Goals and Applications of networks:-

Network goals:- The main goals of network as follows.

Resource sharing:- This is the main aim of a computer network. It means to make all programs peripherals and data available to any one computer on the network to all other computers in the network without regard to the physical locations of them. Thus user at large distances can share the resources or can see data of a computer in the same way that a local user uses them. Another aspect of resource sharing is load sharing. That is if required, a job can be performed using various computers in network by portioning it which reduces time consumption and load both for a particular computer.

High reliability:- A second goal is to provide high reliability by having alternative sources of supply. For example, all files could be replicated on two or three machines, so if one of them is Unavailable, the other copies could be available.

Cost Reduction:- Another goal of networking is reduction of cost. Resource sharing automatically reduces cost and hence money can be saved.

Improve Performance:- Another closely related goal is to increase the systems performance. The performance of a computer can be improved by adding one or more processors to it as the work load on it grows. For example if the system is full instead of replacing it buy a larger one at large expansive it is better to add more processors to it on less cost and less disruption to the user.

Communication Medium:- Computer networks provide a powerful communication medium. A file that was updated/modified on a network can be seen by the other users on the network immediately.

Network application:- Some of the network applications in different fields are the following.

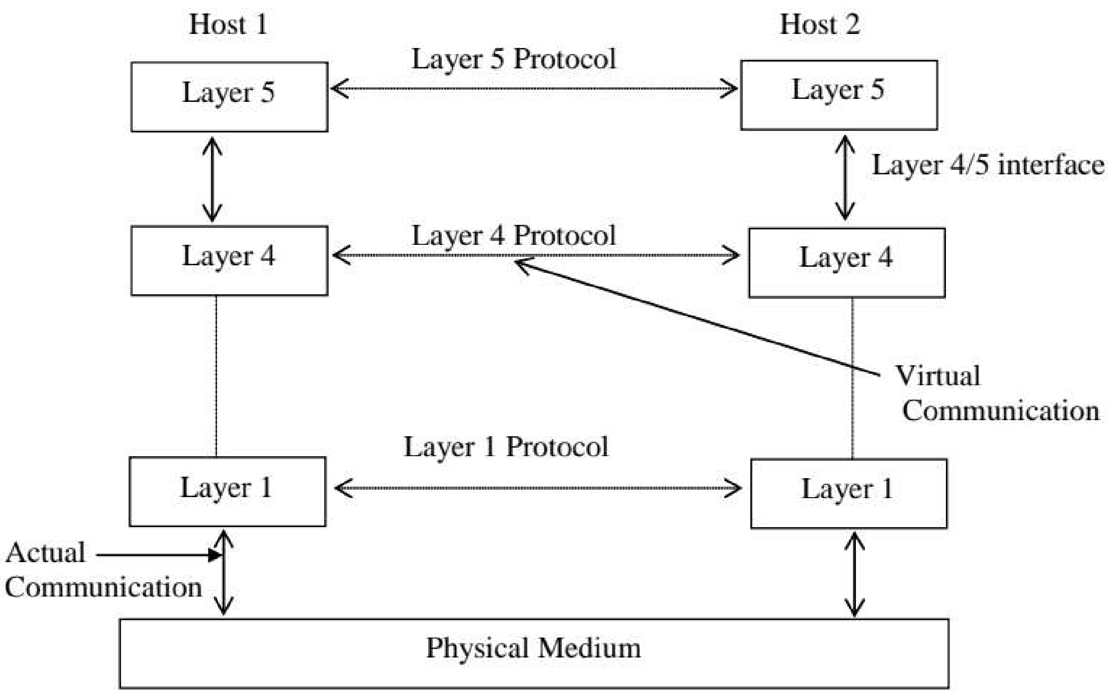
1. Marketing and sales

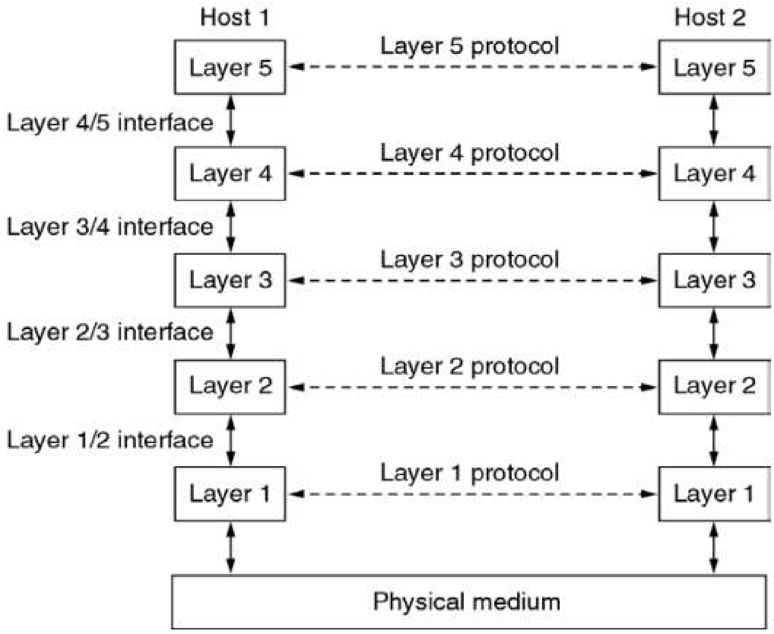
2. Financial services
3. Manufacturing. CAD, CAM etc.
4. Information services
5. Cellular telephone
6. Cable television
7. Teleconferencing
8. EDI
9. E-mail etc.

Network Software : Network Architecture

The first Computer Network were designed with the **hardware as the main (emphasis) concern** and **the Software as an afterthought**. This strategy no longer works. Network Software is now highly structured. The following section deals with network software structuring techniques.

Protocol Hierarchies





To reduce design complexities, most networks are organized as a stack of **layers** or levels, each one built upon the one below it.

The number of layers, the names, the contents of each layer, and the function of each layer may differ from network to network.

However, in all networks, **the purpose** of each layer is to offer certain **services**, to the higher layers, shielding those layers from the details of how the offered services are actually implemented.

Layer n on one machine carries on a conversation with **layer n on another machine**. The rules and conventions used in this conversation are collectively known as the **layer n protocol**.

(Basically, a protocol is an agreement between the communicating parties on how communication is to proceed.)

The active elements in each layer are often called **entities**. An **entity** can be a Software entity (e.g process), or a hardware entity (e.g. an intelligent I/O chip)

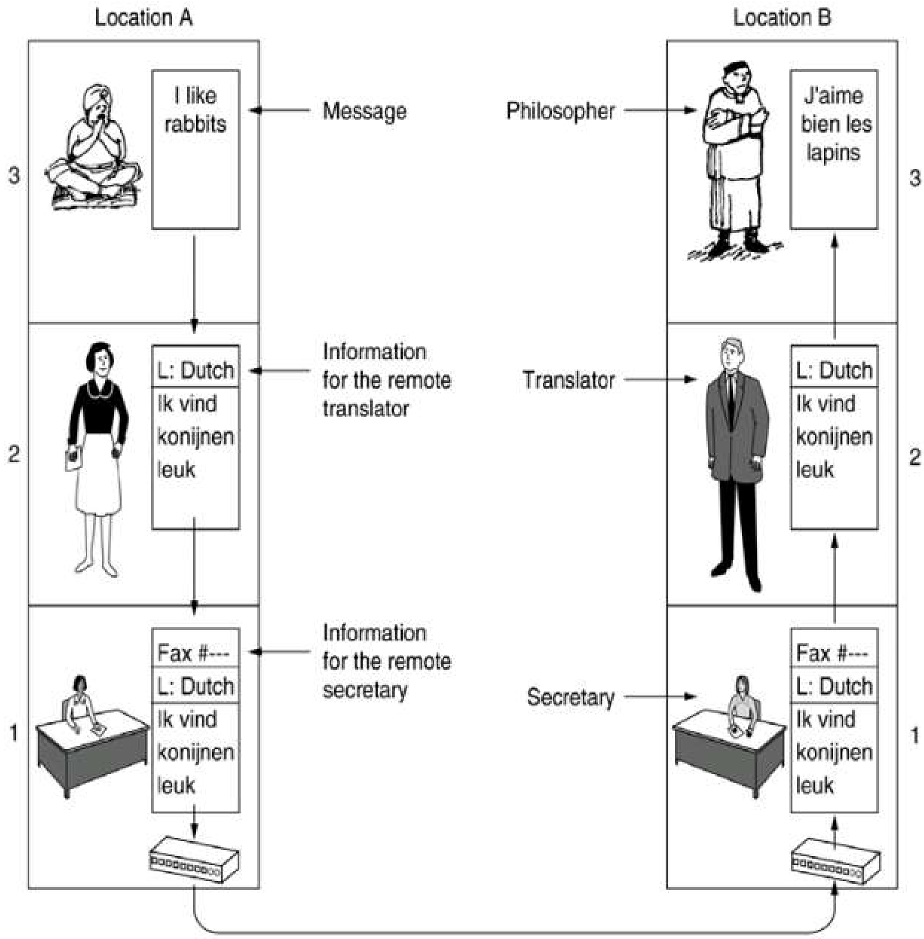
Entities in the same layer on different machines are called **peer entities**. In other words, it is the **peers** that communicate using the protocol.

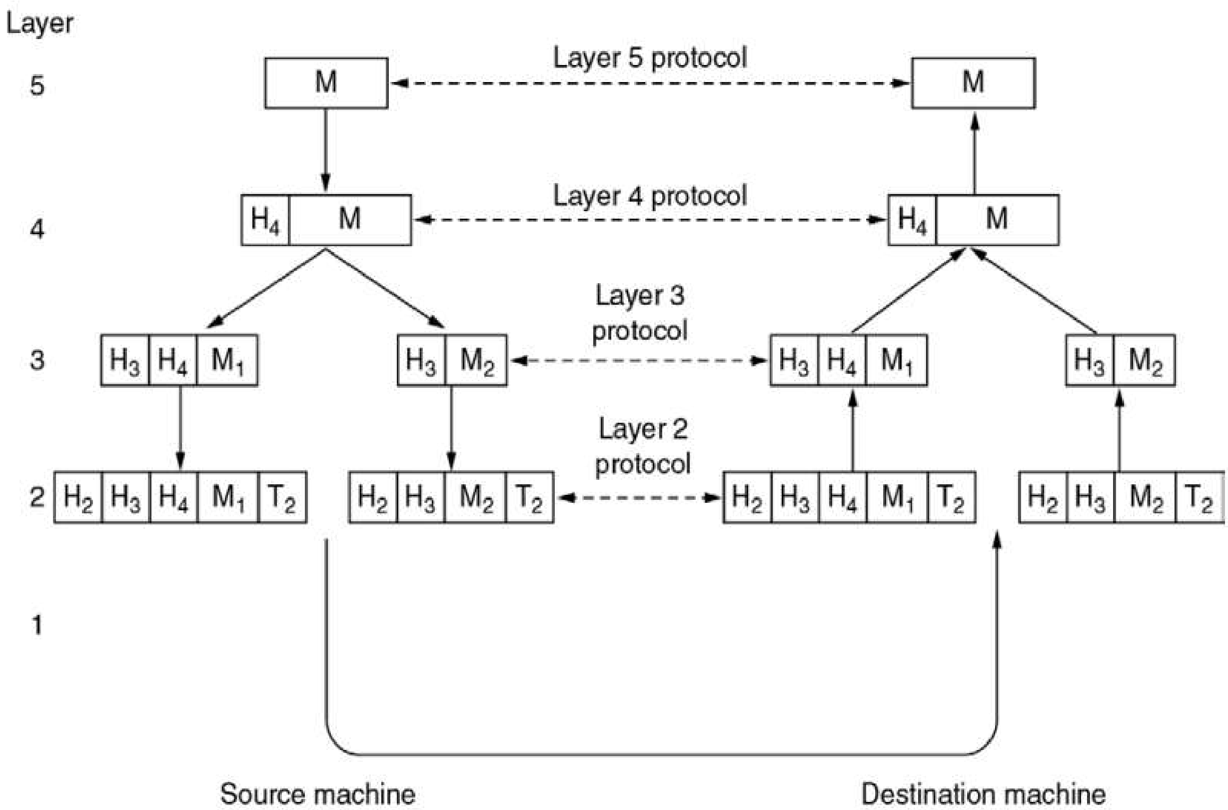
In reality, no data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes **data and control** information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the **physical medium** through which actual communication occurs.

Between each pair of adjacent layers there is an **interface**. The interface defines **which primitive operations and services the lower layer offers to the upper one**. Defining clean interfaces between adjacent layers makes it possible to replace the implementation of one layer with a completely new implementation (of course, offering the same services to upper layer as that of older implementation).

A set of layers and protocols is called a network architecture.

A list of protocols used by a certain system one protocol per layer, is called a **protocol stack**.





Example information flow supporting virtual communication in layer 5

Design Issue for the Layers

1. Mechanism for identifying senders and Receivers (addressing).
2. Rule for data transfer Simplex, Half-Duplex, Full-Duplex.
3. Error control (Detection and Correction).
4. Preserving order of messages (Sequencing).
5. How to keep a fast sender from swamping a slow receiver (flow control).
6. Inability of all process to accept arbitrarily long messages. (Dismantling → Transmit → Reassembling)

7. Multiplexing (Inconvenient and expensive to setup a separate connection for each pair)

8. In case of multiple path, Routing Algorithms are required to choose the optimal path.

9. Connection Establishment, maintenance and termination once the session is over.

Connection-oriented and Connectionless services

Layers may provide Connection oriented service or Connectionless service to the upper layers. The user is generally most concerned about which service that is given end-to-end, but the same alternatives may in principle be provided by any layers.

Connection oriented service: like e.g., the telephone.

1. Establish a connection (dial and get an answer)
2. Communicate over the connection (speak)
3. Delete the connection (hang up)

You get a "pipe" between the endpoints. You do not have to put an address on each piece of data, the address was provided at connection establishment.

The pipe might transfer a sequence of messages (with recognizable boundaries) or a sequence of bits. Normally the transfer is reliable (no data lost or duplicated). Normally the order is maintained.

Connectionless service: A complete address must be put on each message, e.g., like on letters (snail-mails). There is normally no guarantee that messages arrive in the same order they are sent.

Each service can be characterized by a **quality of service**. Some services are reliable in the sense that they never lose data. The service may be reliable or unreliable, the latter is called datagram service.

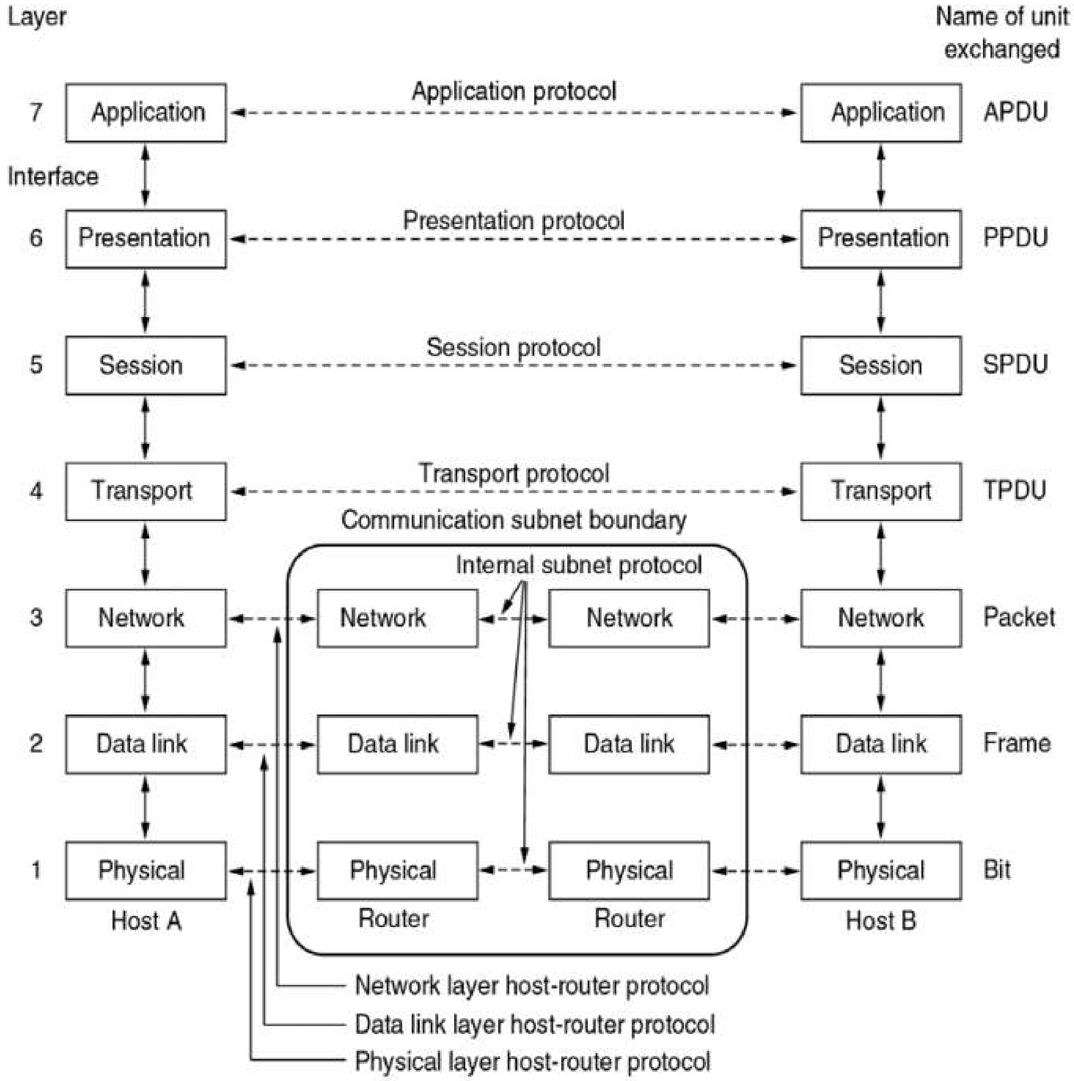
Datagram service may offer acknowledgements to the sender when a message is delivered: "Acknowledged Datagram Service". Then you know what was delivered and what was lost. A reply might also be required: "Request-Reply".

The protocols associated with the OSI model are rarely used any more, the model itself is actually quite general and still valid, and the features discussed a each layer are still very important. The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used.

ISO-OSI reference Model

The OSI model has seen layers. The principles that were applied to arrive at the seven layers can briefly be summarized as follows:

- A layer should be created where a different abstraction is needed.
- Each layer should perform a well-defined function.
- The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
- The layer boundaries should be chosen to minimize the information flow across the interfaces.
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.



1. **Physical layer** is concerned with transmitting raw bits from one node to the next over a communication channel.
   - Make sure bit 1-transmitted should be received as 1- not as 0.
   - How many volts to represent 0 & 1.
   - How many seconds a bit lasts.
   - Whether transmission may proceed simultaneously in both directions.
   - How the initial connection is **established** and how it is **torn down** when both sides are finished.
   - How many pins the network connector has and what each pin is used for.
   - The design issues here largely deals with mechanical, electrical and procedural interfaces and the physical transmission media.
   - **Defines Electrical, mechanical and functional properties of the interface (signal levels, rates, structures etc.) and transmission media.**

2. **Data Link layer**. The main task is to take a raw transmission facility and transform it into a line that appears free of undetected transmission errors to the network layer.

   - Break the input data up into data frames.
   - Transfers frames of a fixed maximum size. A frame is a bit sequence that starts and finishes with a certain (unique) bit pattern.
   - Process the **acknowledgement frames** sent back by the receiver.
   - It is up to this layer to solve the problem caused by damaged, (lost, and duplicate) frames.
   - How to keep **fast transmitter** from drowning a slow receiver in data (traffic Regulation mechanism)
   - How to control access to the shared channel (in Broadcast networks).
   - **Defines the methods for orderly and error-free transmission between two network nodes**.

3. **Network Layer is concerned with controlling the operation of the subset.**

   - How packets are routed from source to destination.
   - Presence of two many packets causes congestion in the subset – congestion control.
   - How many bits, character or packet are sent by each customer, to produce **billing information**.
   - Connection of **heterogeneous** networks.
   - **Defines functions for routing, multiplexing of packets, flow control and network supervision.**

4. **Transport Layer** . The basic function of the transport layer is to accept data from the session Layer, **split it up** into smaller units, if need be, and pass these to the network layer. And **Ensure** that all pieces arrive correctly at the other end.
   Data may have been reordered, doubled or lost on its way through the network. This layer fixes such errors.

   - The transport layer is true end-to-end layers (process-to-process delivery using port address) from source to destination.
   - **Responsible for the reliable transport of the traffic between two network end points as well as the assembly and disassembly of the messages**.

5. **Session Layer** allows users on different machines to establish session between them. (co-ordination between two communicating parties.)
   - A session allows ordinary data transport,
   - A session might be used to allow a user to log into a remote timesharing system or to transfer a file between two machines.
   - To manage dialogue control.
   - Token management and synchronization (2-hr. file transfers between two m/c with a 1-hr mean time between crashes), insert check points.

- **Handles conversation between the process at the two end points.**

6. **Presentation Layer** is concerned with the syntax and semantic of information transmitted.
   - Encoding (Encryption and Compression) data in a standard agreed upon way.
   - To convert encoded transmitted data into forms, which can be displayed on a video terminal or printers.
   - **Manages the differences in syntax among the various representation of information at both endpoints by putting data into a standardized format.**

7. **Application Layer** provides services such a file transfer, Remote file access; data management etc. that directly support the user.
   - **Functions to ensure that two application process co-operate to carry out the desired information processing at the two points**.

A computer network can be classified into different categories. Different criteria are used to classify computer networks. Following are the criteria widely used.

- Geographical spread
- Topology
- Ownership

Classification by Geographical Spread

Based on geographical spread, networks can be classified into the following three categories.

- Local Area Network (LAN)
- Metropolitan Area Network (MAN)
- Wide Area Network (WAN)

Local area network (LAN) LAN is a computer network that consists of few or more computers and other communication devices connected in the form of a network within a well-defined area such as a room or a building.
A typical example is a college or university computer network. Users in a LAN can share both hardware and sharable software resources. For example, hardware resources include expensive laser printer, plotter, fax machines, modem, etc. Almost all local area networks use a single communication media, as it restricted to a limited area. All network resources and their management activities are controlled using special system software called Network Operating System (NOS).

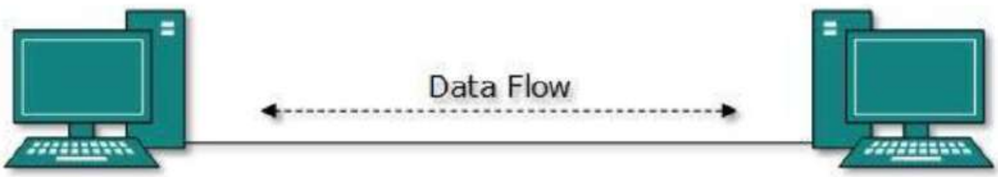
Metropolitan area network (MAN) MAN is a network more extensive than a LAN. The name metropolitan is due to the ability to cover a relatively larger area of a city, from a few tens to a maximum of hundred kilometers. Different hardware and transmission media often used in a MAN for efficient transmission of information.

Wide area network (WAN) WAN is a computer network that spans a large geographical area. It uses dedicated or switched connections to link computers in geographically remote locations. Wide area networks are implemented to connect a large number of LANs and MANs. Due to this reason, it is possible to see a large number of heterogeneous components in a wide area network. Different communication media used, and the network spreads across several national boundaries. Computers connected to a WAN often connected to a public network. They can also be connected through leased lines or satellite links. The

A Network Topology is the arrangement with which computer systems or network devices are connected to each other. Topologies may define both physical and logical aspect of the network. Both logical and physical topologies could be same or different in a same network.

Point-to-Point

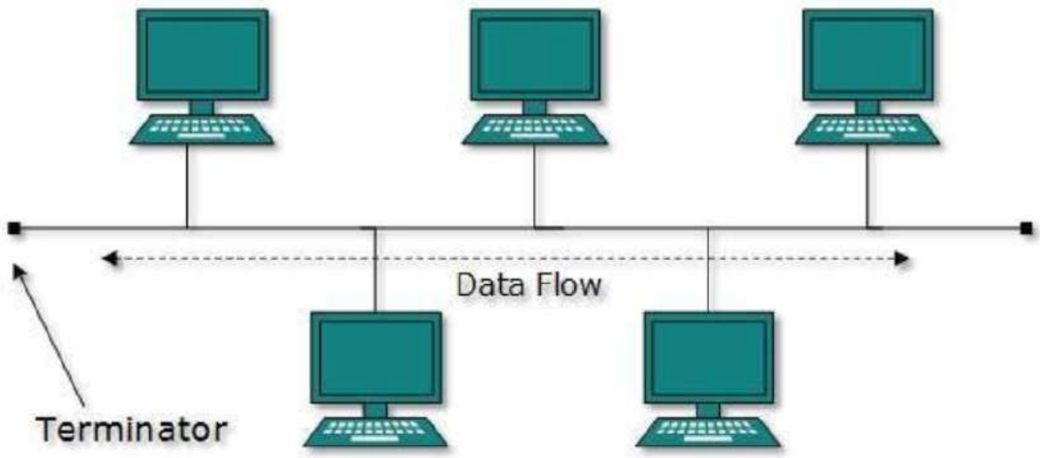
Point-to-point networks contains exactly two hosts such as computer, switches or routers, servers connected back to back using a single piece of cable. Often, the receiving end of one host is connected to sending end of the other and vice-versa.



If the hosts are connected point-to-point logically, then may have multiple intermediate devices. But the end hosts are unaware of underlying network and see each other as if they are connected directly.

Bus Topology

In case of Bus topology, all devices share single communication line or cable.Bus topology may have problem while multiple hosts sending data at the same time. Therefore, Bus topology either uses CSMA/CD technology or recognizes one host as Bus Master to solve the issue. It is one of the simple forms of networking where a failure of a device does not affect the other devices. But failure of the shared communication line can make all other devices stop functioning.

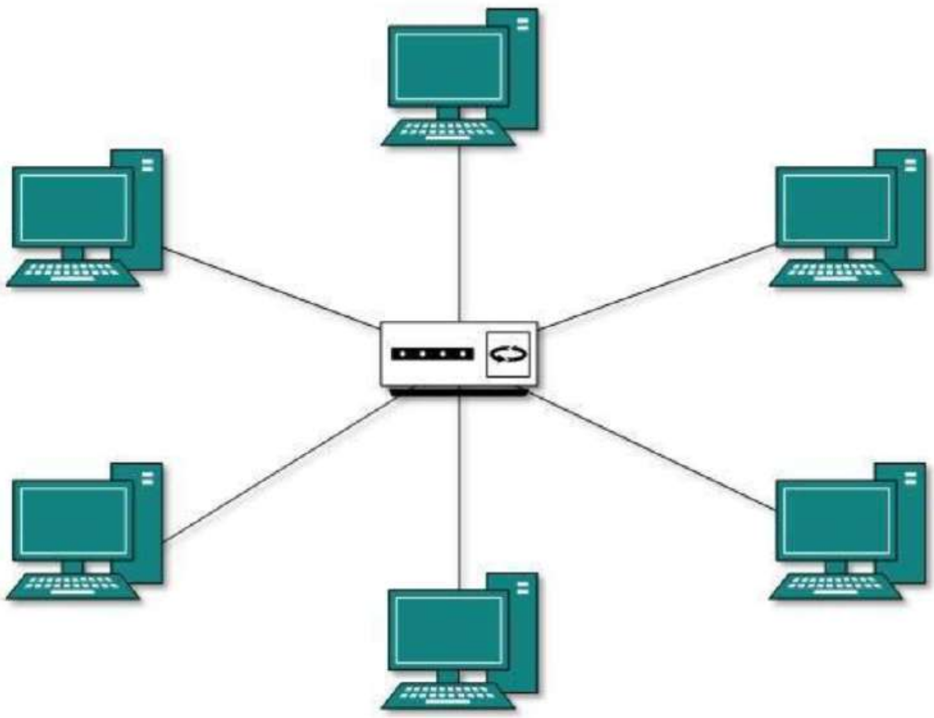


Both ends of the shared channel have line terminator. The data is sent in only one direction and as soon as it reaches the extreme end, the terminator removes the data from the line.

Star Topology

All hosts in Star topology are connected to a central device, known as hub device, using a point-to-point connection. That is, there exists a point to point connection between hosts and hub. The hub device can be any of the following:

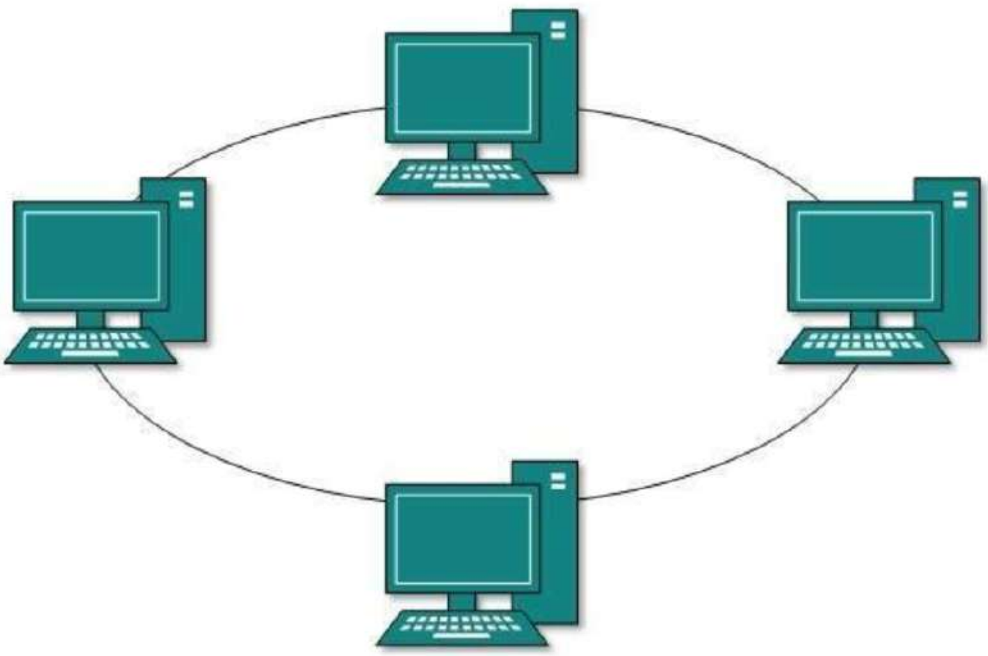
- Layer-1 device such as hub or repeater
- Layer-2 device such as switch or bridge
- Layer-3 device such as router or gateway



As in Bus topology, hub acts as single point of failure. If hub fails, connectivity of all hosts to all other hosts fails. Every communication between hosts, takes place through only the hub.Star topology is not expensive as to connect one more host, only one cable is required and configuration is simple.

Ring Topology

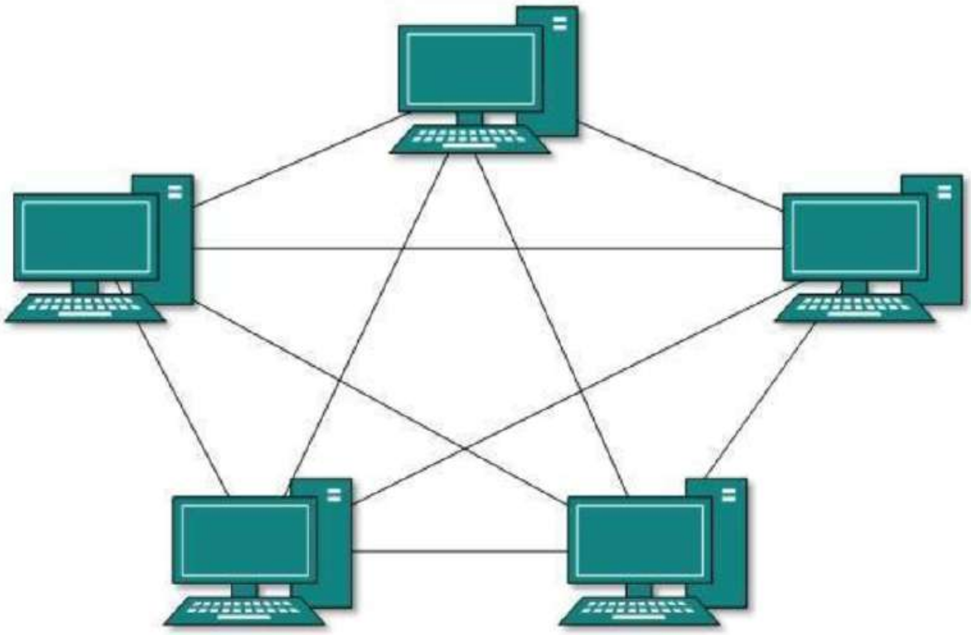
In ring topology, each host machine connects to exactly two other machines, creating a circular network structure. When one host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts. To connect one more host in the existing structure, the administrator may need only one more extra cable.



Failure of any host results in failure of the whole ring.Thus, every connection in the ring is a point of failure. There are methods which employ one more backup ring.

Mesh Topology

In this type of topology, a host is connected to one or multiple hosts.This topology has hosts in point-to-point connection with every other host or may also have hosts which are in point-to-point connection to few hosts only.



Hosts in Mesh topology also work as relay for other hosts which do not have direct point-to-point links. Mesh technology comes into two types:

- **Full Mesh**: All hosts have a point-to-point connection to every other host in the network. Thus for every new host n(n-1)/2 connections are required. It provides the most reliable network structure among all network topologies.
- **Partially Mesh**: Not all hosts have point-to-point connection to every other host. Hosts connect to each other in some arbitrarily fashion. This topology exists where we need to provide reliability to some hosts out of all.

ANALOG AND DIGITAL SIGNAL

One of the major functions of the physical layer is to move data in the form of electromagnetic signals across a transmission medium.

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

Analog and Digital Data

Data can be analog or digital. The term **analog data** refers to information that is continuous; **digital data** refers to information that has discrete states.

For example, an analog clock that has hour, minute, and second hands gives information in a continuous form; the movements of the hands are continuous. On the other hand, a digital clock that reports the hours and the minutes will change suddenly from 8:05 to 8:06.

Analog data, such as the sounds made by a human voice, take on continuous values. When someone speaks, an analog wave is created in the air. This can be captured by a microphone and converted to an analog signal or sampled and converted to a digital signal.

Digital data take on discrete values. For example, data are stored in computer memory in the form of Os and 1s. They can be converted to a digital signal or modulated into an analog signal for transmission across a medium.

Analog and Digital Signals

Like the data they represent, signals can be either analog or digital. An analog signal has infinitely many levels of intensity over a period of time.

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.

Periodic and Non periodic Signals

Both analog and digital signals can take one of two forms: periodic or non periodic.

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.

A non periodic signal changes without exhibiting a pattern or cycle that repeats over time.

Both analog and digital signals can be periodic or non periodic. In data communications, we commonly use periodic analog signals (because they need less bandwidth) and non periodic digital signals (because they can represent variation in data).

Periodic Analog Signals

Periodic analog signals can be classified as simple or composite.

A simple periodic analog signal, a sine wave, cannot be decomposed into simpler signals.
A composite periodic analog signal is composed of multiple sine waves.

Sine Wave

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow.

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

Peak Amplitude

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

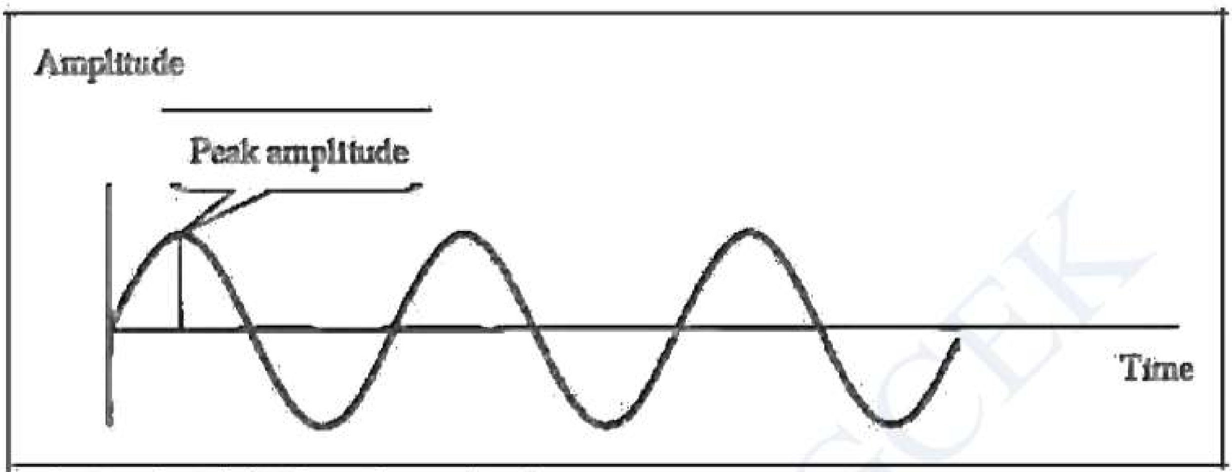
Period and Frequency

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

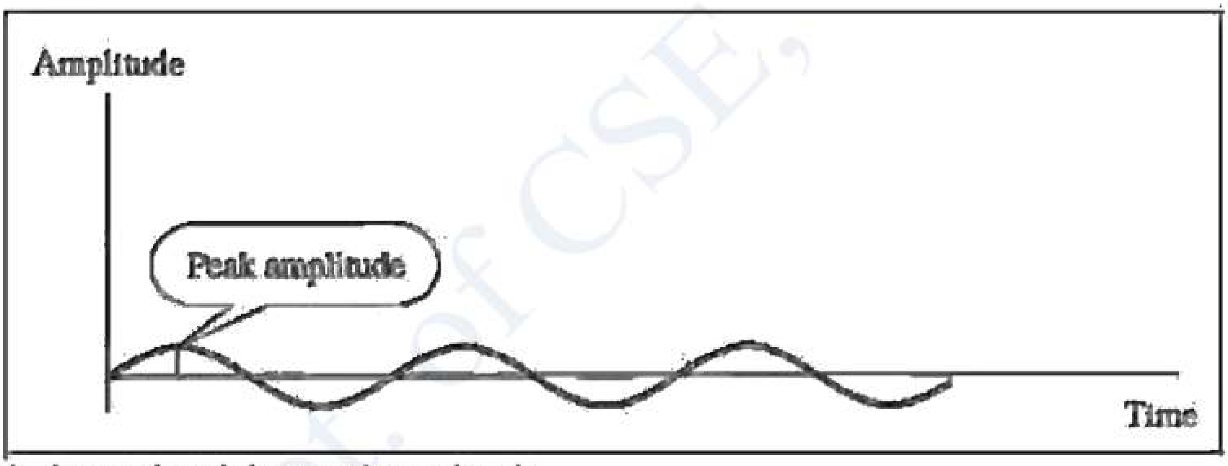
Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show.

$f = 1/T$ and $T = 1/f$

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



a. A signal with high peak amplitude



b. A signal with low peak amplitude

Fig. 2.3: signal with different amplitudes

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.

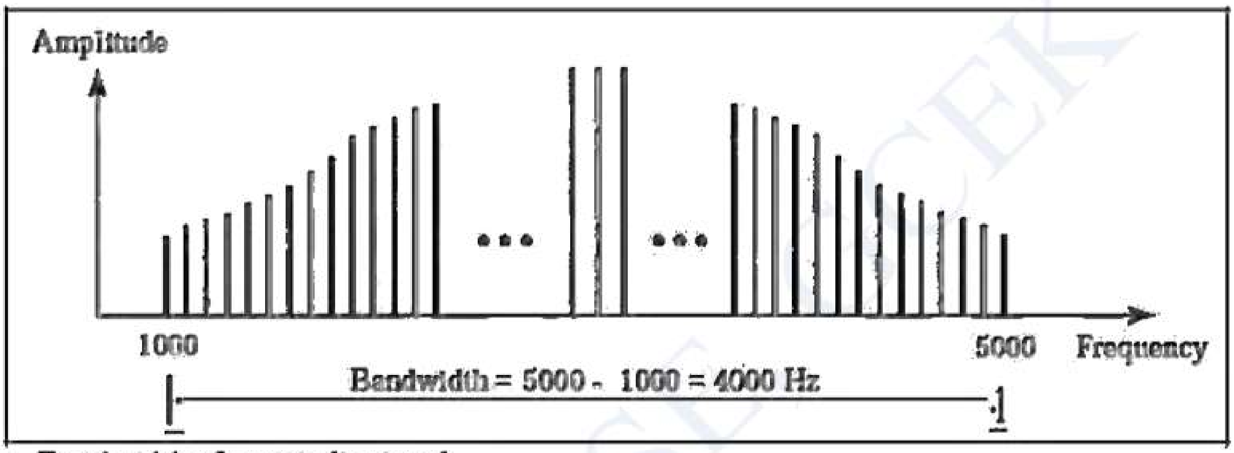
A single frequency sine wave is not useful in data communications; we need to send a composite signal, a signal made of many simple sine waves.

According to Fourier analysis, any composite signal is a combination of simple sine waves with different frequencies, amplitudes, and phases.

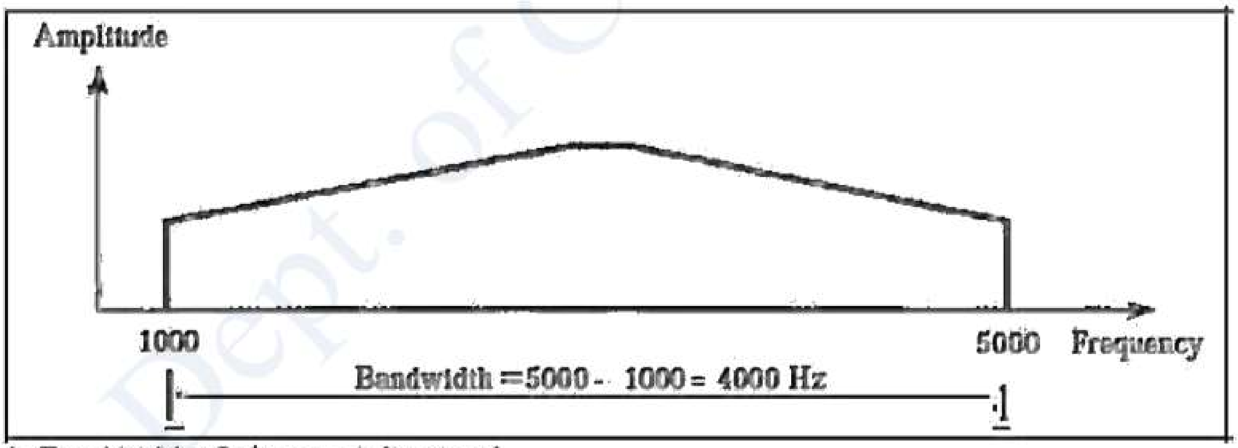
Bandwidth

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

The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal.



a. Bandwidth of a periodic signal



b. Bandwidth of a nonperiodic signal

Example :

A periodic signal has a bandwidth of 20 Hz. The highest frequency is 60 Hz. What is the lowest frequency? Draw the spectrum if the signal contains all frequencies of the same amplitude.

Solution:

Let f_h be the highest frequency, f_z the lowest frequency, and B the bandwidth.

Then $B = f_h - f_z$

$20 = 60 - f_z$ or $f_z = 60 - 20 = 40$ Hz

Digital Signal

In addition to being represented by an analog signal, information can also be represented by a digital signal. For example, 1 can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels.

In general, if a signal has L levels, each level needs $\mathbf{\log_2 L}$ bits.

Example:

A digital signal has eight levels. How many bits are needed per level?

Solution:

We calculate the number of bits from the formula:

Number of bits per level $= \log_2 8 = 3$

Each signal level is represented by 3 bits.

Bit Rate

Most digital signals are non periodic, and thus period and frequency are not appropriate characteristics. Another term-bit rate is used to describe digital signals.

The bit rate is the number of bits sent in 1s, expressed in bits per second (**bps**).

Example :

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

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

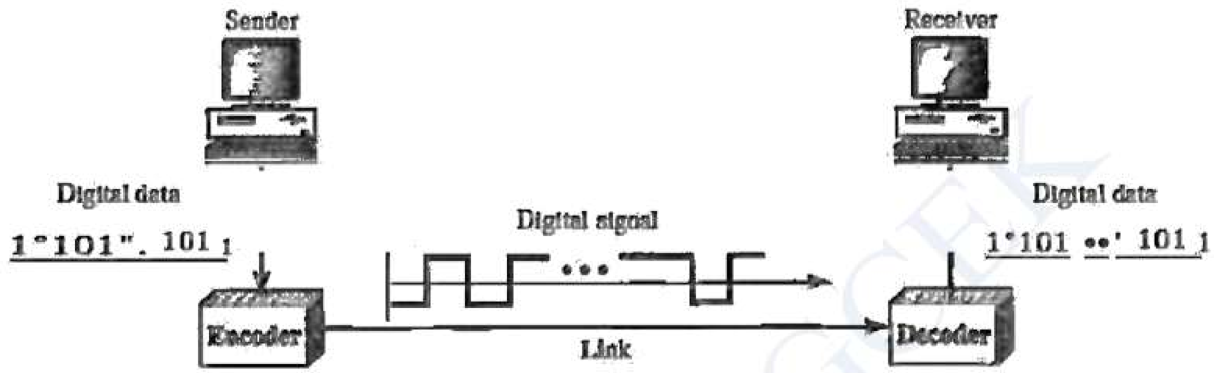
Bit Length

We discussed the concept of the wavelength for an analog signal: the distance one cycle occupies on the transmission medium. We can define something similar for a digital signal: the bit length. The bit length is the distance one bit occupies on the transmission medium.

Bit length=propagation speed x bit duration

DIGITAL TRANSMISSION

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 mayor may not be needed.



Line Coding

Line coding is the process of converting digital data to digital signals.

We assume that data, in the form of text, numbers, graphical images, audio, or video, are stored in computer memory as sequences of bits.

Line coding converts a sequence of bits to a digital signal. At the sender, digital data are

encoded into a digital signal; at the receiver, the digital data are recreated by decoding the digital signal.

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

S =c xNx 1/r baud

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

Example

A signal is carrying data in which one data element is encoded as one signal element (r = 1). If the bit rate is 100 kbps, what is the average value of the baud rate if c is between 0 and 1?

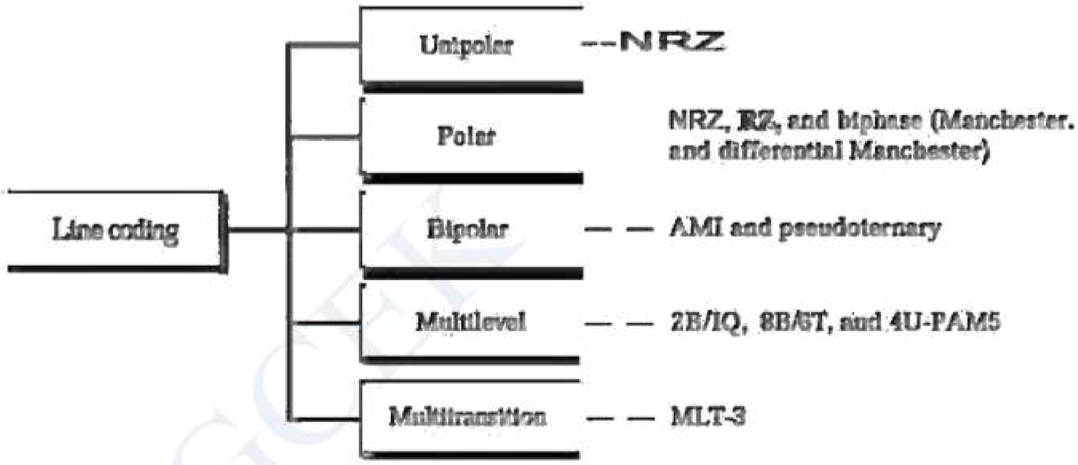
Solution:

We assume that the average value of c is ½. The baud rate is then

S =c x N x1/r = ½ x 100, 000 x 1= 50, 000 = 50 Kbaud

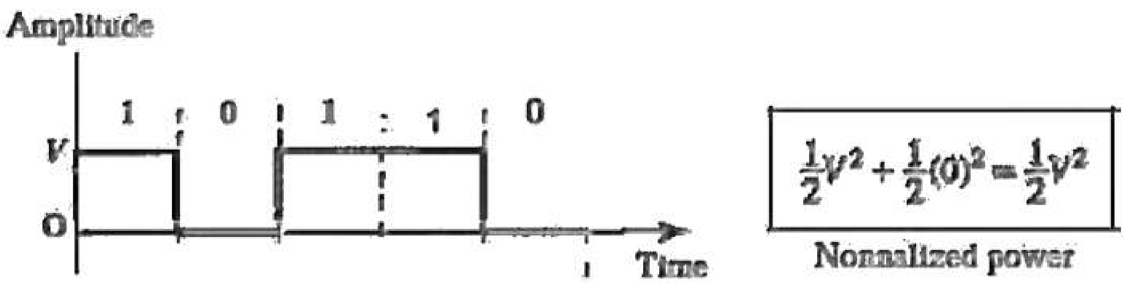
Line Coding Schemes

We can roughly divide line coding schemes into five broad categories, as shown in below:



Unipolar Scheme

In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below. **NRZ** (Non-Return-to-Zero): Traditionally, a unipolar scheme was designed as a non-return-to-zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. ***It is called NRZ because the signal does not return to zero at the middle of the bit***.



Polar Schemes

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.

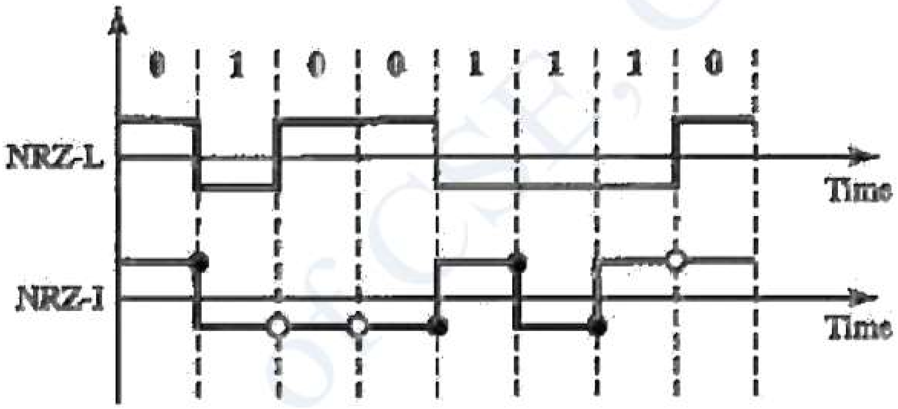
Non-Return-to-Zero (NRZ):

In polar NRZ encoding, we use two levels of voltage amplitude. We can have two versions of polar NRZ: NRZ-Land NRZ-I.

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.**

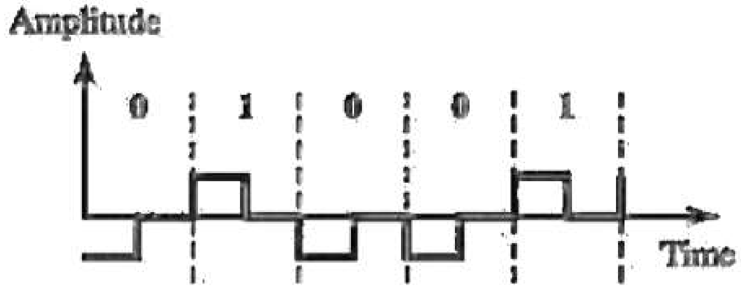
The synchronization problem (sender and receiver clocks are not synchronized) also exists in both schemes. Again, this problem is more serious in **NRZ-L** than in **NRZ-I**. While a long sequence of 0's can cause a problem in both schemes, a long sequence of 1s affects only **NRZ-L**.

Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. NRZ-I does not have this problem. Both schemes have an average signal rate of **N/2 Bd**.



○ No inversion: Next bit is 0 ● Inversion: Next bit is 1

Return to Zero (RZ) The main problem with NRZ encoding occurs when the sender and receiver clocks are not synchronized. The receiver does not know when one bit has ended and the next bit is starting. One solution is the return-to-zero (RZ) scheme, which uses three values: positive, negative, and zero**. In RZ, the signal changes not between bits but during the bit**.



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.