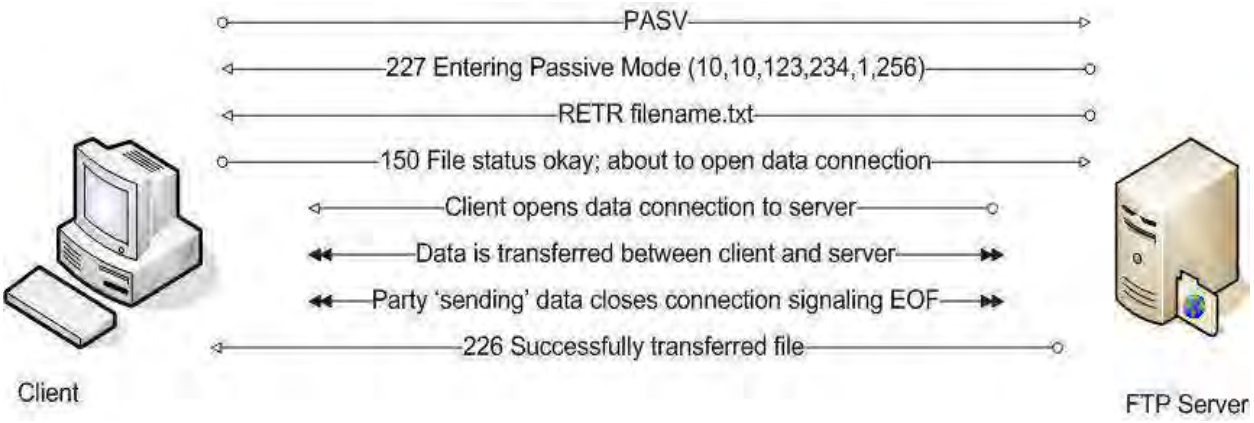
- S1 sends Close_Ctrl to close control connection with H1.

- H1 receives the request and sends Ack_Close to confirm. Now FTP control connection is closed.

## 5.1 File Transfer Protocol (basic Operations):

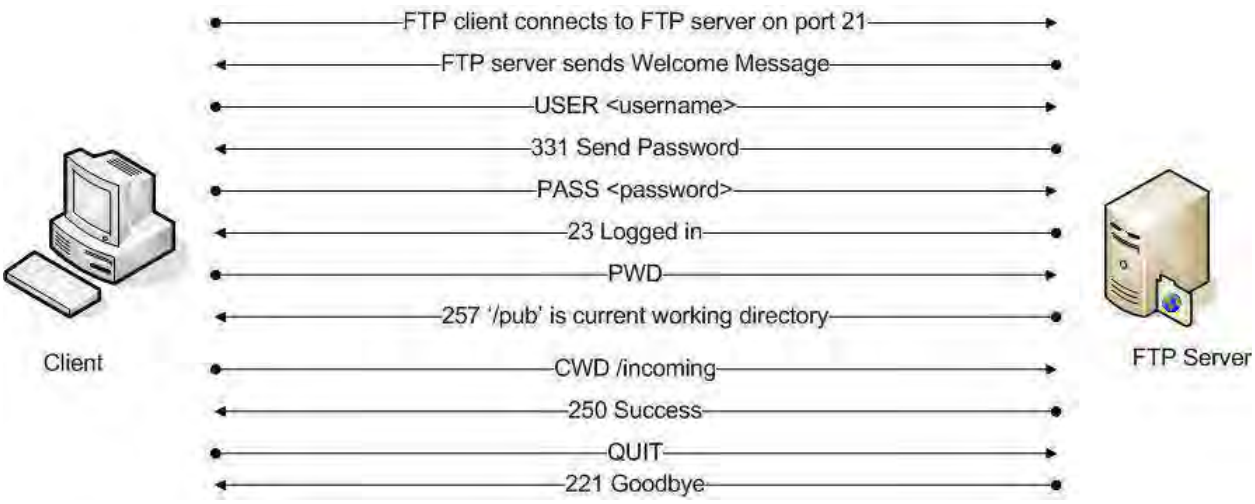


- In Passive mode, the clients are responsible for initiating both the connection control connection as well as data connection.
- In passive mode, the client sends a PASV command to the server. Basically this command asks the server to "listen" on a data port (which is not its default data port 20) and to wait for a connection rather than to initiate one.
- If the server supports the passive mode, it will send a reply to this command including the host (IP address) and port number (unprivileged port > 1023) this server is listening on.
- The client will then establish the data connection from a local random unprivileged port (> 1023) to the IP address and port number learned from the PASV reply.




## 6.3 Login

FTP login utilizes a normal username and password scheme for granting access. The username is sent to the server using the USER command, and the password is sent using the PASS command. If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence. If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.




## 7. List of FTP commands:

These are the FTP **commands** that may be sent to an FTP server, these commands are standardized in RFC 959 by the IETF.

Note that most command-line FTP clients present their own set of commands to users. For example, GET is the common user command to download a file instead of the raw command RETR.

| **Command** | **RFC** | **Description** |
|---|---|---|
| ABOR | | Abort an active file transfer. |
| ACCT | | Account information. |
| ADAT | RFC 2228 | Authentication/Security Data |
| ALLO | | Allocate sufficient disk space to receive a file. |
| APPE | | Append. |
| AUTH | RFC 2228 | Authentication/Security |

| Command | RFC | Description |
|---|---|---|
| | | Mechanism |
| CCC | RFC 2228 | Clear Command Channel |
| CDUP | | Change to Parent Directory. |
| CONF | RFC 2228 | Confidentiality Protection Command |
| CWD | | Change working directory. |
| DELE | | Delete file. |
| ENC | RFC 2228 | Privacy Protected Channel |
| EPRT | RFC 2428 | Specifies an extended address and port to which the server should connect. |
| EPSV | RFC 2428 | Enter extended passive mode. |
| FEAT | RFC 2389 | Get the feature list implemented by the server. |
| HELP | | Returns usage documentation on a command if specified, else a general help document is returned. |
| LANG | RFC 2640 | Language Negotiation |
| LIST | | Returns information of a file or directory if specified, else information of the current working directory is returned. |
| LPRT | RFC 1639 | Specifies a long address and port to which the server should connect. |

| Command | RFC | Description |
|---|---|---|
| LPSV | RFC 1639 | Enter long passive mode. |
| MDTM | RFC 3659 | Return the last-modified time of a specified file. |
| MIC | RFC 2228 | Integrity Protected Command |
| MKD | | Make directory. |
| MLSD | RFC 3659 | Lists the contents of a directory if a directory is named. |
| MLST | RFC 3659 | Provides data about exactly the object named on its command line, and no others. |
| MODE | | Sets the transfer mode (Stream, Block, or Compressed). |
| NLST | | Returns a list of file names in a specified directory. |
| NOOP | | No operation (dummy packet; used mostly on keepalives). |
| OPTS | RFC 2389 | Select options for a feature. |
| PASS | | Authentication password. |
| PASV | | Enter passive mode. |
| PBSZ | RFC 2228 | Protection Buffer Size |
| PORT | | Specifies an address and port to which the server should connect. |
| PROT | RFC 2228 | Data Channel Protection Level. |
| PWD | | Print working directory. Returns the current directory of the host. |

| Command | RFC | Description |
|---|---|---|
| QUIT | | Disconnect. |
| REIN | | Re initializes the connection. |
| REST | | Restart transfer from the specified point. |
| RETR | | Transfer a copy of the file |
| RMD | | Remove a directory. |
| RNFR | | Rename from. |
| RNTO | | Rename to. |
| SITE | | Sends site specific commands to remote server. |
| SIZE | RFC 3659 | Return the size of a file. |
| SMNT | | Mount file structure. |
| STAT | | Returns the current status. |
| STOR | | Accept the data and to store the data as a file at the server site |
| STOU | | Store file uniquely. |
| STRU | | Set file transfer structure. |
| SYST | | Return system type. |
| TYPE | | Sets the transfer mode (ASCII/Binary). |
| USER | | Authentication username. |
| XCUP | RFC 775 | Change to the parent of the current working directory |

| Command | RFC | Description |
|---|---|---|
| XMKD | RFC 775 | Make a directory |
| XPWD | RFC 775 | Print the current working directory |
| XRCP | RFC 743 | |
| XRMD | RFC 775 | Remove the directory |
| XRSQ | RFC 743 | |
| XSEM | RFC 737 | Send, mail if cannot |
| XSEN | RFC 737 | Send to terminal |

## 8. Advantages of FTP

- FTP is the fast and efficient way of transferring bulks of data across the internet.
- Allows transferring multiple files as well as directories.
- Many FTP clients have the ability to schedule transfers.
- No size limitation on single transfers (browsers only allow up to 2 GB)
- Many clients have scripting capabilities through command line
- Most clients have a synchronizing utility
- Faster transfers then HTTP
- It has an automatic backup .Whenever you edit your files in your local system you can update the same by copying it to the host system in your site. So in cases where your site has crashed and all the data is lost you have a copy of it in your own local system. It also works the other way round.
- FTP gives you control over transfer. That is, you can choose the mode in which the data is transferred over the network. The data can be transferred

either in the ASCII mode (for text files) or in the Binary mode (for executable or compressed files).

- You can work with the directories on the remote systems, delete or rename the remote files while transferring data between 2 hosts.

**9. Disadvantages of FTP**

- FTP was not designed to be a secure protocol.
- FTP causes the following attacks during the transfer of data.
    1. Bounce Attacks
    2. Spoof Attacks
    3. Brute Force Attacks
    4. Packet Sniffing
    5. User name protection
    6. Port sealing

- Encryption of data is not done in FTP.
- Usernames, passwords and files are sent in clear text.
- Servers can be spoofed to send data to a random port on an unintended computer
- Filtering active FTP connections is difficult on your local machine (passive is preferred)

**Electronic Mail**

A standout amongst the most prominent Internet administrations is electronic mail (email). The planners of the Internet most likely never envisioned the ubiquity of this application program. At the start of the Internet period, the messages sent by electronic mail were short what's more comprised of content just; they let individuals trade brisk updates. Today, electronic mail is a great deal more intricate. It permits a message to incorporate content, sound, and feature. It moreover permits one message to be sent to one or more beneficiaries.

An Internet email message comprises of three parts, the message envelope, the message header, and the message body. The message header contains control data, including, negligibly, an originator's email location and one or more beneficiary locations. Generally spellbinding data is additionally included, for example, a subject header field and a message accommodation date/time stamp.

***LECTURE NOTE-39***

# WWW and HTTP

The World Wide Web (WWW) is a store of data connected together from focuses everywhere throughout the world. The WWW has a remarkable blend of adaptability, convenience, also easy to understand characteristics that recognize it from different administrations gave by the Internet.

The WWW undertaking was launched by CERN (European Laboratory for Particle Physics) to make a framework to handle circulated assets vital for logical exploration.

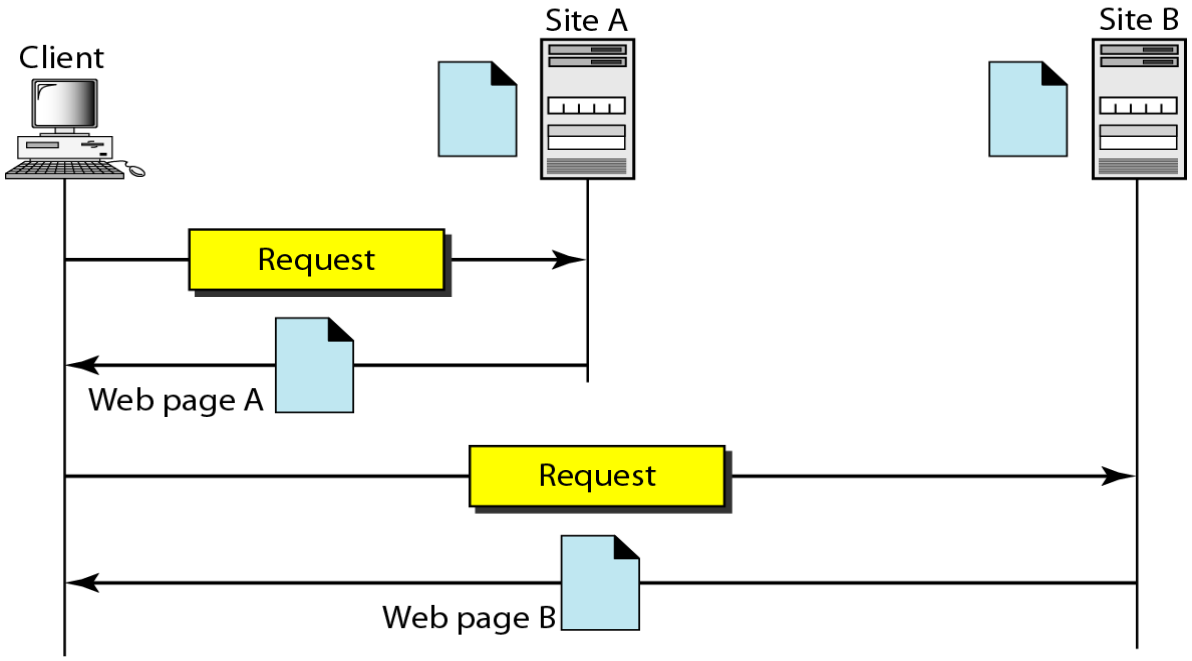
The World Wide Web (WWW, W3) is a data arrangement of interlinked hypertext archives that are gotten to by means of the Internet. It has likewise generally ended up referred to just as the Web. Individual record pages on the World Wide Web are called site pages and are gotten to with a product application running on the client's PC, ordinarily called a web program. Website pages may contain content, pictures, features, and other mixed media parts, and in addition web route gimmicks comprising of hyperlinks.

**History**

Tim Berners-Lee, a British PC researcher and previous CERN representative, is viewed as the innovator of the Web. On 12 March 1989, Berners-Lee composed a proposition for what would in the long run turn into the World Wide Web.

**Architecture**

The WWW today is a dispersed client server administration, in which a customer utilizing a program can get to an administration utilizing a server. Notwithstanding, the administration gave is conveyed over numerous areas called sites, as indicated in Figure.

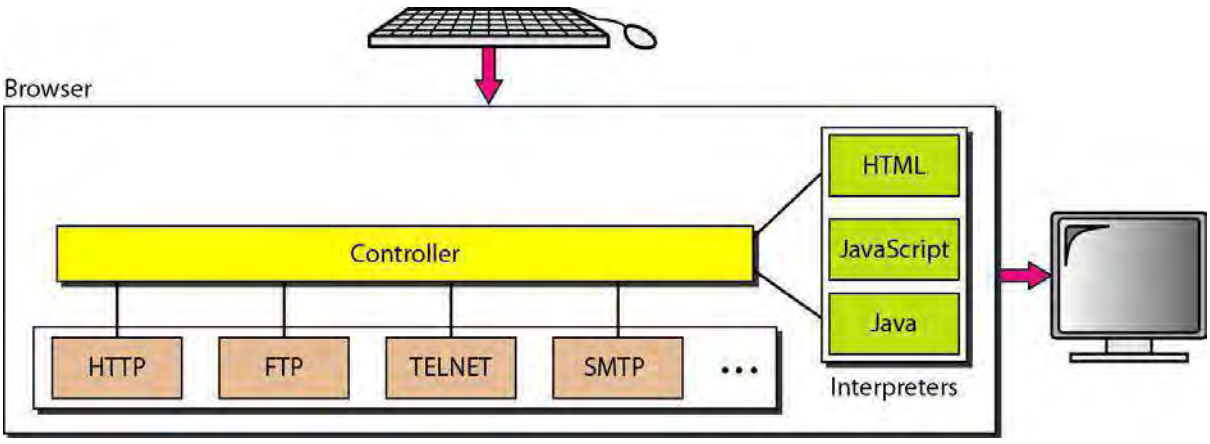



The client needs to see some information that it knows belongs to site A. It sends a request through its browser, a program that is designed to fetch Web documents.

The request, among other information, includes the address of the site and the Web page, called the URL. The server at site A finds the document and sends it to the client. When the user views the document, she finds some references to other documents, including a Web page at site B. The reference has the URL for the new site. The user is also interested in seeing this document. The client sends another request to the new site, and the new page is retrieved.

**Client (Browser)**

Each browser usually consists of three parts: a controller, client protocol, and interpreters. The controller receives input from the keyboard or the mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. The client protocol can be one of the protocols like FTP The interpreter can be HTML, Java, or JavaScript, depending on the type of document.



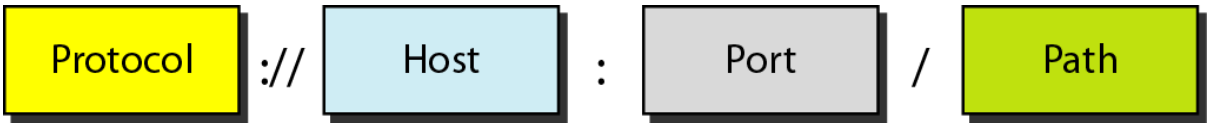

**SERVER**

The Webpage is put away at the server. Each time a customer solicitation arrives, the relating record is sent to the client. To enhance efficiency, servers regularly

store asked records in a store in memory; memory is speedier to get to than disk. A server can likewise become more productive through multithreading or multiprocessing. In this case, a server can answer more than one appeal at once.

**Uniform Resource Locator**

An Uniform Resource Locator (abridged URL; otherwise called a web address, especially when utilized with HTTP) is a particular character string that constitutes a reference to an asset. Most web programs show the URL of a website page over the page in an address bar.




The URL defines four things: protocol, host computer, port and path as shown in above figure.

**Protocol**

The protocol is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP.

**Host**

The host is the computer on which 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.

**Port**

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

**Path**

Path is the pathname of the file where the information is located.

**Function of WWW**

- The WWW works by establishing hypertext/hypermedia links between documents anywhere on the network.
- A document might include many links to other documents held on many different servers.
- Selecting any one of those links will take you to the related document wherever it is.

  E.g. the references at the end of a paper might have hypertext links to the actual documents held elsewhere.

**WWW Hyperlinks**

Hyperlinks can link a part of a hypermedia document to

- Another part of the same document file.

- Another document file on the same server computer.
- Another document file on a server computer located elsewhere in the world.

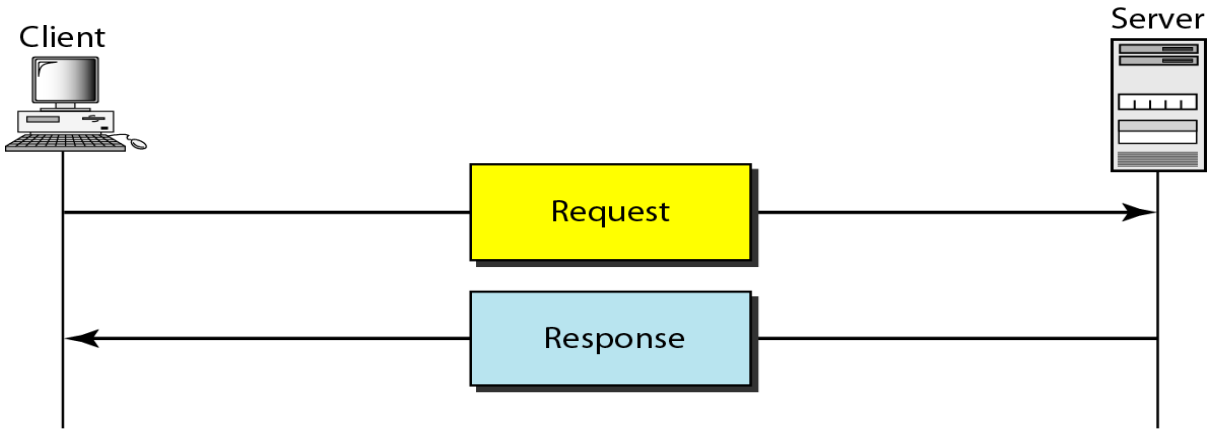
**HTTP**

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

**HTTP uses the services of TCP on well-known port 80.**

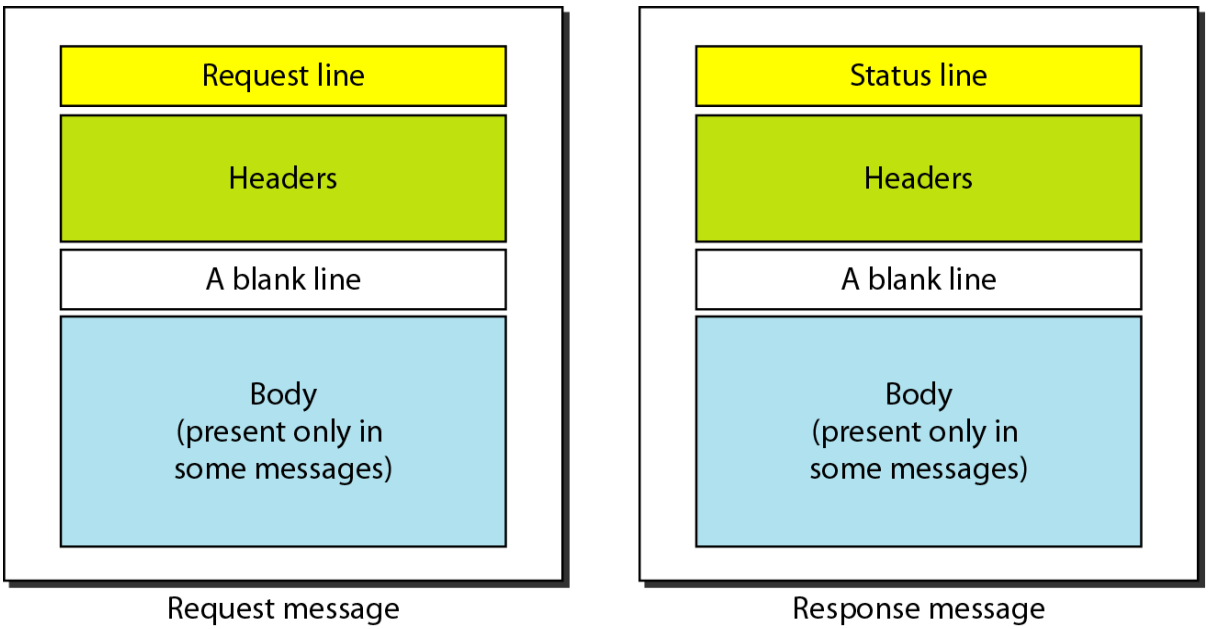
**HTTP Transaction**

HTTP itself is a stateless protocol. The client initializes the transaction by sending a request message. The server replies by sending a response.



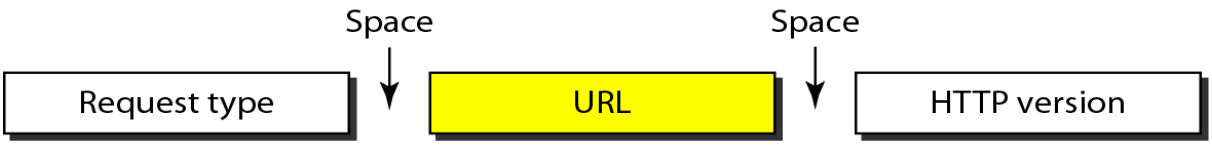
Messages

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

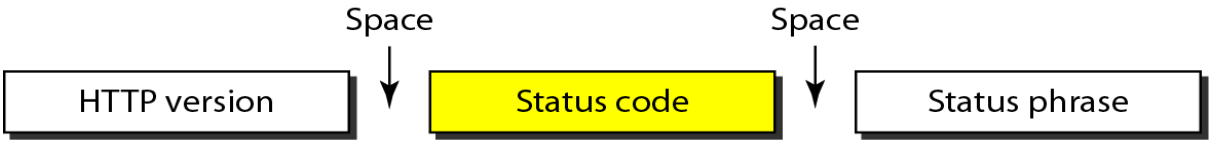



Request and Status Lines

The first line in a request message is called a request line; the first line in the response message is called the status line.




a. Request line




b. Status line

**HTTP Methods**

HTTP allows an open-ended set of methods to be used to indicate the purpose of a request.

The three most often used methods are GET, HEAD, and POST.

| *Method* | *Action* |
|---|---|
| GET | Requests a document from the server |
| HEAD | Requests information about a document but not the document itself |
| POST | Sends some information from the client to the server |
| PUT | Sends a document from the server to the client |
| TRACE | Echoes the incoming request |
| CONNECT | Reserved |
| OPTION | Inquires about available options |

The GET Method

The GET method is used to ask for a specific document - when you click on a hyperlink, GET is being used. GET should probably be used when a URL access will not change the state of a database (by, for example, adding or deleting information) and POST should be used when an access will cause a change. The semantics of the GET method changes to a "conditional GET" if the request message includes an "If-Modified-Since:" header field.

The HEAD Method is used to ask only for information about a document, not for the document itself. HEAD is much faster than GET, as a much smaller amount of data is transferred. It's often used by clients who use caching, to see if the document has changed since it was last accessed. If it was not, then the local copy can be reused, otherwise the updated version must be retrieved with a GET. The meta-information contained in the HTTP headers in response to a HEAD request should be identical to the information sent in response to a GET request.

**The POST Method**

The POST method is used to transfer data from the client to the server; it's designed to allow a uniform method to cover functions like: annotation of existing resources; posting a message to a bulletin board, newsgroup, mailing list, or similar group of articles; providing a block of data (usually a form) to a data-handling process; extending a database through an append operation.

The Status Code

The status code field used in the response message is similar to those in the FTP and the SMTP protocols. It consists of three digits. Whereas the codes in the 100

range are only informational, the codes in the 200 range indicate a successful request.

| *Code* | *Phrase* | *Description* |
|---|---|---|
| | **Informational** | |
| **100** | Continue | The initial part of the request has been received, and the client may continue with its request. |
| **101** | Switching | The server is complying with a client request to switch protocols defined in the upgrade header. |
| | **Success** | |
| **200** | OK | The request is successful. |
| **201** | Created | A new URL is created. |
| **202** | Accepted | The request is accepted, but it is not immediately acted upon. |
| **204** | No content | There is no content in the body. |

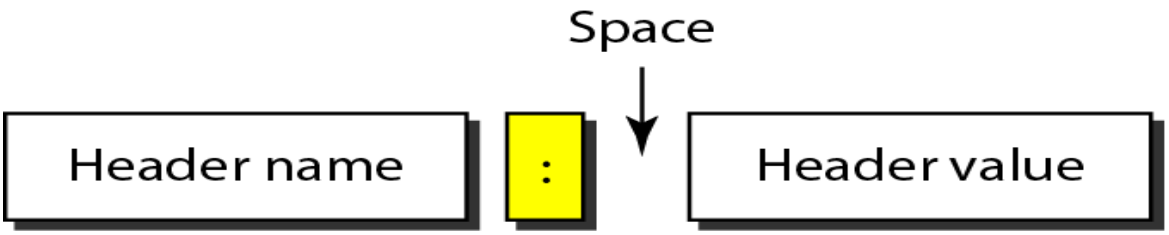
**Status Phase**

This field is used in the response message. It explains the status code in text form.

**HEADER**

The header exchanges additional information between the client and the server. The header can consist of one or more header lines. Each header line has a header name, a colon, a space, and a header value.

A header line belongs to one of four categories: general header, request header, response header, and entity header. A request message can contain only general, request, and entity headers. A response message, on the other hand, can contain only general, response, and entity headers.




**General Header**

The general header gives general information about the message and can be present in both a request and a response.

| *Header* | *Description* |
|---|---|
| Cache-control | Specifies information about caching |
| Connection | Shows whether the connection should be closed or not |
| Date | Shows the current date |
| MIME-version | Shows the MIME version used |
| Upgrade | Specifies the preferred communication protocol |

**Request Header**

The request header can be present only in a request message. It specifies the client's configuration and the client's preferred document format.

| *Header* | *Description* |
|---|---|
| Accept | Shows the medium format the client can accept |
| Accept-charset | Shows the character set the client can handle |
| Accept-encoding | Shows the encoding scheme the client can handle |
| Accept-language | Shows the language the client can accept |
| Authorization | Shows what permissions the client has |
| From | Shows the e-mail address of the user |
| Host | Shows the host and port number of the server |
| If-modified-since | Sends the document if newer than specified date |
| If-match | Sends the document only if it matches given tag |
| If-non-match | Sends the document only if it does not match given tag |
| If-range | Sends only the portion of the document that is missing |
| If-unmodified-since | Sends the document if not changed since specified date |
| Referrer | Specifies the URL of the linked document |
| User-agent | Identifies the client program |

**Response Header**

The response header can be present only in a response message. It specifies the server's configuration and special information about the request.

| *Header* | *Description* |
|---|---|
| Accept-range | Shows if server accepts the range requested by client |
| Age | Shows the age of the document |
| Public | Shows the supported list of methods |
| Retry-after | Specifies the date after which the server is available |
| Server | Shows the server name and version number |

**Entity Header**

The entity header gives information about the body of the document.Although it is mostly present in response messages, some request messages,such as POST or PUT methods, that contain a body also use this type of header.

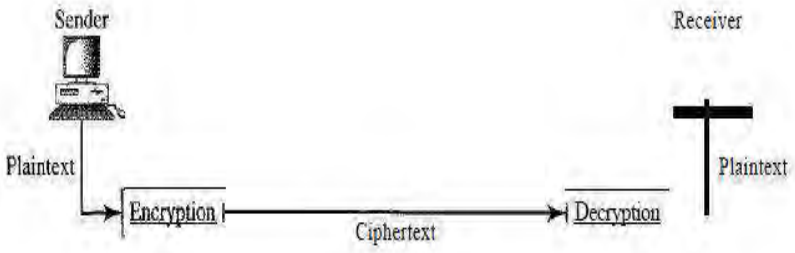
| *Header* | *Description* |
|---|---|
| Allow | Lists valid methods that can be used with a URL |
| Content-encoding | Specifies the encoding scheme |
| Content-language | Specifies the language |
| Content-length | Shows the length of the document |
| Content-range | Specifies the range of the document |
| Content-type | Specifies the medium type |
| Etag | Gives an entity tag |
| Expires | Gives the date and time when contents may change |
| Last-modified | Gives the date and time of the last change |
| Location | Specifies the location of the created or moved document |

**Body**

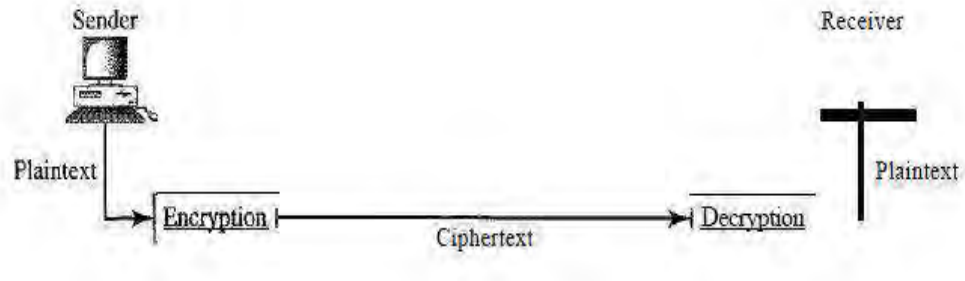
The body can be present in a request or response message. Usually, it contains the document to be sent or received.

*LECTURE NOTE-40*

# ENCRYPTION




- Network security is mostly achieved through the use of cryptography, a science based on abstract algebra.
- **Cryptography,** a word with Greek origins, means "secret writing." However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks.
- The above figure shows the components involved in cryptography:




**Plaintext and Cipher text:**

- The original message, before being transformed, is called plaintext. After the message is transformed, it is called cipher text.
- An encryption algorithm transforms the plaintext into ciphertext; a decryption algorithm transforms the ciphertext back into plaintext.
- The sender uses an encryption algorithm, and the receiver uses a decryption algorithm.

**Cipher:**

- We refer to encryption and decryption algorithms as ciphers.
- The term *cipher* is also used to refer to different categories of algorithms in cryptography.

**Key:**

- A key is a number (or a set of numbers) that the cipher, as an algorithm, operates on.
- To encrypt a message, we need an encryption algorithm, an encryption key, and the plaintext. These create the cipher text.
- To decrypt a message, we need a decryption algorithm, a decryption key, and the cipher text. These reveal the original plaintext.

Encryption is a generic term that refers to the act of encoding data, in this context so that those data can be securely transmitted via the Internet.

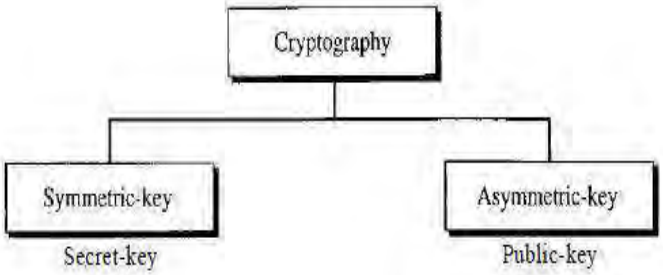
Encryption can protect the data at the simplest level by preventing other people from reading the data. In the event that someone intercepts a data transmission and manages to deceive any user identification scheme, the data that they see appears to be gibberish without a way to decode it.

Encryption technologies can help in other ways as well, by establishing the identity of users (or abusers); control the unauthorized transmission or forwarding of data; verify the integrity of the data (i.e., that it has not been altered in any way); and ensure that users take responsibility for data that they have transmitted.

Encryption can therefore be used either to keep communications secret (defensively) or to identify people involved in communications (offensively).

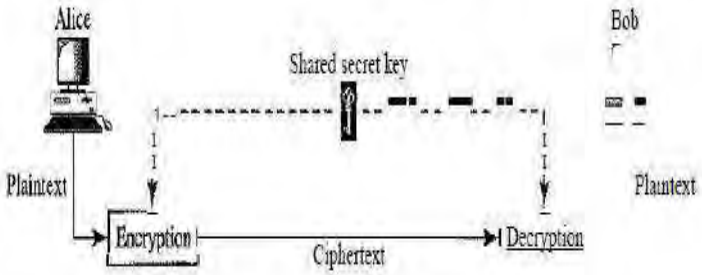
Secure E-Commerce transactions use the encryption technologies below:

- Symmetric-key Encryption
- Asymmetric-key Encryption



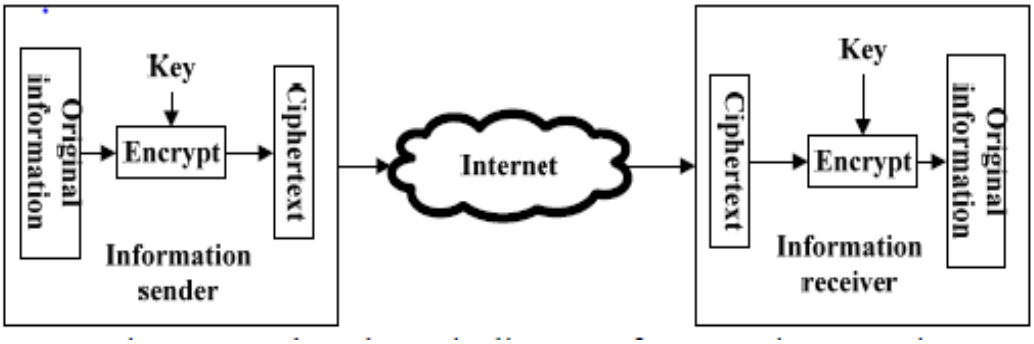

**Symmetric-key Encryption**

- In symmetric-key cryptography, the same key is used by both parties.
- The sender uses this key and an encryption algorithm to encrypt data; the receiver uses the same key and the corresponding decryption algorithm to decrypt the data.




The basic means of encrypting data involves a symmetric cryptosystem. The same key is used to encrypt and to decrypt data.

When sending information, it will be crypted through certain algorithms and keys and the original information will be changed into cipher text. When receiving information, it will be decrypted with the same algorithms and keys and cipher text will be restored.




Symmetric Encryption

At present the most widely used symmetric encryption algorithm is DES (Data Encryption Standard) algorithm proposed by the IBM Company. DES is a binary data encryption algorithm.

The advantages of symmetric encryption are fast speed, high efficiency. It is widely used in encryption of large amount of data.

The disadvantages are that keys are easily intercepted when they are transmitted on the network. That will pose a threat to information security.

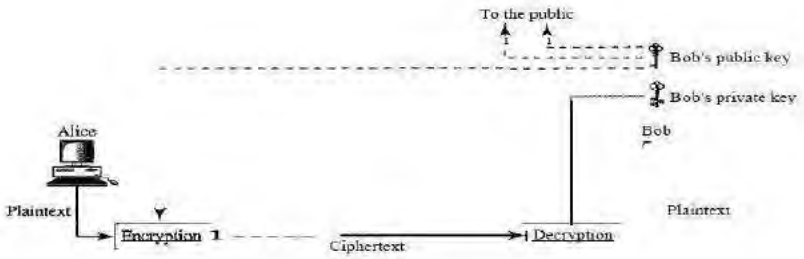
Therefore, when using symmetric encryption the security of key transmission need be guaranteed.

**Asymmetric Key Encryption**

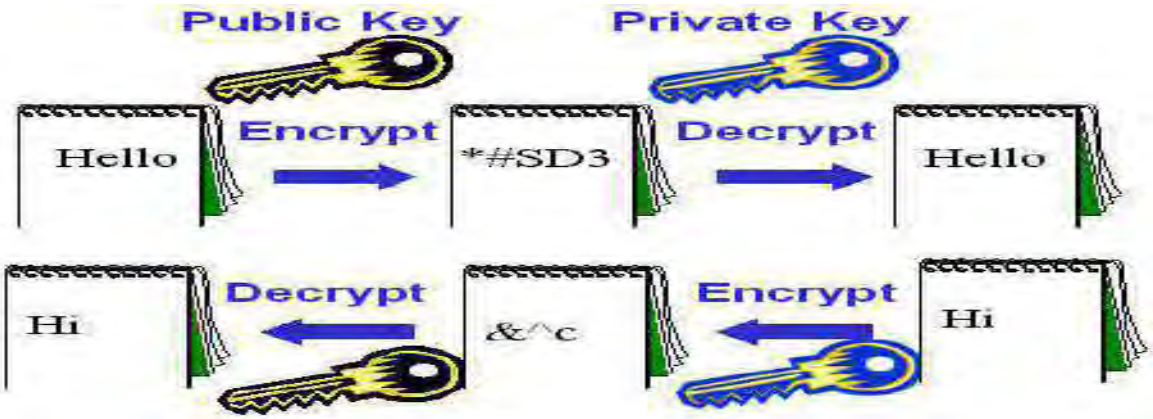
- In asymmetric key cryptography,  two keys are used such as:

1. **Private Key:** The private key is kept by the receiver. The private key is available only to an individual.
2. **Public key:** The public key is announced to the public. The public key is available to the public.

Ex: imagine Alice wants to send a message to Bob. Alice uses the public key to encrypt the message. When the message is received by Bob, the private key is used to decrypt the message




Public Key Encryption, or asymmetric encryption, is much more important than symmetric encryption for the purposes of e-commerce. The big improvement wrought by Public Key Encryption was the introduction of the second key - which makes a world of difference in terms of protecting the integrity of data. Public Key Encryption relies on two keys, one of which is public and one of which is private. If you have one key, you cannot infer the other key.



- We can see that in the asymmetric encryption technology key is decomposed into a pair
- (Private key and public key). There into private key belongs to the owner of key pair and others do not know.
- Public key is open and everyone can know. Information encrypted by public key can be decrypted only by the corresponding private key.
- Information encrypted by private key can be decrypted only by the corresponding public key.


**Asymmetric-Key Encryption**

Typical asymmetric encryption algorithm is the RSA algorithm . The algorithm is proposed by R. Rivest, A.Shamir and L. Adleman from the Massachusetts Institute of Technology. It builds on the basis of the theories of decomposition of large numbers and detection of prime numbers.

The most common use of PKE for e-commerce involves the use of so-called **Digital Certificates** issued by "trusted" third parties.

**Digital Certificates**

- Digital certificates are digital files that certify the identity of an individual or institution seeking access to computer-based information. In enabling such access, they serve the same purpose as a driver's license or library card.
- The digital certificate links the identifier of an individual or institution to a digital *public key*.
- The certificate includes information about the key, information about its owner's identity, and the **digital signature** of an entity that has verified the certificate's contents are correct.
- If the signature is valid, and the person examining the certificate trusts the signer, then they know they can use that key to communicate with its owner.
- In a typical public-key infrastructure (PKI) scheme, the signer is a certificate authority (CA), usually a company such as VeriSign which charges customers to issue certificates for them.

**How these certificates are issued**

Digital certificates are issued by certificate authorities, just as state governments issue driver's licenses. There are several public companies in the business of issuing certificates. Also, many campuses are setting up their own certificate authorities and issuing certificates to their faculty members, staff, and students. This is similar to campuses issuing ID cards to the members of their communities. How campuses issue certificates will depend on the technical infrastructure and institutional policies that are established. Certificate authorities are responsible for managing the life cycle of certificates, including their revocation.

**Digital Signatures**

- A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document.

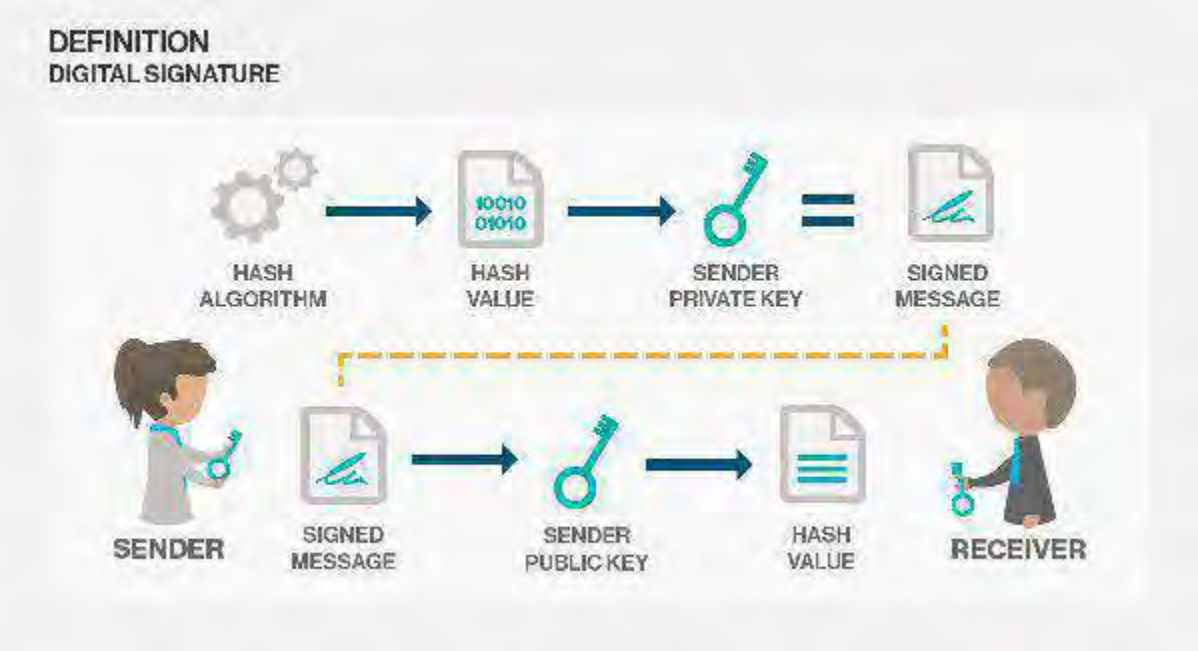
- A valid digital signature gives a recipient reason to believe that the message was created by a known sender, such that the sender cannot deny having sent the message (authentication and non-repudiation) and that the message was not altered in transit (integrity).
- Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.
- Digital signatures are based on a combination of the traditional idea of data hashing with public-key based encryption. Most hash functions are similar to encryption functions; in fact, some hash functions are just slightly modified encryption functions.
- The digital equivalent of a handwritten signature or stamped seal, but offering far more inherent security, a digital signature is intended to solve the problem of tampering and impersonation in digital communications.
- Digital signatures can provide the added assurances of evidence to origin, identity and status of an electronic document, transaction or message, as well as acknowledging informed consent by the signer.

**How digital signatures work?**

Digital signatures are based on public key cryptography, also known as asymmetric. Using a public key algorithm such as RSA, one can generate two keys that are mathematically linked: one private and one public. To create a digital signature, signing software (such as an email program) creates a one-way hash of the electronic data to be signed. The private is then used to encrypt the hash. The encrypted hash -- along with other information, such as the hashing algorithm -- is the digital signature. The reason for encrypting the hash instead of the entire message or document is that a hash function can convert an arbitrary input into a

fixed length value, which is usually much shorter. This saves time since hashing is much faster than signing.

The value of the hash is unique to the hashed data. Any change in the data, even changing or deleting a single character, results in a different value. This attribute enables others to validate the integrity of the data by using the signer's public key to decrypt the hash. If the decrypted hash matches a second computed hash of the same data, it proves that the data hasn't changed since it was signed. If the two hashes don't match, the data has either been tampered with in some way (integrity) or the signature was created with a private key that doesn't correspond to the public key presented by the signer (authentication).




**Applications of digital signatures**

As organizations move away from paper documents with ink signatures or authenticity stamps, digital signatures can provide added assurances of the

evidence to provenance, identity, and status of an electronic document as well as acknowledging informed consent and approval by a signatory.

Some common reasons for applying a digital signature to communications:

**Authentication**

Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user.

**Integrity**

The sender and receiver of a message may have a need for confidence that the message has not been altered during transmission.

**Non-repudiation**

Non-repudiation, or more specifically *non-repudiation of origin*, is an important aspect of digital signatures. By this property, an entity that has signed some information cannot at a later time deny having signed it. Similarly, access to the public key only does not enable a fraudulent party to fake a valid signature.

As the digital signature process is central to the idea of a digital certificate - and in turn, the digital certificate is the primary tool to ensure e-commerce security.

# REFERENCES


1. **Data Communication and Networking – B.A. Forouzan, TMH**

2. http://www.tutorialspoint.com/