

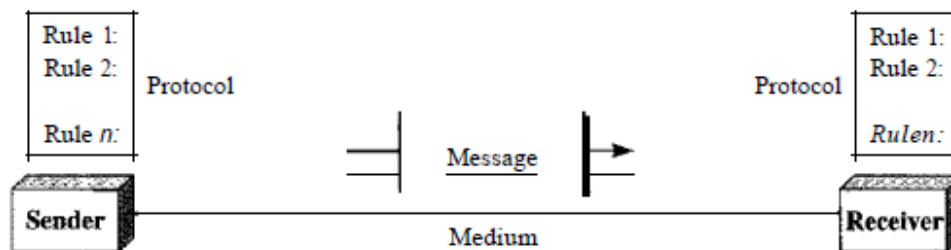
UNIT -I

Introduction to Computer Networks

1.1 Data Communication: When we communicate, we are sharing information. This sharing can be local or remote. Between individuals, local communication usually occurs face to face, while remote communication takes place over distance.

1.1.1 Components:

A data communications system has five components.



1. Message. The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.
2. Sender. The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.
3. Receiver. The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.
4. Transmission medium. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves
5. Protocol. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

1.1.2 Data Representation:

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

Text:

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

Numbers:

Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations. Appendix B discusses several different numbering systems.

Images:

Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the *resolution*. For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image (better resolution), but more memory is needed to store the image. After an image is divided into pixels, each pixel is assigned a bit pattern. The size and the value of the pattern depend on the image. For an image made of only black and white dots (e.g., a chessboard), a 1-bit pattern is enough to represent a pixel. If an image is not made of pure white and pure black pixels, you can increase the size of the bit pattern to include gray scale. For example, to show four levels of gray scale, you can use 2-bit patterns. A black pixel can be represented by 00, a dark gray pixel by 01, a light gray pixel by 10, and a white pixel by 11. There are several methods to represent color images. One method is called RGB, so called because each color is made of a combination of three primary colors: *red*, *green*, and *blue*. The intensity of each color is measured, and a bit pattern is assigned to it. Another method is called YCM, in which a color is made of a combination of three other primary colors: *yellow*, *cyan*, and *magenta*.

Audio:

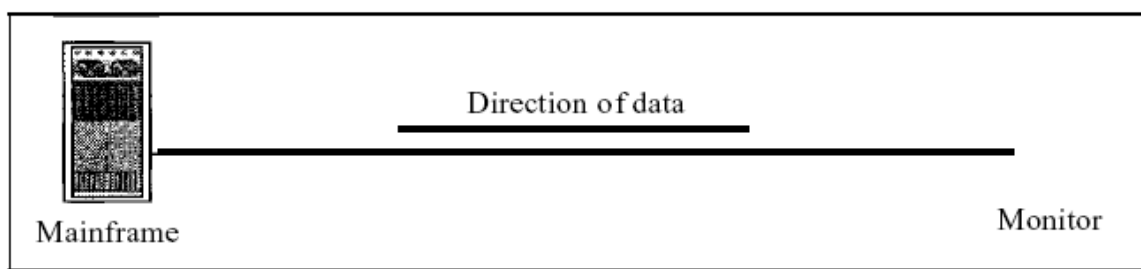
Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal. In Chapters 4 and 5, we learn how to change sound or music to a digital or an analog signal.

Video:

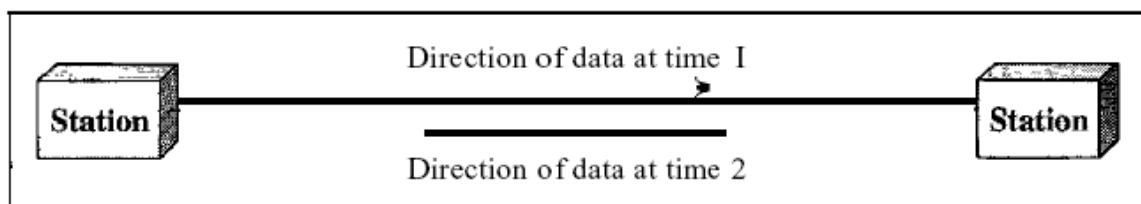
Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion. Again we can change video to a digital or an analog signal.

1.1.3 Data Flow

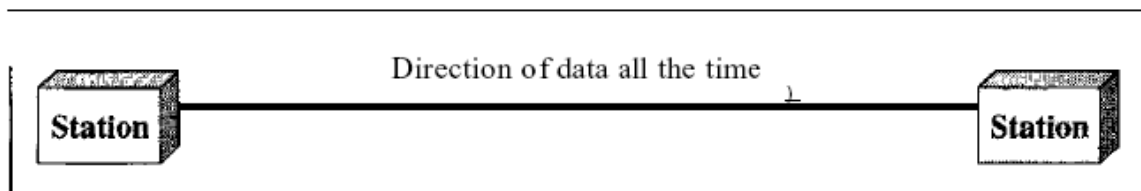
Communication between two devices can be simplex, half-duplex, or full-duplex as shown in Figure



a. Simplex



b. Half-duplex



c. Full-duplex

Simplex:

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

Half-Duplex:

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa. The half-duplex mode is like a one-lane road with traffic allowed in both directions.

When cars are traveling in one direction, cars going the other way must wait. In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkie-talkies and CB (citizens band) radios are both half-duplex systems.

The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

Full-Duplex:

In full-duplex both stations can transmit and receive simultaneously (see Figure c). The full-duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in one direction share the capacity of the link; with signals going in the other direction. This sharing can occur in two ways: Either the link must contain two physically separate transmission paths, one for sending and the other for receiving; or the capacity of the channel is divided between signals traveling in both directions. One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time. The full-duplex mode is used when communication in both directions is required all the time. The capacity of the channel, however, must be divided between the two directions.

1.2 NETWORKS

A network is a set of devices (often referred to as *nodes*) 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.

1.2.1 Distributed Processing

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

1.2.2 Network Criteria

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

Performance:

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software. Performance is often evaluated by two networking metrics: throughput and delay. We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability:

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

Security:

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

1.2.3 Physical Structures:

Type of Connection

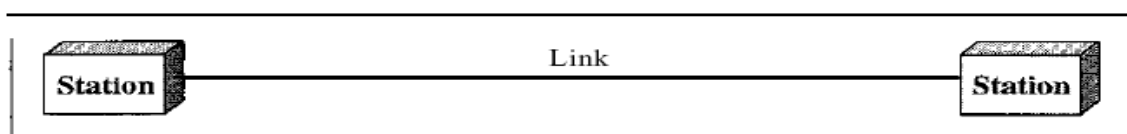
A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. For visualization purposes, it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time. There are two possible types of connections: point-to-point and multipoint.

Point-to-Point

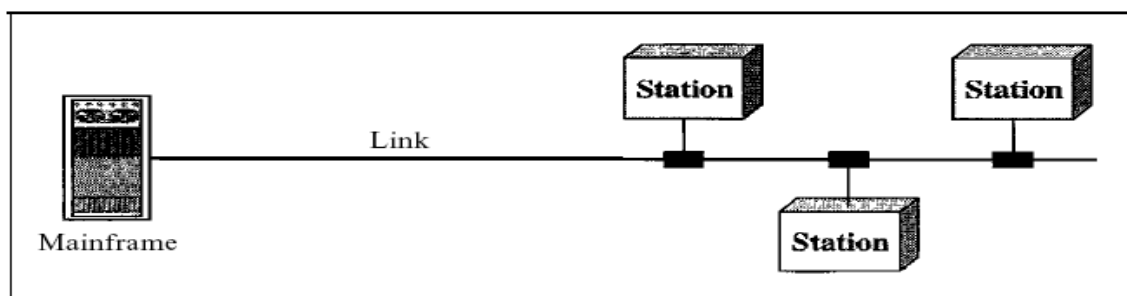
A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible. When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

Multipoint

A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.



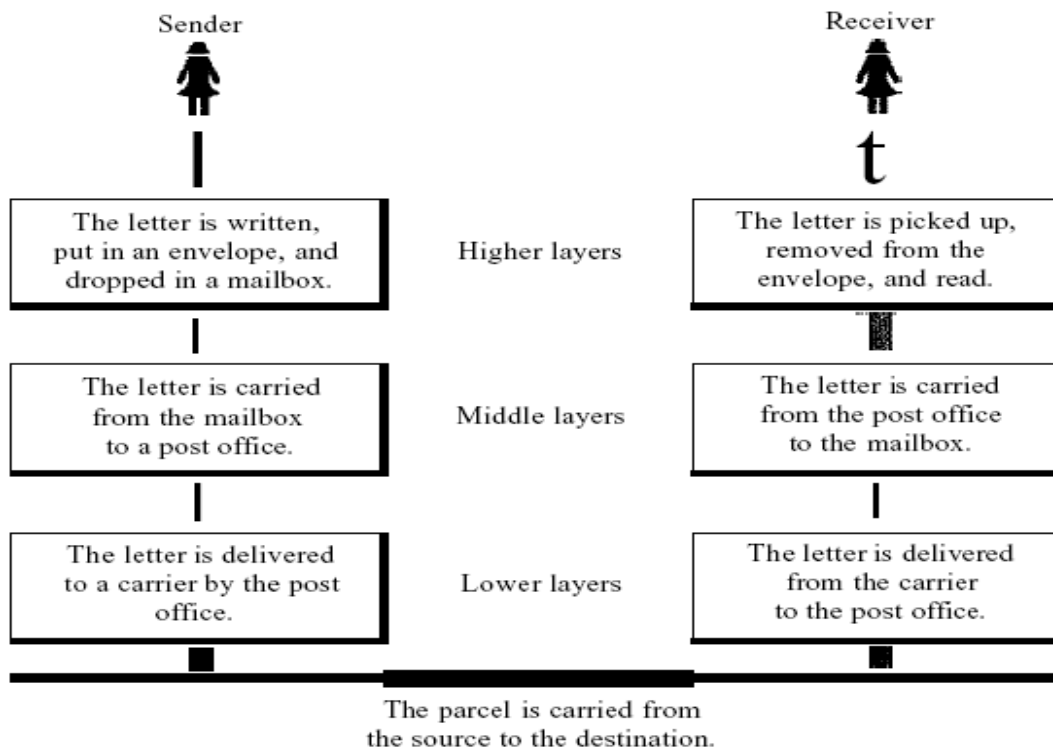
a. Point-to-point



b. Multipoint

1.5 LAYERED TASKS

We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail. The process of sending a letter to a friend would be complex if there were no services available from the post office. Below Figure shows the steps in this task.



Sender, Receiver, and Carrier

In Figure we have a sender, a receiver, and a carrier that transports the letter. There is a hierarchy of tasks.

At the Sender Site

Let us first describe, in order, the activities that take place at the sender site.

- o Higher layer. The sender writes the letter, inserts the letter in an envelope, writes the sender and receiver addresses, and drops the letter in a mailbox.
- o Middle layer. The letter is picked up by a letter carrier and delivered to the post office.
- o Lower layer. The letter is sorted at the post office; a carrier transports the letter.

On the Way: The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the Receiver Site

- o Lower layer. The carrier transports the letter to the post office.
- o Middle layer. The letter is sorted and delivered to the recipient's mailbox.
- o Higher layer. The receiver picks up the letter, opens the envelope, and reads it.

1.6 The OSI Reference Model:

The OSI model (minus the physical medium) is shown in Fig. This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995). The model is called the ISO-OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems.

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

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. 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.

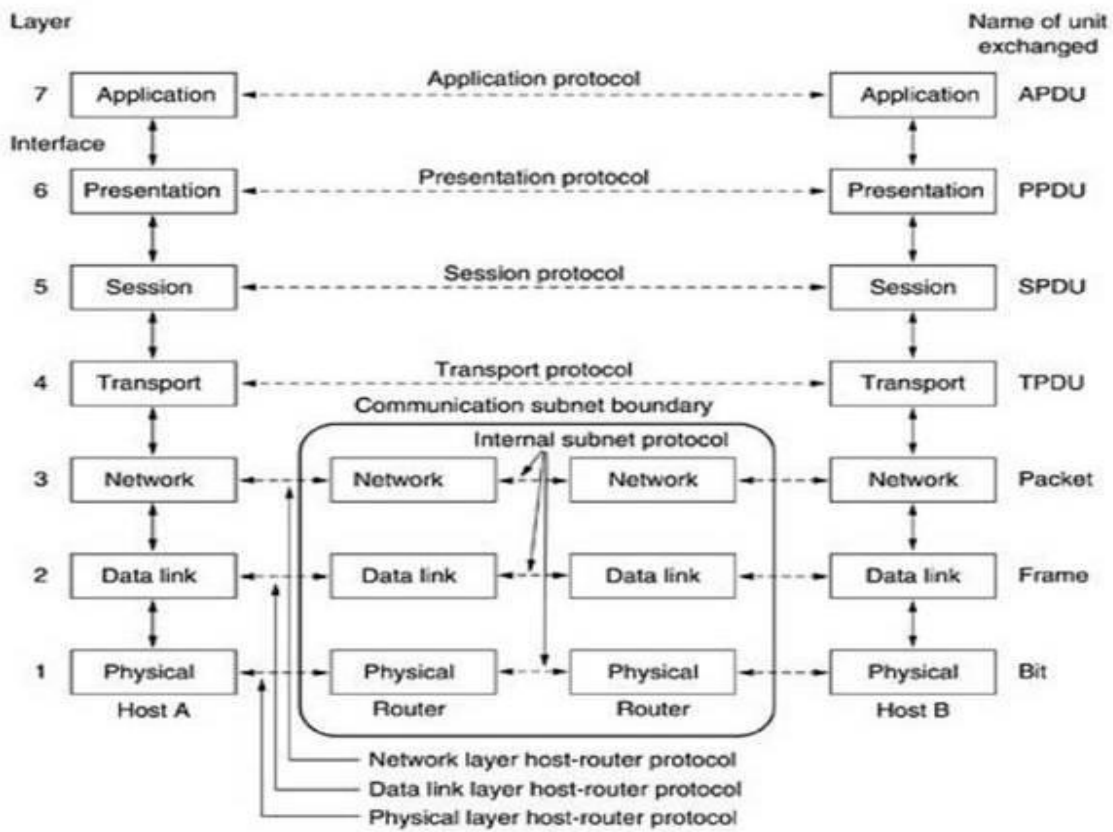


Fig.4: The OSI reference model

The Physical Layer:

The physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit.

The Data Link Layer:

The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break up the input data into data frames (typically a few hundred or a few thousand bytes) and transmits the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgement frame.

Another issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism is often needed to let the transmitter know how much buffer space the receiver has at the moment. Frequently, this flow regulation and the error handling are integrated.

The Network Layer:

The network layer controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into" the network and rarely changed. They can also be determined at the start of each conversation, for example, a terminal session (e.g., a login to a remote machine). Finally, they can be highly dynamic, being determined anew for each packet, to reflect the current network load.

If too many packets are present in the subnet at the same time, they will get in one another's way, forming bottlenecks. The control of such congestion also belongs to the network layer. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected. In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

The Transport Layer:

The basic function of the transport layer is to accept data from above, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology. The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service are the transporting of isolated messages, with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established.

The transport layer is a true end-to-end layer, all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers,

the protocols are between each machine and its immediate neighbours, and not between the ultimate source and destination machines, which may be separated by many routers.

The Session Layer:

The session layer allows users on different machines to establish sessions between them. Sessions offer various services, including dialog control (keeping track of whose turn it is to transmit), token management (preventing two parties from attempting the same critical operation at the same time), and synchronization (check pointing long transmissions to allow them to continue from where they were after a crash).

The Presentation Layer:

The presentation layer is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used "on the wire." The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records), to be defined and exchanged.

The Application Layer:

The application layer contains a variety of protocols that are commonly needed by users. One widely-used application protocol is HTTP (Hypertext Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

1.7 The TCP/IP Reference Model:

The TCP/IP reference model was developed prior to OSI model. The major design goals of this model were,

1. To connect multiple networks together so that they appear as a single network.
2. To survive after partial subnet hardware failures.
3. To provide a flexible architecture.

Unlike OSI reference model, TCP/IP reference model has only 4 layers. They are,

1. Host-to-Network Layer
2. Internet Layer

3. Transport Layer

4. Application Layer

Application Layer

Transport Layer

Internet Layer

Host-to-Network Layer

Host-to-Network Layer:

The TCP/IP reference model does not really say much about what happens here, except to point out that the host has to connect to the network using some protocol so it can send IP packets to it. This protocol is not defined and varies from host to host and network to network.

Internet Layer:

This layer, called the internet layer, is the linchpin that holds the whole architecture together. Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet.

The internet layer defines an official packet format and protocol called IP (Internet Protocol). The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer. Fig. shows this correspondence.

The Transport Layer:

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in the OSI transport layer. Two end-to-end transport protocols have been defined here. The first one, TCP (Transmission Control Protocol), is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It fragments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control

to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

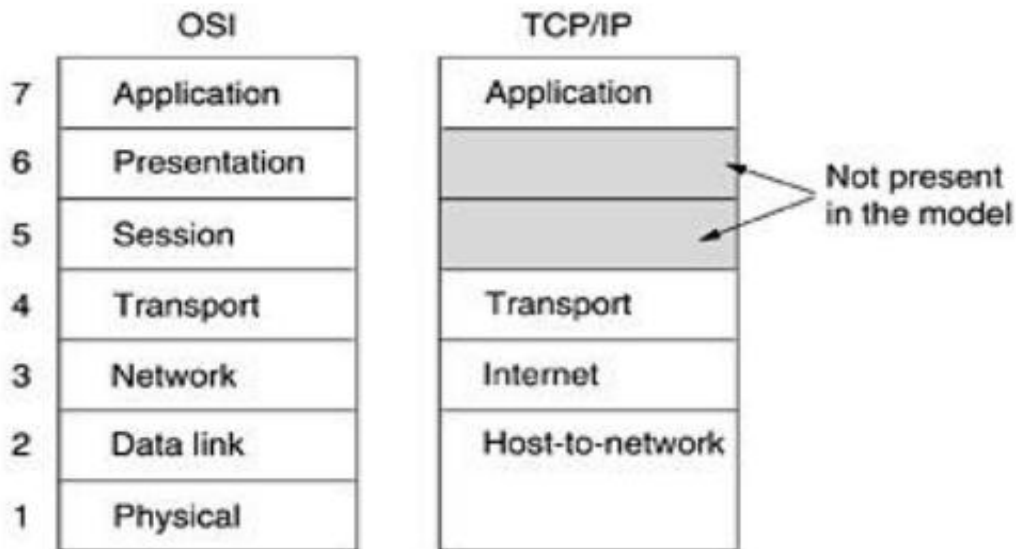


Fig.1: The TCP/IP reference model.

The second protocol in this layer, UDP (User Datagram Protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Fig.2. Since the model was developed, IP has been implemented on many other networks.

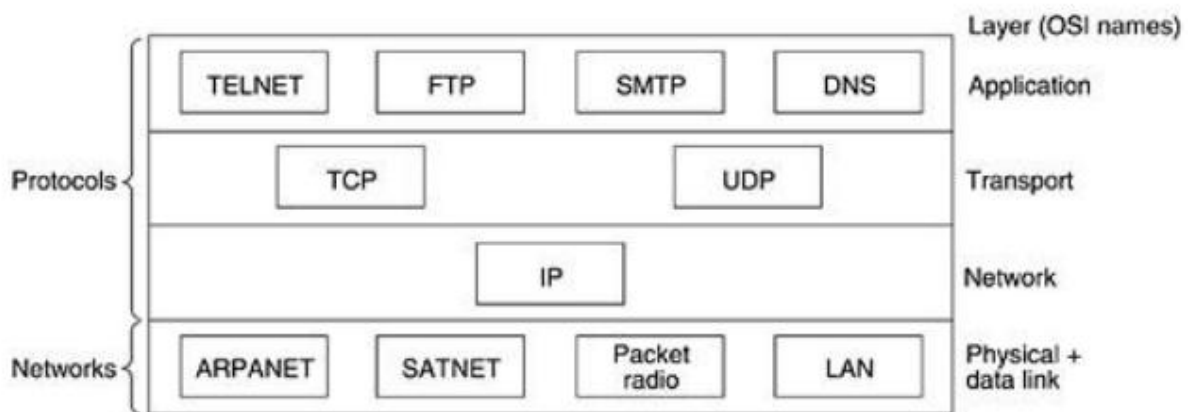


Fig.2: Protocols and networks in the TCP/IP model initially.

The Application Layer:

The TCP/IP model does not have session or presentation layers. On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP), as shown in Fig.6.2. The virtual terminal protocol allows a user on one machine to log onto a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a kind of file transfer, but later a specialized protocol (SMTP) was developed for it. Many other protocols have been added to these over the years: the Domain Name System (DNS) for mapping host names onto their network addresses, NNTP, the protocol for moving USENET news articles around, and HTTP, the protocol for fetching pages on the World Wide Web, and many others.

Comparison of the OSI and TCP/IP Reference Models:

The OSI and TCP/IP reference models have much in common. Both are based on the concept of a stack of independent protocols. Also, the functionality of the layers is roughly similar. For example, in both models the layers up through and including the transport layer are there to provide an end-to-end, network-independent transport service to processes wishing to communicate. These layers form the transport provider. Again in both models, the layers above transport are application-oriented users of the transport service. Despite these fundamental similarities, the two models also have many differences. Three concepts are central to the OSI model:

1. Services.
2. Interfaces.
3. Protocols.

Probably the biggest contribution of the OSI model is to make the distinction between these three concepts explicit. Each layer performs some services for the layer above it. The service definition tells what the layer does, not how entities above it access it or how the layer works. It defines the layer's semantics.

A layer's interface tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside.

Finally, the peer protocols used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higher layers.

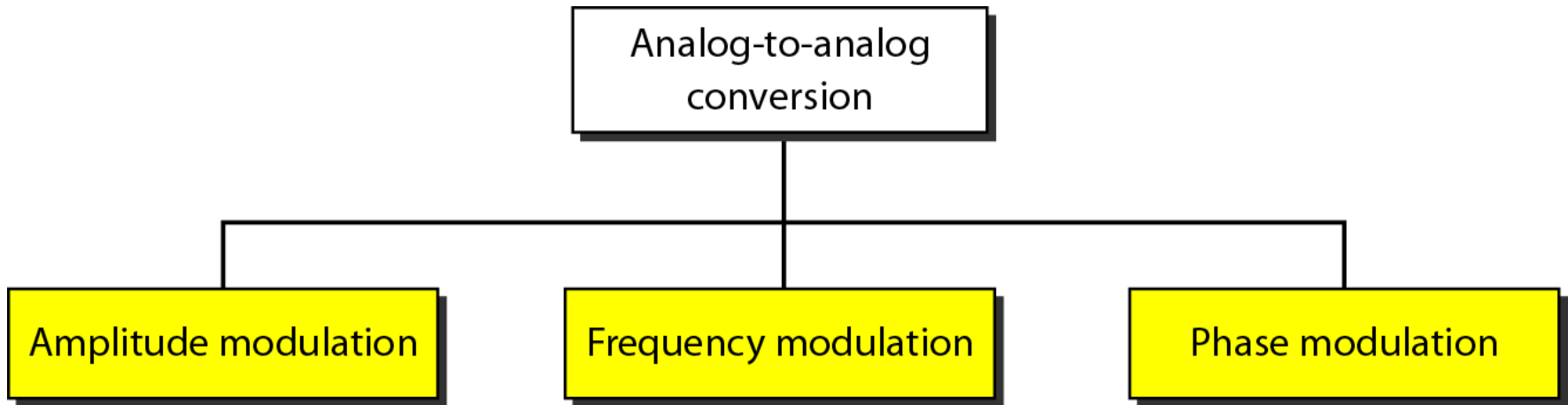
The TCP/IP model did not originally clearly distinguish between service, interface, and protocol, although people have tried to retrofit it after the fact to make it more OSI-like. For example, the only real services offered by the internet layer are SEND IP PACKET and RECEIVE IP PACKET.

As a consequence, the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes. Being able to make such changes is one of the main purposes of having layered protocols in the first place. The OSI reference model was devised before the corresponding protocols were invented. This ordering means that the model was not biased toward one particular set of protocols, a fact that made it quite general. The downside of this ordering is that the designers did not have much experience with the subject and did not have a good idea of which functionality to put in which layer.

Another difference is in the area of connectionless versus connection-oriented communication. The OSI model supports both connectionless and connection-oriented communication in the network layer, but only connection-oriented communication in the transport layer, where it counts (because the transport service is visible to the users). The TCP/IP model has only one mode in the network layer (connectionless) but supports both modes in the transport layer, giving the users a choice. This choice is especially important for simple request-response protocols.

Analog Modulation

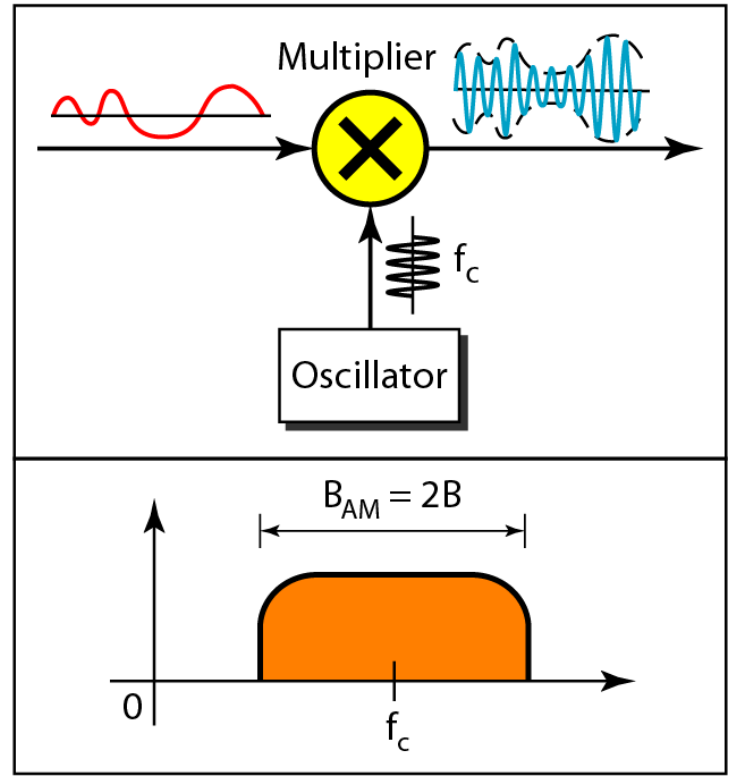
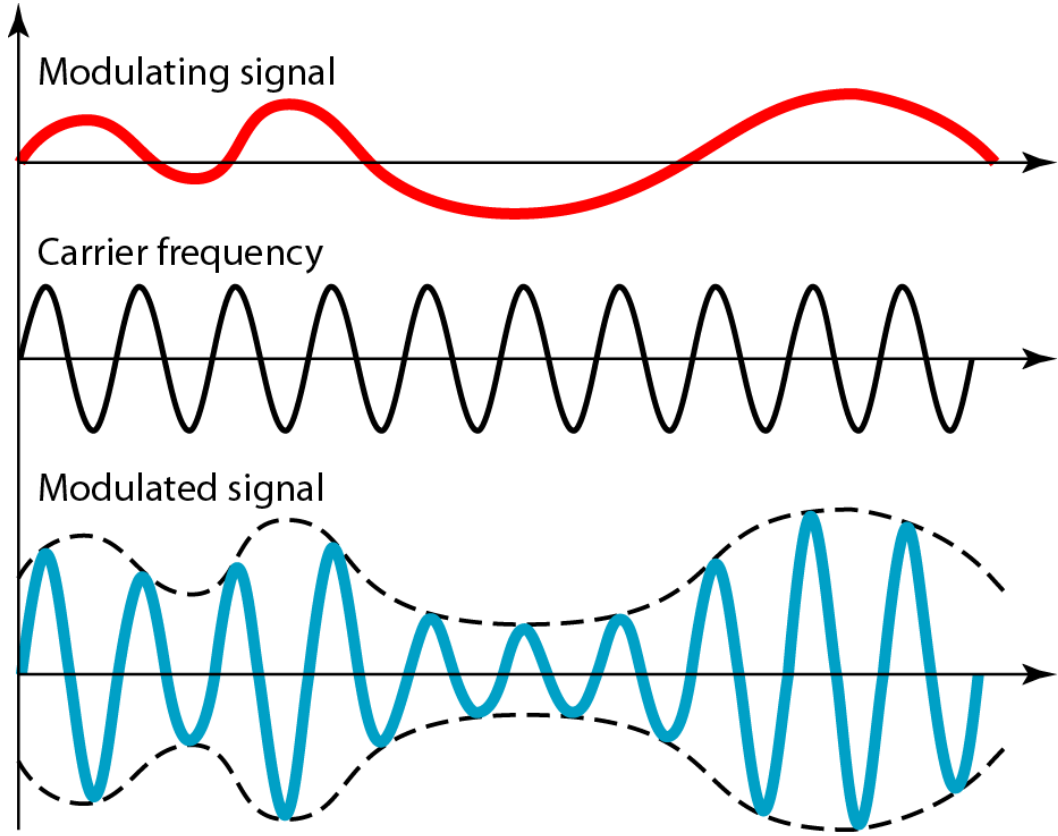
Types of analog-to-analog modulation



Amplitude Modulation

- A carrier signal is modulated only in amplitude value
- The modulating signal is the envelope of the carrier
- The required bandwidth is $2B$, where B is the bandwidth of the modulating signal
- Since on both sides of the carrier freq. f_c , the spectrum is identical, we can discard one half, thus requiring a smaller bandwidth for transmission.

Amplitude modulation

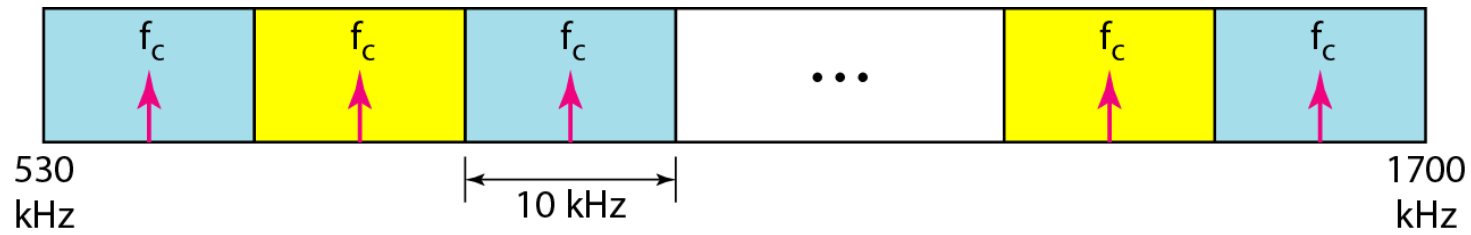




Note

**The total bandwidth required for AM
can be determined
from the bandwidth of the audio
signal: $B_{AM} = 2B$.**

AM band allocation



Frequency Modulation

- The modulating signal changes the freq. f_c of the carrier signal
- The bandwidth for FM is high
- It is approx. 10x the signal frequency

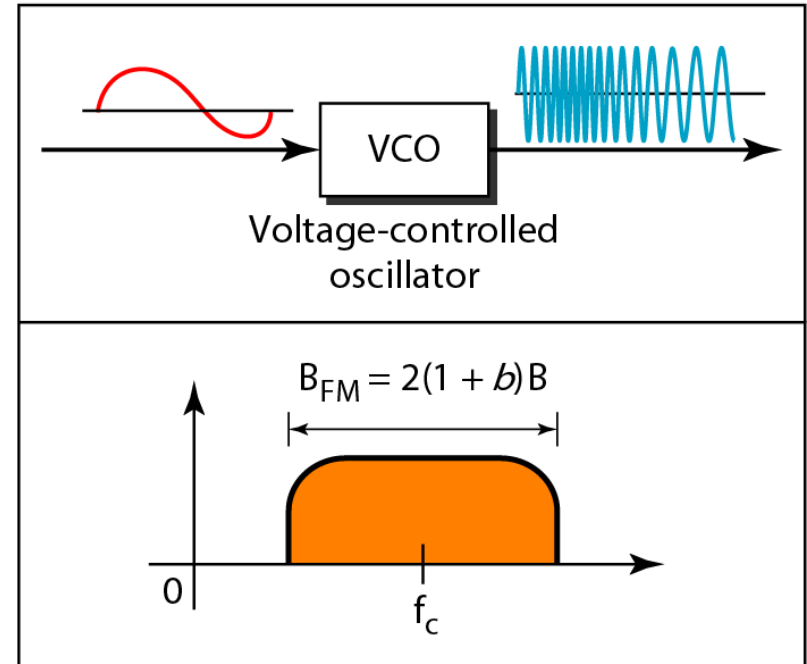
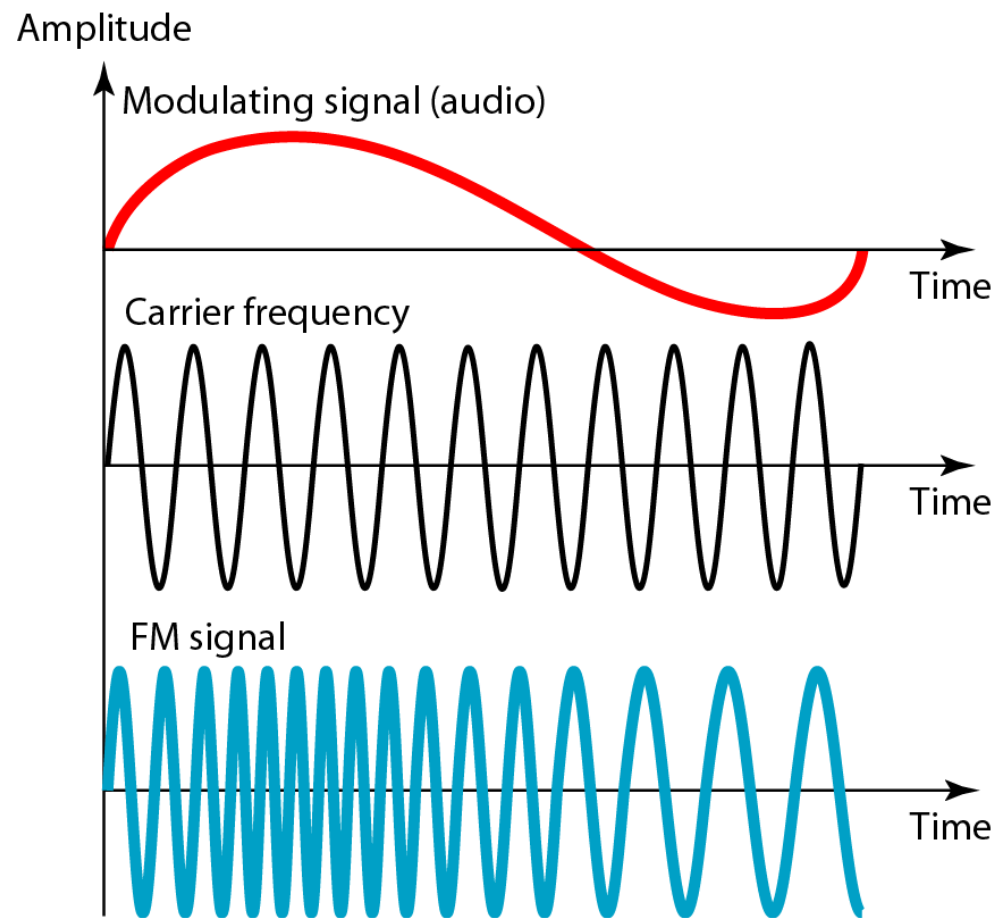


Note

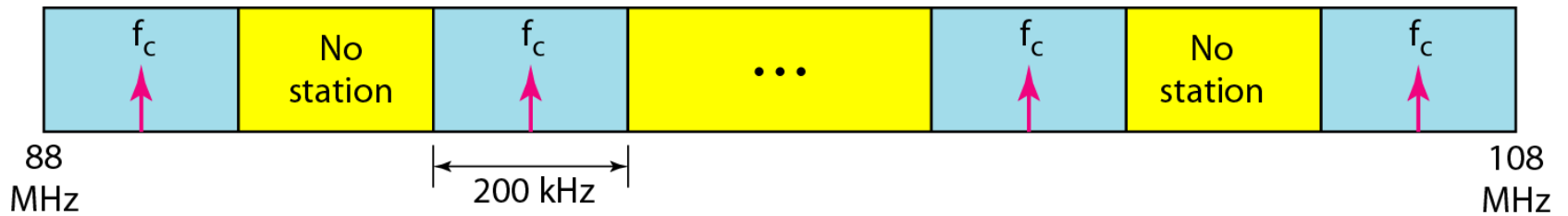
The total bandwidth required for FM can be determined from the bandwidth of the audio signal: $B_{FM} = 2(1 + \beta)B$.

Where β is usually 4.

Frequency modulation



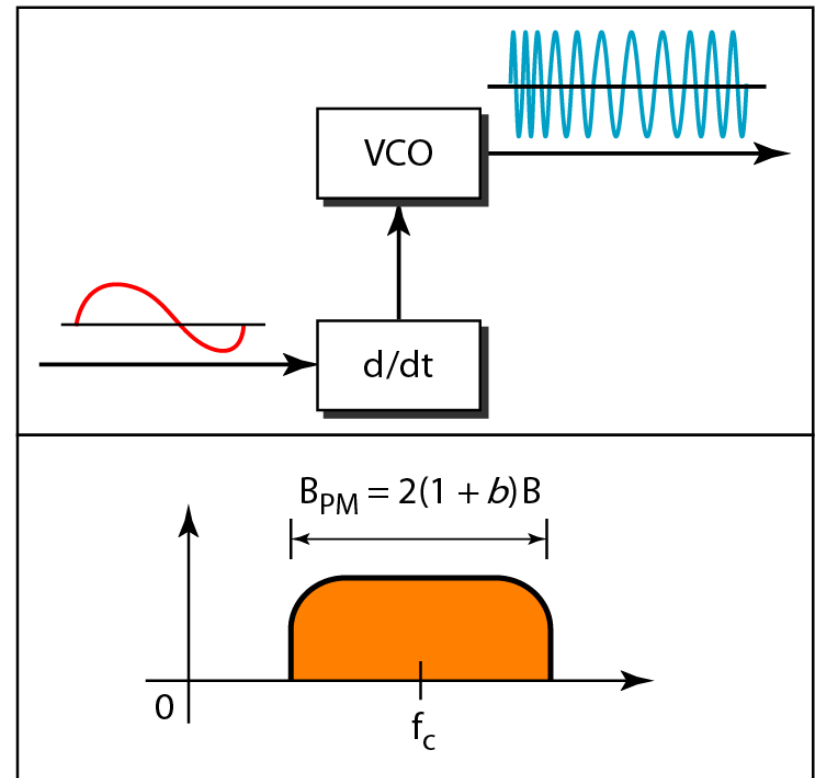
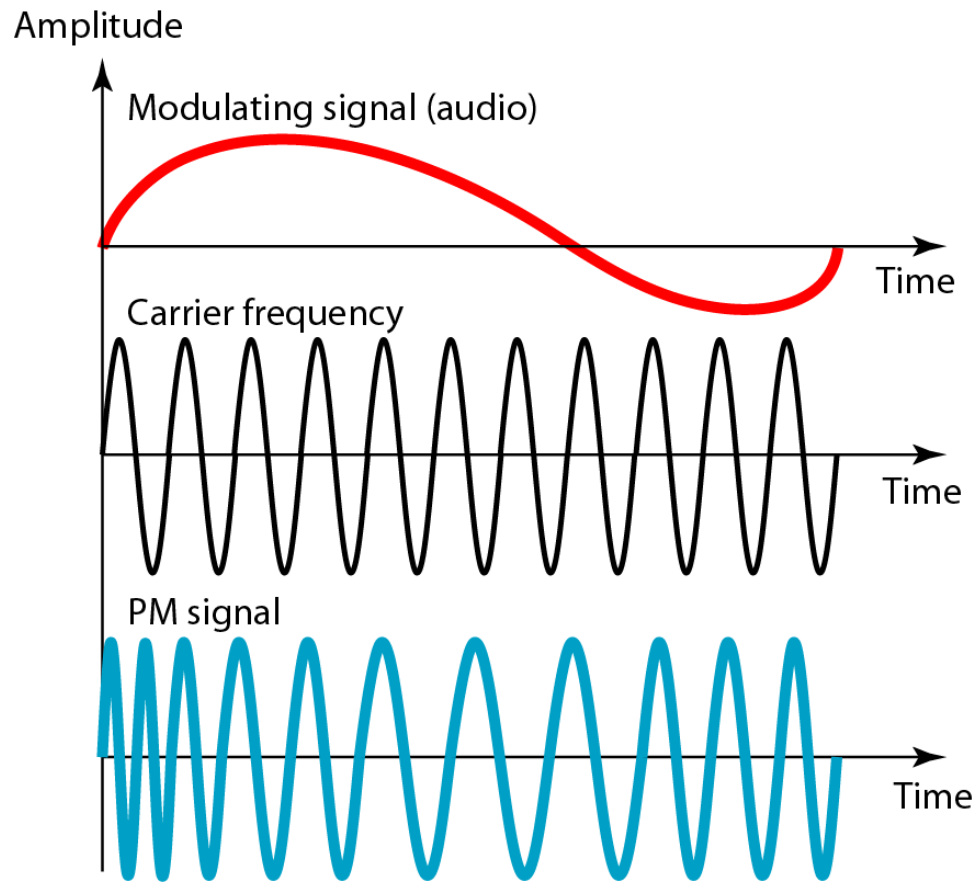
FM band allocation



Phase Modulation (PM)

- The modulating signal only changes the phase of the carrier signal.
- The phase change manifests itself as a frequency change but the instantaneous frequency change is proportional to the derivative of the amplitude.
- The bandwidth is higher than for AM.

Phase modulation





Note

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

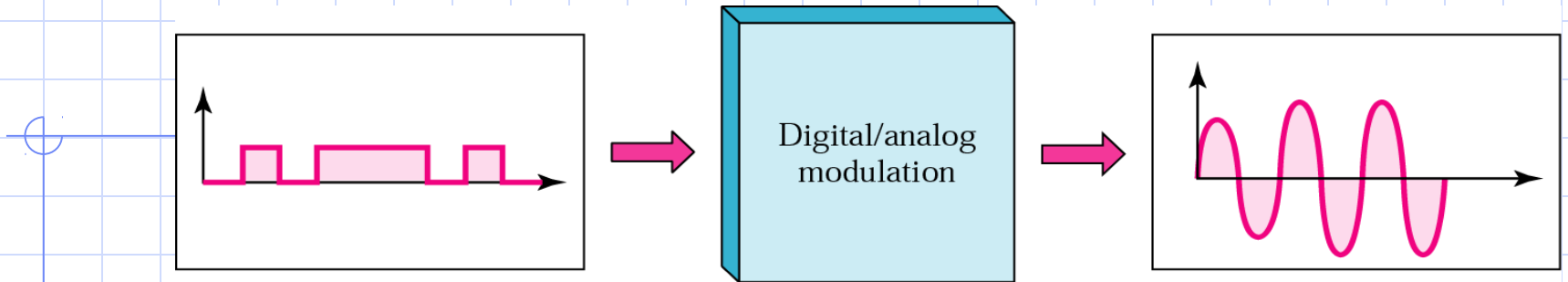
$$B_{PM} = 2(1 + \beta)B.$$

Where $\beta = 2$ most often.

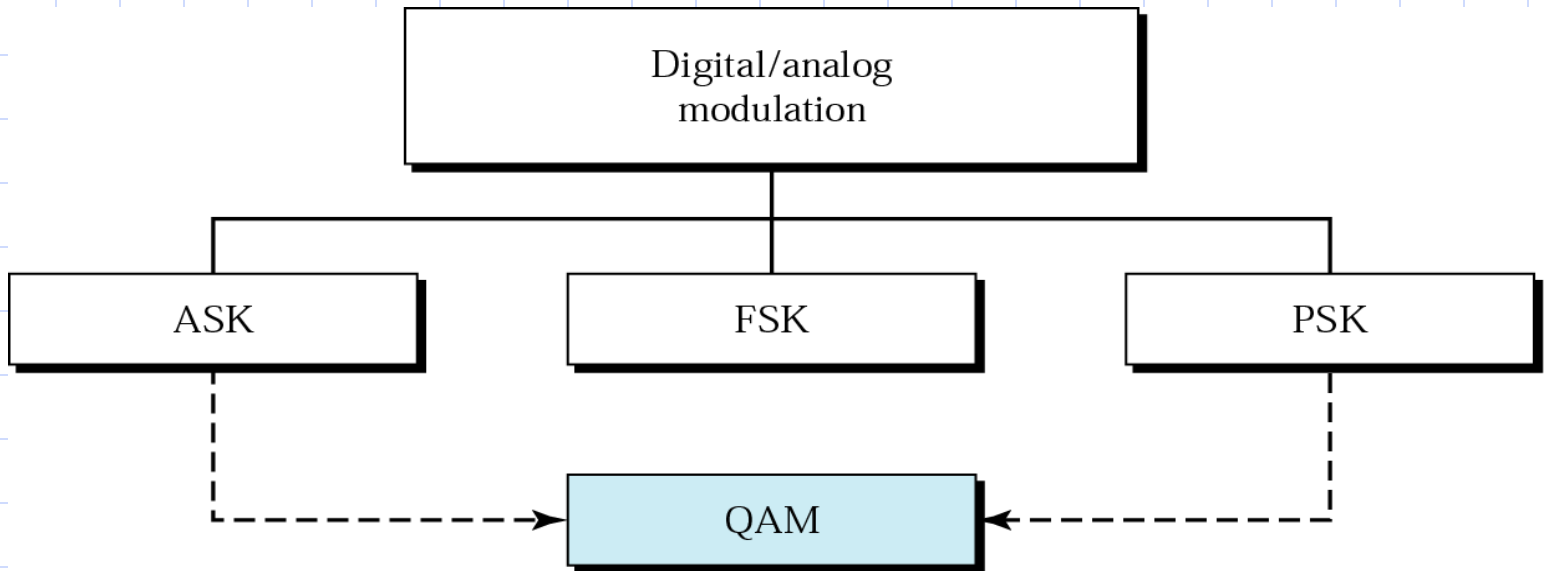
Modulation of Digital Data

1. Digital-to-Analog Conversion
2. Amplitude Shift Keying (ASK)
3. Frequency Shift Keying (FSK)
4. Phase Shift Keying (PSK)
5. Quadrature Amplitude Modulation (QAM)
6. Bit/Baud Comparison

Digital-to-analog modulation



Types of digital-to-analog modulation

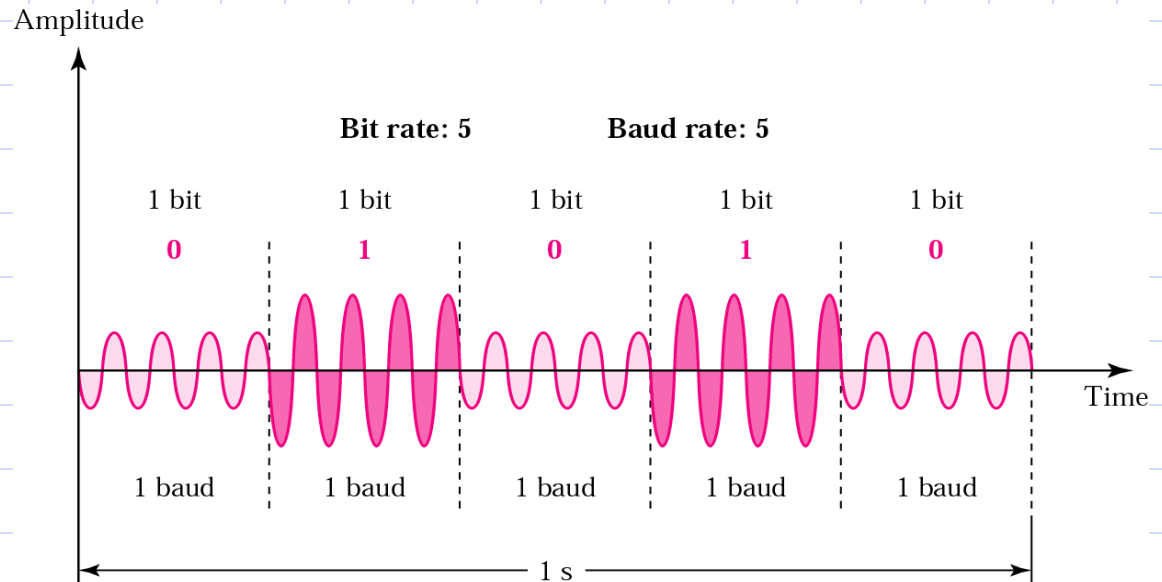


Aspects to digital-to Analog conversion

◆ Bit Rate / Baud Rate

- *Bit rate is the number of bits per second. Baud rate is the number of signal units per second. Baud rate is less than or equal to the bit rate.*
- *Bit rate is important in computer efficiency*
- *Baud rate is important in data transmission.*
 - ◆ *Baud rate determines the bandwidth required to send signal*
- *Baud rate = bit rate / # bits per signal unit*
- *An analog signal carries 4 bits in each signal unit. If 1000 signal units are sent per second, find the baud rate and the bit rate*
 - ◆ **Baud rate = 1000 bauds per second (baud/s) Bit rate = $1000 \times 4 = 4000$ bps**
- *The bit rate of a signal is 3000. If each signal unit carries 6 bits, what is the baud rate?*
 - ◆ **Baud rate = $3000/6 = 500$ bauds/sec**

Amplitude Shift Keying (ASK)

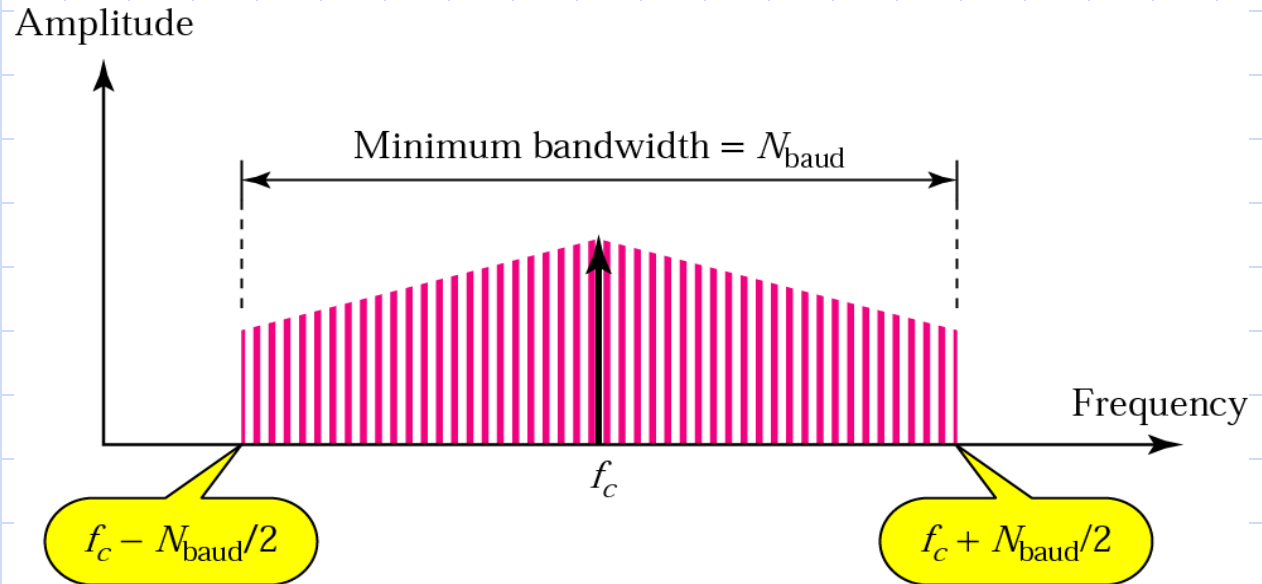


- ◆ The strength of the carrier signal is varied to represent binary 1 and 0.
- ◆ Frequency and phase remains the same.
- ◆ Highly susceptible to noise interference.
- ◆ Used up to 1200 bps on voice grade lines, and on optical fiber.

ASK

- ◆ The simplest and most common form of ASK operates as a switch, using the presence of a carrier wave to indicate a binary one and its absence to indicate a binary zero. This type of modulation is called on-off keying (OOK), and is used at radio frequencies to transmit Morse code (referred to as continuous wave operation)

Relationship between baud rate and bandwidth in ASK



◆ $BW = (1 + d) * N_d$

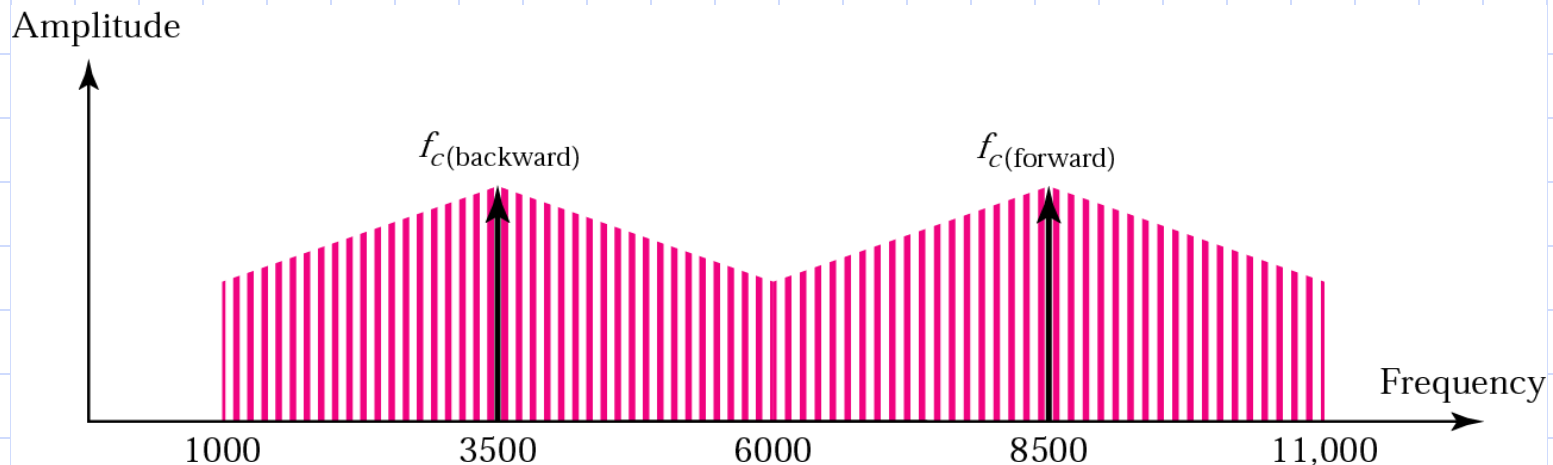
◆ Find the minimum bandwidth for an ASK signal transmitting at 2000 bps. The transmission mode is half-duplex.

- In ASK the baud rate and bit rate are the same. The baud rate is therefore 2000. An ASK signal requires a minimum bandwidth equal to its baud rate. Therefore, the minimum bandwidth is 2000 Hz.

Full duplex ASK

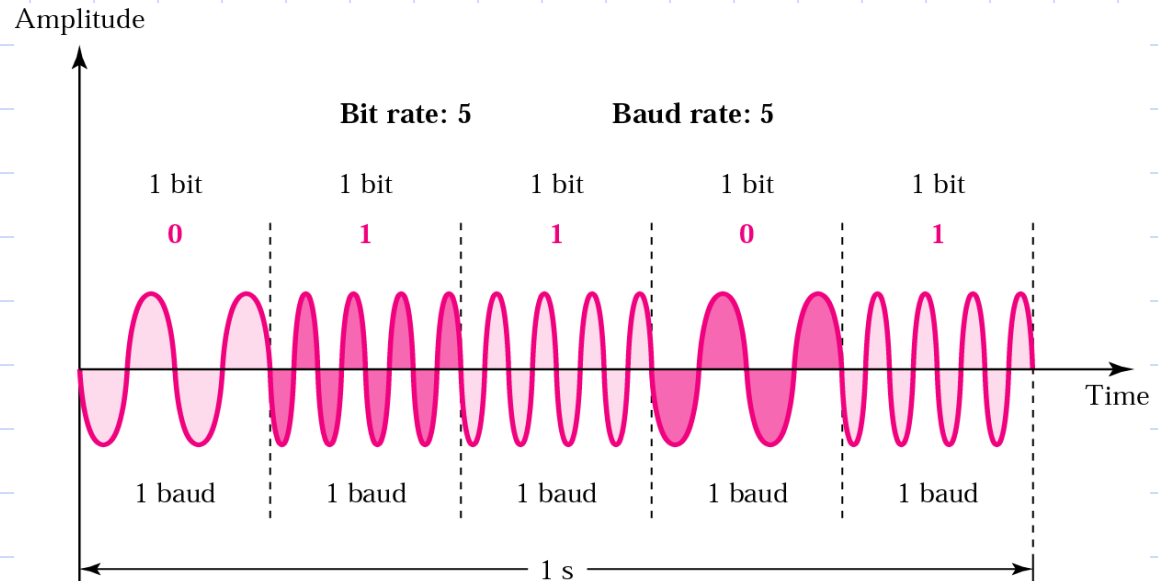
Given a bandwidth of 10,000 Hz (1000 to 11,000 Hz), if draw the full-duplex ASK diagram of the system. We can find the carriers and the bandwidths in each direction. Assume there is no gap between the bands in the two directions.

- For full-duplex ASK, the bandwidth for each direction is
 - $BW = 10000 / 2 = 5000$ Hz
 - The carrier frequencies can be chosen at the middle of each band
 - $f_c(\text{forward}) = 1000 + 5000/2 = 3500$ Hz
 - $f_c(\text{backward}) = 11000 - 5000/2 = 8500$ Hz

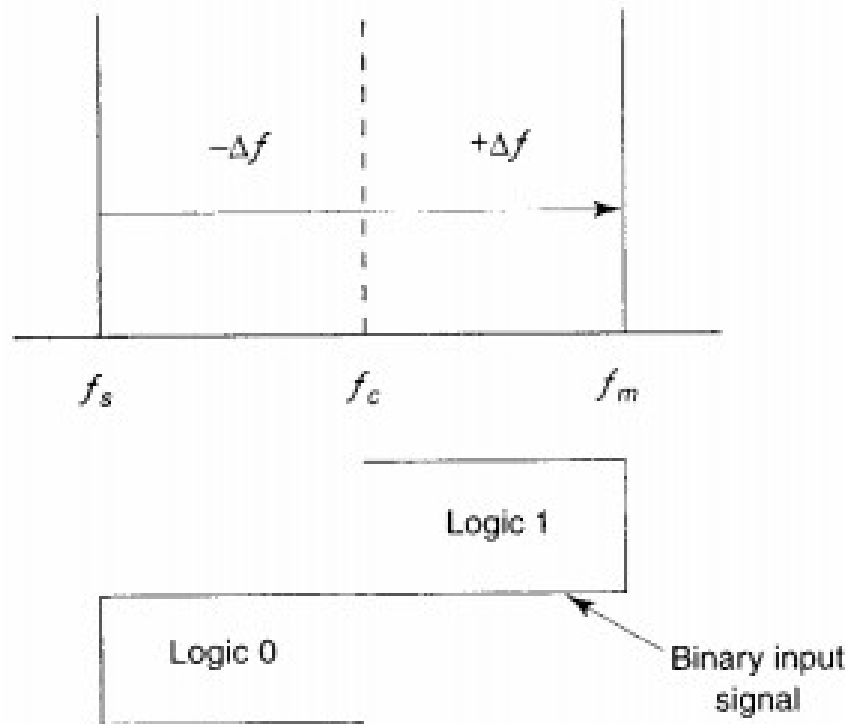


Frequency Shift Keying

- ◆ Frequency of the carrier is varied to represent digital data (binary 0/1)
- ◆ Peak amplitude and phase remain constant.
- ◆ Avoid noise interference by looking at frequencies (change of a signal) and ignoring amplitudes.
- ◆ Limitations of FSK is the physical capabilities of the carrier.
- ◆ f_1 and f_2 equally offset by equal opposite amounts to the carrier freq.
- ◆ In MFSK more than 2 freq are used, each signal element represents more than one bit



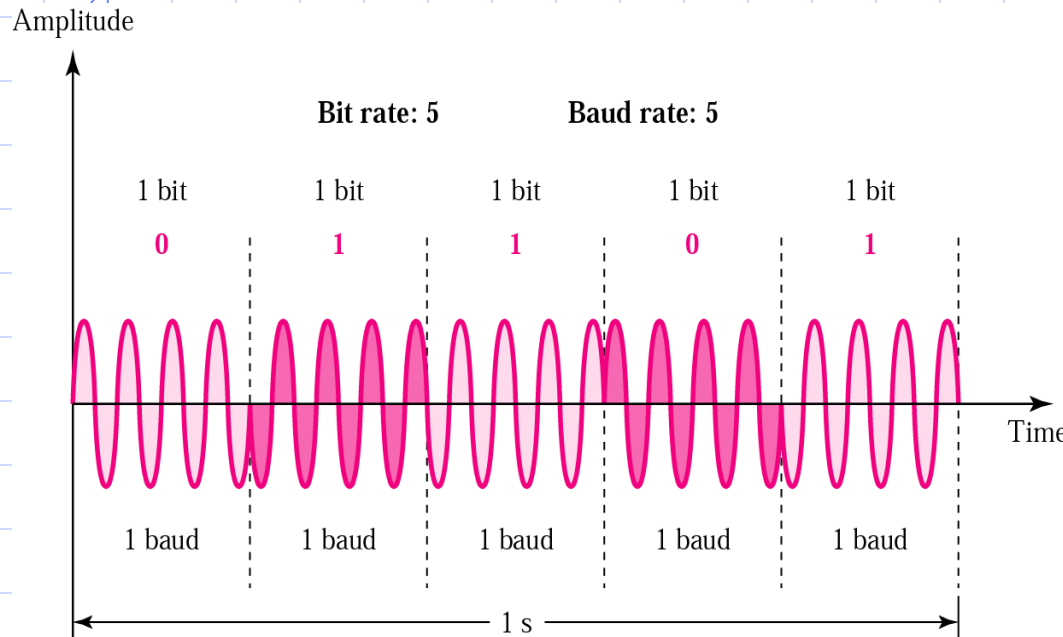
With binary FSK, the carrier center frequency (f_c) is shifted (deviated) up and down in the frequency domain by the binary input signal as shown in Figure 2-3.





As the binary input signal changes from a logic 0 to a logic 1 and vice versa, the output frequency shifts between two frequencies: a mark, or logic 1 frequency (f_m), and a space, or logic 0 frequency (f_s). The mark and space frequencies are separated from the carrier frequency by the peak frequency deviation (Δf) and from each other by $2 \Delta f$.

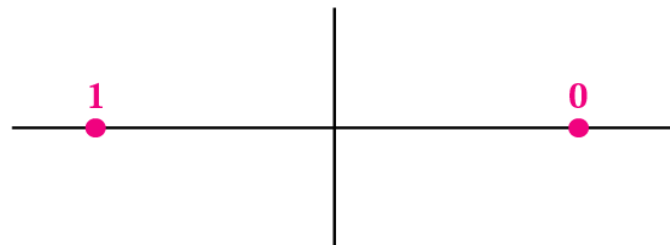
Phase Shift Keying



- ◆ Phase of the carrier is varied to represent digital data (binary 0 or 1)
- ◆ Amplitude and frequency remains constant.
- ◆ If phase 0 deg to represent 0, 180 deg to represent 1. (2-PSK)
- ◆ PSK is not susceptible to noise degradation that affects ASK or bandwidth limitations of FSK

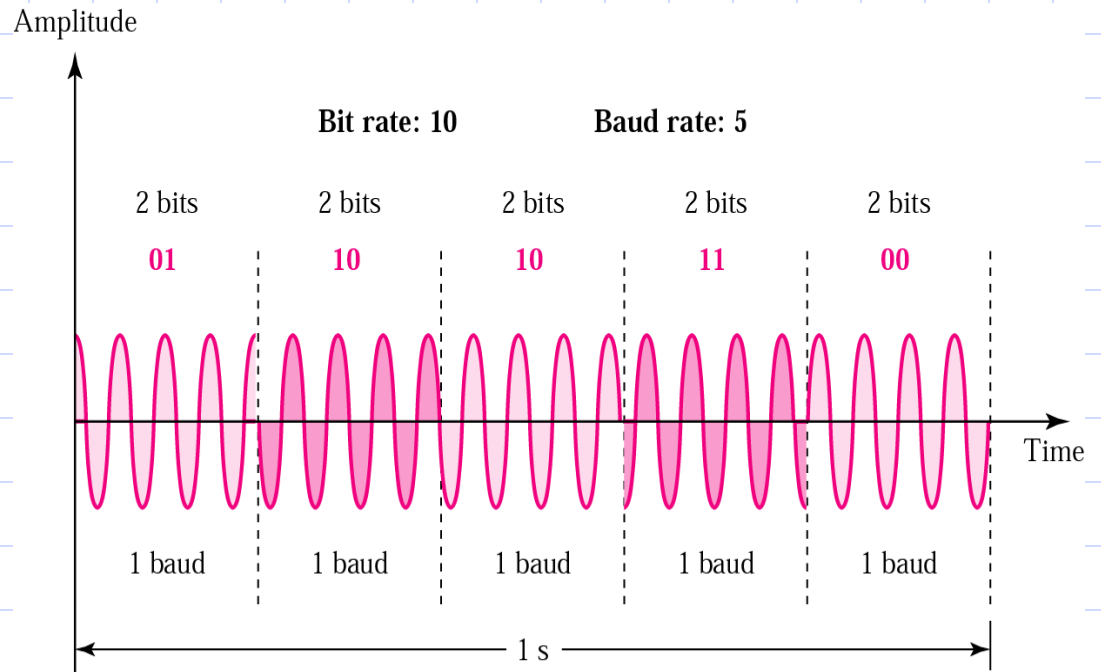
Bit	Phase
0	0
1	180

Bits



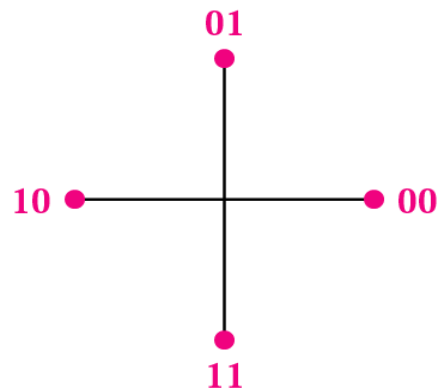
Constellation diagram

4-PSK (QPSK) method



Dibit	Phase
00	0
01	90
10	180
11	270

Dibit
(2 bits)



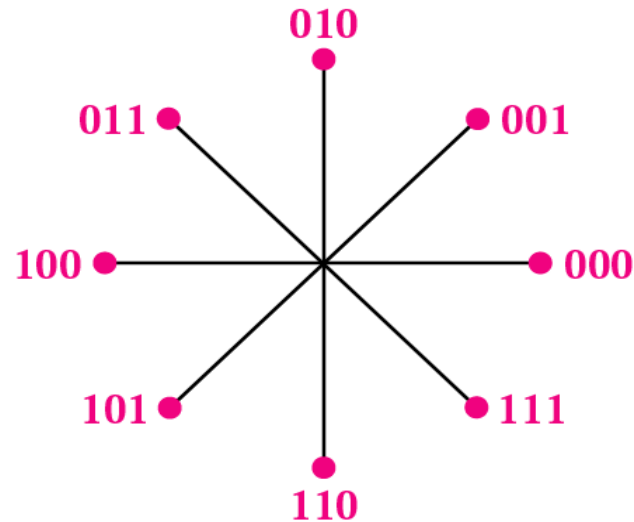
Constellation diagram

8-PSK

- ◆ We can extend, by varying the the signal by shifts of 45 deg (instead of 90 deg in 4-PSK)
- ◆ With $8 = 2^3$ different phases, each phase can represents 3 bits (tribit)

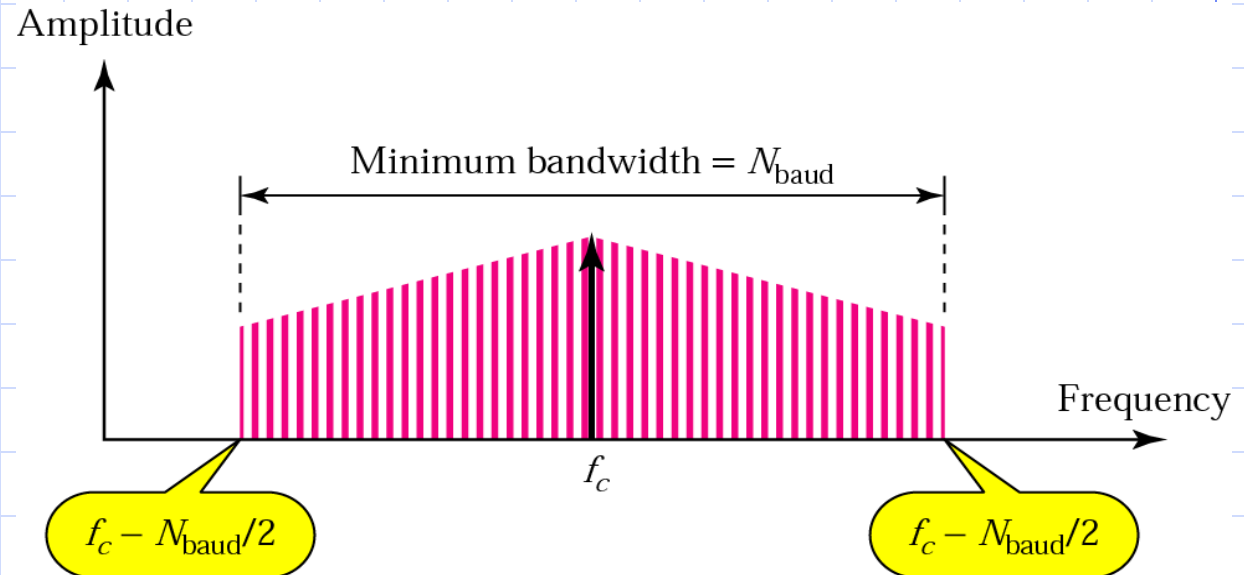
Tribit	Phase
000	0
001	45
010	90
011	135
100	180
101	225
110	270
111	315

Tribits
(3 bits)



Constellation diagram

Relationship between baud rate and bandwidth in PSK

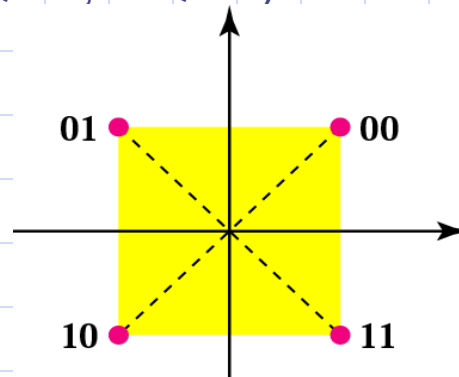


- ◆ Bandwidth similar to ASK, but data rate can 2 or more times greater.
- ◆ What is the bandwidth for a 4-PSK signal transmitting at 2000 bps. Transmission is in half-duplex mode.
 - For PSK the baud rate is the same as the bandwidth, which means the baud rate is 5000. But in 8-PSK the bit rate is 3 times the baud rate, so the bit rate is 15,000 bps.
- ◆ Given a bandwidth of 5000 Hz for an 8-PSK signal, what are the baud rate and bit rate?
 - For PSK the baud rate is the same as the bandwidth, which means the baud rate is 5000. But in 8-PSK the bit rate is 3 times the baud rate, so the bit rate is 15,000 bps.

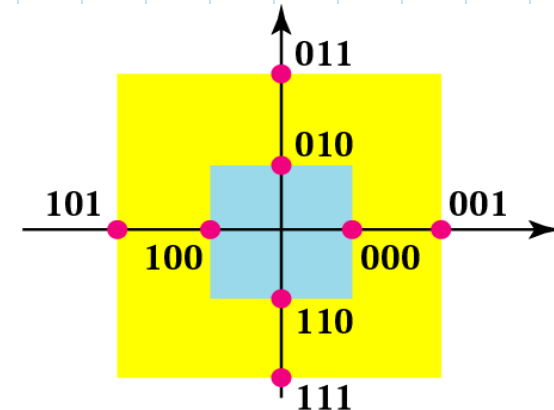
Quadrature Amplitude Modulation

- ◆ PSK is limited by the ability of the equipment to distinguish between small differences in phases.
 - Limits the potential data rate.
- ◆ Quadrature amplitude modulation is a combination of ASK and PSK so that a maximum contrast between each signal unit (bit, dibit, tritbit, and so on) is achieved.
 - We can have x variations in phase and y variations of amplitude
 - $x \cdot y$ possible variation (greater data rates)
- ◆ Numerous variations. (4-QAM, 8-QAM)

of phase shifts > # of amplitude shifts

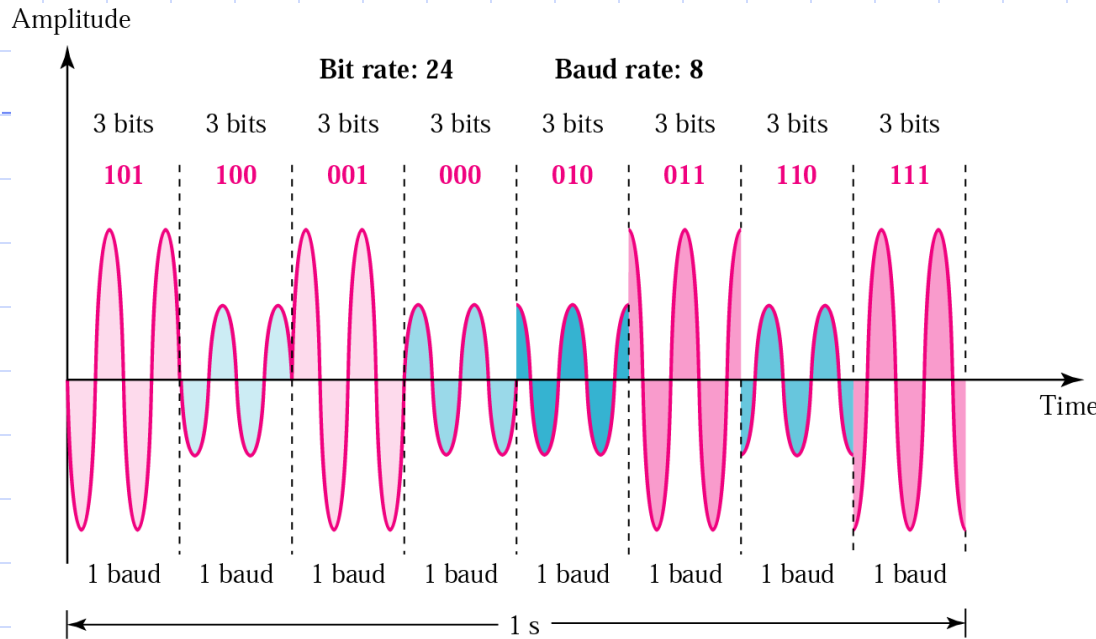


4-QAM
1 amplitude, 4 phases



8-QAM
2 amplitudes, 4 phases

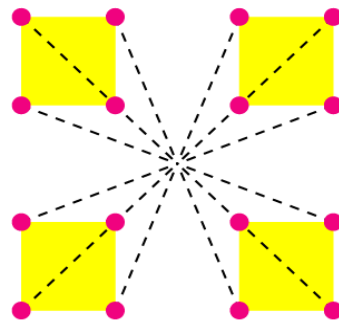
8-QAM and 16-QAM



Second example, recommendation of OSI. not all possibilities are used, to increase readability of signal, measurable differences between shifts are increased

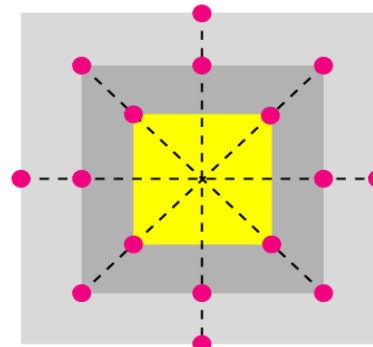
First example handles noise best Because of ratio of phases to amplitudes ITU-T recommendation.

3 amplitudes, 12 phases



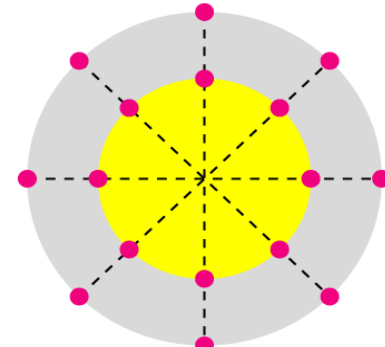
16-QAM

4 amplitudes, 8 phases



16-QAM

2 amplitudes, 8 phases



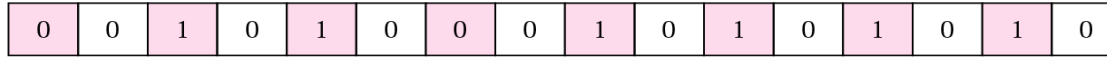
16-QAM

Bit Baud comparison

Bit

Baud rate = N

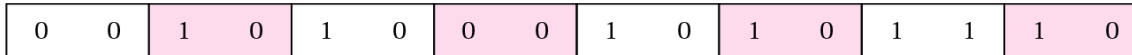
Bit rate = N



Dibit

Baud rate = N

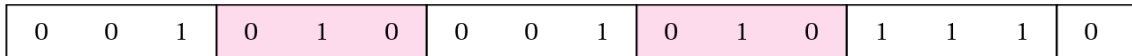
Bit rate = $2N$



Tribit

Baud rate = N

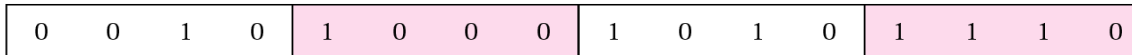
Bit rate = $3N$



Quadbit

Baud rate = N

Bit rate = $4N$



Assuming a FSK signal over voice-grade phone line can send 1200 bps, it requires 1200 signal units to send 1200 bits (each frequency shift represents one bit, baud rate 1200)

Assuming 8-QAM, baud rate is only 400 to achieve same data rate.

Modulation	Units	Bits/Baud	Baud rate	Bit Rate
ASK, FSK, 2-PSK	Bit	1	N	N
4-PSK, 4-QAM	Dibit	2	N	$2N$
8-PSK, 8-QAM	Tribit	3	N	$3N$
16-QAM	Quadbit	4	N	$4N$
32-QAM	Pentabit	5	N	$5N$
64-QAM	Hexabit	6	N	$6N$
128-QAM	Septabit	7	N	$7N$
256-QAM	Octabit	8	N	$8N$

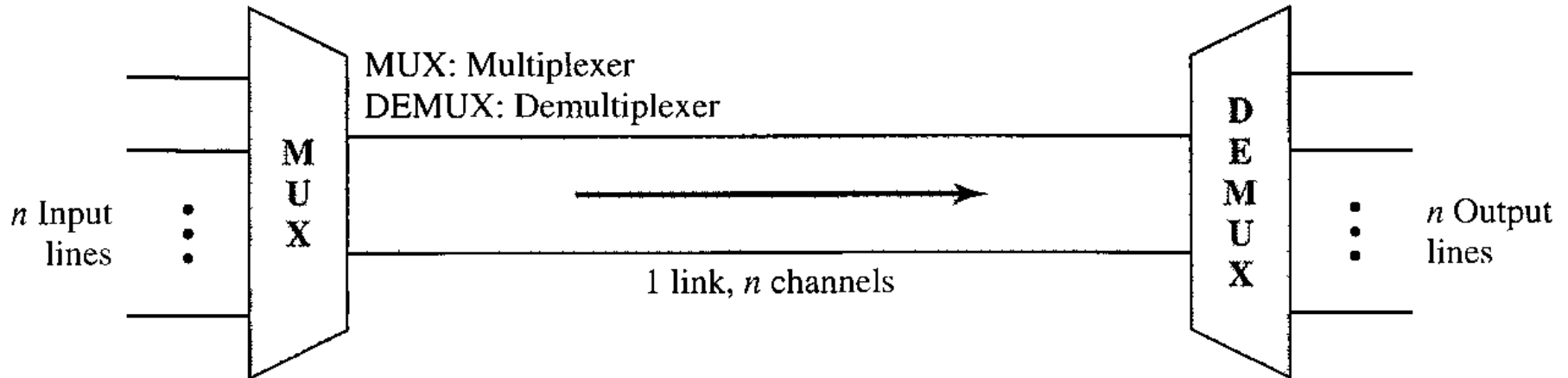
Multiplexing

- **Many to one/one to many**
- **Types of multiplexing**

Multiplexing

- It is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.
- Multiplexing is done using a device called Multiplexer (MUX) that combine n input lines to generate one output line i.e. (*many to one*).
- At the receiving end a device called Demultiplexer (DEMUX) is used that separate signal into its component signals i.e. one input and several outputs (*one to many*).

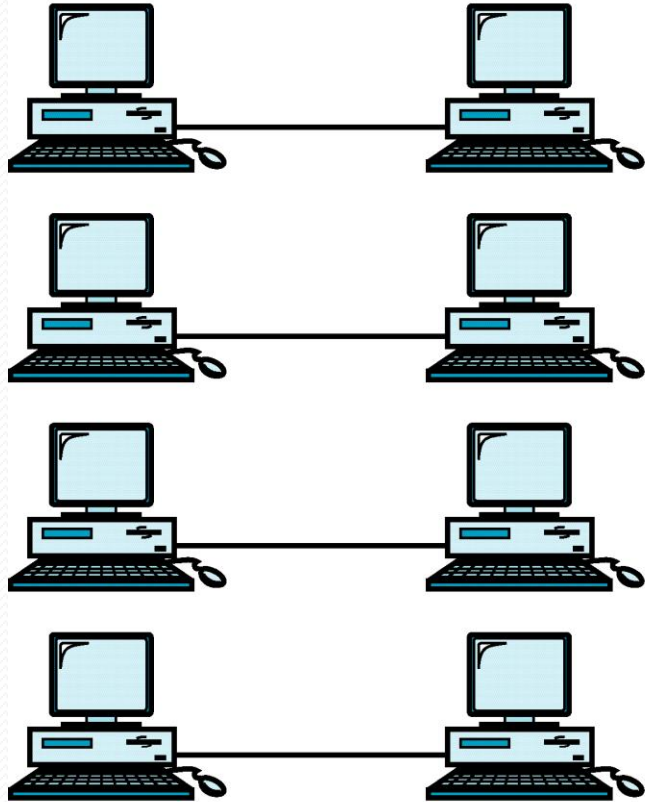
Multiplexing...



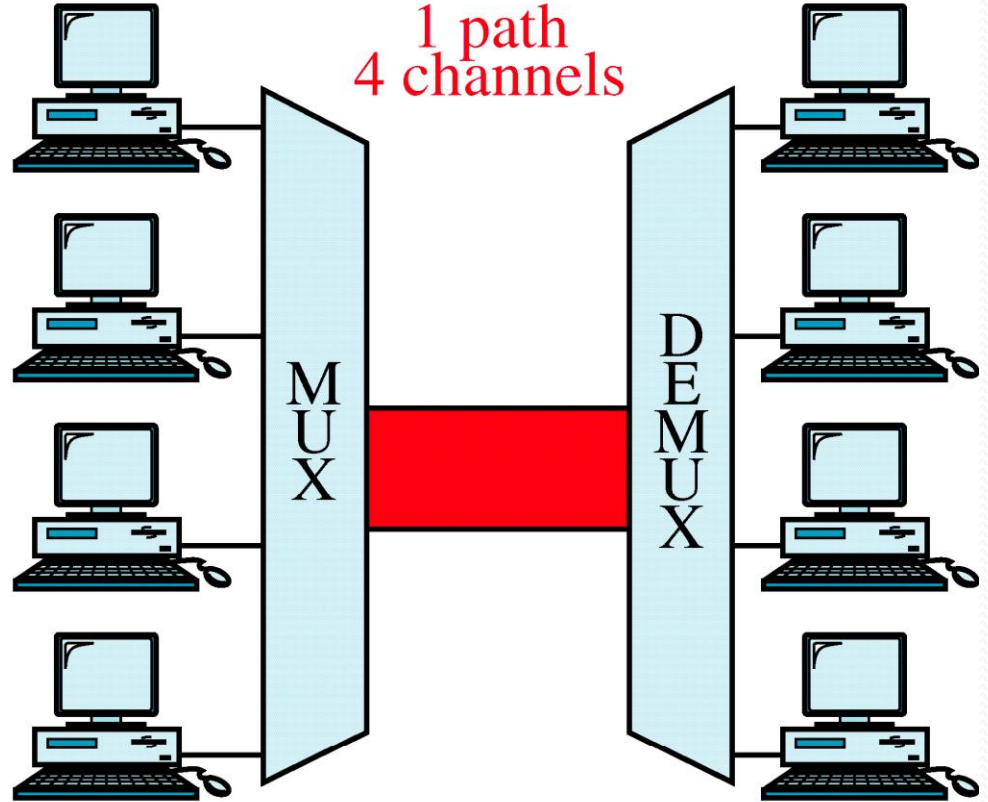
Advantages of Multiplexing

- More than one signals can be sent over single medium or link
- Effective use of the bandwidth of medium

Multiplexing vs. No Multiplexing

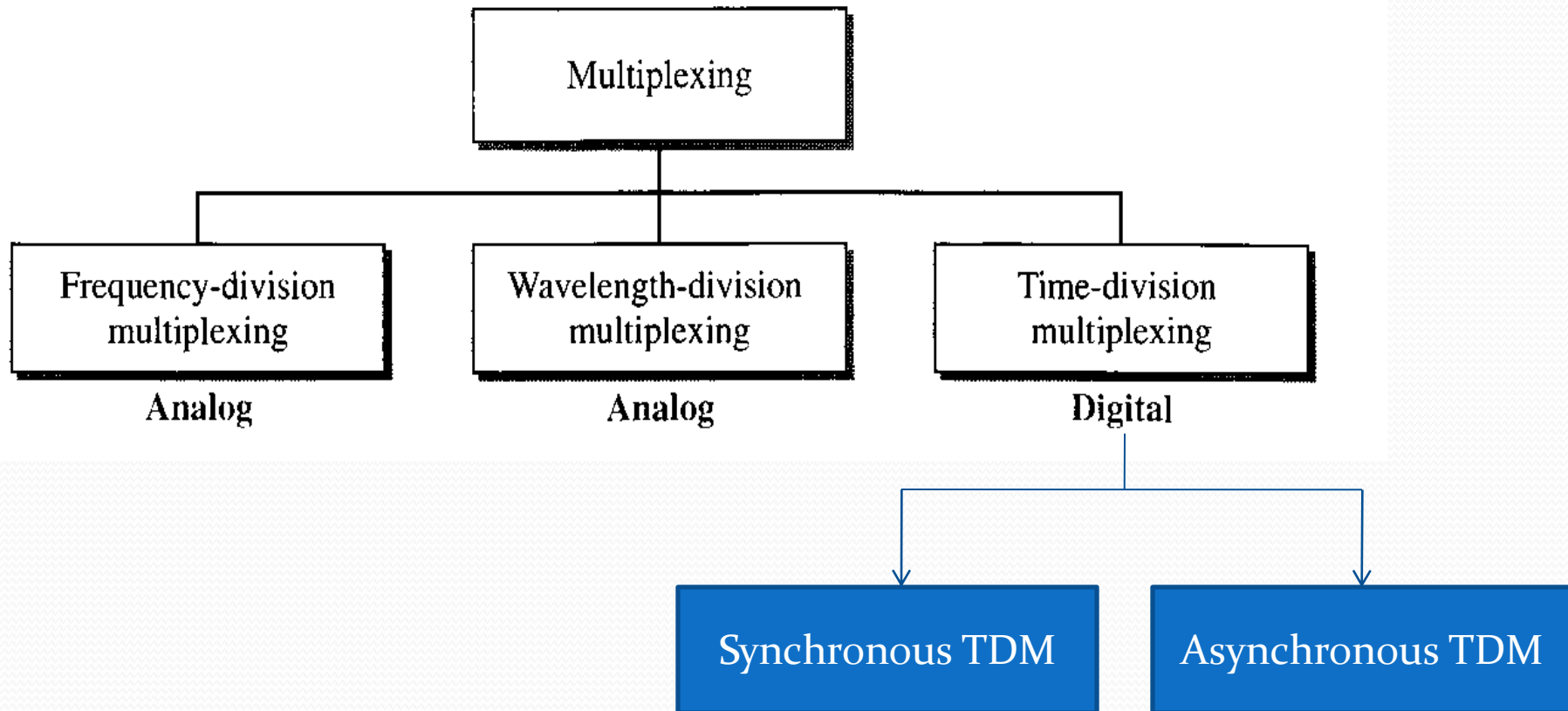


a. No multiplexing



b. Multiplexing

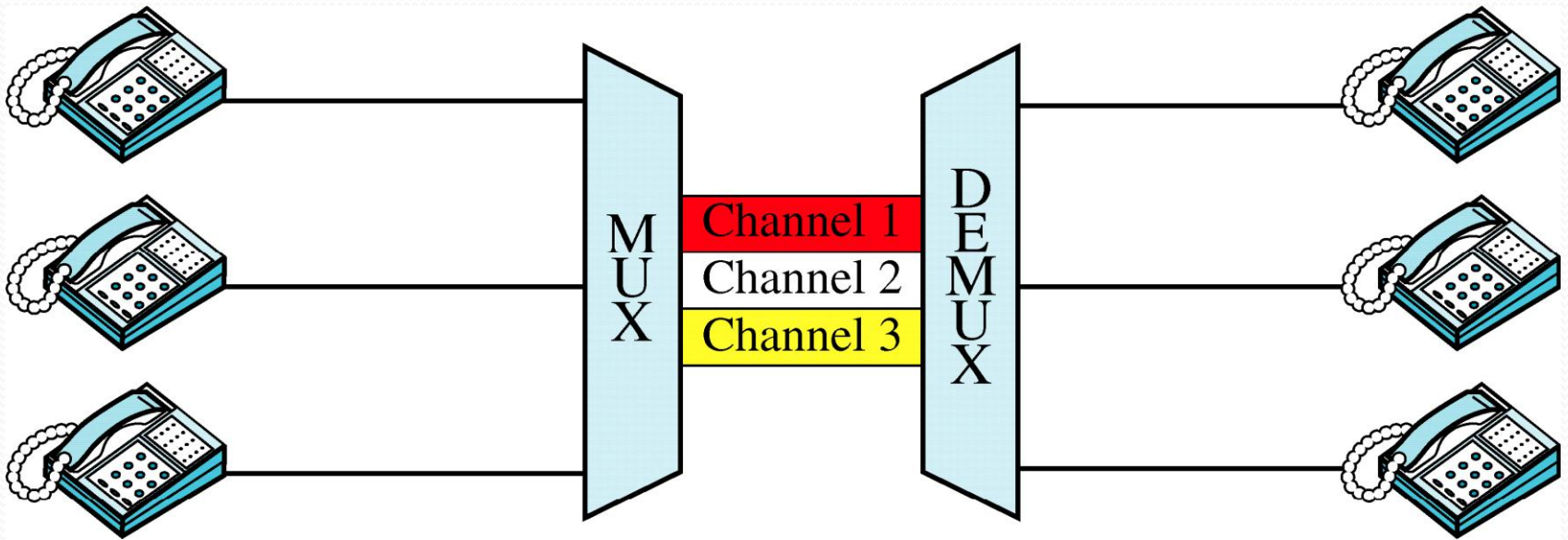
Types of Multiplexing



Frequency Division Multiplexing

- It is an analog technique.
- Signals of different frequencies are combined into a composite signal and is transmitted on the single link.
- Bandwidth of a link should be greater than the combined bandwidths of the various channels.
- Each signal is having different frequency.
- Channels are separated by the strips of unused bandwidth called *Guard Bands* (to prevent overlapping).

FDM

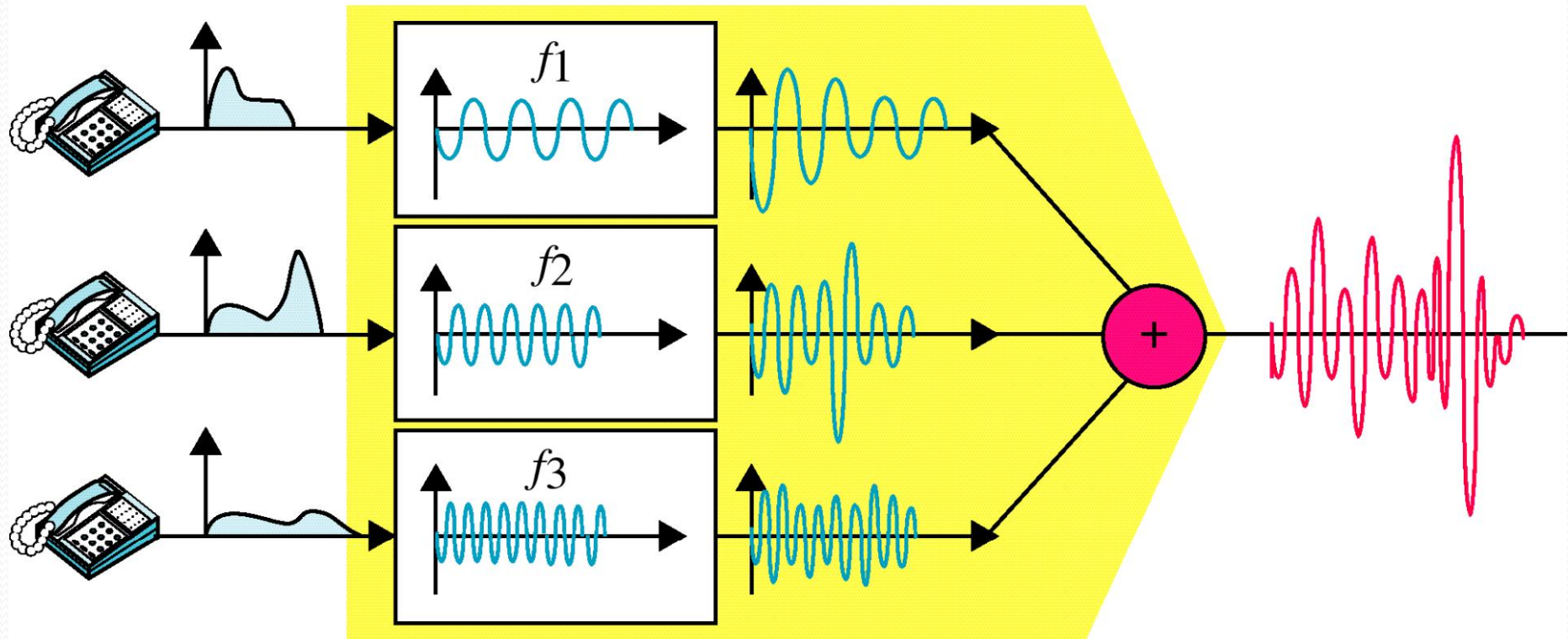


Applications of FDM

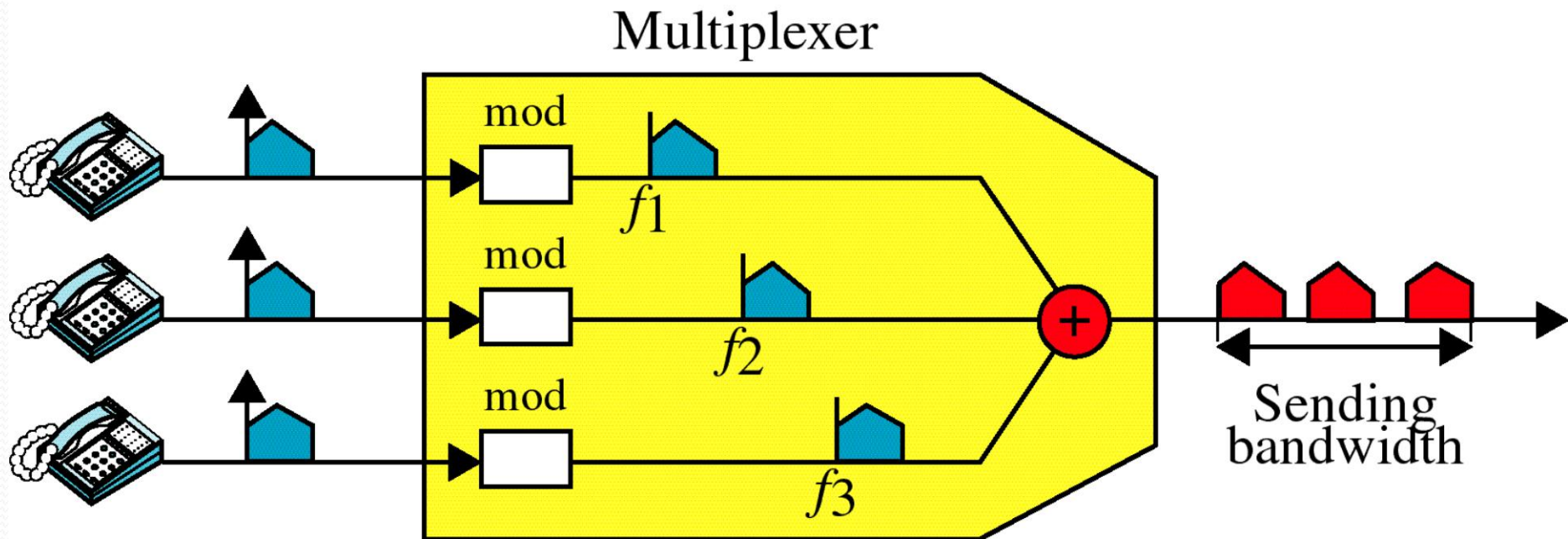
- FDM is used for FM & AM radio broadcasting.
- AM frequency = 530 to 1700 kHz.
- FM frequency = 88 to 108 MHz.
- FDM is used in television broadcasting.
- First generation cellular telephone also uses FDM.

FDM, Time Domain

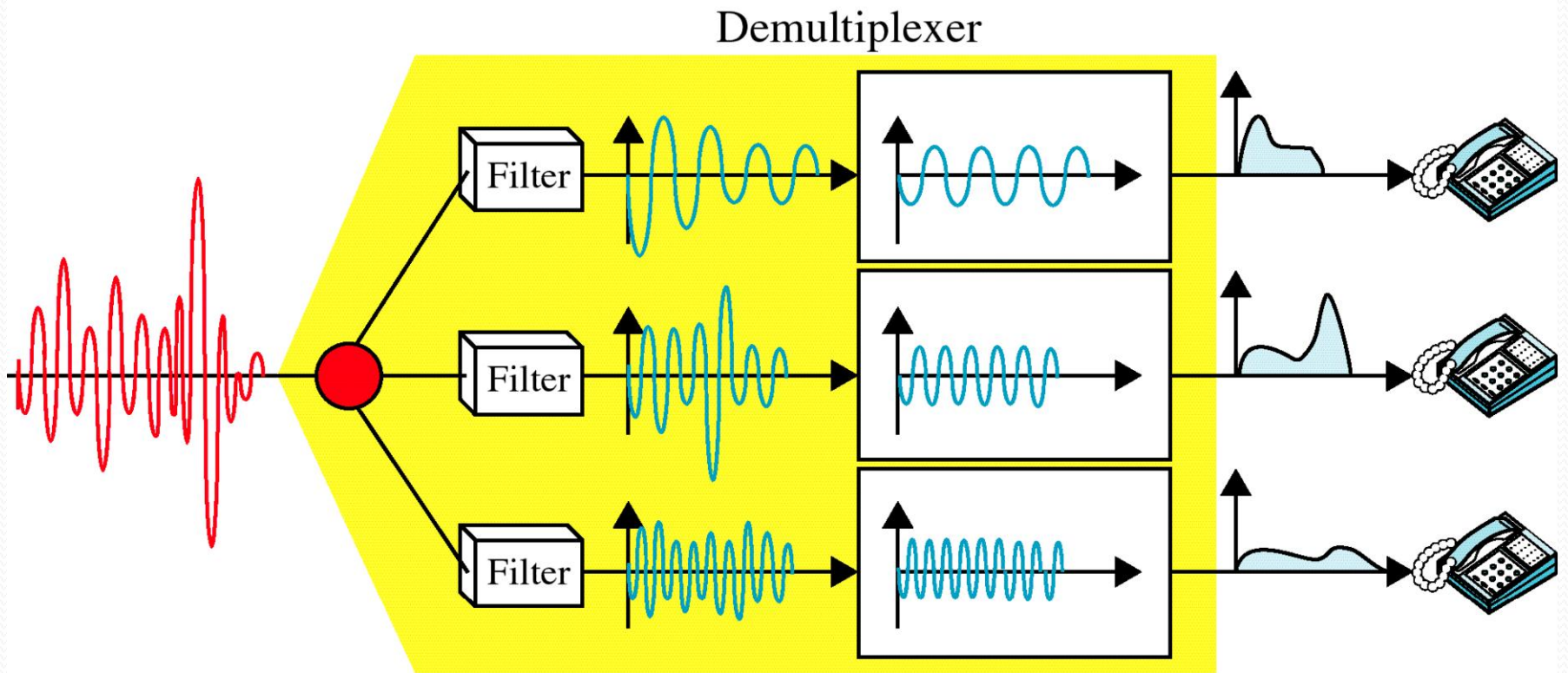
Multiplexer



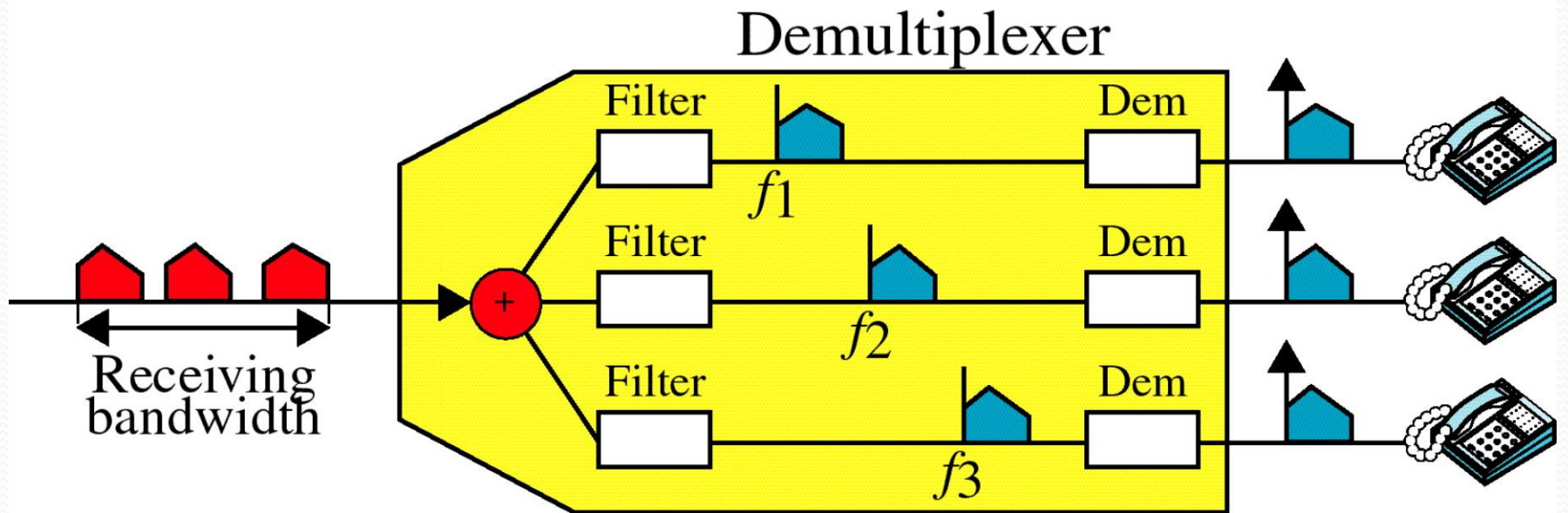
Multiplexing, Frequency Domain



Demultiplexing, Time Domain



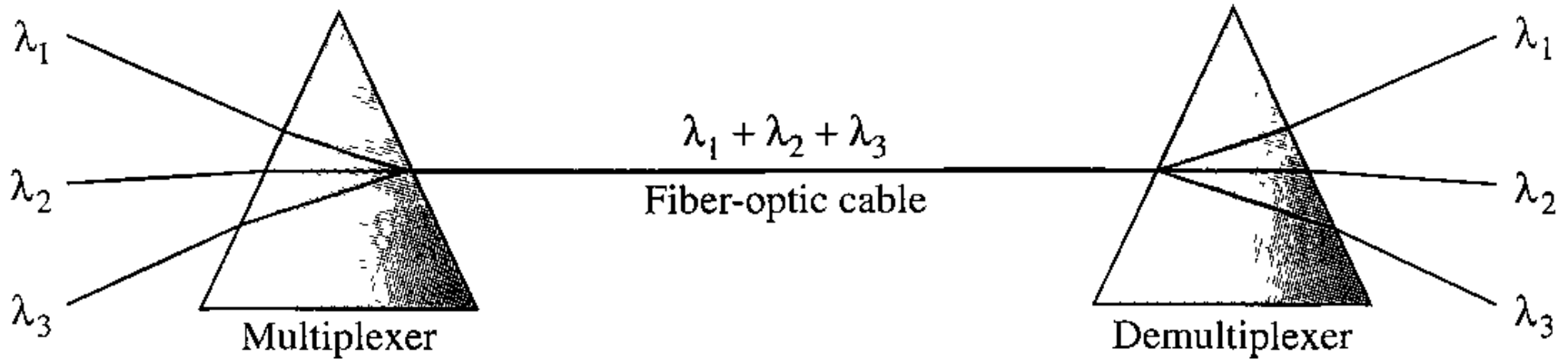
Demultiplexing, Frequency Domain



Wave Division Multiplexing

- WDM is an analog multiplexing technique.
- Working is same as FDM.
- In WDM different signals are *optical or light* signals that are transmitted through optical fiber.
- Various light waves from different sources are combined to form a composite light signal that is transmitted across the channel to the receiver.
- At the receiver side, this composite light signal is broken into different light waves by Demultiplexer.
- This Combining and the Splitting of light waves is done by using a PRISM. Prism bends beam of light based on the angle of incidence and the frequency of light wave.

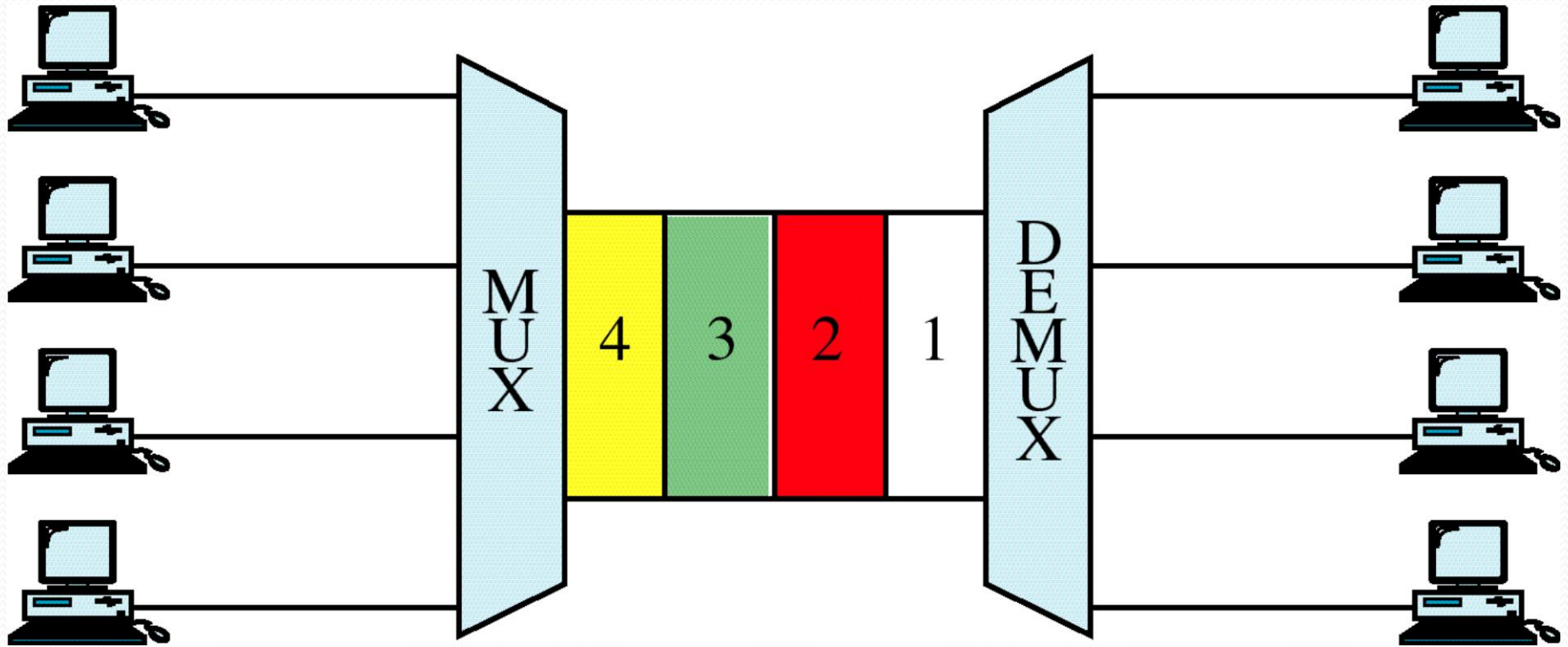
Wave Division Multiplexing...



Time Division Multiplexing

- It is the digital multiplexing technique.
- Channel/Link is not divided on the basis of *frequency* but on the *basis of time*.
- Total time available in the channel is divided between several users.
- Each user is allotted a particular time interval called *time slot* or *slice*.
- In TDM the data rate capacity of the transmission medium should be greater than the data rate required by sending or receiving devices.

TDM



Types of TDM

- Synchronous TDM
- Asynchronous TDM

Synchronous TDM

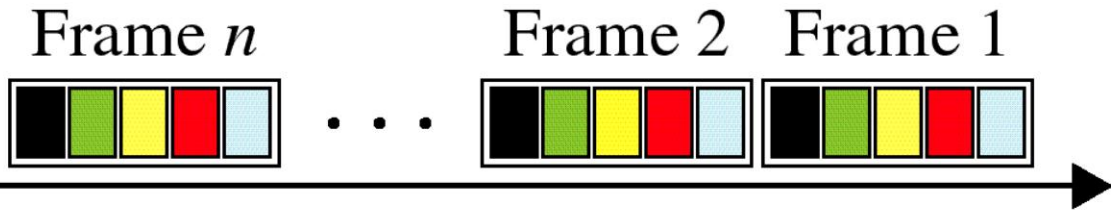
- Each device is given same Time Slot to transmit the data over the link, whether the device has any data to transmit or not.
- Each device places its data onto the link when its *Time Slot* arrives, each device is given the possession of line turn by turn.
- If any device does not have data to send then its time slot remains empty.
- Time slots are organized into *Frames* and each frame consists of one or more time slots.
- If there are n sending devices there will be n slots in frame.

Synchronous TDM

5 Inputs



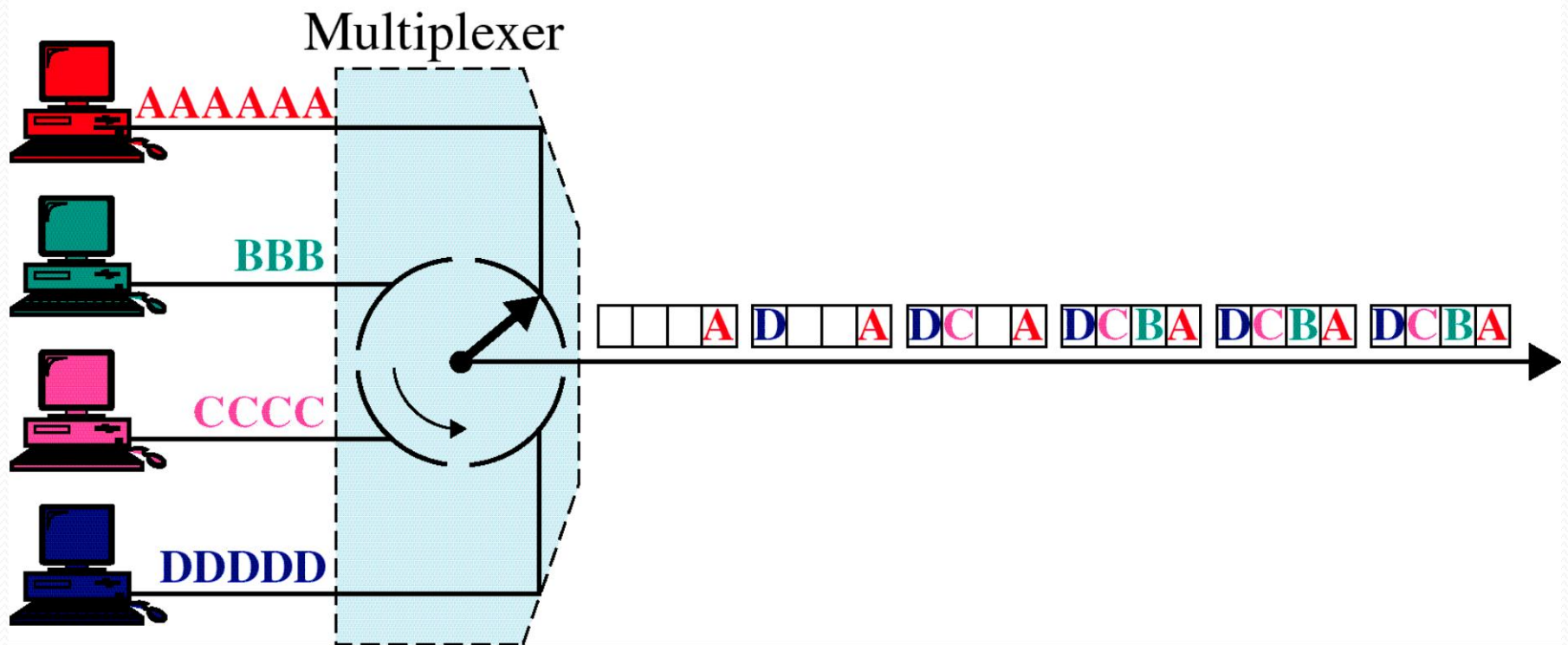
Number of inputs: 5
Number of slots in each frame: 5



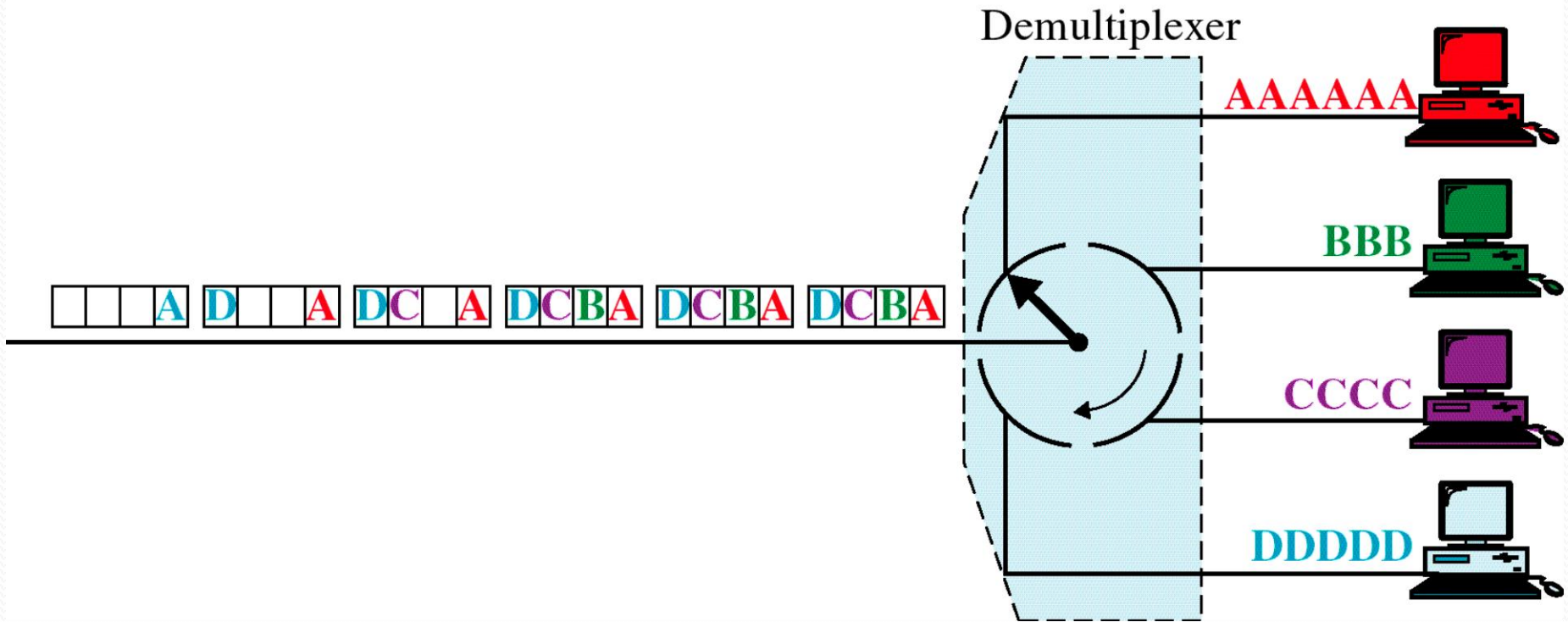
Multiplexing Process in STDM

- In STDM every device is given opportunity to transmit a specific amount of data onto the link.
- Each device gets its turn in fixed order and for fixed amount of time = INTERLEAVING.
- Interleaving is done by a character (one byte).
- Each frame consist of four slots as there are four input devices.
- Slots of some devices go empty if they do not have any data to send.

TDM, Multiplexing



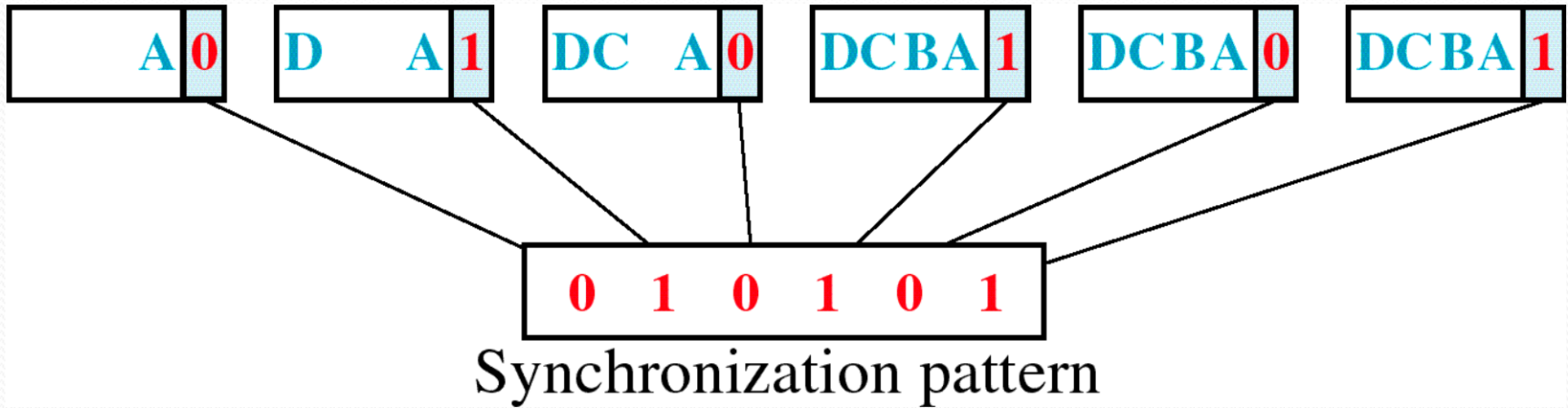
TDM, Demultiplexing



Disadvantages of STDM

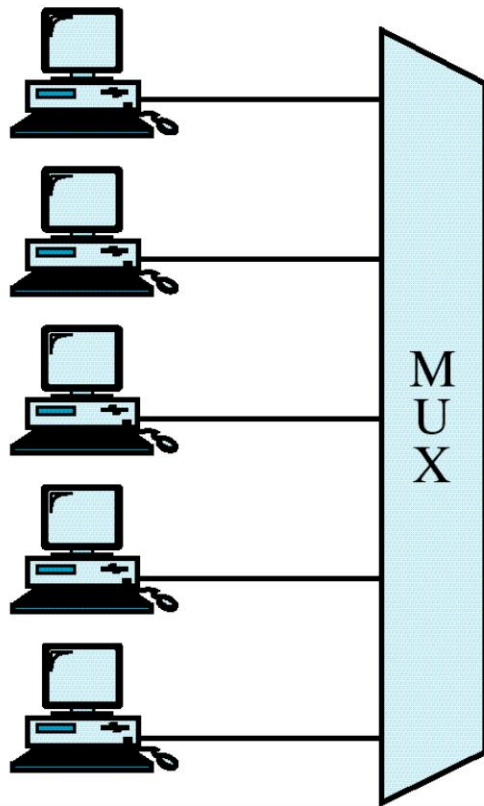
- The channel capacity cannot be fully utilized. Some of the slots go empty in certain frames.

Framing Bits

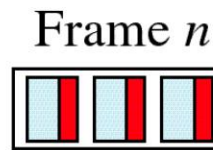


Asynchronous TDM

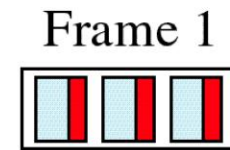
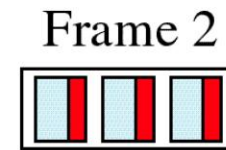
5 Inputs



Number of inputs: 5
Number of slots in each frame: 3



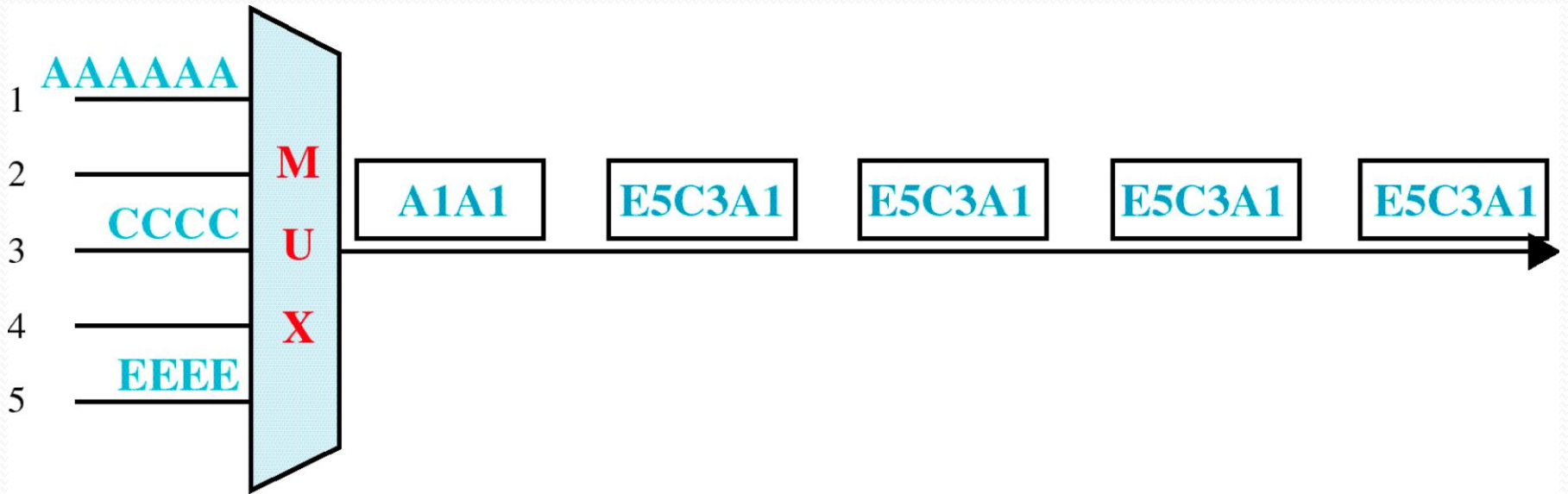
...



Asynchronous TDM

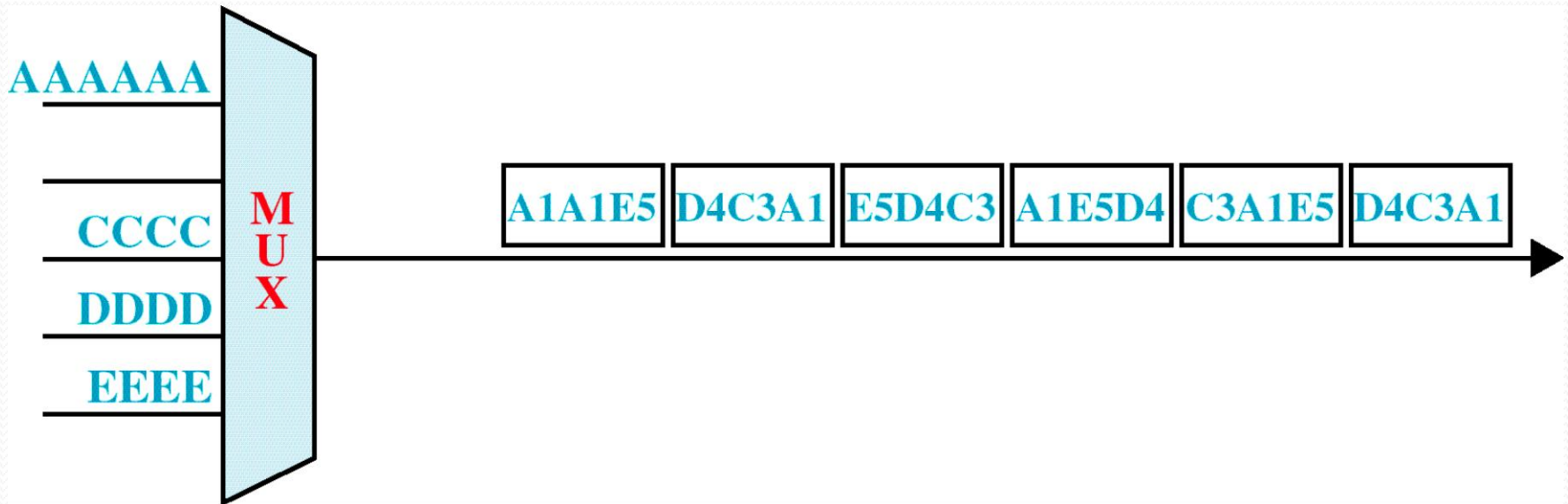
- Also known as Statistical Time Division *multiplexing*.
- In this time slots are not *Fixed* i.e. slots are Flexible.
- Total speed of the input lines can be greater than the capacity of the path.
- In ASTDM we have n input lines and m slots i.e. m less than n ($m < n$).
- Slots are not predefined rather slots are allocated to any of the device that has data to send.

Frames and Addresses



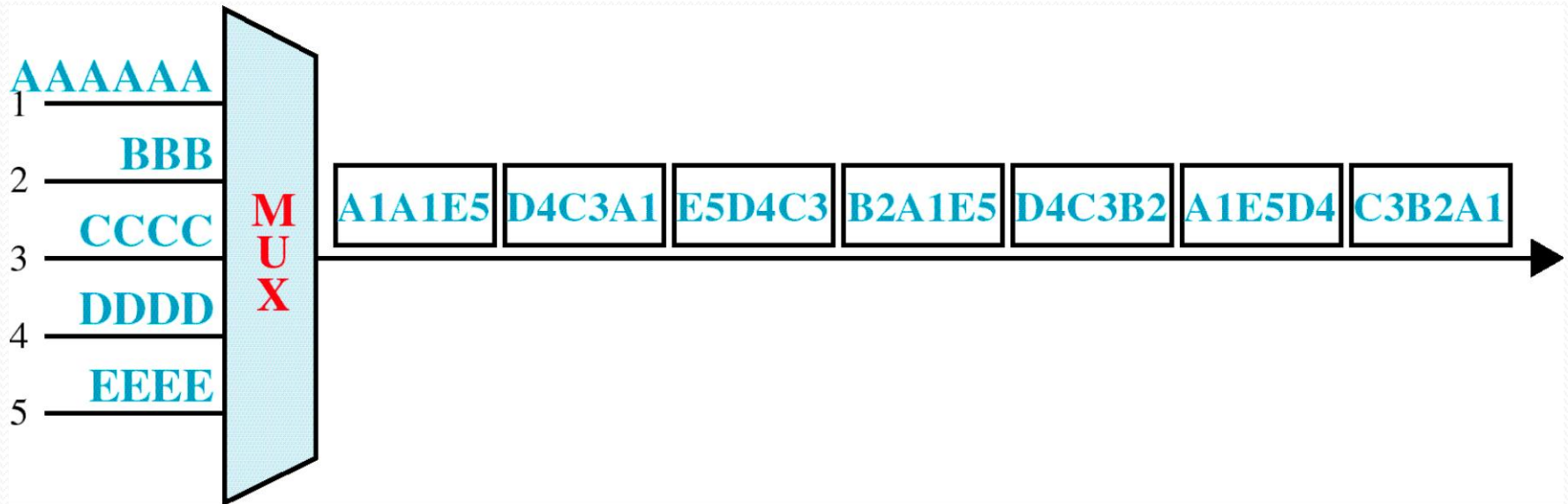
a. Only three lines sending data

Frames and Addresses



b. Only four lines sending data

Frames and Addresses



c. All five lines sending data