Unit 1: Introduction to computer network

Content

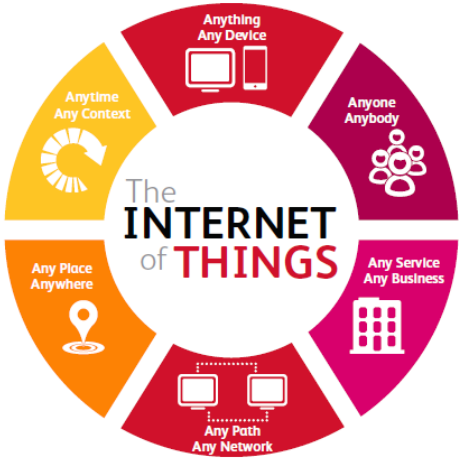
1.1 Communicating in a network centric world
1.2 Network as a platform
1.3 Architecture of internet
1.4 Classification of networks
1.5 Components of a network
1.6 Network topology and transmission
1.7 Layered models

Communicating in a network centric world

- Networks in Our Daily Lives
- Technology Then and Now
- The Global Community
- Networks Support the Way We Learn
- Networks Support the Way We Communicate
- Networks Support the Way We Work
- Networks Support the Way We Play

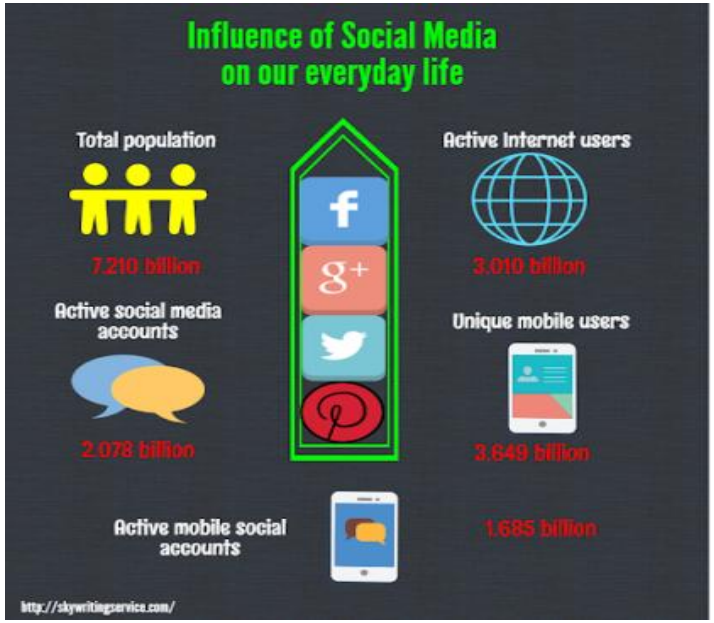
Networks in Our Daily Lives

- Communication is almost as important to us as our reliance on air, water, food, and shelter.
- The methods that we use to communicate are constantly changing and evolving. Whereas we were once limited to face-to-face interactions, breakthroughs in technology have significantly extended the reach of our communications.



Networks in Our Daily Lives

- Networks connect people and promote unregulated communication.
- The Internet is the largest network in existence. In fact, the term *Internet* means a network of networks. It is actually a collection of interconnected private and public networks.



Technology Then and Now

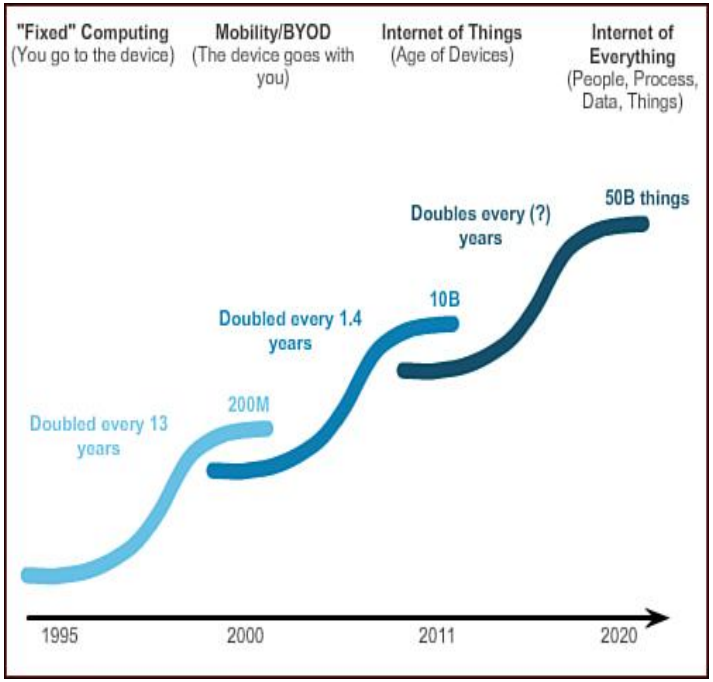
- Imagine a world without the Internet.
  - No more Google, YouTube, instant messaging, Facebook, Wikipedia, online gaming, Netflix, shopping online, or quickly looking up phone numbers and map directions to various locations at the click of a finger.
  - How different would our lives be without all of this?
  - Time would go back to the 80's!

Technology Then and Now

- A whole new world with the Internet!
  - Innovators are figuring out new ways to use the Internet more every day.
  - Consider the changes that have happened within the last couple of decades.

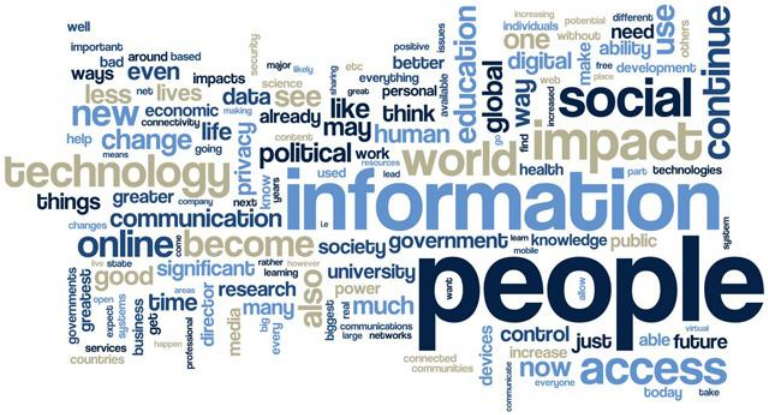
Technology Then and Now

- A whole new world with the Internet!
  - Now consider what changes will happen within the next decade.
  - What else do you think we will be able to do using the network as the platform?



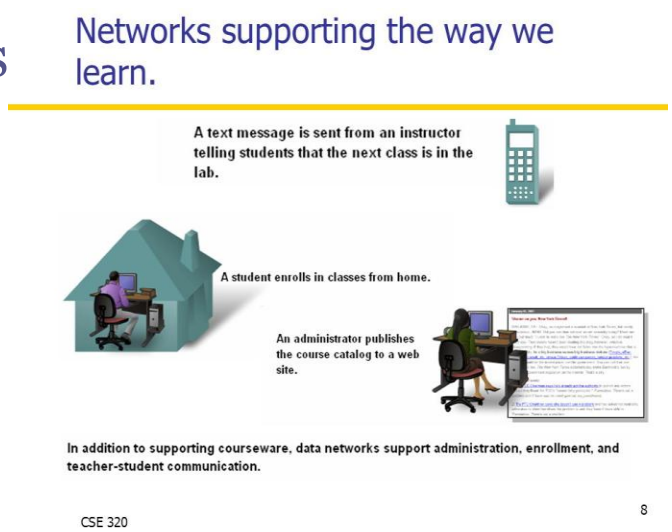
The Global Community

- The human network
  - The internet has changed the manner in which social, commercial, political, and personal interactions occur.
  - The Internet encourages the creation of global communities.
  - Global communities allow for social interaction that is independent of location or time zone.
  - We refer to this as the *human network*.



Networks Support the Way We Learn

- Changing the way we learn.
  - Robust and reliable networks support and enrich student learning experiences. They deliver learning material in a wide range of formats, including interactive activities, assessments, and feedback. Networks now
    - Support the creation of virtual classrooms
    - Provide on-demand video
    - Enable collaborative learning spaces
    - Enable mobile learning



Networks Support the Way We Communicate

- Changing the Way We Communicate
  - The globalization of the Internet has ushered in new forms of communication that empower individuals to create information that can be accessed by a global audience.
  - Some forms of communications include:
    - Instant messaging (IM) and texting
    - Social media
    - Collaboration tools
    - Weblogs (blogs)
    - Wikis
    - Podcasting
    - Peer-to-peer (P2P) file sharing

Networks Support the Way We Work

- Changing the Way We Work
  - Data networks were initially used by business world.
  - These business networks evolved to enable the transmission of many different types of information services, including email, video, messaging, and telephony.

Networks Support the Way We play

- Changing the Way We Play
  - The widespread adoption of the Internet by travel industries enhances the ability to enjoy and share many forms of recreation.
  - In addition, the Internet is used for traditional forms of entertainment. Live sporting events and concerts can be experienced as they are happening, or recorded and viewed on demand.
  - Networks enable the creation of new forms of entertainment, such as online games.

Network as a platform

- The Converging Network:
  - The network has become a platform for distributing a wide range of services to end users in a reliable, efficient, and secure manner.
  - Advances in technology are enabling us to consolidate these different kinds of networks onto one platform, referred to as the ***converged network***.
  - Unlike dedicated networks, converged networks are capable of delivering voice, video streams, text, and graphics among many different types of devices over the same communication channel and network structure.

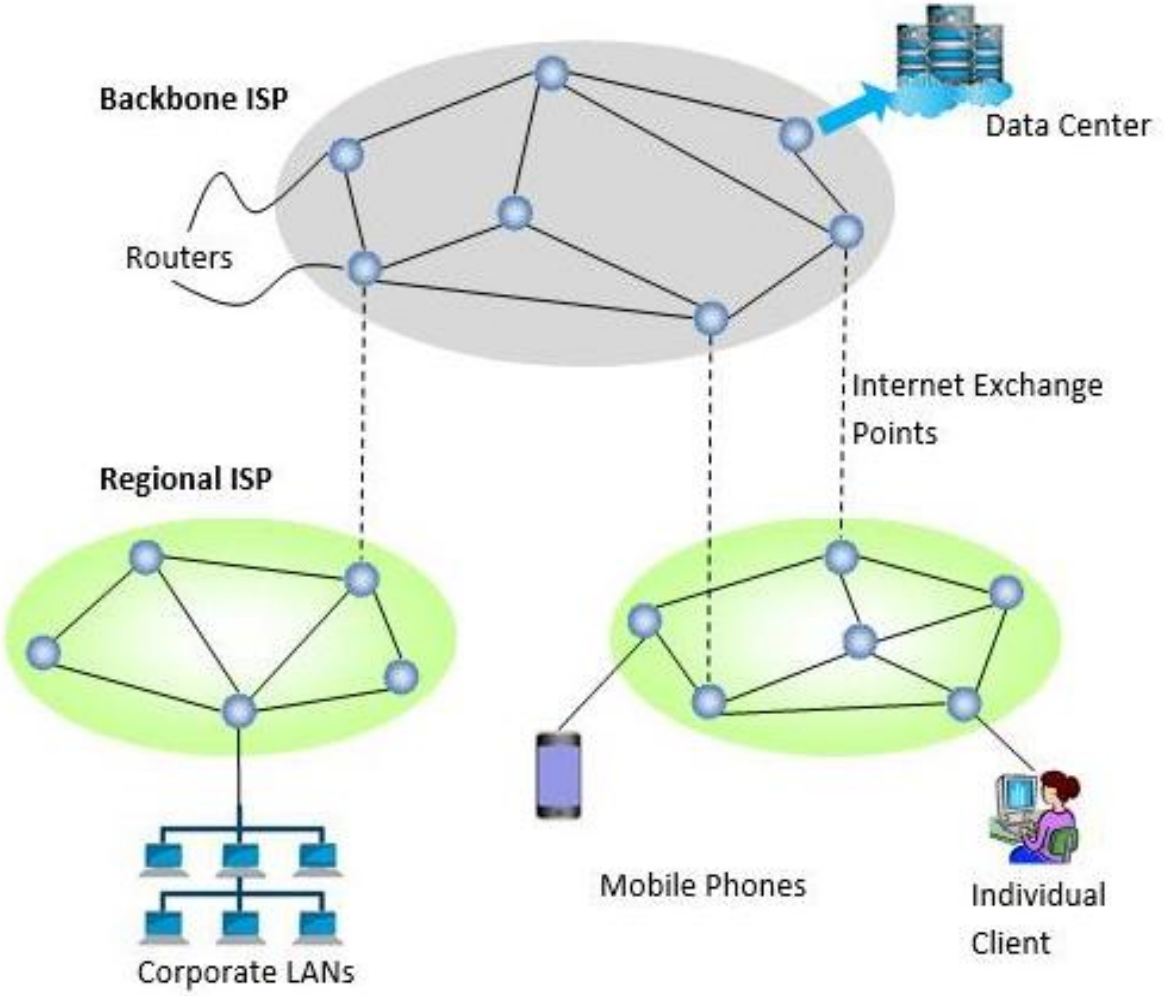
Network as a platform

- Previously separate and distinct communication forms have converged onto a common platform.
- In a converged network, there are still many points of contact and many specialized devices, such as personal computers, phones, TVs, and tablet computers, but there is one common network infrastructure.
- This network infrastructure uses the same set of rules, agreements, and implementation standards.

Architecture of internet

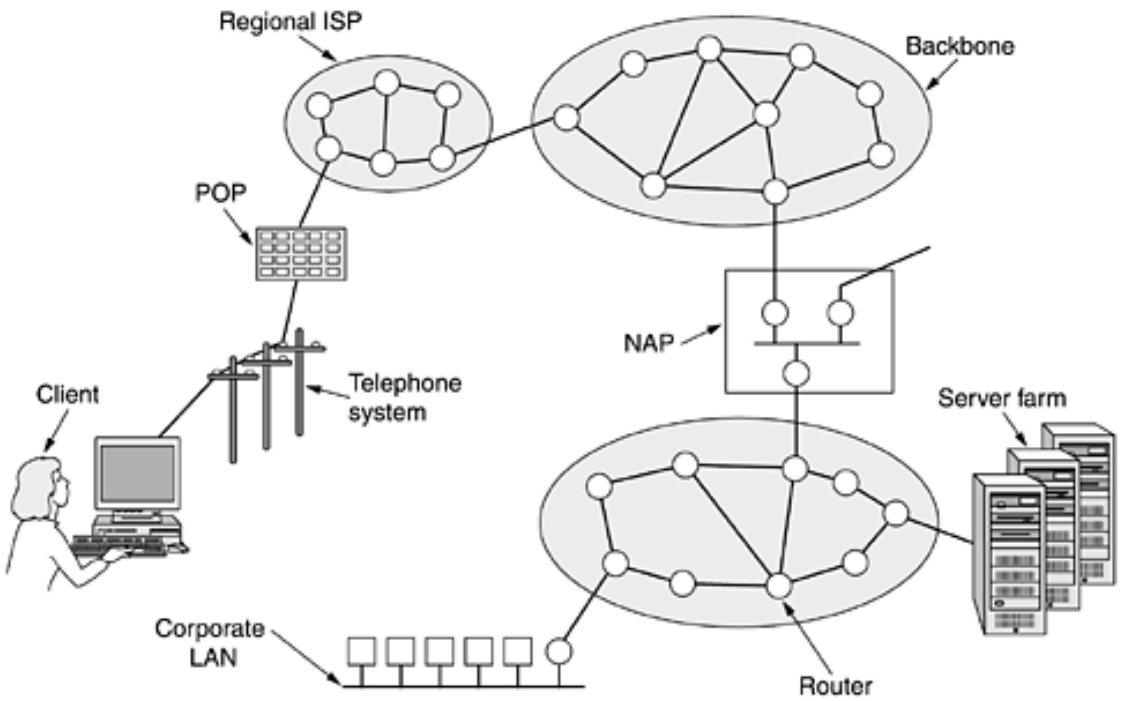
- The architecture of the Internet is ever-changing due to continuous changes in the technologies as well as the nature of the service provided.
- The heterogeneity and vastness of the Internet make it difficult to describe every aspect of its architecture.
- The overall architecture can be described in three levels –
  - Backbone ISP (Internet Service Provider)
  - Regional ISPs
  - Clients

- The following diagram shows the three levels –



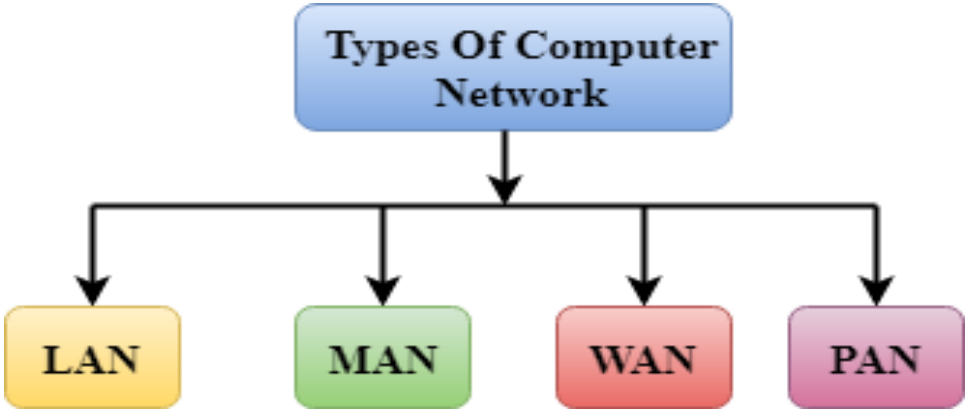
- Backbone ISP (Internet Service Provider) – Backbone ISPs are large international backbone networks. They are equipped with thousands of routers and store enormous amounts of information in data centers, connected through high bandwidth fiber optic links.
- Everyone needs to connect with a backbone ISP to access the entire Internet.
- There are different ways through which a client can connect to the ISP. A commonly used way is DSL (Digital Subscriber Line) which reuses the telephone connection of the user for transmission of digital data.
- The user uses a dial-up connection instead of the telephone call.

- For high-speed Internet access, the connectivity can be done through FTTH (Fiber to the Home), that uses optical fibers for transmitting data.
- Nowadays, most Internet access is done through the wireless connection to mobile phones from fixed subscribers, who transmit data within their coverage area.



Classification of networks

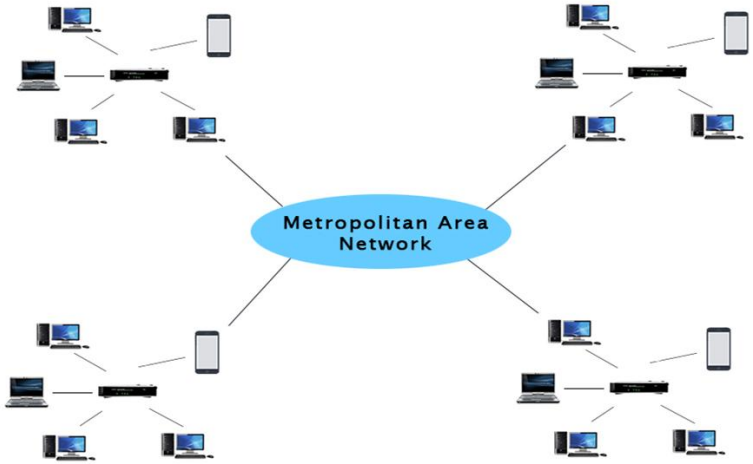
- The network can be classified in following manner:
- There are many different network protocols, each has its own characteristics and application domain.
  - Personal area network (PAN)
  - Local area network (LAN)
  - Metropolitan area network (MAN)
  - Wide area network (WAN)



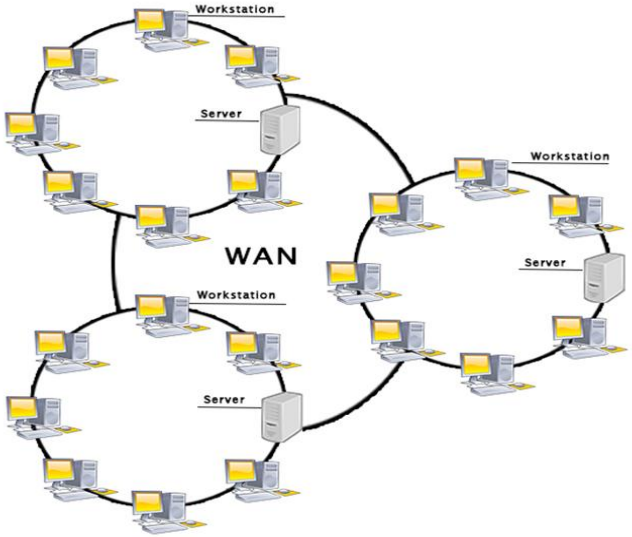
- Personal area network (PAN):
  - The PAN provides interconnection of devices within the range of an individual person, typically within a range of 10 meters.
  - Personal Area Network is used for connecting the computer devices of personal use is known as Personal Area Network.
    - Example of wired PAN: Infrared, Bluetooth etc
    - Examples of wireless PAN: Cell phone headsets, wireless keyboards, wireless mice, printers, bar code scanners etc

- Local area network (LAN):
  - Local Area Network is a group of computers connected to each other in a small area such as building, office.
  - LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc.
  - The data is transferred at an extremely faster rate in Local Area Network.
  - Local Area Network provides higher security.
    - Example: Networking in the home, school, library, laboratory, college/ university campus, or office.

- Metropolitan area network (MAN):
  - A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.
  - MAN or Metropolitan area Network covers a larger area than that of a LAN and smaller area as compared to WAN.
  - Uses Of Metropolitan Area Network:
    - MAN is used in communication between the banks in a city.
    - It can be used in a college within a city.



- Wide area network (WAN):
  - A Wide Area Network is a network that extends over a large geographical area such as states or countries.
  - A Wide Area Network is not limited to a single location, but it spans over a large geographical area through a telephone line, fiber optic cable or satellite links.
  - The internet is one of the biggest WAN in the world.
  - A Wide Area Network is widely used in the field of:
    - Business, government, and education.



Components of a network

- Computer network components are the major parts which are needed to install the software.
- Computer networks components comprise both physical parts as well as the software required for installing computer networks.
- Some important network components are:
  - NIC
  - Hub
  - Switch
  - Router
  - Modem
  - Cable

- Network interface card (NIC):
  - Each system or computer in a computer network must have a card called network interface card (NIC).
  - The main purpose of NIC is to format the data, send the data and receive the data at the receiving node.

- Hub:
  - A Hub is a hardware device that divides the network connection among multiple devices.
  - When computer requests for some information from a network, it first sends the request to the Hub through cable.
  - Hub will broadcast this request to the entire network. All the devices will check whether the request belongs to them or not. If not, the request will be dropped.
  - The process used by the Hub consumes more bandwidth and limits the amount of communication.
  - Nowadays, the use of hub is obsolete, and it is replaced by more advanced computer network components such as Switches, Routers.

- Switch:
  - A switch is a hardware device that connects multiple devices on a computer network. A Switch contains more advanced features than Hub.
  - The Switch contains the updated table that decides where the data is transmitted or not. Switch delivers the message to the correct destination based on the physical address present in the incoming message.
  - A Switch does not broadcast the message to the entire network like the Hub. It determines the device to whom the message is to be transmitted.
  - Therefore, we can say that switch provides a direct connection between the source and destination.
  - It increases the speed of the network.

- Router:
  - When we talk about computer network components, the other device that used to connect a LAN with an internet connection is called Router.
  - When you have two distinct networks (LANs) or want to share a single internet connection to multiple computers, we use a Router.
  - There are two types of Router: wired and wireless. The choice depends on your physical office/home setting, speed and cost.

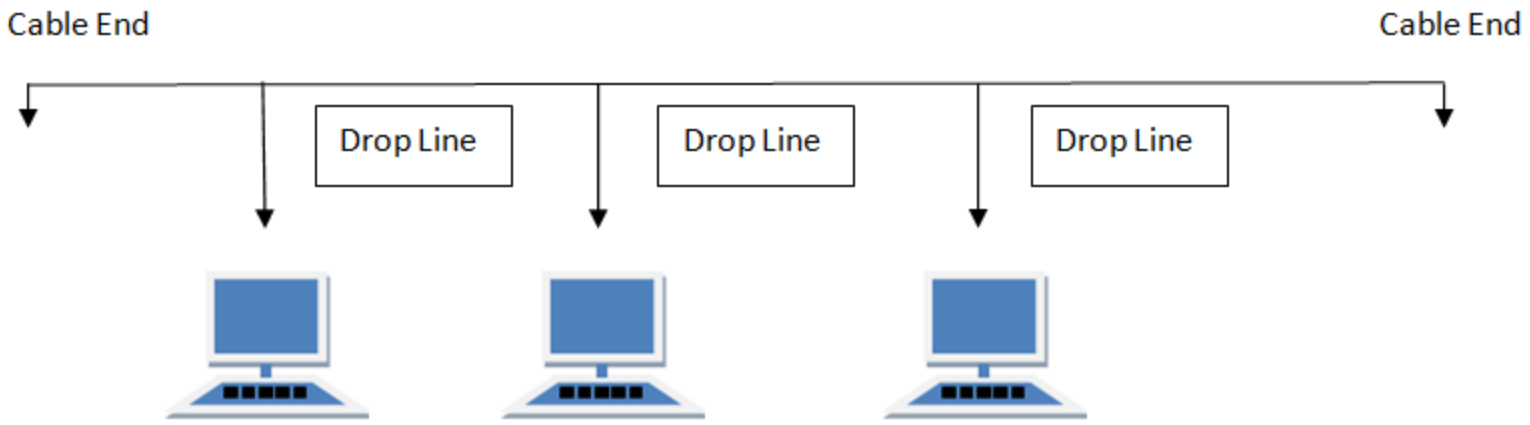
- Modem:
  - A modem is a hardware device that allows the computer to connect to the internet over the existing telephone line.
  - A modem is not integrated with the motherboard rather than it is installed on the PCI slot found on the motherboard.
  - It stands for Modulator/Demodulator. It converts the digital data into an analog signal over the telephone lines.

- LAN Cable:
  - A local area Network cable is also known as data cable or Ethernet cable which is a wired cable used to connect a device to the internet or to other devices like computer, printers, etc.

Network topology and transmission

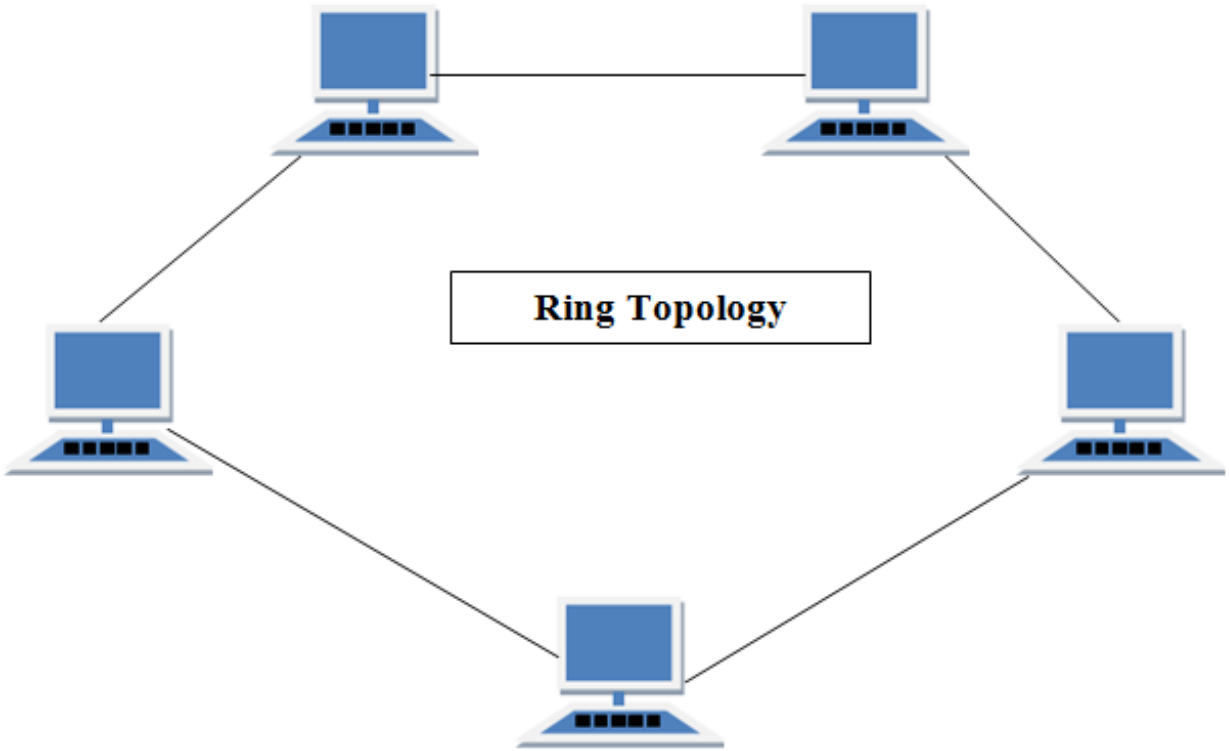
- Geometric representation of how the computers are connected to each other is known as topology.
- Topologies may define both physical and logical aspect of the network.
- There are six types of topologies–
- Bus
- Ring
- Mesh
- Star
- Tree
- Hybrid

- Bus Topology:
  - In bus topology there is a main cable and all the devices are connected to this main cable through drop lines.
  - There is a device called tap that connects the drop line to the main cable.
  - Since all the data is transmitted over the main cable, there is a limit of drop lines and the distance a main cable can have.

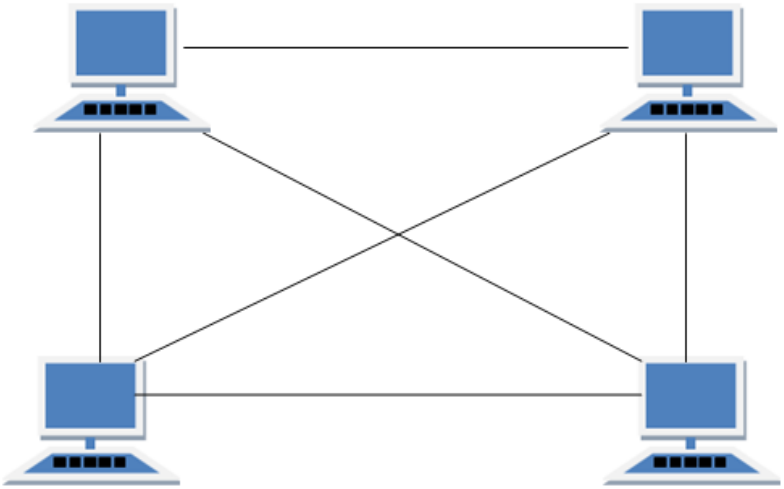


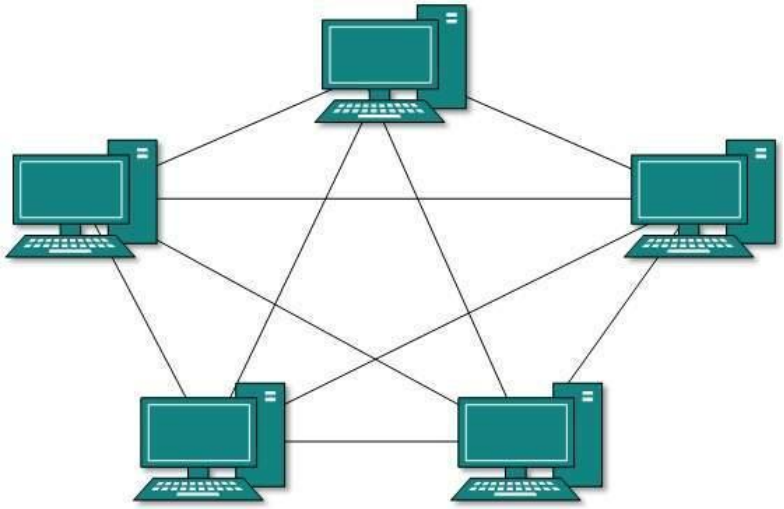
- Ring Topology:
  - In ring topology each device is connected with the two devices on either side of it.
  - There are two dedicated point to point links a device has with the devices on the either side of it.
  - This structure forms a ring thus it is known as ring topology.
  - If a device wants to send data to another device then it sends the data in one direction.

- Ring Topology:
  - Each device in ring topology has a repeater, if the received data is intended for other device then repeater forwards this data until the intended device receives it.

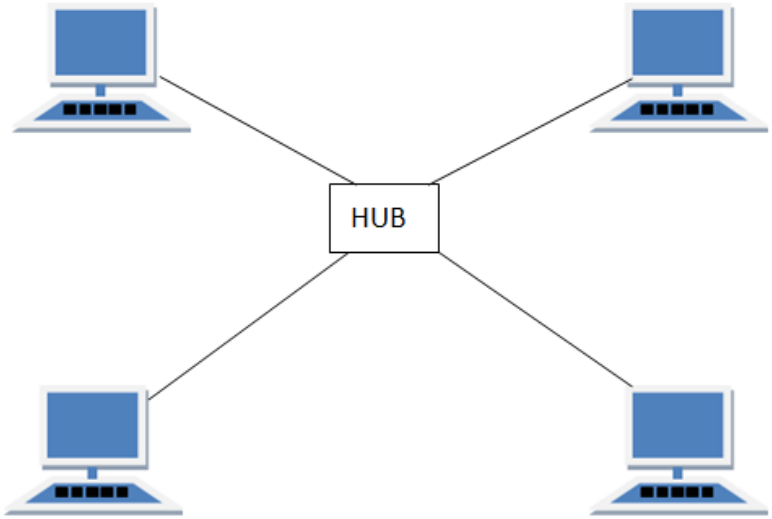


- Mesh Topology:
  - In mesh topology each device is connected to every other device on the network through a dedicated point-to-point link.
  - When we say dedicated it means that the link only carries data for the two connected devices only.
  - Lets say we have n devices in the network then each device must be connected with (n-1) devices of the network.
  - Number of links in a mesh topology of n devices would be n(n-1)/2.

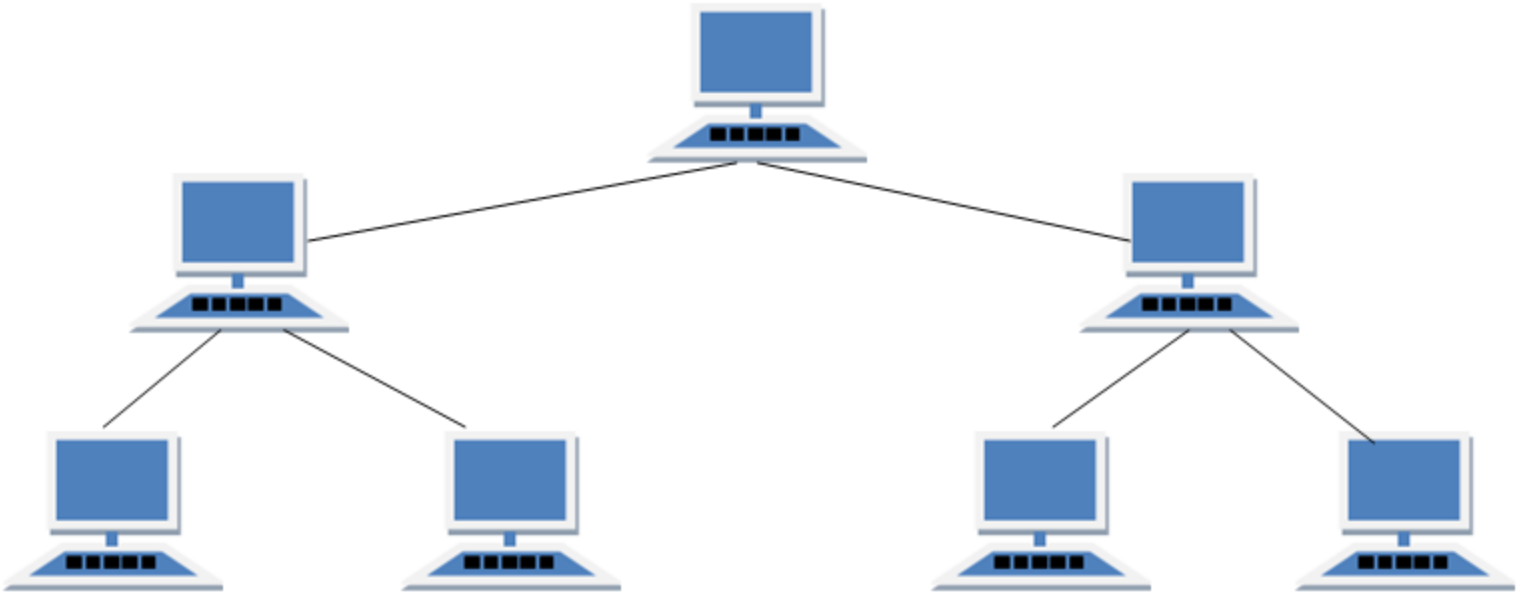




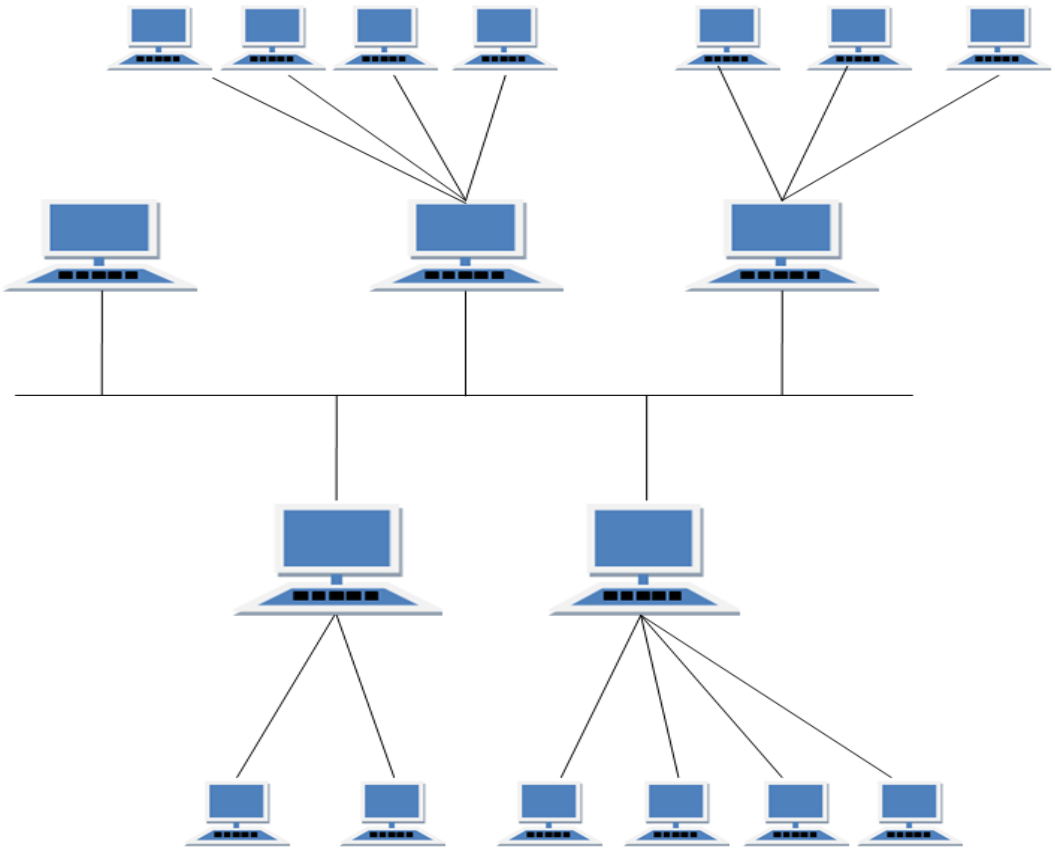
- Star Topology:
  - In star topology each device in the network is connected to a central device called hub.
  - Unlike Mesh topology, star topology doesn't allow direct communication between devices, a device must have to communicate through hub.
  - If one device wants to send data to other device, it has to first send the data to hub and then the hub transmit that data to the designated device.



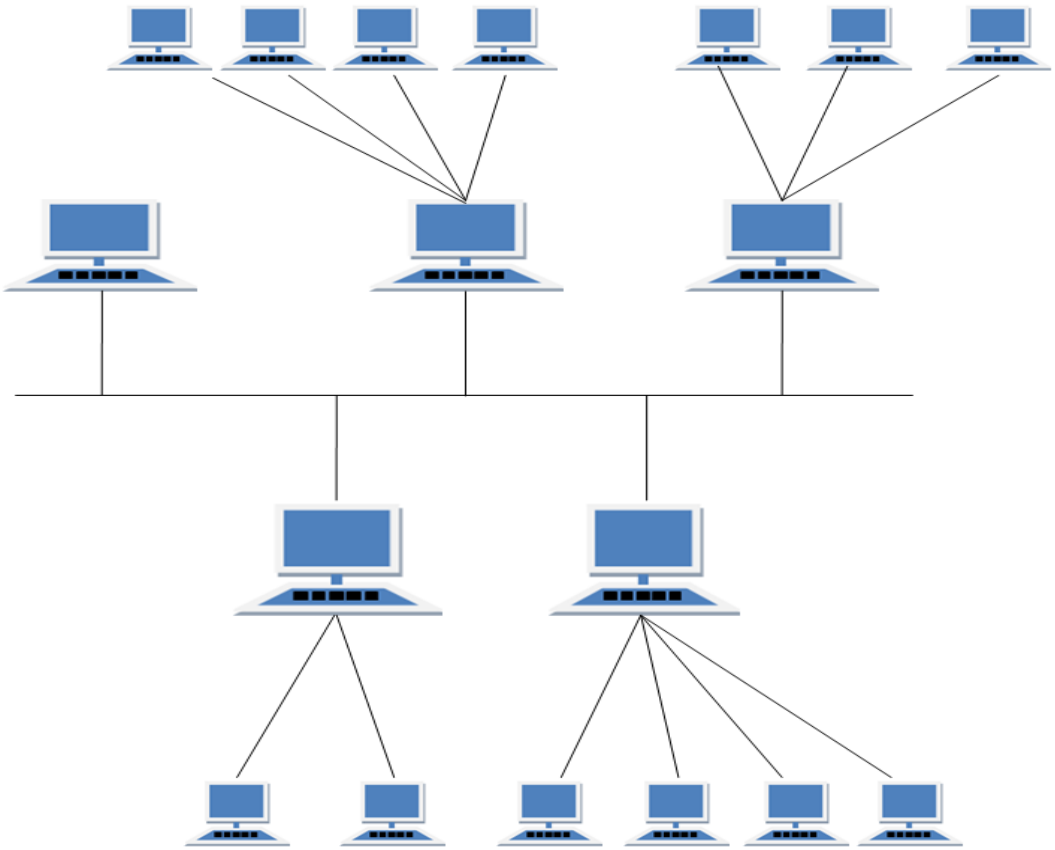
- Tree Topology:
  - This is also known as Hierarchical Topology, this is the most common form of network topology in use presently.
  - This topology imitates as extended Star topology and inherits properties of bus topology.
  - This topology divides the network in to multiple levels/layers of network.



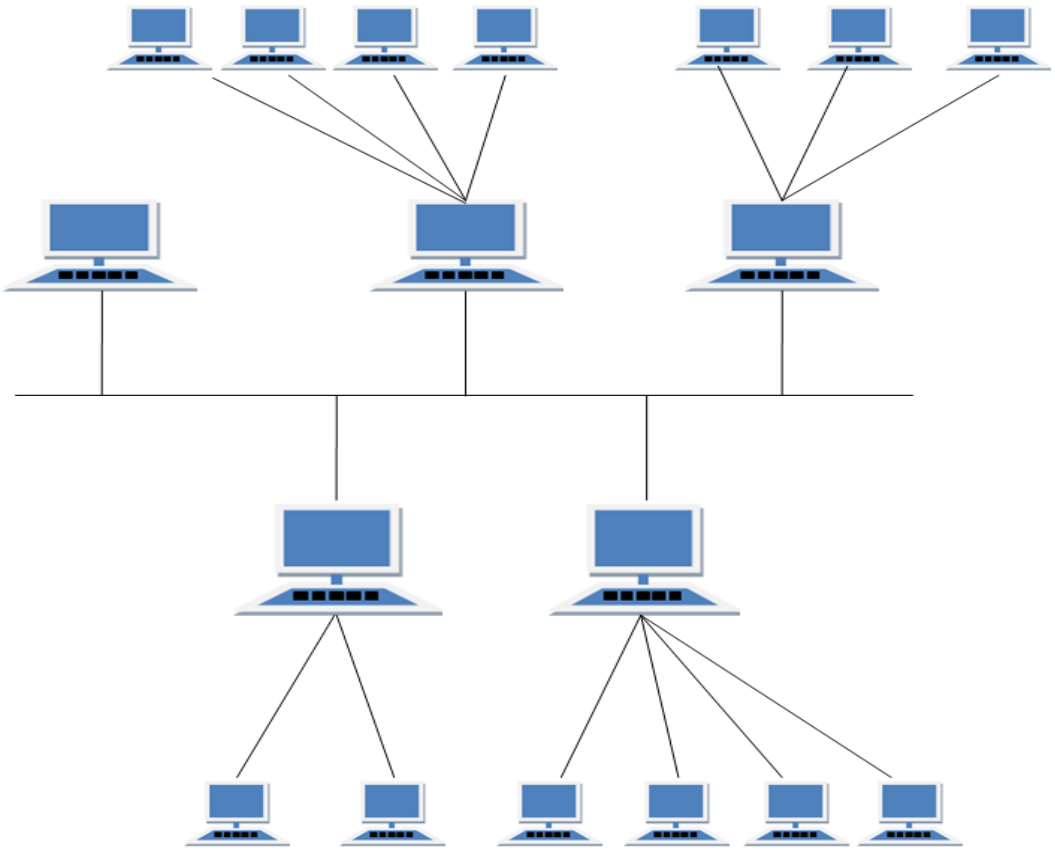
- Hybrid Topology:
  - A combination of two or more topology is known as hybrid topology.
  - For example a combination of star and mesh topology is known as hybrid topology.



- Hybrid Topology:
  - A combination of two or more topology is known as hybrid topology.
  - For example a combination of star and mesh topology is known as hybrid topology.



- Hybrid Topology:
  - A combination of two or more topology is known as hybrid topology.
  - For example a combination of star and mesh topology is known as hybrid topology.

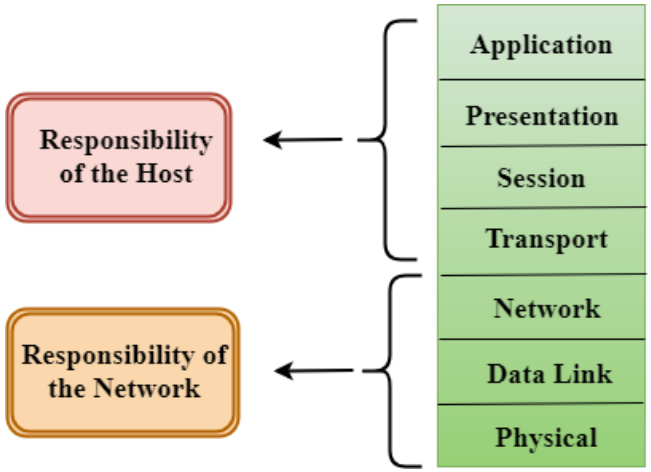


Layered models

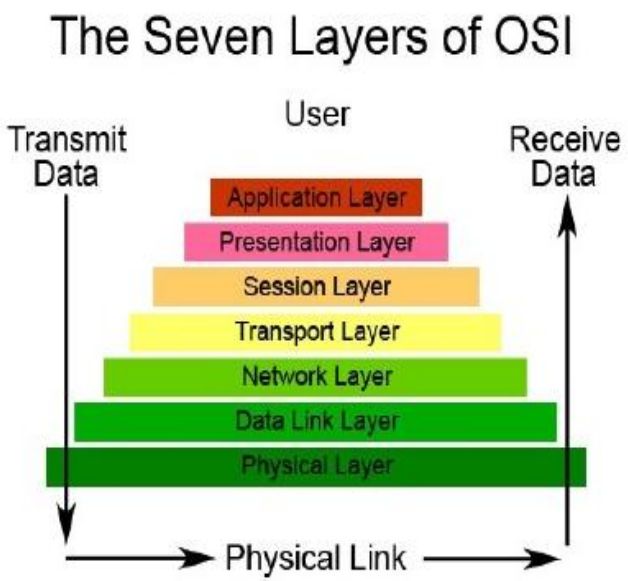
- In computer science, the concept of network layers is a framework that helps to understand complex network interactions.
- There are two models that are widely referenced today: OSI and TCP/IP.
- The concepts are similar, but the layers themselves differ between the two models.
- While TCP/IP is the newer model, the Open Systems Interconnection (OSI) model is still referenced a lot to describe network layers.

- OSI Reference Model:
  - OSI stands for **Open System Interconnection** is a reference model that describes how information from a software application in one computer moves through a physical medium to the software application in another computer.
  - OSI consists of seven layers, and each layer performs a particular network function.
  - OSI model was developed by the International Organization for Standardization (ISO) in 1984, and it is now considered as an architectural model for the inter-computer communications.
  - OSI model divides the whole task into seven smaller and manageable tasks. Each layer is assigned a particular task.
  - Each layer is self-contained, so that task assigned to each layer can be performed independently.

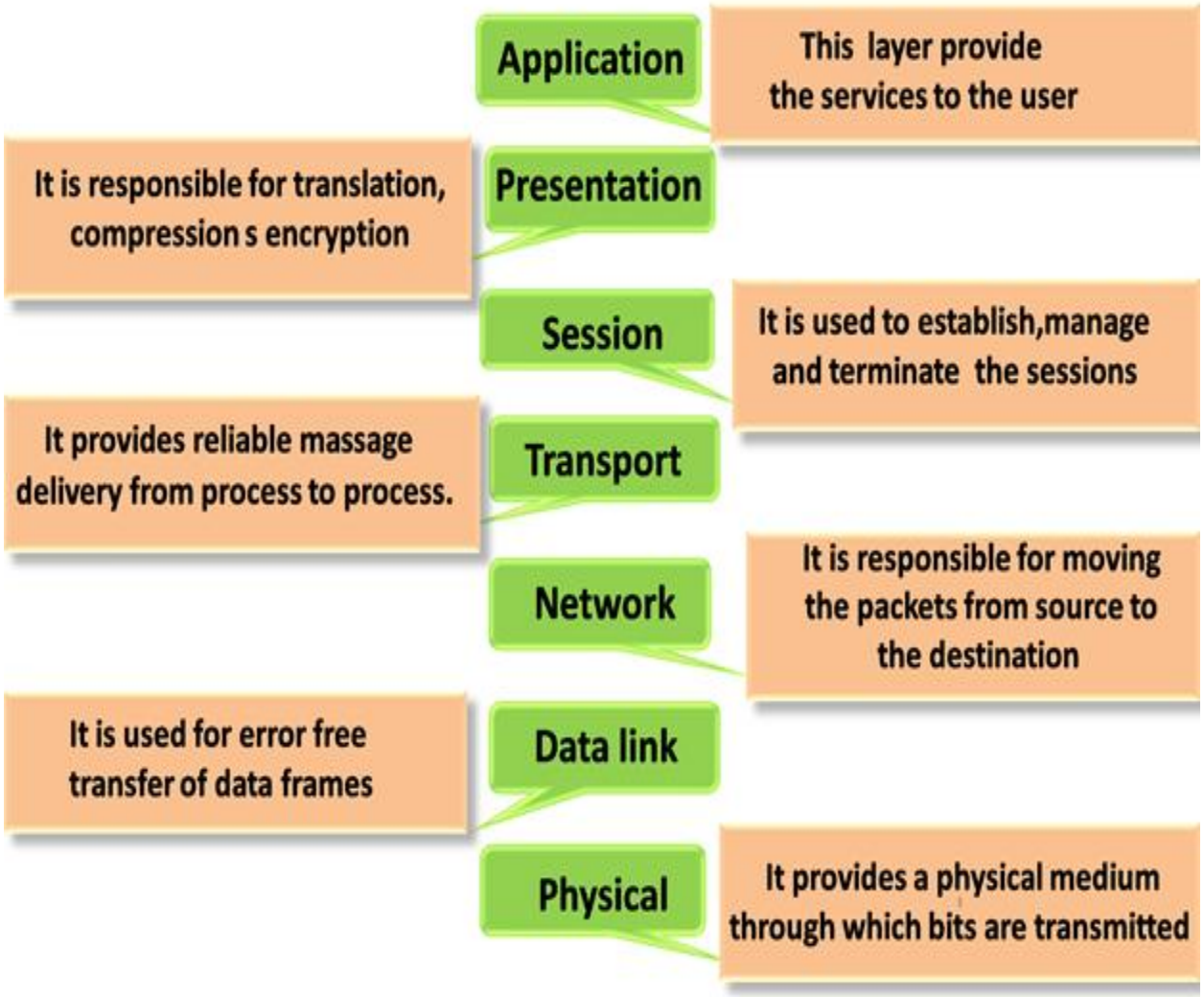
- The OSI model is divided into two layers: Upper layers and Lower layers.
- The upper layer:
  - This layer of the OSI model mainly deals with the application related issues, and they are implemented only in the software. The application layer is closest to the end user.
- The lower layer:
  - This layer of the OSI model deals with the data transport issues. The data link layer and the physical layer are implemented in hardware and software. The physical layer is the lowest layer of the OSI model and is closest to the physical medium.



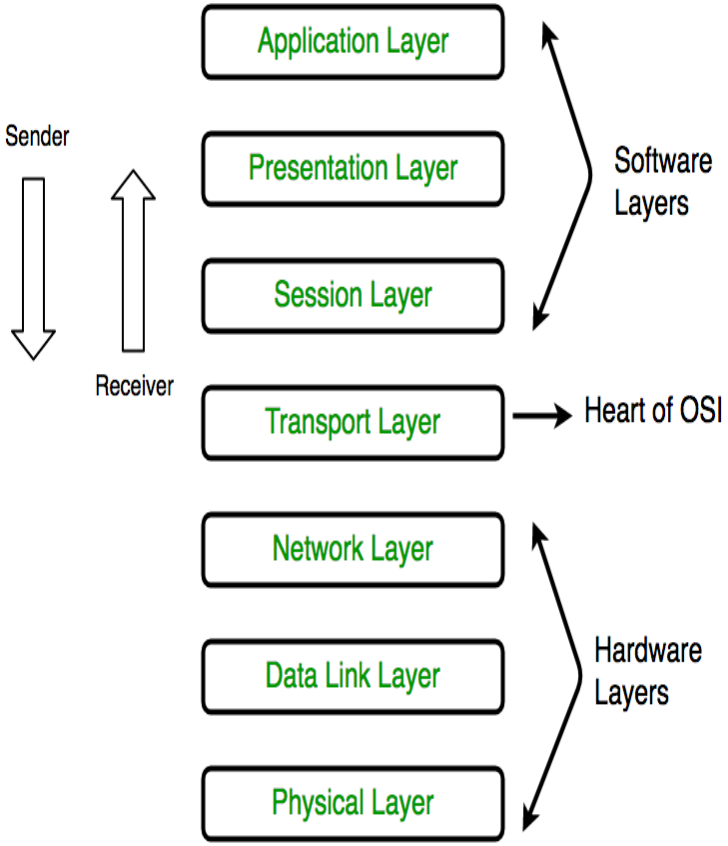
- There are the seven OSI layers. A list of seven layers are given below:
  1. Physical Layer
  2. Data-Link Layer
  3. Network Layer
  4. Transport Layer
  5. Session Layer
  6. Presentation Layer
  7. Application Layer



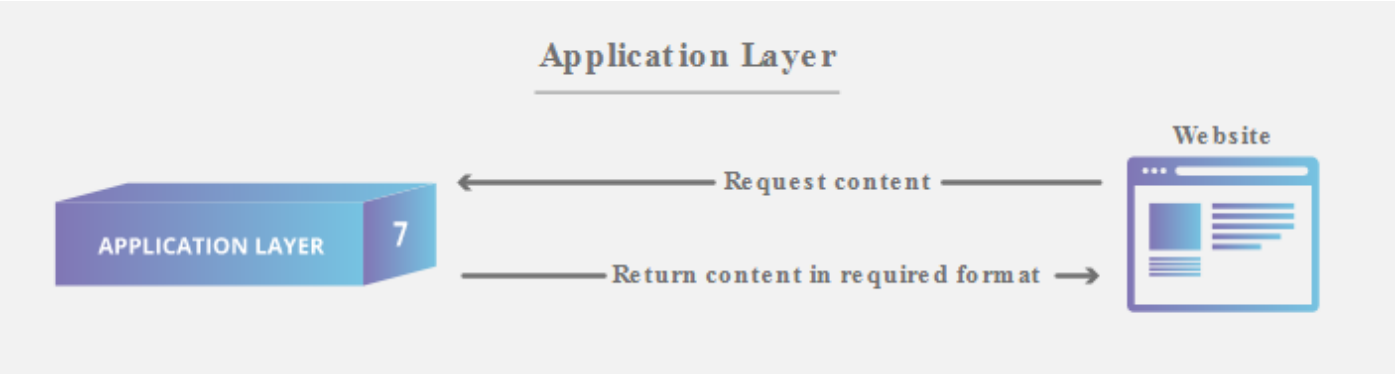
- Functions of the OSI Layers:
  - Each layer in the OSI model has a different function:



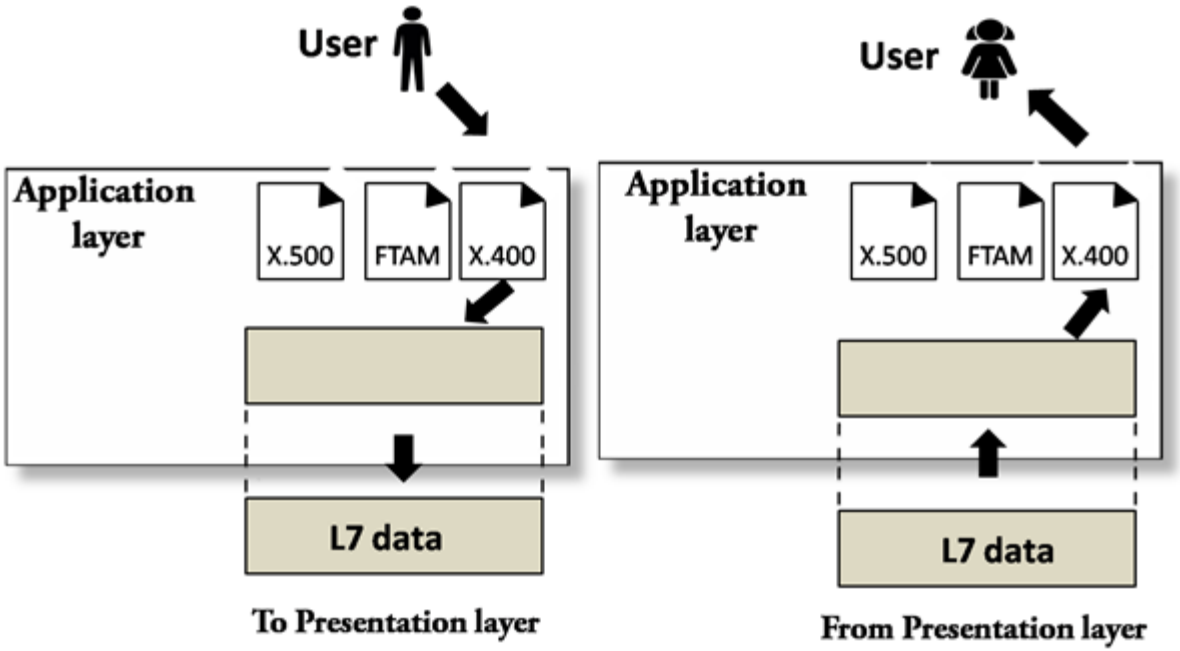
- Functions of the OSI Layers:
  - The seven abstraction layers of the OSI model can be explained as follows:
  - For the OSI model, let's start at the top (Application layer) and work our way down to bottom (Physical layer):



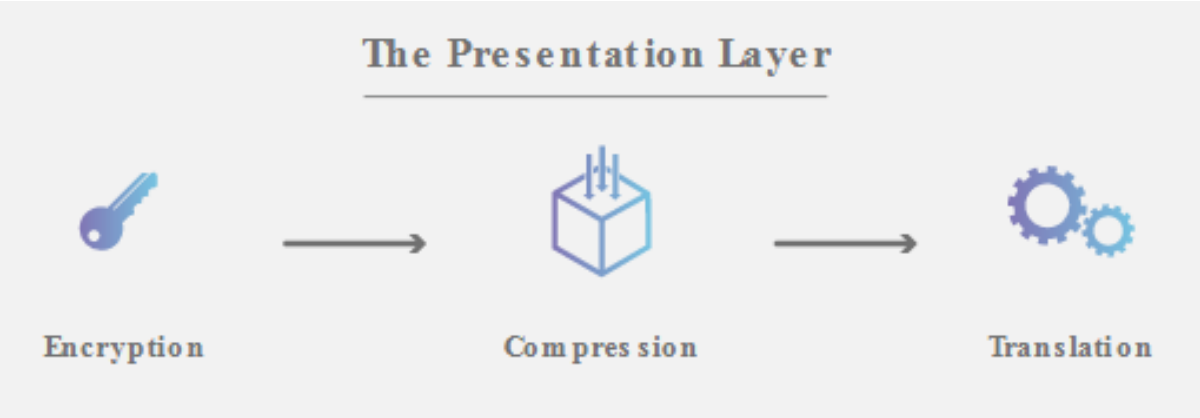
- 7. The Application Layer:
  - This is the only layer that directly interacts with data from the user.
  - Software applications like web browsers and email clients rely on the application layer to initiate communications.
  - But it should be made clear that the application layer is responsible for the protocols and data manipulation that the software relies on to present meaningful data to the user.



- 7. The Application Layer:
  - Application layer protocols include HTTP as well as SMTP (Simple Mail Transfer Protocol is one of the protocols that enables email communications).

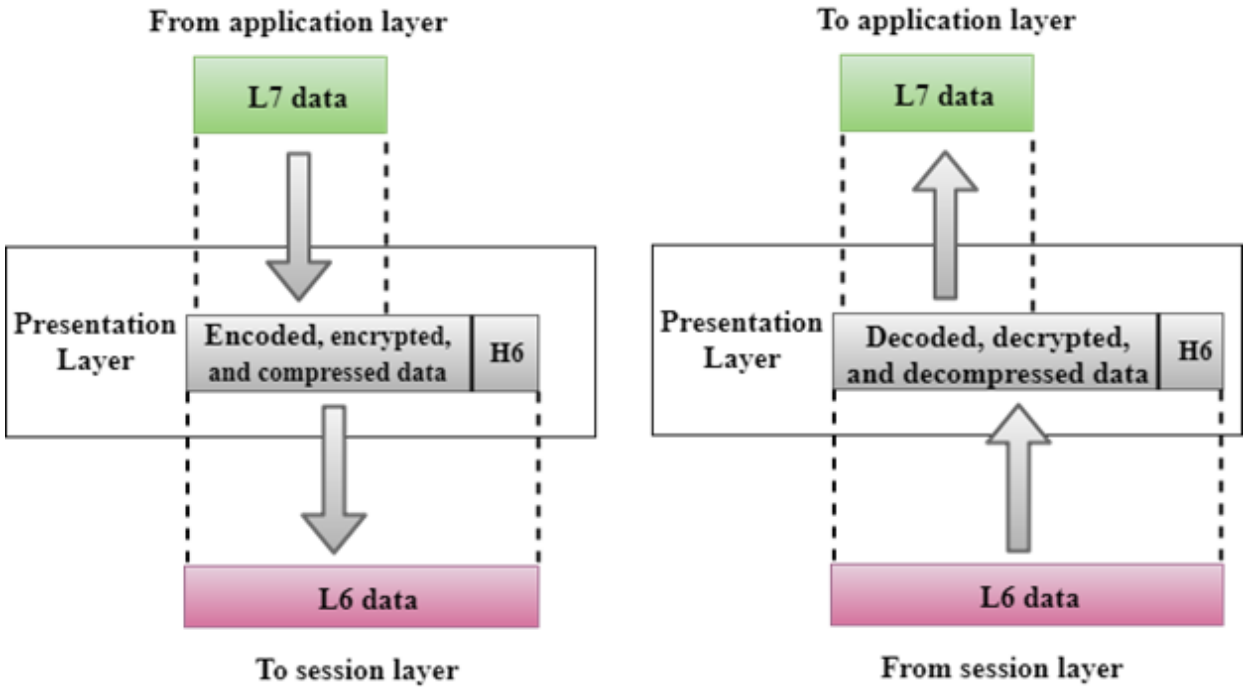


- 6. The Presentation Layer
  - This layer is primarily responsible for preparing data so that it can be used by the application layer; in other words, layer 6 makes the data presentable for applications to consume.
  - A Presentation layer is mainly concerned with the syntax and semantics of the information exchanged between the two systems.
  - It acts as a data translator for a network.
  - This layer is a part of the operating system that converts the data from one presentation format to another format.



- 6. The Presentation Layer
  - The presentation layer is responsible for translation (Encoding/Decoding), Encryption/Decryption, and Compression/Decompression of data.
  - Translation:
    - Two communicating devices communicating may be using different encoding methods, so layer 6 converts the data from **sender-dependent format** into a common format and changes the common format into **receiver-dependent format** at the receiving end.
  - Encryption/Decryption:
    - Encryption is needed to maintain privacy.
    - If the devices are communicating over an encrypted connection, layer 6 is responsible for adding the encryption on the sender's end as well as converting the encryption on the receiver's end so that it can present the application layer with decrypted readable data.

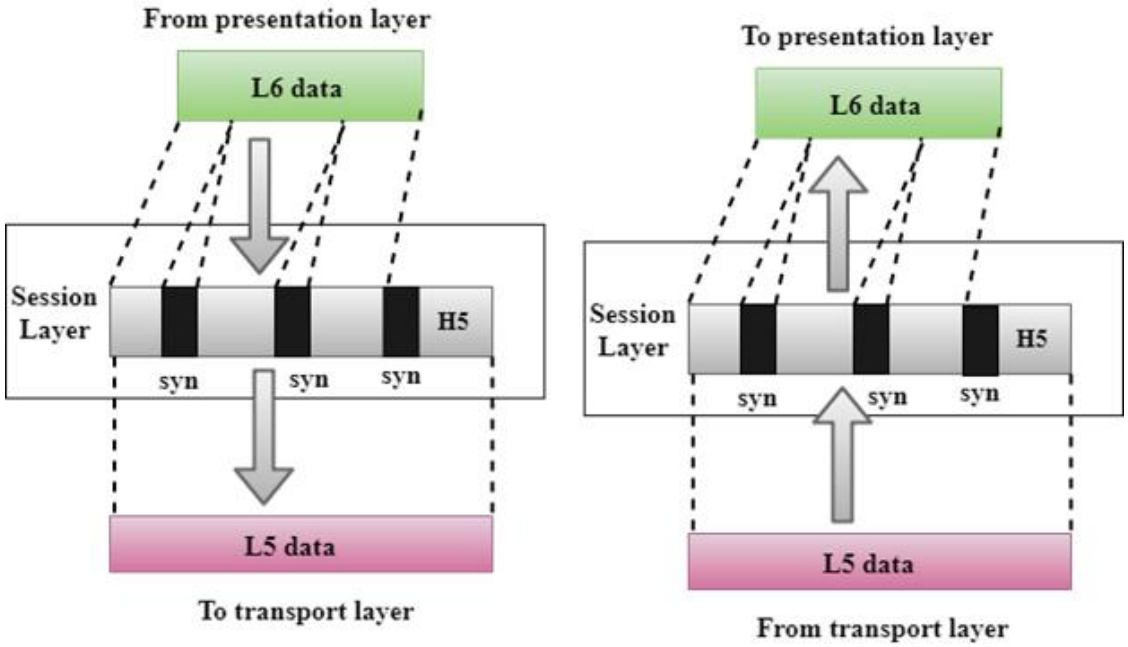
- 6. The Presentation Layer
  - Compression/Decompression:
    - Data compression is a process of compressing the data, i.e., it reduces the number of bits to be transmitted.
    - The presentation layer is also responsible for compressing data it receives from the application layer before delivering it to layer 5 and decompressing while transferring the data to layer 7.
    - This helps improve the speed and efficiency of communication.



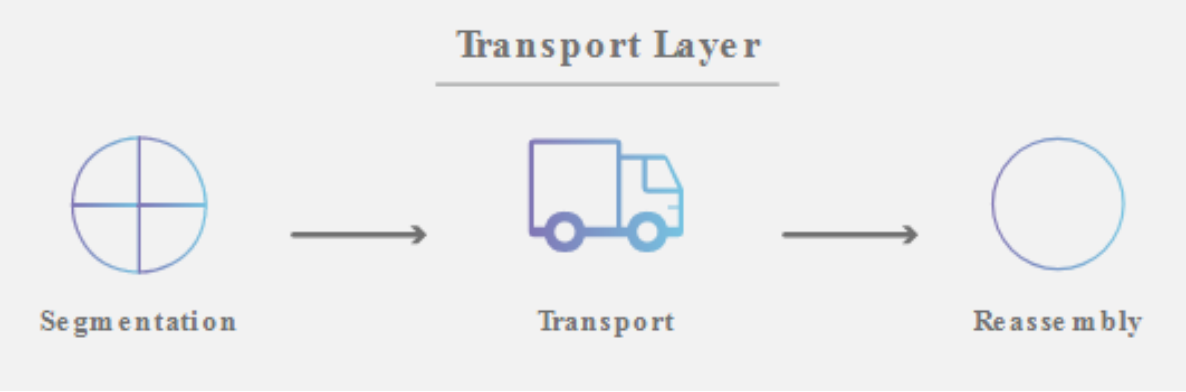
- 5. The Session Layer:
  - The Session layer is used to maintain and synchronizes the interaction between communicating devices.
  - This is the layer responsible for opening and closing communication between the two devices.
  - The time between when the communication is opened and closed is known as the session.
  - The session layer ensures that the session stays open long enough to transfer all the data being exchanged, and then promptly closes the session in order to avoid wasting resources.

- 5. The Session Layer:
  - The session layer also synchronizes data transfer with checkpoints.
  - This means, it adds some checkpoints in between when transmitting the data in a sequence.
  - If some error occurs in the middle of the transmission of data, then the transmission will take place again from the checkpoint.
  - This process is known as Synchronization and recovery.

- 5. The Session Layer:
  - For example, if a 100 megabyte file is being transferred, the session layer could set a checkpoint every 5 megabytes. In the case of a disconnect or a crash after 52 megabytes have been transferred, the session could be resumed from the last checkpoint, meaning only 50 more megabytes of data need to be transferred. Without the checkpoints, the entire transfer would have to begin again from scratch.

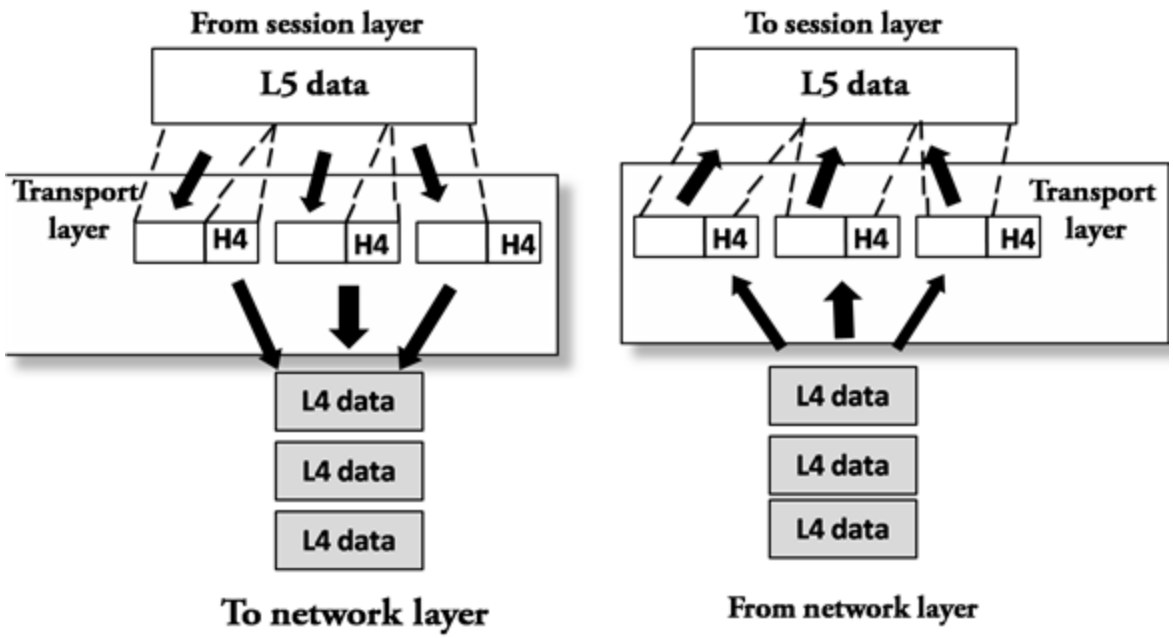


- 4. The Transport Layer
  - The Transport layer ensures that messages are transmitted in the order in which they are received and there is no duplication of data.
  - The main responsibility of the transport layer is to transfer the data completely.
  - It receives the data from the upper layer and converts them into smaller units known as **segments**.
  - This layer provides a point-to-point connection between source and destination and delivers the data reliably.

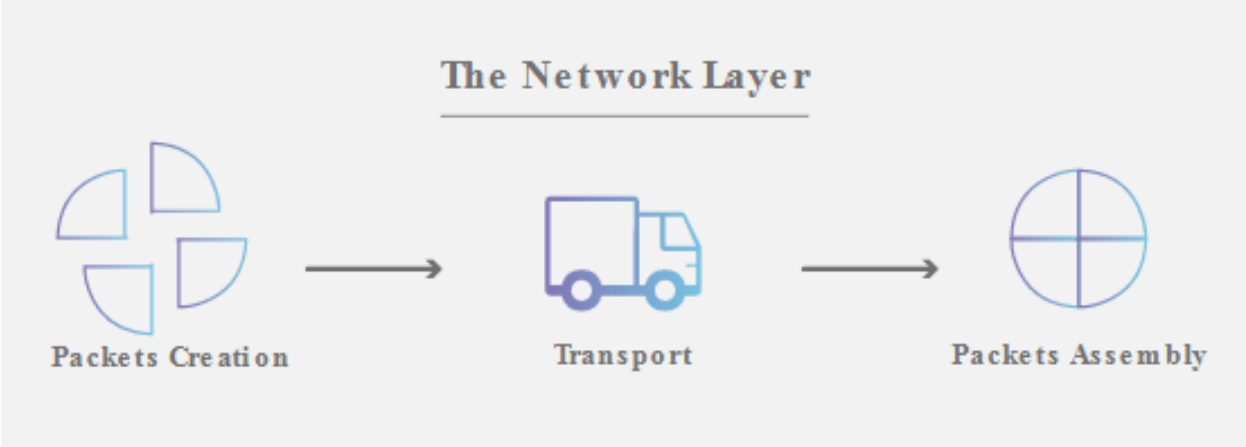


- 4. The Transport Layer
  - The Layer 4 is responsible for end-to-end communication between the two devices.
  - This includes taking data from the session layer and breaking it up into chunks called **segments** before sending it to layer 3.
  - The transport layer on the receiving device is responsible for **reassembling** the segments into data which the session layer can consume.
  - Computers run several programs simultaneously due to this reason, the transport layer adds the **header** to each segment that contains the address known as a **service-point address**.
  - The transport layer is also responsible for flow control and error control.

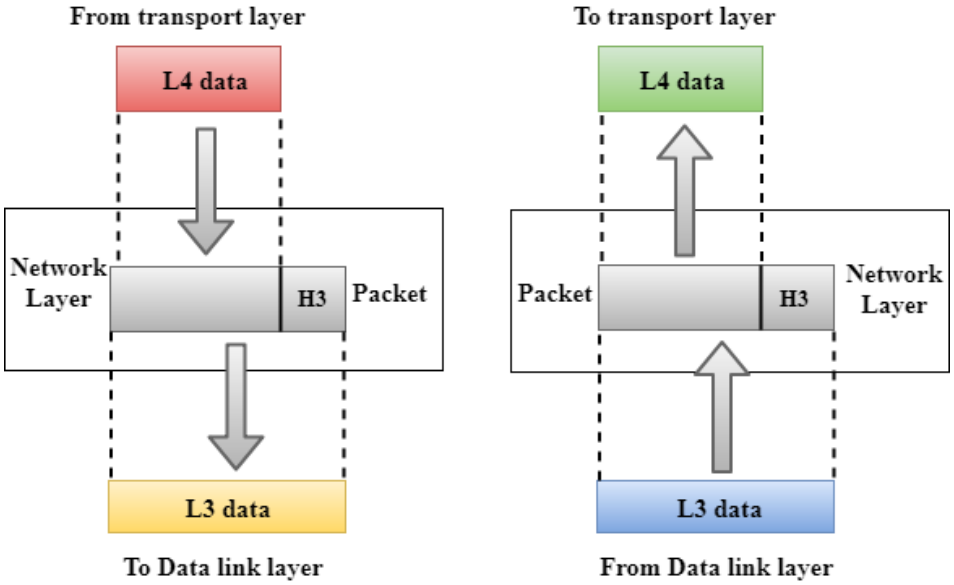
- 4. The Transport Layer
  - The flow control determines an optimal speed of transmission to ensure that a sender with a fast connection doesn't overwhelm a receiver with a slow connection.
  - The transport layer performs error control on the receiving end by ensuring that the data received is complete, and the sender transport layer ensures that message reach at the destination without any error.



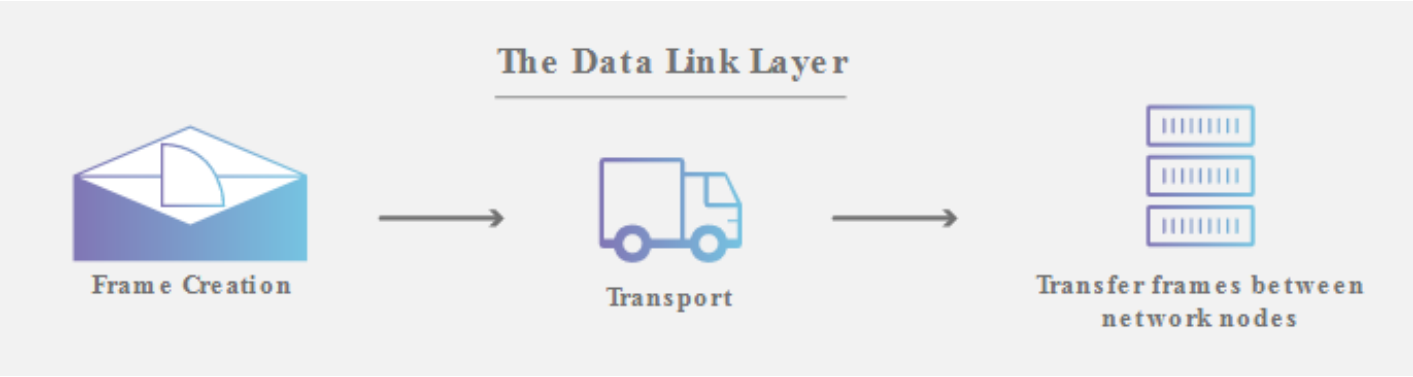
- 3. The Network Layer
  - This layer manages device addressing and keeps a track of the location of devices on the network.
  - The network layer is also responsible for facilitating data transfer between two different networks. If the two devices communicating are on the same network, then the network layer is unnecessary.
  - It determines the best path to move data from source to the destination based on the network conditions, the priority of service, and other factors.



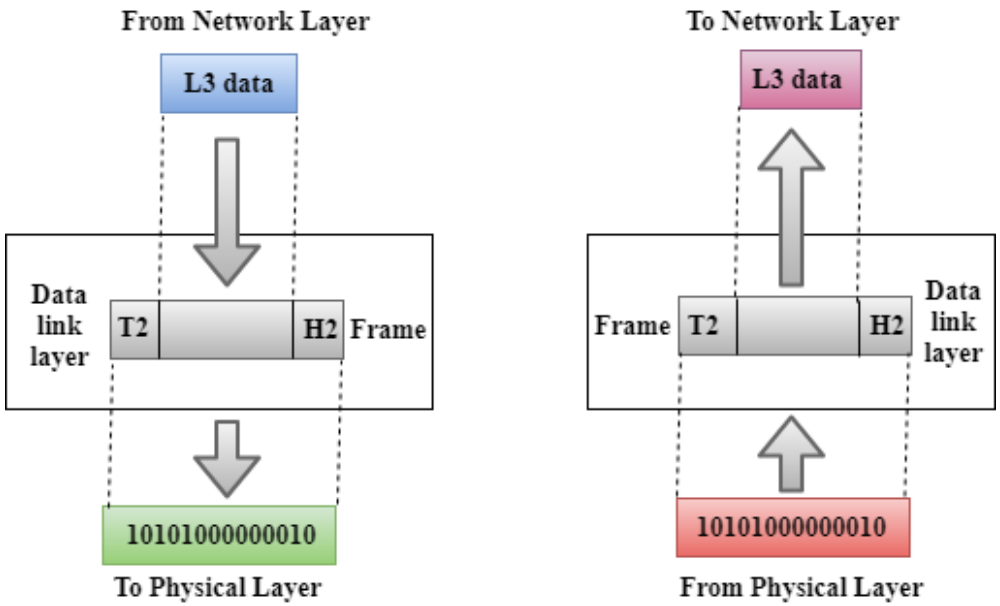
- 3. The Network Layer
  - A Network Layer receives the segments from the transport layer and converts them into packets.
  - This process is known as Packetizing.
  - A Network layer adds the source and destination address to the header of the packet. This address is used to identify the device on the internet.
  - The network layer also finds the best physical path for the data to reach its destination; this is known as routing.



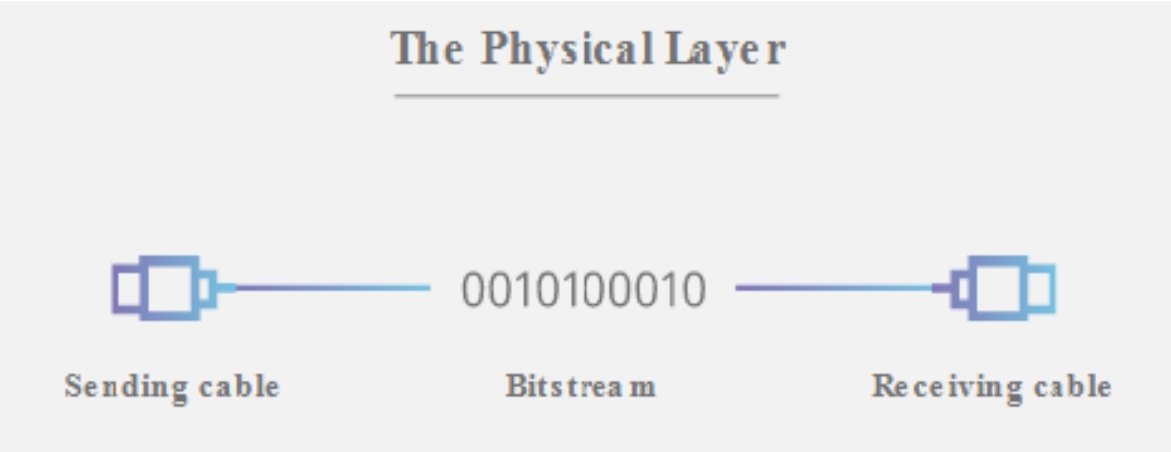
- 2. The Data Link Layer
  - The data link layer is very similar to the network layer.
  - This layer is responsible for the error-free transfer of data.
  - It provides a reliable and efficient communication between two or more devices.
  - The data link layer takes packets from the network layer and breaks them into smaller pieces called frames. This is called as framing.
  - The Data link layer adds a header and a trailer to the frame.
  - The header which is added to the frame contains the hardware destination and source address.



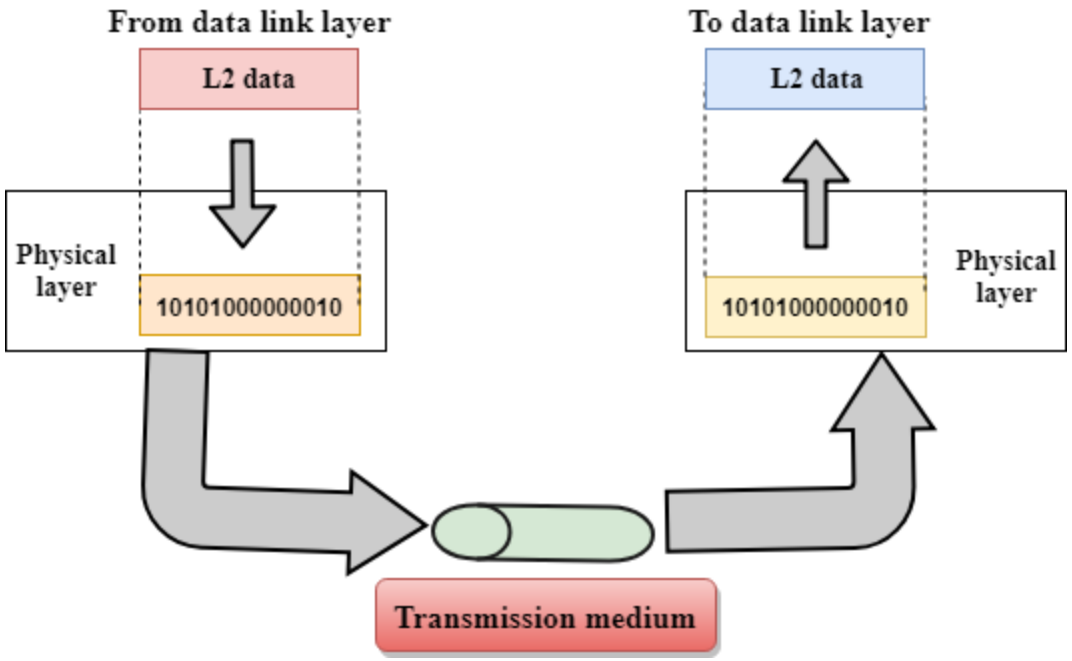
- 2. The Data Link Layer
  - The trailer which is added to the message frame before it is sent to the physical layer controls the errors in the frame.
  - If any error seems to occur, then the receiver sends the acknowledgment for the retransmission of the corrupted frames.
  - The data link layer is also responsible for flow control. It is the technique through which the constant data rate is maintained on both the sides so that no data get corrupted.



- 1. The Physical Layer
  - The main functionality of the physical layer is to transmit the individual bits from one node to another node.
  - It is the lowest layer of the OSI model.
  - It establishes, maintains and deactivates the physical connection.
  - This layer includes the physical equipment involved in the data transfer, such as the cables and switches

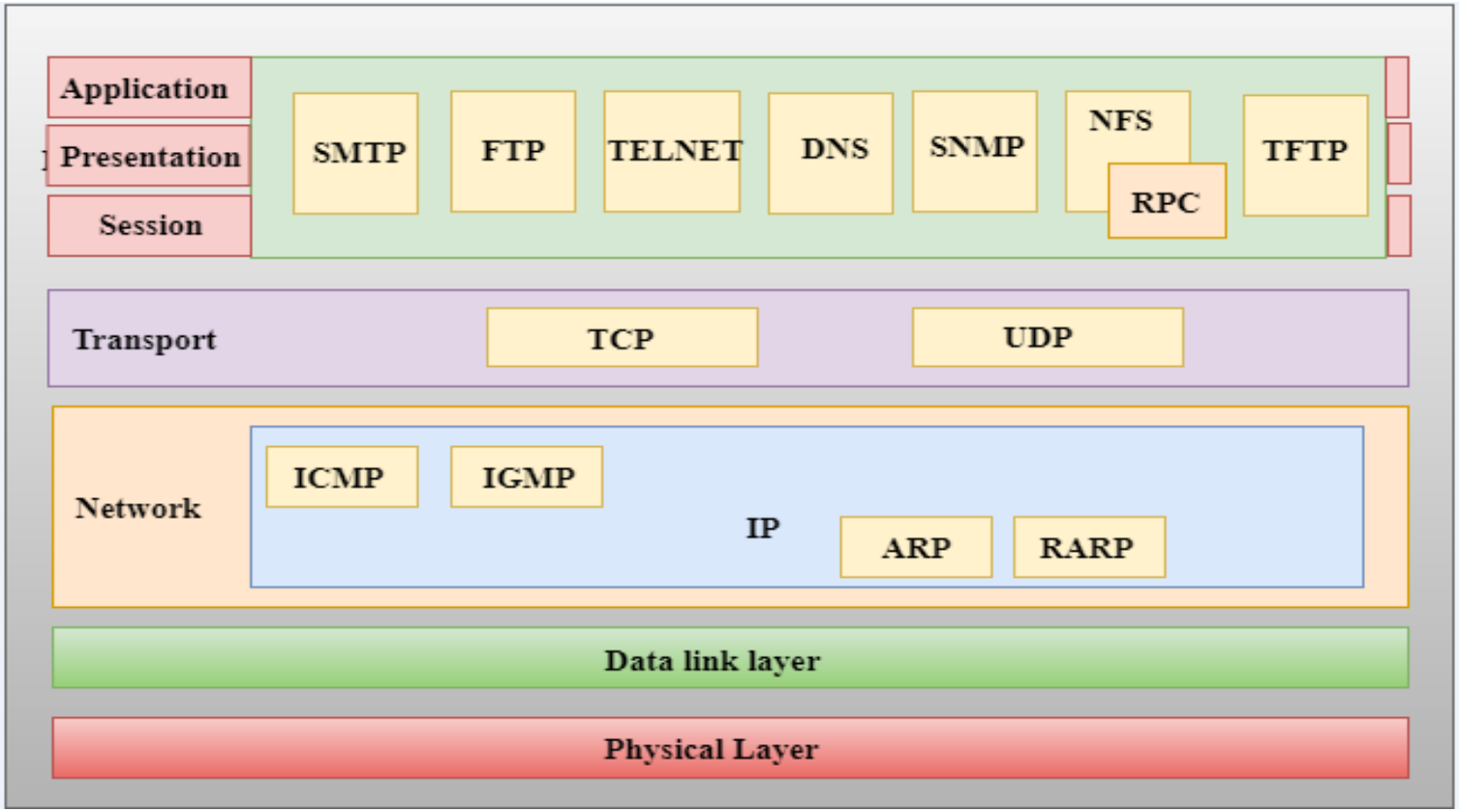


- 1. The Physical Layer
  - It defines the way how two or more devices can be connected physically.
  - This is also the layer where the data gets converted into a bit stream, which is a string of 1s and 0s.
  - The physical layer of both devices must also agree on a signal convention so that the 1's can be distinguished from the 0's on both devices.



- TCP/IP Reference Model:
  - The TCP/IP model was developed prior to the OSI model.
  - The TCP/IP model is not exactly similar to the OSI model.
  - The TCP/IP model consists of five layers: the application layer, transport layer, network layer, data link layer and physical layer.
  - The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model and these four layers are represented in TCP/IP model by a single layer called the application layer.
  - TCP/IP is a hierarchical protocol made up of interactive modules, and each of them provides specific functionality.
  - Here, hierarchical means that each upper-layer protocol is supported by two or more lower-level protocols.

- Layers in TCP/IP Model:
  - The TCP/IP Model has four layers:
    - Physical Layer and data link layer – combined known as host-to-network layer
    - Network layer
    - Transport
    - Application



- 1. Physical and Data Link Layer
  - Physical and Data Link Layers in TCP/IP model does not define any protocols, they support all the standard protocols.
  - They are combined known as host-to-network layer.
  - A network in TCP/IP internetwork can be LAN or WAN.

- 2. Network Layer
  - In the network layer, the TCP/IP model supports internetworking protocol in short known as IP.
  - The IP uses four protocols internally: ARP, RARP, ICMP & IGMP.
  - Internetworking protocol(IP):
    1. It is an unreliable connectionless protocol used by TCP/IP Model.
    2. This protocol is used for data transmission.
    3. This protocol doesn't do error checking or tracking of data, thus we cannot be sure that the data is actually reached its destination. This is why it is also known as best effort delivery service which means this protocol tries it best to send the data to its destination but doesn't take an guarantee.
    4. IP protocol transmits the data in form of small packets known as datagrams. Each of these datagrams are transmitted separately, thus they can take different routes and sometime duplicate datagrams can be reached to destination, also they are reached in no particular order at the destination.

- 2. Network Layer
  - Four protocol the Internetworking protocol(IP) uses:
  - Address Resolution Protocol (ARP)
    - ARP finds the physical address by using the IP address.
    - In a network, each device is known as physical address usually imprinted on Network interface card (NIC).
    - ARP protocol is used to find the physical address of a device whose internet address (IP address) is known.
  - Reverse Address Resolution Protocol (RARP)
    - RARP protocol helps to find the internet address of a device whose physical address is known.

- 2. Network Layer
  - Internet Control Message Protocol (ICMP)
    - IP in network layer sends data in form of small packets known as datagrams. ICMP protocol sends the datagrams problems back to sender. It is used for query and error reporting messages.
  - Internet Group Message Protocol (IGMP)
    - This protocol is used for simultaneous transmission of a message to a group of recipients.

- 3. Transport Layer
  - Transport layer in TCP/Model can be represented by three protocols:
    - Transmission control protocol (TCP), User data gram protocol (UDP) and Stream Control Transmission Protocol (SCTP).
    - These three protocols in transport layer are responsible for delivery of messages from one process to another.
    - The SCTP protocol was later introduced to meet the needs of newer applications.

- 3. Transport Layer
  - User Datagram protocol (UDP):

    1. UDP is a process to process protocol used for data transmission.
    2. It takes the data from upper layer of TCP/IP Model and adds following information to the data:
       a) Port Address – Source port address of 16 bits and destination port address of 16 bits added to the data so that it reaches to correct destination and displays the correct source of data.
       b) Checksum error control – 16 bits of checksum data is added to the data received from upper layer, this is used for error control.
       c) Length of data – Length defines the total bytes of data present in datagram.
    3. Although this protocol finds the error in the transmission of data, it doesn't specify the error which makes it hard to identify the actual error in transmission.

- 3. Transport Layer
  - Transmission control protocol (TCP)

    1. Unlike UDP which is a connectionless protocol, the TCP is a connection oriented protocol.
    2. A connection must be made between sender and receiver before the transmission of data.
    3. TCP protocol divides the data in small units called segments. Each segment contains the sequence number which makes it possible to rearrange the segments in correct order at the receiver side to make the complete data.
    4. It also adds acknowledgement number to the segments to verify that the data is actually reached its destination or not.
    5. Unlike UDP which is unable to specify the exact error in transmission, this protocol does error control and specifies the exact error which makes TCP a reliable protocol.

- 3. Transport Layer
  - Stream Control Transmission Protocol (SCTP)
    1. The SCTP protocol was later introduced to meet the needs of newer applications.
    2. This protocol combines the best features of TCP and UDP protocols.
    3. It was discovered later to support newer applications such as voice data transmission over the internet.

- 4. Application Layer
  1. This is the top most layer of TCP/IP model.
  2. It is used for interaction between user and application.
  3. There are several protocols used by the application layer for user interaction such as: HTTP, SNMP, SMTP, DNS, TELNET, FTP etc.
  - HTTP:
    - HTTP stands for Hypertext transfer protocol, it allows the user to interact with world wide web through the browser application.
    - HTTP can be used to transfer various types of data such as plain text, audio, video etc.
  - FTP:
    - FTP stands for File Transfer Protocol.
    - This protocol is used for transmitting files from one system to another system.

- 4. Application Layer
  - TELNET:
    - TELNET is a client-server protocol. It is a reliable connection oriented protocol.
    - This protocol is used on internet or on LAN to provide a bidirectional (both ways) text based communication through a virtual terminal connection.
  - DNS:
    - DNS stands for Domain Name System.
    - Each computer on a network has different IP address, a computer is known by its IP address.
    - DNS provides a mapping to a name to the IP Address so a name can be used to identify a system on network rather than IP address itself.

- 4. Application Layer
  - SNMP:
    - SNMP stands for Simple Network Management Protocol.
    - It managers the devices connected to the internet using TCP/IP protocol.
  - SMTP:
    - SMTP stands for Simple mail transfer protocol.
    - It is used for email services, using this protocol a email containing data can be sent to another email address.

Thank you

Unit – 2
Physical layer

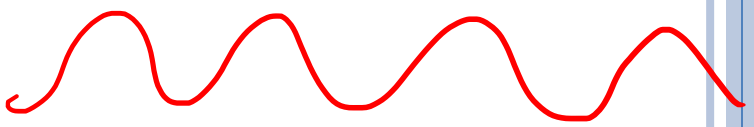
1.1 Introduction

- Physical layer in the OSI model plays the role of interacting with actual hardware and signaling mechanism.
- Physical layer is the only layer of OSI network model which actually deals with the physical connectivity of two different stations.
- This layer defines the hardware equipment, cabling, wiring, frequencies, pulses used to represent binary signals etc.
- Information can be transmitted or received using cables. The main purpose of the physical layer is to transmit information (data or text or even animated pictures) from a source to a destination.

  → x → 0|0|0|→
- Any form of such information cannot be transmitted as it is, just as in a postal service. Information is transmitted from a source computer to a destination computer in the form of electromagnetic signals using the transmission medium.

INTRODUCTION

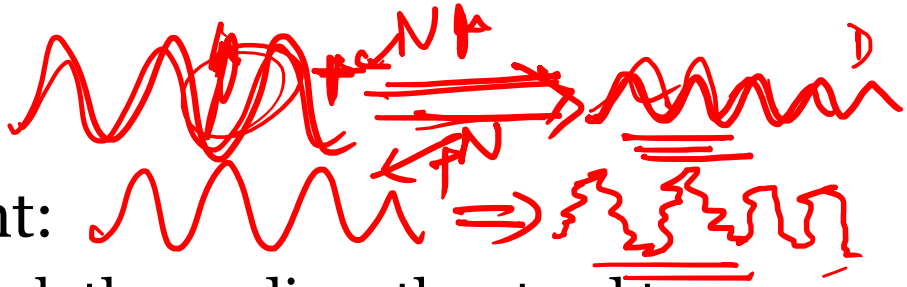
- Some basic definition:
  - Signals: Electric or electromagnetic encoding of data.
  - Signaling: It is the act of propagating the signal along a suitable medium.
- Types of signal:
  - Analog signal and Digital signal



  - Analog signal – A continuously varying electromagnetic wave that may be propagated over a variety of medium depending on the spectrum (e.g., wire, twisted pair, coaxial cable, fiber optic cable and atmosphere or space propagation).
  - Digital signal – A sequence of voltage pulses that may be transmitted over a wire medium.

  - Note – Analog signals which represent analog data and Digital signals which represent digital data are **not the only possibilities.**

INTRODUCTION



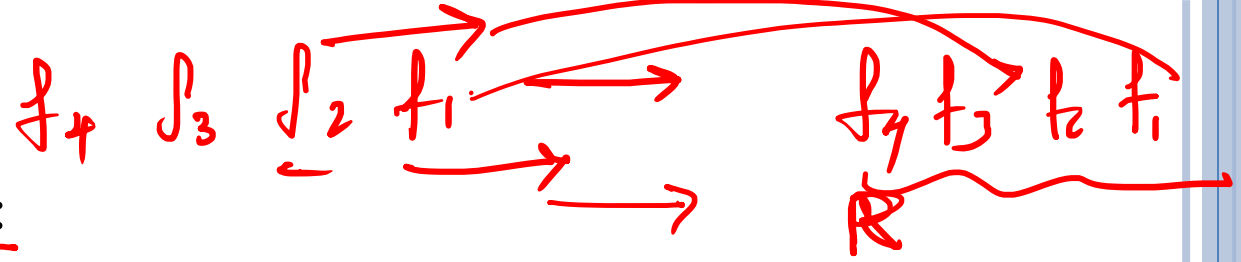
- Transmission Impairment:
  - When signals travel through the medium they tend to deteriorate. This may have many reasons as given:

1. Attenuation of a signal:
   - The reduction or loss of signal strength (power) as it transferred across a system.
   - Attenuation is an increasing function of frequency.
   - The strength of the received signal must be strong enough for detection and must be higher than the noise to be received without error.

2. Dispersion:

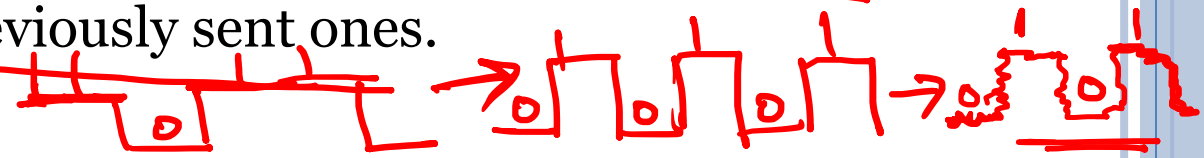
   - As signal travels through the media, it tends to spread and overlaps.
   - The amount of dispersion depends upon the frequency used.

INTRODUCTION



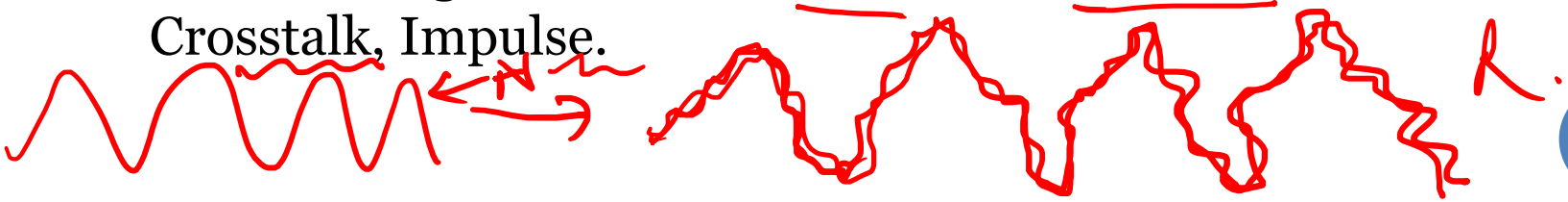
3. Delay distortion:

- Signals are sent over media with pre-defined speed and frequency.
- If the signal speed and frequency do not match, there are possibilities that signal reaches destination in arbitrary fashion.
- In digital media, this is very critical that some bits reach earlier than the previously sent ones.



4. Noise:

- Random disturbance or fluctuation in analog or digital signal is said to be Noise in signal, which may distort the actual information being carried. Noise can be characterized in one of the following class: Thermal Noise, Intermodulation, Crosstalk, Impulse.



Introduction

- Thermal Noise:
  - Heat agitates the electronic conductors of a medium which may introduce noise in the media.
  - Up to a certain level, thermal noise is unavoidable.
- Intermodulation:
  - When multiple frequencies share a medium, their interference can cause noise in the medium.
  - Intermodulation noise occurs if two different frequencies are sharing a medium and one of them has excessive strength or the component itself is not functioning properly, then the resultant frequency may not be delivered as expected.

Introduction

- Crosstalk:
  - This sort of noise happens when a foreign signal enters into the media.
  - This is because signal in one medium affects the signal of second medium.
- Impulse:
  - This noise is introduced because of irregular disturbances such as lightening, electricity, short-circuit, or faulty components.
  - Digital data is mostly affected by this sort of noise.

INTRODUCTION

- Analog and Digital Transmissions:
  - Transmissions: Communication of data by the propagation and processing of signals.
  - Both analog and digital signals may be transmitted on suitable transmission media.
- Analog transmission:
  - A means of transmitting analog signals without regard to their content (i.e., the signals may represent analog data or digital data).
  - Transmissions are attenuated over a distance.
  - Analog signal – The analog transmission system uses amplifiers to boost the energy in the signal.
  - Amplifiers boost the energy of the signal but they amplify the noise also.

INTRODUCTION

- Digital transmissions:
  - They are concerned with the digital content of the signal.
  - Attenuation is overcome without amplifying the noise.
  - With retransmission devices [repeater]at appropriate points the device recovers the digital data from the digital signal.
  - Digital signals – The digital repeaters are used to attain transmission at greater distances.
  - The digital repeater receives the digital signal, recovers the patterns of 0's and 1's and retransmits a new digital signal.

INTRODUCTION

- Digital versus Analog Transmissions:
  - Digital transmission advantages:-

1. Superior cost of digital technology
   – Low cost LSI/VLSI technology
   – Repeaters have low cost versus amplifiers costs
2. Superior quality {Data integrity}
   – Longer distances over lines with lower error rates
3. Capacity utilization
   – Economical to build high bandwidth links
   – High degree of multiplexing easier with digital techniques
4. Security & Privacy
   – Encryption techniques readily applied to digitized data
5. Integration
   – Can treat analog and digital data similarly
   – Economies of scale from integrating voice, video and data

INTRODUCTION

- Digital versus Analog Transmissions:
  - Analog transmission advantages:

    1. Digital signaling not as versatile or practical (digital impossible for satellite and microwave systems)

    2. LAN star topology limits the severity of the noise and attenuation problems.
- Channel Capacity:
  - The speed of transmission of information is said to be the channel capacity. We count it as data rate in digital world.
  - It depends on numerous factors such as:
    - **Bandwidth:** The physical limitation of underlying media.
    - **Error-rate:** Incorrect reception of information because of noise.
    - **Encoding:** The number of levels used for signaling.

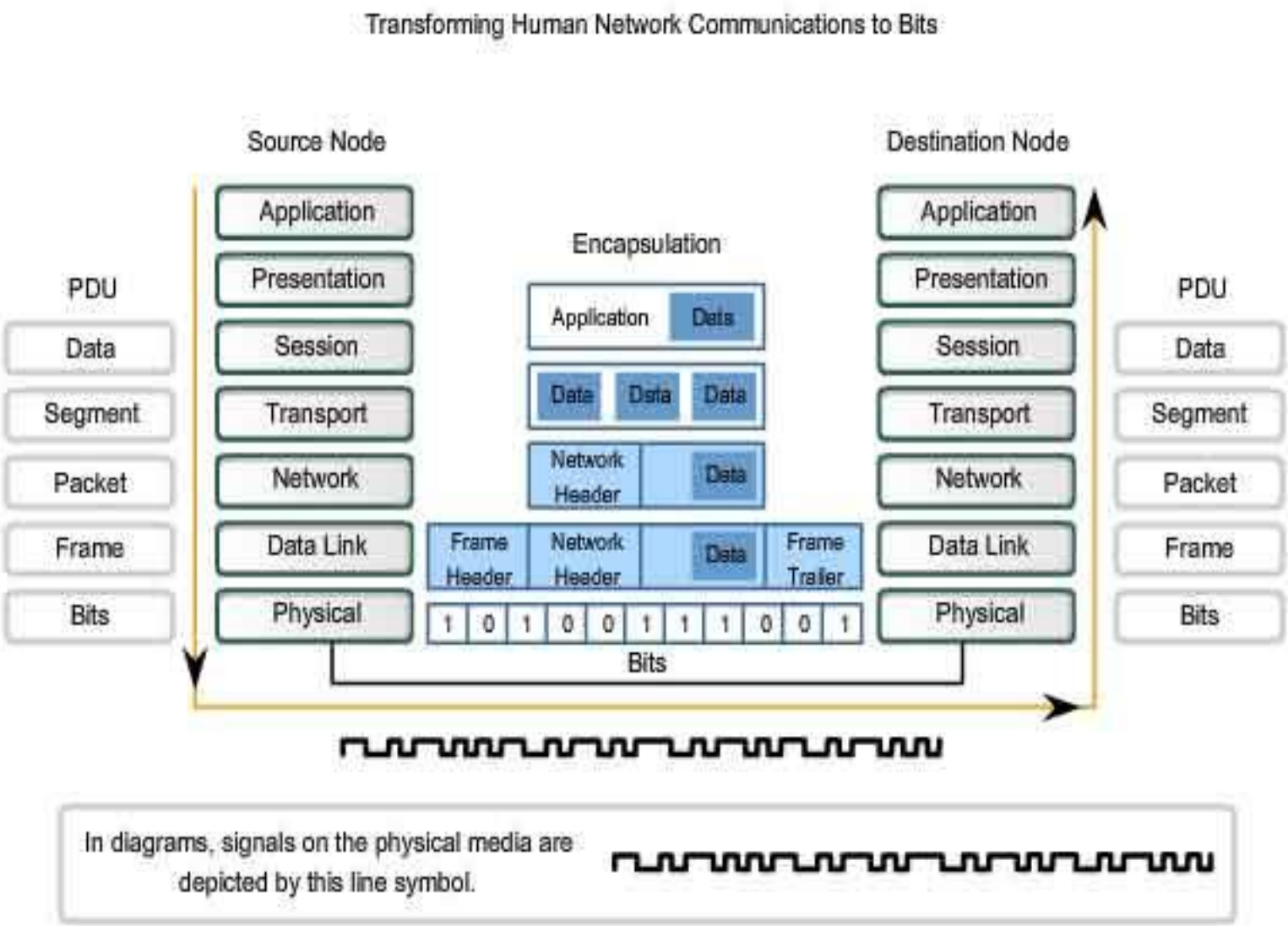
1.2 Physical layer - Purpose

- The OSI Physical layer provides the means to transport a signal across the network media the bits that make up a Data Link layer frame.
- This layer accepts a complete frame from the Data Link layer and encodes it as a series of signals that are transmitted onto the local media.
- The encoded bits that comprise a frame are received by either an end device or an intermediate device.
- The delivery of frames across the local media requires the following Physical layer elements:
  - The physical media and associated connectors.
  - A representation of bits on the media.
  - Encoding of data and control information.
  - Transmitter and receiver circuitry on the network devices.

1.2 Physical layer - Purpose

- At this stage of the communication process, the user data has been segmented by the Transport layer, placed into packets by the Network layer, and further encapsulated as frames by the Data Link layer.
- The purpose of the Physical layer is to create the electrical, optical, or microwave signal that represents the bits in each frame.
- These signals are then sent on the media one at a time.
- It is also the job of the Physical layer to retrieve these individual signals from the media, restore them to their bit representations, and pass the bits up to the Data Link layer as a complete frame.

1.2 Physical layer - Purpose



1.3 Physical Layer - Operation

- The media does not carry the frame as a single entity. The media carries signals, one at a time, to represent the bits that make up the frame.
- There are three basic forms of network media on which data is represented:
  - Copper cable
  - Fiber
  - Wireless
- The representation of the bits - that is, the type of signal - depends on the type of media.
  - For copper cable media, the signals are patterns of electrical pulses.
  - For fiber, the signals are patterns of light.
  - For wireless media, the signals are patterns of radio transmissions.

1.3 Physical Layer - Operation

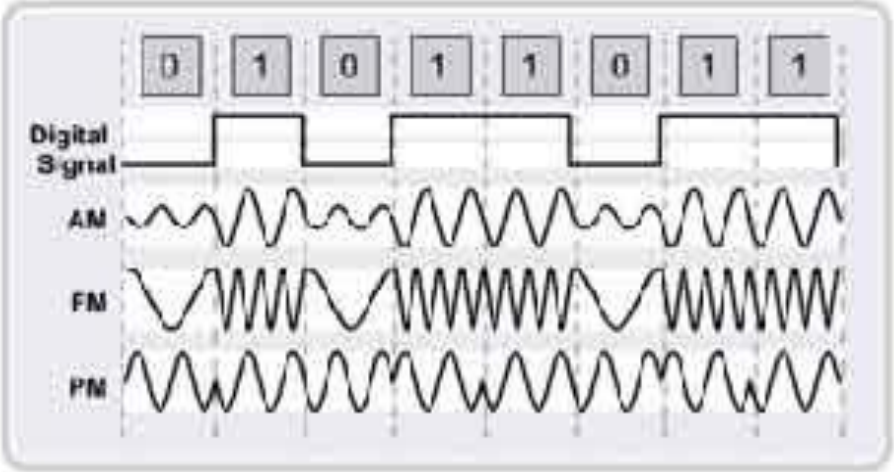
- Identifying a Frame:
  - When the Physical layer encodes the bits into the signals for a particular medium, it must also distinguish where one frame ends and the next frame begins.
  - Otherwise, the devices on the media would not recognize when a frame has been fully received.
  - In that case, the destination device would only receive a string of signals and would not be able to properly reconstruct the frame.
  - As described in the previous chapter, indicating the beginning of frame is often a function of the Data Link layer.
  - However, in many technologies, the Physical layer may add its own signals to indicate the beginning and end of the frame.
  - To enable a receiving device to clearly recognize a frame boundary, the transmitting device adds signals to designate the start and end of a frame.
  - These signals represent particular bit patterns that are only used to denote the start or end of a frame.

1.3 Physical Layer - Operation

Representations of Signals on the Physical Media

Sample electrical signals transmitted on copper cable

Representative light pulse fiber signals



Microwave (wireless) signals

1.4 Physical Layer - Standards

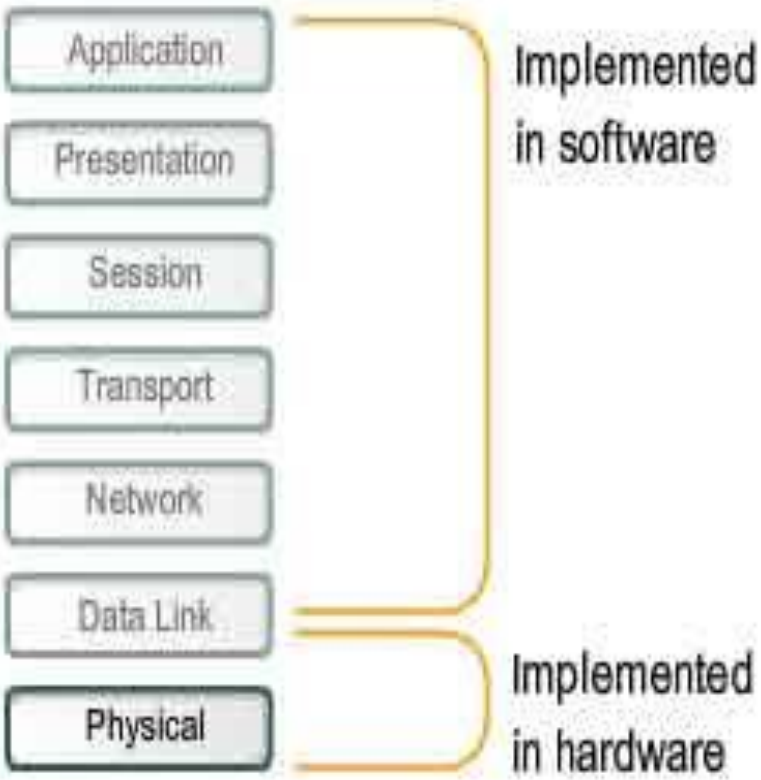
- The Physical layer consists of hardware, developed by engineers, in the form of electronic circuitry, media, and connectors.
- Therefore, it is appropriate that the standards governing this hardware are defined by the relevant electrical and communications engineering organizations.
- By comparison, the protocols and operations of the upper OSI layers are performed by software and are designed by software engineers and computer scientists.
- The services and protocols in the TCP/IP suite are defined by the Internet Engineering Task Force (IETF) in RFCs.

1.4 Physical Layer - Standards

- Similar to technologies associated with the Data Link layer, the Physical layer technologies are defined by organizations such as:
  - The International Organization for Standardization (ISO)
  - The Institute of Electrical and Electronics Engineers (IEEE)
  - The American National Standards Institute (ANSI)
  - The International Telecommunication Union (ITU)
  - The Electronics Industry Alliance/Telecommunications Industry Association (EIA/TIA)
  - National telecommunications authorities such as the Federal Communication Commission (FCC) in the USA.

1.4 Physical Layer - Standards

Comparison of Physical Layer Standards and Upper Layer Standards



TCP/IP Standards set by:
IETF

Standards set by:

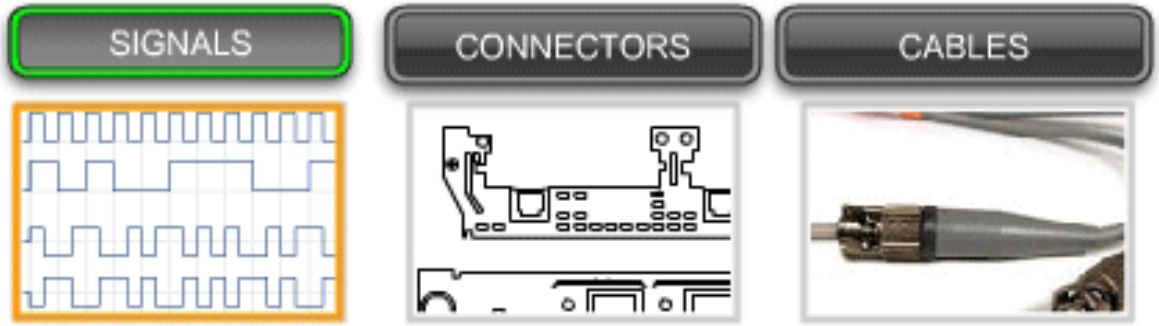
ISO	IEEE
ANSI	ITU
EIA/TIA	FCC

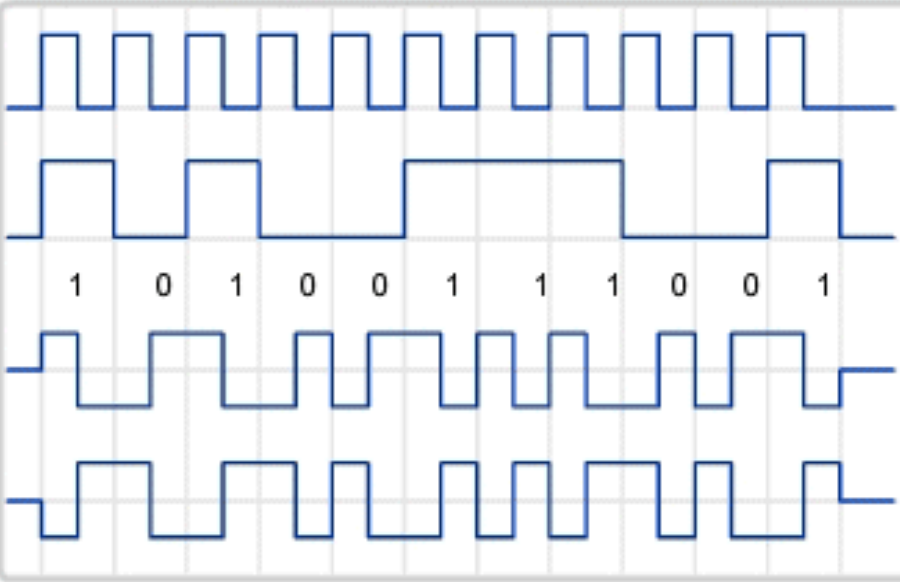
1.5 Physical Layer Technologies and Hardware

- The technologies defined by these organizations include four areas of the Physical layer standards:
  - Physical and electrical properties of the media.
  - Mechanical properties (materials, dimensions, pin-outs) of the connectors.
  - Bit representation by the signals (encoding).
  - Definition of control information signals.
- Hardware components such as network adapters (NICs), interfaces and connectors, cable materials, and cable designs are all specified in standards associated with the Physical layer.

1.5 Physical Layer Technologies and Hardware

Standards for the Physical layer specify signal, connector, and cabling requirements.





Signal standards allow a variety of devices to interoperate.

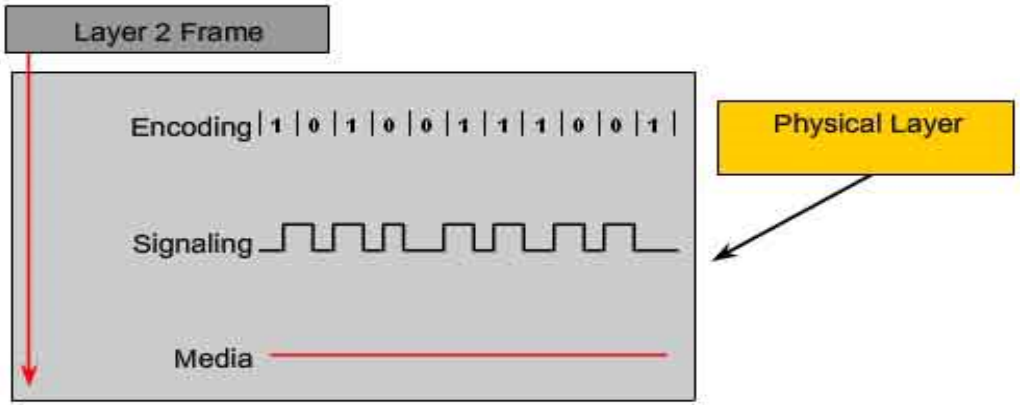
1.6 Physical layer Fundamental Principles

- The two fundamental functions of the Physical layer are:
  - Data encoding
  - Signaling
- Encoding:
  - Encoding is a method of converting a stream of data bits into a predefined code.
  - Codes are groupings of bits used to provide a predictable pattern that can be recognized by both the sender and the received.
  - Using predictable patterns helps to distinguish data bits from control bits and provide better media error detection.
  - In addition to creating codes for data, encoding methods at the Physical layer may also provide codes for control purposes such as identifying the beginning and end of a frame.
  - The transmitting host will transmit the specific pattern of bits or a code to identify the beginning and end of the frame.

1.6 Physical layer Fundamental Principles

- Signaling:
  - The Physical layer must generate the electrical, optical, or wireless signals that represent the "1" and "0" on the media.
  - The method of representing the bits is called the signaling method.
  - The Physical layer standards must define what type of signal represents a "1" and a "0".
  - This can be as simple as a change in the level of an electrical signal or optical pulse or a more complex signaling method.

Physical Layer Fundamental Principles



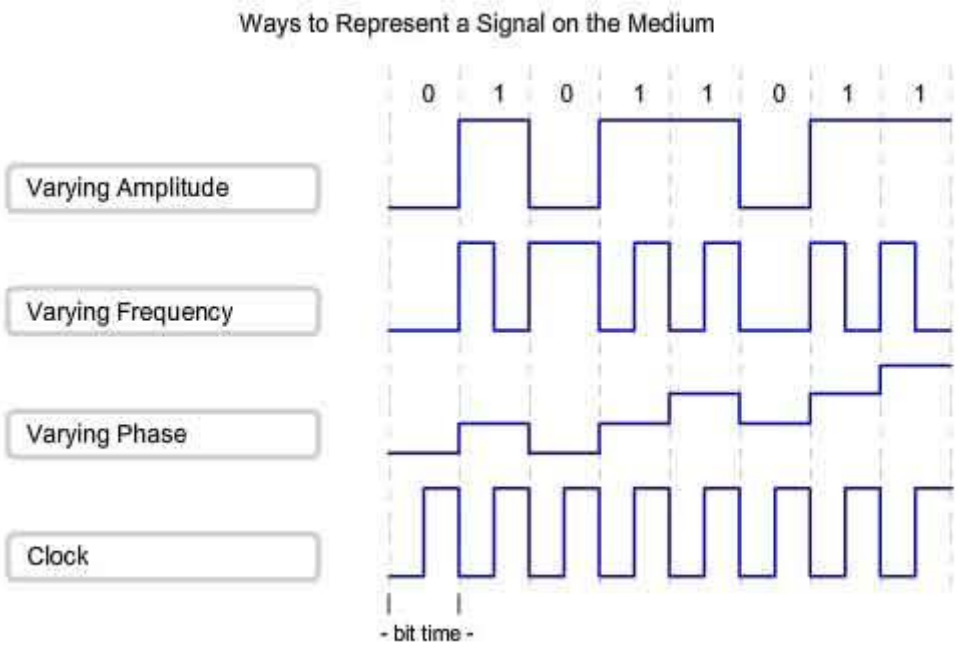
1.6 Physical layer Fundamental Principles

- Physical Signaling and Encoding: Representing Bits.
  - Signaling Bits for the Media:
    - The Physical layer represents each of the bits in the frame as a signal.
    - Each signal placed onto the media has a specific amount of time to occupy the media.
    - This is referred to as its bit time. Signals are processed by the receiving device and returned to its representation as bits.
    - At the Physical layer of the receiving node, the signals are converted back into bits.
    - The bits are then examined for the start of frame and end of frame bit patterns to determine that a complete frame has been received.
    - The Physical layer then delivers all the bits of a frame to the Data Link layer. Successful delivery of the bits requires some method of synchronization between transmitter and receiver.
    - The synchronization is accomplished by the use of a clock. In LANs, each end of the transmission maintains its own clock.

1.6 Physical layer Fundamental Principles

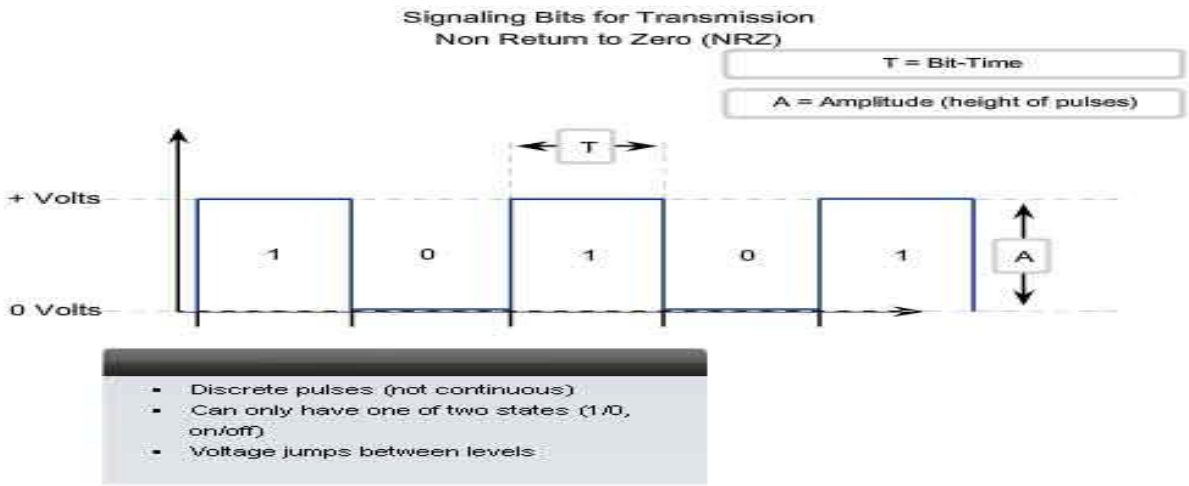
- Signaling Methods:
  - Bits are represented on the medium by changing one or more of the following characteristics of a signal:

- Amplitude
- Frequency
- Phase



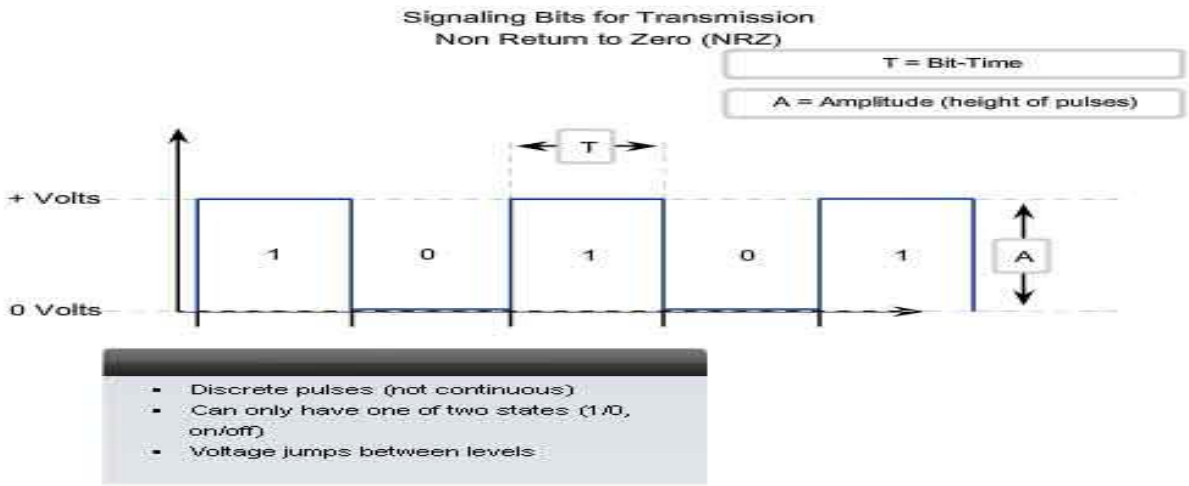
1.6 Physical layer Fundamental Principles

- NRZ Signaling: Non Return to Zero (NRZ).
  - In NRZ, the bit stream is transmitted as a series of voltage values.
  - A low voltage value represents a logical 0 and a high voltage value represents a logical 1.
  - NRZ signaling uses bandwidth inefficiently and is susceptible to electromagnetic interference.
  - Additionally, the boundaries between individual bits can be lost when long strings of 1s or 0s are transmitted consecutively.
  - In that case, no voltage transitions are detectable on the media.



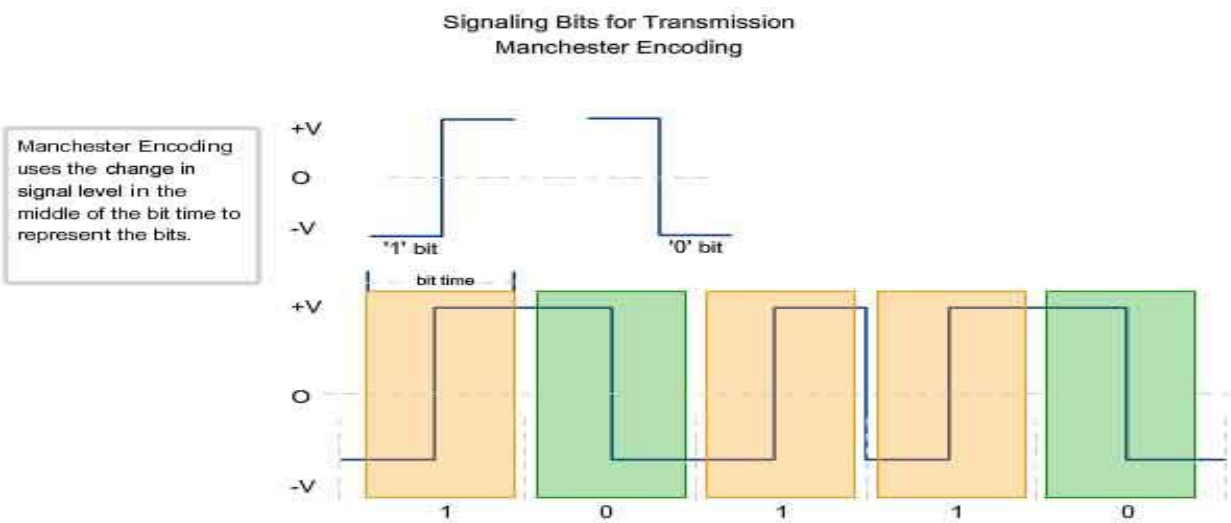
1.6 Physical layer Fundamental Principles

- NRZ Signaling: Non Return to Zero (NRZ).
  - In NRZ, the bit stream is transmitted as a series of voltage values.
  - A low voltage value represents a logical 0 and a high voltage value represents a logical 1.
  - NRZ signaling uses bandwidth inefficiently and is susceptible to electromagnetic interference.
  - Additionally, the boundaries between individual bits can be lost when long strings of 1s or 0s are transmitted consecutively.
  - In that case, no voltage transitions are detectable on the media.



1.6 Physical layer Fundamental Principles

- Manchester Encoding:
  - Instead of representing bits as pulses of simple voltage values, in the Manchester Encoding scheme, bit values are represented as voltage transitions.
  - For example, a transition from a low voltage to a high voltage represents a bit value of 1.
  - A transition from a high voltage to a low voltage represents a bit value of 0.

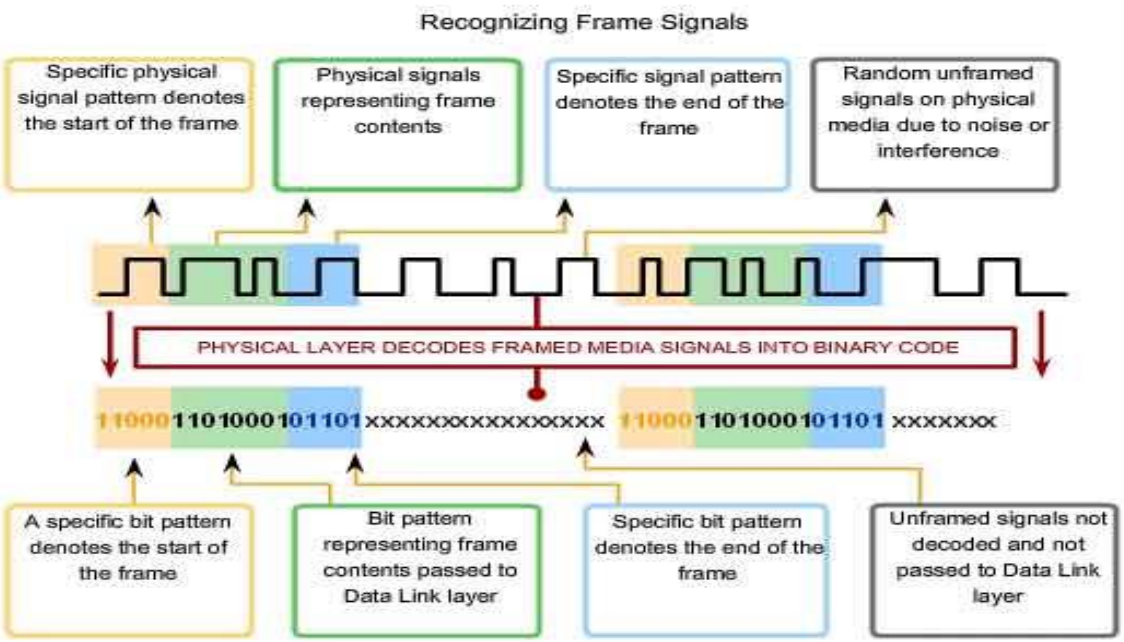


1.6 Physical layer Fundamental Principles

- Encoding - Grouping Bits:
  - In the prior section, we describe the signaling process as how bits are represented on physical media.
  - In this section, we use of the word encoding to represent the symbolic grouping of bits prior to being presented to the media.
  - By using an encoding step before the signals are placed on the media, we improve the efficiency at higher speed data transmission.
  - As we use higher speeds on the media, we have the possibility that data will be corrupted. By using the coding groups, we can detect errors more efficiently.
  - Additionally, as the demand for data speeds increase, we seek ways to represent more data across the media, by transmitting fewer bits.
  - Coding groups provide a method of making this data representation.

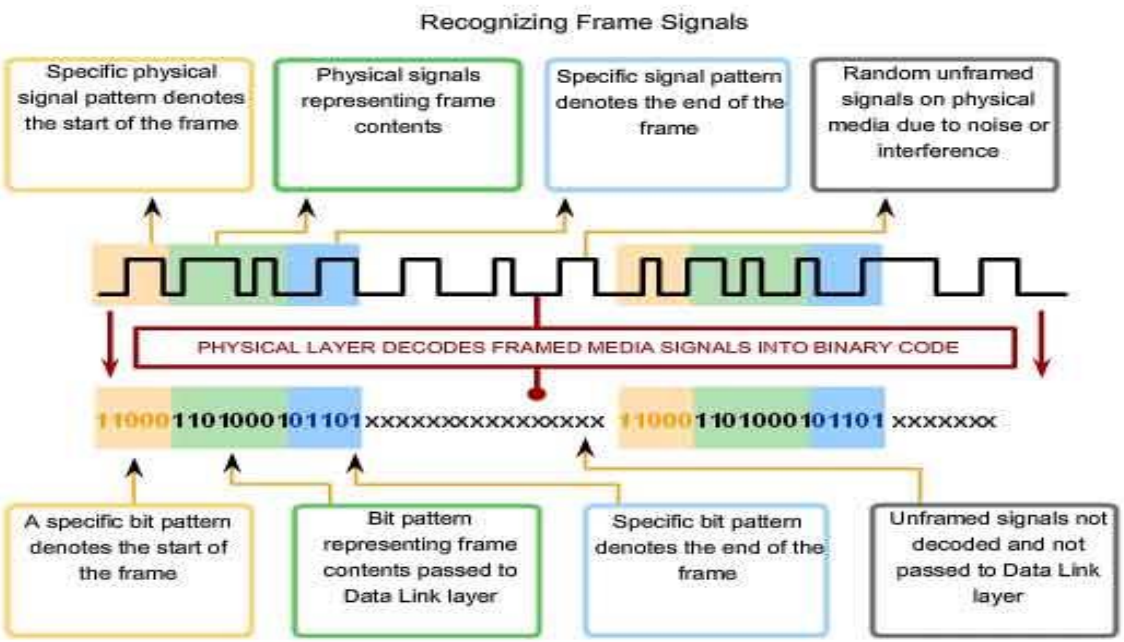
1.6 Physical layer Fundamental Principles

- Signal Patterns:
  - One way to provide frame detection is to begin each frame with a pattern of signals representing bits that the Physical layer recognizes as denoting the start of a frame.
  - Another pattern of bits will signal the end of the frame. Signal bits not framed in this manner are ignored by the Physical layer standard being used.



1.7 Physical media

- Signal Patterns:
  - One way to provide frame detection is to begin each frame with a pattern of signals representing bits that the Physical layer recognizes as denoting the start of a frame.
  - Another pattern of bits will signal the end of the frame. Signal bits not framed in this manner are ignored by the Physical layer standard being used.



1.8 SWITCHING

- Switching techniques:
  - In large networks, there can be multiple paths from sender to receiver. The switching technique will decide the best route for data transmission.
  - Switching technique is used to connect the systems for making one-to-one communication.
  - The different switching techniques used in networking are:

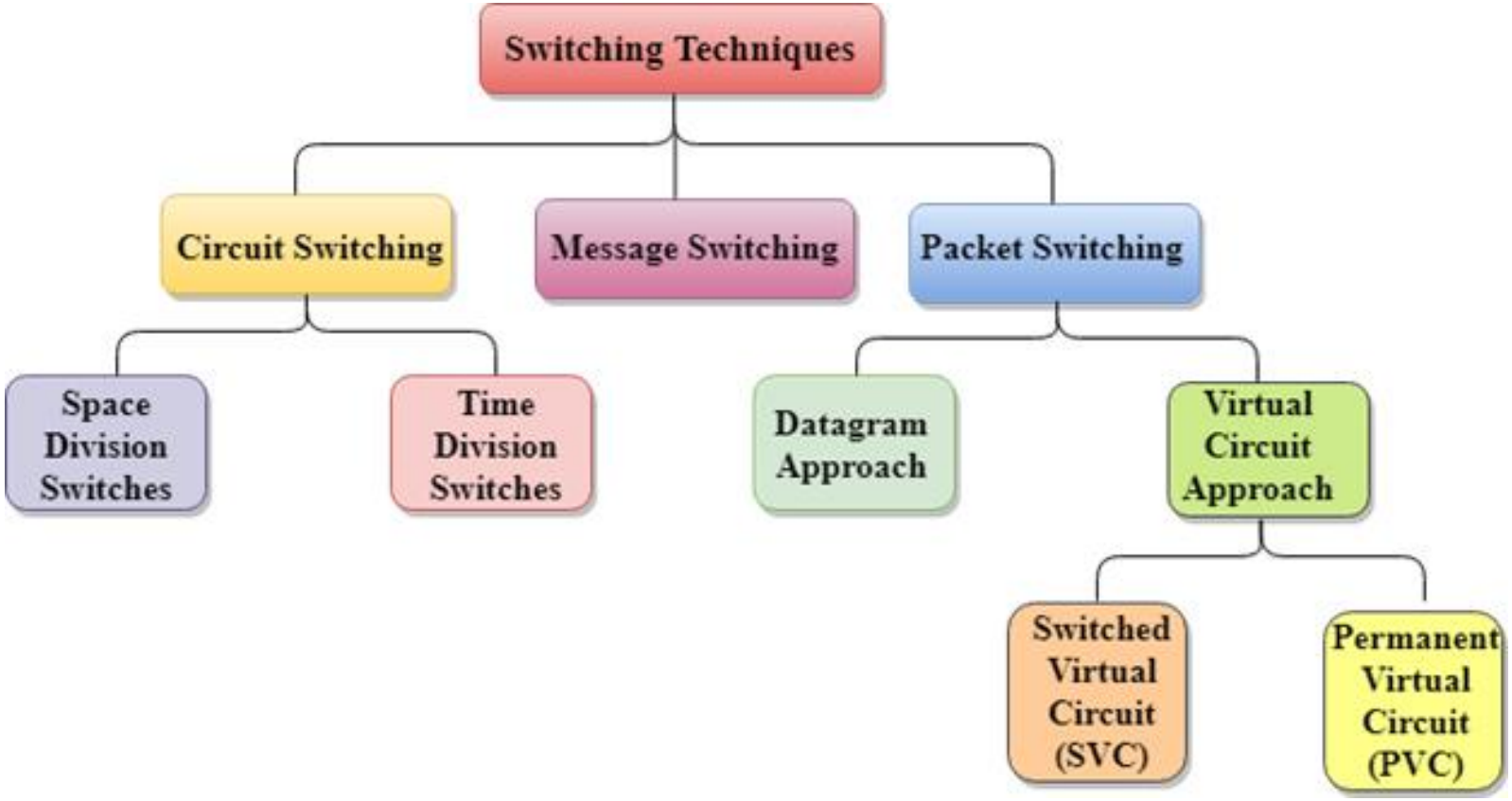
    (1) Circuit switching,

    (2) Packet switching, and

    (3) Message switching.

1.8 SWITCHING

- Classification Of Switching Techniques:

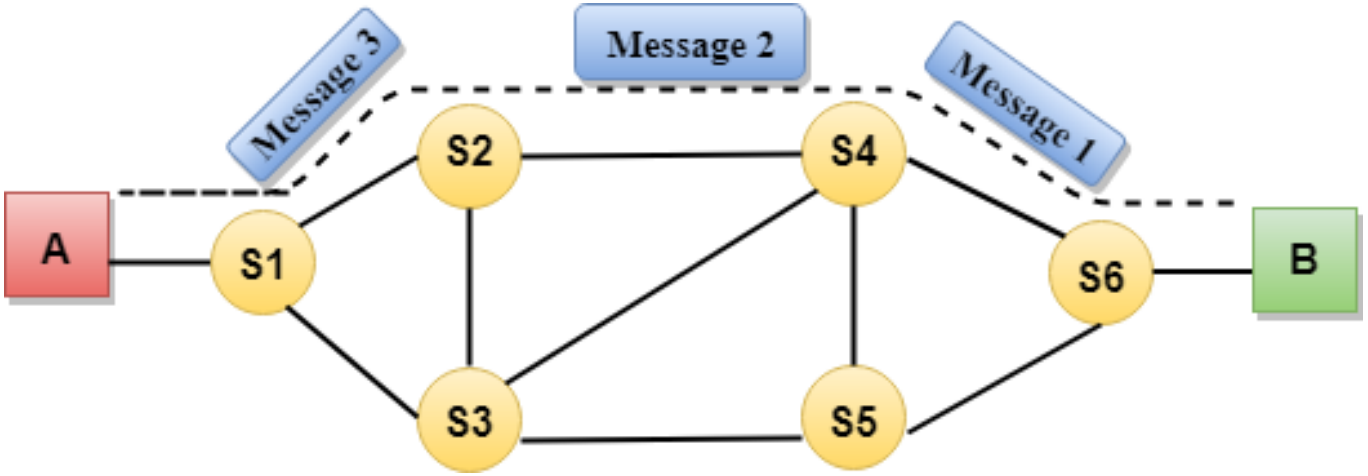


1.8 SWITCHING

- Circuit Switching:
  - Circuit switching is a switching technique that establishes a dedicated path between sender and receiver.
  - In the Circuit Switching Technique, once the connection is established then the dedicated path will remain to exist until the connection is terminated.
  - Circuit switching in a network operates in a similar way as the telephone works.
  - A complete end-to-end path must exist before the communication takes place.
  - In case of circuit switching technique, when any user wants to send the data, voice, video, a request signal is sent to the receiver then the receiver sends back the acknowledgment to ensure the availability of the dedicated path. After receiving the acknowledgment, dedicated path transfers the data.

1.8 SWITCHING

- Circuit Switching:
  - Circuit switching is used in public telephone network. It is used for voice transmission.
  - Fixed data can be transferred at a time in circuit switching technology.
  - Communication through circuit switching has 3 phases:
    - Circuit establishment
    - Data transfer
    - Circuit Disconnect



1.8 SWITCHING

- Circuit Switching:
  - Circuit Switching can uses technology:
    - Space-division circuit switching.
- Space-division circuit switching:
  - Space Division Switching is a circuit switching technology in which a single transmission path is accomplished in a switch by using a physically separate set of cross points.
  - Space Division Switching can be achieved by using crossbar switch.
  - The crossbar switch has intersection points known as cross points.
  - A crossbar switch is a metallic cross point or semiconductor gate that can be enabled or disabled by a control unit.
  - The Crossbar switch is made by using the semiconductor.
  - Space Division Switching has high speed, high capacity, and non-blocking switches.

1.8 SWITCHING

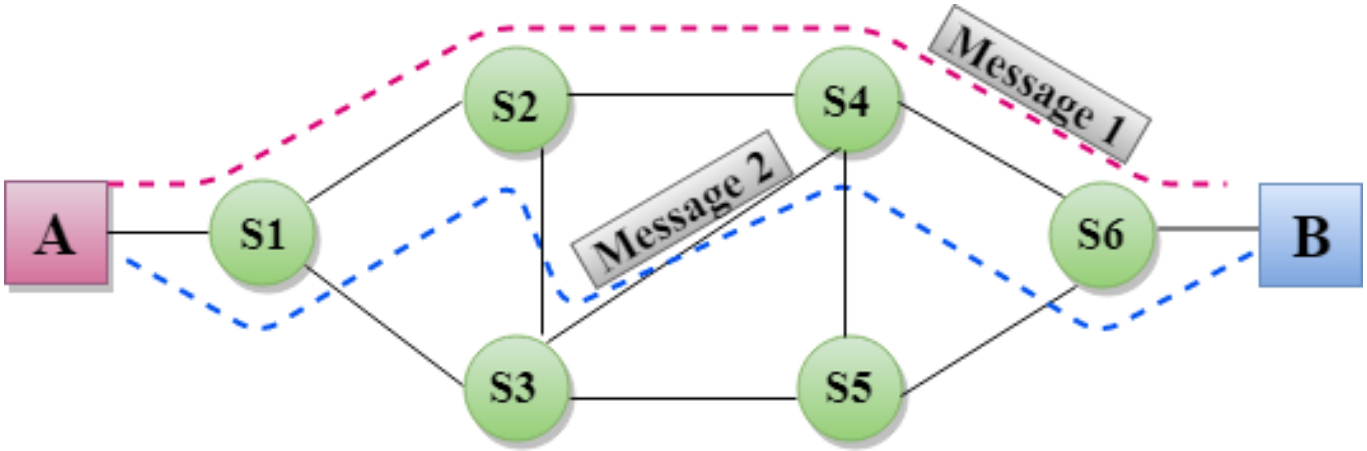
- Advantages Of Circuit Switching:
  - In the case of Circuit Switching technique, the communication channel is dedicated.
  - It has fixed bandwidth.
- Disadvantages Of Circuit Switching:
  - Once the dedicated path is established, the only delay occurs in the speed of data transmission.
  - It takes a long time to establish a connection approx 10 seconds during which no data can be transmitted.
  - It is more expensive than other switching techniques as a dedicated path is required for each connection.
  - It is inefficient to use because once the path is established and no data is transferred, then the capacity of the path is wasted.
  - In this case, the connection is dedicated therefore no other data can be transferred even if the channel is free.

1.8 SWITCHING

- Message Switching:
  - Message Switching is a switching technique in which a message is transferred as a complete unit and routed through intermediate nodes at which it is stored and forwarded.
  - In Message Switching technique, there is no establishment of a dedicated path between the sender and receiver.
  - The destination address is appended to the message. Message Switching provides a dynamic routing as the message is routed through the intermediate nodes based on the information available in the message.
  - Message switches are programmed in such a way so that they can provide the most efficient routes.

1.8 SWITCHING

- Message Switching:
  - Each and every node stores the entire message and then forward it to the next node. This type of network is known as store and forward network.
  - Message switching treats each message as an independent entity.



1.8 SWITCHING

- Advantages Of Message Switching:
  - Data channels are shared among the communicating devices that improve the efficiency of using available bandwidth.
  - Traffic congestion can be reduced because the message is temporarily stored in the nodes.
  - Message priority can be used to manage the network.
  - The size of the message which is sent over the network can be varied. Therefore, it supports the data of unlimited size.
- Disadvantages Of Message Switching:
  - The message switches must be equipped with sufficient storage to enable them to store the messages until the message is forwarded.
  - The Long delay can occur due to the storing and forwarding facility provided by the message switching technique.

1.8 SWITCHING

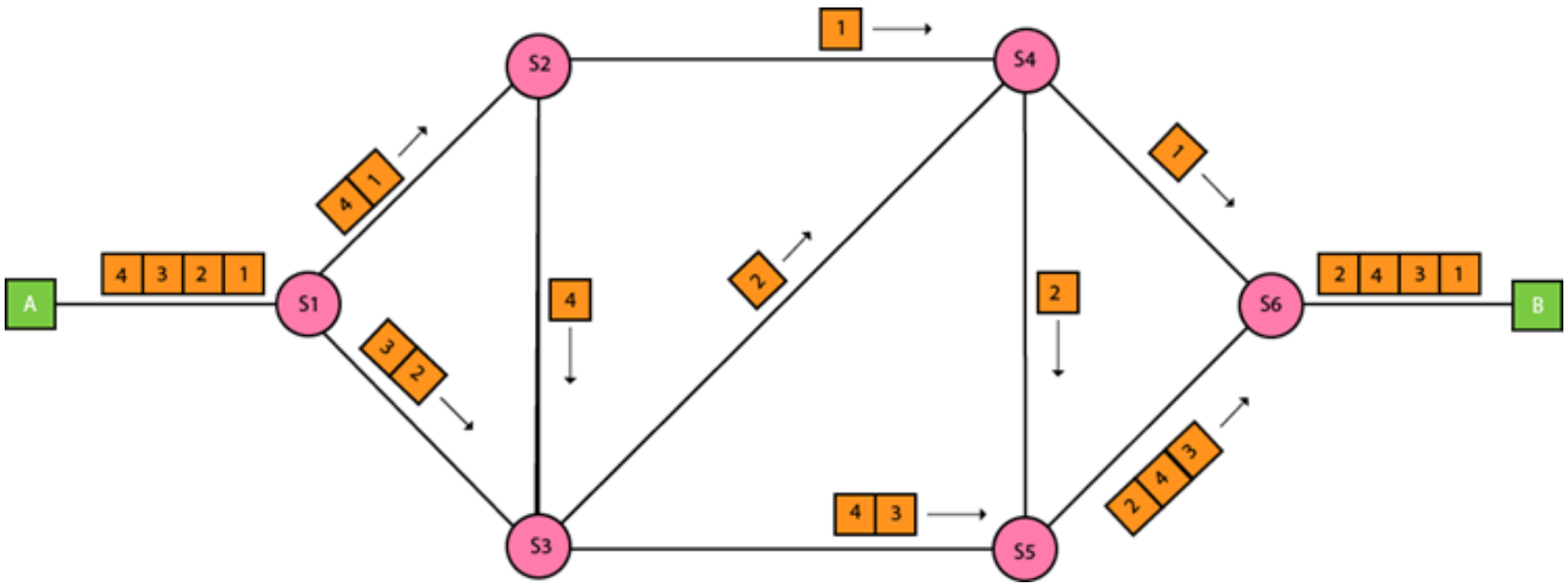
- Packet Switching:
  - The packet switching is a switching technique in which the message is sent in one go, but it is divided into smaller pieces, and they are sent individually.
  - The message splits into smaller pieces known as packets and packets are given a unique number to identify their order at the receiving end.
  - Every packet contains some information in its headers such as source address, destination address and sequence number.
  - Packets will travel across the network, taking the shortest path as possible.
  - All the packets are reassembled at the receiving end in correct order.
  - If any packet is missing or corrupted, then the message will be sent to resend the message.
  - If the correct order of the packets is reached, then the acknowledgment message will be sent.

1.8 SWITCHING

- Approaches Of Packet Switching:
  - There are two approaches to Packet Switching:
    - Datagram Packet switching
    - Virtual Circuit Switching
- Datagram Packet switching:
  - It is a packet switching technology in which packet is known as a datagram, is considered as an independent entity.
  - Each packet contains the information about the destination and switch uses this information to forward the packet to the correct destination.
  - The packets are reassembled at the receiving end in correct order.

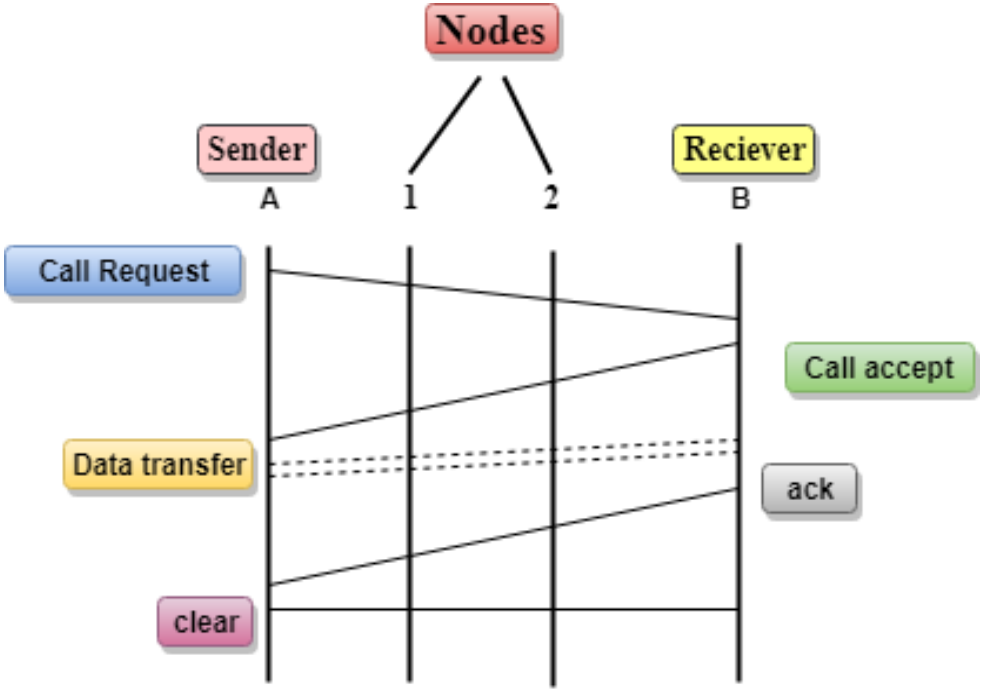
1.8 SWITCHING

- In Datagram Packet Switching technique, the path is not fixed.
- Intermediate nodes take the routing decisions to forward the packets.
- Datagram Packet Switching is also known as connectionless switching.



1.8 SWITCHING

- Virtual Circuit Switching
  - Virtual Circuit Switching is also known as connection-oriented switching.
  - In the case of Virtual circuit switching, a preplanned route is established before the messages are sent.
  - Call request and call accept packets are used to establish the connection between sender and receiver.
  - In this case, the path is fixed for the duration of a logical connection.



1.8 SWITCHING

- Let's understand the concept of virtual circuit switching through a diagram:
  - In the diagram, A and B are the sender and receiver respectively. 1 and 2 are the nodes.
  - Call request and call accept packets are used to establish a connection between the sender and receiver.
  - When a route is established, data will be transferred.
  - After transmission of data, an acknowledgment signal is sent by the receiver that the message has been received.
  - If the user wants to terminate the connection, a clear signal is sent for the termination.

1.8 SWITCHING

- Advantages Of Packet Switching:
  - Cost-effective: In packet switching technique, switching devices do not require massive secondary storage to store the packets, so cost is minimized to some extent. Therefore, we can say that the packet switching technique is a cost-effective technique.
  - Reliable: If any node is busy, then the packets can be rerouted. This ensures that the Packet Switching technique provides reliable communication.
  - Efficient: Packet Switching is an efficient technique. It does not require any established path prior to the transmission, and many users can use the same communication channel simultaneously, hence makes use of available bandwidth very efficiently.
- Disadvantages Of Packet Switching:
  - Packet Switching technique cannot be implemented in those applications that require low delay and high-quality services.
  - The protocols used in a packet switching technique are very complex and requires high implementation cost.
  - If the network is overloaded or corrupted, then it requires retransmission of lost packets. It can also lead to the loss of critical information if errors are nor recovered.

Thank You

Data Communications and Networking Fourth Edition

Forouzan

Chapter 6

Bandwidth Utilization: Multiplexing and Spreading

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Note

Bandwidth utilization is the wise use of available bandwidth to achieve specific goals.

Efficiency can be achieved by multiplexing; privacy and anti-jamming can be achieved by spreading.

6-1 MULTIPLEXING

Whenever the bandwidth of a medium linking two devices is greater than the bandwidth needs of the devices, the link can be shared. Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link. As data and telecommunications use increases, so does traffic.

Topics discussed in this section:

Frequency-Division Multiplexing
Wavelength-Division Multiplexing
Synchronous Time-Division Multiplexing
Statistical Time-Division Multiplexing

Figure 6.1 *Dividing a link into channels*

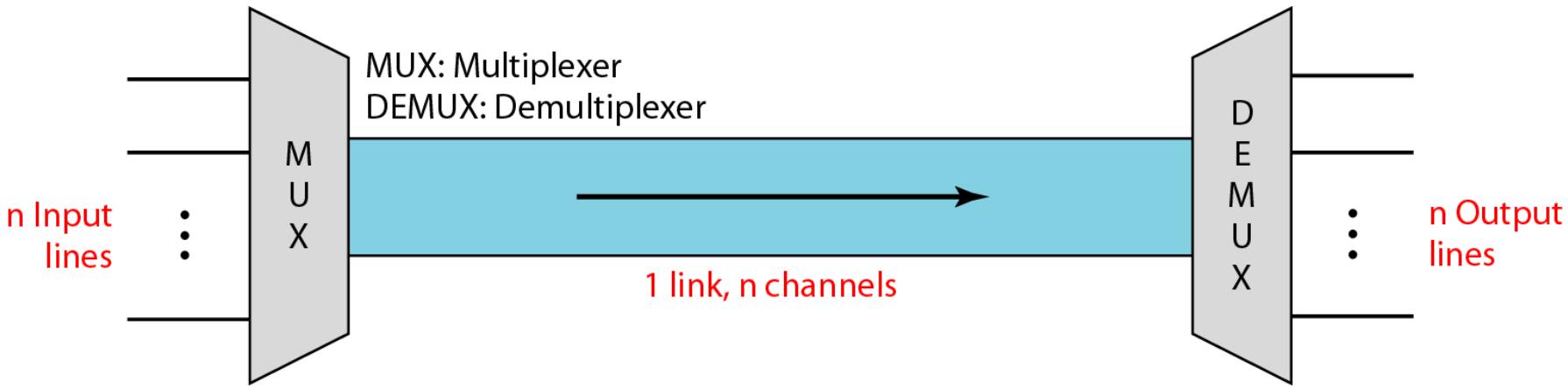


Figure 6.2 *Categories of multiplexing*

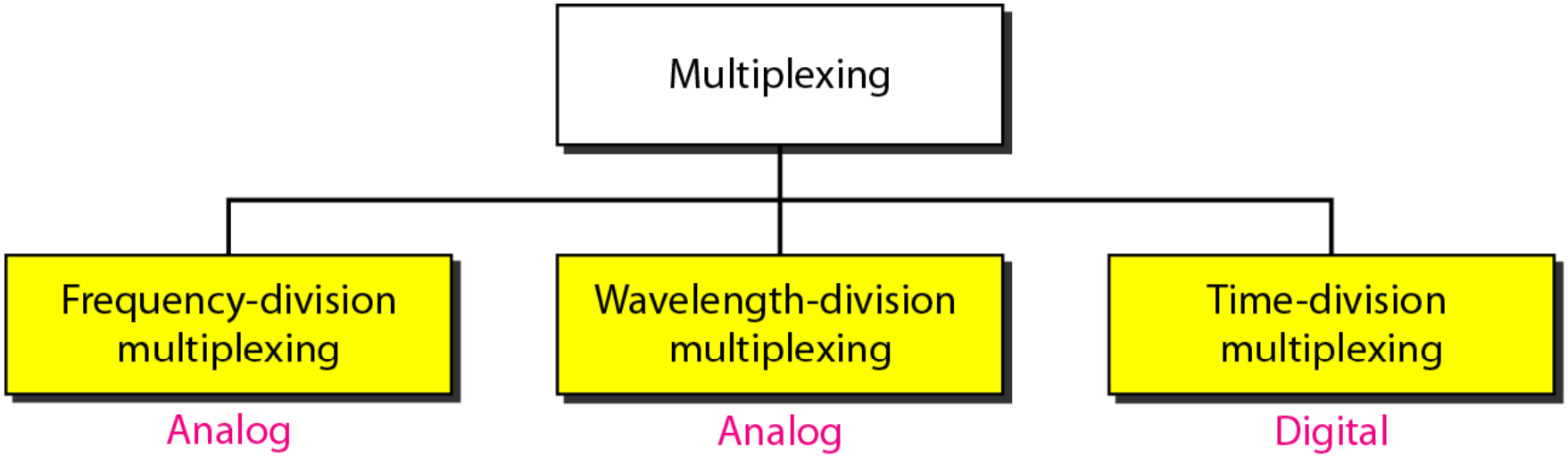


Figure 6.3 *Frequency-division multiplexing*

Note

FDM is an analog multiplexing technique that combines analog signals.

Figure 6.4 *FDM process*

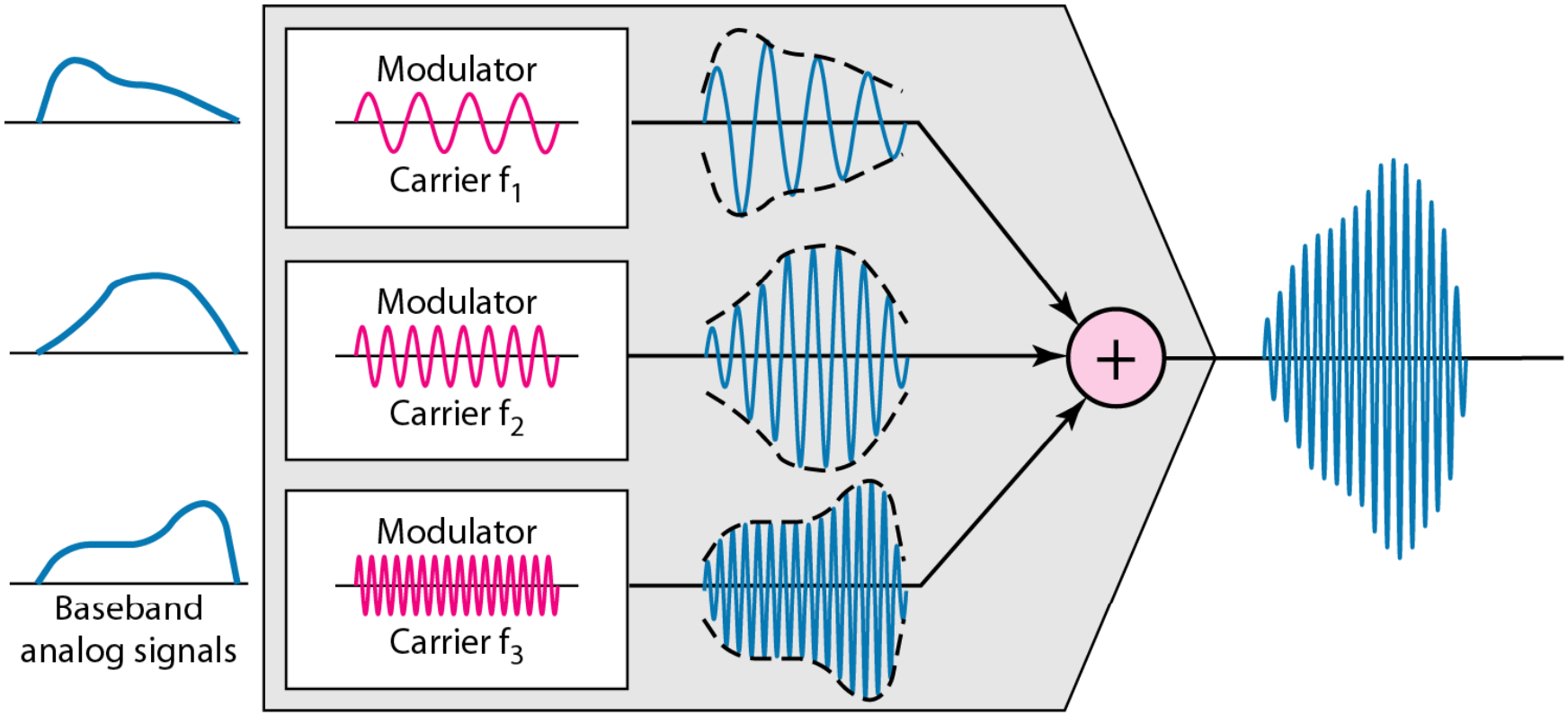
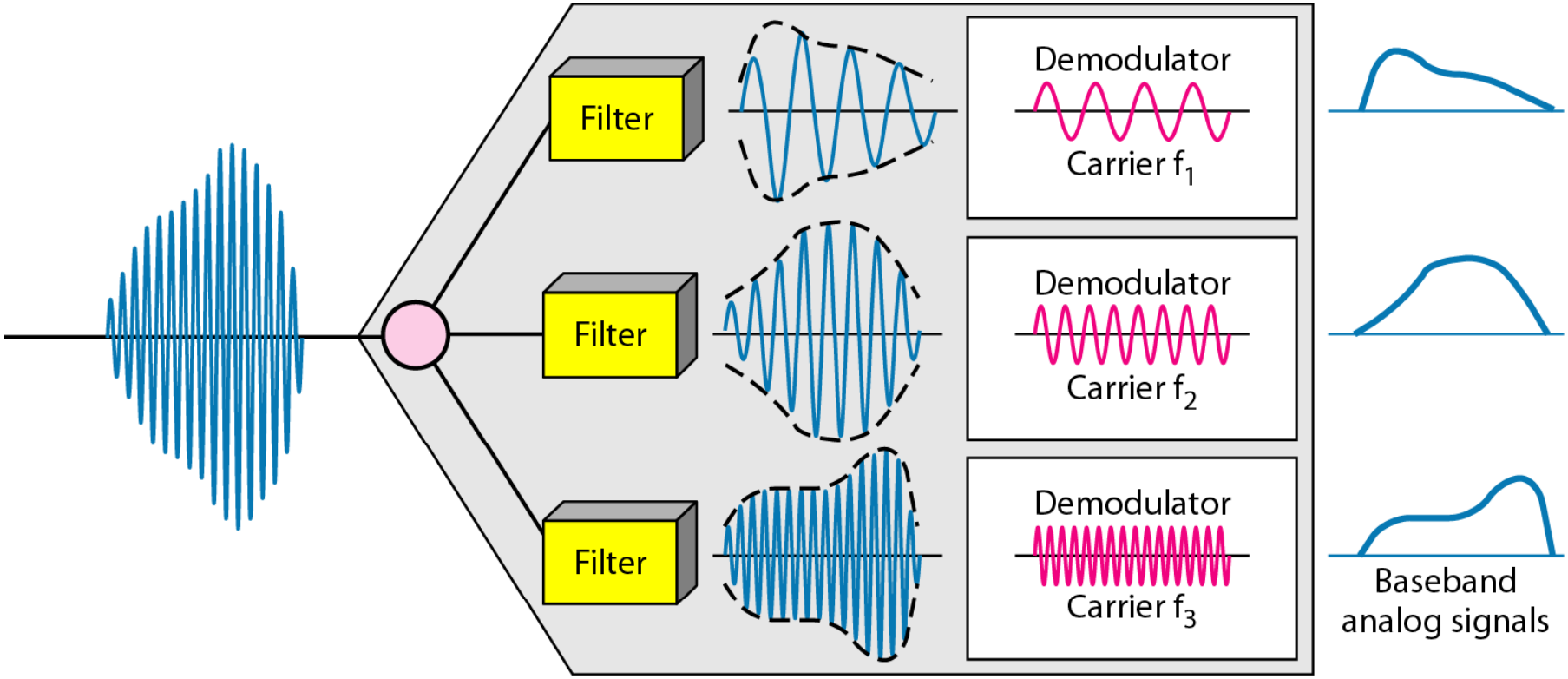


Figure 6.5 *FDM demultiplexing example*



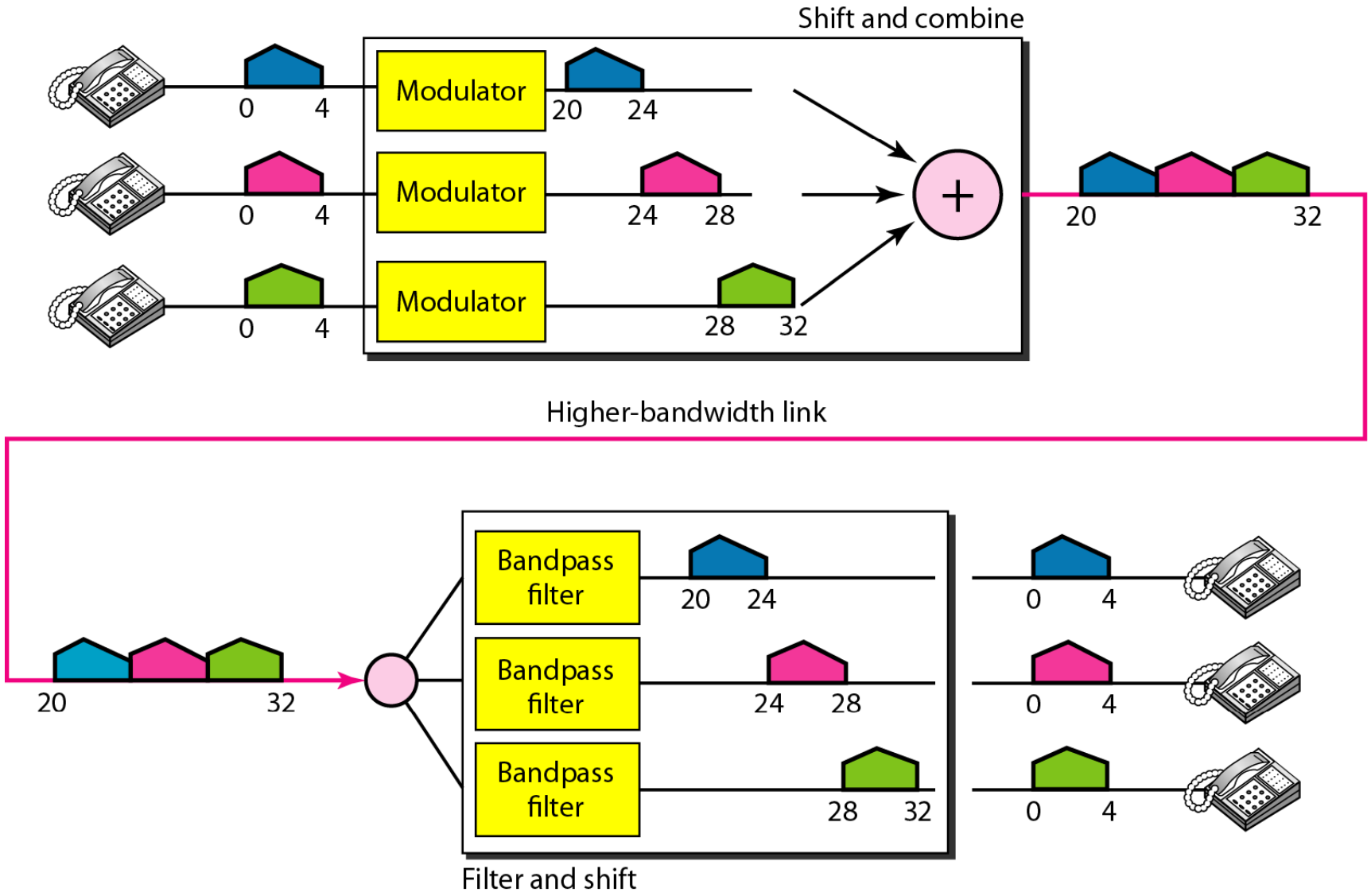
Example 6.1

Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands.

Solution

We shift (modulate) each of the three voice channels to a different bandwidth, as shown in Figure 6.6. We use the 20- to 24-kHz bandwidth for the first channel, the 24- to 28-kHz bandwidth for the second channel, and the 28- to 32-kHz bandwidth for the third one. Then we combine them as shown in Figure 6.6.

Figure 6.6 *Example 6.1*



Example 6.2

Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

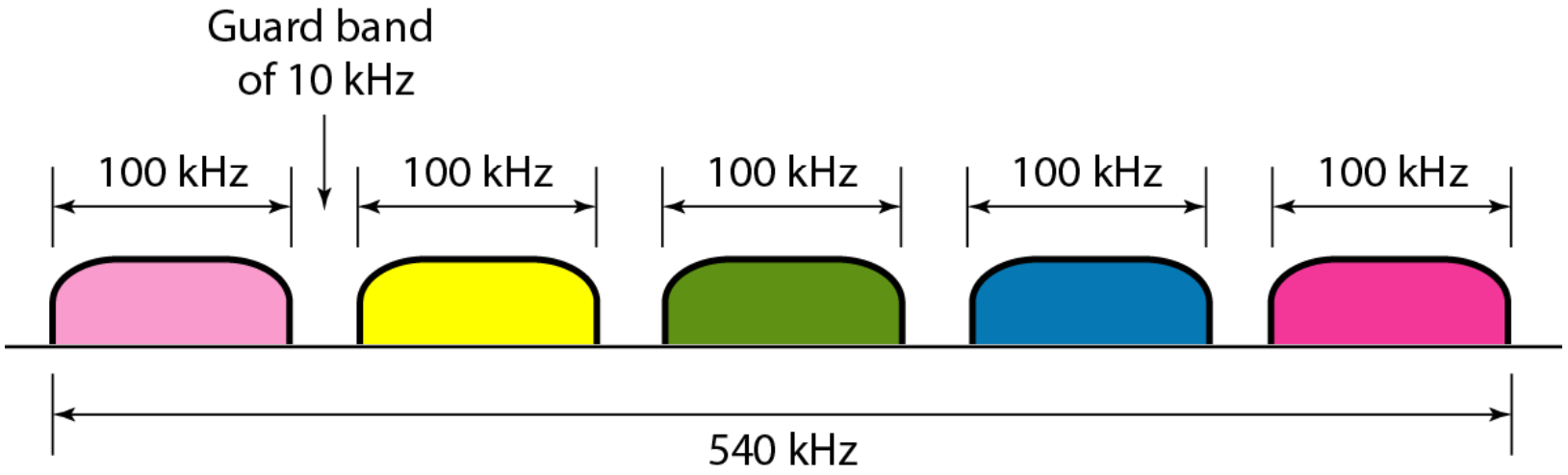
Solution

For five channels, we need at least four guard bands. This means that the required bandwidth is at least

$$5 \times 100 + 4 \times 10 = 540 \text{ kHz},$$

as shown in Figure 6.7.

Figure 6.7 *Example 6.2*



Example 6.3

Four data channels (digital), each transmitting at 1 Mbps, use a satellite channel of 1 MHz. Design an appropriate configuration, using FDM.

Solution

The satellite channel is analog. We divide it into four channels, each channel having a 250-kHz bandwidth. Each digital channel of 1 Mbps is modulated such that each 4 bits is modulated to 1 Hz. One solution is 16-QAM modulation. Figure 6.8 shows one possible configuration.

Figure 6.8 *Example 6.3*

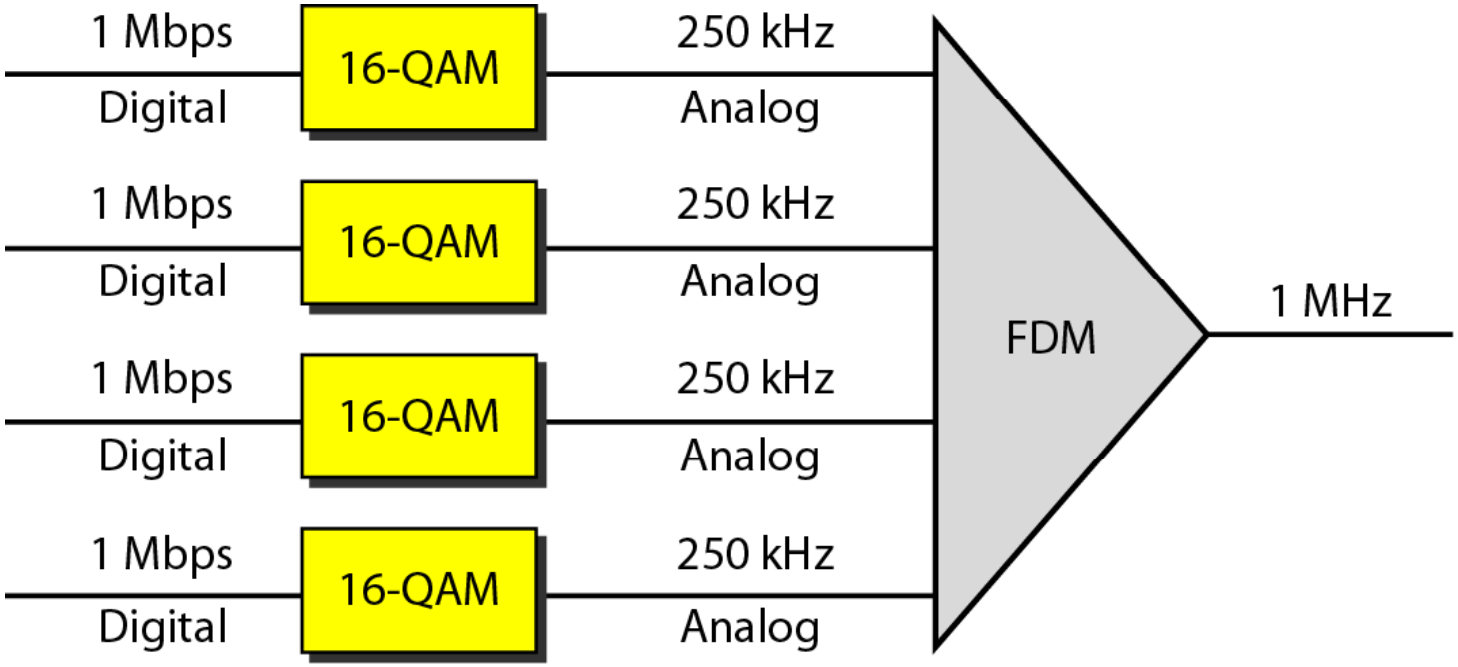
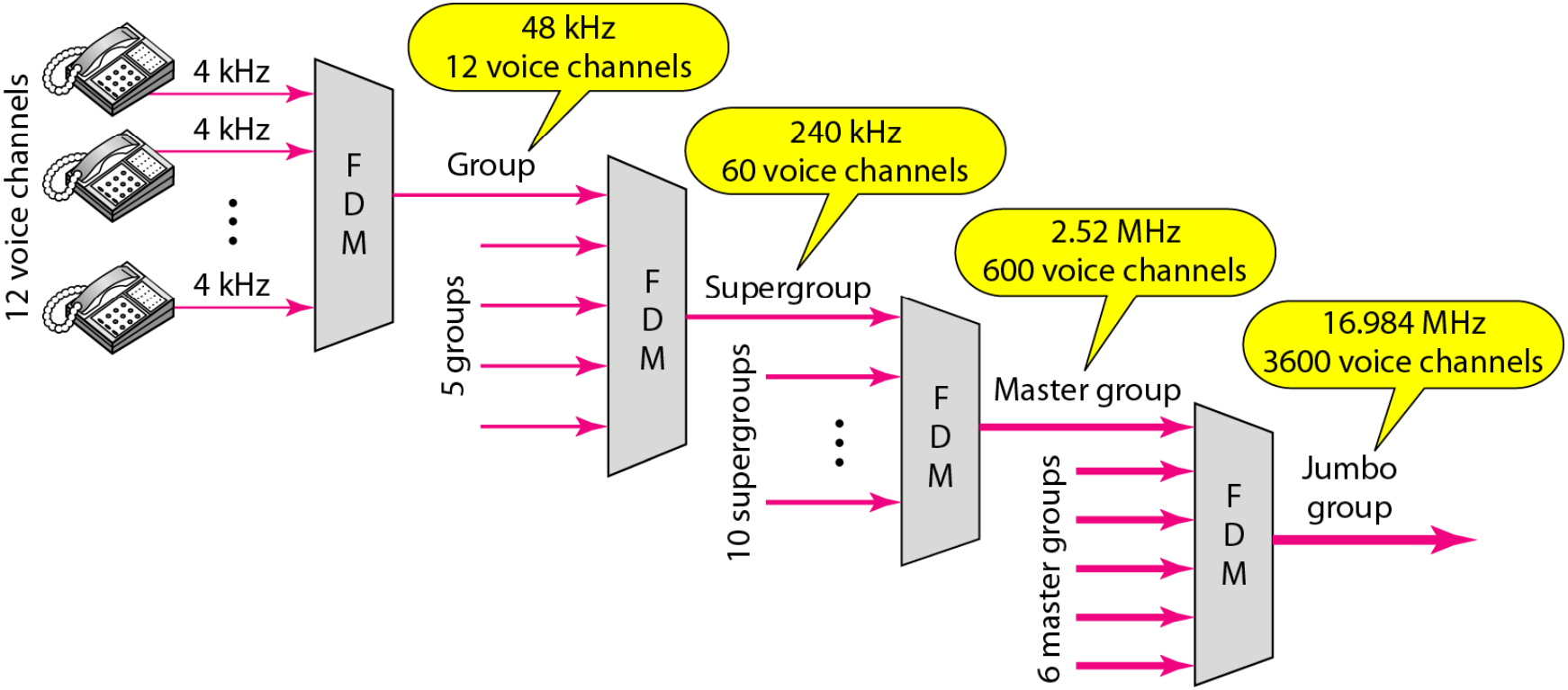


Figure 6.9 *Analog hierarchy*



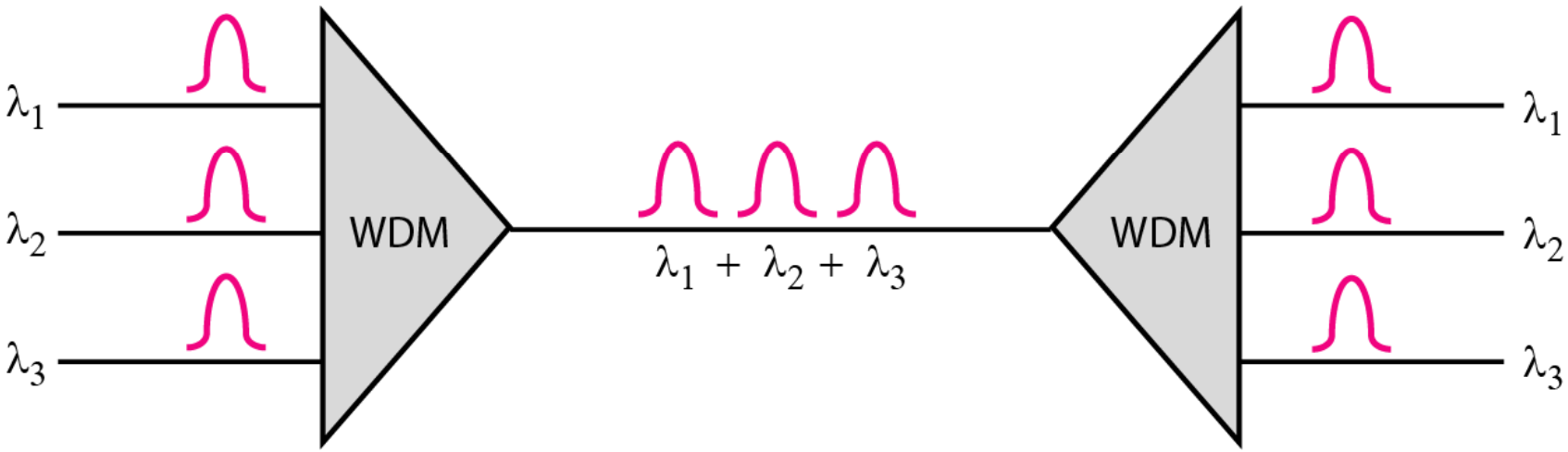
Example 6.4

The Advanced Mobile Phone System (AMPS) uses two bands. The first band of 824 to 849 MHz is used for sending, and 869 to 894 MHz is used for receiving. Each user has a bandwidth of 30 kHz in each direction. How many people can use their cellular phones simultaneously?

Solution

Each band is 25 MHz. If we divide 25 MHz by 30 kHz, we get 833.33. In reality, the band is divided into 832 channels. Of these, 42 channels are used for control, which means only 790 channels are available for cellular phone users.

Figure 6.10 *Wavelength-division multiplexing*



Note

WDM is an analog multiplexing technique to combine optical signals.

Figure 6.11 ***Prisms in wavelength-division multiplexing and demultiplexing***

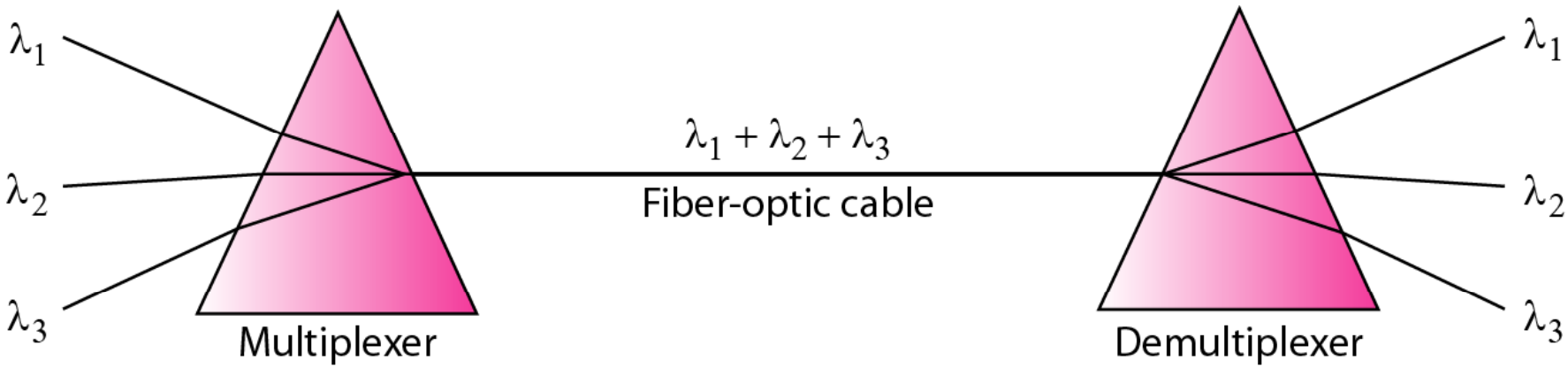
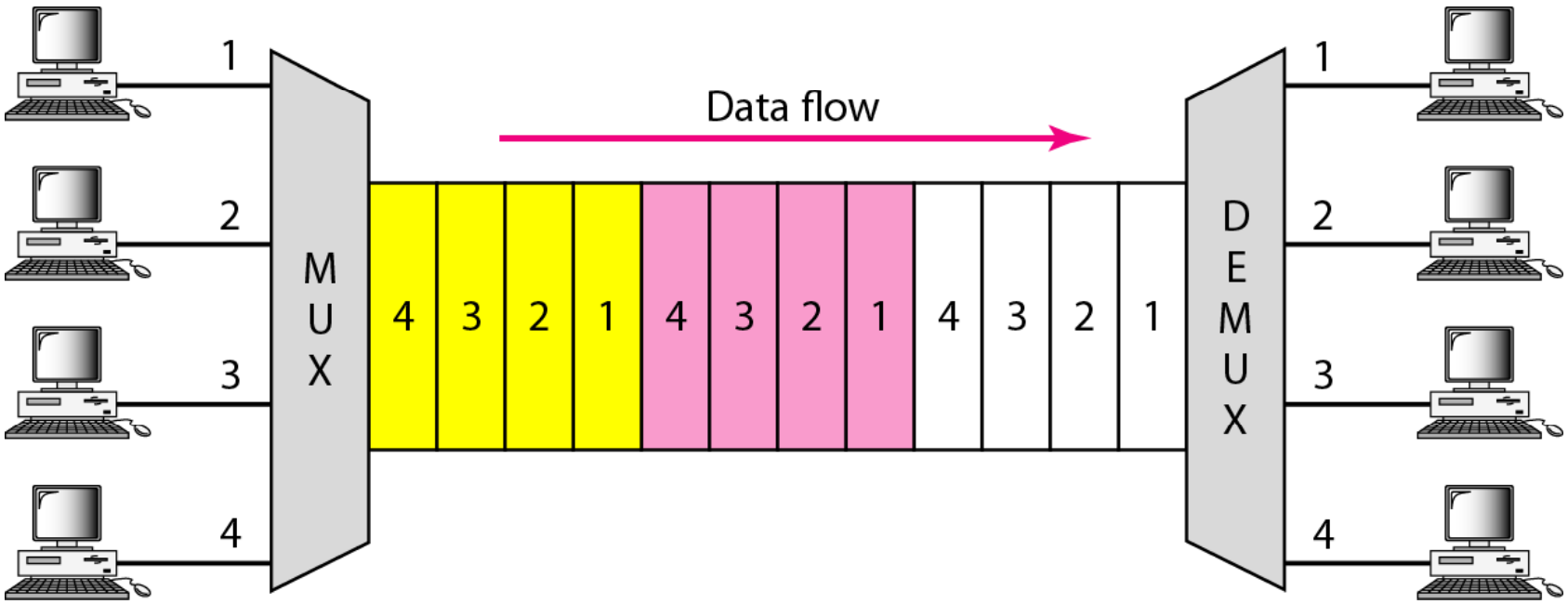


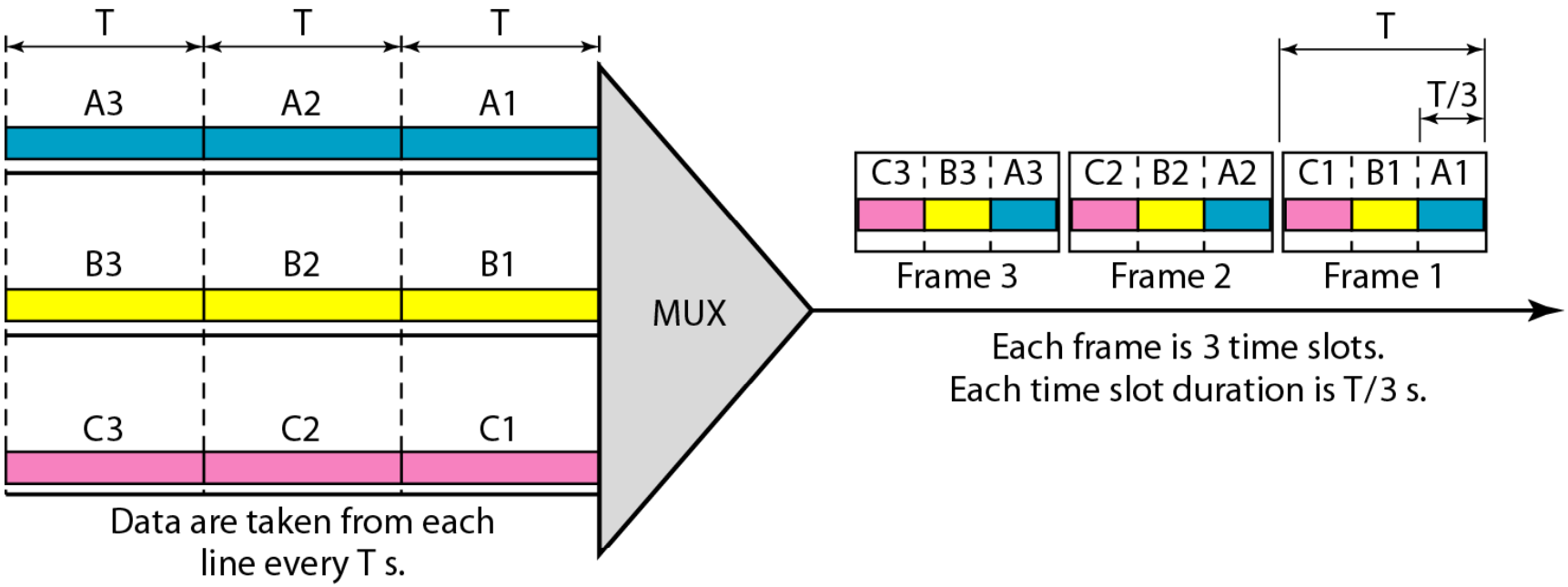
Figure 6.12 *TDM*



Note

TDM is a digital multiplexing technique for combining several low-rate channels into one high-rate one.

Figure 6.13 *Synchronous time-division multiplexing*



Note

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

Example 6.5

In Figure 6.13, the data rate for each input connection is 3 kbps. If 1 bit at a time is multiplexed (a unit is 1 bit), what is the duration of (a) each input slot, (b) each output slot, and (c) each frame?

Solution

We can answer the questions as follows:

a. The data rate of each input connection is 1 kbps. This means that the bit duration is 1/1000 s or 1 ms. The duration of the input time slot is 1 ms (same as bit duration).

Example 6.5 (continued)

b. ***The duration of each output time slot is one-third of the input time slot. This means that the duration of the output time slot is 1/3 ms.***

c. ***Each frame carries three output time slots. So the duration of a frame is 3 × 1/3 ms, or 1 ms. The duration of a frame is the same as the duration of an input unit.***

Example 6.6

Figure 6.14 shows synchronous TDM with a data stream for each input and one data stream for the output. The unit of data is 1 bit. Find (a) the input bit duration, (b) the output bit duration, (c) the output bit rate, and (d) the output frame rate.

Solution

We can answer the questions as follows:

a. The input bit duration is the inverse of the bit rate: 1/1 Mbps = 1 μs.

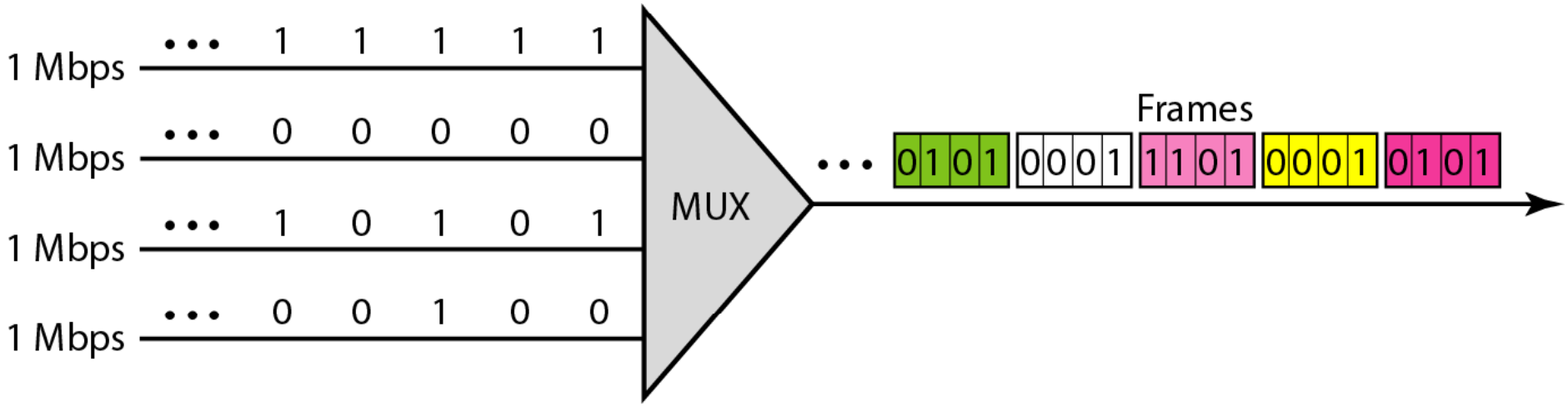
b. The output bit duration is one-fourth of the input bit duration, or ¼ μs.

Example 6.6 (continued)

c. ***The output bit rate is the inverse of the output bit duration or 1/(4μs) or 4 Mbps. This can also be deduced from the fact that the output rate is 4 times as fast as any input rate; so the output rate = 4 × 1 Mbps = 4 Mbps.***

d. ***The frame rate is always the same as any input rate. So the frame rate is 1,000,000 frames per second. Because we are sending 4 bits in each frame, we can verify the result of the previous question by multiplying the frame rate by the number of bits per frame.***

Figure 6.14 *Example 6.6*



Example 6.7

Four 1-kbps connections are multiplexed together. A unit is 1 bit. Find (a) the duration of 1 bit before multiplexing, (b) the transmission rate of the link, (c) the duration of a time slot, and (d) the duration of a frame.

Solution

We can answer the questions as follows:

a. The duration of 1 bit before multiplexing is 1 / 1 kbps, or 0.001 s (1 ms).

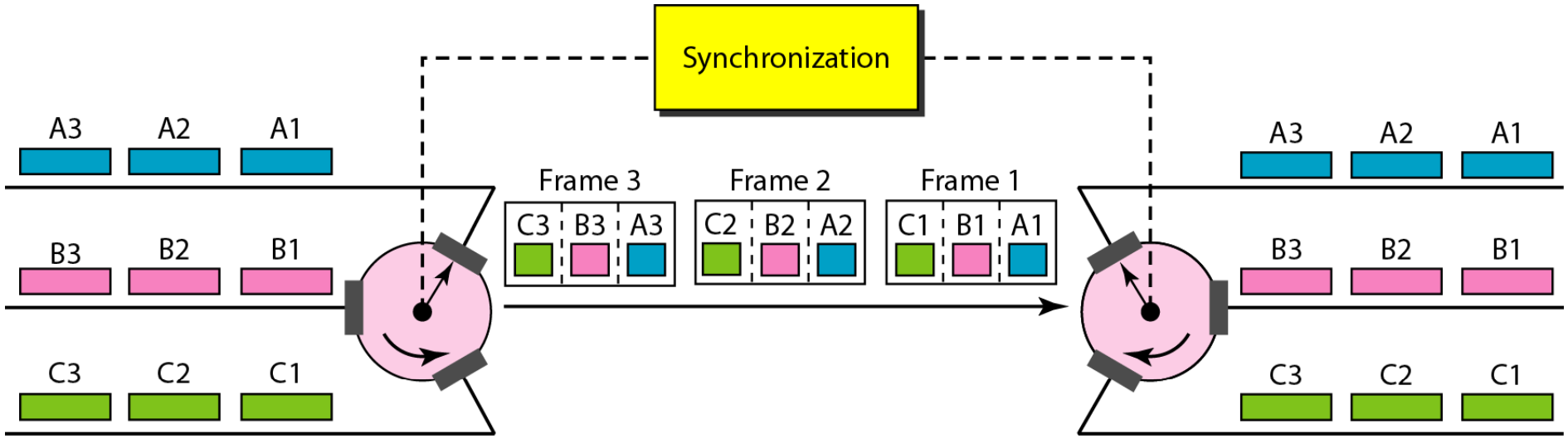
b. The rate of the link is 4 times the rate of a connection, or 4 kbps.

Example 6.7 (continued)

c. *The duration of each time slot is one-fourth of the duration of each bit before multiplexing, or 1/4 ms or 250 μs. Note that we can also calculate this from the data rate of the link, 4 kbps. The bit duration is the inverse of the data rate, or 1/4 kbps or 250 μs.*

d. *The duration of a frame is always the same as the duration of a unit before multiplexing, or 1 ms. We can also calculate this in another way. Each frame in this case has four time slots. So the duration of a frame is 4 times 250 μs, or 1 ms.*

Figure 6.15 *Interleaving*



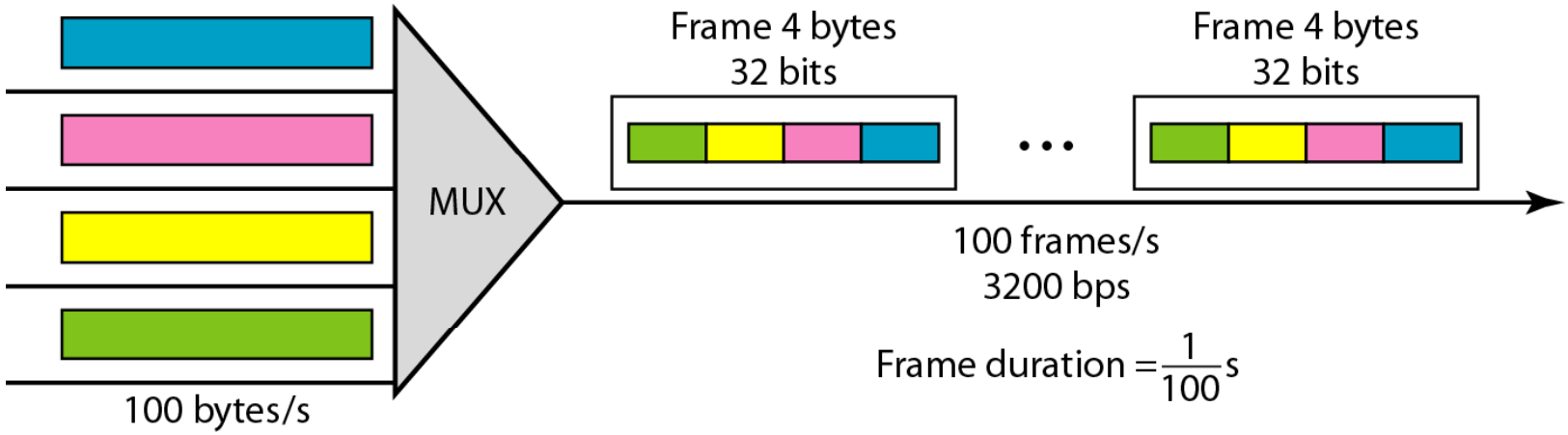
Example 6.8

Four channels are multiplexed using TDM. If each channel sends 100 bytes /s and we multiplex 1 byte per channel, show the frame traveling on the link, the size of the frame, the duration of a frame, the frame rate, and the bit rate for the link.

Solution

The multiplexer is shown in Figure 6.16. Each frame carries 1 byte from each channel; the size of each frame, therefore, is 4 bytes, or 32 bits. Because each channel is sending 100 bytes/s and a frame carries 1 byte from each channel, the frame rate must be 100 frames per second. The bit rate is 100 × 32, or 3200 bps.

Figure 6.16 *Example 6.8*



Example 6.9

A multiplexer combines four 100-kbps channels using a time slot of 2 bits. Show the output with four arbitrary inputs. What is the frame rate? What is the frame duration? What is the bit rate? What is the bit duration?

Solution

Figure 6.17 shows the output for four arbitrary inputs. The link carries 50,000 frames per second. The frame duration is therefore 1/50,000 s or 20 μs. The frame rate is 50,000 frames per second, and each frame carries 8 bits; the bit rate is 50,000 × 8 = 400,000 bits or 400 kbps. The bit duration is 1/400,000 s, or 2.5 μs.

Figure 6.17 *Example 6.9*

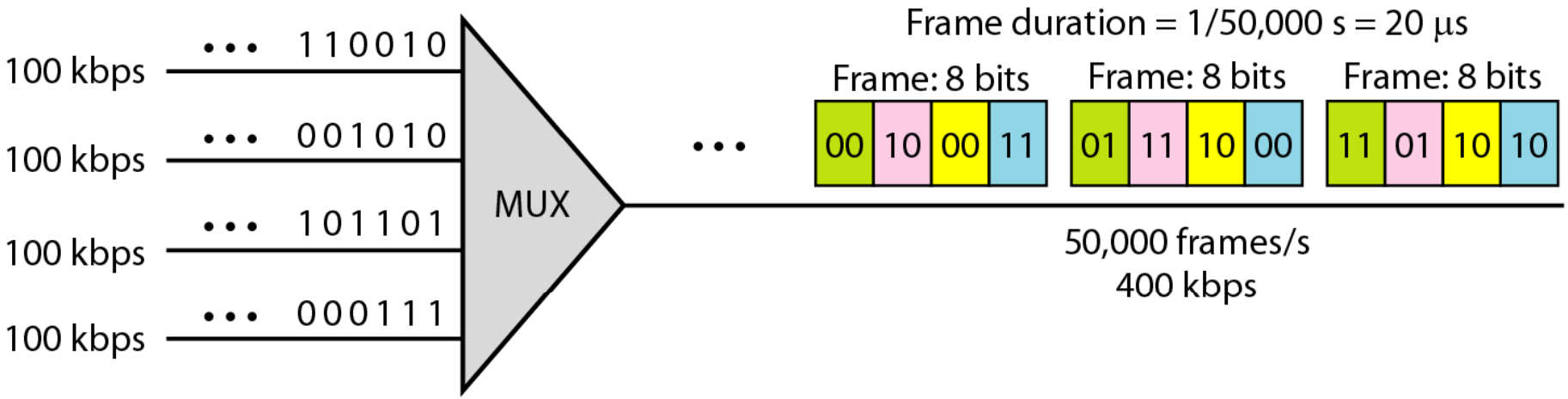


Figure 6.18 *Empty slots*

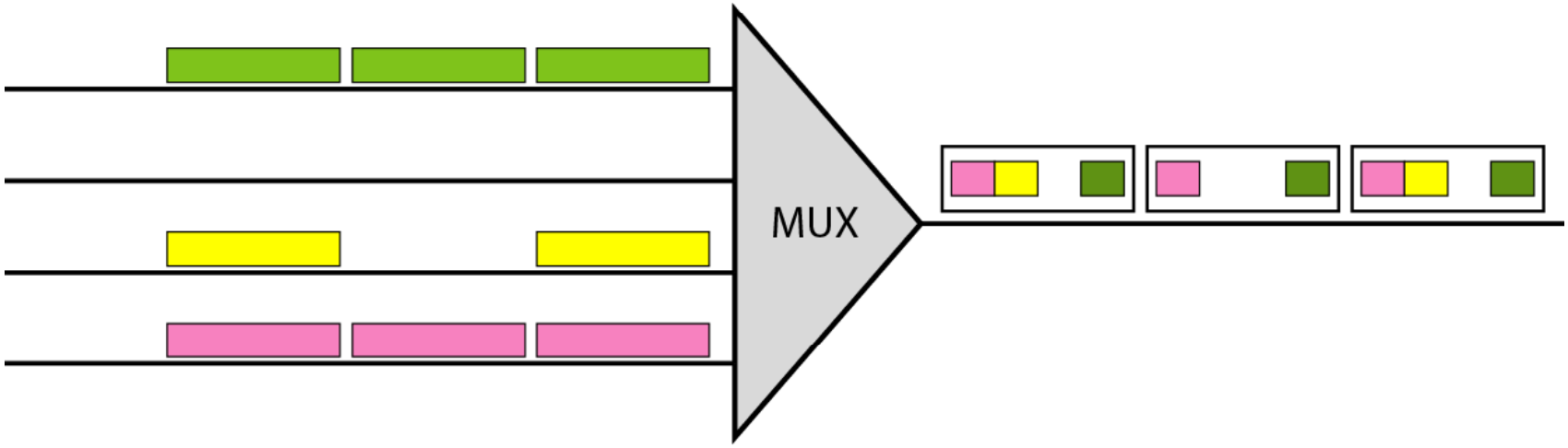


Figure 6.19 *Multilevel multiplexing*

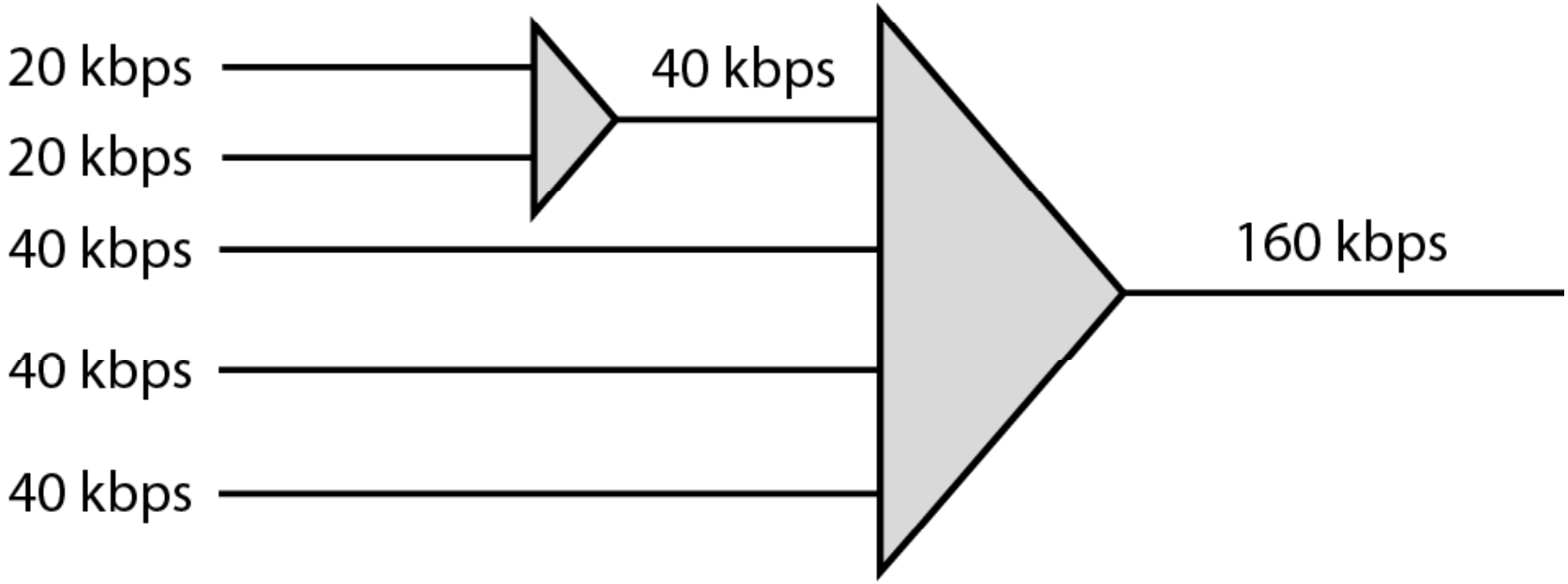


Figure 6.20 *Multiple-slot multiplexing*

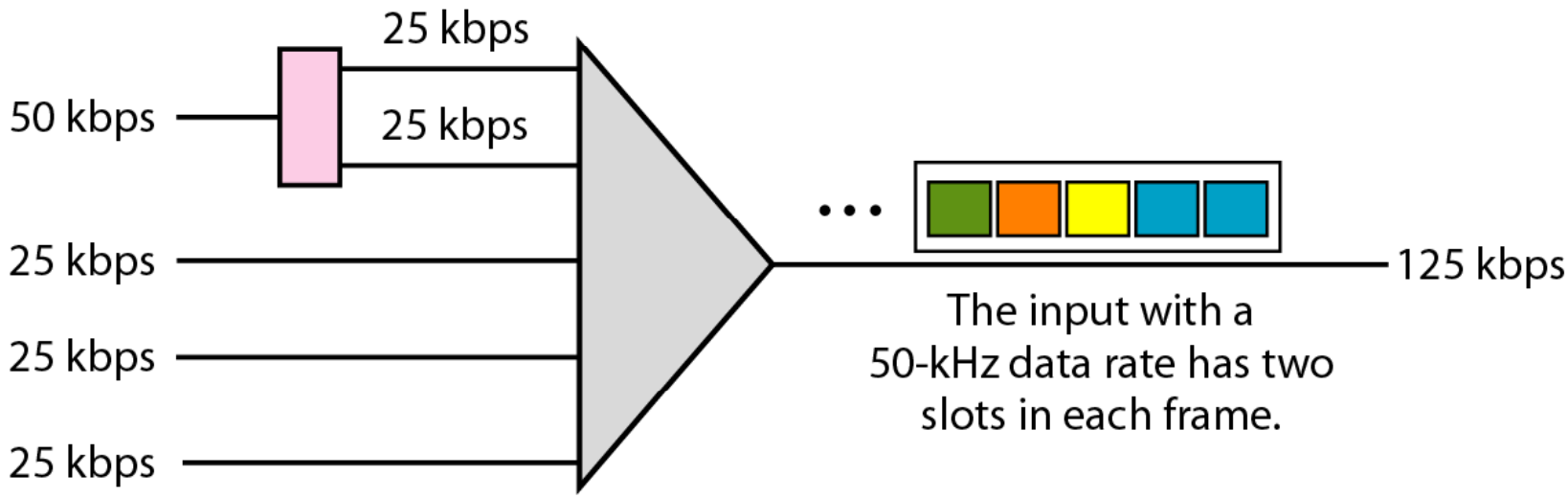


Figure 6.21 *Pulse stuffing*

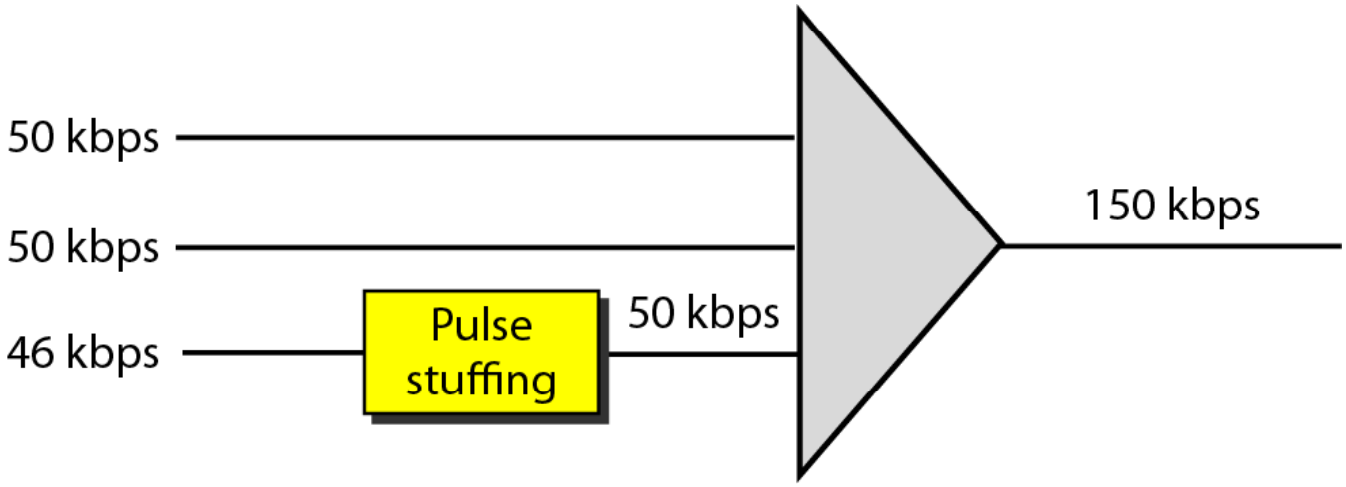
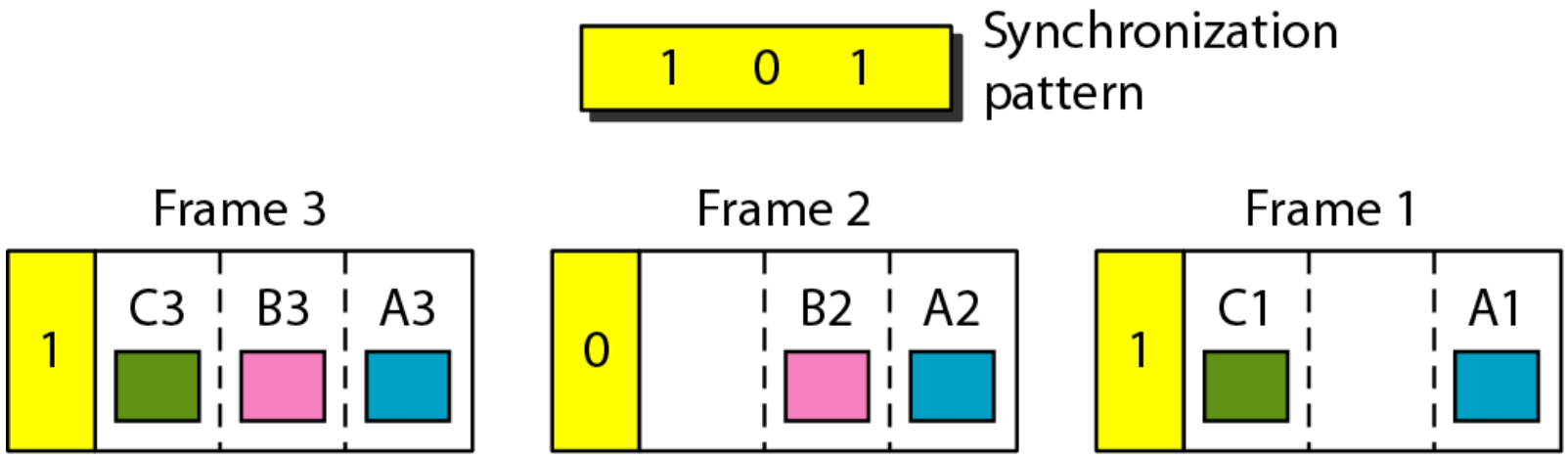


Figure 6.22 *Framing bits*



Example 6.10

We have four sources, each creating 250 characters per second. If the interleaved unit is a character and 1 synchronizing bit is added to each frame, find (a) the data rate of each source, (b) the duration of each character in each source, (c) the frame rate, (d) the duration of each frame, (e) the number of bits in each frame, and (f) the data rate of the link.

Solution

We can answer the questions as follows:

a. The data rate of each source is 250 × 8 = 2000 bps = 2 kbps.

Example 6.10 (continued)

b. *Each source sends 250 characters per second; therefore, the duration of a character is 1/250 s, or 4 ms.*

c. *Each frame has one character from each source, which means the link needs to send 250 frames per second to keep the transmission rate of each source.*

d. *The duration of each frame is 1/250 s, or 4 ms. Note that the duration of each frame is the same as the duration of each character coming from each source.*

e. *Each frame carries 4 characters and 1 extra synchronizing bit. This means that each frame is* $4 \times 8 + 1 = 33$ *bits.*

Example 6.11

Two channels, one with a bit rate of 100 kbps and another with a bit rate of 200 kbps, are to be multiplexed. How this can be achieved? What is the frame rate? What is the frame duration? What is the bit rate of the link?

Solution

We can allocate one slot to the first channel and two slots to the second channel. Each frame carries 3 bits. The frame rate is 100,000 frames per second because it carries 1 bit from the first channel. The bit rate is 100,000 frames/s × 3 bits per frame, or 300 kbps.

Figure 6.23 *Digital hierarchy*

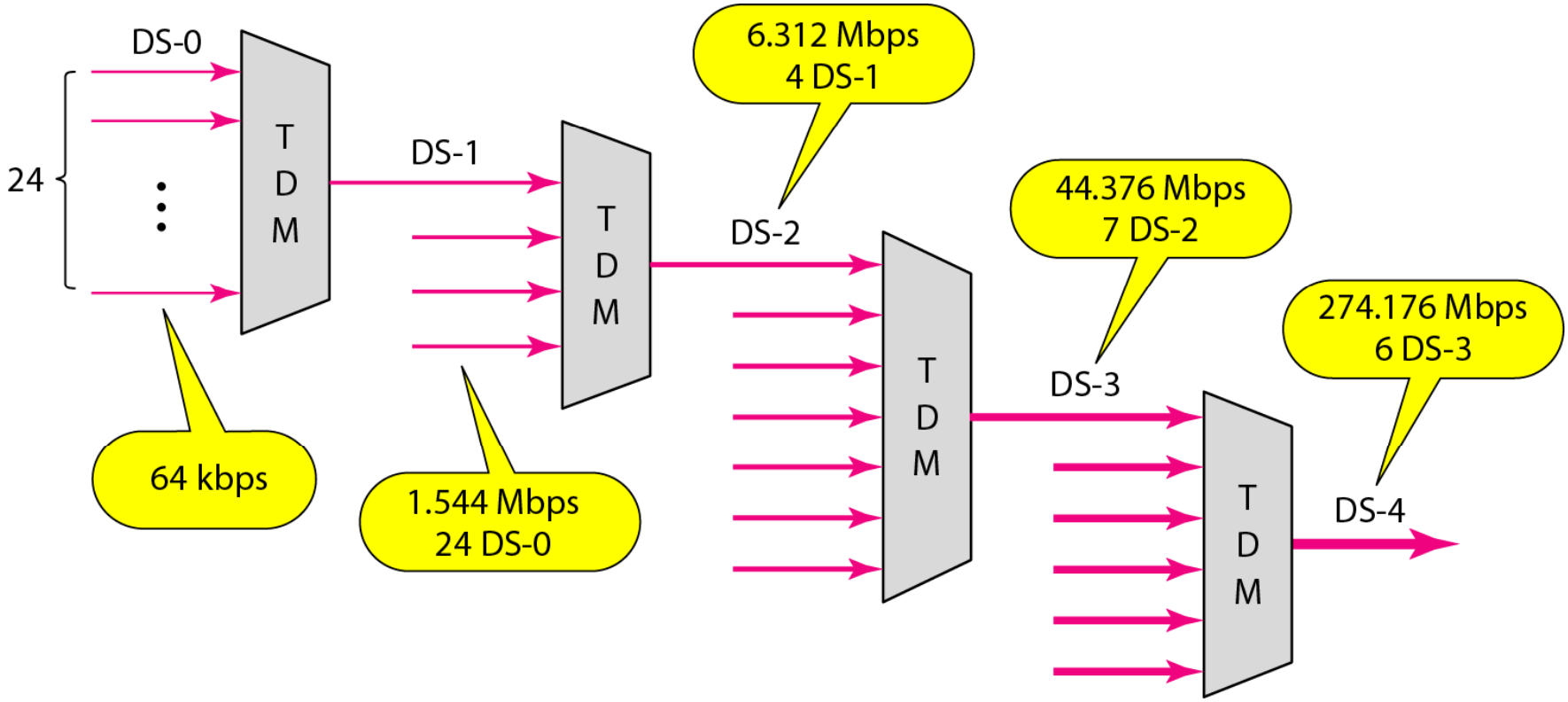


Table 6.1 *DS and T line rates*

Service	*Line*	*Rate (Mbps)*	*Voice Channels*
DS-1	T-1	1.544	24
DS-2	T-2	6.312	96
DS-3	T-3	44.736	672
DS-4	T-4	274.176	4032

Figure 6.24 *T-1 line for multiplexing telephone lines*

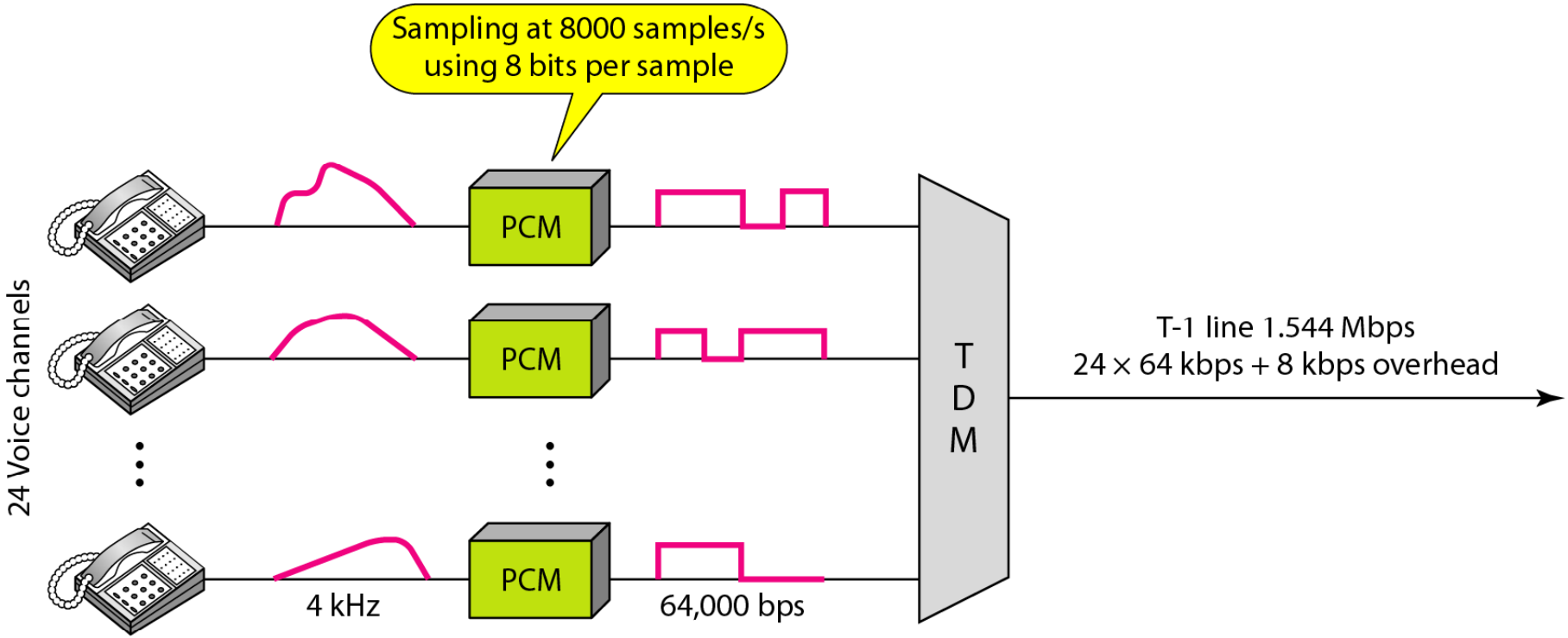


Figure 6.25 *T-1 frame structure*

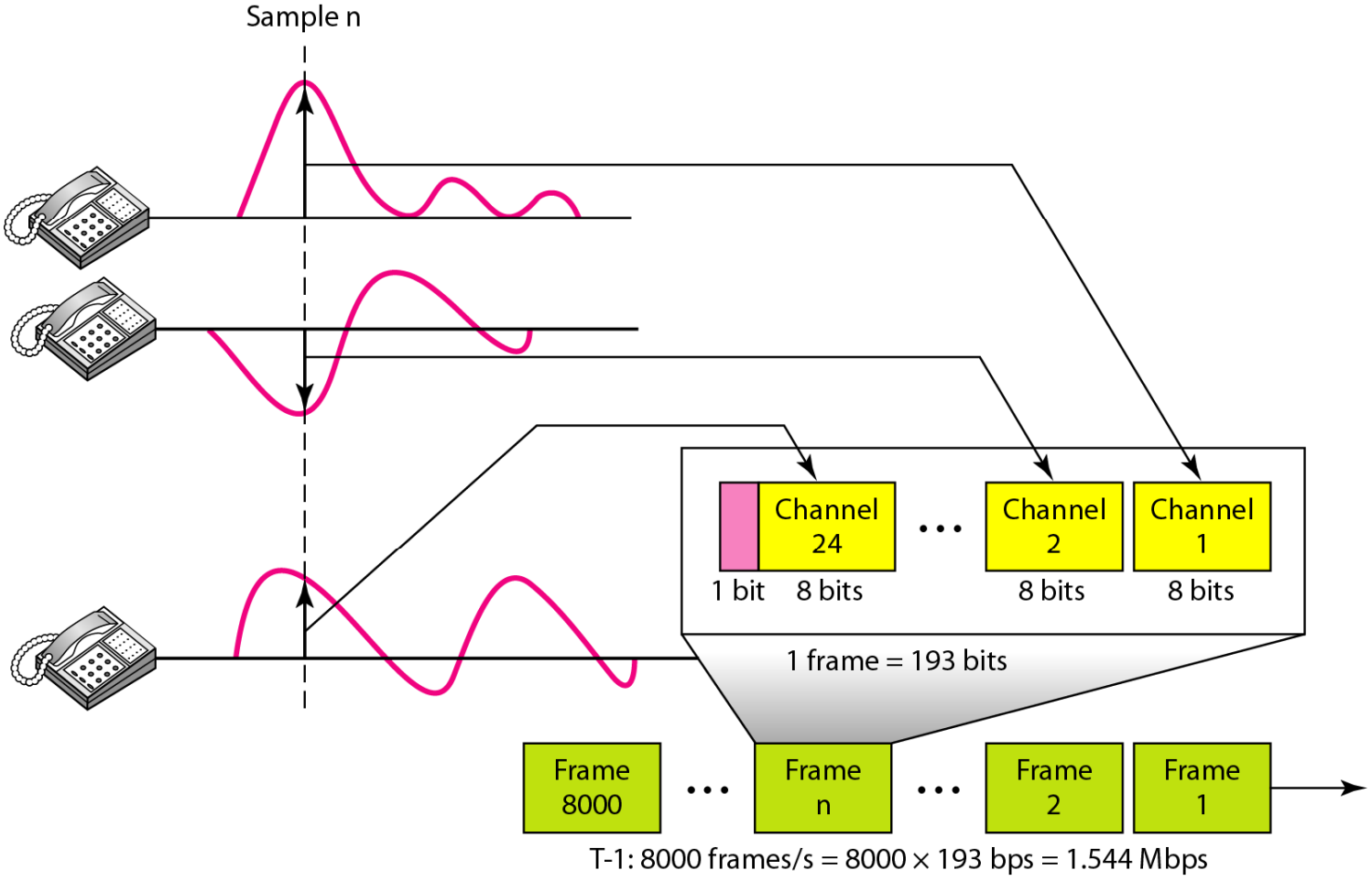
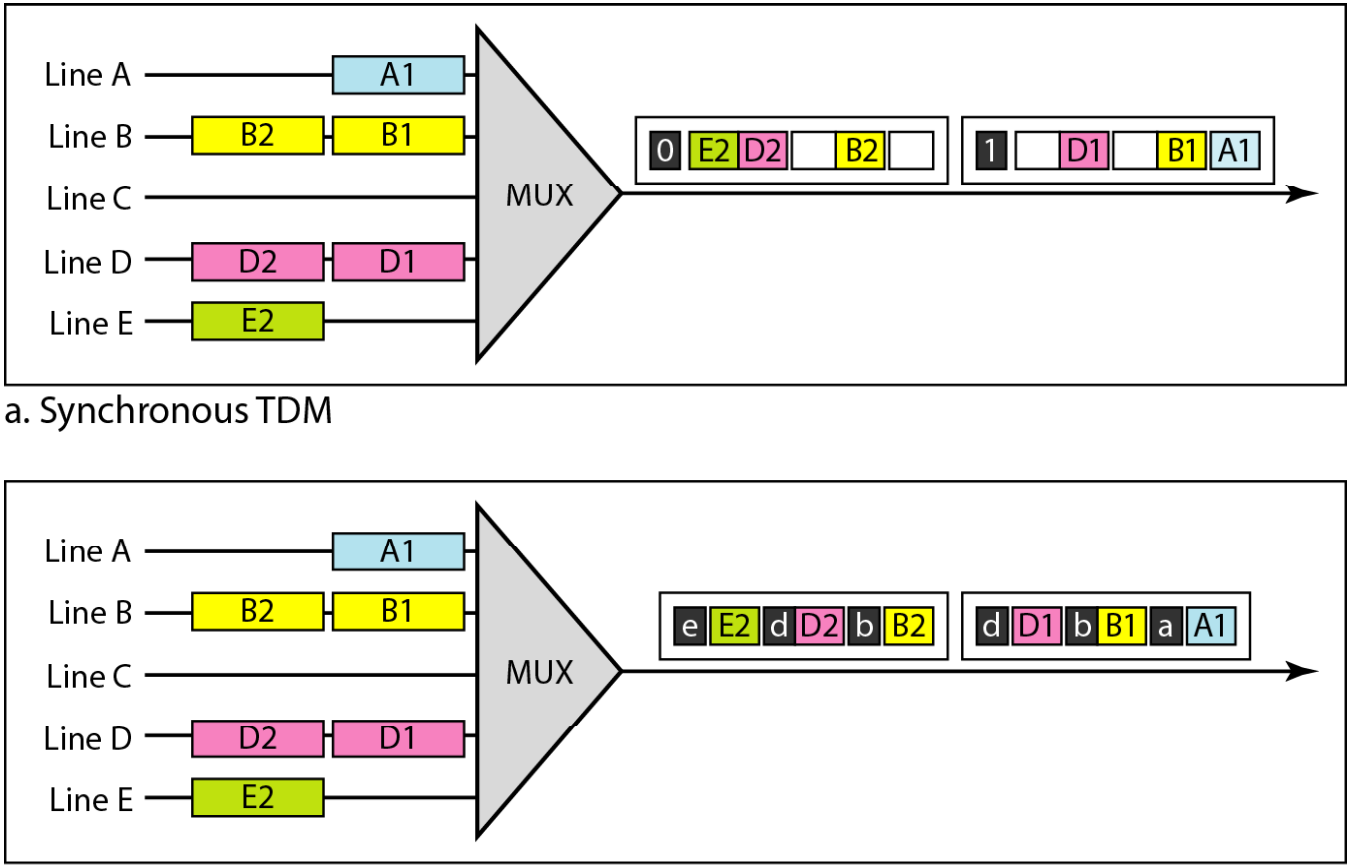


Table 6.2 ***E line rates***

Line	*Rate (Mbps)*	*Voice Channels*
E-1	2.048	30
E-2	8.448	120
E-3	34.368	480
E-4	139.264	1920

Figure 6.26 *TDM slot comparison*



a. Synchronous TDM

b. Statistical TDM

6-1 SPREAD SPECTRUM

In spread spectrum (SS), we combine signals from different sources to fit into a larger bandwidth, but our goals are to prevent eavesdropping and jamming. To achieve these goals, spread spectrum techniques add redundancy.

Topics discussed in this section:

Frequency Hopping Spread Spectrum (FHSS)
Direct Sequence Spread Spectrum Synchronous (DSSS)

Figure 6.27 *Spread spectrum*

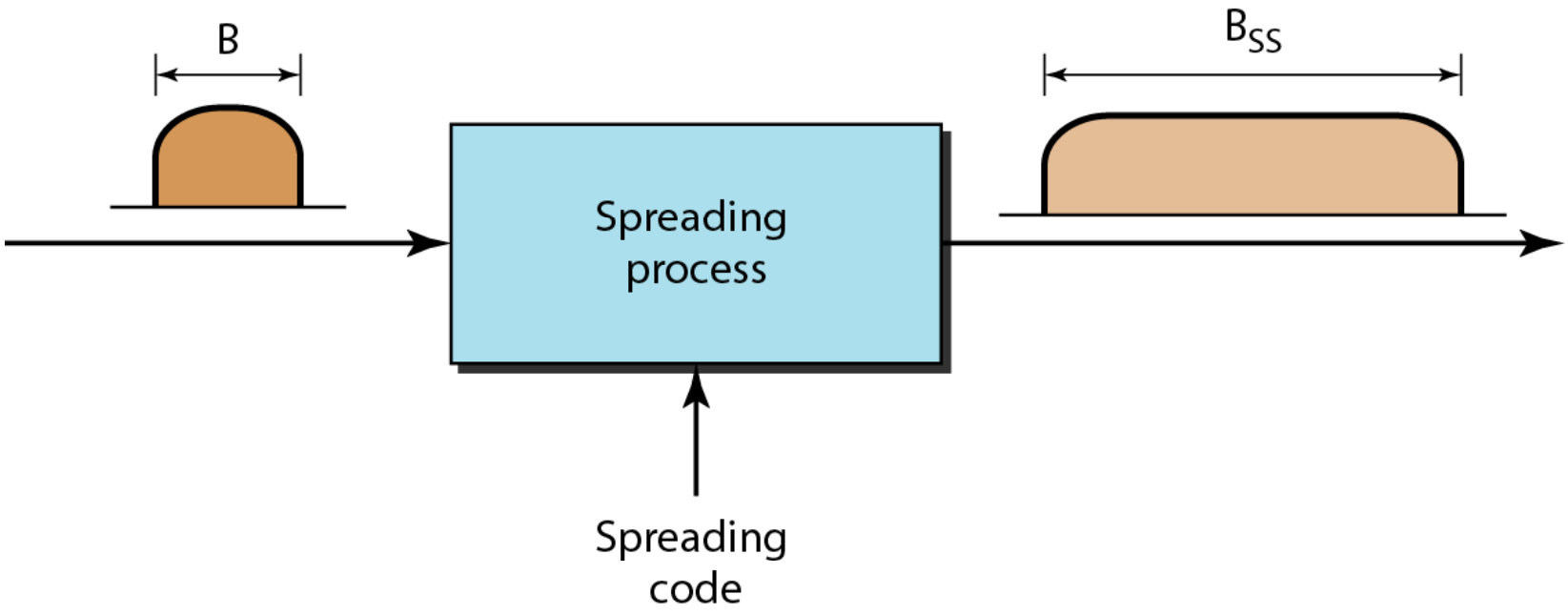


Figure 6.28 *Frequency hopping spread spectrum (FHSS)*

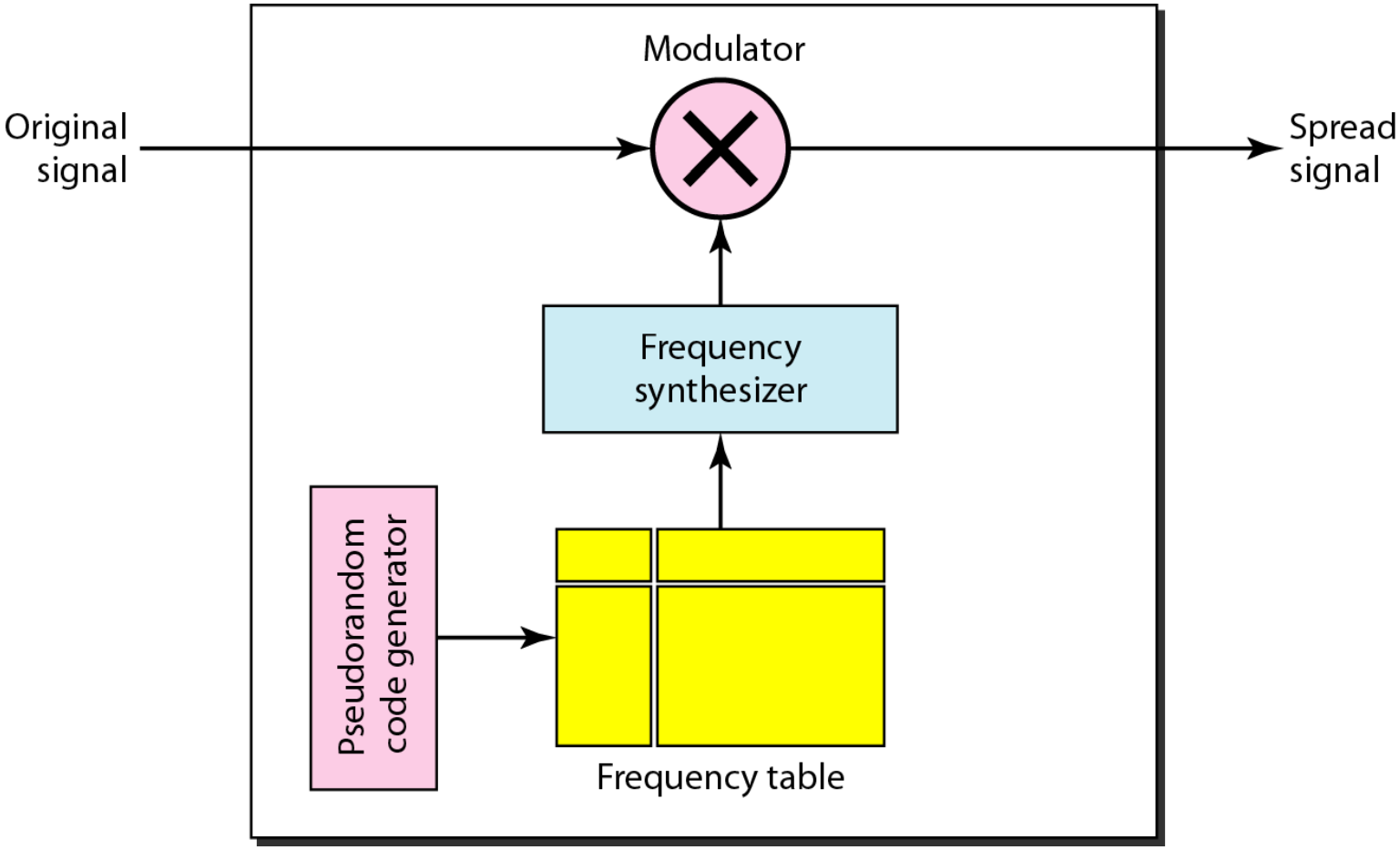


Figure 6.29 *Frequency selection in FHSS*

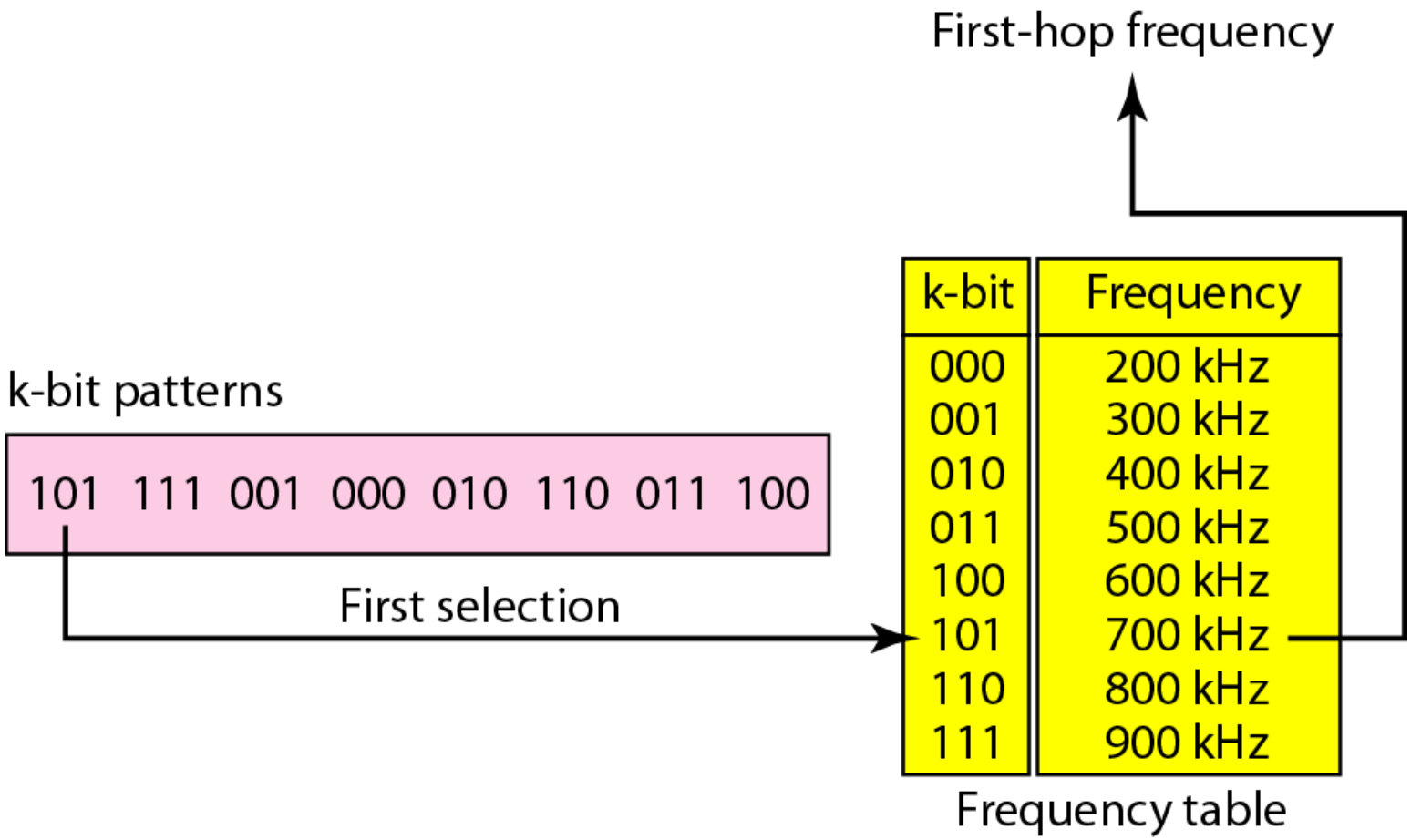


Figure 6.30 *FHSS cycles*

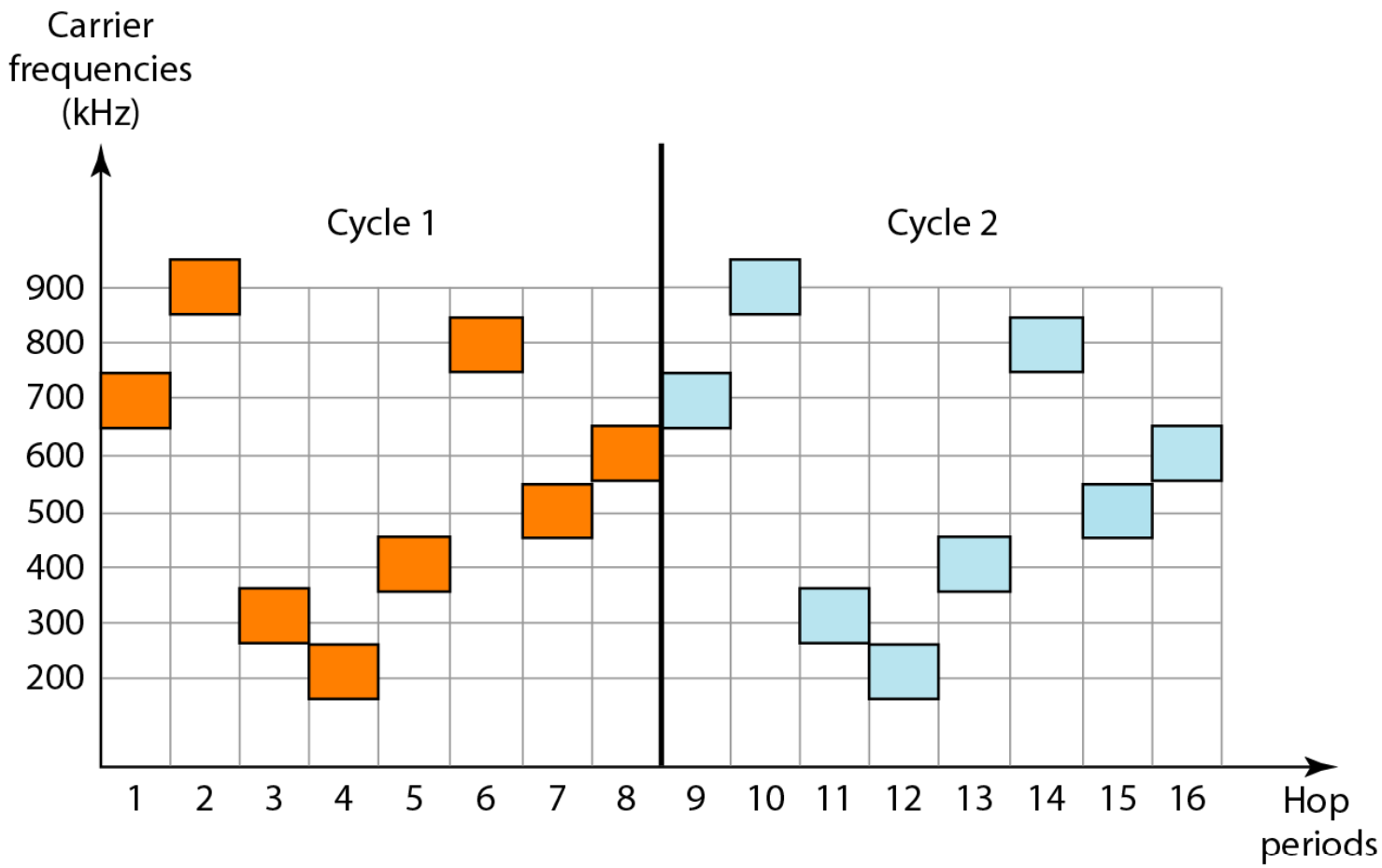
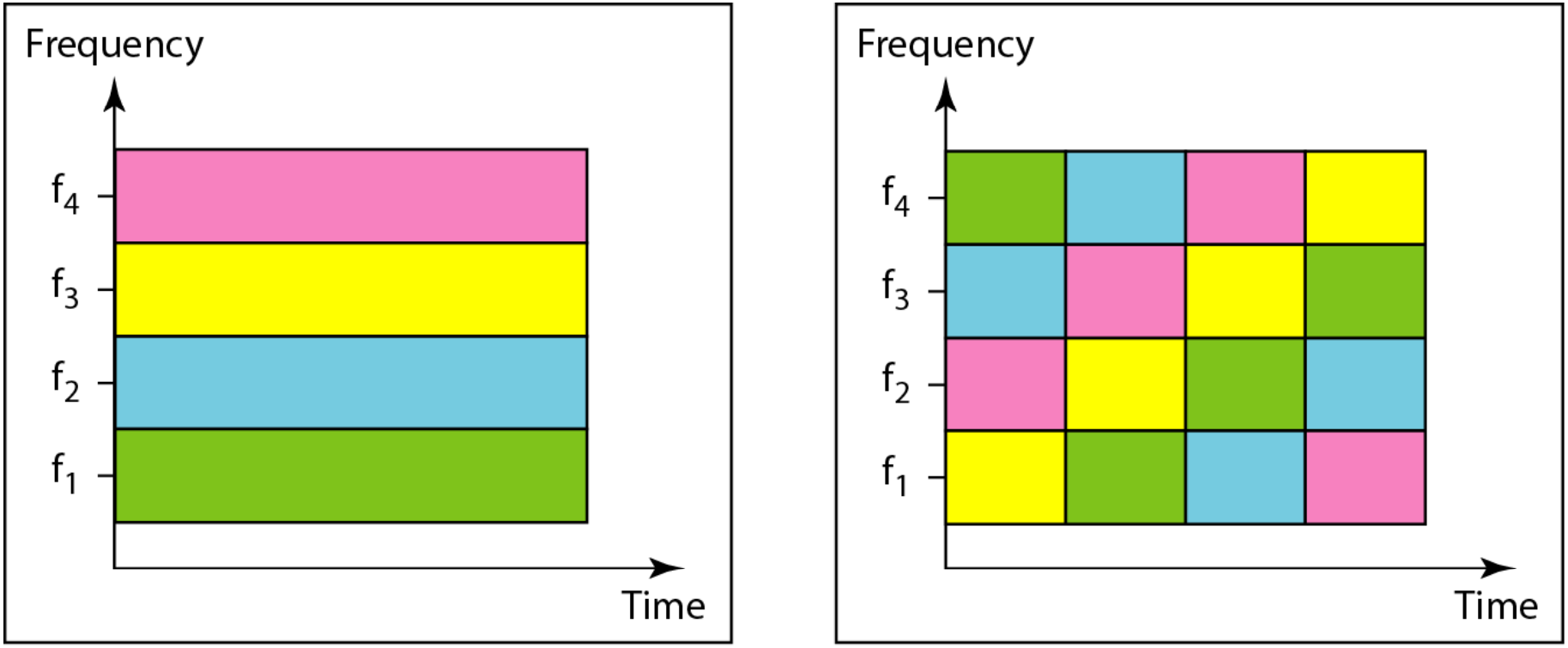


Figure 6.31 *Bandwidth sharing*



a. FDM

b. FHSS

Figure 6.32 *DSSS*

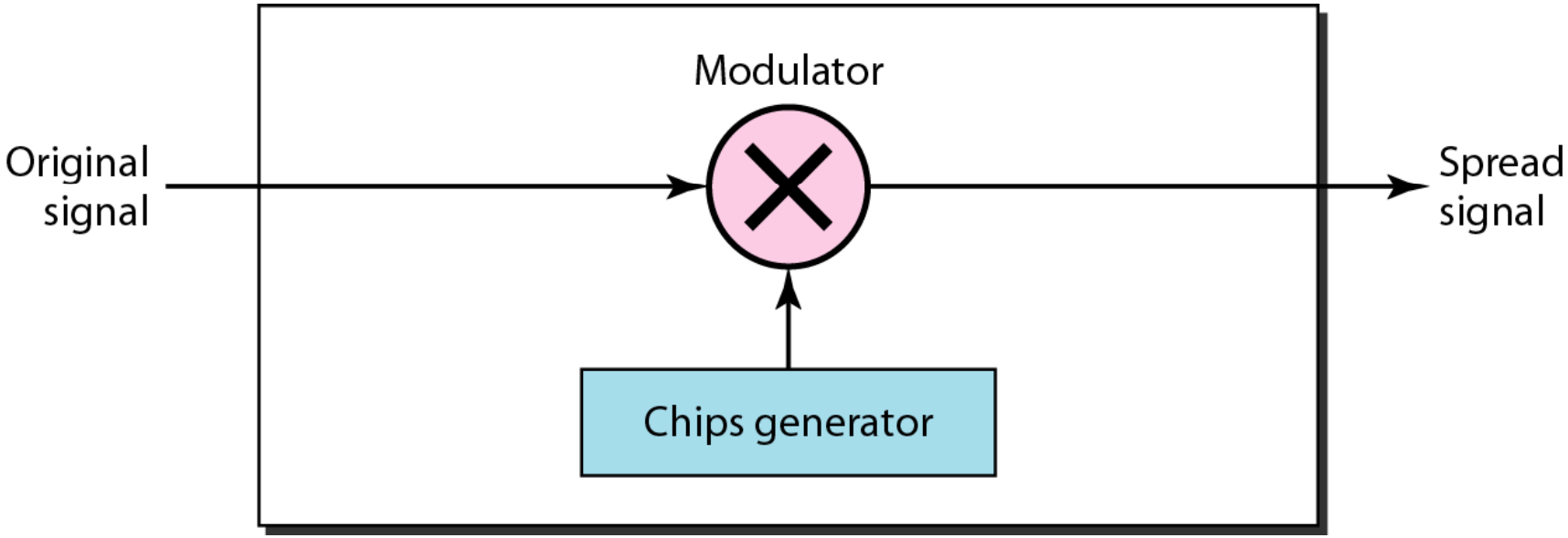
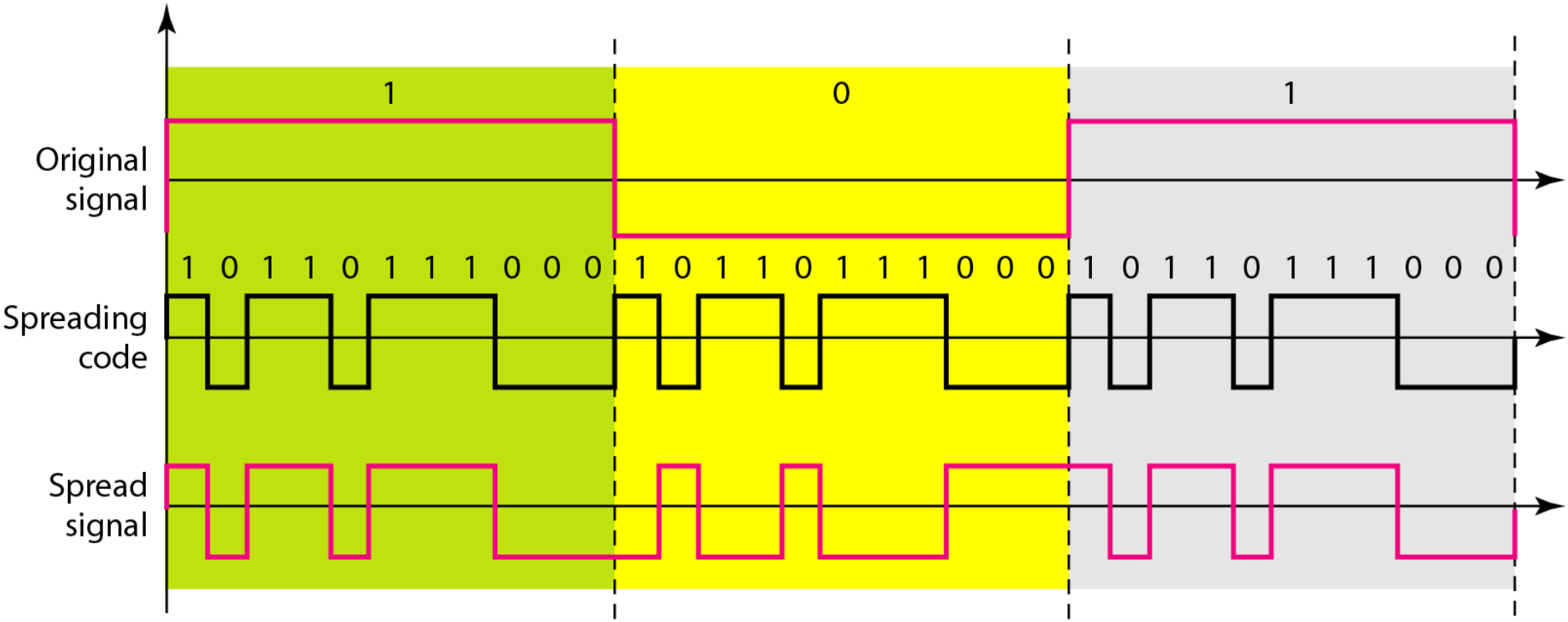


Figure 6.33 *DSSS example*



WIRED-LAN, WIRELESS-LAN & VIRTUAL LAN

Note: This is chapter 3 (Underlying Technologies) from Text book 2 Behrouz_Forouzan

WIRED LOCAL AREA NETWORKS

- A local area network (LAN) is a computer network that is designed for a limited geographic area such as a building or a campus. Although a LAN can be used as an isolated network to connect computers in an organization for the sole purpose of sharing resources, most LANs today are also linked to a wide area network (WAN) or the Internet.

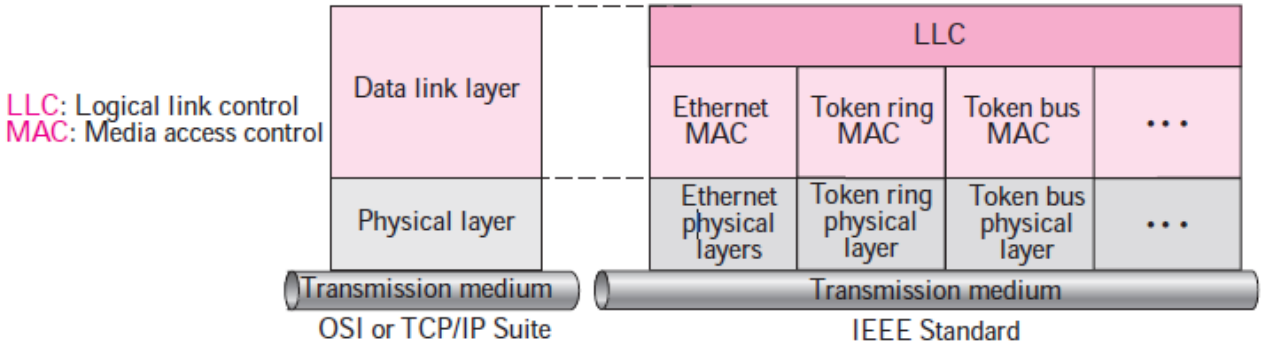
- In this section, we first briefly discuss the IEEE Standard Project 802, designed to regulate the manufacturing and interconnectivity between different LANs. We then concentrate on the Ethernet LANs.

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 does not seek to replace any part of the OSI or the Internet model.

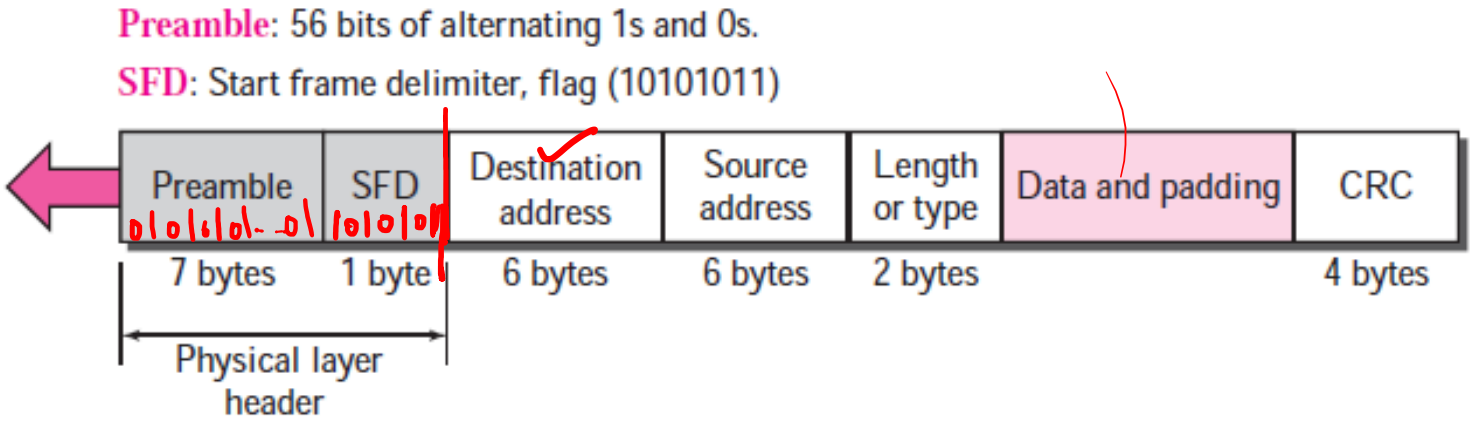
- Instead, it is a way of specifying functions of the physical layer and the data link layer of major LAN protocols. The standard was adopted by the American National Standards Institute (ANSI). In 1987, the International Standards Organization (ISO) also approved it as an international standard under the designation ISO 8802. IEEE 802.

- The relationship of the 802 Standard to the traditional OSI model is shown in Figure. The IEEE has subdivided the data link layer into two sublayers: logical link control (LLC) and media access control (MAC). IEEE has also created several physical layer standards for different LAN protocols.



Frame Format

- The Ethernet frame contains seven fields: preamble, SFD, DA, SA, length or type of data unit, 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.



Frame Format

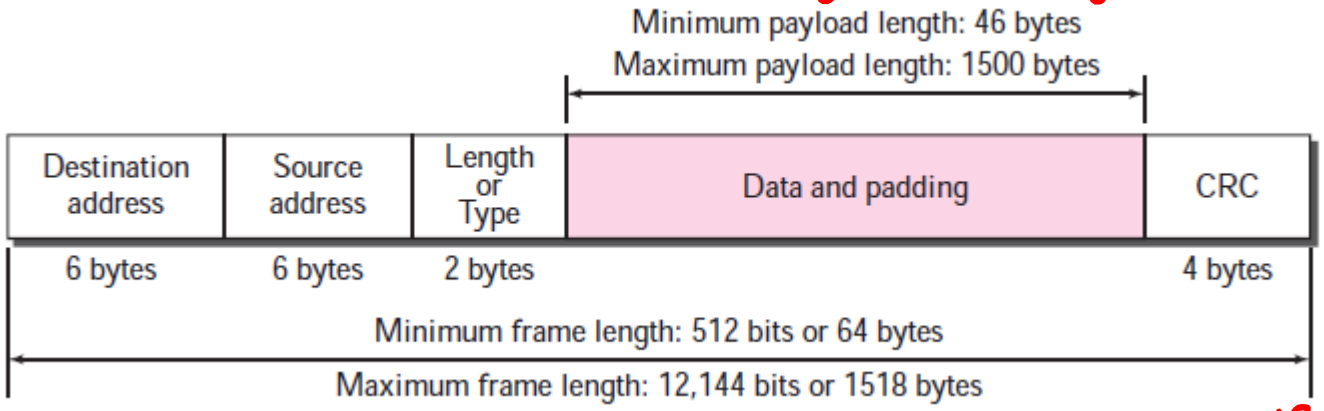
- **Preamble.** The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the stations to 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.

(010101....(8-bits)..01|10101011

- **Start frame delimiter (SFD).** The second field (1 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 are 11 and alert the receiver that the next field is the destination address. The SFD is also added at the physical layer.
- **Destination address (DA).** The DA field is 6 bytes and contains the physical address of the destination station or stations to receive the packet. We will discuss addressing shortly.
- **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. Both uses are common today.
- **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

Frame Length

32-byte + 00000 46 - 32 bytes.
14 bytes = 14 bytes (zeros)



→ Allocated by → 18 bytes.

→ Total frame = 64 bytes. (min)

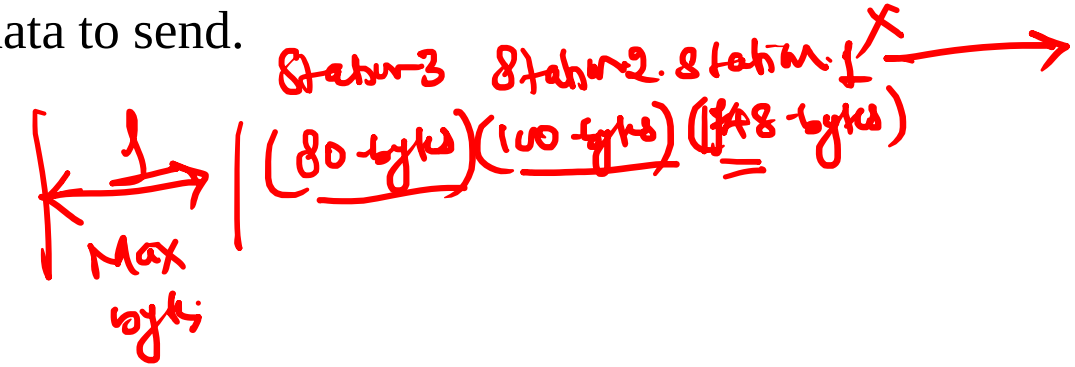
1500 + 18 = 1518 bytes.

64 - 18 = 46.

- The minimum length restriction is required for the correct operation of CSMA/CD, as we will see shortly. 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.

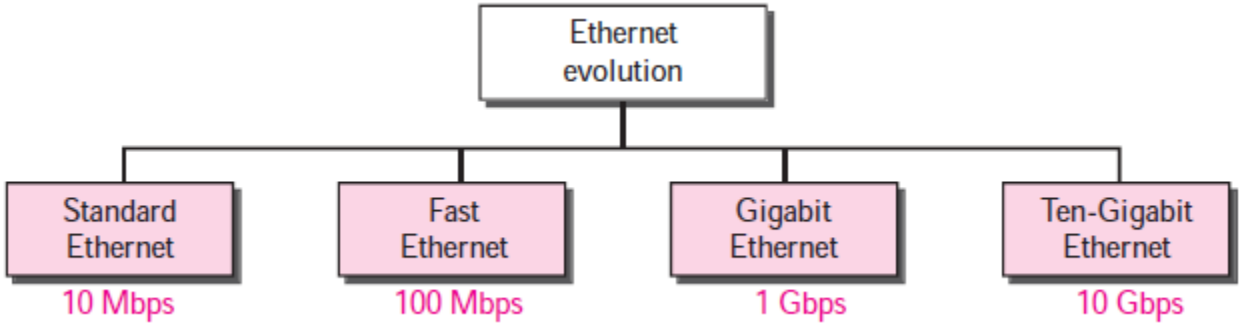
Frame Length

- The standard defines the maximum length of a frame (without preamble and SFD field) 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 historical 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.



Ethernet Evolution

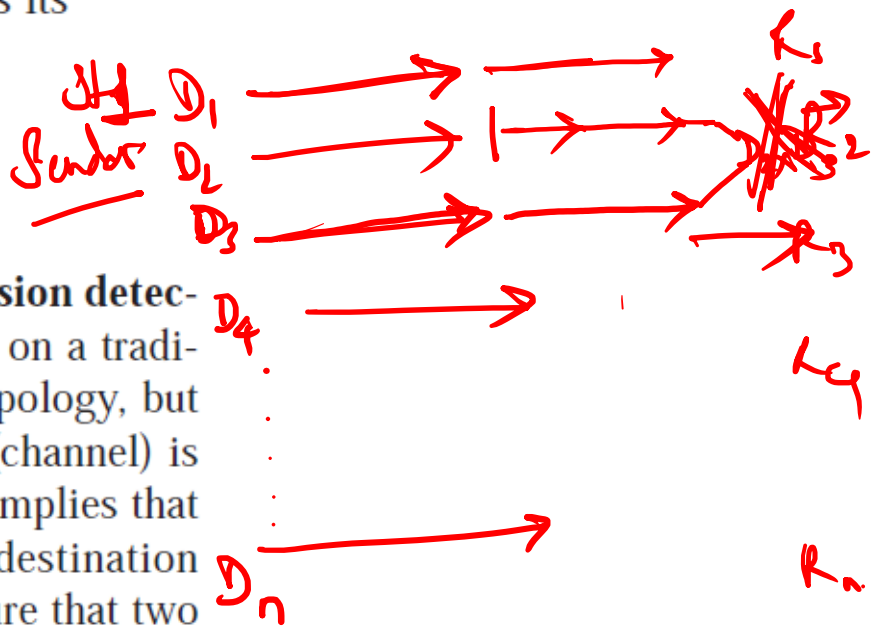
Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations: **Standard Ethernet** (10 Mbps), **Fast Ethernet** (100 Mbps), **Gigabit Ethernet** (1 Gbps), and **Ten-Gigabit Ethernet** (10 Gbps), as shown in Figure 3.6. We briefly discuss all these generations starting with the first, Standard (or traditional) Ethernet.



Ethernet Evolution

Standard Ethernet

The original Ethernet with 10-Mbps data rate is now history, but we briefly discuss its characteristics to pave the way for understanding other Ethernet versions.

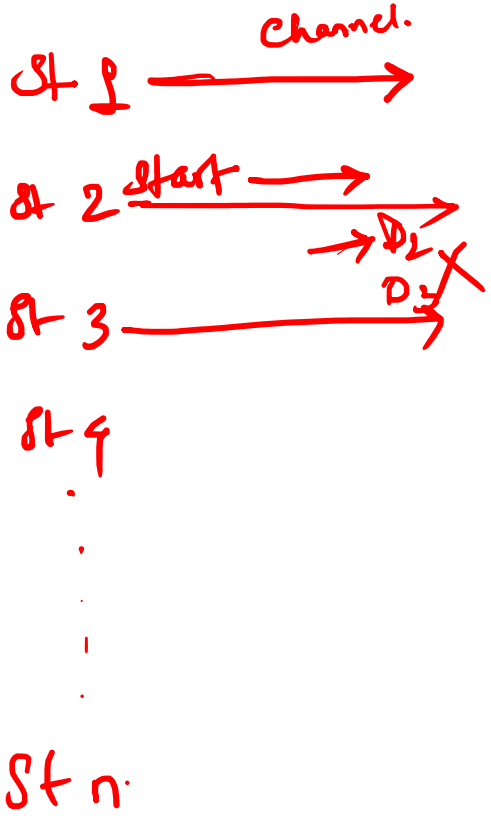


Access Method: CSMA/CD

The IEEE 802.3 standard defines **carrier sense multiple access with collision detection (CSMA/CD)** as the access method for traditional Ethernet. Stations on a traditional Ethernet can be connected together using a physical bus or star topology, but the logical topology is always a bus. By this, we mean that the medium (channel) is shared between stations and only one station at a time can use it. It also implies that all stations receive a frame sent by a station (broadcasting). The real destination keeps the frame while the rest drop it. In this situation, how can we be sure that two stations are not using the medium at the same time? If they do, their frames will collide with each other.

Ethernet Evolution

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. **Carrier sense multiple access (CSMA)** requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk." CSMA can reduce the possibility of collision, but it cannot eliminate it. The reason for this is shown in Figure 3.7, a space and time model of a CSMA network. Stations are connected to a shared channel (usually a dedicated medium).



Ethernet Evolution

Standard Ethernet

The original Ethernet with 10-Mbps data rate is now history, but we briefly discuss its characteristics to pave the way for understanding other Ethernet versions.

Implementation

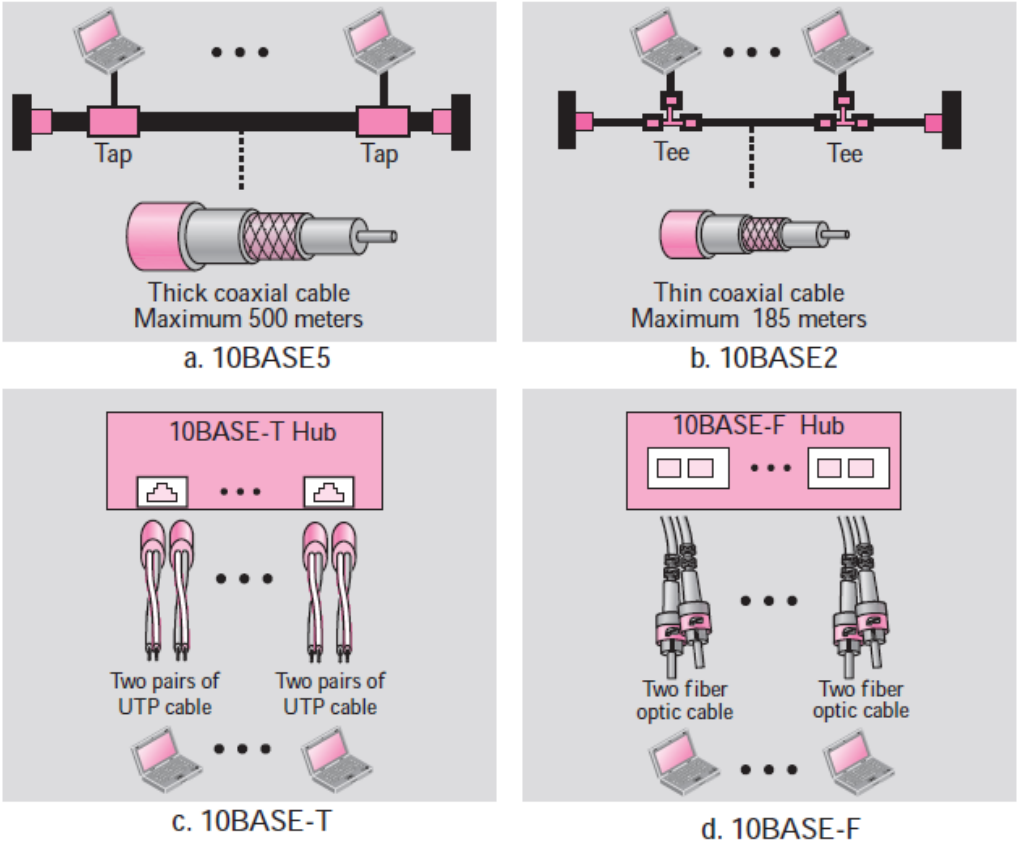
The Standard Ethernet defined several implementations, but only four of them became popular during '80s. Table 3.1 shows a summary of Standard Ethernet implementations. In the nomenclature 10Base-X, the number defines the data rate (10 Mbps), the term Base means baseband (digital) signal, and X approximately defines either the maximum size of the cable in 100 meters (for example 5 for 500 or 2 for 185 meters) or the type of the cable, T for unshielded twisted pair cable (UTP) and F for fiber-optic.

Table 3.1 *Summary of Standard Ethernet implementations*

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Medium	Thick coax	Thin coax	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m

Ethernet Evolution

Standard Ethernet: Implementation



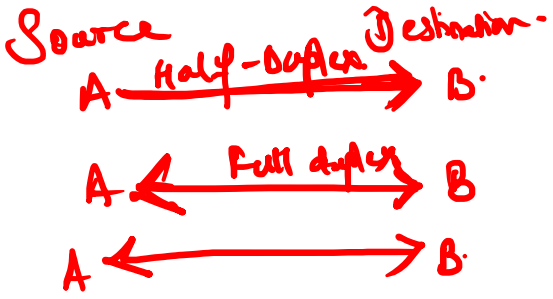
Ethernet Evolution

Fast Ethernet

Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber 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.

Ethernet Evolution



MAC Sublayer

A main consideration in the evolution of Ethernet from 10 to 100 Mbps was to keep the MAC sublayer untouched. However, a decision was made to drop the bus topologies and keep only the star topology. For the star topology, there are two choices: half duplex and full duplex. In the half-duplex approach, the stations are connected via a hub; in the full-duplex approach, the connection is made via a switch with buffers at each port (see Section 3.5, Connecting Devices, at the end of the chapter).

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. However, the implementations keep CSMA/CD for backward compatibility with Standard Ethernet.

Autonegotiation

A new feature added to Fast Ethernet is called **autonegotiation.** It allows a station or a hub a range of capabilities. Autonegotiation 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.

Ethernet Evolution

Fast Ethernet

Implementation

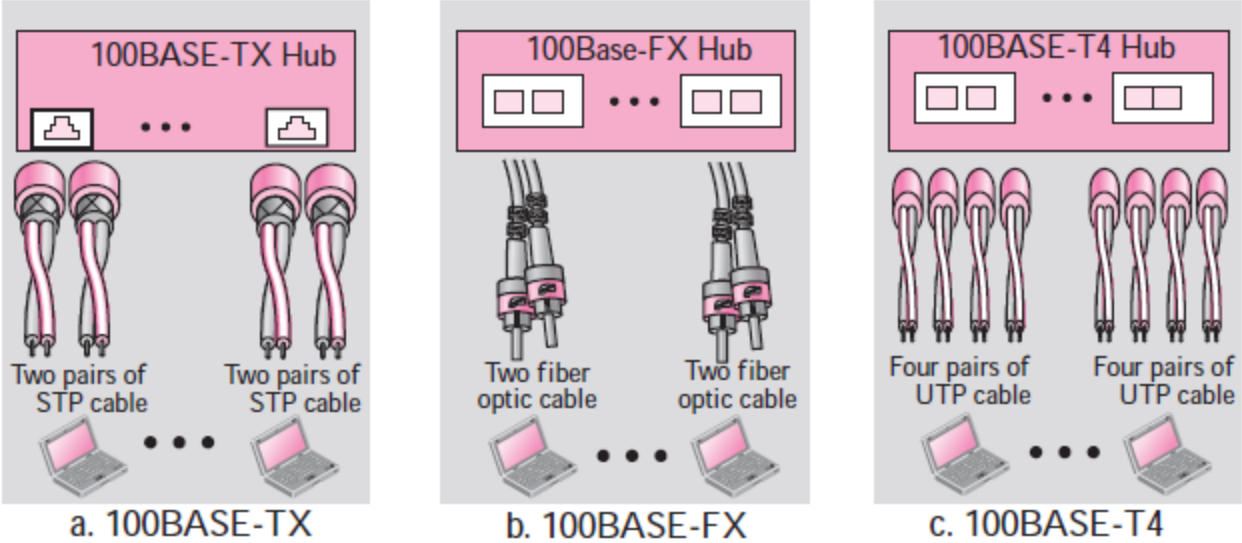
Fast Ethernet implementation at the physical layer can be categorized as either two-wire or four-wire. The two-wire implementation can be either shielded twisted pair, STP (**100Base-TX**) or fiber-optic cable (**100Base-FX**). The four-wire implementation is designed only for unshielded twist pair, UTP (**100Base-T4**). Table 3.2 is a summary of the Fast Ethernet implementations.

Table 3.2 *Summary of Fast Ethernet implementations*

Characteristics	*100Base-TX*	*100Base-FX*	*100Base-T4*
Media	STP	Fiber	UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m

Ethernet Evolution

Fast Ethernet : Implementation



Ethernet Evolution

Gigabit Ethernet

The need for an even higher data rate resulted in the design of the Gigabit Ethernet Protocol (1000 Mbps). The IEEE committee calls 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.
5. Keep the same minimum and maximum frame lengths.
6. To support autonegotiation as defined in Fast Ethernet.

Ethernet Evolution

Gigabit Ethernet

Implementation

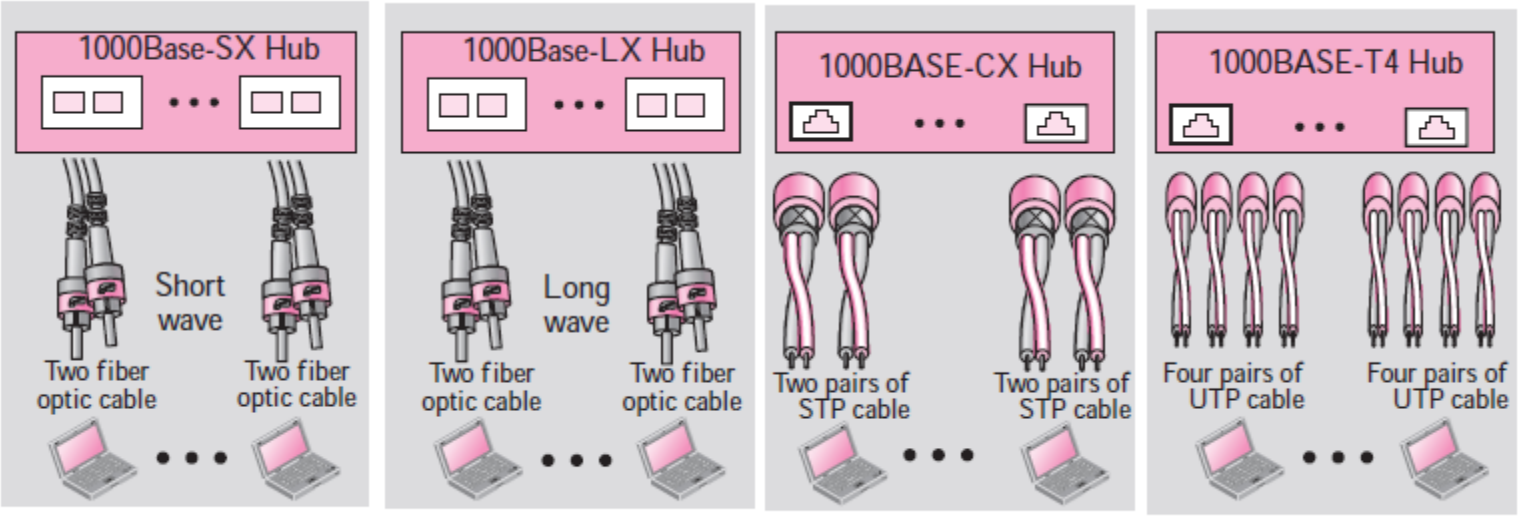
Table 3.3 is a summary of the Gigabit Ethernet implementations.

Table 3.3 *Summary of Gigabit Ethernet implementations*

Characteristics	*1000Base-SX*	*1000Base-LX*	*1000Base-CX*	*1000Base-T4*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m

Ethernet Evolution

Gigabit Ethernet : Implementation



a. 1000BASE-SX b. 100BASE-LX c. 1000BASE-CX d. 1000BASE-T4

Ethernet Evolution

Ten-Gigabit Ethernet

The IEEE committee created Ten-Gigabit Ethernet and called it Standard 802.3ae. The goals of the Ten-Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 10 Gbps.
2. Make it compatible with Standard, Fast, and Gigabit Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. Allow the interconnection of existing LANs into a metropolitan area network (MAN) or a wide area network (WAN).
7. Make Ethernet compatible with technologies such as Frame Relay and ATM.

Ethernet Evolution

Ten-Gigabit Ethernet

Implementation

Ten-Gigabit Ethernet operates only in full duplex mode, which means there is no need for contention; CSMA/CD is not used in Ten-Gigabit Ethernet. Three implementations are the most common: 10GBase-S, 10GBase-L, and 10GBase-E. Table 3.4 shows a summary of the Ten-Gigabit Ethernet implementation.

Table 3.4 *Ten-Gigabit Ethernet Implementation*

Characteristics	*10GBase-S*	*10GBase-L*	*10GBase-E*
Media	multi-mode fiber	single-mode fiber	single-mode fiber
Number of wires	2	2	2
Maximum length	300 m	10,000 m	40,000 m

WIRELESS LANS

Introduction

IEEE 802

- 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.

- In this section, we concentrate on two wireless technologies for LANs: 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.

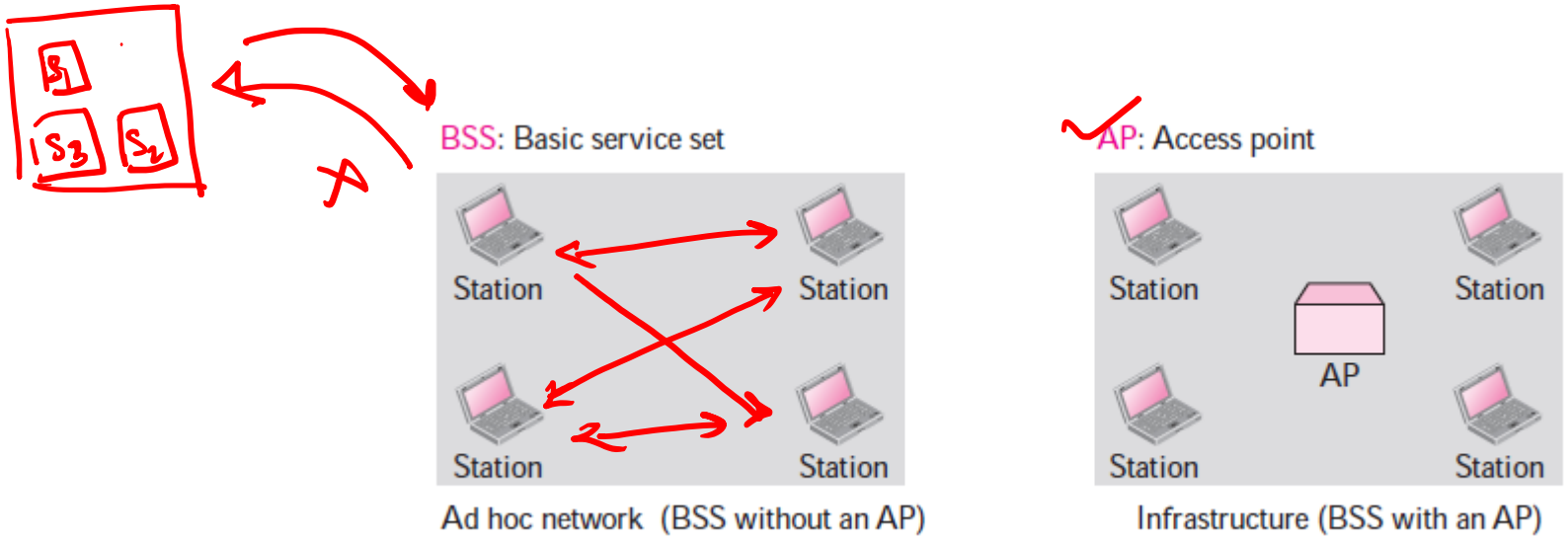
Architecture: The standard defines two kinds of services: the basic service set (BSS) and the extended service set (ESS).

- **Basic Service Set:** IEEE 802.11 defines the **basic service set (BSS)** as the building block of a wireless LAN. A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the **access point (AP).**

  - The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an ad hoc architecture. In this architecture, stations can form a 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 sometimes referred to as an infrastructure network.

IEEE 802.11

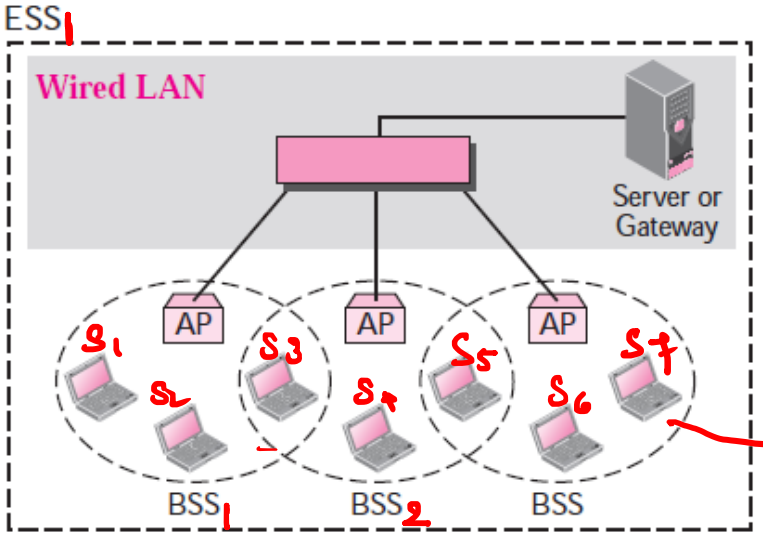
- The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called an ad hoc architecture.
- In this architecture, stations can form a network without the need of an AP; they can locate one another and agree to be part of a BSS. A BSS with an AP is sometimes referred to as an infrastructure network.

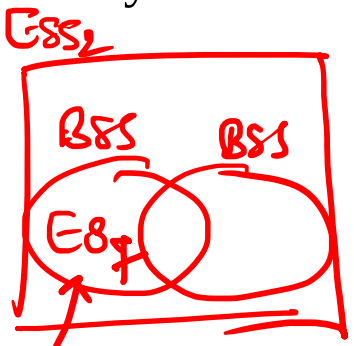


IEEE 802.11

- Extended Service Set: An extended service set (ESS) is made up of two or more BSSs with Aps.
- 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.

ESS: Extended service set
BSS: Basic service set
AP: Access point





IEEE 802.11

- **Station Types:**

(Stationary).

- 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.

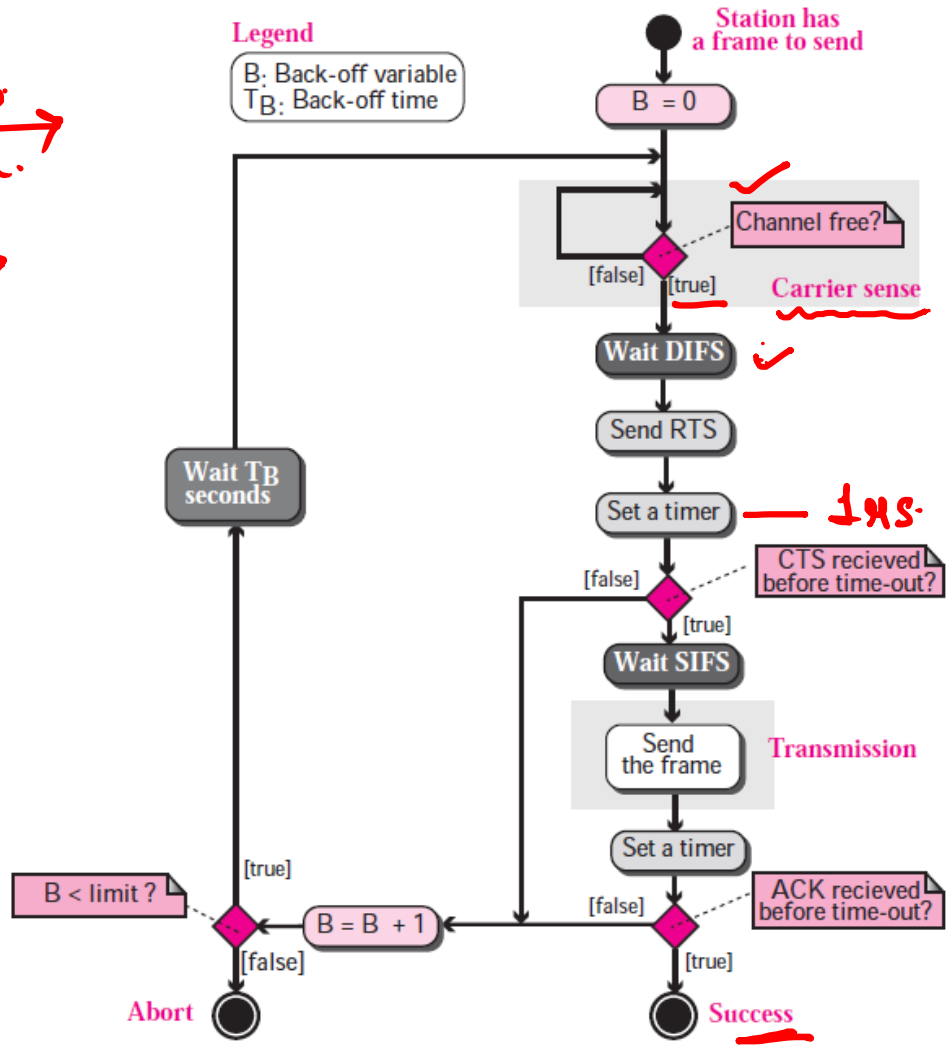
IEEE 802.11

Datalink layer → LLC / MAC.

- **MAC Sublayer:**
- There are two different MAC sublayers in this protocol, however; the one that is used most of the time is based on CSMA/CA (carrier sense multiple access with collision avoidance).

① CSMA/CD → Loss DXD path.

② CSMA → Station empty



* Reasons for not implementing CSMA/CD → (Detection).

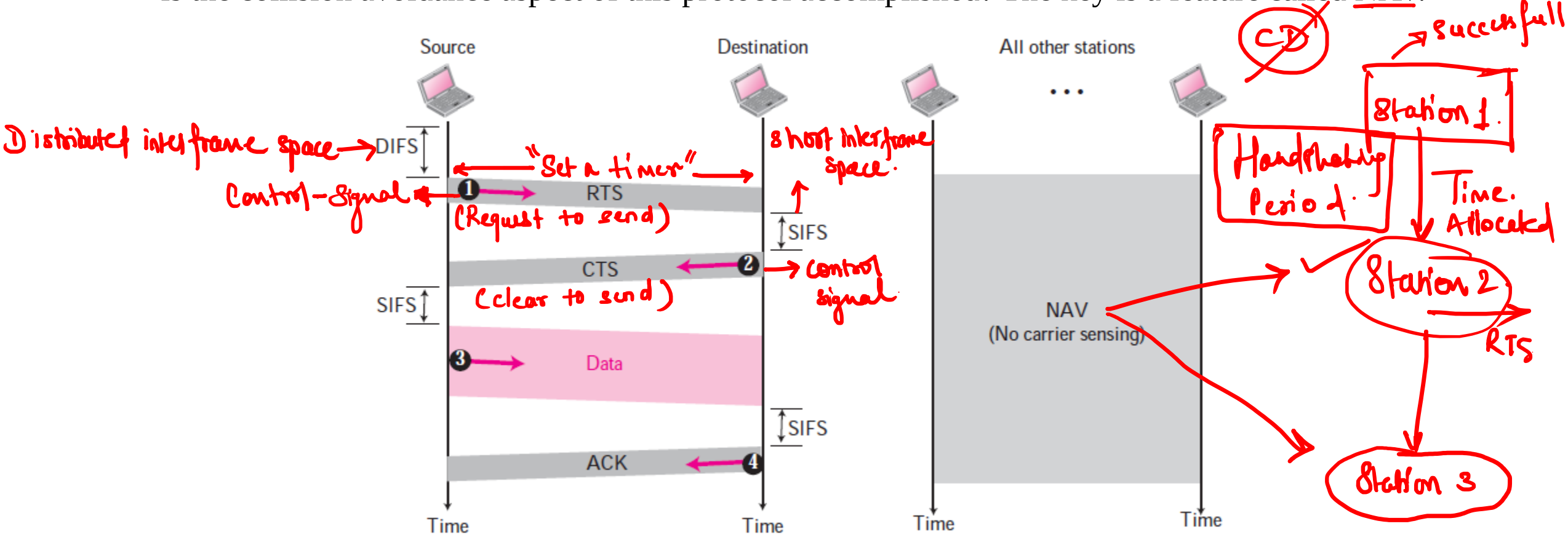
① Collision detection station can send & receive collision signals. This requires higher B.W.

② Collion may not be detected because of hidden station problem.

③ Due to large distance betn station (sending & receiving) there can be fading of signals. (collision signals).

IEEE 802.11

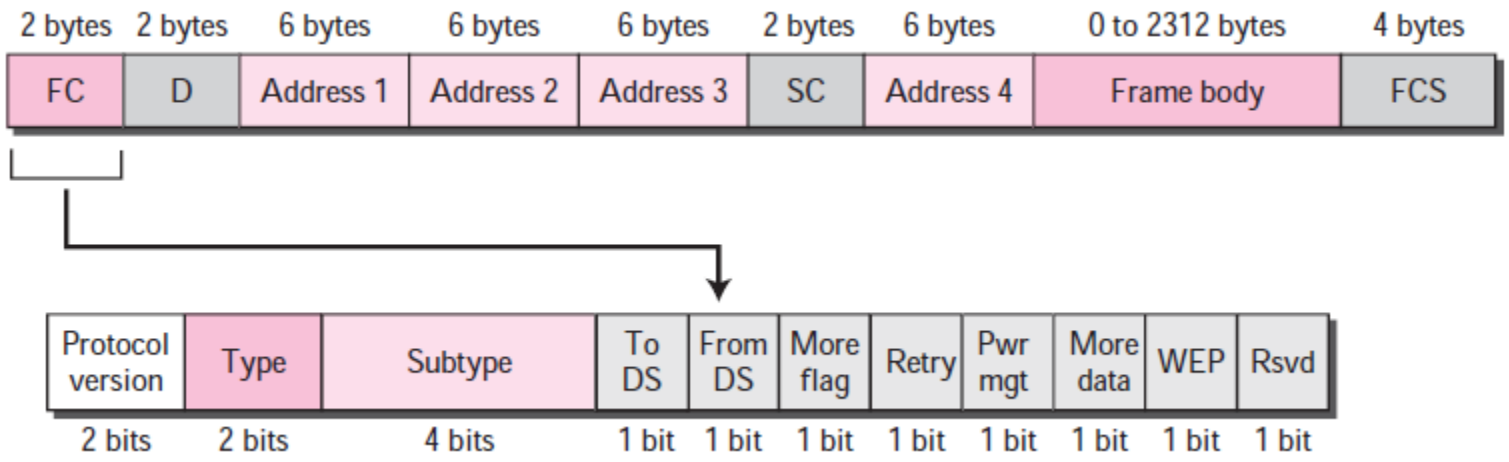
- **Frame Exchange Time Line:**
- Before sending a frame, the source station senses the medium by checking the energy level at the carrier frequency.
- **Network Allocation Vector:** (NAV)
- How do other stations defer sending their data if one station acquires access? In other words, how is the collision avoidance aspect of this protocol accomplished? The key is a feature called NAV.



IEEE 802.11

- **Fragmentation:**
- The wireless environment is very noisy; a corrupt frame has to be retransmitted. The protocol, therefore, recommends fragmentation—the division of a large frame into smaller ones. It is more efficient to resend a small frame than a large one.

- **Frame Format:**
- The MAC layer frame consists of nine fields:



IEEE 802.11

- **Frame Format:** The MAC layer frame consists of nine fields:

- **Frame control (FC):** The FC field is 2 bytes long and defines the type of frame and some control information. Table describes the subfields.

Field	*Explanation*
Version	Current version is 0
Type	Type of information: management (00), control (01), or data (10)
Subtype	Subtype of each type (see Table 3.6)
To DS	Defined later
From DS	Defined later
More flag	When set to 1, means more fragments
Retry	When set to 1, means retransmitted frame
Pwr mgt	When set to 1, means station is in power management mode
More data	When set to 1, means station has more data to send
WEP	Wired equivalent privacy (encryption implemented)
Rsvd	Reserved

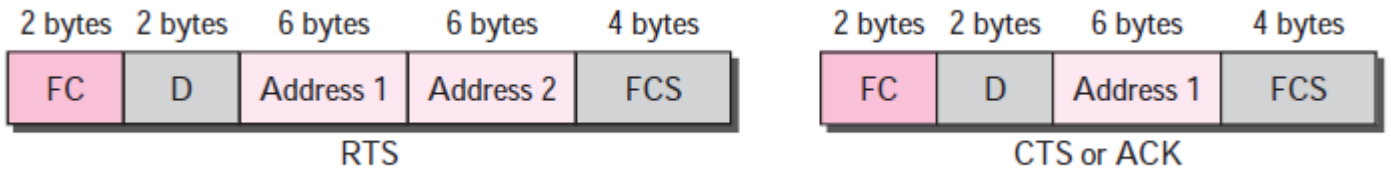
IEEE 802.11

- **D:** In all frame types except one, this field defines the duration of the transmission that is used to set the value of NAV. In one control frame, this field defines the ID of the frame.
- **Addresses:** There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the *To DS* and *From DS* subfields and will be discussed later.
- **Sequence control:** This field defines the sequence number of the frame to be used in flow control.
- **Frame body:** This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.
- **FCS:** The FCS field is 4 bytes long and contains a CRC-32 error detection sequence.

IEEE 802.11

- **Frame Types:** A wireless LAN defined by IEEE 802.11 has three categories of frames: management frames, control frames, and data frames.

- **Management Frames:** Management frames are used for the initial communication between stations and access points.
- **Control Frames:** Control frames are used for accessing the channel and acknowledging frames.



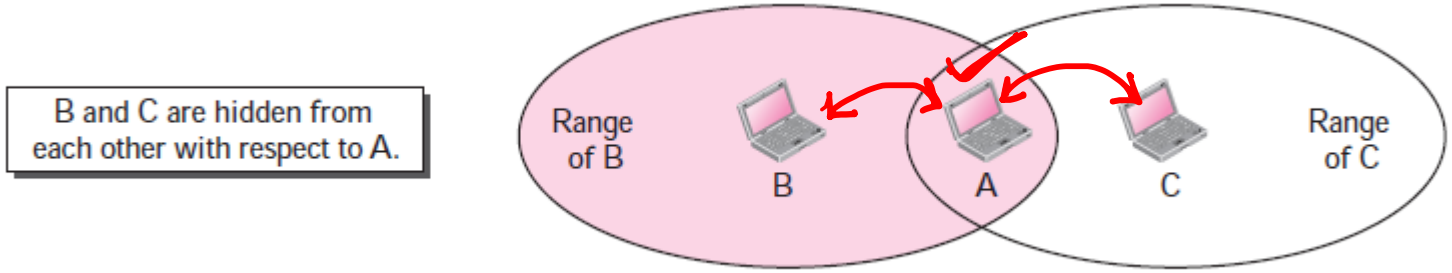
Control frames

- **Data Frames:** Data frames are used for carrying data and control information.

IEEE 802.11

- **Hidden and Exposed Station Problems:** We will refer to hidden and exposed station problems and their effects.

- **Hidden Station Problem:** Fig below shows an example of the hidden station problem.
- Station B has a transmission range shown by the left oval (sphere in space); every station in this range can hear any signal transmitted by station B. Station C has a transmission range shown by the right oval (sphere in space); every station located in this range can hear any signal transmitted by C.
- Station C is outside the transmission range of B; likewise, station B is outside the transmission range of C. Station A, however, is in the area covered by both B and C; it can hear any signal transmitted by B or C.

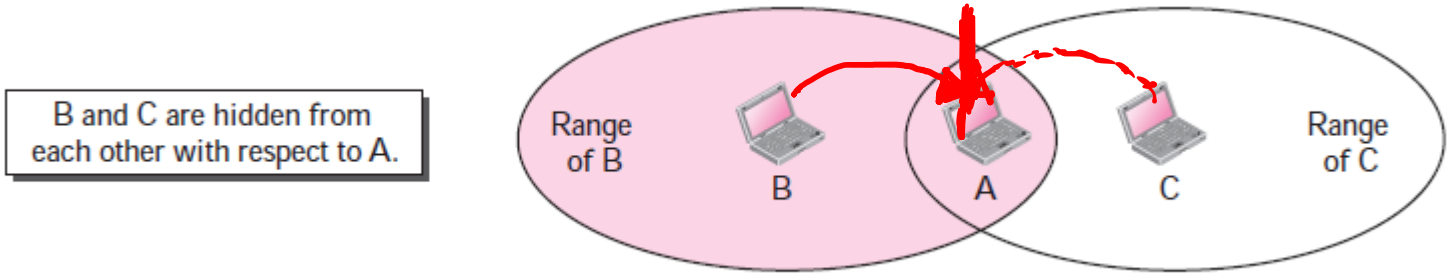


Hidden station problem

IEEE 802.11

- Assume that station B is sending data to station A. In the middle of this transmission, station C also has data to send to station A. However, station C is out of B's range and transmissions from B cannot reach C. Therefore C thinks the medium is free.
- Station C sends its data to A, which results in a collision at A because this station is receiving data from both B and C. In this case, we say that stations B and C are hidden from each other with respect to A. Hidden stations can reduce the capacity of the network because of the possibility of collision.

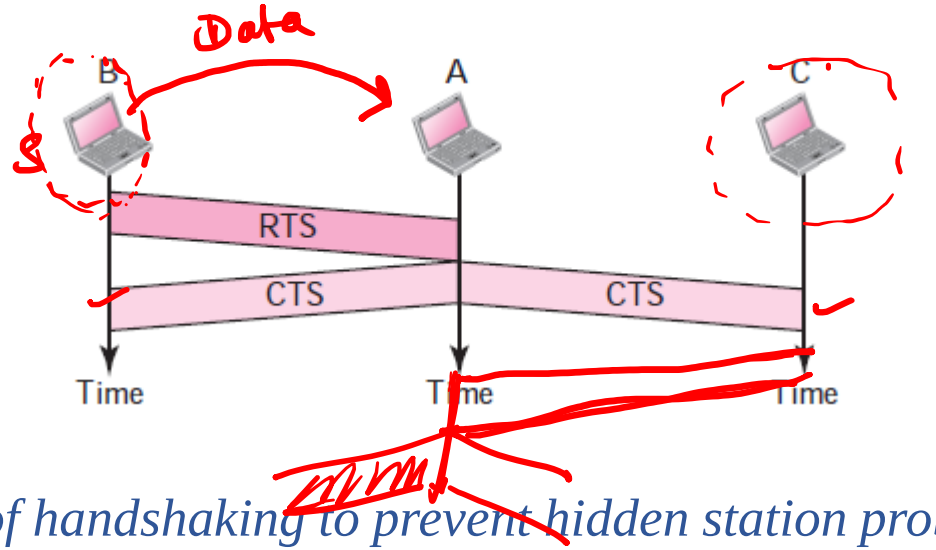
CSMA/CA.



Hidden station problem

IEEE 802.11

- **Hidden Station Solution:** The solution to the hidden station problem is the use of the handshake frames (RTS and CTS) that we discussed earlier. Fig below shows that the RTS message from B reaches A, but not C.
- However, because both B and C are within the range of A, the CTS message, which contains the duration of data transmission from B to A reaches C.
- Station C knows that some hidden station is using the channel and refrains from transmitting until that duration is over.



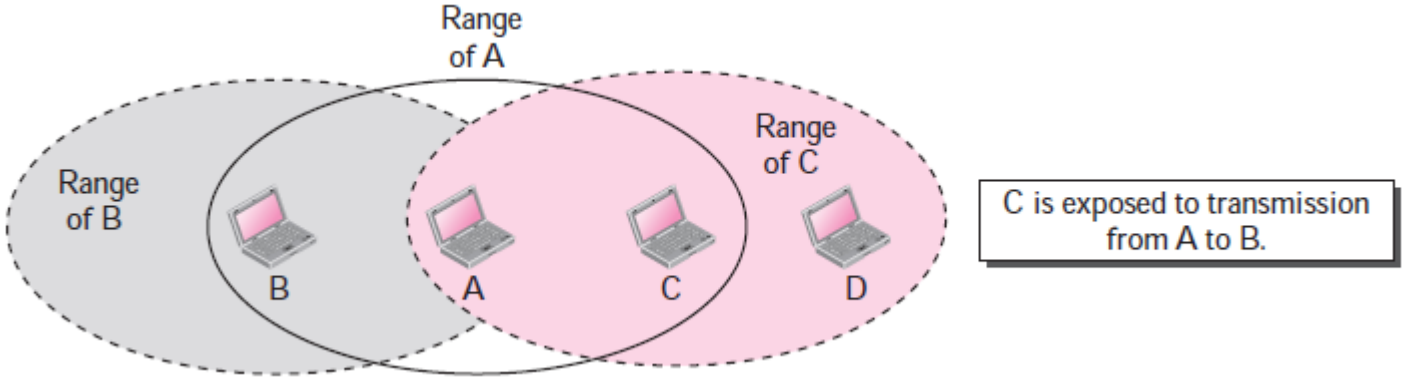
Use of handshaking to prevent hidden station problem

Note: The CTS frame in CSMA/CA handshake can prevent collision from a hidden station.

IEEE 802.11

- **Exposed Station Problem:** Fig below shows an example of the exposed station problem.

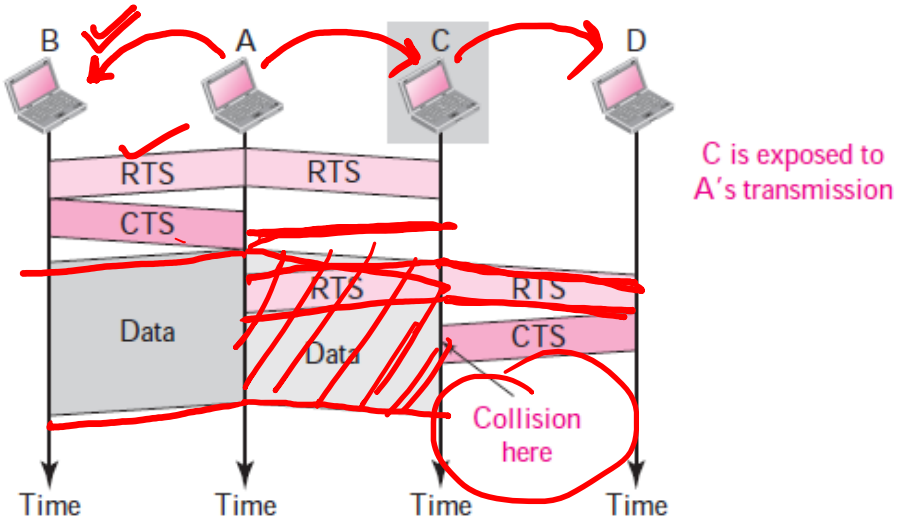
- In this problem a station refrains from using a channel when it is, in fact, available. In Fig, station A is transmitting to station B.
- Station C has some data to send to station D, which can be sent without interfering with the transmission from A to B. However, station C is exposed to transmission from A; it hears what A is sending and thus refrains from sending.
- In other words, C is too conservative and wastes the capacity of the channel.



Exposed station problem

IEEE 802.11

- **Exposed Station Problem:** The handshaking messages RTS and CTS cannot help in exposed station problem.
- Station C hears the RTS from A, but does not hear the CTS from B. Station C, after hearing the RTS from A, can wait for a time so that the CTS from B reaches A; it then sends an RTS to D to show that it needs to communicate with D.
- Both stations D and A may hear this RTS, but station A is in the sending state, not the receiving state. Station D, however, responds with a CTS. The problem is here.
- If station A has started sending its data, station C cannot hear the CTS from station D because of the collision; it cannot send its data to D. It remains exposed until A finishes sending its data.



Use of handshaking in exposed station problem

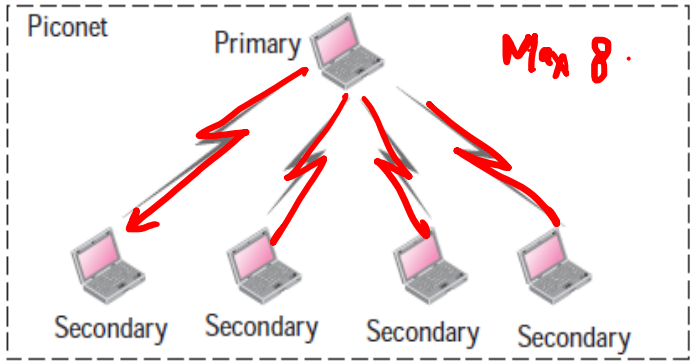
Bluetooth

- 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, sometimes called gadgets, find each other and make a network called a piconet.
- A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability.
- A Bluetooth LAN, by nature, cannot be large. If there are many gadgets that try to connect, there is chaos.
- 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: piconet and scatternet.

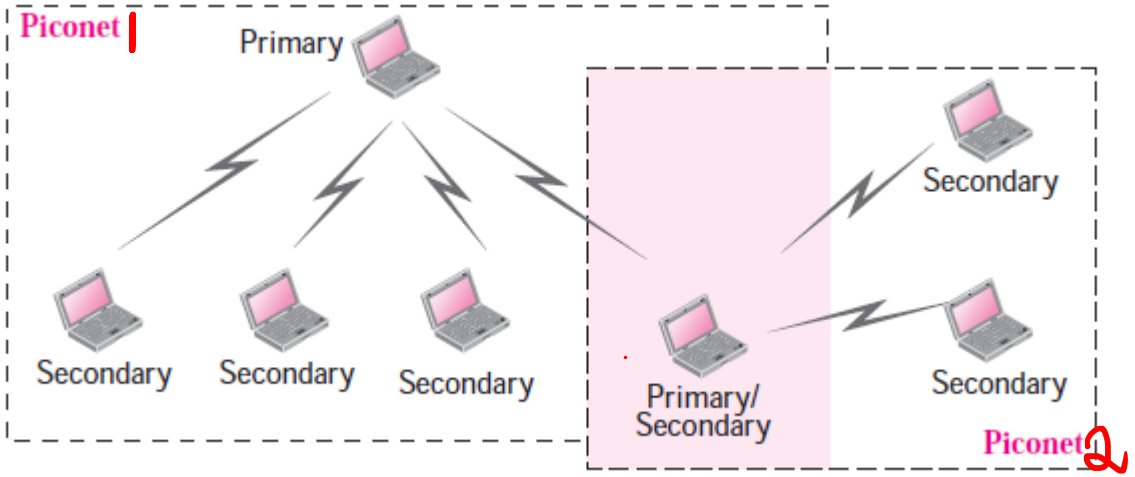
- **Piconets:**
- A Bluetooth network is called a piconet, or a small net. A piconet can have up to eight stations, one of which is called the primary; the rest are called secondaries.
- All the secondary stations synchronize their clocks and hopping sequence with the primary. Note that a piconet can have only one primary station. The communication between the primary and the secondary can be one-to-one or one-to-many. Figure shows a piconet.
- Although a piconet can have a maximum of seven secondaries, an additional eight secondaries can 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.



Bluetooth

- **Architecture:** Bluetooth defines two types of networks: piconet and scatternet.

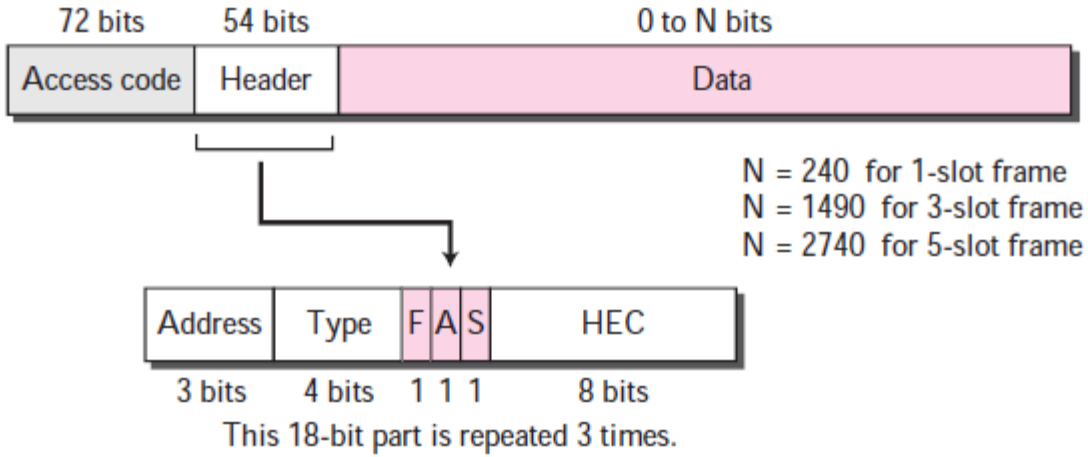
- **Scatternet:**
- Piconets can be combined to form what is called a scatternet. A secondary station in one piconet can be the primary in another piconet. This station can receive messages from the primary in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in the second piconet. A station can be a member of two piconets. Figure illustrates a scatternet.



Bluetooth

Frame Format:

- **Access code.** This 72-bit field normally contains synchronization bits and the identifier of the primary to distinguish the frame of one piconet from another.



Bluetooth

- **Header.** This 54-bit field is a repeated 18-bit pattern. Each pattern has the following subfields:
    - **Address:** The 3-bit address subfield can define up to seven secondaries (1 to 7). If the address is zero, it is used for broadcast communication from the primary to all secondaries.
    - **Type:** The 4-bit type subfield defines the type of data coming from the upper layers. We discuss these types later.
    - **F:** This 1-bit subfield is for flow control. When set (1), it indicates that the device is unable to receive more frames (buffer is full).
    - **A:** This 1-bit subfield is for acknowledgment. Bluetooth uses stop-and-wait ARQ; 1 bit is sufficient for acknowledgment.
    - **S:** This 1-bit subfield holds a sequence number. Bluetooth uses stop-and-wait ARQ; 1 bit is sufficient for sequence numbering.
    - **HEC:** The 8-bit header error correction subfield is a checksum to detect errors in each 18-bit header section.

- **Data:** This subfield can be 0 to 2740 bits long. It contains data or control information coming from the upper layers.

Bluetooth

- **Data:** This subfield can be 0 to 2740 bits long. It contains data or control information coming from the upper layers.

  a. The sending station, after sensing that the medium is idle, sends a special small frame called request to send (RTS). In this message, the sender defines the total time it needs the medium.

  b. The receiver acknowledges the request (broadcast to all stations) by sending a small packet called clear to send (CTS).

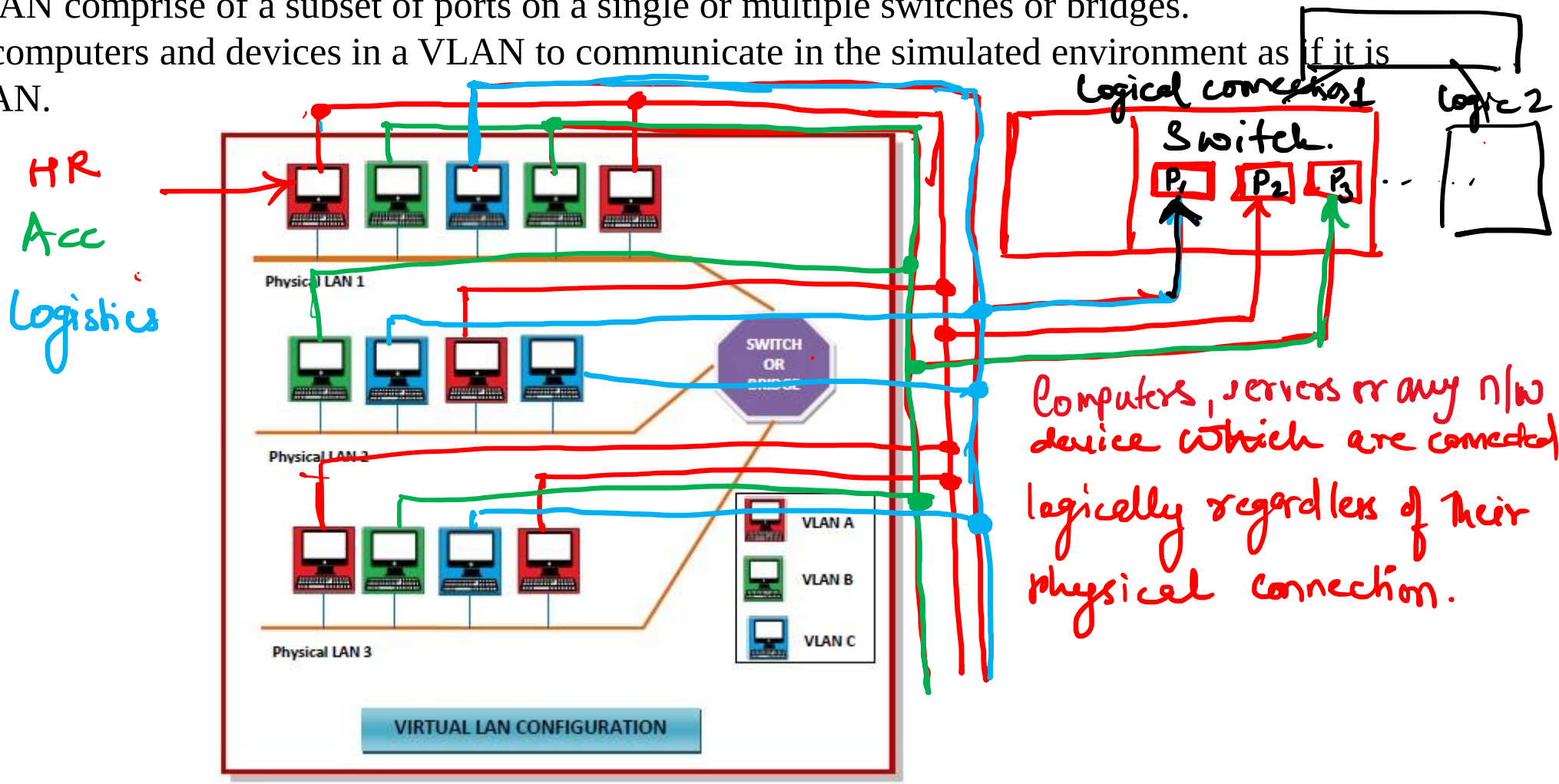
  c. The sender sends the data frame.

  d. The receiver acknowledges the receipt of data.

Virtual LANS

Introduction

- Virtual Local Area Networks or Virtual LANs (VLANs) are a logical group of computers that appear to be on the same LAN irrespective of the configuration of the underlying physical network.
- Network administrators partition the networks to match the functional requirements of the VLANs so that each VLAN comprise of a subset of ports on a single or multiple switches or bridges.
- This allows computers and devices in a VLAN to communicate in the simulated environment as if it is a separate LAN.

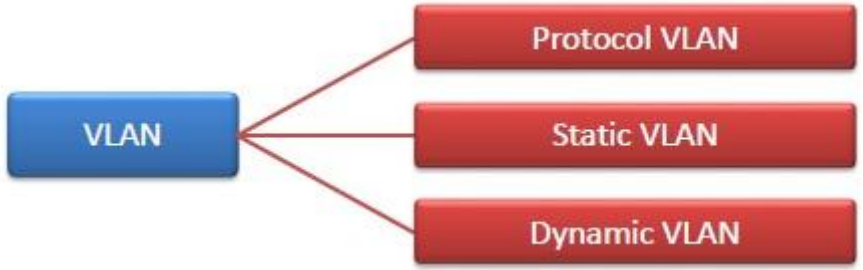


Features of VLANs

- A VLAN forms sub-network grouping together devices on separate physical LANs.
- VLAN's help the network manager to segment LANs logically into different broadcast domains.
- VLANs function at layer 2, i.e. Data Link Layer of the OSI model.
- There may be one or more network bridges or switches to form multiple, independent VLANs.
- Using VLANs, network administrators can easily partition a single switched network into multiple networks depending upon the functional and security requirements of their systems.
- VLANs eliminate the requirement to run new cables or reconfiguring physical connections in the present network infrastructure.
- VLANs help large organizations to re-partition devices aiming improved traffic management.
- VLANs also provide better security management allowing partitioning of devices according to their security criteria and also by ensuring a higher degree of control connected devices.
- VLANs are more flexible than physical LANs since they are formed by logical connections. This aids is quicker and cheaper reconfiguration of devices when the logical partitioning needs to be changed.

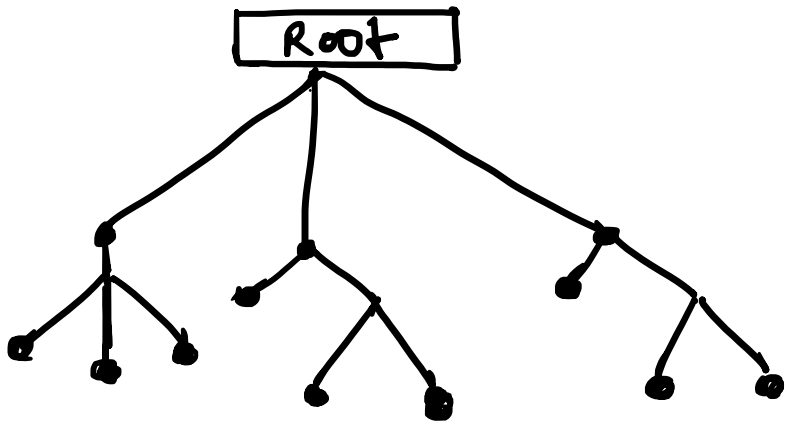
Types of VLANs

- Protocol VLAN − Here, the traffic is handled based on the protocol used. A switch or bridge segregates, forwards or discards frames the come to it based upon the traffics protocol.
- Port-based VLAN − This is also called static VLAN. Here, the network administrator assigns the ports on the switch / bridge to form a virtual network.
- Dynamic VLAN − Here, the network administrator simply defines network membership according to device characteristics.



Virtual LAN Protocol and Design

- The VLAN protocol and design are based on IEEE 802. 1Q, where IEEE 802.1 format is a group of multiplexed VLAN that supports VLAN multivendor. Other protocols like ISC (Cisco Inter-switch link) and VLT (Virtual LAN trunk) are also implemented on VLANs to carry the data of VLAN in a frame header of IEEE 802.10.
- Spanning tree protocol (STP) creates loop-free network topology on each data link layer (layer 2) of the OSI model among the switch links. This type of virtual LAN protocol and design is used to minimize the total STP of similar network topologies or multi-instance STP (MISTP) and blocks the connections of loop propagations of data. It is mainly used when the breakdown occurs in another part of the network during the transmission of data packets through a selected switch or route.



Advantages

- The use of VLAN reduces the traffic of broadcast and multicasting domains in computer networks. So, that data packets easily reach the destinations.
- Enhances performance.
- It forms the workgroups and logical groups virtually to increase the communication between the users within the workgroups
- Easy to control and manage the VLAN's because there is no need to reconfigure the routers reduces the use of devices on the network topology and cablings.
- Reduces the installation and maintenance cost
- It increases the security during the broadcasting of sensitive data on the computer network by controlling firewalls, intrusions, and restrict access to the network.
- Easy to resolve all the broadcasting issues and reduce the size of the domains of broadcast on the network.

Thankyou

Unit-3

The Data Link Layer

Content:

1. The data link layer issues
2. Error detection and correction
3. Elementary data link protocols
4. Sliding window protocols
5. Examples of data link protocol: HDLC

I. Introduction-

In this chapter we will study the design principles for layer 2, the data link layer. This study deals with the algorithms for achieving reliable, efficient communication between two adjacent machines at the data link layer. By adjacent, we mean that the two machines are connected by a communication channel that acts conceptually like a wire (e.g., a coaxial cable, telephone line, or point-to-point wireless channel). The essential property of a channel that makes it ''wire-like'' is that the bits are delivered in exactly the same order in which they are sent.

After an introduction to the key design issues present in the data link layer, we will start our study of its protocols by looking at the nature of errors, their causes, and how they can be detected and corrected. Then we will study a series of increasingly complex protocols, each one solving more and more of the problems present in this layer. Finally, we will conclude with an examination of protocol modeling and correctness and give some examples of data link protocols.

3.1 Data Link Layer Design Issues:

The data link layer has a number of specific functions it can carry out. These functions include:

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.

To accomplish these goals, the data link layer takes the packets it 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, as illustrated in Fig. 3-1.

Frame management forms the heart of what the data link layer does. In the following sections we will examine all the above-mentioned issues in detail.

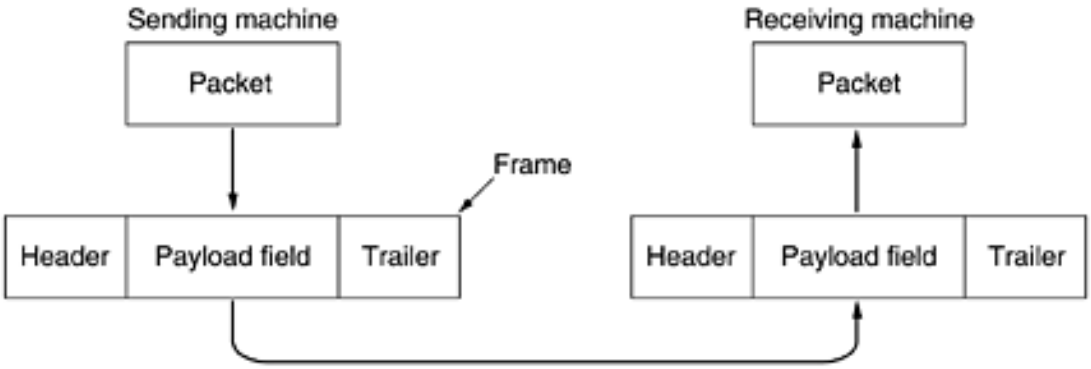


Figure 3-1. Relationship between packets and frames.

3.1.1 Services Provided to the Network Layer:

The function of the data link layer is to provide services to the network layer.

The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine.

On the source machine is an entity, call it a process, in the network layer that hands some bits to the data link layer for transmission to the destination.

The job of the data link layer is to transmit the bits to the destination machine so they can be handed over to the network layer there, as shown in Fig. 3-2(a).

The actual transmission follows the path of Fig. 3-2(b), but it is easier to think in terms of two data link layer processes communicating using a data link protocol. For this reason, we will implicitly use the model of Fig. 3-2(a) throughout this chapter.

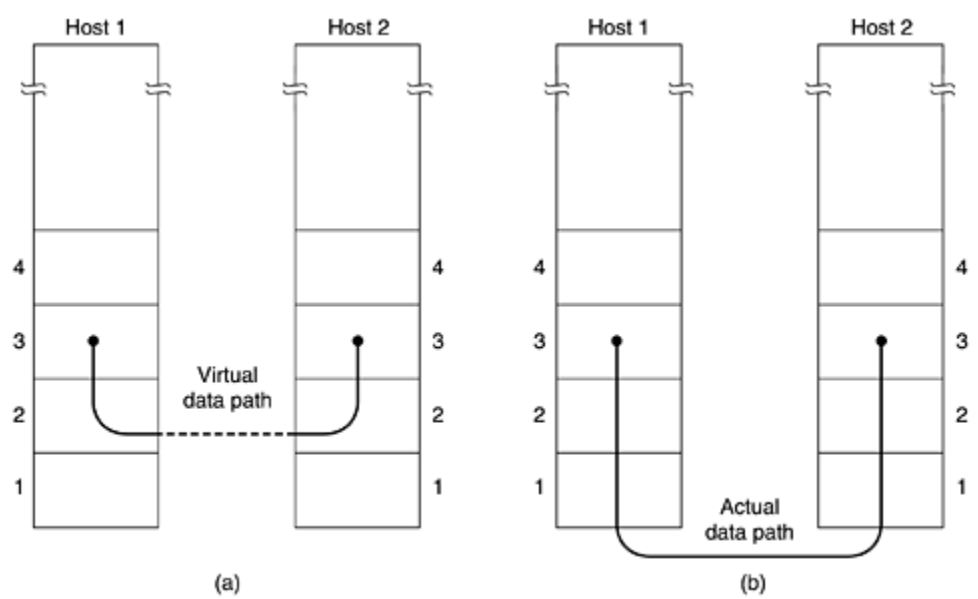


Figure 3-2. (a) Virtual communication. (b) Actual communication.

The data link layer can be designed to offer various services. The actual services offered can vary from system to system. Three reasonable possibilities that are commonly provided are

1. Unacknowledged connectionless service.
2. Acknowledged connectionless service.
3. Acknowledged connection-oriented service.

- **Unacknowledged connectionless service:**

Unacknowledged connectionless service consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them.

No logical connection is established beforehand or released afterward.

If a frame is lost due to noise on the line, no attempt is made to detect the loss or recover from it in the data link layer.

This class of service is appropriate when the error rate is very low so that recovery is left to higher layers.

- **Acknowledged connectionless service:**

When this service is offered, there are still no logical connections used, but each frame sent is individually acknowledged.

In this way, the sender knows whether a frame has arrived correctly.

If it has not arrived within a specified time interval, it can be sent again. This service is useful over unreliable channels, such as wireless systems.

It is perhaps worth emphasizing that providing acknowledgements in the data link layer is just an optimization, never a requirement.

The network layer can always send a packet and wait for it to be acknowledged.

If the acknowledgement is not forthcoming before the timer expires, the sender can just send the entire message again.

If individual frames are acknowledged and retransmitted, entire packets get through much faster.

On reliable channels, such as fiber, the overhead of a heavyweight data link protocol may be unnecessary, but on wireless channels, with their inherent unreliability, it is well worth the cost.

- **Acknowledged connection-oriented service:**

The most sophisticated service the data link layer can provide to the network layer is connection-oriented service.

With this service, the source and destination machines establish a connection before any data are transferred.

Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received.

Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order.

With connectionless service, in contrast, it is conceivable that a lost acknowledgement causes a packet to be sent several times and thus received several times.

Connection-oriented service, in contrast, provides the network layer processes with the equivalent of a reliable bit stream.

When connection-oriented service is used, transfers go through three distinct phases.

In the first phase, the connection is established by having both sides initialize variables and counters needed to keep track of which frames have been received and which ones have not.

In the second phase, one or more frames are actually transmitted. In the third and final phase, the connection is released, freeing up the variables, buffers, and other resources used to maintain the connection.

3.1.2 Framing:

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination.

This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to detect and, if necessary, correct errors.

The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame.

When a frame arrives at the destination, the checksum is recomputed. If the newly-computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report).

Breaking the bit stream up into frames is more difficult than it at first appears. One way to achieve this framing is to insert time gaps between frames, much like the spaces between words in ordinary text.

However, networks rarely make any guarantees about timing, so it is possible these gaps might be squeezed out or other gaps might be inserted during transmission.

Since it is too risky to count on timing to mark the start and end of each frame, other methods have been devised. In this section we will look at four methods:

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

The first framing method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is.

This technique is shown in Fig. 3-4(a) for four frames of sizes 5, 5, 8, and 8 characters, respectively.

The trouble with this algorithm is that the count can be garbled by a transmission error.

For example, if the character count of 5 in the second frame of Fig. 3-4(b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts.

Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission.

For this reason, the character count method is rarely used anymore.

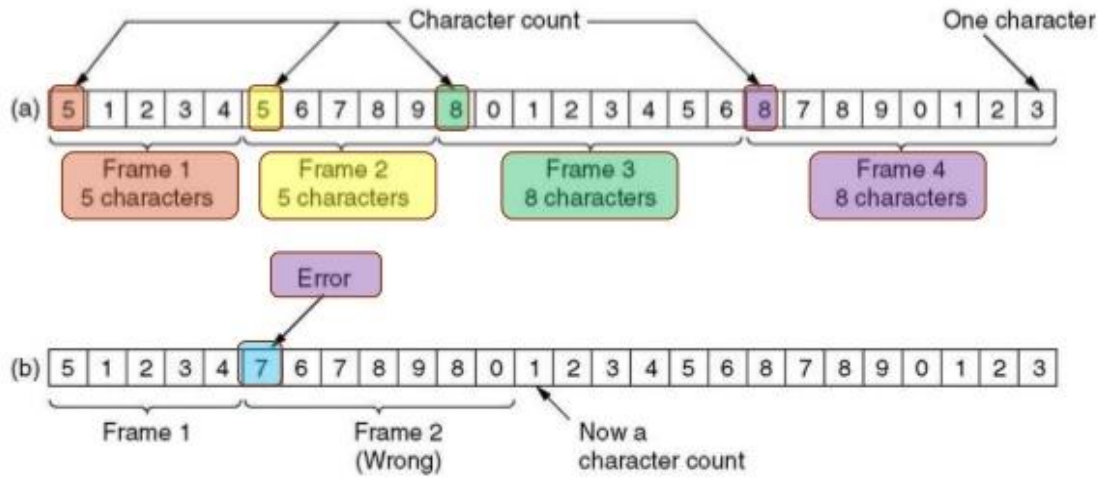


Figure 3-4. A character stream. (a) Without errors. (b) With one error.

The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes.

In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in Fig. 3-5(a) as FLAG.

In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.

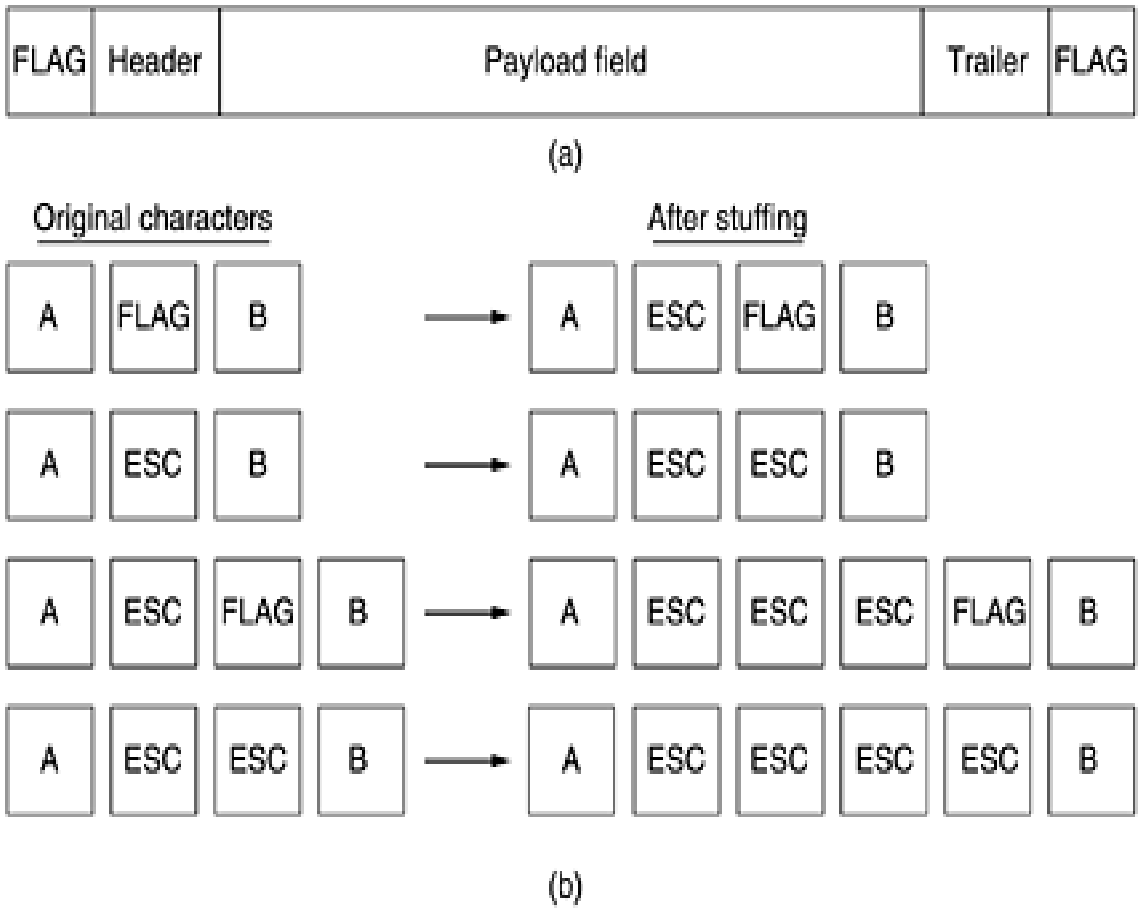


Figure 3-5. (a) A frame delimited by flag bytes. (b) Four examples of byte sequences before and after byte stuffing.

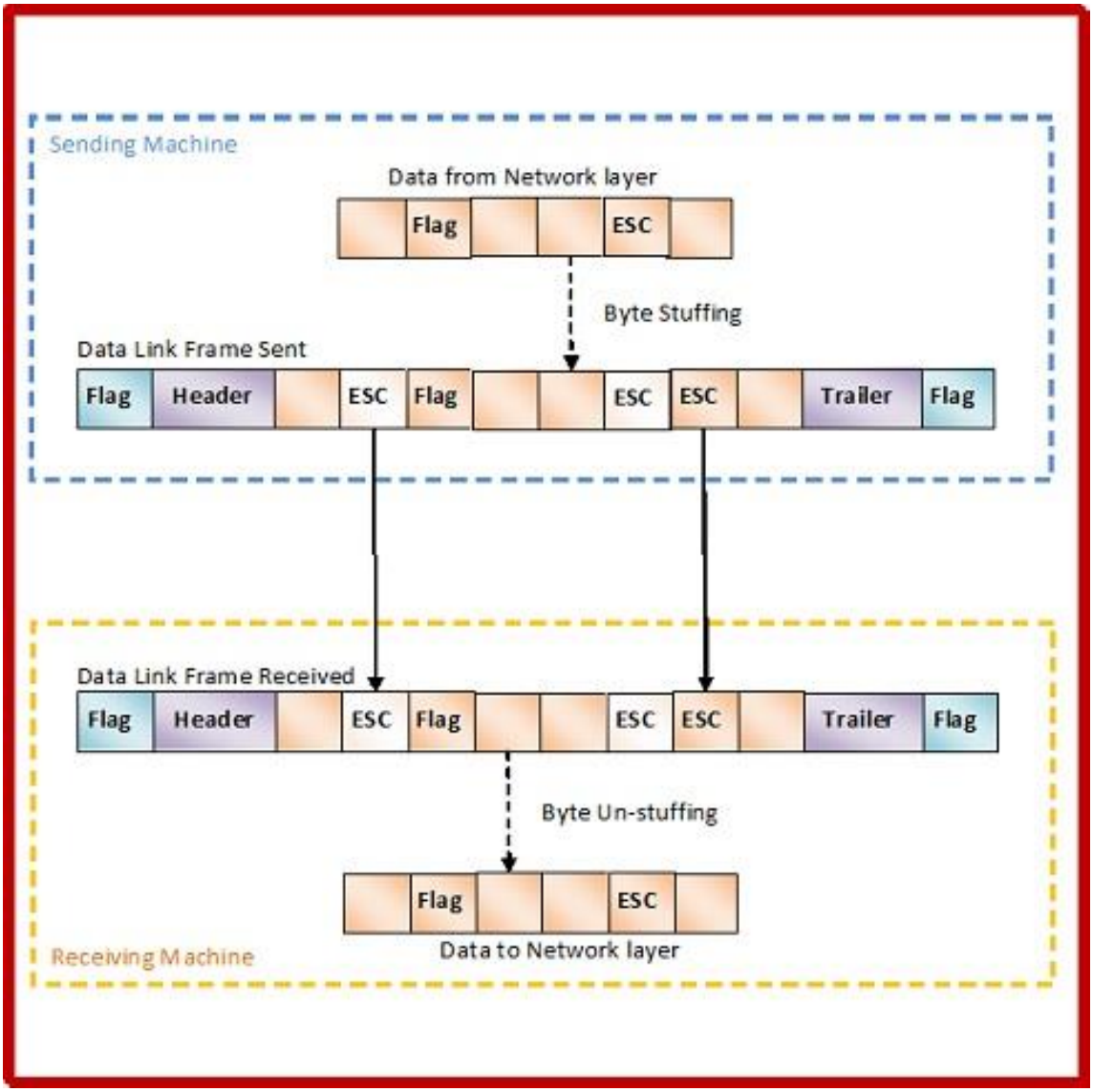


Figure 3-5. (c) Byte stuffing mechanism

A serious problem occurs with this method when binary data, such as object programs or floating-point numbers, are being transmitted. It may easily happen that the flag byte's bit pattern occurs in the data. This situation will usually interfere with the framing.

One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each ''accidental'' flag byte in the

data. The data link layer on the receiving end removes the escape byte before the data are given to the network layer.

This technique is called byte stuffing or character stuffing. Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

Of course, the next question is: What happens if an escape byte occurs in the middle of the data? The answer is that it, too, is stuffed with an escape byte.

Thus, any single escape byte is part of an escape sequence, whereas a doubled one indicates that a single escape occurred naturally in the data. Some examples are shown in Fig. 3-5(b).

In all cases, the byte sequence delivered after de-stuffing is exactly the same as the original byte sequence.

The new technique allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character.

It works like this. Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte).

Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream.

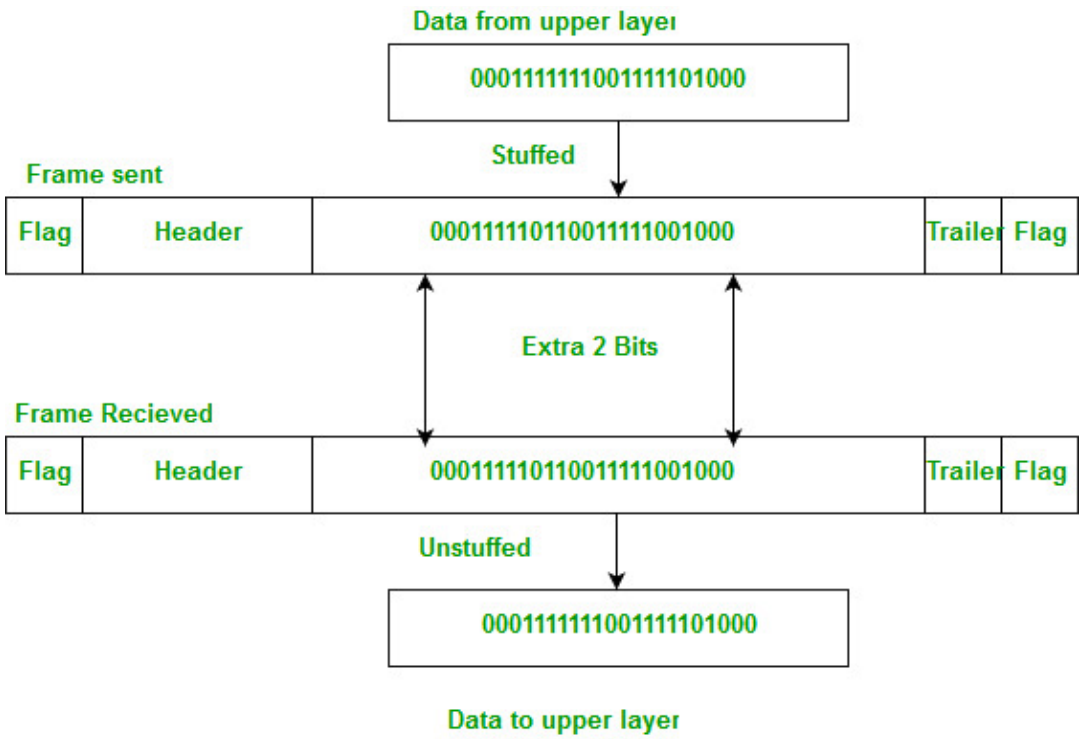
This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de-stuffs (i.e., deletes) the 0 bit.

Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing.

If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

Figure 3.6 (a)Explains Bit Stuffing Mechanism (b) gives an example of bit stuffing.



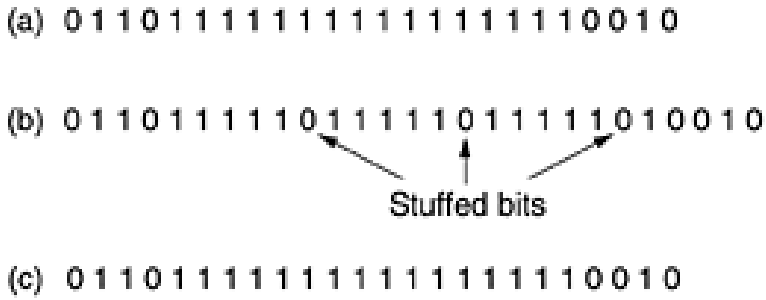


Figure 3-6. (a) Bit Stuffing Mechanism (b) An example of Bit Stuffing.

With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern.

Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

The last method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy.

As a final note on framing, many data link protocols use a combination of a character count with one of the other methods for extra safety.

When a frame arrives, the count field is used to locate the end of the frame.

Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter.

3.1.3 Error Control

Error control in data link layer is the process of detecting and correcting data frames that have been corrupted or lost during transmission.

The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line.

Typically, the protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames.

If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely.

On the other hand, a negative acknowledgement means that something has gone wrong, and the frame must be transmitted again.

In case of lost or corrupted frames, the receiver does not receive the correct data-frame and sender is ignorant about the loss.

Data link layer follows a technique to detect transit errors and take necessary actions, which is retransmission of frames whenever error is detected or frame is lost. The process is called Automatic Repeat Request (ARQ).

Phases in Error Control

The error control mechanism in data link layer involves the following phases –

- **Detection of Error –** Transmission error, if any, is detected by either the sender or the receiver.
- **Acknowledgment –** acknowledgment may be positive or negative.
    - **Positive ACK –** On receiving a correct frame, the receiver sends a positive acknowledge.
    - **Negative ACK –** On receiving a damaged frame or a duplicate frame, the receiver sends a negative acknowledgment back to the sender.
- **Retransmission –** The sender maintains a clock and sets a timeout period. If an acknowledgment of a data-frame previously transmitted does not arrive before the timeout, or a negative acknowledgment is received, the sender retransmits the frame.

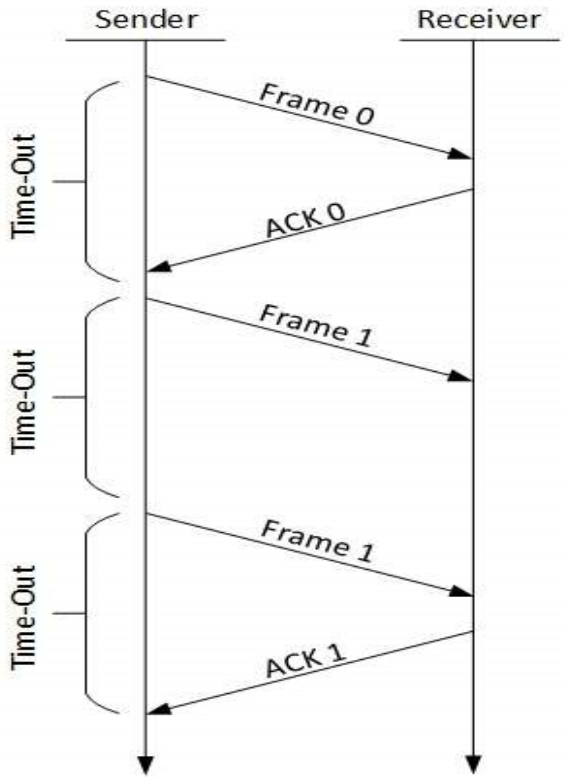
Error Control Techniques

There are three main techniques for error control –

- **Stop and Wait ARQ**

  This protocol involves the following transitions −

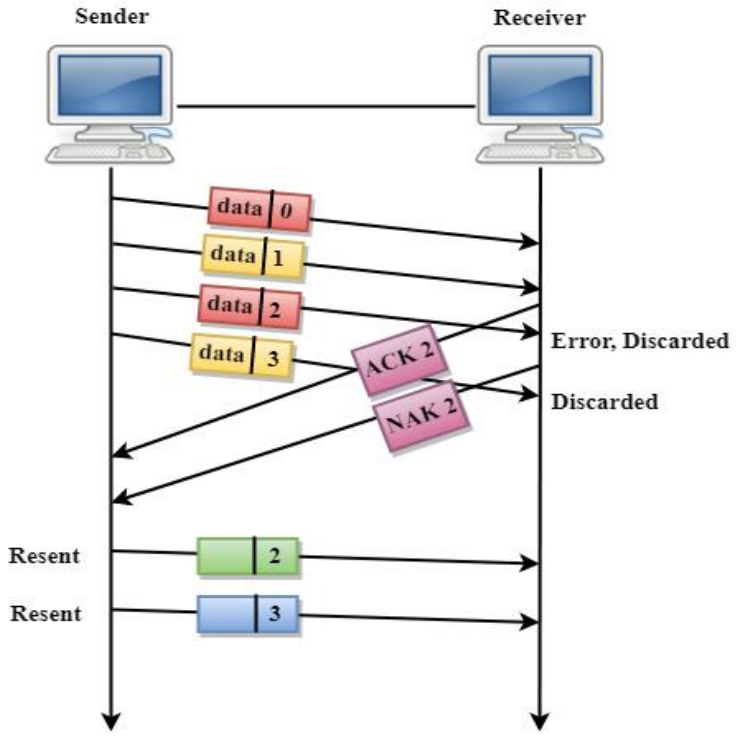
  - A timeout counter is maintained by the sender, which is started when a frame is sent.
  - If the sender receives acknowledgment of the sent frame within time, the sender is confirmed about successful delivery of the frame. It then transmits the next frame in queue.
  - If the sender does not receive the acknowledgment within time, the sender assumes that either the frame or its acknowledgment is lost in transit. It then retransmits the frame.
  - If the sender receives a negative acknowledgment, the sender retransmits the frame.



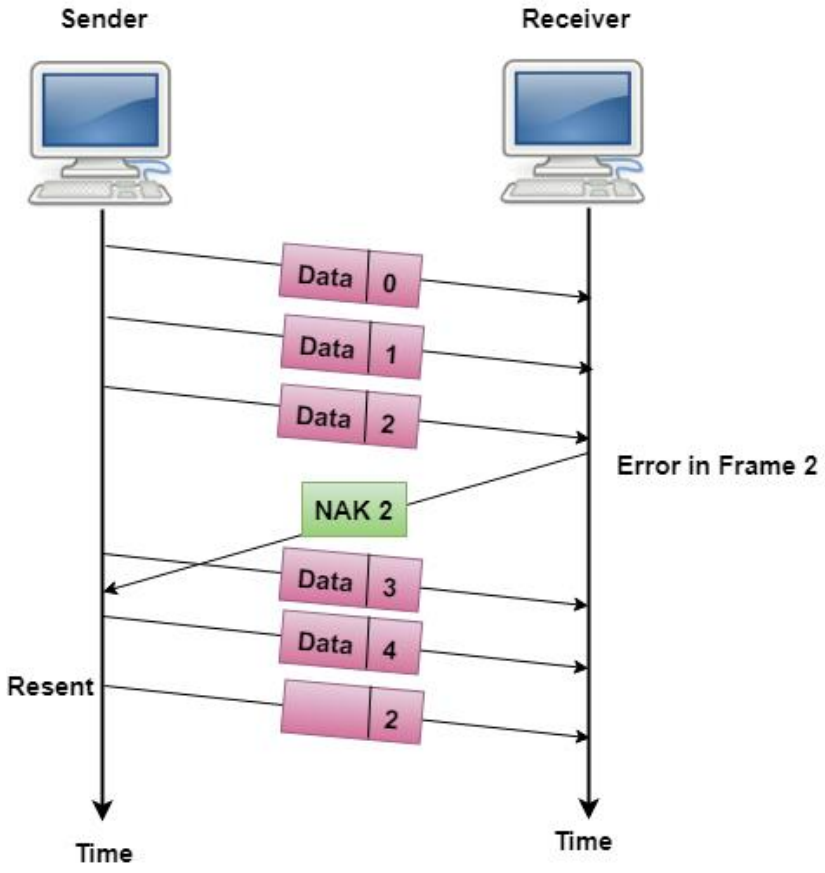
- **Go-Back-N ARQ**

  The working principle of this protocol is –

  - The sender has buffers called sending window.The sender sends multiple frames based upon the sending-window size, without receiving the acknowledgment of the previous ones.
  - The receiver receives frames one by one. It keeps track of incoming frame's sequence number and sends the corresponding acknowledgment frames.
  - After the sender has sent all the frames in window, it checks up to what sequence number it has received positive acknowledgment.
  - If the sender has received positive acknowledgment for all the frames, it sends next set of frames.
  - If sender receives 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.



- **Selective Repeat ARQ**
    - Both the sender and the receiver have buffers called sending window and receiving window respectively.
    - The sender sends multiple frames based upon the sending-window size, without receiving the acknowledgment of the previous ones.
    - The receiver also receives multiple frames within the receiving window size.
    - The receiver keeps track of incoming frame's sequence numbers, buffers the frames in memory.
    - It sends ACK for all successfully received frames and sends NACK for only frames which are missing or damaged.
    - The sender in this case, sends only packet for which NACK is received.



3.1.4 Flow Control:

Another important design issue that occurs in the data link layer (and higher layers as well) is what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them.

This situation can easily occur when the sender is running on a fast (or lightly loaded) computer and the receiver is running on a slow (or heavily loaded) machine.

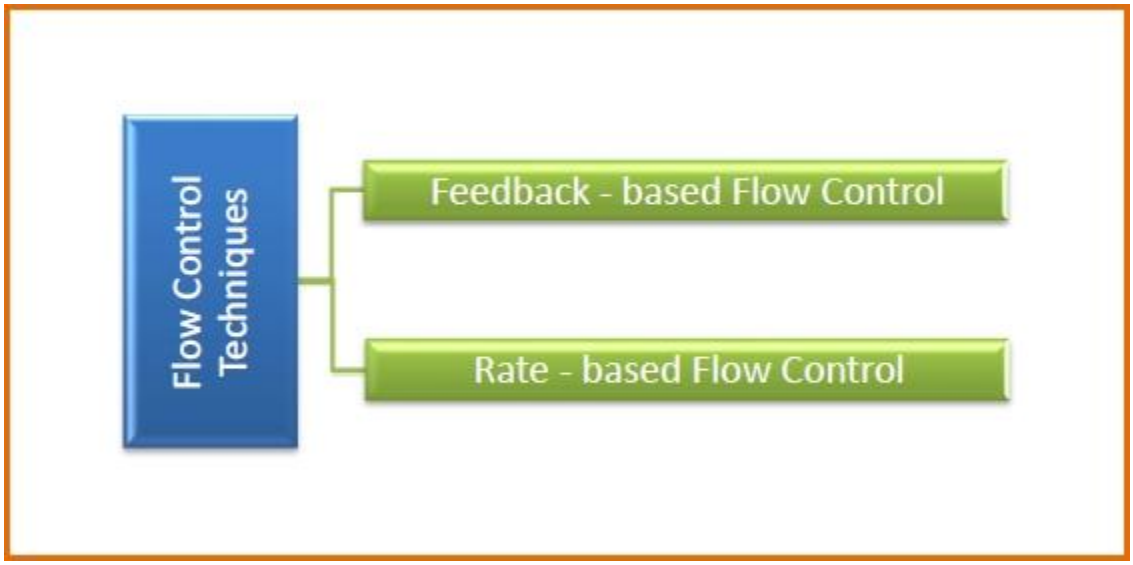
The sender keeps pumping the frames out at a high rate until the receiver is completely swamped.

Even if the transmission is error free, at a certain point the receiver will simply be unable to handle the frames as they arrive and will start to lose some.

In data link layer, flow control restricts the number of frames the sender can send before it waits for an acknowledgment from the receiver.

Approaches of Flow Control

Flow control can be broadly classified into two categories −



Feedback based Flow Control:

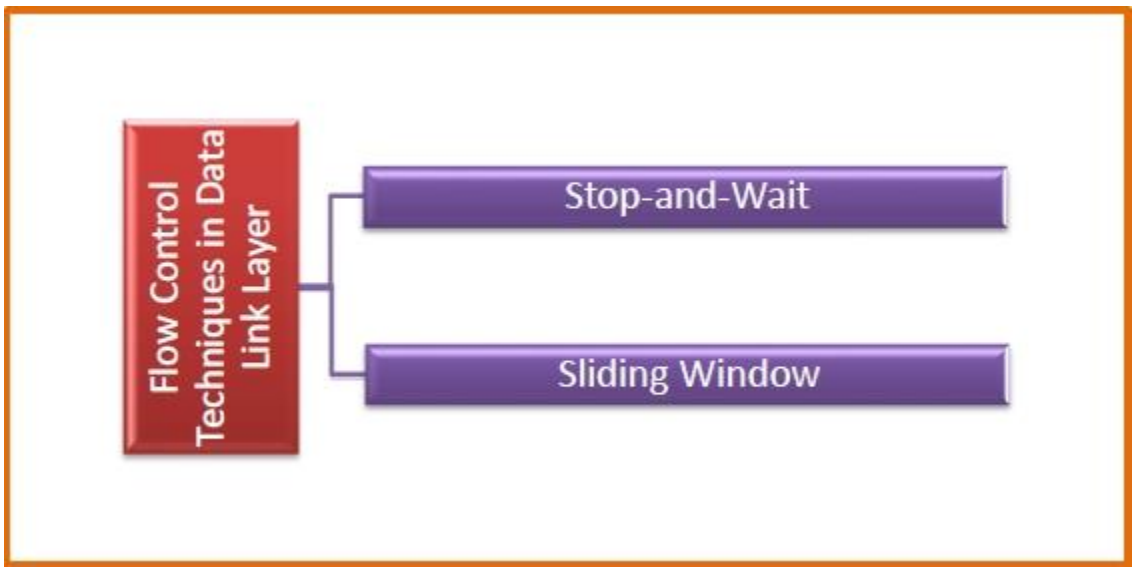
In the first one, feedback-based flow control, the receiver sends back information to the sender giving it permission to send more data or at least telling the sender how the receiver is doing.

Rate based Flow Control:

In the second one, rate-based flow control, the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver.

Flow Control Techniques in Data Link Layer:

Data link layer uses feedback based flow control mechanisms. There are two main techniques –



Stop and Wait

This protocol involves the following transitions –

- The sender sends a frame and waits for acknowledgment.
- Once the receiver receives the frame, it sends an acknowledgment frame back to the sender.

- On receiving the acknowledgment frame, the sender understands that the receiver is ready to accept the next frame. So it sender the next frame in queue.

Sliding Window

This protocol improves the efficiency of stop and waits protocol by allowing multiple frames to be transmitted before receiving an acknowledgment.

The working principle of this protocol can be described as follows −

Both the sender and the receiver have finite sized buffers called windows. The sender and the receiver agree upon the number of frames to be sent based upon the buffer size.

The sender sends multiple frames in a sequence, without waiting for acknowledgment. When its sending window is filled, it waits for acknowledgment.

On receiving acknowledgment, it advances the window and transmits the next frames, according to the number of acknowledgments received.

3.2 Error Detection and Correction:

There are many reasons such as noise, cross-talk etc., which may help data to get corrupted during transmission. The upper layers work on some generalized view of network architecture and are not aware of actual hardware data processing. Hence, the upper layers expect error-free transmission between the systems. Most of the applications would not function expectedly if they receive erroneous data. Applications such as voice and video may not be that affected and with some errors they may still function well.

Data-link layer uses error control techniques to ensure that frames, i.e. bit streams of data, are transmitted from the source to the destination with a certain extent of accuracy.

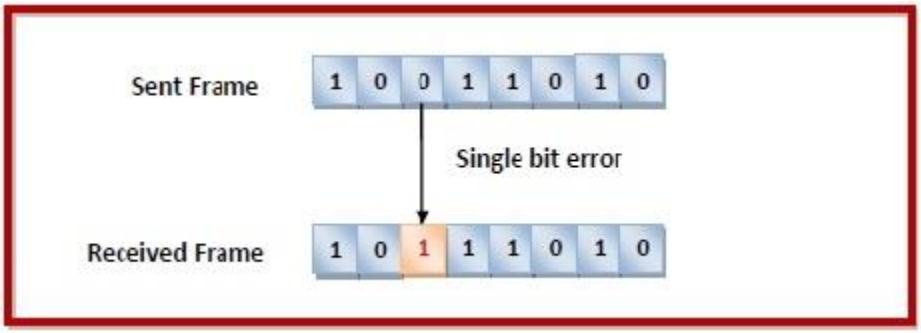
Errors

When bits are transmitted over the computer network, they are subject to get corrupted due to interference and network problems. The corrupted bits leads to spurious data being received by the destination and are called errors.

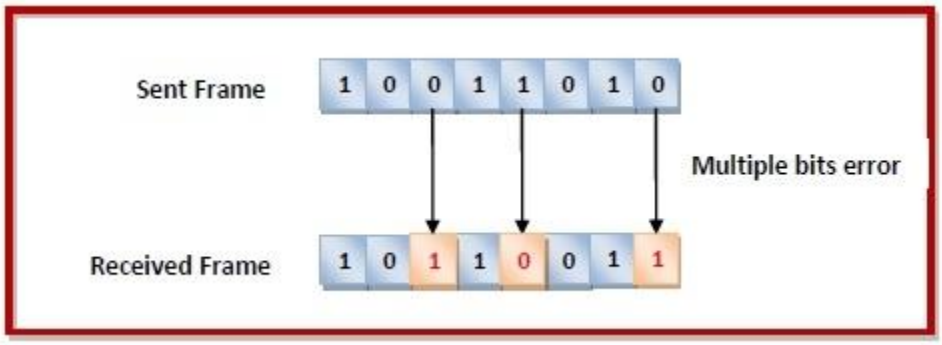
Types of Errors

Errors can be of three types, namely single bit errors, multiple bit errors, and burst errors.

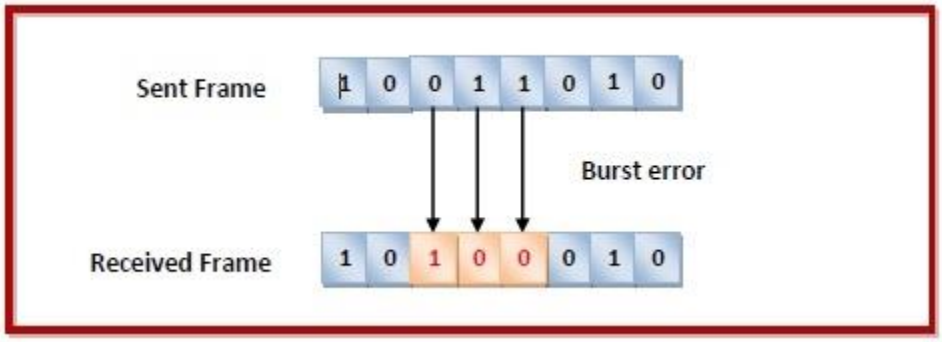
- **Single bit error** – In the received frame, only one bit has been corrupted, i.e. either changed from 0 to 1 or from 1 to 0.



- **Multiple bits error** – In the received frame, more than one bits are corrupted.



- **Burst error** – In the received frame, more than one consecutive bits are corrupted.



Error Control

Error control can be done in two ways:

- **Error detection** − Error detection involves checking whether any error has occurred or not. The number of error bits and the type of error does not matter.
- **Error correction** − Error correction involves ascertaining the exact number of bits that has been corrupted and the location of the corrupted bits.

  For both error detection and error correction, the sender needs to send some additional bits along with the data bits. The receiver performs necessary checks based upon the additional redundant bits. If it finds that the data is free from errors, it removes the redundant bits before passing the message to the upper layers.

Error Detection Techniques

There are three main techniques for detecting errors in frames:

- Parity Check
- Checksum
- Cyclic Redundancy Check (CRC).

Parity Check

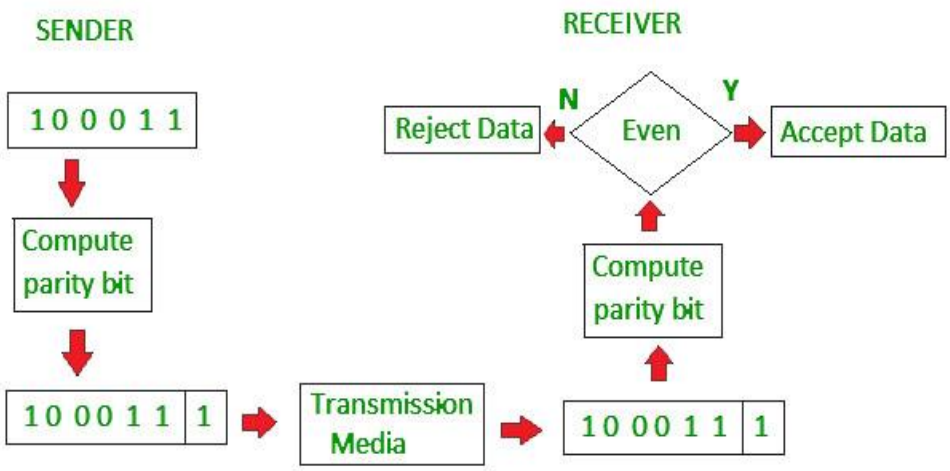
The parity check is done by adding an extra bit, called parity bit to the data to make a number of 1s either even in case of even parity or odd in case of odd parity.

While creating a frame, the sender counts the number of 1s in it and adds the parity bit in the following way

- In case of even parity: If a number of 1s is even then parity bit value is 0. If the number of 1s is odd then parity bit value is 1.
- In case of odd parity: If a number of 1s is odd then parity bit value is 0. If a number of 1s is even then parity bit value is 1.

On receiving a frame, the receiver counts the number of 1s in it. In case of even parity check, if the count of 1s is even, the frame is accepted, otherwise, it is rejected. A similar rule is adopted for odd parity check.

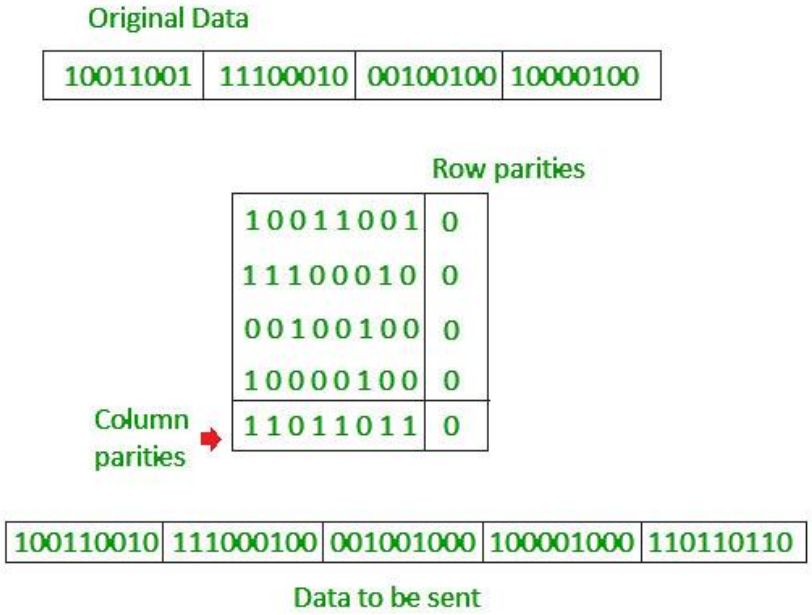
The parity check is suitable for single bit error detection only.



Two-dimensional Parity check:

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, and then both are sent along with the data.

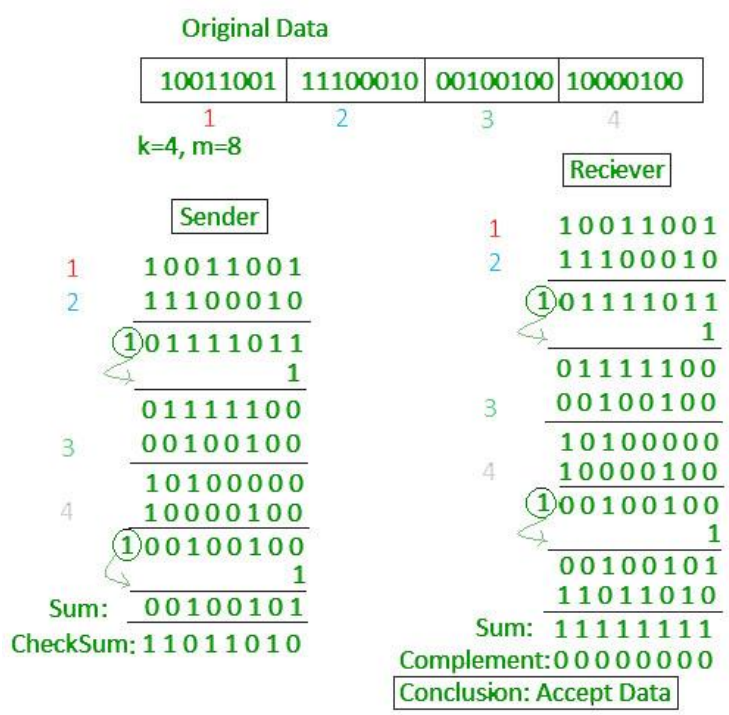
At the receiving end these are compared with the parity bits calculated on the received data.



Checksum

In this error detection scheme, the following procedure is applied

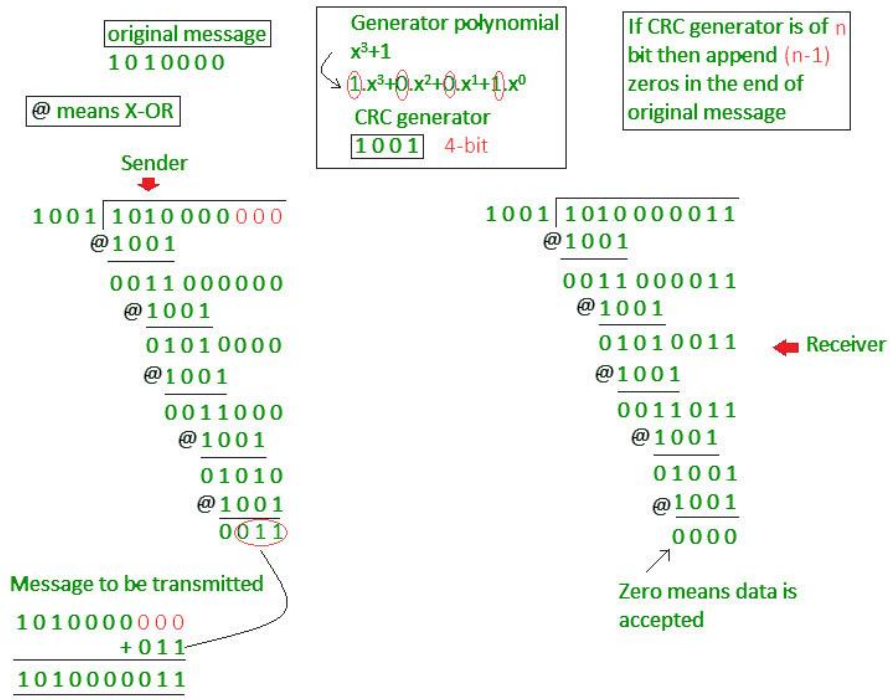
- Data is divided into fixed sized frames or segments.
- The sender adds the segments using 1's complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.
- The receiver adds the incoming segments along with the checksum using 1's complement arithmetic to get the sum and then complements it.
- If the result is zero, the received frames are accepted; otherwise, they are discarded.



Cyclic Redundancy Check (CRC)

Cyclic Redundancy Check (CRC) involves binary division of the data bits being sent by a predetermined divisor agreed upon by the communicating system. The divisor is generated using polynomials.

- Here, the sender performs binary division of the data segment by the divisor. It then appends the remainder called CRC bits to the end of the data segment. This makes the resulting data unit exactly divisible by the divisor.
- The receiver divides the incoming data unit by the divisor. If there is no remainder, the data unit is assumed to be correct and is accepted. Otherwise, it is understood that the data is corrupted and is therefore rejected.



Error Correction Techniques

Error correction techniques find out the exact number of bits that have been corrupted and as well as their locations. There are two principle ways

- **Backward Error Correction (Retransmission)** – If the receiver detects an error in the incoming frame, it requests the sender to retransmit the frame. It is a relatively simple technique. But it can be efficiently used only where retransmitting is not expensive as in fiber optics and the time for retransmission is low relative to the requirements of the application.

- **Forward Error Correction** – If the receiver detects some error in the incoming frame, it executes error-correcting code that generates the actual frame. This saves bandwidth required for retransmission. It is inevitable in real-time systems. However, if there are too many errors, the frames need to be retransmitted.

The main error correction code is: Hamming Codes

Hamming Code:

A single additional bit can detect the error, but cannot correct it.

For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.

Suppose r is the number of redundant bits and d is the total number of the data bits. The number of redundant bits r can be calculated by using the formula: $2^r>=d+r+1$

The value of r is calculated by using the above formula. For example, if the value of d is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

Algorithm of Hamming code:

- An information of 'd' bits are added to the redundant bits 'r' to form d+r.
- The location of each of the (d+r) digits is assigned a decimal value.
- The 'r' bits are placed in the positions $1,2,.....2^{k-1}$.
- At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Let's understand the concept of Hamming code through an example:

Suppose the original data is 1010 which is to be sent.

Total number of data bits'd' = 4

Number of redundant bits r: $2^r >= d+r+1$

$2^r >= 4+r+1$

Therefore, the value of r is 3 that satisfy the above relation.

Total number of bits = d+r = 4+3 = 7;

Determining the position of the redundant bits:

The number of redundant bits is 3. The three bits are represented by r1, r2, r4. The position of the redundant bits is calculated with corresponds to the raised power of 2. Therefore, their corresponding positions are **1, 2^1, 2^2**. Therefore,

- The position of r1 = 1
- The position of r2 = 2
- The position of r4 = 4

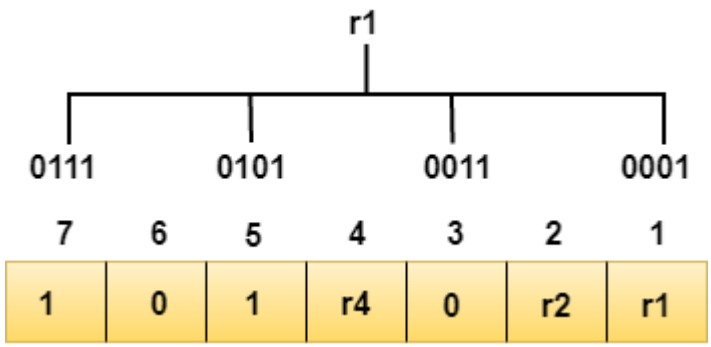
Representation of Data on the addition of parity bits:

7	6	5	4	3	2	1
1	0	1	r4	0	r2	r1

Determining the Parity bits:

Determining the r1 bit

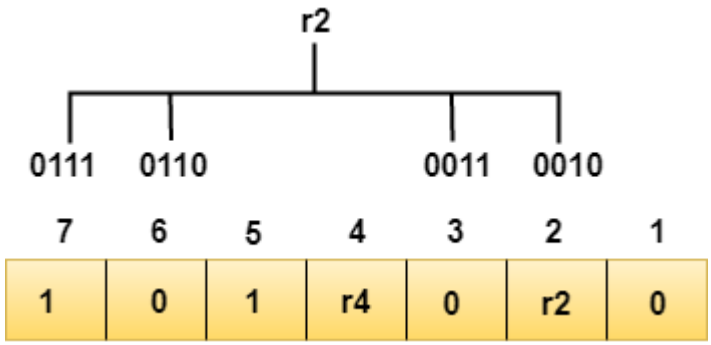
The r1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



We observe from the above figure that the bit positions that include 1 in the first position are 1, 3, 5, and 7. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r1 is **even, therefore, the value of the r1 bit is 0**.

Determining r2 bit

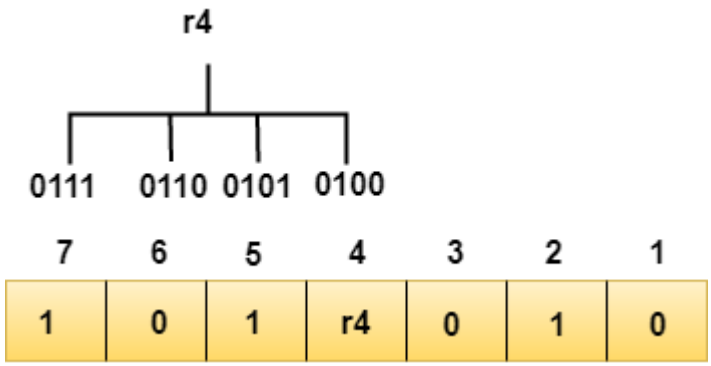
The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



We observe from the above figure that the bit positions that include 1 in the second position are **2, 3, 6, and 7**. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r2 is **odd; therefore, the value of the r2 bit is 1**.

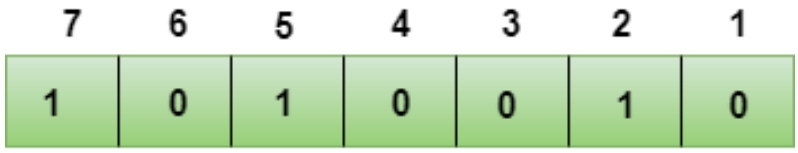
Determining r4 bit

The r4 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



We observe from the above figure that the bit positions that include 1 in the third position are **4, 5, 6, and 7**. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r4 is **even, therefore, the value of the r4 bit is 0**.

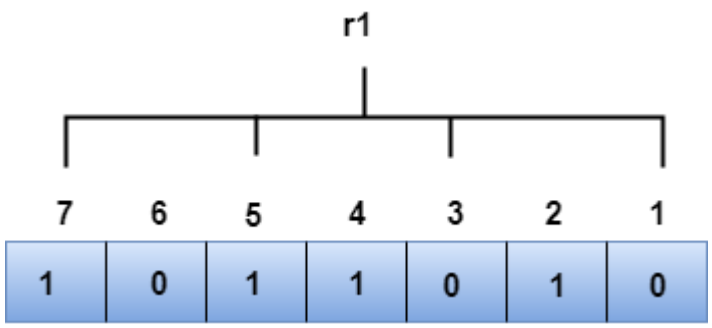
Data transferred is given below:



Suppose the 4th bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

R1 bit

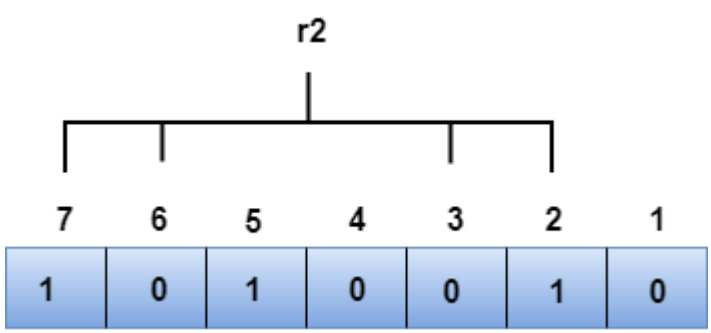
The bit positions of the r1 bit are 1,3,5,7



We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

R2 bit

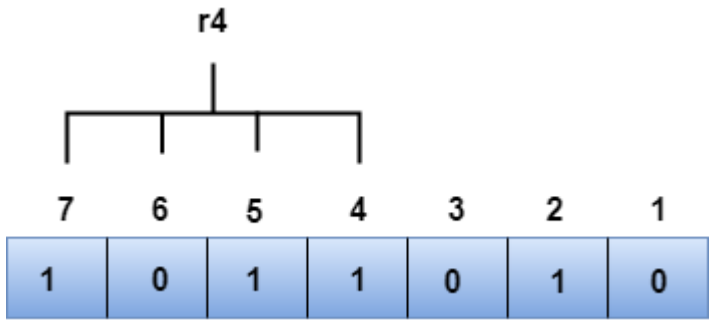
The bit positions of r2 bit are 2,3,6,7.



We observe from the above figure that the binary representation of r2 is 1001. Now, we perform the even-parity check, the total number of 1s appearing in the r2 bit is an even number. Therefore, the value of r2 is 0.

R4 bit

The bit positions of r4 bit are 4,5,6,7.



We observe from the above figure that the binary representation of r4 is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the r4 bit is an odd number. Therefore, the value of r4 is 1.

- **The binary representation of redundant bits, i.e., r4r2r1 is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4th bit position. The bit value must be changed from 1 to 0 to correct the error.**

Alpha Numeric Codes:

Alphanumeric codes are basically binary codes which are used to represent the alphanumeric data. As these codes represent data by characters, alphanumeric codes are also called “Character codes”.

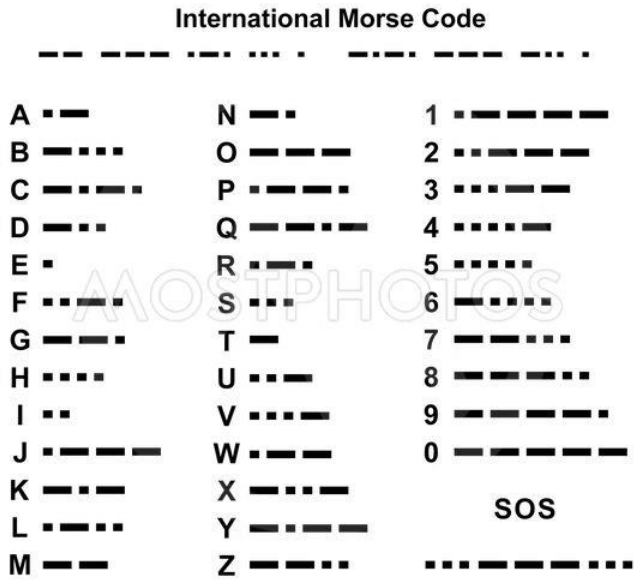
These codes can represent all types of data including alphabets, numbers, punctuation marks and mathematical symbols in the acceptable form by computers. These codes are implemented in I/O devices like key boards, monitors, printers etc.

In earlier days, punch cards are used to represent the alphanumeric codes. They are listed below as:

- MORSE code
- BAUDOT code
- HOLLERITH code
- ASCII code
- EBCDI code
- UNICODE

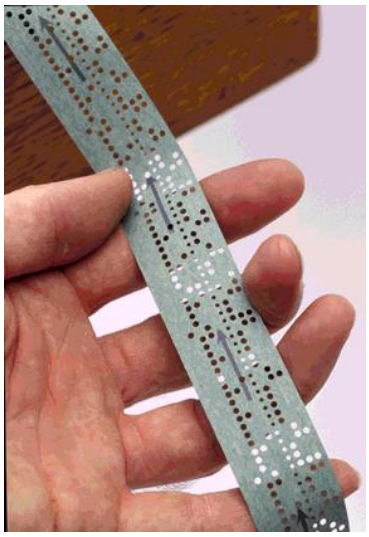
MORSE Code:

At the starting stage of computer and digital electronics era, Morse code is very popular and most used code. This was invented by Samuel F.B.Morse, in 1837. It was the first ever telegraphic code used in telecommunication. It is mainly used in Telegraph channels, Radio channels and in air traffic control units.



BOUDOT Code:

This code is invented by a French Engineer Emile Baudot, in 1870. It is a 5 unit code, means it uses 5 elements to represent an alphabet. It is also used in Telegraph networks to transfer Roman numeric.



HOLLERITH Code:

This code is developed by a company founded by Herman Hollerith in 1896. The 12 bit code used to punch cards according to the transmitting information is called “Hollerith code”.

TABLE 6.1 HOLLERITH CODE

Character	Punch at Rows	Character	Punch at Rows
1	1	O	11,6
2	2	P	11,7
3	3	Q	11,8
4	4	R	11,9
5	5	S	0,2
6	6	T	0,3
7	7	U	0,4
8	8	V	0,5
9	9	W	0,6
A	12,1	X	0,7
B	12,2	Y	0,8
C	12,3	Z	0,9
D	12,4	+	12
E	12,5	–	11
F	12,6	*	11,4,8
G	12,7	/	0,1
H	12,8	=	3,8
I	12,9	(	0,4,8
J	11,1	)	12,4,8
K	11,2		12,3,8
L	11,3	, comma	0,3,8
M	11,4	' quote	4,8
N	11,5	$	11,3,8

ASCII CODE:

ASCII means American Standard Code for Information Interchange. It is the world's most popular and widely used alphanumeric code. This code was developed and first published in 1967. ASCII code is a 7 bit code that means this code uses 27 = 128 characters.

This includes 26 lower case letters (a – z), 26 upper case letters (A – Z), 33 special characters and symbols (! @ # $ etc), 33 control characters (* – + / and %) and 10 digits (0 – 9).

Dec	Hex	Oct	Binary	Char		Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char
0	00	000	0000000	NUL	(null character)	32	20	040	0100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	01	001	0000001	SOH	(start of header)	33	21	041	0100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	02	002	0000010	STX	(start of text)	34	22	042	0100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	03	003	0000011	ETX	(end of text)	35	23	043	0100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	04	004	0000100	EOT	(end of transmission)	36	24	044	0100100	$	68	44	104	1000100	D	100	64	144	1100100	d
5	05	005	0000101	ENQ	(enquiry)	37	25	045	0100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	06	006	0000110	ACK	(acknowledge)	38	26	046	0100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	07	007	0000111	BEL	(bell (ring))	39	27	047	0100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	08	010	0001000	BS	(backspace)	40	28	050	0101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	09	011	0001001	HT	(horizontal tab)	41	29	051	0101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	012	0001010	LF	(line feed)	42	2A	052	0101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	013	0001011	VT	(vertical tab)	43	2B	053	0101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	014	0001100	FF	(form feed)	44	2C	054	0101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	015	0001101	CR	(carriage return)	45	2D	055	0101101	-	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	016	0001110	SO	(shift out)	46	2E	056	0101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	017	0001111	SI	(shift in)	47	2F	057	0101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	020	0010000	DLE	(data link escape)	48	30	060	0110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	021	0010001	DC1	(device control 1)	49	31	061	0110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	022	0010010	DC2	(device control 2)	50	32	062	0110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	023	0010011	DC3	(device control 3)	51	33	063	0110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	024	0010100	DC4	(device control 4)	52	34	064	0110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	025	0010101	NAK	(negative acknowledge)	53	35	065	0110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	026	0010110	SYN	(synchronize)	54	36	066	0110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	027	0010111	ETB	(end transmission block)	55	37	067	0110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	030	0011000	CAN	(cancel)	56	38	070	0111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	031	0011001	EM	(end of medium)	57	39	071	0111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	032	0011010	SUB	(substitute)	58	3A	072	0111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	033	0011011	ESC	(escape)	59	3B	073	0111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	034	0011100	FS	(file separator)	60	3C	074	0111100	<	92	5C	134	1011100	\	124	7C	174	1111100	|
29	1D	035	0011101	GS	(group separator)	61	3D	075	0111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	036	0011110	RS	(record separator)	62	3E	076	0111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	037	0011111	US	(unit separator)	63	3F	077	0111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL

Example:

If we want to print the name LONDAN, the ASCII code is?

The ASCII-7 equivalent of L = 100 1100

The ASCII-7 equivalent of O = 100 1111

The ASCII-7 equivalent of N = 100 1110

The ASCII-7 equivalent of D = 100 0100

The ASCII-7 equivalent of A = 100 0001

The ASCII-7 equivalent of N = 100 1110

The output of LONDAN in ASCII code is 1 0 0 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 1 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 1 1 1 0.

EBCDI CODE:

EBCDI stands for Extended Binary Coded Decimal Interchange code. This code is developed by IBM Inc Company.

It is an 8 bit code, so we can represent 28 = 256 characters by using EBCDI code.

This include all the letters and symbols like 26 lower case letters (a – z), 26 upper case letters (A – Z), 33 special characters and symbols (! @ # $ etc), 33 control characters (* – + / and % etc) and 10 digits (0 – 9).

In the EBCDI code, the 8 bit code the numbers are represented by 8421 BCD code preceded by 1111.

EBCDIC Format

Bits																		
			5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
			6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			7	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
			8	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	2	3	4																
0	0	0	0	NUL	SOH	STX	ETX	PF	HT	LC	DEL			SMM	VT	FF	CR	SO	SI
0	0	0	1	DLE	DC1	DC2	DC3	RES	NL	BS	IL	CAN	EM	CC		IFS	IGS	IRS	IUS
0	0	1	0	DS	SOS	FS		BYP	LF	EOB	PRE			SM			ENQ	ACK	BEL
0	0	1	1			SYN		PN	RS	US	EOT					DC4	NAK		SUB
0	1	0	0	SP										¢		<	(	+	|
0	1	0	1	&										!	$	*	)	;	¬
0	1	1	0	–	/										,	%	_	>	?
0	1	1	1											:	#	@	'	=	"
1	0	0	0		a	b	c	d	e	f	g	h	i						
1	0	0	1		j	k	l	m	n	o	p	q	r						
1	0	1	0			s	t	u	v	w	x	y	z						
1	0	1	1																
1	1	0	0		A	B	C	D	E	F	G	H	I						
1	1	0	1		J	K	L	M	N	O	P	Q	R						
1	1	1	0			S	T	U	V	W	X	Y	Z						
1	1	1	1	0	1	2	3	4	5	6	7	8	9						

PF	Punch off
HT	Horizontal tab
LC	Lower case
DEL	Delete
SP	Space
UC	Upper case
RES	Restore
NL	New line
BS	Backspace
IL	Idle
PN	Punch on
EOT	End of transmission
BYP	Bypass
LF	Line feed
EOB	End of block
PRE	Prefix (ESC)
RS	Reader stop
SM	Start message
DS	Digit select
SOS	Start of significance
IFS	Interchange file separator
IGS	Interchange group separator
IRS	Interchange record separator
IUS	Interchange unit separator
Others	Same as ASCII

UNICODE:

The draw backs in ASCII code and EBCDI code are that they are not compatible to all languages and they do not have sufficient set of characters to represent all types of data.

To overcome these drawback this UNICODE is developed.

UNICODE is the new concept of all digital coding techniques. In this we have a different character to represent every number.

It is the most advanced and sophisticated language with the ability to represent any type of data.

SO this is known as “Universal code”. It is a 16 bit code, with which we can represent 216 = 65536 different characters.

UNICODE is developed by the combined effort of UNICODE consortium and ISO (International organization for Standardization).

Graphic character symbol	Hexadecimal character value

0020	0 0030	@ 0040	P 0050	` 0060	p 0070	00A0	° 00B0	À 00C0	Ð 00D0	à 00E0	ð 00F0
! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071	¡ 00A1	± 00B1	Á 00C1	Ñ 00D1	á 00E1	ñ 00F1
" 0022	2 0032	B 0042	R 0052	b 0062	r 0072	¢ 00A2	² 00B2	Â 00C2	Ò 00D2	â 00E2	ò 00F2
# 0023	3 0033	C 0043	S 0053	c 0063	s 0073	£ 00A3	³ 00B3	Ã 00C3	Ó 00D3	ã 00E3	ó 00F3
$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074	¤ 00A4	´ 00B4	Ä 00C4	Ô 00D4	ä 00E4	ô 00F4
% 0025	5 0035	E 0045	U 0055	e 0065	u 0075	¥ 00A5	µ 00B5	Å 00C5	Õ 00D5	å 00E5	õ 00F5
& 0026	6 0036	F 0046	V 0056	f 0066	v 0076	¦ 00A6	¶ 00B6	Æ 00C6	Ö 00D6	æ 00E6	ö 00F6
' 0027	7 0037	G 0047	W 0057	g 0067	w 0077	§ 00A7	· 00B7	Ç 00C7	× 00D7	ç 00E7	÷ 00F7
(0028	8 0038	H 0048	X 0058	h 0068	x 0078	¨ 00A8	¸ 00B8	È 00C8	Ø 00D8	è 00E8	ø 00F8
) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079	© 00A9	¹ 00B9	É 00C9	Ù 00D9	é 00E9	ù 00F9
* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A	ª 00AA	º 00BA	Ê 00CA	Ú 00DA	ê 00EA	ú 00FA
+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B	« 00AB	» 00BB	Ë 00CB	Û 00DB	ë 00EB	û 00FB
, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C	¬ 00AC	¼ 00BC	Ì 00CC	Ü 00DC	ì 00EC	ü 00FC
- 002D	= 003D	M 004D	] 005D	m 006D	} 007D	- 00AD	½ 00BD	Í 00CD	Ý 00DD	í 00ED	ý 00FD
. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E	® 00AE	¾ 00BE	Î 00CE	Þ 00DE	î 00EE	þ 00FE
/ 002F	? 003F	O 004F	_ 005F	o 006F	007F	¯ 00AF	¿ 00BF	Ï 00CF	ß 00DF	ï 00EF	ÿ 00FF

Elementary Data Link Protocols:

When the data link layer accepts a packet, it encapsulates the packet in a frame by adding a data link header and trailer to it. Thus, a frame consists of an **embedded packet**, **control information** (in the header), and a **checksum** (in the trailer).

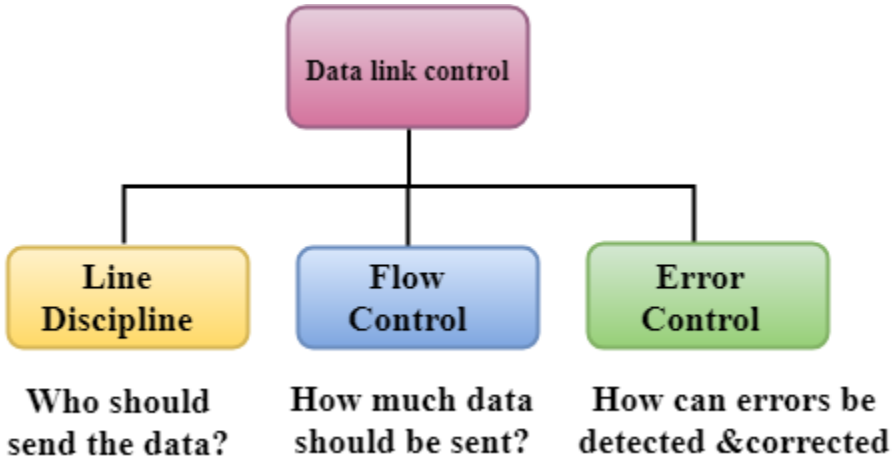
The frame is then transmitted to the data link layer on the other machine. We will assume that there exist suitable library procedures to physical layer to send and receive a frame.

Data Link Controls

Data Link Control is the service provided by the Data Link Layer to provide reliable data transfer over the physical medium. For example, In the half-duplex transmission mode, one device can only transmit the data at a time. If both the devices at the end of the links transmit the data simultaneously, they will collide and leads to the loss of the information. The Data link layer provides the coordination among the devices so that no collision occurs.

The Data link layer provides three functions:

- Line discipline
- Flow Control
- Error Control



Line Discipline

- Line Discipline is a functionality of the Data link layer that provides the coordination among the link systems. It determines which device can send, and when it can send the data.

  Line Discipline can be achieved in two ways:

- ENQ/ACK
- Poll/select

END/ACK

END/ACK stands for Enquiry/Acknowledgement is used when there is no wrong receiver available on the link and having a dedicated path between the two devices so that the device capable of receiving the transmission is the intended one.

END/ACK coordinates which device will start the transmission and whether the recipient is ready or not.

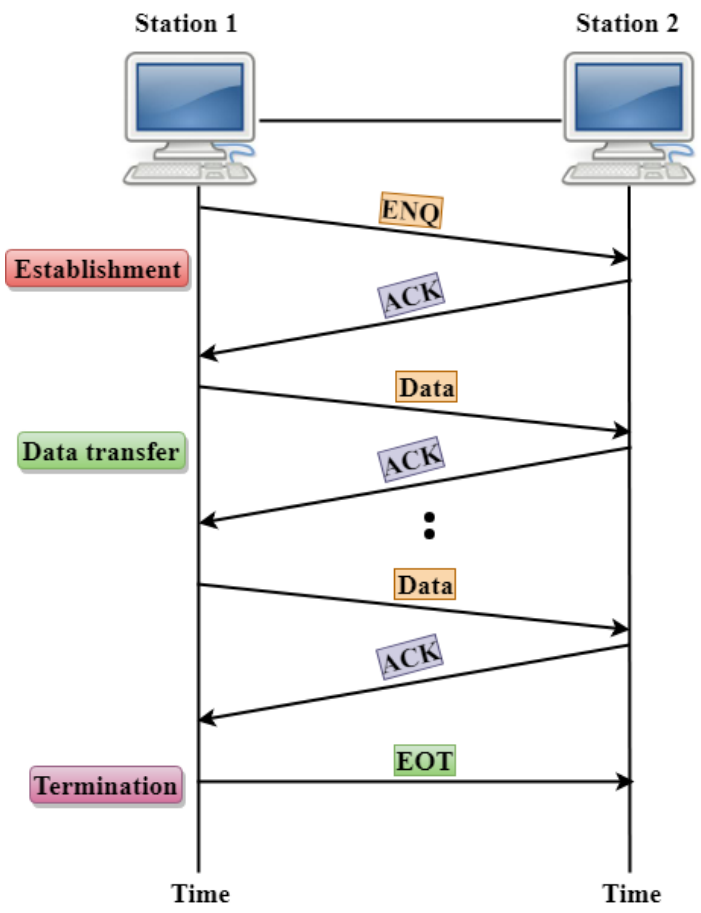
Working of END/ACK

The transmitter transmits the frame called an Enquiry (ENQ) asking whether the receiver is available to receive the data or not.

The receiver responses either with the positive acknowledgement(ACK) or with the negative acknowledgement(NACK) where positive acknowledgement means that the receiver is ready to receive the transmission and negative acknowledgement means that the receiver is unable to accept the transmission.

Following are the responses of the receiver:

- If the response to the ENQ is positive, the sender will transmit its data, and once all of its data has been transmitted, the device finishes its transmission with an EOT (END-of-Transmission) frame.
- If the response to the ENQ is negative, then the sender disconnects and restarts the transmission at another time.
- If the response is neither negative nor positive, the sender assumes that the ENQ frame was lost during the transmission and makes three attempts to establish a link before giving up.



Poll/Select

The Poll/Select method of line discipline works with those topologies where one device is designated as a primary station, and other devices are secondary stations.

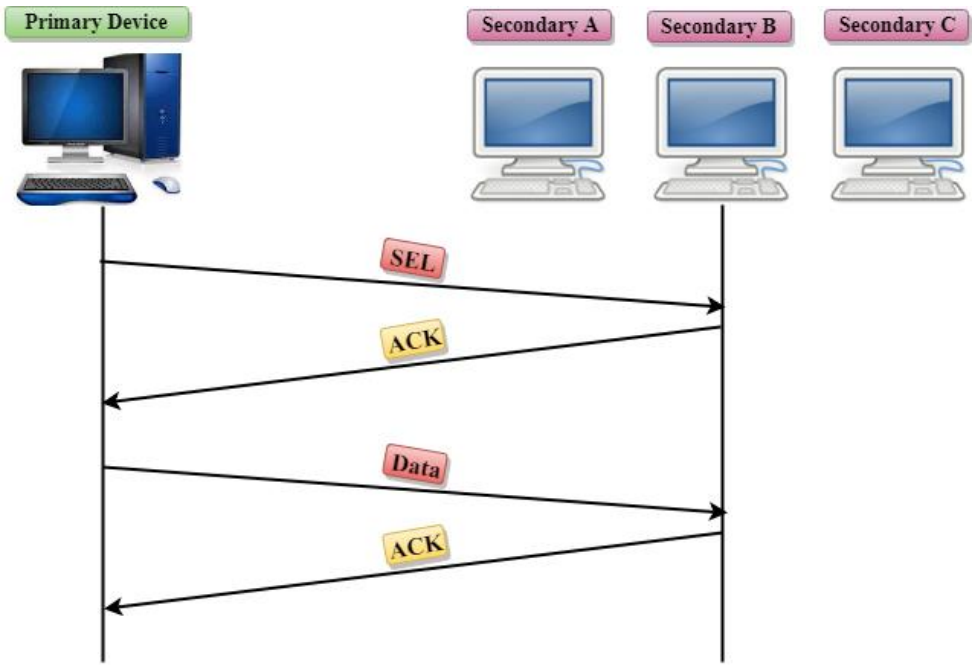
Working of Poll/Select

- In this, the primary device and multiple secondary devices consist of a single transmission line, and all the exchanges are made through the primary device even though the destination is a secondary device.
- The primary device has control over the communication link, and the secondary device follows the instructions of the primary device.

- The primary device determines which device is allowed to use the communication channel. Therefore, we can say that it is an initiator of the session.
- If the primary device wants to receive the data from the secondary device, it asks the secondary device that they anything to send, this process is known as polling.
- If the primary device wants to send some data to the secondary device, then it tells the target secondary to get ready to receive the data, this process is known as selecting.

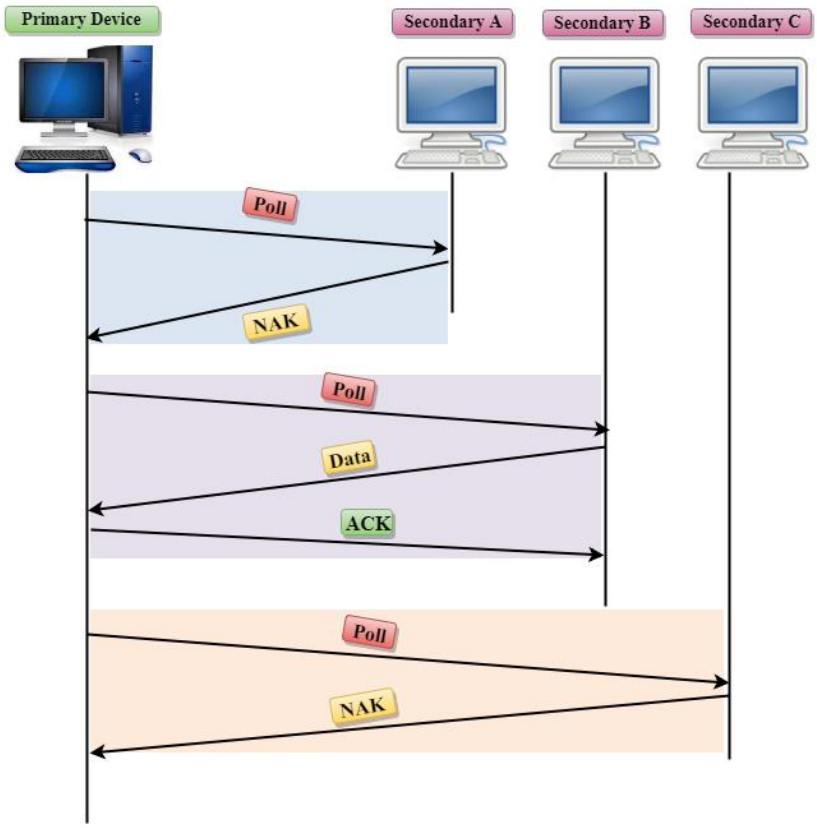
Select

- The select mode is used when the primary device has something to send.
- When the primary device wants to send some data, then it alerts the secondary device for the upcoming transmission by transmitting a Select (SEL) frame, one field of the frame includes the address of the intended secondary device.
- When the secondary device receives the SEL frame, it sends an acknowledgement that indicates the secondary ready status.
- If the secondary device is ready to accept the data, then the primary device sends two or more data frames to the intended secondary device. Once the data has been transmitted, the secondary sends an acknowledgement specifies that the data has been received.



Poll

- The Poll mode is used when the primary device wants to receive some data from the secondary device.
- When a primary device wants to receive the data, then it asks each device whether it has anything to send.
- Firstly, the primary asks (poll) the first secondary device, if it responds with the NACK (Negative Acknowledgement) means that it has nothing to send. Now, it approaches the second secondary device, it responds with the ACK means that it has the data to send. The secondary device can send more than one frame one after another or sometimes it may be required to send ACK before sending each one, depending on the type of the protocol being used.



Elementary Data Link Protocols: When a frame arrives at the receiver, the hardware computes the checksum. If the checksum is incorrect, the data link layer is so informed (event = cksum err).

If the inbound frame arrived undamaged, thedata link layer is also informed (event = frame arrival) so that it can acquire the frame for inspection using from physical layer.

As soon as the receiving data link layer has acquired an undamaged frame, it checks the control information in the header, and if everything is all right, passes the packet portion to the network layer. Under no circumstances is a frame header ever given to a network layer.

There is a good reason why the network layer must never be given any part of the frame header: to keep the network and data link protocols completely separate.

Five data structures are defined there:

1. Boolean
2. seq_nr
3. packet
4. frame_kind
5. frame

A **Boolean** is an enumerated type and can take on the values true and false.

A **seq_nr** is a small integer used to number the frames. These sequence numbers run from 0 up to and including MAX_SEQ, which is defined in each protocol needing it.

A **packet** is the unit of information exchanged between the network layer and the data link layer on the same machine, or between network layer peers.

In our model it always contains **MAX_PKT** bytes, but more realistically it would be of variable length.

A frame is composed of four fields: **kind, seq, ack,** and **info**, the first three of which contain control information and the last of which may contain actual data to be transferred.

These control fields are collectively called the **frame header**.

The **kind** field tells whether there are any data in the frame. The procedures **to network layer** and **from network layer** are used by the data link layer to pass packets to the network layer and accept packets from the network layer respectively.

```
#define MAX_PKT 1024                                  /* determines packet size in bytes */

typedef enum {false, true} boolean;                   /* boolean type */
typedef unsigned int seq_nr;                          /* sequence or ack numbers */
typedef struct {unsigned char data[MAX_PKT];} packet;/*    packet definition */
typedef enum {data, ack, nak} frame_kind;             /* frame_kind definition */

typedef struct {                                      /* frames are transported in this layer */
  frame_kind kind;                                    /* what kind of a frame is it? */
  seq_nr seq;                                         /* sequence number */
  seq_nr ack;                                         /* acknowledgement number */
  packet info;                                        /* the network layer packet */
} frame;

/* Wait for an event to happen; return its type in event. */
void wait_for_event(event_type *event);

/* Fetch a packet from the network layer for transmission on the channel. */
void from_network_layer(packet *p);

/* Deliver information from an inbound frame to the network layer. */
void to_network_layer(packet *p);

/* Go get an inbound frame from the physical layer and copy it to r. */
void from_physical_layer(frame *r);

/* Pass the frame to the physical layer for transmission. */
void to_physical_layer(frame *s);

/* Start the clock running and enable the timeout event. */
void start_timer(seq_nr k);

/* Stop the clock and disable the timeout event. */
void stop_timer(seq_nr k);

/* Start an auxiliary timer and enable the ack_timeout event. */
void start_ack_timer(void);

/* Stop the auxiliary timer and disable the ack_timeout event. */
void stop_ack_timer(void);

/* Allow the network layer to cause a network_layer_ready event. */
void enable_network_layer(void);

/* Forbid the network layer from causing a network_layer_ready event. */
void disable_network_layer(void);

/* Macro inc is expanded in-line: Increment k circularly. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

In most of the protocols, we assume that the channel is unreliable and loses entire frames upon occasion. To be able to recover from such calamities, the sending data link layer must start an internal timer or clock whenever it sends a frame.

If no reply has been received within a certain predetermined time interval, the clock timeout and the data link layer receive an interrupt signal.

Three different protocols have been proposed to demonstrate the same process on different channels.

1. An Unrestricted Simplex Protocol.
2. A Simplex Stop-and-Wait Protocol.
3. A Simplex Protocol for a Noisy Channel.

An Unrestricted Simplex Protocol

As an initial example we will consider a protocol that is as simple as it can be. Data are transmitted in one direction only. Both the transmitting and receiving network layers are always ready. Processing time can be ignored. Infinite buffer space is available. And best of all, the communication channel between the data link layers never damages or loses frames.

```
/* Protocol 1 (utopia) provides for data transmission in one direction only, from
   sender to receiver.  The communication channel is assumed to be error free
   and the receiver is assumed to be able to process all the input infinitely quickly.
   Consequently, the sender just sits in a loop pumping data out onto the line as
   fast as it can. */

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender1(void)
{
  frame s;                          /* buffer for an outbound frame */
  packet buffer;                    /* buffer for an outbound packet */

  while (true) {
       from_network_layer(&buffer); /* go get something to send */
       s.info = buffer;             /* copy it into s for transmission */
       to_physical_layer(&s);       /* send it on its way */
  }                                 /* Tomorrow, and tomorrow, and tomorrow,
                                       Creeps in this petty pace from day to day
                                       To the last syllable of recorded time.
                                          - Macbeth, V, v */
}


void receiver1(void)
{
  frame r;
  event_type event;                 /* filled in by wait, but not used here */

  while (true) {
       wait_for_event(&event);      /* only possibility is frame_arrival */
       from_physical_layer(&r);     /* go get the inbound frame */
       to_network_layer(&r.info);   /* pass the data to the network layer */
  }
}
```

The protocol consists of two distinct procedures, a sender and a receiver. The sender runs in the data link layer of the source machine, and the receiver runs in the data link layer of the destination machine. No sequence numbers or acknowledgements are used here, so MAX_SEQ is not needed. The only event type possible is frame_arrival (i.e., the arrival of an undamaged frame).

The sender is in an infinite while loop just pumping data out onto the line as fast as it can.

The body of the loop consists of three actions:

1. Go fetch a packet from the (always obliging) network layer
2. Construct an outbound frame using the variable s
3. Send the frame on its way.

Only the info field of the frame is used by this protocol, because the other fields have to do with error and flow control and there are no errors or flow control restrictions here.

The receiver is equally simple. Initially, it waits for something to happen, the only possibility being the arrival of an undamaged frame.

Eventually, the frame arrives and the procedure wait_for_event returns, with event set to frame_arrival (which is ignored anyway).

The call to from_physical_layer removes the newly arrived frame from the hardware buffer and puts it in the variable r, where the receiver code can get at it. Finally, the data portion is passed on to the network layer, and the data link layer settles back to wait for the next frame, effectively suspending itself until the frame arrives.

A Simplex Stop-and-Wait Protocol

Now we will drop the most unrealistic restriction used in protocol 1: the ability of the receiving network layer to process incoming data infinitely quickly (or equivalently, the presence in the receiving data link layer of an infinite amount of buffer space in which to store all incoming frames while they are waiting their respective turns). The communication channel is still assumed to be error free however, and the data traffic is still simplex.

The main problem we have to deal with here is how to prevent the sender from flooding the receiver with data faster than the latter is able to process them. In essence, if the receiver requires a time (t) to execute from physical layer plus to network layer, the sender must transmit at an average rate less than one frame per time (t).

A more general solution to this dilemma is to have the receiver provide feedback to the sender. After having passed a packet to its network layer, the receiver sends a little dummy frame back to the sender which, in effect, gives the sender permission to transmit the next frame. After having sent a frame, the sender is required by the protocol to bide its time until the little dummy (i.e., acknowledgement) frame arrives. Using feedback from the receiver to let the sender know when it may send more data is an example of the flow control mentioned earlier.

Protocols in which the sender sends one frame and then waits for an acknowledgement before proceeding are called stop-and-wait.

```
/* Protocol 2 (stop-and-wait) also provides for a one-directional flow of data from
   sender to receiver. The communication channel is once again assumed to be error
   free, as in protocol 1. However, this time, the receiver has only a finite buffer
   capacity and a finite processing speed, so the protocol must explicitly prevent
   the sender from flooding the receiver with data faster than it can be handled. */

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender2(void)
{
  frame s;                          /* buffer for an outbound frame */
  packet buffer;                    /* buffer for an outbound packet */
  event_type event;                 /* frame_arrival is the only possibility */

  while (true) {
      from_network_layer(&buffer);  /* go get something to send */
      s.info = buffer;              /* copy it into s for transmission */
      to_physical_layer(&s);        /* bye-bye little frame */
      wait_for_event(&event);       /* do not proceed until given the go ahead */
  }
}

void receiver2(void)
{
  frame r, s;                       /* buffers for frames */
  event_type event;                 /* frame_arrival is the only possibility */
  while (true) {
      wait_for_event(&event);       /* only possibility is frame_arrival */
      from_physical_layer(&r);      /* go get the inbound frame */
      to_network_layer(&r.info);    /* pass the data to the network layer */
      to_physical_layer(&s);        /* send a dummy frame to awaken sender */
  }
}
```

unlike in protocol 1, the sender must wait until an acknowledgement frame arrives before looping back and fetching the next packet from the network layer. The sending data link layer need not even inspect the incoming frame: there is only one possibility. The incoming frame is always an acknowledgement.

The only difference between receiver1 and receiver2 is that after delivering a packet to the network layer, receiver2 sends an acknowledgement frame back to the sender before entering the wait loop again. Because only the arrival of the frame back at the sender is important, not its contents, the receiver need not put any particular information in it.

A Simplex Protocol for a Noisy Channel

Now let us consider the normal situation of a communication channel that makes errors. Frames may be either damaged or lost completely. However, we assume that if a frame is damaged in transit, the receiver hardware will detect this when it computes the checksum.

If the frame is damaged in such a way that the checksum is nevertheless correct, an unlikely occurrence, this protocol (and all other protocols) can fail (i.e., deliver an incorrect packet to the network layer).

At first glance it might seem that a variation of protocol 2 would work: adding a timer. The sender could send a frame, but the receiver would only send an acknowledgement frame if the data were correctly received. If a damaged frame arrived at the receiver, it would be discarded.

After a while the sender would time out and send the frame again. This process would be repeated until the frame finally arrived intact.

The above scheme has a fatal flaw in it. Think about the problem and try to discover what might go wrong before reading further.

To see what might go wrong, remember that it is the task of the data link layer processes to provide error-free, transparent communication between network layer processes.

The network layer on machine A gives a series of packets to its data link layer, which must ensure that an identical series of packets are delivered to the network layer on machine B by its data link layer.

In particular, the network layer on B has no way of knowing that a packet has been lost or duplicated, so the data link layer must guarantee that no combination of transmission errors, however unlikely, can cause a duplicate packet to be delivered to a network layer.

Consider the following scenario:

1. The network layer on A gives packet 1 to its data link layer. The packet is correctly received at B and passed to the network layer on B. B sends an acknowledgement frame back to A.

2. The acknowledgement frame gets lost completely. It just never arrives at all. Life would be a great deal simpler if the channel mangled and lost only data frames and not control frames, but sad to say, the channel is not very discriminating.

3. The data link layer on A eventually times out. Not having received an acknowledgement, it (incorrectly) assumes that its data frame was lost or damaged and sends the frame containing packet 1 again.

4. The duplicate frame also arrives at the data link layer on B perfectly and is unwittingly passed to the network layer there. If A is sending a file to B, part of the file will be duplicated (i.e., the copy of the file made by B will be incorrect and the error will not have been detected). In other words, the protocol will fail.

Clearly, what is needed is some way for the receiver to be able to distinguish a frame that it is seeing for the first time from a retransmission. The obvious way to achieve this is to have the sender put a sequence number in the header of each frame it sends. Then the receiver can check the sequence number of each arriving frame to see if it is a new frame or a duplicate to be discarded.

Protocols in which the sender waits for a positive acknowledgement before advancing to the next data item are often called PAR (Positive Acknowledgement with Retransmission) or ARQ (Automatic Repeat Request). Like protocol 2, this one also transmits data only in one direction.

```
/* Protocol 3 (par) allows unidirectional data flow over an unreliable channel. */

#define MAX_SEQ 1                          /* must be 1 for protocol 3 */
typedef enum  {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"

void sender3(void)
{
  seq_nr next_frame_to_send;               /* seq number of next outgoing frame */
  frame s;                                 /* scratch variable */
  packet buffer;                           /* buffer for an outbound packet */
  event_type event;

  next_frame_to_send = 0;                  /* initialize outbound sequence numbers */
  from_network_layer(&buffer);             /* fetch first packet */
  while (true) {
      s.info = buffer;                     /* construct a frame for transmission */
      s.seq = next_frame_to_send;          /* insert sequence number in frame */
      to_physical_layer(&s);               /* send it on its way */
      start_timer(s.seq);                  /* if answer takes too long, time out */
      wait_for_event(&event);              /* frame_arrival, cksum_err, timeout */
      if (event == frame_arrival) {
            from_physical_layer(&s);       /* get the acknowledgement */
            if (s.ack == next_frame_to_send) {
                  stop_timer(s.ack);       /* turn the timer off */
                  from_network_layer(&buffer); /* get the next one to send */
                  inc(next_frame_to_send);  /* invert next_frame_to_send */
            }
      }
  }
}

void receiver3(void)
{
  seq_nr frame_expected;
  frame r, s;
  event_type event;

  frame_expected = 0;
  while (true) {
      wait_for_event(&event);              /* possibilities: frame_arrival, cksum_err */
      if (event == frame_arrival) {        /* a valid frame has arrived. */
            from_physical_layer(&r);       /* go get the newly arrived frame */
            if (r.seq == frame_expected) {  /* this is what we have been waiting for. */
                  to_network_layer(&r.info); /* pass the data to the network layer */
                  inc(frame_expected);     /* next time expect the other sequence nr */
            }
            s.ack = 1 – frame_expected;    /* tell which frame is being acked */
            to_physical_layer(&s);         /* send acknowledgement */
      }
  }
}
```

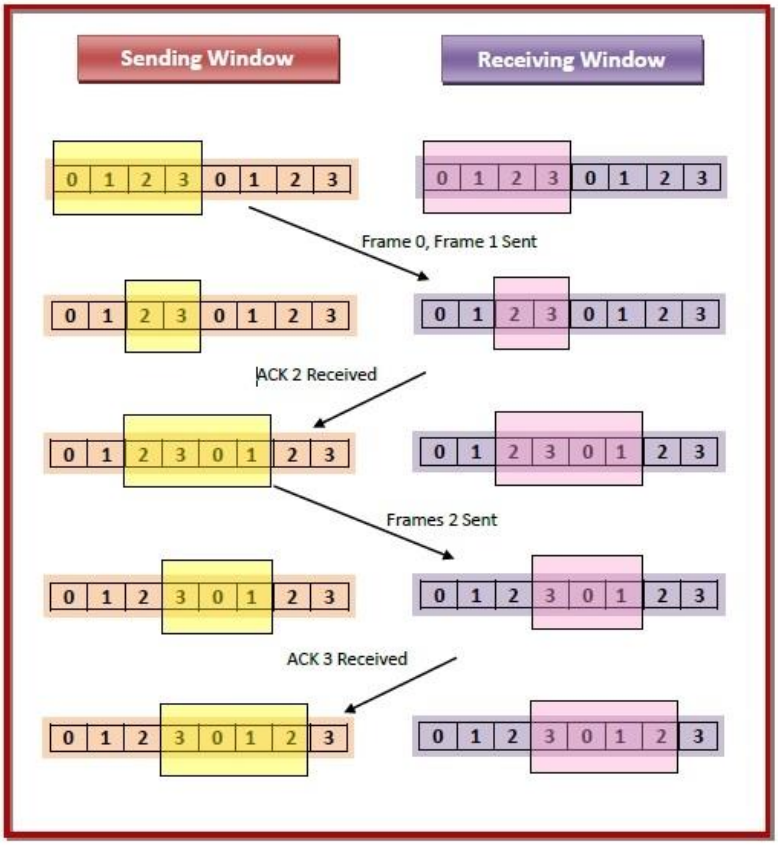
Sliding Window Protocol:

Sliding window protocols are data link layer protocols for reliable and sequential delivery of data frames. The sliding window is also used in Transmission Control Protocol.

In this protocol, multiple frames can be sent by a sender at a time before receiving an acknowledgment from the receiver. The term sliding window refers to the imaginary boxes to hold frames. Sliding window method is also known as windowing.

Example:

Suppose that we have sender window and receiver window each of size 4. So, the sequence numbering of both the windows will be 0,1,2,3,0,1,2 and so on. The following diagram shows the positions of the windows after sending the frames and receiving acknowledgments.



Types of Sliding Window Protocols

The Sliding Window ARQ (Automatic Repeat re-quest) protocols are of two categories −

1) **Go-Back-N ARQ:**

Go-Back-N ARQ protocol is also known as Go-Back-N Automatic Repeat Request. It is a data link layer protocol that uses a sliding window method. In this, if any frame is corrupted or lost, all subsequent frames have to be sent again.

The size of the sender window is N in this protocol. Example, Go-Back-8, the size of the sender window, will be 8. The receiver window size is always 1.

If the receiver receives a corrupted frame, it cancels it. The receiver does not accept a corrupted frame. When the timer expires, the sender sends the correct frame again.

2) **Selective Repeat ARQ:**

Selective Repeat ARQ is also known as the Selective Repeat Automatic Repeat Request. It is a data link layer protocol that uses a sliding window method.

The Go-back-N ARQ protocol works well if it has fewer errors. But if there is a lot of error in the frame, lots of bandwidth loss in sending the frames again. So, we use the Selective Repeat ARQ protocol.

In this protocol, the size of the sender window is always equal to the size of the receiver window. The size of the sliding window is always greater than 1.

If the receiver receives a corrupt frame, it does not directly discard it. It sends a negative acknowledgment to the sender.

The sender sends that frame again as soon as on the receiving negative acknowledgment. There is no waiting for any time-out to send that frame.

****

Unit- 4

The Medium Access Sub-layer

Content:

1. The channel allocation problems
2. Multiple access protocols
3. Ethernet
4. Data link layer switching

I. Introduction-

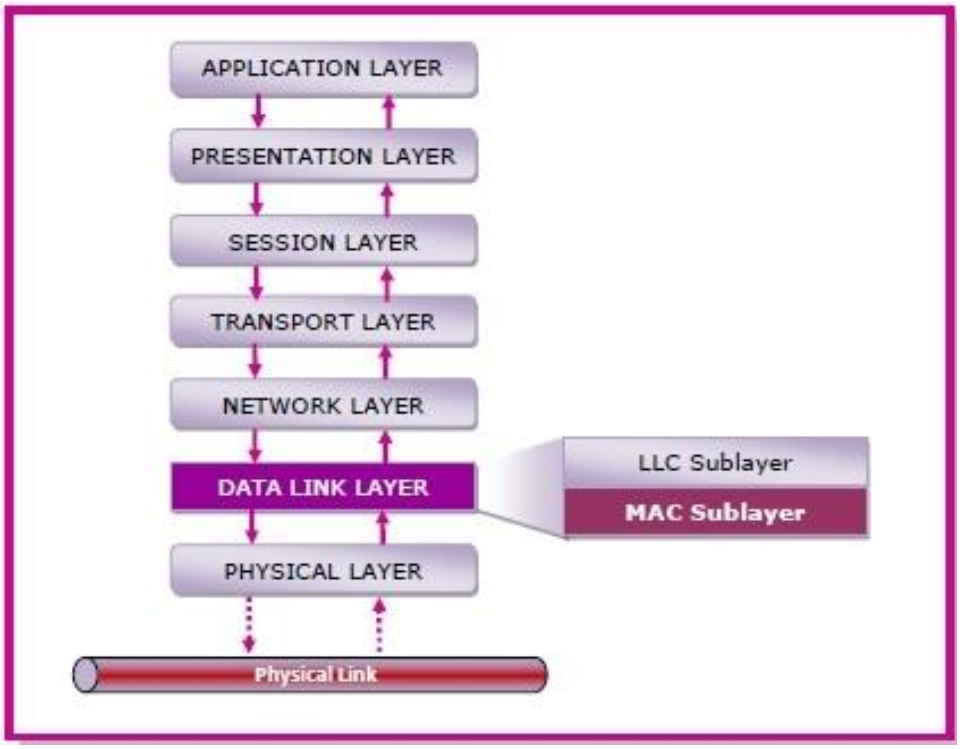
The medium access control (MAC) is a sublayer of the data link layer of the open system interconnections (OSI) reference model for data transmission. It is responsible for flow control and multiplexing for transmission medium. It controls the transmission of data packets via remotely shared channels. It sends data over the network interface card.

MAC Layer in the OSI Model:

The Open System Interconnections (OSI) model is a layered networking framework that conceptualizes how communications should be done between heterogeneous systems. The data link layer is the second lowest layer. It is divided into two sublayers –

1. The logical link control (LLC) sublayer
2. The medium access control (MAC) sublayer

The following diagram depicts the position of the MAC layer –



Functions of MAC Layer:

- It provides an abstraction of the physical layer to the LLC and upper layers of the OSI network.
- It is responsible for encapsulating frames so that they are suitable for transmission via the physical medium.
- It resolves the addressing of source station as well as the destination station, or groups of destination stations.
- It performs multiple access resolutions when more than one data frame is to be transmitted. It determines the channel access methods for transmission.
- It also performs collision resolution and initiating retransmission in case of collisions.
- It generates the frame check sequences and thus contributes to protection against transmission errors.

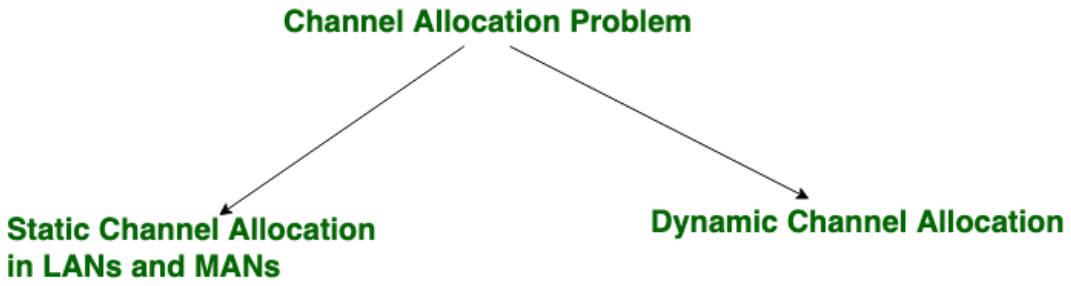
MAC Addresses:

MAC address or media access control address is a unique identifier allotted to a network interface controller (NIC) of a device. It is used as a network address for data transmission within a network segment like Ethernet, Wi-Fi, and Bluetooth.

MAC address is assigned to a network adapter at the time of manufacturing. It is hardwired or hard-coded in the network interface card (NIC). A MAC address comprises of six groups of two hexadecimal digits, separated by hyphens, colons, or no separators. An example of a MAC address is 00:0A:89:5B:F0:11.

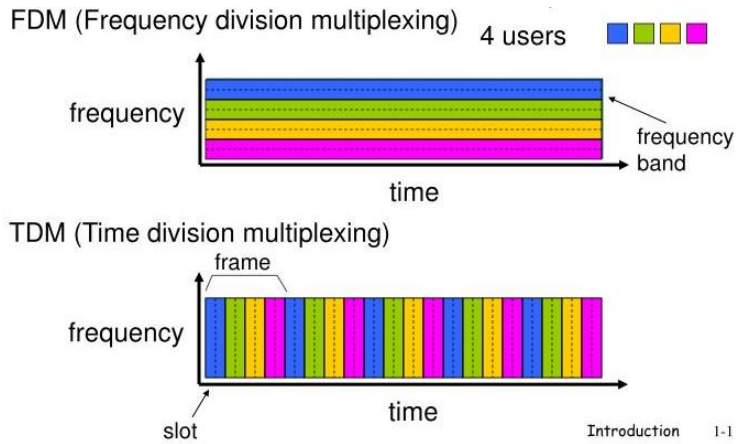
4.1 The Channel Allocation Problem:

Channel allocation is a process in which a single channel is divided and allotted to multiple users in order to carry user specific tasks. There is user's quantity may vary every time the process takes place. The central theme of this chapter is how to allocate a single broadcast channel among competing users.



4.1.1 Static Channel Allocation in LANs and MANs:

The traditional way of allocating a single channel, such as a telephone trunk, among multiple competing users is Frequency Division Multiplexing (FDM).



If there are *N* users, the bandwidth is divided into *N* equal-sized portions, each user being assigned one portion. Since each user has a private frequency band, there is no interference between users.

When there is only a small and constant number of users, each of which has a heavy (buffered) load of traffic (e.g., carriers' switching offices), FDM is a simple and efficient allocation mechanism.

However, when the number of senders is large and continuously varying or the traffic is bursty, FDM presents some problems.

If the spectrum is cut up into *N* regions and fewer than *N* users are currently interested in communicating, a large piece of valuable spectrum will be wasted.

If more than *N* users want to communicate, some of them will be denied permission for lack of bandwidth, even if some of the users who have been assigned a frequency band hardly ever transmit or receive anything.

The poor performance of static FDM can easily be seen from a simple queueing theory calculation.

Let us start with the mean time delay, *T*, for a channel of capacity *C* bps, with an arrival rate of λ frames/sec, each frame having a length drawn from an exponential probability density function with mean $1/\mu$ bits/frame.

With these parameters the arrival rate is λ frames/sec and the service rate is μ*C* frames/sec.

From queueing theory it can be shown that for Poisson arrival and service times,

$$T = \frac{1}{\mu C - \lambda}$$

Now let us divide the single channel into *N* independent subchannels, each with capacity *C/N* bps. The mean input rate on each of the subchannels will now be λ/*N.* Recomputing *T* we get,

$$T_{\text{FDM}} = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

4.1.2 Dynamic Channel Allocation in LANs and MANs:

Underlying all the work done in this area are five key assumptions, described below.

Station Model:

The model consists of *N* independent stations (e.g., computers, telephones, or personal communicators), each with a program or user that generates frames for transmission.

Stations are sometimes called terminals. The probability of a frame being generated in an interval of length Δ*t* is λΔ*t*, where λ is a constant (the arrival rate of new frames).

Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

Single Channel Assumption:

A single channel is available for all communication. All stations can transmit on it and all can receive from it. As far as the hardware is concerned, all stations are equivalent, although protocol software may assign priorities to them.

Collision Assumption:

If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a collision. All stations can detect collisions. A collided frame must be transmitted again later. There are no errors other than those generated by collisions.

4a. Continuous Time. Frame transmission can begin at any instant. There is no master clock dividing time into discrete intervals.

4b. Slotted Time. Time is divided into discrete intervals (slots). Frame transmissions always begin at the start of a slot. A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.

5a. Carrier Sense. Stations can tell if the channel is in use before trying to use it. If the channel is sensed as busy, no station will attempt to use it until it goes idle.

5b. No Carrier Sense. Stations cannot sense the channel before trying to use it. They just go ahead and transmit. Only later can they determine whether the transmission was successful.

Random Access Protocol: In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state (idle or busy). It has two features:

1. There is no fixed time for sending data
2. There is no fixed sequence of stations sending data

The Random-access protocols are further subdivided as:

(a) ALOHA – It was designed for wireless LAN but is also applicable for shared medium. In this, multiple stations can transmit data at the same time and can hence lead to collision and data being garbled.

- **Pure Aloha:**
  When a station sends data it waits for an acknowledgement. If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases.
- Vulnerable Time = 2* Frame transmission time
- Throughput = G exp{-2*G}

  Maximum throughput = 0.184 for G=0.5

- **Slotted Aloha:**
  It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots. If a station misses out the allowed time, it must wait for the next slot. This reduces the probability of collision.
- Vulnerable Time = Frame transmission time
- Throughput = G exp{-*G}

  Maximum throughput = 0.368 for G=1

  For more information on ALOHA refer – LAN Technologies

(b) CSMA – Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data. If it is idle then it sends data, otherwise it waits till the

channel becomes idle. However there is still chance of collision in CSMA due to propagation delay. For example, if station A wants to send data, it will first sense the medium.If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

CSMA access modes-

- **1-persistent:** The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle.
- **Non-Persistent:** The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.
- **P-persistent:** The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.
- **O-persistent:** Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

(c) CSMA/CD – Carrier sense multiple access with collision detection. Stations can terminate transmission of data if collision is detected. For more details refer – Efficiency of CSMA/CD

(d) CSMA/CA – Carrier sense multiple access with collision avoidance. The process of collisions detection involves sender receiving acknowledgement signals. If there is just one signal (its own) then the data is successfully sent but if there are two signals(its own and the one with which it has collided)

then it means a collision has occurred. To distinguish between these two cases, collision must have a lot of impact on received signal. However, it is not so in wired networks, so CSMA/CA is used in this case.

CSMA/CA avoids collision by:

1. **Interframe space –** Station waits for medium to become idle and if found idle it does not immediately send data (to avoid collision due to propagation delay) rather it waits for a period of time called Interframe space or IFS. After this time, it again checks the medium for being idle. The IFS duration depends on the priority of station.
2. **Contention Window –** It is the amount of time divided into slots. If the sender is ready to send data, it chooses a random number of slots as wait time which doubles every time medium is not found idle. If the medium is found busy it does not restart the entire process, rather it restarts the timer when the channel is found idle again.
3. **Acknowledgement –** The sender re-transmits the data if acknowledgement is not received before time-out.

4.1.3 Collision-Free Protocols

Although collisions do not occur with CSMA/CD once a station has unambiguously captured the channel, they can still occur during the contention period. These collisions adversely affect the system performance, especially when the cable is long (i.e., large τ) and the frames are short. And CSMA/CD is not universally applicable. In this section, we will examine some protocols that resolve the contention for the channel without any collisions at all, not even during the contention period.

4.1.3.1 A Bit-Map Protocol:

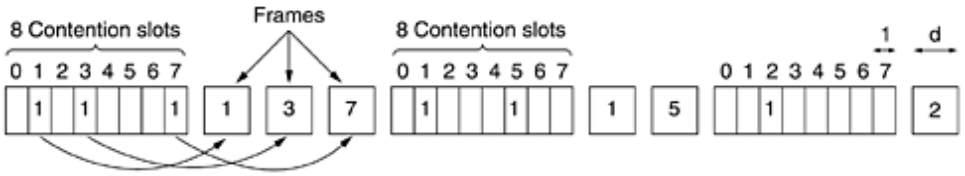
In our first collision-free protocol, the basic bit-map method, each contention period consists of exactly *N* slots. If station 0 has a frame to send,

it transmits a 1 bit during the zeroth slot. No other station is allowed to transmit during this slot. Regardless of what station 0 does,

station 1 gets the opportunity to transmit a 1 during slot 1, but only if it has a frame queued. In general, station *j* may announce that it has a frame to send by inserting a 1 bit into slot *j.* After all *N* slots have passed by, each station has complete knowledge of which stations wish to transmit. At that point, they begin transmitting in numerical order.

Since everyone agrees on who goes next, there will never be any collisions. After the last ready station has transmitted its frame, an event all stations can easily monitor, another *N* bit contention period is begun.

If a station becomes ready just after its bit slot has passed by, it is out of luck and must remain silent until every station has had a chance and the bit map has come around again. Protocols like this in which the desire to transmit is broadcast before the actual transmission are called **reservation protocols**.



4.1.3.2 Binary Countdown:

Binary Countdown Protocol is a collision-free protocol that operates in the Medium Access Control (MAC) layer of the OSI model. In computer networks, when more than one station tries to transmit simultaneously via a shared channel, the transmitted data is garbled, an event called collision. Collision-free protocols resolves channel access while the stations are contending for the shared channel, thus eliminating any possibilities of collisions.

Working Principle of Binary Countdown

In a binary countdown protocol, each station is assigned a binary address. The binary addresses are bit strings of equal lengths. When a station wants to transmit, it broadcasts its address to all the stations in the channel, one bit at a time starting with the highest order bit.

In order to decide which station gets the channel access, the addresses of the stations which are broadcasted are ORed. The higher numbered station gets the channel access.

Example

Suppose that six stations contend for channel access which have the addresses: 1011, 0010, 0101, 1100, 1001 and 1101.

The iterative steps are –

- All stations broadcast their most significant bit, i.e. 1, 0, 0, 1, 1, 1. Stations 0010 and 0101 sees 1 bit in other stations, and so they give up competing for the channel.
- The stations 1011, 1100, 1001 and 1101 continue. They broadcast their next bit, i.e. 0, 1, 0, 1. Stations 1011 and 1001 sees 1 bit in other stations, and so they give up competing for the channel.
- The stations 1100 and 1101 continue. They broadcast their next bit, i.e. 0, 0. Since both of them have same bit value, both of them broadcast their next bit.
- The stations 1100 and 1101 broadcast their least significant bit, i.e. 0 and 1. Since station 1101 has 1 while the other 0, station 1101 gets the access to the channel.
- After station 1101 has completed frame transmission, or there is a time-out, the next contention cycle starts.

Station Address	Bit Time 0	1	2	3	Station Status
1011	1	0	-	-	Gives up after bit time 1
0010	0	-	-	-	Gives up after bit time 0
0101	0	-	-	-	Gives up after bit time 0
1100	1	1	0	0	Gives up after bit time 3
1001	1	0	-	-	Gives up after bit time 0
1101	1	1	0	1	Gets channel access

4.1.3.3 Limited Contention Protocols:

- Collision based protocols (pure and slotted ALOHA, CSMA/CD) are good when the network load is low.
- Collision free protocols (bitmap, binary Countdown) are good when load is high.
- How about combining their advantages?

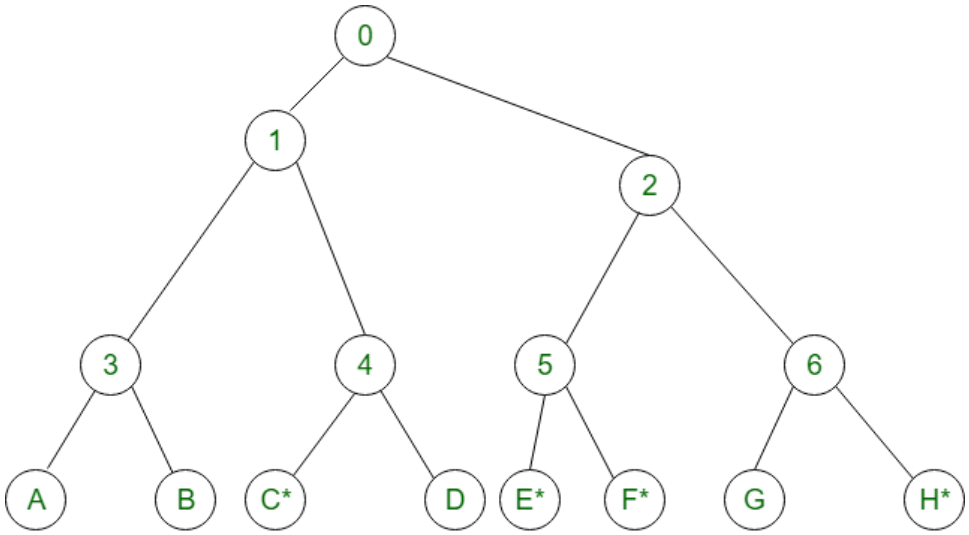
1. Behave like the ALOHA scheme under light load
2. Behave like the bitmap scheme under heavy load.

4.1.3.3.1 Adaptive Tree Walk Protocol:

- partition the group of station and limit the contention for each slot.
- Under light load, everyone can try for each slot like aloha
- Under heavy load, only a group can try for each slot
- **How do we do it:**

1. Treat every station as the leaf of a binary tree
2. First slot (after successful transmission), all stations can try to get the slot (under the root node).
3. If no conflict, fine
4. n case of conflict, only nodes under a subtree get to try for the next one. (depth first search)

For Example:

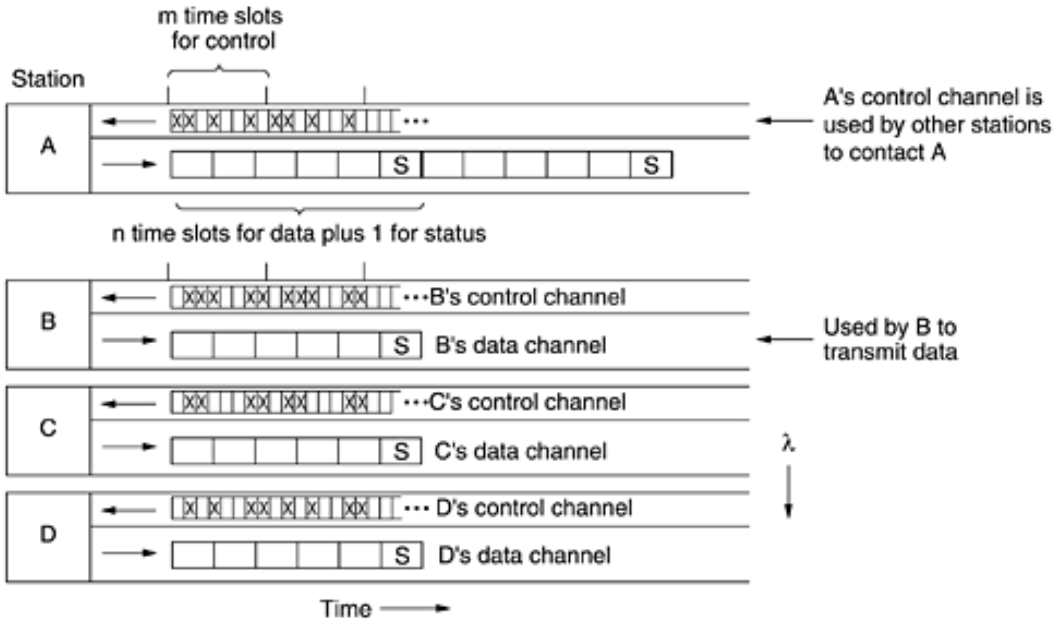


- **Slot-0:** C*, E*, F*, H* (all nodes under node 0 can try which are going to send), conflict
- **Slot-1:** C* (all nodes under node 1can try}, C sends
- **Slot-2:** E*, F*, H*(all nodes under node 2 can try}, conflict
- **Slot-3:** E*, F* (all nodes under node 5 can try to send), conflict
- **Slot-4:** E* (all nodes under E can try), E sends
- **Slot-5:** F* (all nodes under F can try), F sends
- **Slot-6:** H* (all nodes under node 6 can try to send), H sends.

4.1.3.3 Wavelength Division Multiple Access Protocols:

A different approach to channel allocation is to divide the channel into subchannels using FDM, TDM, or both, and dynamically allocate them as needed. Schemes like this are commonly used on fiber optic LANs to permit different conversations to use different wavelengths (i.e., frequencies) at the same time.

In this protocol, **WDMA** (**Wavelength Division Multiple Access**), each station is assigned two channels. A narrow channel is provided as a control channel to signal the station, and a wide channel is provided so the station can output data frames.



The protocol supports three traffic classes: (1) constant data rate connection-oriented traffic, such as uncompressed video, (2) variable data rate connection-oriented traffic, such as file transfer, and (3) datagram traffic, such as UDP packets. For the two connection-oriented protocols, the idea is that for *A* to communicate with *B*, it must first insert a CONNECTION REQUEST frame in a free slot on *B*'s control channel. If *B* accepts, communication can take place on *A*'s data channel.

Each station has two transmitters and two receivers, as follows:

1. A fixed-wavelength receiver for listening to its own control channel.

2. A tunable transmitter for sending on other stations' control channels.

3. A fixed-wavelength transmitter for outputting data frames.

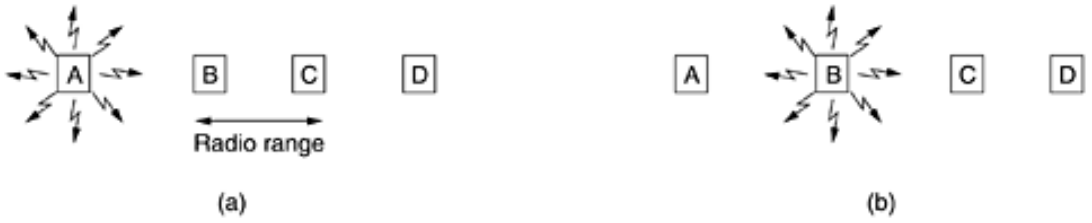
4. A tunable receiver for selecting a data transmitter to listen to.

4.1.3.4 Wireless LAN Protocols:

In our discussions below, we will make the simplifying assumption that all radio transmitters have some fixed range. When a receiver is within range of two active transmitters, the resulting signal will generally be garbled and useless, in other words, we will not consider CDMA-type systems further in this discussion. It is important to realize that in some wireless LANs, not all stations are within range of one another, which leads to a variety of complications. Furthermore, for indoor wireless LANs, the presence of walls between stations can have a major impact on the effective range of each station.

A naive approach to using a wireless LAN might be to try CSMA: just listen for other transmissions and only transmit if no one else is doing so. The trouble is, this protocol is not really appropriate because what matters is interference at the receiver, not at the sender. The radio range is such that A

and B are within each other's range and can potentially interfere with one another. C can also potentially interfere with both B and D, but not with A.



First consider what happens when A is transmitting to B, as depicted in Fig. 4-11(a). If C senses the medium, it will not hear A because A is out of range, and thus falsely conclude that it can transmit to B. If C does start transmitting, it will interfere at B, wiping out the frame from A. The problem of a station not being able to detect a potential competitor for the medium because the competitor is too far away is called the hidden station problem.

Now let us consider the reverse situation: B transmitting to A, as shown in Fig. 4-11(b). If C senses the medium, it will hear an ongoing transmission and falsely conclude that it may not send to D, when in fact such a transmission would cause bad reception only in the zone between B and C, where neither of the intended receivers is located. This is called the exposed station problem.

The problem is that before starting a transmission, a station really wants to know whether there is activity around the receiver. CSMA merely tells it whether there is activity around the station sensing the carrier. In a system based on short-range radio waves, multiple transmissions can occur simultaneously if they all have different destinations and these destinations are out of range of one another.

MACA and MACAW:

An early protocol designed for wireless LANs is **MACA (Multiple Access with Collision Avoidance)**. The basic idea behind it is for the sender to stimulate the receiver into outputting a short frame, so stations nearby can detect this

transmission and avoid transmitting for the duration of the upcoming (large) data frame.

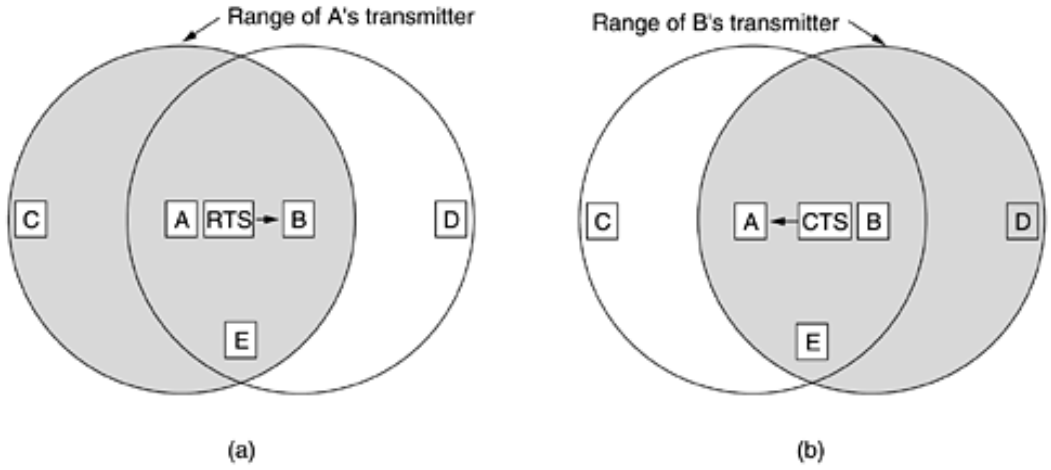


***Figure. The MACA protocol. (a)* A *sending an RTS to* B. *(b)* B *responding with a CTS to* A.**

Let us now consider how A sends a frame to B. A start by sending an RTS (Request To Send) frame to B, as shown in Fig.(a). This short frame (30 bytes) contains the length of the data frame that will eventually follow. Then B replies with a CTS (Clear to Send) frame, as shown in Fig.(b). The CTS frame contains the data length (copied from the RTS frame). Upon receipt of the CTS frame, A begins transmission.

In Fig, C is within range of A but not within range of B. Therefore, it hears the RTS from A but not the CTS from B. As long as it does not interfere with the CTS, it is free to transmit while the data frame is being sent. In contrast, D is within range of B but not A. It does not hear the RTS but does hear the CTS. Hearing the CTS tips, it off that it is close to a station that is about to receive a frame, so it defers sending anything until that frame is expected to be finished. Station E hears both control messages and, like D, must be silent until the data frame is complete.

Despite these precautions, collisions can still occur. For example, B and C could both send RTS frames to A at the same time. These will collide and be lost. In the event of a collision, an unsuccessful transmitter (i.e., one that does not hear a CTS within the expected time interval) waits a random

amount of time and tries again later. The algorithm used is binary exponential backoff, which we will study when we come to Ethernet.

Based on simulation studies of MACA, fine-tuned MACA to improve its performance and renamed their new protocol **MACAW (MACA for Wireless).** To start with, they noticed that without data link layer acknowledgements, lost frames were not retransmitted until the transport layer noticed their absence, much later.

They solved this problem by introducing an ACK frame after each successful data frame. They also observed that CSMA has some use, namely, to keep a station from transmitting an RTS at the same time another nearby station is also doing so to the same destination, so carrier sensing was added.

In addition, they decided to run the backoff algorithm separately for each data stream (source-destination pair), rather than for each station. This change improves the fairness of the protocol.

Finally, they added a mechanism for stations to exchange information about congestion and a way to make the backoff algorithm react less violently to temporary problems, to improve system performance.

4.2 Ethernet:

IEEE has standardized a number of local area networks and metropolitan area networks under the name of IEEE 802. The most important are 802.3 (Ethernet) and 802.11 (wireless LAN). We will focus on the technical details of Ethernet, the protocols, and recent developments in high-speed (gigabit) Ethernet.

1. Ethernet Cabling:

The most common kinds of Ethernet cabling are:

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

2. Manchester Encoding:

None of the versions of Ethernet uses straight binary encoding with 0 volts for a 0 bit and 5 volts for a 1 bit because it leads to ambiguities. This problem can be solved by using +1 volts for a 1 and -1 volts for a 0, but there is still the problem of a receiver sampling the signal at a slightly different frequency than the sender used to generate it. Different clock speeds can cause the receiver and sender to get out of synchronization about where the bit boundaries are, especially after a long run of consecutive 0s or a long run of consecutive 1s.

What is needed is a way for receivers to unambiguously determine the start, end, or middle of each bit without reference to an external clock. Two such approaches are called **Manchester encoding** and **differential Manchester encoding**.

Manchester encoding:

With Manchester encoding, each bit period is divided into two equal intervals. A binary 1 bit is sent by having the voltage set high during the first interval and low in the second one. A binary 0 is just the reverse: first low and then high.

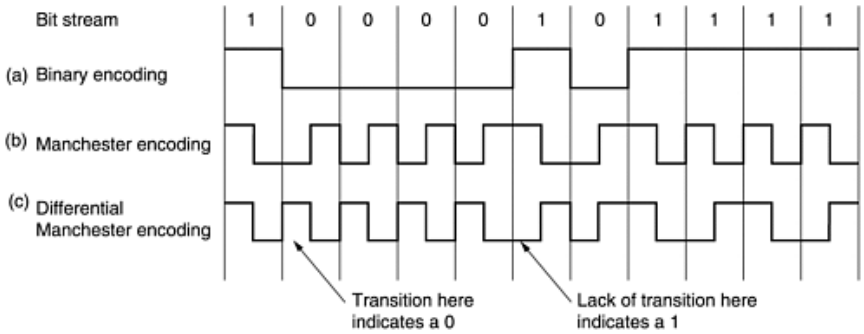
This scheme ensures that every bit period has a transition in the middle, making it easy for the receiver to synchronize with the sender.

A disadvantage of Manchester encoding is that it requires twice as much bandwidth as straight binary encoding because the pulses are half the width.

Differential Manchester encoding:

Differential Manchester encoding, is a variation of basic Manchester encoding. In it, a 1 bit is indicated by the absence of a transition at the start of the interval. A 0 bit is indicated by the presence of a transition at the start of the interval.

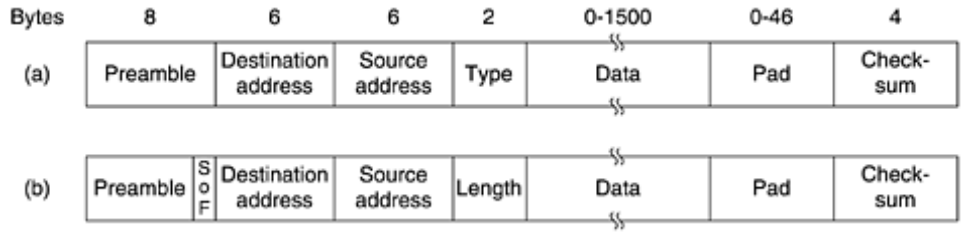
In both cases, there is a transition in the middle as well. The differential scheme requires more complex equipment but offers better noise immunity. All Ethernet systems use Manchester encoding due to its simplicity.



3. The Ethernet MAC Sublayer Protocol:

The original DIX (DEC, Intel, Xerox) frame structure is shown below in Fig.(a). Each frame starts with a *Preamble* of 8 bytes, each containing the bit pattern 10101010.

The Manchester encoding of this pattern produces a 10-MHz square wave for 6.4 µsec to allow the receiver's clock to synchronize with the senders. They are required to stay synchronized for the rest of the frame, using the Manchester encoding to keep track of the bit boundaries.



The frame contains two addresses, one for the destination and one for the source. The standard allows 2-byte and 6-byte addresses, but the

parameters defined for the 10-Mbps baseband standard use only the 6-byte addresses. The high-order bit of the destination address is a 0 for ordinary addresses and 1 for group addresses. Group addresses allow multiple stations to listen to a single address. When a frame is sent to a group address, all the stations in the group receive it. Sending to a group of stations is called multicast. The address consisting of all 1 bits is reserved for broadcast. A frame containing all 1s in the destination field is accepted by all stations on the network. The difference between multicast and broadcast is important enough to warrant repeating. A multicast frame is sent to a selected group of stations on the Ethernet; a broadcast frame is sent to all stations on the Ethernet. Multicast is more selective, but involves group management. Broadcasting is coarser but does not require any group management.

Next comes the *Type* field, which tells the receiver what to do with the frame. Multiple network-layer protocols may be in use at the same time on the same machine, so when an Ethernet frame arrives, the kernel has to know which one to hand the frame to. The *Type* field specifies which process to give the frame to.

Next come the data, up to 1500 bytes. This limit was chosen somewhat arbitrarily at the time the DIX standard was cast in stone, mostly based on the fact that a transceiver needs enough RAM to hold an entire frame and RAM was expensive in 1978. A larger upper limit would have meant more RAM, hence a more expensive transceiver.

Another interesting feature of the addressing is the use of bit 46 (adjacent to the high-order bit) to distinguish local from global addresses. Local addresses are assigned by each network administrator and have no significance outside the local network. Global addresses, in contrast, are assigned centrally by IEEE to ensure that no two stations anywhere in the world have the same global address. The final Ethernet field is the *Checksum*. It is effectively a 32-bit hash code of the data. If some data bits are erroneously received (due to

noise on the cable), the checksum will almost certainly be wrong and the error will be detected.

When IEEE standardized Ethernet, the committee made two changes to the DIX format, as shown in Fig.(b). The first one was to reduce the preamble to 7 bytes and use the last byte for a *Start of Frame* delimiter, for compatibility with 802.4 and 802.5. The second one was to change the *Type* field into a *Length* field.

Unfortunately, by the time 802.3 was published, so much hardware and software for DIX Ethernet was already in use that few manufacturers and users were enthusiastic about converting the *Type* field into a *Length* field. In 1997 IEEE threw in the towel and said that both ways were fine with it.

Fortunately, all the *Type* fields in use before 1997 were greater than 1500. Consequently, any number there less than or equal to 1500 can be interpreted as *Length*, and any number greater than 1500 can be interpreted as *Type*.

4. The Binary Exponential Backoff Algorithm:

Let us now see how randomization is done when a collision occurs. After a collision, time is divided into discrete slots whose length is equal to the worst-case round-trip propagation time on the ether (2τ). To accommodate the longest path allowed by Ethernet, the slot time has been set to 512 bit times, or 51.2 µsec as mentioned above.

After the first collision, each station waits either 0 or 1 slot times before trying again. If two stations collide and each one picks the same random number, they will collide again. After the second collision, each one picks either 0, 1, 2, or 3 at random and waits that number of slot times. If a third collision occurs (the probability of this happening is 0.25), then the next time the number of slots to wait is chosen at random from the interval 0 to $2^3 - 1$.

In general, after *i* collisions, a random number between 0 and $2^i - 1$ is chosen, and that number of slots is skipped. However, after ten collisions have been

reached, the randomization interval is frozen at a maximum of 1023 slots. After 16 collisions, the controller throws in the towel and reports failure back to the computer. Further recovery is up to higher layers.

This algorithm, called **binary exponential backoff**, was chosen to dynamically adapt to the number of stations trying to send. If the randomization interval for all collisions was 1023, the chance of two stations colliding for a second time would be negligible, but the average wait after a collision would be hundreds of slot times, introducing significant delay.

On the other hand, if each station always delayed for either zero or one slots, then if 100 stations ever tried to send at once, they would collide over and over until 99 of them picked 1 and the remaining station picked 0. This might take years.

By having the randomization interval grow exponentially as more and more consecutive collisions occur, the algorithm ensures a low delay when only a few stations collide but also ensures that the collision is resolved in a reasonable interval when many stations collide. Truncating the backoff at 1023 keeps the bound from growing too large.

As described so far, CSMA/CD provides no acknowledgements. Since the mere absence of collisions does not guarantee that bits were not garbled by noise spikes on the cable, for reliable communication the destination must verify the checksum, and if correct, send back an acknowledgement frame to the source.

Normally, this acknowledgement would be just another frame as far as the protocol is concerned and would have to fight for channel time just like a data frame. However, a simple modification to the contention algorithm would allow speedy confirmation of frame receipt.

5. Switched Ethernet:

With the growth of multimedia, transmission using 100-Mbps or 1-Gbps Ethernet can even become saturated.

Fortunately, there is an additional way to deal with increased load: switched Ethernet, as shown in Fig. The heart of this system is a **switch** containing a high-speed backplane and room for typically 4 to 32 plug-in line cards, each containing one to eight connectors. Most often, each connector has a 10Base-T twisted pair connection to a single host computer.

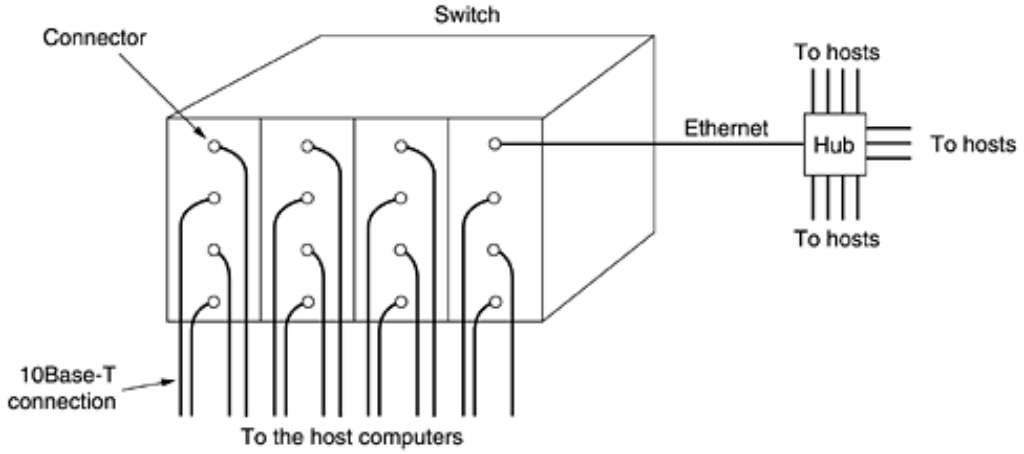


Figure (a). A simple example of switched Ethernet.

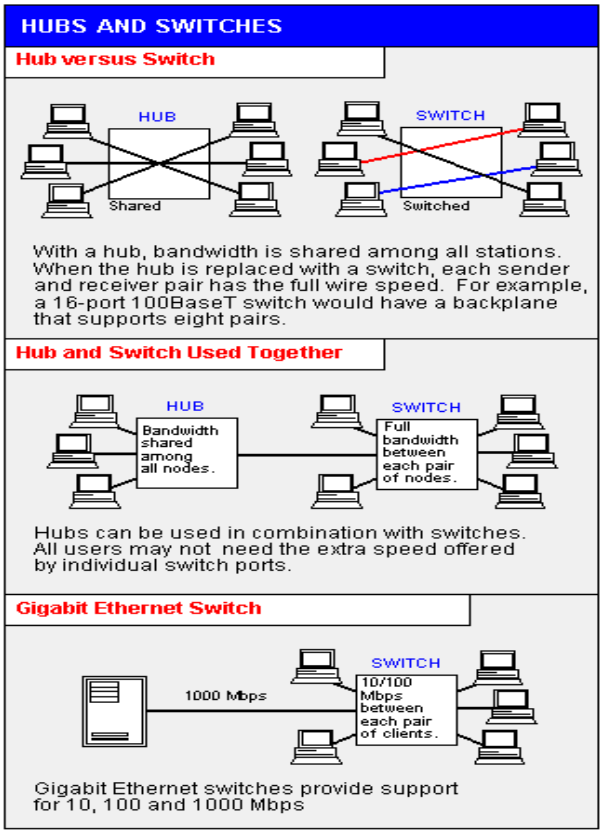


Figure (b). Hubs and Switches used in Ethernet.

When a station wants to transmit an Ethernet frame, it outputs a standard frame to the switch. The plug-in card getting the frame may check to see if it is destined for one of the other stations connected to the same card. If so, the frame is copied there. If not, the frame is sent over the high-speed backplane to the destination station's card. The backplane typically runs at many Gbps, using a proprietary protocol.

If two machines attached to the same plug-in card transmit frames at the same time. It depends on how the card has been constructed. One possibility is for all the ports on the card to be wired together to form a local on-card LAN. Collisions on this on-card LAN will be detected and handled the same as any other collisions on a CSMA/CD network—with retransmissions using the binary exponential backoff algorithm.

With this kind of plug-in card, only one transmission per card is possible at any instant, but all the cards can be transmitting in parallel. With this design, each card forms its own **collision domain**, independent of the others. With only one station per collision domain, collisions are impossible and performance is improved.

With the other kind of plug-in card, each input port is buffered, so incoming frames are stored in the card's on-board RAM as they arrive. This design allows all input ports to receive (and transmit) frames at the same time, for parallel, full-duplex operation, something not possible with CSMA/CD on a single channel. Once a frame has been completely received, the card can then check to see if the frame is destined for another port on the same card or for a distant port.

In the former case, it can be transmitted directly to the destination. In the latter case, it must be transmitted over the backplane to the proper card. With this design, each port is a separate collision domain, so collisions do not occur.

Since the switch just expects standard Ethernet frames on each input port, it is possible to use some of the ports as concentrators. In Fig (a), the port in

the upper-right corner is connected not to a single station, but to a 12-port hub. As frames arrive at the hub, they contend for the ether in the usual way, including collisions and binary backoff. Successful frames make it to the switch and are treated there like any other incoming frames: they are switched to the correct output line over the high-speed backplane. Hubs are cheaper than switches, but due to falling switch prices, they are rapidly becoming obsolete. Nevertheless, legacy hubs still exist.

6. IEEE 802.2: Logical Link Control

IEEE has defined one that can run on top of Ethernet and the other 802 protocols. In addition, this protocol, called LLC (Logical Link Control), hides the differences between the various kinds of 802 networks by providing a single format and interface to the network layer. LLC forms the upper half of the data link layer, with the MAC sublayer below it, as shown in Fig.

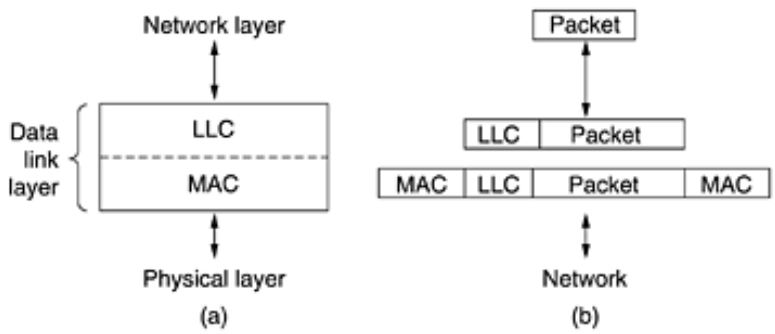


Figure. (a) Position of LLC. (b) Protocol formats.

Typical usage of LLC is as follows. The network layer on the sending machine passes a packet to LLC, using the LLC access primitives. The LLC sublayer then adds an LLC header, containing sequence and acknowledgement numbers. The resulting structure is then inserted into the payload field of an 802 frame and transmitted. At the receiver, the reverse process takes place.

LLC provides three service options: unreliable datagram service, acknowledged datagram service, and reliable connection-oriented service. The LLC header contains three fields: a destination access point, a source access point, and a control field.

The access points tell which process the frame came from and where it is to be delivered, replacing the DIX *Type* field. For the Internet, best-efforts attempts to deliver IP packets is sufficient, so no acknowledgements at the LLC level are required.

4.3 Data Link Layer Switching:

Many organizations have multiple LANs and wish to connect them. LANs can be connected by devices called bridges, which operate in the data link layer. Bridges examine the data layer link addresses to do routing. Before getting into the technology of bridges, it is worthwhile taking a look at some common situations in which bridges are used. We will mention six reasons why a single organization may end up with multiple LANs.

First, many university and corporate departments have their own LANs, primarily to connect their own personal computers, workstations, and servers. Since the goals of the various departments differ, different departments choose different LANs, without regard to what other departments are doing. Sooner or later, there is a need for interaction, so bridges are needed. In this example, multiple LANs came into existence due to the autonomy of their owners.

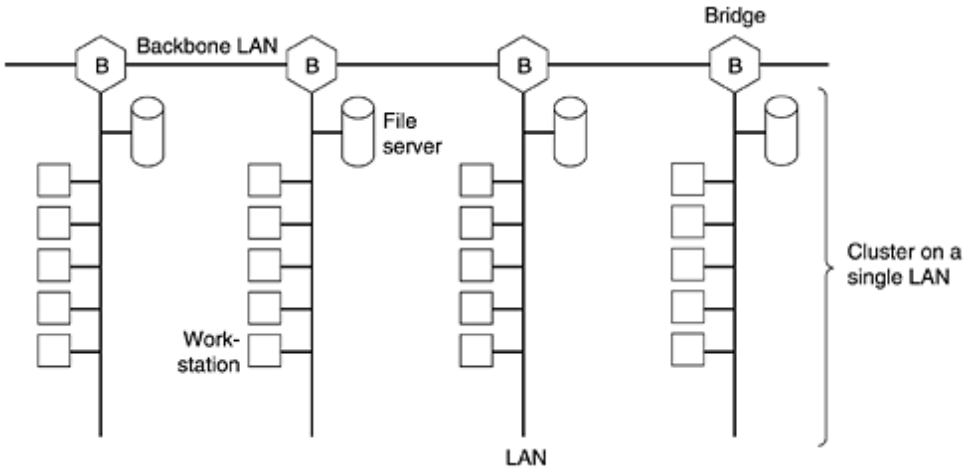
Second, the organization may be geographically spread over several buildings separated by considerable distances. It may be cheaper to have separate LANs in each building and connect them with bridges and laser links than to run a single cable over the entire site.

Third, it may be necessary to split what is logically a single LAN into separate LANs to accommodate the load. At many universities, for example, thousands of workstations are available for student and faculty computing.

Files are normally kept on file server machines and are downloaded to users' machines upon request.

The enormous scale of this system precludes putting all the workstations on a single LAN—the total bandwidth needed is far too high. Instead, multiple LANs connected by bridges are used, as shown in Fig below.

Each LAN contains a cluster of workstations with its own file server so that most traffic is restricted to a single LAN and does not add load to the backbone.



It is worth noting that although we usually draw LANs as multidrop cables as in Fig. 4-39 (the classic look), they are more often implemented with hubs or especially switches nowadays. However, a long multidrop cable with multiple machines plugged into it and a hub with the machines connected inside the hub are functionally identical. In both cases, all the machines belong to the same collision domain, and all use the CSMA/CD protocol to send frames. Switched LANs are different, however, as we saw before and will see again shortly.

Fourth, in some situations, a single LAN would be adequate in terms of the load, but the physical distance between the most distant machines is too

great (e.g., more than 2.5 km for Ethernet). Even if laying the cable is easy to do, the network would not work due to the excessively long round-trip delay. The only solution is to partition the LAN and install bridges between the segments. Using bridges, the total physical distance covered can be increased.

Fifth, there is the matter of reliability. On a single LAN, a defective node that keeps outputting a continuous stream of garbage can cripple the LAN. Bridges can be inserted at critical places, like fire doors in a building, to prevent a single node that has gone berserk from bringing down the entire system. Unlike a repeater, which just copies whatever it sees, a bridge can be programmed to exercise some discretion about what it forwards and what it does not forward.

Sixth, and last, bridges can contribute to the organization's security. Most LAN interfaces have a promiscuous mode, in which all frames are given to the computer, not just those addressed to it. Spies and busybodies love this feature. By inserting bridges at various places and being careful not to forward sensitive traffic, a system administrator can isolate parts of the network so that its traffic cannot escape and fall into the wrong hands.

Ideally, bridges should be fully transparent, meaning it should be possible to move a machine from one cable segment to another without changing any hardware, software, or configuration tables.

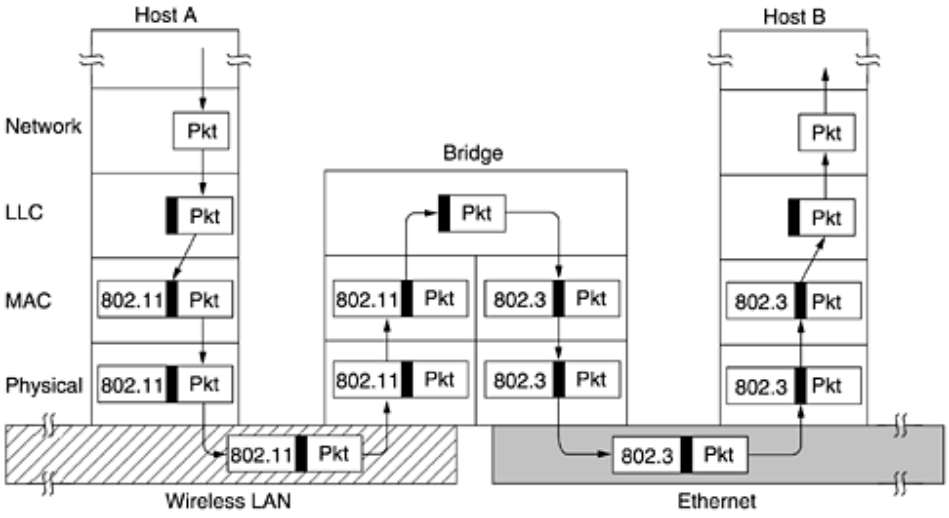
4.1 Bridges from 802.x to 802.y:

Having seen why bridges are needed, let us now turn to the question of how they work. Figure 4-40 illustrates the operation of a simple two-port bridge. Host A on a wireless (802.11) LAN has a packet to send to a fixed host, B, on

an (802.3) Ethernet to which the wireless LAN is connected. The packet descends into the LLC sublayer and acquires an LLC header (shown in black in the figure). Then it passes into the MAC sublayer and an 802.11 header is prepended to it (also a trailer, not shown in the figure).

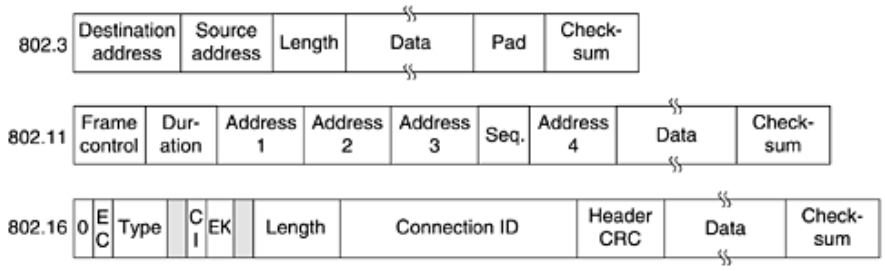
This unit goes out over the air and is picked up by the base station, which sees that it needs to go to the fixed Ethernet. When it hits the bridge connecting the 802.11 network to the 802.3 network, it starts in the physical layer and works

its way upward. In the MAC sublayer in the bridge, the 802.11 header is stripped off. The bare packet (with LLC header) is then handed off to the LLC sublayer in the bridge.



So far it looks like moving a frame from one LAN to another is easy. Such is not the case. In this section we will point out some of the difficulties that one encounters when trying to build a bridge between the various 802 LANs (and MANs). We will focus on 802.3, 802.11, and 802.16, but there are others as well, each with its unique problems.

1) To start with, each of the LANs uses a different frame format.



2) A second problem is that interconnected LANs do not necessarily run at the same data rate. When forwarding a long run of back-to-back frames from a fast LAN to a slower one, the bridge will not be able to get rid of the frames as fast as they come in. For example, if a gigabit Ethernet is pouring bits into an 11-Mbps 802.11b LAN at top speed, the bridge will have to buffer them, hoping not to run out of memory.

3) A third problem, and potentially the most serious of all, is that different 802 LANs have different maximum frame lengths. An obvious problem arises when a long frame must be forwarded onto a LAN that cannot accept it. Splitting the frame into pieces is out of the question in this layer. All the protocols assume that frames either arrive or they do not. There is no provision for reassembling frames out of smaller units.

4) Another point is security. Both 802.11 and 802.16 support encryption in the data link layer. Ethernet does not. This means that the various encryption services available to the wireless networks are lost when traffic passes over an Ethernet. Worse yet, if a wireless station uses data link layer encryption, there will be no way to decrypt it when it arrives over an Ethernet. One solution to the security problem is to do encryption in a higher layer.

5) A final point is quality of service. Both 802.11 and 802.16 provide it in various forms, the former using PCF mode and the latter using constant bit rate connections. Ethernet has no concept of quality of service, so traffic from

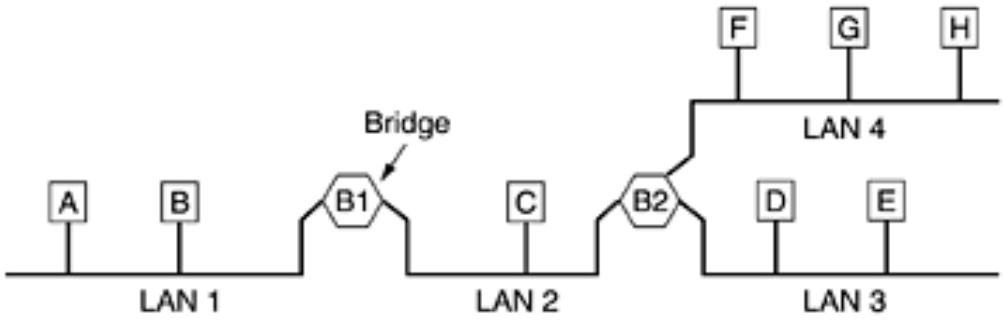
either of the others will lose its quality of service when passing over an Ethernet.

4.2 Local Internetworking:

The previous section dealt with the problems encountered in connecting two different IEEE 802 LANs via a single bridge. However, in large organizations with many LANs, just interconnecting them all raises a variety of issues, even if they are all just Ethernet. Ideally, it should be possible to go out and buy bridges designed to the IEEE standard, plug the connectors into the bridges, and everything should work perfectly, instantly.

There should be no hardware changes required, no software changes required, no setting of address switches, no downloading of routing tables or parameters, nothing. Just plug in the cables and walk away. Furthermore, the operation of the existing LANs should not be affected by the bridges at all.

In its simplest form, a transparent bridge operates in promiscuous mode, accepting every frame transmitted on all the LANs to which it is attached. As an example, consider the configuration of Fig. Bridge B1 is connected to LANs 1 and 2, and bridge B2 is connected to LANs 2, 3, and 4. A frame arriving at bridge B1 on LAN 1 destined for A can be discarded immediately, because it is already on the correct LAN, but a frame arriving on LAN 1 for C or F must be forwarded.



When a frame arrives, a bridge must decide whether to discard or forward it, and if the latter, on which LAN to put the frame. This decision is made by looking up the destination address in a big (hash) table inside the bridge. The table can list each possible destination and tell which output line (LAN) it belongs on. For example, B2's table would list A as belonging to LAN 2, since all B2 has to know is which LAN to put frames for A on. That, in fact, more forwarding happens later is not of interest to it.

When the bridges are first plugged in, all the hash tables are empty. None of the bridges know where any of the destinations are, so they use a flooding algorithm: every incoming frame for an unknown destination is output on all the LANs to which the bridge is connected except the one it arrived on. As time goes on, the bridges learn where destinations are, as described below. Once a destination is known, frames destined for it are put on only the proper LAN and are not flooded.

The algorithm used by the transparent bridges is backward learning. As mentioned above, the bridges operate in promiscuous mode, so they see every frame sent on any of their LANs. By looking at the source address, they can tell which machine is accessible on which LAN.

For example, if bridge B1 in Fig. sees a frame on LAN 2 coming from C, it knows that C must be reachable via LAN 2, so it makes an entry in its hash table noting that frames going to C should use LAN 2. Any subsequent frame addressed to C coming in on LAN 1 will be forwarded, but a frame for C coming in on LAN 2 will be discarded.

The topology can change as machines and bridges are powered up and down and moved around. To handle dynamic topologies, whenever a hash table entry is made, the arrival time of the frame is noted in the entry. Whenever a

frame whose source is already in the table arrives, its entry is updated with the current time. Thus, the time associated with every entry tells the last time a frame from that machine was seen.

Periodically, a process in the bridge scans the hash table and purges all entries more than a few minutes old. In this way, if a computer is unplugged from its LAN, moved around the building, and plugged in again somewhere else, within a few minutes it will be back in normal operation, without any manual intervention.

The routing procedure for an incoming frame depends on the LAN it arrives on (the source LAN) and the LAN its destination is on (the destination LAN), as follows:

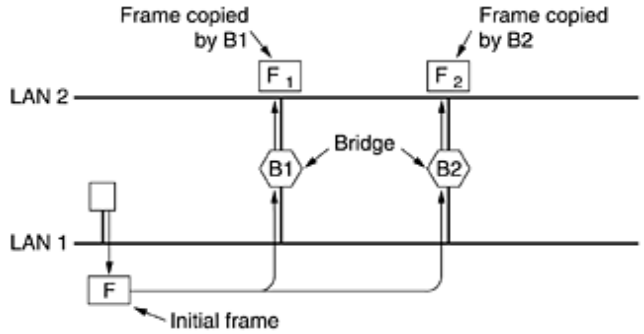
1. If destination and source LANs are the same, discard the frame.

2. If the destination and source LANs are different, forward the frame.

3. If the destination LAN is unknown, use flooding.

4.3 Spanning Tree Bridges:

To increase reliability, some sites use two or more bridges in parallel between pairs of LANs, as shown in Fig. 4-43. This arrangement, however, also introduces some additional problems because it creates loops in the topology. Fig. Shows two parallel transparent bridges.



A simple example of these problems can be seen by observing how a frame, F, with unknown destination is handled in Fig. 4-43. Each bridge, following the normal rules for handling unknown destinations, uses flooding, which in this example just means copying it to LAN 2. Shortly thereafter, bridge 1 sees F2, a frame with an unknown destination, which it copies to LAN 1, generating F3 (not shown). Similarly, bridge 2 copies F1 to LAN 1 generating F4 (also not shown). Bridge 1 now forwards F4 and bridge 2 copies F3. This cycle goes on forever.

The solution to this difficulty is for the bridges to communicate with each other and overlay the actual topology with a spanning tree that reaches every LAN. In effect, some potential connections between LANs are ignored in the interest of constructing a fictitious loop-free topology.

For example, in Fig.(a) we see nine LANs interconnected by ten bridges. This configuration can be abstracted into a graph with the LANs as the nodes. An arc connects any two LANs that are connected by a bridge. The graph can be reduced to a spanning tree by dropping the arcs shown as dotted lines in Fig.(b).

Using this spanning tree, there is exactly one path from every LAN to every other LAN. Once the bridges have agreed on the spanning tree, all forwarding between LANs follows the spanning tree. Since there is a unique path from each source to each destination, loops are impossible.

To build the spanning tree, first the bridges have to choose one bridge to be the root of the tree. They make this choice by having each one broadcast its serial number, installed by the manufacturer and guaranteed to be unique worldwide.

The bridge with the lowest serial number becomes the root. Next, a tree of shortest paths from the root to every bridge and LAN is constructed. This tree is the spanning tree. If a bridge or LAN fails, a new one is computed.

The result of this algorithm is that a unique path is established from every LAN to the root and thus to every other LAN. Although the tree spans all the LANs, not all the bridges are necessarily present in the tree (to prevent loops). Even after the spanning tree has been established, the algorithm continues to run during normal operation in order to automatically detect topology changes and update the tree.

4.4 Remote Bridges:

A common use of bridges is to connect two (or more) distant LANs. For example, a company might have plants in several cities, each with its own LAN. Ideally, all the LANs should be interconnected, so the complete system acts like one large LAN.

This goal can be achieved by putting a bridge on each LAN and connecting the bridges pairwise with point-to-point lines (e.g., lines leased from a telephone company). A simple system, with three LANs, is illustrated in Fig. 4-45. The usual routing algorithms apply here. The simplest way to see this is to regard the three point-to-point lines as host less LANs. Then we have a normal system of six LANS interconnected by four bridges.

Various protocols can be used on the point-to-point lines. One possibility is to choose some standard point-to-point data link protocol such as PPP, putting complete MAC frames in the payload field. This strategy works best if all the LANs are identical, and the only problem is getting frames to the correct LAN. Another option is to strip off the MAC header and trailer at the source bridge and put what is left in the payload field of the point-to-point

protocol. A new MAC header and trailer can then be generated at the destination bridge. A disadvantage of this approach is that the checksum that arrives at the destination host is not the one computed by the source host, so errors caused by bad bits in a bridge's memory may not be detected.

4.5 Repeaters, Hubs, Bridges, Switches, Routers, and Gateways: [To be read from Text book Pg. No. 242]

4.6 Virtual LANs: [To be read from Text book Pg. No. 244]

****

Unit- 5

The Network Layer

Content:

1. Network Layer Design Issues
2. Routing Algorithms
3. Congestion Control Algorithms
4. IPV4 Address

I. Introduction-

The network layer is concerned with getting packets from the source all the way to the destination. Getting to the destination may require making many hops at intermediate routers along the way. This function clearly contrasts with that of the data link layer, which has the more modest goal of just moving frames from one end of a wire to the other. Thus, the network layer is the lowest layer that deals with end-to-end transmission.

To achieve its goals, the network layer must know about the topology of the communication subnet (i.e., the set of all routers) and choose appropriate paths through it. It must also take care to choose routes to avoid overloading some of the communication lines and routers while leaving others idle. Finally, when the source and destination are in different networks, new problems occur. It is up to the network layer to deal with them. In this chapter we will study all these issues and illustrate them, primarily using the Internet and its network layer protocol, IP, although wireless networks will also be addressed.

1. Network Layer Design Issues:

In the following sections we will provide an introduction to some of the issues that the designers of the network layer must grapple with. These issues include the service provided to the transport layer and the internal design of the subnet.

1.1 Store-and-Forward Packet Switching:

The major components of the system are the carrier's equipment (routers connected by transmission lines), shown inside the shaded oval, and the customers' equipment, shown outside the oval.

Host *H1* is directly connected to one of the carrier's routers, *A*, by a leased line. In contrast, *H2* is on a LAN with a router, *F*, owned and operated by the customer. This router also has a leased line to the carrier's equipment. We have shown *F* as being outside the oval because it does not belong to the carrier, but in terms of construction, software, and protocols, it is probably no different from the carrier's routers.

Whether it belongs to the subnet is arguable, but for the purposes of this chapter, routers on customer premises are considered part of the subnet because they run the same algorithms as the carrier's routers (and our main concern here is algorithms).

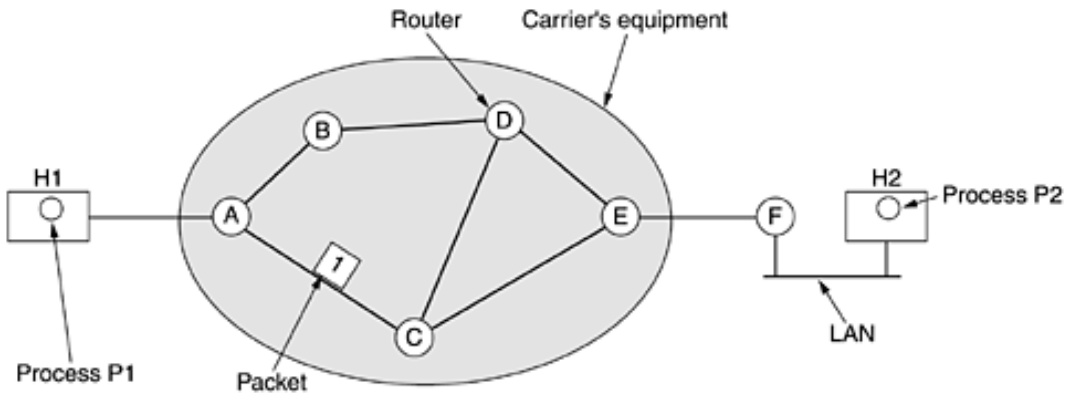


Figure-1. The environment of the network layer protocols.

This equipment is used as follows. A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier. The packet is stored there until it has fully arrived so the checksum can be verified. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered.

1.2 Services Provided to the Transport Layer:

The network layer provides services to the transport layer at the network layer/transport layer interface. An important question is what kind of services the network layer provides to the transport layer. The network layer services have been designed with the following goals in mind.

1. The services should be independent of the router technology.

2. The transport layer should be shielded from the number, type, and topology of the routers present.

3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Given these goals, the designers of the network layer have a lot of freedom in writing detailed specifications of the services to be offered to the transport layer. This freedom often degenerates into a raging battle between two warring factions. The discussion centres on whether the network layer should provide connection-oriented service or connectionless service.

5.1.3 Implementation of Connectionless Service:

If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called **datagrams** (in analogy with telegrams) and the subnet is called a **datagram subnet**.

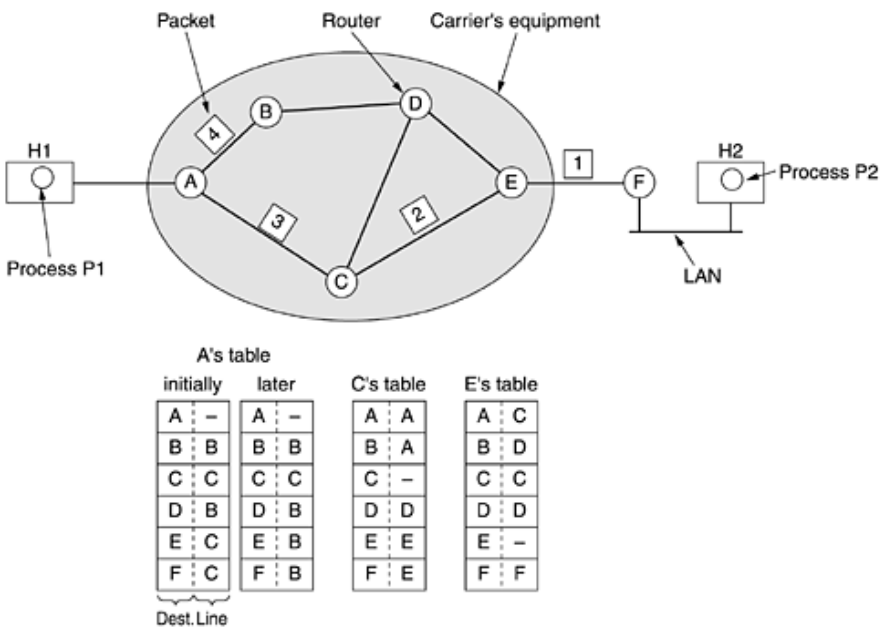


Figure-2. Routing within a datagram subnet.

Let us now see how a datagram subnet works. Suppose that the process *P1* in Fig-2 has a long message for *P2*. It hands the message to the transport layer with instructions to deliver it to process *P2* on host *H2*. The transport layer code runs on *H1*, typically within the operating system. It prepends a transport header to the front of the message and hands the result to the network layer, probably just another procedure within the operating system.

Let us assume that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4 and sends each of them in turn to router A using some point-to-point protocol. At this point the carrier takes over. Every router has an internal table telling it where to send packets for each possible destination. Each table entry is a pair consisting of a destination and the outgoing line to use for that destination. Only directly-connected lines can be used.

For example, in Fig-2, A has only two outgoing lines—to B and C—so every incoming packet must be sent to one of these routers, even if the ultimate destination is some other router. A's initial routing table is shown in the figure under the label ''initially.''

As they arrived at A, packets 1, 2, and 3 were stored briefly (to verify their checksums). Then each was forwarded to C according to A's table. Packet 1 was then forwarded to E and then to F. When it got to F, it was encapsulated in a data link layer frame and sent to H2 over the LAN. Packets 2 and 3 follow the same route.

However, something different happened to packet 4. When it got to A it was sent to router B, even though it is also destined for F. For some reason, A decided to send packet 4 via a different route than that of the first three. Perhaps it learned of a traffic jam somewhere along the ACE path and updated its routing table, as shown under the label ''later.'' The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

1.4 Implementation of Connection-Oriented Service:

If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a **VC** (**virtual circuit**), in analogy with the physical circuits set up by the telephone system, and the subnet is called a **virtual-circuit subnet**.

The idea behind virtual circuits is to avoid having to choose a new route for every packet sent, as in Fig-2. Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

As an example, consider the situation of Fig-3. Here, host H1 has established connection 1 with host H2. It is remembered as the first entry in each of the routing tables. The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1.

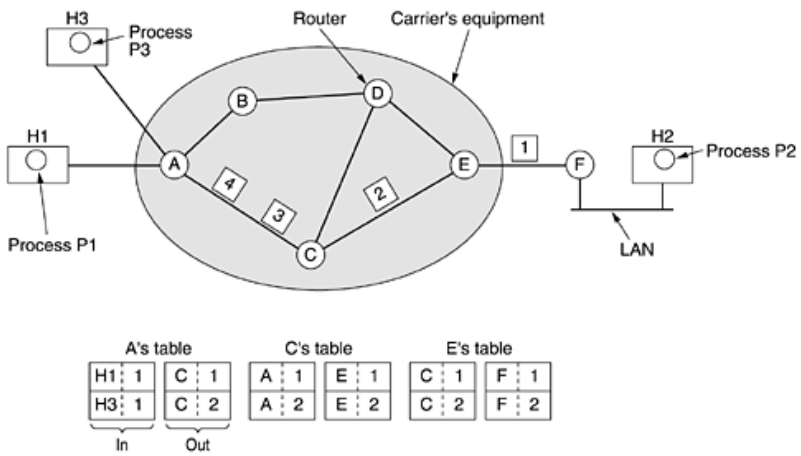


Figure-3. Routing within a virtual-circuit subnet.

Now let us consider what happens if H3 also wants to establish a connection to H2. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the subnet to establish the virtual circuit. This leads to the second row in the tables. Note that we have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets. In some contexts, this is called label switching.

1.5 Comparison of Virtual-Circuit and Datagram Subnets:

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

2. Routing Algorithms:

Routing is concerned with the question: Which line should router *J* use when forwarding a packet to router *K*?

There are two types of algorithms:

- *Adaptive* algorithms use such dynamic information as current topology, load, delay, etc. to select routes.
- In *nonadaptive* algorithms, routes never change once initial routes have been selected. Also called *static* routing.

Obviously, adaptive algorithms are more interesting, as nonadaptive algorithms don't even make an attempt to handle failed links.

Adaptive algorithms can be further divided in the following types:

1. *Isolated*: each router makes its routing decisions using only the *local* information it has on hand. Specifically, routers do not even exchange information with their neighbors.
2. *Centralized*: a centralized node makes all routing decisions. Specifically, the centralized node has access to *global* information.
3. *Distributed*: algorithms that use a combination of local and global information.

Non-Adaptive Algorithms

Flooding is a form of isolated routing. Does not select a specific route. When a router receives a packet, it sends a copy of the packet out on each line (except the one on which it arrived):

- To prevent packets from looping forever, each router decrements a hop count contained in the packet header. Whenever the hop count decrements to zero, the router discards the packet.
- To reduce looping even further:

1. Add a sequence number to each packet's header.
2. Each router maintains a private sequence number. When it sends a new packet, it copies the sequence number into the packet, and increments its private sequence number.
3. For each source router *S*, a router:
    - keeps track of the highest sequence number seen from *S*.
    - whenever it receives a packet from *S* containing a sequence number lower than the one stored in its table, it discards the packet.
    - otherwise, it updates the entry for *S* and forwards the packet on

Flooding has several important uses:

1. in military applications, the network must remain robust in the face of (extreme) hostility
2. sending routing updates, because updates can't rely on the correctness of a router's routing table.
3. theoretical--chooses all possible paths, so it chooses the shortest one.

In *selective flooding*, a router sends packets out only on those lines in the general direction of the destination. That is, don't send packets out on lines that clearly lead in the wrong direction.

In *random walk* a router selects one or more lines at random.

- equalize load (on network)
- robust

Shortest Path Algorithm

What if we ``know'' the complete topology of the network? Can look at computing the optimal path. What if we have the following network and we want to route a packet from node A to node G. What is the shortest path (do not initially show distance.

Use Dijkstra's algorithm (or variation). Basic idea:

- Choose the source, and put nodes connected to source in list to consider.
- From the list to consider choose the nearest node.

Algorithm results:

nodes	list to consider
(A, -, 0)	B, D, E
(B, A, 2)	C, D, E
(C, B, 3)	D, E, H
(H, C, 4)	D, E, G
(D, A, 5)	E, F, G
(E, C, 6)	G, F
(G, H, 7)	F
(F, E, 7)	-

Guaranteed to get the shortest path? How to prove? If an alternate shorter path to a node then we would have already tried the path.

Static Routing (Multipath Routing)

May not be a single *best* route between two nodes. Can use a routing table. Can compute alternate routes between source and destination by first computing shortest path and then removing these links.

Problem is that these routing tables are initially created and not changed.

Adaptive/Dynamic Routing

Centralized--use a *routing control center*. Creates, modifies, and distributes routing tables to other routers. Gathers information from the routers.

Good: adaptive routing, relieve burden on the routers of computing tables.

Problems:

- Does not adapt quickly.
- Quicker the adaption, the more overhead it causes.
- Synchronization of updates (some routers change, but not others, send at each other).
- If RCC crashes the network becomes stale.
- Lines near the RCC are overloaded.

Decentralized Routing

Isolated Routing

Base decisions on local traffic and conditions.

- hot potato--choose output line with the shortest queue
- backward learning--each packet contains source address and number of hops so far. Use this information to learn shortest path to each source. Will learn shortest path to all routers. What is wrong? Only deal with good news, not bad. ``Good'' may no longer be good due to down router or congestion.

Must periodically forget and start over (with suboptimal performance after a purge).

Delta Routing

Combine centralized and isolated routing. RCC (routing control center) computes optimal paths for each router as before. Has a parameter δ to decide if two lines are equivalent. Are equivalent if the delay is within δ of each other.

$\delta \to 0$, centralized
$\delta \to \infty$, isolated, up to the routers.

Modern networks use two dynamic (adaptive) algorithms--distance vector routing and link state routing.

Distance Vector Routing

Distributed routing algorithm (Old Arpanet Routing Algorithm). routers work together.

1. Each router maintains a table (vector) giving the best-known distance to a destination and the line to use for sending there. Tables are updated by exchanging information with neighbors.
2. Each router knows the distance (cost) of reaching its neighbors (e.g. send echo requests).
3. routers periodically exchange routing tables with each of their neighbors.
4. Upon receipt of an update, for each destination in its table, a router:
    1. compares the metric in its local table with the metric in the neighbor's table plus the cost of reaching that neighbor
    2. if the path via the neighbor has a lower cost, the router updates its local table to forward packets to the neighbor

This algorithm was used in the original ARPANET. Unfortunately, it suffers from the problem: good news travels quickly, bad news travels slowly (count-to-infinity problem).

The fundamental problem with the old Arpanet algorithm is that it continues to use ``old'' information that is invalid, even after newer information becomes available.

Link State Routing

The ``old'' Arpanet routing algorithm was replaced in 1979. Problems with old algorithm included:

1. High-priority routing update packets were large, adversely affecting traffic.
2. Network was too slow in adapting to congestion, too fast to react to minor changes.
3. Average queue length was used to estimate delay. This works only if all lines have the same capacity and propagation delay, and doesn't take into account that packets have varying sizes.

In the new algorithm:

1. Each router maintains a database describing the topology and link delays between each router. That is, each router keeps track of the full graph of links and nodes.
2. Each router periodically discovers it neighbors (sends ``hello'' message on booting) and measures the delays across its links (echo requests--should load be taken into account?), then forwards that information to all other routers (link state packets)
3. Updates are propagated at high priority using flooding. Updates contain sequence numbers, and a router forwards ``new'' copies of the packet.

Why use flooding? Because that way routing updates propagate even when routing tables aren't quite correct. ACKs are sent to neighbors.

4. Each router uses an SPF algorithm to calculate shortest paths based on the current values in its database.
5. Because each router makes its calculation using the same information, better routing decisions are made.

3. Congestion Control Algorithms:

Introduction:

As Internet can be considered as a *Queue of packets,* where transmitting nodes are constantly adding packets and some of them (receiving nodes) are removing packets from the queue. So, consider a situation where too many packets are present in this queue (or internet or a part of internet), such that constantly transmitting nodes are pouring packets at a higher rate than receiving nodes are removing them. This degrades the performance, and such a situation is termed as *Congestion*. Main reason of congestion is more number of packets into the network than it can handle. So, the objective of congestion control can be summarized as to maintain the number of packets in the network below the level at which performance falls off dramatically. The nature of a Packet switching network can be summarized in following points:

A network of queues:

At each node, there is a queue of packets for each outgoing channel

If packet arrival rate exceeds the packet transmission rate, the queue size grows without bound

When the line for which packets are queuing becomes more than 80% utilized, the queue length grows alarmingly. When the number of packets dumped into the network is within the carrying capacity, they all are delivered, expect a few that have too be rejected due to transmission errors). And then the number delivered is proportional to the number of packets sent. However, as traffic increases too far, the routers are no longer able to cope, and they begin to lose packets. This tends to make matter worse. At very high traffic, performance collapse completely, and almost no packet is delivered. In the following sections, the causes of congestion, the effects of congestion and various congestion control techniques are discussed in detail.

Causes Of Congestion:

Congestion can occur due to several reasons. For example, if all of a sudden a stream of packets arrive on several input lines and need to be out on the same output line, then a long queue will be build up for that output. If there is *insufficient memory* to hold these packets, then packets will be lost (dropped). Adding more memory also may not help in certain situations. If router have an infinite amount of memory even then instead of congestion being reduced, it gets worse; because by the time packets gets at the head of the queue, to be dispatched out to the output line, they have already timed-out (repeatedly), and duplicates may also be present. All the packets will be forwarded to next router up to the destination, all the way only increasing the load to the network more and more. Finally when it arrives at the destination, the packet will be discarded, due to time out, so instead of been dropped at any intermediate router (in case memory is restricted) such a packet goes all the way up to the destination, increasing the network load throughout and then finally gets dropped there.

Slow processors also cause Congestion. If the router CPU is slow at performing the task required for them (Queuing buffers, updating tables, reporting any exceptions etc.), queue can build up even if there is excess of line capacity. Similarly, *Low-Bandwidth* lines can also cause congestion. Upgrading lines but not changing slow processors, or vice-versa, often helps a little; these can just shift the bottleneck to some other point. The real problem is the mismatch between different parts of the system.

Congestion tends to feed upon itself to get even worse. Routers respond to overloading by dropping packets. When these packets contain TCP segments, the segments don't reach their destination, and they are therefore left unacknowledged, which eventually leads to timeout and retransmission.

So, the major cause of congestion is often the *bursty* nature of traffic. If the hosts could be made to transmit at a uniform rate, then congestion problem will be less common and all other causes will not even led to congestion

because other causes just act as an enzyme which boosts up the congestion when the traffic is bursty (i.e., other causes just add on to make the problem more serious, main cause is the bursty traffic).

This means that when a device sends a packet and does not receive an acknowledgment from the receiver, in most the cases it can be assumed that the packets have been dropped by intermediate devices due to congestion. By detecting the rate at which segments are sent and not acknowledged, the source or an intermediate router can infer the level of congestion on the network. In the following section we shall discuss the ill effects of congestion.

Effects of Congestion:

Congestion affects two vital parameters of the network performance, namely *throughput* and *delay*. In simple terms, the throughput can be defined as the percentage utilization of the network capacity. Figure (a) shows how throughput is affected as offered load increases. Initially throughput increases linearly with offered load, because utilization of the network increases. However, as the offered load increases beyond certain limit, say 60% of the capacity of the network, the throughput drops. If the offered load increases further, a point is reached when not a single packet is delivered to any destination, which is commonly known as *deadlock* situation. There are three curves in Fig.(a), the ideal one corresponds to the situation when all the packets introduced are delivered to their destination up to the maximum capacity of the network. The second one corresponds to the situation when there is no congestion control. The third one is the case when some congestion control technique is used. This prevents the throughput collapse, but provides lesser throughput than the ideal condition due to overhead of the congestion control technique.

The delay also increases with offered load, as shown in Fig.(b). And no matter what technique is used for congestion control, the delay grows without bound as the load approaches the capacity of the system. It may be noted that initially

there is longer delay when congestion control policy is applied. However, the network without any congestion control will saturate at a lower offered load.

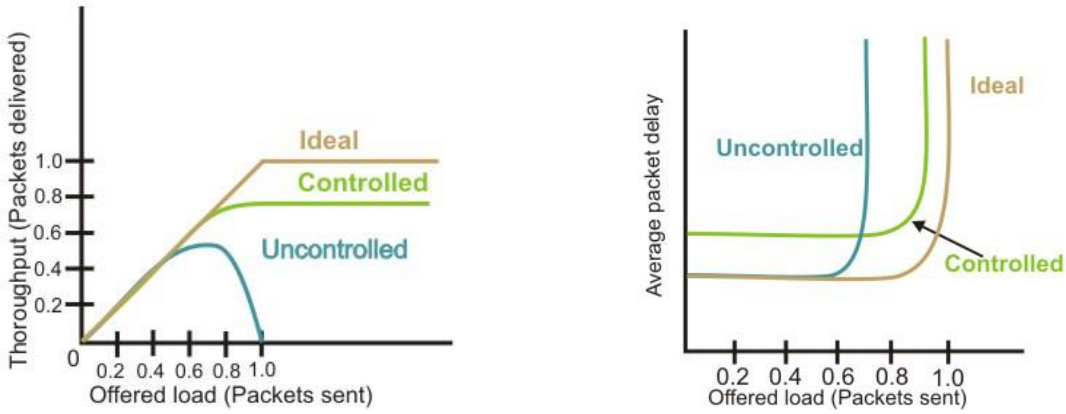


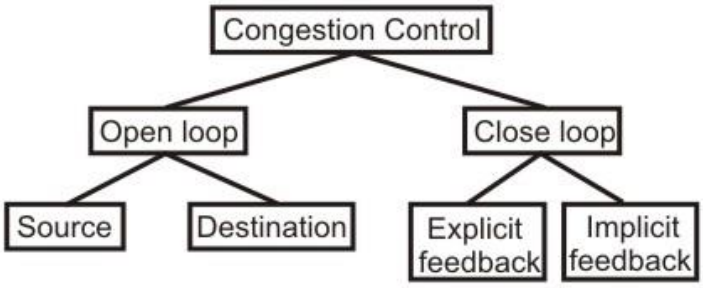
Figure (a) Effect of congestion on throughput (b) Effect of congestion on delay

Congestion Control Techniques:

Congestion control refers to the mechanisms and techniques used to control congestion and keep the traffic below the capacity of the network. As shown in Fig, the congestion control techniques can be broadly classified two broad categories:

Open loop: Protocols to prevent or avoid congestion, ensuring that the system (or network under consideration) never enters a Congested State.

Close loop: Protocols that allow system to enter congested state, detect it, and remove it.



The first category of solutions or protocols attempt to solve the problem by a good design, at first, to make sure that it doesn't occur at all. Once system is up and running midcourse corrections are not made. These solutions are somewhat static in nature, as the policies to control congestion don't change much according to the current state of the system. Such Protocols are also known as *Open Loop* solutions. These rules or policies include deciding upon when to accept traffic, when to discard it, making scheduling decisions and so on. Main point here is that they make decision without taking into consideration the current state of the network. The open loop algorithms are further divided on the basis of whether these acts on source versus that act upon destination.

The second category is based on the concept of feedback. During operation, some system parameters are measured and feed back to portions of the subnet that can take action to reduce the congestion. This approach can be divided into 3 steps:

- Monitor the system (network) to detect whether the network is congested or not and what's the actual location and devices involved.
- To pass this information to the places where actions can be taken
- Adjust the system operation to correct the problem.

These solutions are known as *Closed Loop* solutions. Various Metrics can be used to monitor the network for congestion. Some of them are: the average queue length, number of packets that are timed-out, average packet delay, number of packets discarded due to lack of buffer space, etc. A general feedback step would be, say a router, which detects the congestion send special packets to the source (responsible for the congestion) announcing the problem. These extra packets increase the load at that moment of time, but are necessary to bring down the congestion at a later time. Other approaches are also used at times to curtail down the congestion. For example, hosts or routers send out probe packets at regular intervals to explicitly ask about the congestion and source itself regulate its transmission rate, if congestion is

detected in the network. This kind of approach is a *pro-active* one, as source tries to get knowledge about congestion in the network and act accordingly.

Yet another approach may be where instead of sending information back to the source an intermediate router which detects the congestion send the information about the congestion to rest of the network, piggy backed to the outgoing packets. This approach will in no way put an extra load on the network (by not sending any kind of special packet for feedback). Once the congestion has been detected and this information has been passed to a place where the action needed to be done, then there are two basic approaches that can overcome the problem. These are: either to increase the resources or to decrease the load. For example, separate dial-up lines or alternate links can be used to increase the bandwidth between two points, where congestion occurs. Another example could be to decrease the rate at which a particular sender in transmitting packets out into the network.

The closed loop algorithms can also be divided into two categories, namely *explicit feedback* and *implicit feedback* algorithms. In the explicit approach, special packets are sent back to the sources to curtail down the congestion. While in implicit approach, the source itself acts pro-actively and tries to deduce the existence of congestion by making local observations.

In the following sections we shall discuss about some of the popular algorithms from the above categories.

Leaky Bucket Algorithm:

Consider a Bucket with a small hole at the bottom, whatever may be the rate of water pouring into the bucket, the rate at which water comes out from that small hole is constant. This scenario is depicted in figure(a). Once the bucket is full, any additional water entering it spills over the sides and is lost (i.e. it doesn't appear in the output stream through the hole underneath).

The same idea of leaky bucket can be applied to packets, as shown in Fig. (b). Conceptually each network interface contains a *leaky bucket*. And the following steps are performed:

When the host has to send a packet, the packet is thrown into the bucket.
The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
Bursty traffic is converted to a uniform traffic by the leaky bucket.
In practice the bucket is a finite queue that outputs at a finite rate.

This arrangement can be simulated in the operating system or can be built into the hardware. Implementation of this algorithm is easy and consists of a finite queue. Whenever a packet arrives, if there is room in the queue it is queued up and if there is no room then the packet is discarded.

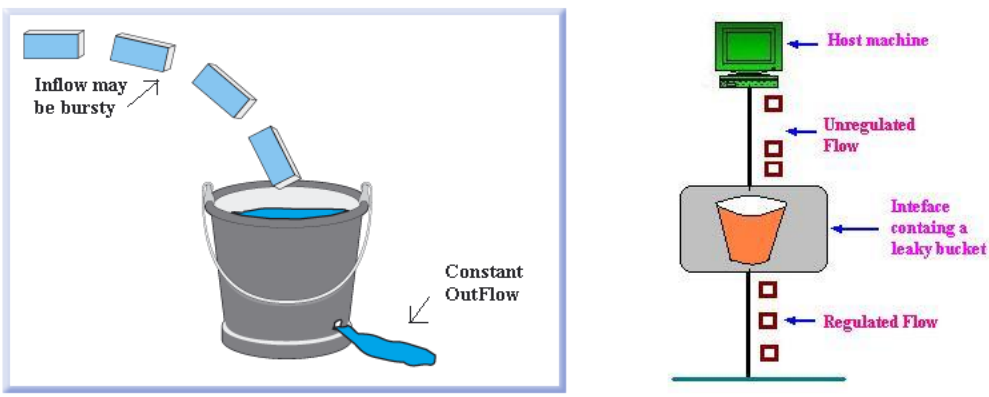


Figure (a) Leaky bucket (b) Leaky bucket implementation

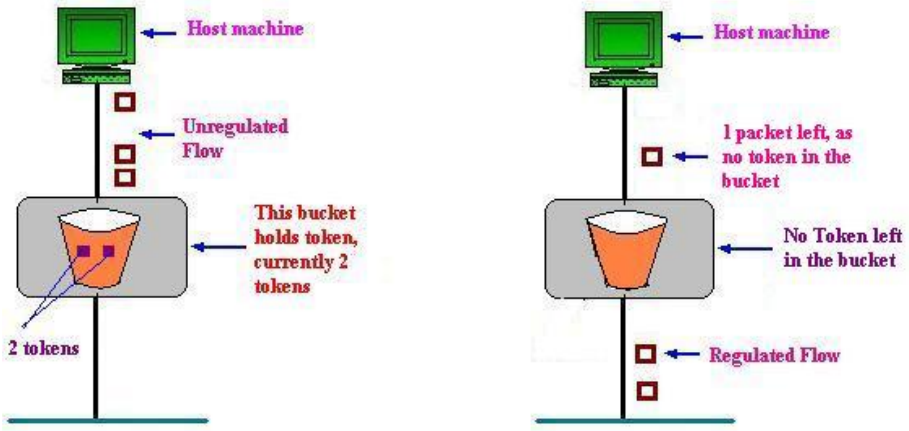
Token Bucket Algorithm:

The leaky bucket algorithm described above, enforces a rigid pattern at the output stream, irrespective of the pattern of the input. For many applications it is better to allow the output to speed up somewhat when a larger burst arrives than to loose the data. Token Bucket algorithm provides such a solution. In this algorithm leaky bucket holds token, generated at regular intervals. Main steps of this algorithm can be described as follows:

- In regular intervals tokens are thrown into the bucket.
- The bucket has a maximum capacity.
- If there is a ready packet, a token is removed from the bucket, and the packet is send.
- If there is no token in the bucket, the packet cannot be send.

Figure, shows the two scenarios before and after the tokens present in the bucket have been consumed. In Fig.(a) the bucket holds two tokens, and three packets are waiting to be sent out of the interface, in Fig.(b) two packets have been sent out by consuming two tokens, and 1 packet is still left.

The token bucket algorithm is less restrictive than the leaky bucket algorithm, in a sense that it allows bursty traffic. However, the limit of burst is restricted by the number of tokens available in the bucket at a particular instant of time.



Figure(a) Token bucket holding two tokens, before packets are send out, (b) Token bucket after two packets are send, one packet still remains as no token is left

Congestion control in virtual Circuit:

Till now we have discussed two open loop algorithms, where the policy decisions are made in the beginning, irrespective of the current state. Both leaky bucket algorithm and token bucket algorithm are open loop algorithms.

In this section we shall have a look at how the congestion is tackled in a virtual-circuit network. *Admission control* is one such closed-loop technique, where action is taken once congestion is detected in the network. Different approaches can be followed:

- Simpler one being: do not set-up new connections, once the congestion is signalled. This type of approach is often used in normal telephone networks. When the exchange is overloaded, then no new calls are established.

- Another approach, which can be followed is: to allow new virtual connections, but route these carefully so that none of the congested router (or none of the problem area) is a part of this route.
- Yet another approach can be: To negotiate different parameters between the host and the network, when the connection is setup. During the setup time itself, Host specifies the volume and shape of traffic, quality of service, maximum delay and other parameters, related to the traffic it would be offering to the network. Once the host specifies its requirement, the resources needed are reserved along the path, before the actual packet follows.

Choke Packet Technique:

The *choke packet* technique, a closed loop control technique, can be applied in both virtual circuit and datagram subnets. Each router monitors its resources and the utilization at each of its output line. There is a threshold set by the administrator, and whenever any of the resource utilization crosses this threshold and action is taken to curtail down this. Actually each output line has a utilization associated with it, and whenever this utilization crosses the threshold, the output line enters a “warning” state. If so, the router sends a *choke packet* back to the source, giving it a feedback to reduce the traffic. And the original packet is tagged (a bit is manipulated in the header field) so that it will not generate other choke packets by other intermediate router, which comes in place and is forwarded in usual way. It means that the first router (along the way of a packet), which detects any kind of congestion, is the only one that sends the choke packets.

When the source host gets the choke packet, it is required to reduce down the traffic send out to that particular destination (choke packet contains the destination to which the original packet was send out). After receiving the choke packet the source reduces the traffic by a particular fixed percentage, and this percentage decreases as the subsequent choke packets are received. Figure, depicts the functioning of choke packets.

For Example, when source A receives a choke packet with destination B at first, it will curtail down the traffic to destination B by 50%, and if again after affixed duration of time interval it receives the choke packet again for the same destination, it will further curtail down the traffic by 25% more and so on. As stated above that a source will entertain another subsequent choke packet only after a fixed interval of time, not before that. The reason for this is that when the first choke packet arrives at that point of time other packets destined to the same destination would also be there in the network and they will generate other choke packets too, the host should ignore these choke packets which refer to the same destination for a fixed time interval.

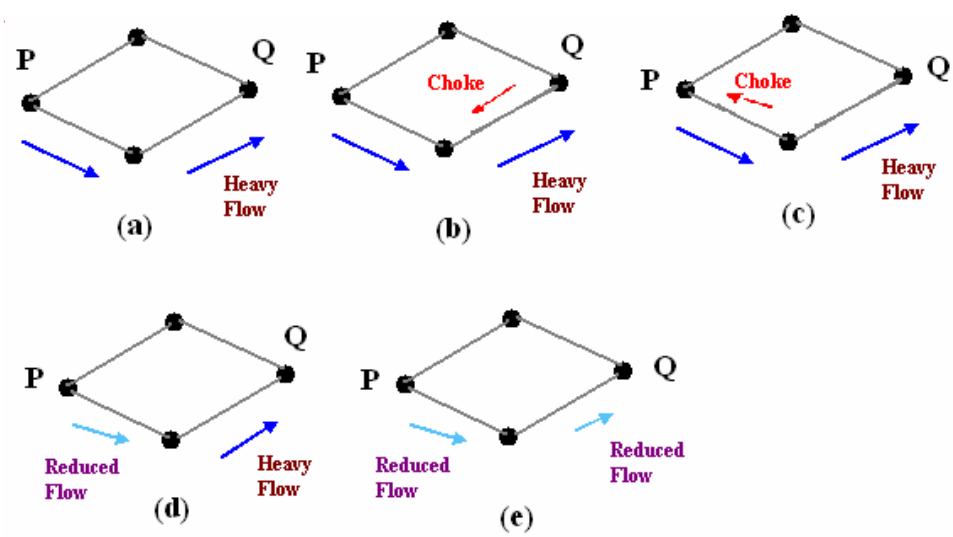


Figure: Depicts the functioning of choke packets, (a) Heavy traffic between nodes P and Q, (b) Node Q sends the Choke packet to P, (c) Choke packet reaches P, (d) P reduces the flow and send a reduced flow out, (e) Reduced flow reaches node Q

Hop-by Hop Choke Packets:

This technique is an advancement over Choked packet method. At high speed over long distances, sending a packet all the way back to the source doesn't help much, because by the time choke packet reach the source, already a lot of packets destined to the same original destination would be out from the source. So to help this, Hop-by-Hop Choke packets are used. In this approach, the choke packet affects each and every intermediate router through which it passes by. Here, as soon as choke packet reaches a router back to its path to the source, it curtails down the traffic between those intermediate routers. In this scenario, intermediate nodes must dedicate few more buffers for the incoming traffic as the outflow through that node will be curtailed down immediately as choke packet arrives it, but the input traffic flow will only be curtailed down when choke packet reaches the node which is before it in the original path. This method is illustrated in Fig.

As compared to choke packet technique, hop-by-hop choke packet algorithm is able to restrict the flow rapidly. One-step reduction is seen in controlling the traffic, this single step advantage is because in our example there is only one intermediate router. Hence, in a more complicated network, one can achieve a significant advantage by using hop-by-hop choke packet method.

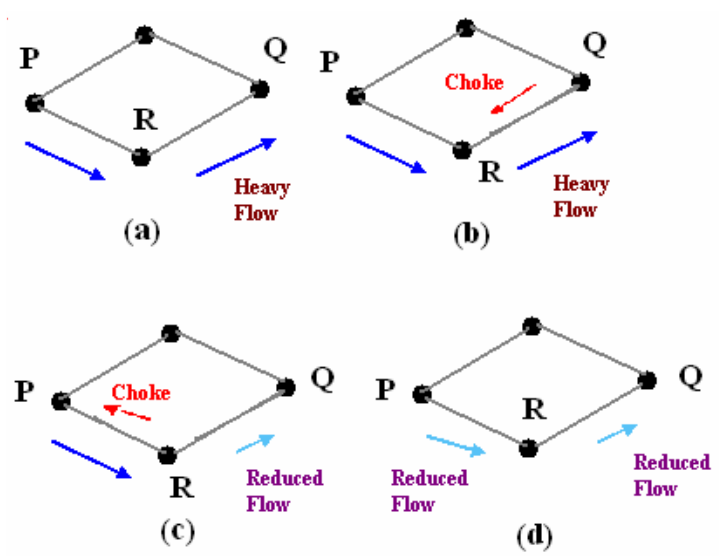


Figure: Depicts the functioning of Hop-by-Hop choke packets, (a) Heavy traffic between nodes P and Q, (b) Node Q sends the Choke packet to P, (c) Choke packet reaches R, and the flow between R and Q is curtail down, Choke packer reaches P, and P reduces the flow out

Load Shedding:

Another simple closed loop technique is *Load Shedding*; it is one of the simplest and more effective techniques. In this method, whenever a router finds that there is congestion in the network, it simply starts dropping out the packets. There are different methods by which a host can find out which packets to drop. Simplest way can be just choose the packets randomly which has to be dropped. More effective ways are there but they require some kind of cooperation from the sender too. For many applications, some packets are more important than others. So, sender can mark the packets in priority classes to indicate how important they are. If such a priority policy is implemented than intermediate nodes can drop packets from the lower priority classes and use the available bandwidth for the more important packets.

Slow Start - a Pro-active technique:

This is one of the pro-active techniques, which is used to avoid congestion. In the original implementation of TCP, as soon as a connection was established between two devices, they could each go “hog wild”, sending segments as fast as they liked as long as there was room in the other devices receive window. In a busy internet, the sudden appearance of a large amount of new traffic could aggravate any existing congestion.

To alleviate this, modern TCP devices are restrained in the rate at which they initially send segments. Each sender is at first restricted to sending only an amount of data equal to one “full-sized” segment—that is, equal to the MSS (maximum segment size) value for the connection. Each time an acknowledgment is received, the amount of data the device can send is increased by the size of another full-sized segment. Thus, the device “starts

slow” in terms of how much data it can send, with the amount it sends increasing until either the full window size is reached or congestion is detected on the link. In the latter case, the congestion avoidance feature is used.

When potential congestion is detected on a TCP link, a device responds by throttling back the rate at which it sends segments. A special algorithm is used that allows the device to drop the rate at which segments are sent quickly when congestion occurs. The device then uses the *Slow Start* algorithm just above to gradually increase the transmission rate back up again to try to maximize throughput without congestion occurring again.

Flow Control Versus Congestion control:

Let’s have a look at the difference between *Flow Control* and *Congestion Control*, which are mixed up at times.

Flow control is a very important part of regulating the transmission of data between devices, but it is limited in a way that it only considers what is going on within each of the devices on the connection, and not what is happening in devices between them. It relates to the point-point traffic between a given sender and a receiver. Flow control always involves some kind of feedback from receiver to sender to tell sender how things are at other end of the network. Since we are dealing with how TCP works between a typical server and client at layer four, we don't worry about how data gets between them; that's the job of the Internet Protocol at layer three.

In practice, what is going on at layer three can be quite important. Considered from an abstract point of view, our server and client may be connected “directly” using TCP, but all the packets we transmit are carried across an internet and routers between different networks. These networks and routers are also carrying data from many other connections and higher-layer protocols. If the internet becomes very busy, the speed at which segments are carried between the endpoints of our connection will be reduced, and they could even be dropped. This is called *congestion control*. Congestion control has to do with

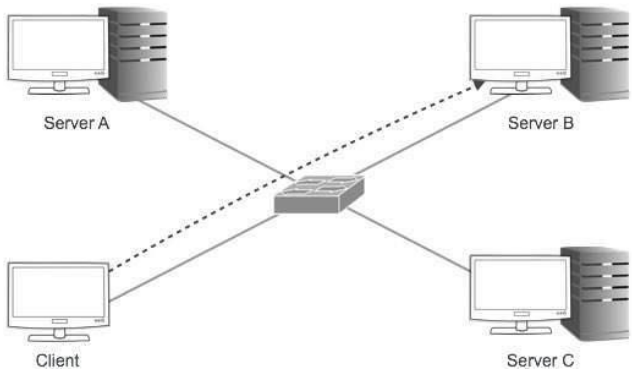
making sure that subnet carry the offered traffic. It is the global issue, involving the behaviour of all the hosts, router, link, store and forward mechanism between them in the entire subnet or internet.

4. IPV4 Address:

Addressing Modes:

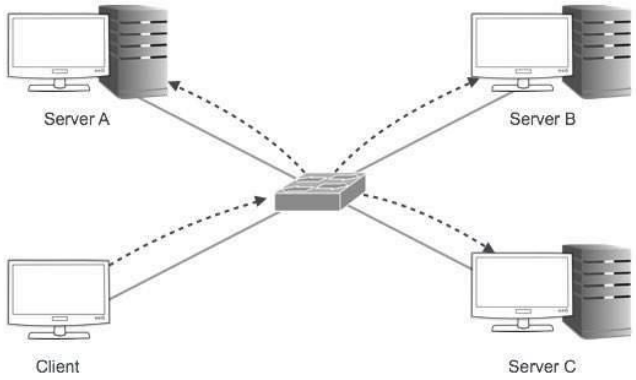
- Unicast Addressing Mode:

In this mode, data is sent only to one destined host. The Destination Address field contains 32- bit IP address of the destination host. Here the client sends data to the targeted server −



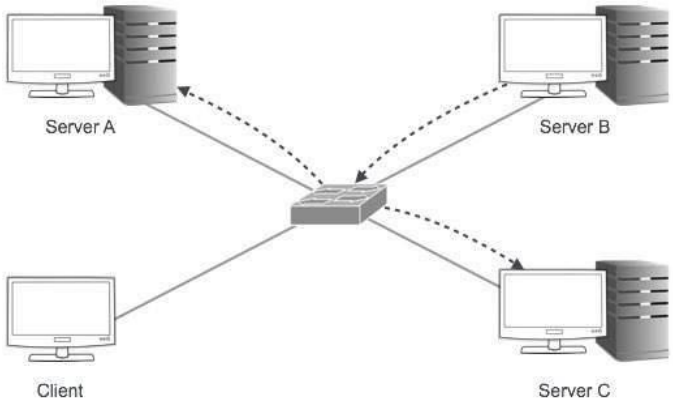
- Broadcast Addressing Mode:

In this mode, the packet is addressed to all the hosts in a network segment. The Destination Address field contains a special broadcast address, i.e. **255.255.255.255**. When a host sees this packet on the network, it is bound to process it. Here the client sends a packet, which is entertained by all the Servers −



- Multicast Addressing Mode:

  This mode is a mix of the previous two modes, i.e. the packet sent is neither destined to a single host nor all the hosts on the segment. In this packet, the Destination Address contains a special address which starts with 224.x.x.x and can be entertained by more than one host.



  Here a server sends packets which are entertained by more than one servers. Every network has one IP address reserved for the Network Number which represents the network and one IP address reserved for the Broadcast Address, which represents all the hosts in that network.

- Hierarchical Addressing Scheme:

  IPv4 uses hierarchical addressing scheme. An IP address, which is 32-bits in length, is divided into two or three parts as depicted –

8 bits	8 bits	8 bits	8 bits
Network	Network	Sub-Network	Host

A single IP address can contain information about the network and its sub-network and ultimately the host. This scheme enables the IP Address to be hierarchical where a network can have many sub-networks which in turn can have many hosts.

- Subnet Mask

The 32-bit IP address contains information about the host and its network. It is very necessary to distinguish both. For this, routers use Subnet Mask, which is as long as the size of the network address in the IP address. Subnet Mask is also 32 bits long. If the IP address in binary is ANDed with its Subnet Mask, the result yields the Network address. For example, say the IP Address is 192.168.1.152 and the Subnet Mask is 255.255.255.0 then −

IP	192.168.1.152	11000000	10101000	00000001	10011000	ANDed
Mask	255.255.255.0	11111111	11111111	11111111	00000000	
Network	192.168.1.0	11000000	10101000	00000001	00000000	Result

This way the Subnet Mask helps extract the Network ID and the Host from an IP Address. It can be identified now that 192.168.1.0 is the Network number and 192.168.1.152 is the host on that network.

- Binary Representation

The positional value method is the simplest form of converting binary from decimal value. IP address is 32 bit value which is divided into 4 octets. A binary octet contains 8 bits and the value of each bit can be determined by the position of bit value '1' in the octet.

MSB	8th	7th	6th	5th	4th	3rd	2nd	1st	LSB
	1	1	1	1	1	1	1	1	
Positional Value	128	64	32	16	8	4	2	1	

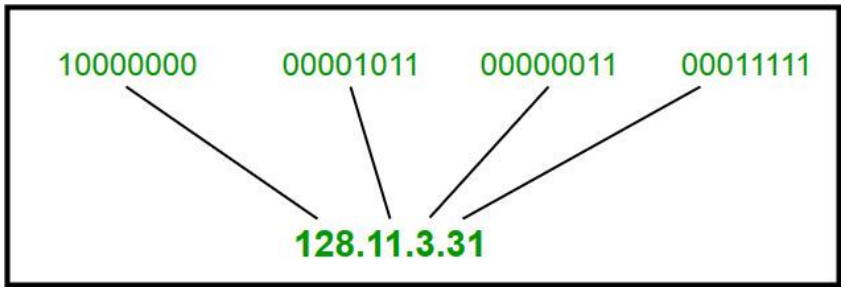
Positional value of bits is determined by 2 raised to power (position – 1), that is the value of a bit 1 at position 6 is 2^(6-1) that is 2^5 that is 32. The total value of the octet is determined by adding up the positional value of bits. The value of 11000000 is 128+64 = 192. Some examples are shown in the table below –

128	64	32	16	8	4	2	1	Value
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	4
0	0	0	0	0	1	0	1	5
0	0	0	0	0	1	1	0	6
0	0	0	0	0	1	1	1	7
0	0	0	0	1	0	0	0	8
0	0	0	0	1	0	0	1	9
0	0	0	0	1	0	1	0	10
0	0	0	1	0	0	0	0	16
0	0	1	0	0	0	0	0	32
0	1	0	0	0	0	0	0	64
0	1	1	0	0	1	0	0	100
0	1	1	1	1	1	1	1	127
1	0	0	0	0	0	0	0	128
1	0	1	0	1	0	0	0	168
1	1	0	0	0	0	0	0	192
1	1	1	1	1	1	1	1	255

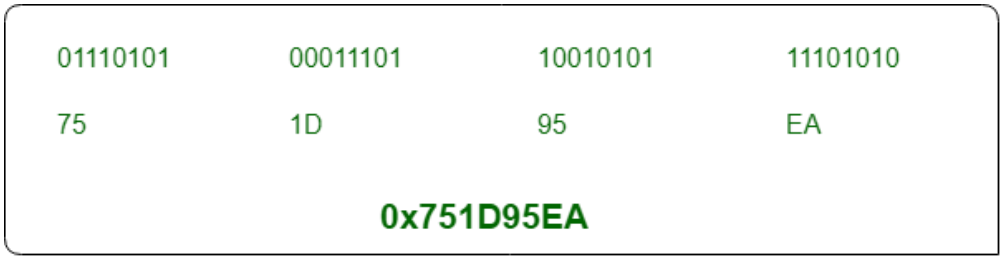
Introduction of Classful IP Addressing

IP address is an address having information about how to reach a specific host, especially outside the LAN. An IP address is a 32 bit unique address having an address space of 2^{32}. Generally, there are two notations in which IP address is written, dotted decimal notation and hexadecimal notation.

Dotted Decimal Notation:



Hexadecimal Notation:



Some points to be noted about dotted decimal notation:

1. The value of any segment (byte) is between 0 and 255 (both included).
2. There are no zeroes preceding the value in any segment (054 is wrong, 54 is correct).

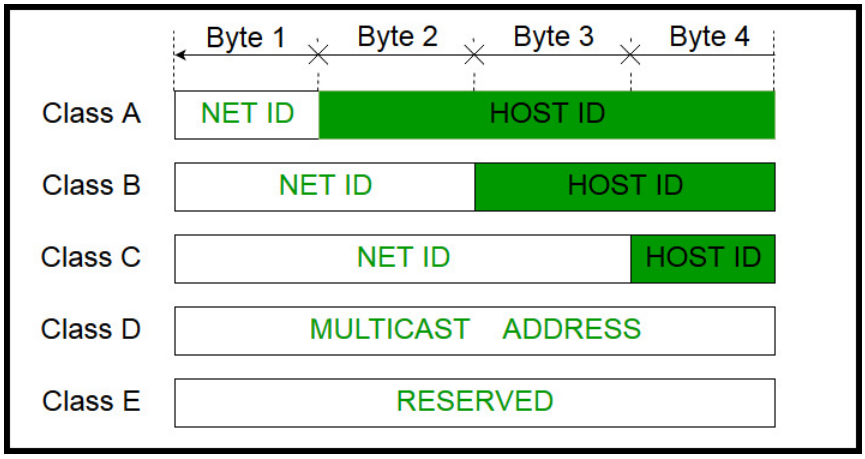
Classful Addressing
The 32 bit IP address is divided into five sub-classes. These are:

- Class A
- Class B
- Class C
- Class D
- Class E

Each of these classes has a valid range of IP addresses. Classes D and E are reserved for multicast and experimental purposes respectively. The order of bits in the first octet determine the classes of IP address.
IPv4 address is divided into two parts:

- Network ID
- Host ID

The class of IP address is used to determine the bits used for network ID and host ID and the number of total networks and hosts possible in that particular class. Each ISP or network administrator assigns IP address to each device that is connected to its network.



Note: IP addresses are globally managed by Internet Assigned Numbers Authority(IANA) and regional Internet registries(RIR).

Note: While finding the total number of host IP addresses, 2 IP addresses are not counted and are therefore, decreased from the total count because the first IP address of any network is the network number and whereas the last IP address is reserved for broadcast IP.

Class A:

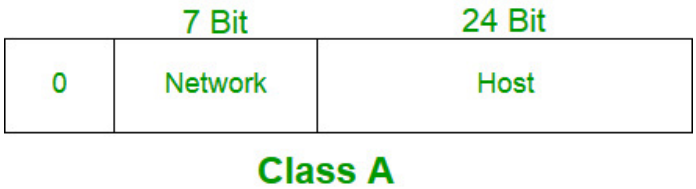
IP address belonging to class A are assigned to the networks that contain a large number of hosts.

- The network ID is 8 bits long.
- The host ID is 24 bits long.

The higher order bit of the first octet in class A is always set to 0. The remaining 7 bits in first octet are used to determine network ID. The 24 bits of host ID are used to determine the host in any network. The default subnet mask for class A is 255.x.x.x. Therefore, class A has a total of:

- 2^7-2= 126 network ID(Here 2 address is subtracted because 0.0.0.0 and 127.x.y.z are special address.)
- 2^24 – 2 = 16,777,214 host ID

IP addresses belonging to class A ranges from 1.x.x.x – 126.x.x.x



Class B:

IP address belonging to class B are assigned to the networks that ranges from medium-sized to large-sized networks.

- The network ID is 16 bits long.
- The host ID is 16 bits long.

The higher order bits of the first octet of IP addresses of class B are always set to 10. The remaining 14 bits are used to determine network ID. The 16 bits of host ID is used to determine the host in any network. The default sub-net mask for class B is 255.255.x.x. Class B has a total of:

- 2^14 = 16384 network address
- 2^16 – 2 = 65534 host address

IP addresses belonging to class B ranges from 128.0.x.x – 191.255.x.x.

Class B

Class C:

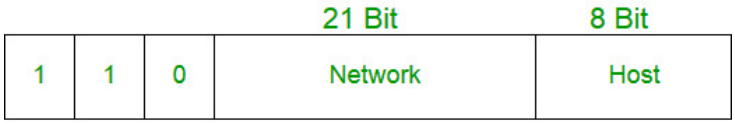
IP address belonging to class C are assigned to small-sized networks.

- The network ID is 24 bits long.
- The host ID is 8 bits long.

The higher order bits of the first octet of IP addresses of class C are always set to 110. The remaining 21 bits are used to determine network ID. The 8 bits of host ID is used to determine the host in any network. The default sub-net mask for class C is 255.255.255.x. Class C has a total of:

- 2^21 = 2097152 network address
- 2^8 – 2 = 254 host address

IP addresses belonging to class C ranges from 192.0.0.x – 223.255.255.x.

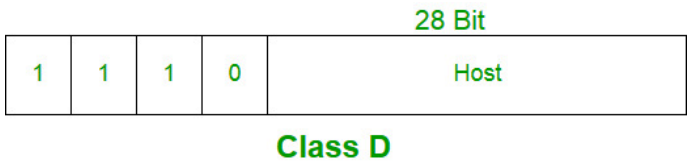


Class C

Class D:

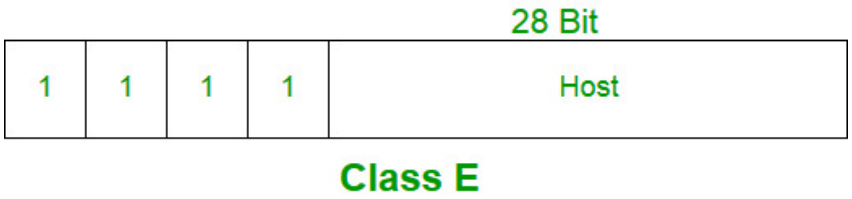
IP address belonging to class D are reserved for multi-casting. The higher order bits of the first octet of IP addresses belonging to class D are always set to 1110. The remaining bits are for the address that interested hosts recognize.

Class D does not posses any sub-net mask. IP addresses belonging to class D ranges from 224.0.0.0 – 239.255.255.255.



Class E:

IP addresses belonging to class E are reserved for experimental and research purposes. IP addresses of class E ranges from 240.0.0.0 – 255.255.255.254. This class doesn't have any sub-net mask. The higher order bits of first octet of class E are always set to 1111.



Range of special IP addresses:

169.254.0.0 – 169.254.0.16 : Link local addresses
127.0.0.0 – 127.0.0.8 : Loop-back addresses
0.0.0.0 – 0.0.0.8 : used to communicate within the current network.

Rules for assigning Host ID:

Host ID's are used to identify a host within a network. The host ID are assigned based on the following rules:

- Within any network, the host ID must be unique to that network.
- Host ID in which all bits are set to 0 cannot be assigned because this host ID is used to represent the network ID of the IP address.
- Host ID in which all bits are set to 1 cannot be assigned because this host ID is reserved as a broadcast address to send packets to all the hosts present on that particular network.

Rules for assigning Network ID:

Hosts that are located on the same physical network are identified by the network ID, as all host on the same physical network is assigned the same network ID. The network ID is assigned based on the following rules:

- The network ID cannot start with 127 because 127 belongs to class A address and is reserved for internal loop-back functions.
- All bits of network ID set to 1 are reserved for use as an IP broadcast address and therefore, cannot be used.
- All bits of network ID set to 0 are used to denote a specific host on the local network and are not routed and therefore, aren't used.

Classless Addressing

Classless Inter-Domain Routing (CIDR) is another name for classless addressing. This addressing type aids in the more efficient allocation of IP addresses. This technique assigns a block of IP addresses based on specified conditions when the user demands a specific amount of IP addresses. This block is known as a "CIDR block", and it contains the necessary number of IP addresses.

When allocating a block, classless addressing is concerned with the following three rules.

- **Rule 1** – The CIDR block's IP addresses must all be contiguous.
- **Rule 2** – The block size must be a power of two to be attractive. Furthermore, the block's size is equal to the number of IP addresses in the block.
- **Rule 3** – The block's first IP address must be divisible by the block size.

For example, assume the classless address is 192.168.1.35/27.

- The network component has a bit count of 27, whereas the host portion has a bit count of 5. (32-27)
- The binary representation of the address is: (00100011 . 11000000 . 10101000 . 00000001).
- (11000000.10101000.00000001.00100000) is the first IP address (assigns 0 to all host bits), that is, 192.168.1.32
- (11000000.10101000.00000001.00111111) is the most recent IP address (assigns 1 to all host bits), that is, 192.168.1.63
- The IP address range is 192.168.1.32 to 192.168.1.63.

Difference Between Classful and Classless Addressing

- Classful addressing is a technique of allocating IP addresses that divides them into five categories. Classless addressing is a technique of allocating IP addresses that is intended to replace classful addressing in order to reduce IP address depletion.
- The utility of classful and classless addressing is another distinction. Addressing without a class is more practical and helpful than addressing with a class.
- The network ID and host ID change based on the classes in classful addressing. In classless addressing, however, there is no distinction between network ID and host ID. As a result, another distinction between classful and classless addressing may be made.

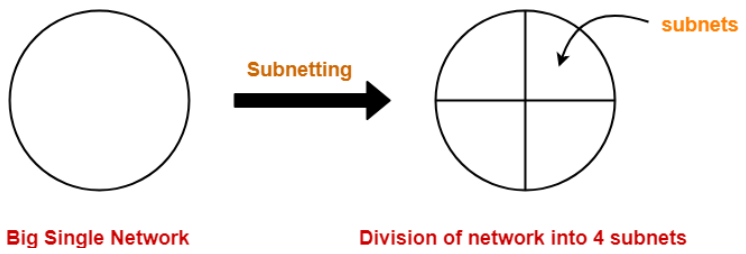
Subnetting in Networking-

In networking,

- The process of dividing a single network into multiple sub networks is called as **subnetting**.
- The sub networks so created are called as **subnets**.

Example-

Following diagram shows the subnetting of a big single network into 4 smaller subnets-



Advantages-

The two main advantages of subnetting a network are-

- It improves the security.
- The maintenance and administration of subnets is easy.

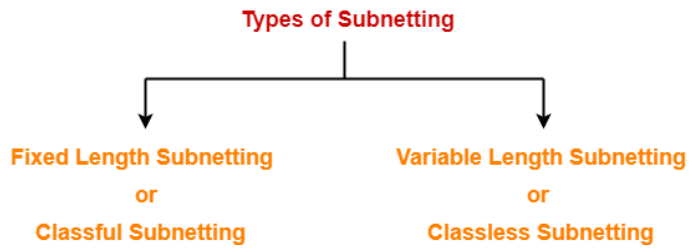
Subnet ID-

- Each subnet has its unique network address known as its **Subnet ID**.
- The subnet ID is created by borrowing some bits from the Host ID part of the IP Address.

- The number of bits borrowed depends on the number of subnets created.

Types of Subnetting-

Subnetting of a network may be carried out in the following two ways-



1. Fixed Length Subnetting
2. Variable Length Subnetting

1. Fixed Length Subnetting-

Fixed length subnetting also called as **classful subnetting** divides the network into subnets where-

- All the subnets are of same size.
- All the subnets have equal number of hosts.
- All the subnets have same subnet mask.

2. Variable Length Subnetting-

Variable length subnetting also called as **classless subnetting** divides the network into subnets where-

- All the subnets are not of same size.
- All the subnets do not have equal number of hosts.
- All the subnets do not have same subnet mask.

Subnetting Examples-

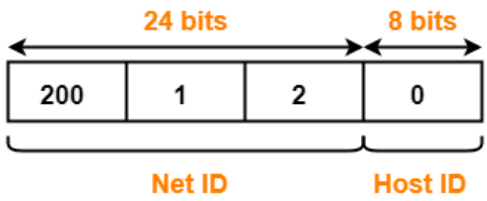
Now, we shall discuss some examples of subnetting a network-

Example-01:

Consider-

- We have a big single network having IP Address 200.1.2.0.
- We want to do subnetting and divide this network into 2 subnets.

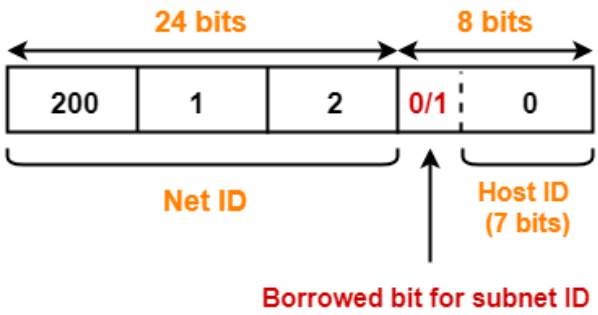
Clearly, the given network belongs to class C.



For creating two subnets and to represent their subnet IDs, we require 1 bit.

So,

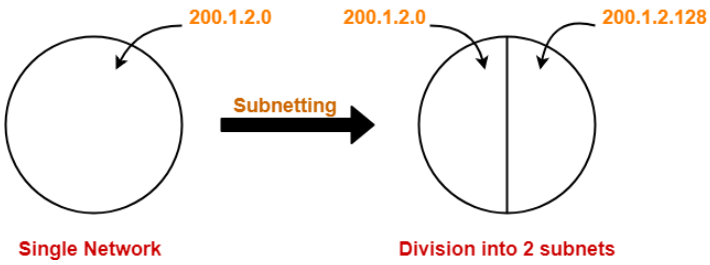
- We borrow one bit from the Host ID part.
- After borrowing one bit, Host ID part remains with only 7 bits.



- If borrowed bit = 0, then it represents the first subnet.
- If borrowed bit = 1, then it represents the second subnet.

IP Address of the two subnets are-

- 200.1.2.**0**0000000 = 200.1.2.0
- 200.1.2.**1**0000000 = 200.1.2.128



For 1st Subnet-

- IP Address of the subnet = 200.1.2.0
- Total number of IP Addresses = 2^7 = 128
- Total number of hosts that can be configured = 128 – 2 = 126
- Range of IP Addresses = [200.1.2.**0**0000000, 200.1.2.**0**1111111] = [200.1.2.0, 200.1.2.127]
- Direct Broadcast Address = 200.1.2.**0**1111111 = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255

For 2nd Subnet-

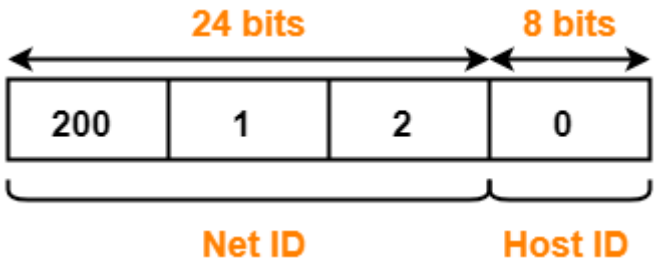
- IP Address of the subnet = 200.1.2.128
- Total number of IP Addresses = 2^7 = 128
- Total number of hosts that can be configured = 128 – 2 = 126
- Range of IP Addresses = [200.1.2.**1**0000000, 200.1.2.**1**1111111] = [200.1.2.128, 200.1.2.255]
- Direct Broadcast Address = 200.1.2.**1**1111111 = 200.1.2.255
- Limited Broadcast Address = 255.255.255.255

Example-02:

Consider-

- We have a big single network having IP Address 200.1.2.0.
- We want to do subnetting and divide this network into 4 subnets.

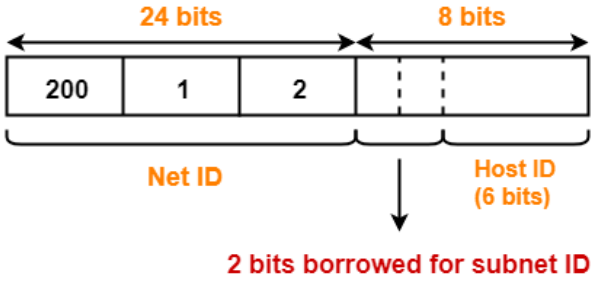
Clearly, the given network belongs to class C.



For creating four subnets and to represent their subnet IDs, we require 2 bits.

So,

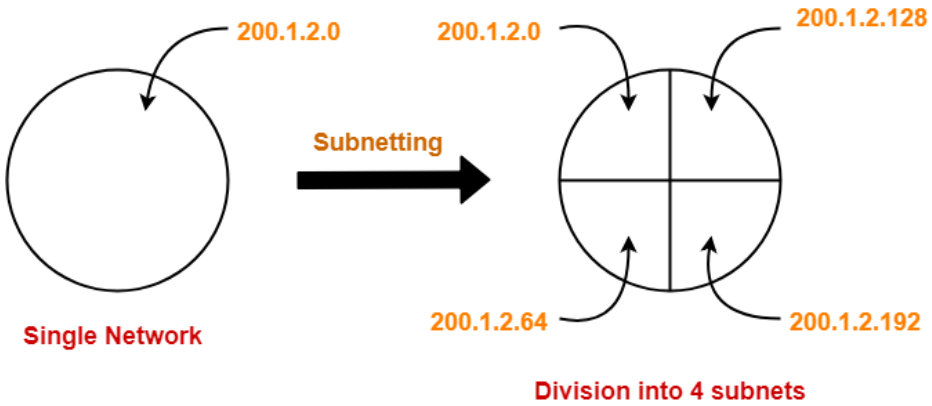
- We borrow two bits from the Host ID part.
- After borrowing two bits, Host ID part remains with only 6 bits.



- If borrowed bits = 00, then it represents the 1st subnet.
- If borrowed bits = 01, then it represents the 2nd subnet.
- If borrowed bits = 10, then it represents the 3rd subnet.
- If borrowed bits = 11, then it represents the 4th subnet.

IP Address of the four subnets are-

- 200.1.2.**00**000000 = 200.1.2.0
- 200.1.2.**01**000000 = 200.1.2.64
- 200.1.2.**10**000000 = 200.1.2.128
- 200.1.2.**11**000000 = 200.1.2.192



<u>For 1st Subnet-</u>

- IP Address of the subnet = 200.1.2.0
- Total number of IP Addresses = 2^6 = 64
- Total number of hosts that can be configured = 64 – 2 = 62
- Range of IP Addresses = [200.1.2.**00**000000, 200.1.2.**00**111111] = [200.1.2.0, 200.1.2.63]
- Direct Broadcast Address = 200.1.2.**00**111111 = 200.1.2.63

- Limited Broadcast Address = 255.255.255.255

For 2nd Subnet-

- IP Address of the subnet = 200.1.2.64
- Total number of IP Addresses = 2^6 = 64
- Total number of hosts that can be configured = 64 – 2 = 62
- Range of IP Addresses = [200.1.2.**01**000000, 200.1.2.**01**111111] = [200.1.2.64, 200.1.2.127]
- Direct Broadcast Address = 200.1.2.**01**111111 = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255

For 3rd Subnet-

- IP Address of the subnet = 200.1.2.128
- Total number of IP Addresses = 2^6 = 64
- Total number of hosts that can be configured = 64 – 2 = 62
- Range of IP Addresses = [200.1.2.**10**000000, 200.1.2.**10**111111] = [200.1.2.128, 200.1.2.191]
- Direct Broadcast Address = 200.1.2.**10**111111 = 200.1.2.191
- Limited Broadcast Address = 255.255.255.255

For 4th Subnet-

- IP Address of the subnet = 200.1.2.192
- Total number of IP Addresses = 2^6 = 64
- Total number of hosts that can be configured = 64 – 2 = 62
- Range of IP Addresses = [200.1.2.**11**000000, 200.1.2.**11**111111] = [200.1.2.192, 200.1.2.255]
- Direct Broadcast Address = 200.1.2.**11**111111 = 200.1.2.255
- Limited Broadcast Address = 255.255.255.255

Example-03:

Consider-

- We have a big single network having IP Address 200.1.2.0.
- We want to do subnetting and divide this network into 3 subnets.

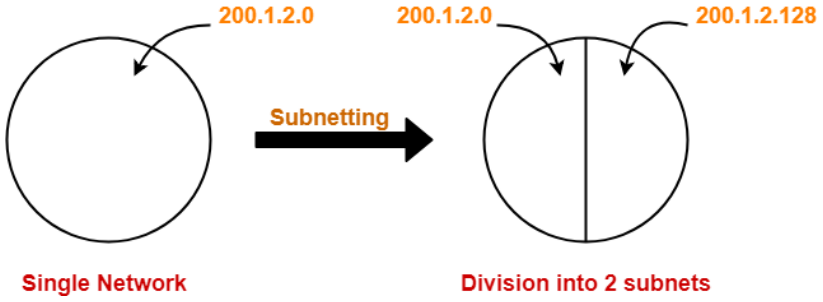
Here, the subnetting will be performed in two steps-

1. Dividing the given network into 2 subnets
2. Dividing one of the subnets further into 2 subnets

Step-01: Dividing Given Network into 2 Subnets-

The subnetting will be performed exactly in the same way as performed in Example-01.

After subnetting, we have-

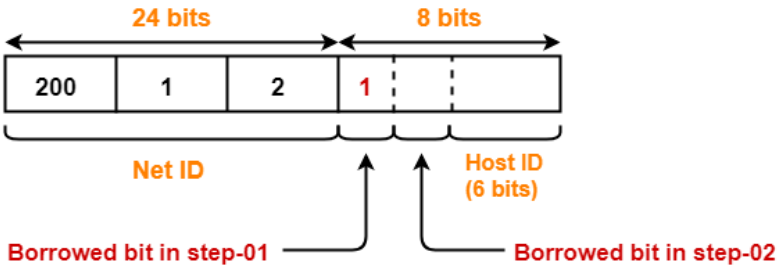


Step-02: Dividing One Subnet into 2 Subnets-

- We perform the subnetting of one of the subnets further into 2 subnets.
- Consider we want to do subnetting of the 2nd subnet having IP Address 200.1.2.128.

For creating two subnets and to represent their subnet IDs, we require 1 bit. So,

- We borrow one more bit from the Host ID part.
- After borrowing one bit, Host ID part remains with only 6 bits.

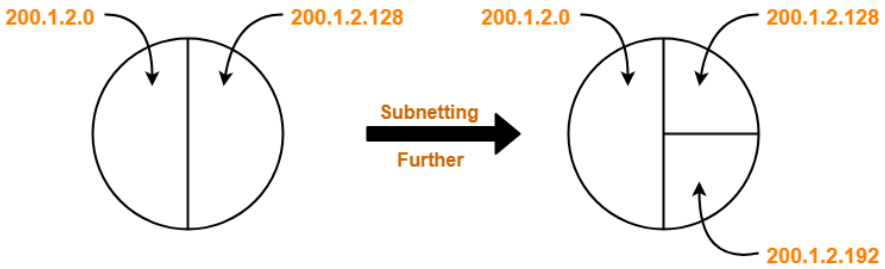


- If 2nd borrowed bit = 0, then it represents one subnet.
- If 2nd borrowed bit = 1, then it represents the other subnet.

IP Address of the two subnets are-

- 200.1.2.**10**000000 = 200.1.2.128

- 200.1.2.**11**000000 = 200.1.2.192



Finally, the given single network is divided into 3 subnets having IP Address-

- 200.1.2.0
- 200.1.2.128
- 200.1.2.192

<u>For 1st Subnet-</u>

- IP Address of the subnet = 200.1.2.0
- Total number of IP Addresses = 2^7 = 128
- Total number of hosts that can be configured = 128 – 2 = 126
- Range of IP Addresses = [200.1.2.**0**0000000, 200.1.2.**0**1111111] = [200.1.2.0, 200.1.2.127]
- Direct Broadcast Address = 200.1.2.**0**1111111 = 200.1.2.127
- Limited Broadcast Address = 255.255.255.255

<u>For 2nd Subnet-</u>

- IP Address of the subnet = 200.1.2.128
- Total number of IP Addresses = 2^6 = 64
- Total number of hosts that can be configured = 64 – 2 = 62
- Range of IP Addresses = [200.1.2.**10**000000, 200.1.2.**10**111111] = [200.1.2.128, 200.1.2.191]
- Direct Broadcast Address = 200.1.2.**10**111111 = 200.1.2.191
- Limited Broadcast Address = 255.255.255.255

<u>For 3rd Subnet-</u>

- IP Address of the subnet = 200.1.2.192
- Total number of IP Addresses = 2^6 = 64
- Total number of hosts that can be configured = 64 – 2 = 62

- Range of IP Addresses = [200.1.2.**11**000000, 200.1.2.**11**111111] = [200.1.2.192, 200.1.2.255]
- Direct Broadcast Address = 200.1.2.**11**111111 = 200.1.2.255
- Limited Broadcast Address = 255.255.255.255

****

Unit- 7

The Application Layer

Content:

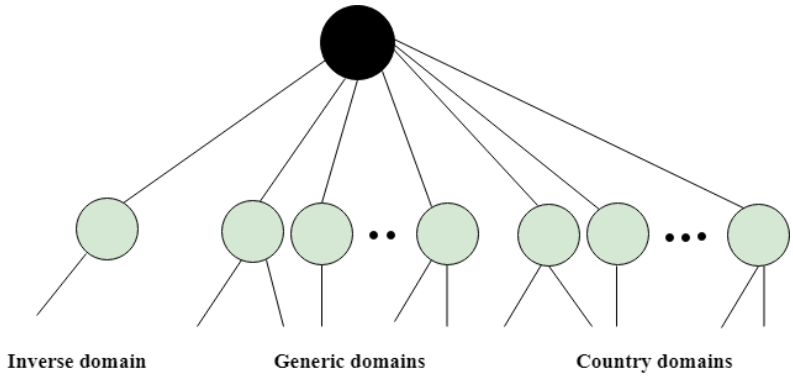
1. Domain Name System
2. E-mail Services
3. SMTP/POP protocols
4. File transfer protocols
5. WWW Service and HTTP
6. SNMP Protocol

1. Domain Name System (DNS):

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.

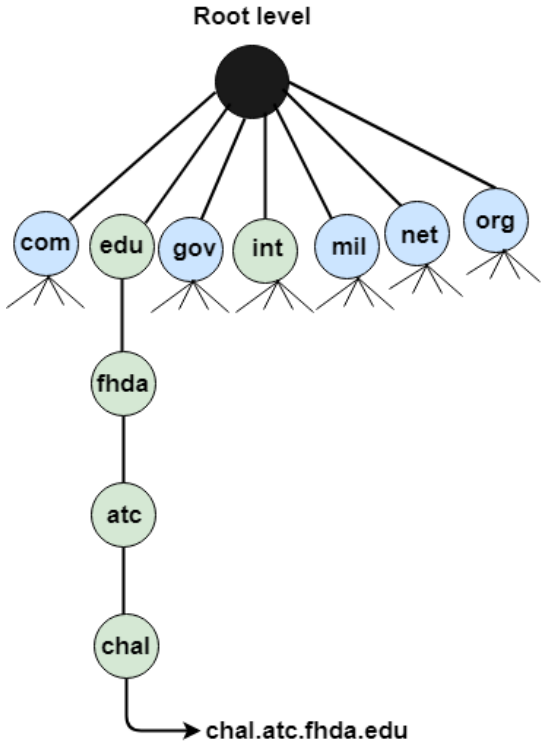


Generic Domains

- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers

int	International Organizations
mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations



Country Domain

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookup while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as DNS resolver sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

2. E-mail Services:

Email is a service which allows us to send the message in electronic mode over the internet. It offers an efficient, inexpensive and real time mean of distributing information among people.

E-Mail Address

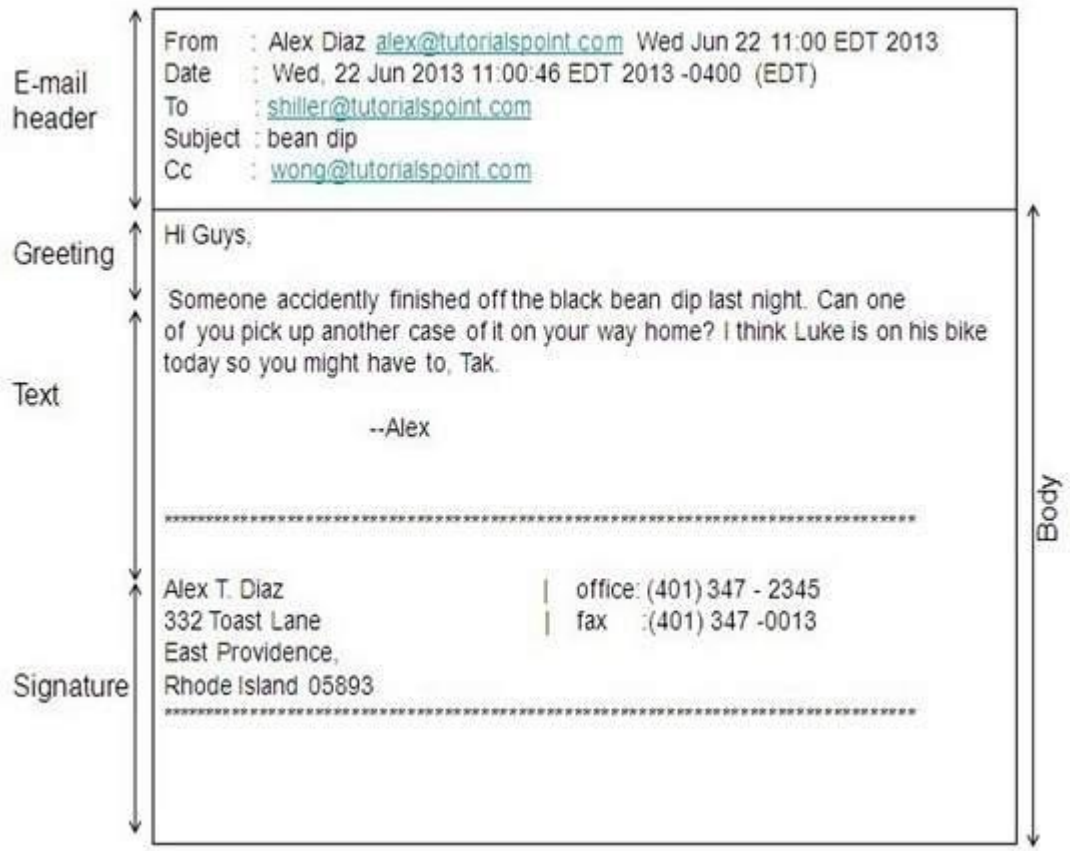
Each user of email is assigned a unique name for his email account. This name is known as E-mail address. Different users can send and receive messages according to the e-mail address.

E-mail is generally of the form username@domainname. For example, webmaster@tutorialspoint.com is an e-mail address where webmaster is username and tutorialspoint.com is domain name.

- The username and the domain name are separated by **@ (at)** symbol.
- E-mail addresses are not case sensitive.
- Spaces are not allowed in e-mail address.

E-mail Message Components:

E-mail message comprises of different components: E-mail Header, Greeting, Text, and Signature. These components are described in the following diagram:



E-mail Header:

The first five lines of an E-mail message is called E-mail header. The header part comprises of following fields:

- From
- Date
- To
- Subject
- CC
- BCC

- From-
  The **From** field indicates the sender's address i.e. who sent the e-mail.
- Date-
  The **Date** field indicates the date when the e-mail was sent.
- To-
  The **To** field indicates the recipient's address i.e. to whom the e-mail is sent.
- Subject-
  The **Subject** field indicates the purpose of e-mail. It should be precise and to the point.
- CC-
  CC stands for Carbon copy. It includes those recipient addresses whom we want to keep informed but not exactly the intended recipient.
- BCC-
  BCC stands for Black Carbon Copy. It is used when we do not want one or more of the recipients to know that someone else was copied on the message.
- Greeting-
  Greeting is the opening of the actual message. Eg. Hi Sir or Hi Guys etc.
- Text-
  It represents the actual content of the message.
- Signature-
  This is the final part of an e-mail message. It includes Name of Sender, Address, and Contact Number.

Advantages:

E-mail has prooved to be powerful and reliable medium of commmunication. Here are the benefits of **E-mail:**

- Reliable
  Many of the mail systems notify the sender if e-mail message was undeliverable.
- Convenience
  There is no requirement of stationary and stamps. One does not have to go to post office. But all these things are not required for sending or receiving an mail.
- Speed
  E-mail is very fast. However, the speed also depends upon the underlying network.
- Inexpensive
  The cost of sending e-mail is very low.
- Printable
  It is easy to obtain a hardcopy of an e-mail. Also an electronic copy of an e-mail can also be saved for records.
- Global

E-mail can be sent and received by a person sitting across the globe.

- Generality

  It is also possible to send graphics, programs and sounds with an e-mail.

Disadvantages:

Apart from several benefits of E-mail, there also exists some disadvantages as discussed below:

- Forgery

  E-mail doesn't prevent from forgery, that is, someone impersonating the sender, since sender is usually not authenticated in any way.

- Overload

  Convenience of E-mail may result in a flood of mail.

- Misdirection

  It is possible that you may send e-mail to an unintended recipient.

- Junk

  Junk emails are undesirable and inappropriate emails. Junk emails are sometimes referred to as spam.

- No Response

  It may be frustrating when the recipient does not read the e-mail and respond on a regular basis.

  E-mail System

  E-mail system comprises of the following three components:

- **Mailer:**

  It is also called **mail program, mail application** or **mail client.** It allows us to manage, read and compose e-mail.

- **Mail Server:**

  The function of mail server is to receive, store and deliver the email. It is must for mail servers to be Running all the time because if it crashes or is down, email can be lost.

- **Mailboxes:**

  Mailbox is generally a folder that contains emails and information about them.

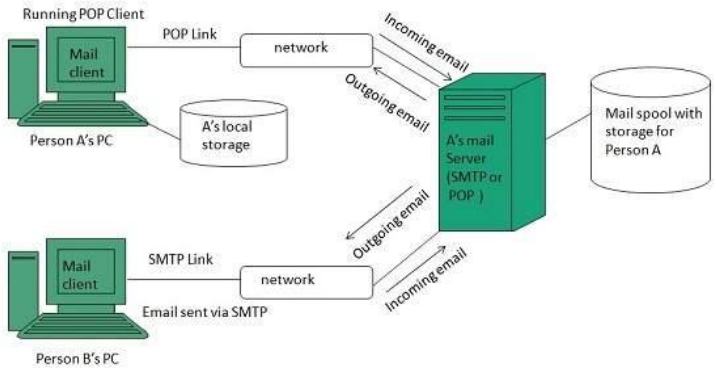
Working of E-mail:

Email working follows the client server approach. In this client is the mailer i.e. the mail application or mail program and server is a device that manages emails.

Following example will take you through the basic steps involved in sending and receiving emails and will give you a better understanding of working of email system:

- Suppose person A wants to send an email message to person B.
- Person A composes the messages using a mailer program i.e. mail client and then select Send option.
- The message is routed to **Simple Mail Transfer Protocol** to person B's mail server.
- The mail server stores the email message on disk in an area designated for person B.
- The disk space area on mail server is called mail spool.
- Now, suppose person B is running a POP client and knows how to communicate with B's mail server.
- It will periodically poll the POP server to check if any new email has arrived for B.As in this case, person B has sent an email for person B, so email is forwarded over the network to B's PC. This is message is now stored on person B's PC.

The following diagram gives pictorial representation of the steps discussed above:



Composing and Sending Email:

Before sending an email, we need to compose a message. When we are composing an email message, we specify the following things:

- Sender's address in To field
- Cc (if required)
- Bcc (if required)
- Subject of email message
- Text
- Signature

You should specify the correct email address; otherwise it will send an error back to the sender.

Once you have specified all the above parameters, It's time to send the email. The mailer program provides a Send button to send email, when you click Send, it is sent to the mail server and a message **mail sent successfully** is shown at the above.

Reading Email:

Every email program offers you an interface to access email messages. Like in Gmail, emails are stored under different tabs such as primary, social, and promotion. When you click one of tab, it displays a list of emails under that tab.

In order to read an email, you just have to click on that email. Once you click a particular email, it gets opened. The opened email may have some file attached with it. The attachments are shown at the bottom of the opened email with an option called **download attachment.**

Replying Email:

After reading an email, you may have to reply that email. To reply an email, click **Reply** option shown at the bottom of the opened email.

Once you click on Reply, it will automatically copy the sender's address in to the To field. Below the To field, there is a text box where you can type the message.

Once you are done with entering message, click Send button. It's that easy. Your email is sent.

Forwarding Email:

It is also possible to send a copy of the message that you have received along with your own comments if you want. This can be done using **forward** button available in mail client software.

The difference between replying and forwarding an email is that when you reply a message to a person who has send the mail but while forwarding you can send it to anyone.

When you receive a forwarded message, the message is marked with a > character in front of each line and **Subject:** field is prefixed with **Fw.**

Deleting Email:

If you don't want to keep email into your inbox, you can delete it by simply selecting the message from the message list and clicking **delete** or pressing the appropriate command.

Some mail clients offers the deleted mails to be stored in a folder called deleted items or trash from where you can recover a deleted email.

Email Features:

Now a day, the mail client comes with enhanced features such as attachment, address book, and MIME support. Here in this chapter we will discuss all of these features which will give you a better understanding of added feature of a mail client program.

Attachment:

Ability to attach file(s) along with the message is one of the most useful features of email. The attachment may be a word document, PowerPoint presentation, audio/video files, or images.

- In order to attach file(s) to an email, click the attach button. As a result, a dialog box appears asking for specifying the name and location of the file you want to attach.
- Once you have selected the appropriate file, it is attached to the mail.
- Usually a paper clip icon appears in the email which indicates that it has an attachment.
- When adding an attachment it is better to compress the attached files so as to reduce the file size and save transmission time as sending and downloading large files consumes a lot of space and time.

Address Book:

Address book feature of a mail program allows the users to store information about the people whom they communicate regularly by sending emails. Here are some of the key features of an Address book:

- Address book includes the nick names, email addresses, phone number etc. of the people.
- Using address book allows us not to memorize email of address of a person, you just have to select recipient name from the list.
- When you select a particular name from the list, the corresponding email address link automatically get inserted in to the To: field.
- Address book also allows creating a group so that you can send a email to very member of the group at once instead of giving each person email address one by one.

MIME Types:

MIME is acronym of Multipurpose Internet Mail Extensions. MIME compliant mailer allows us to send files other than simple text i.e. It allows us to send audio, video, images, document, and pdf files as an attachment to an email.

Suppose if you want to send a word processor document that has a group of tabular columns with complex formatting.

If we transfer the file as text, all the formatting may be lost. MIME compliant mailer takes care of messy details and the message arrives as desired.

The following table describes commonly used MIME Types:

1.	Type	Subtype	Description	File extension(s)
2.	Application	postscript tex troff	Printable postscript document TEX document Printable troff document	.eps, .ps .tex .t, .tr, .roff
3.	Audio	aiff au midi real audio	Apple sound Sun Microsystems sound Musical Instrument Digital Interface Progressive Network sound	.aif, .aiff,.aifc .au, .snd .midi, .mid .ra, .ram
4.	image	gif jpeg png triff	Graphics Interchange Format Joint Photographic Experts Group Portable Network Graphics Tagged Image Modeling Language	.gif .jpeg, .jpg, .jpe .png .tiff, .tif
5.	Model	vrml	Virual reality Modelling Language	.wrl
6.	Text plain sgml	html	Hyper Text Markup Language Unformatted text Standard Generalized Markup language	.html, .htm .txt .sgml
7.	Video	avi mpeg quicktime sgi-movie	Microsoft Audio Video Interleaved Moving Pictures Expert Group Apple QuickTime movie silicon graphic movie	.avi .mpeg, .mpg .qt, .mov .movie

The Email service providers:

There are several email service providers available in the market with their enabled features such as sending, receiving, drafting, storing an email and much more.

The following table shows the popular email service providers:

S.N.	Service and Description
1.	**Gmail** Gmail is an email service that allows users to collect all the messages. It also offers approx 7 GB of free storage.
2.	**Hotmail** Hotmail offers free email and practically unlimited storage accessible on web.
3.	**Yahoo Mail** Yahoo Mail offers unlimited storage, SMS texting, social networking and instant messaging to boot.
4.	**iCloud Mail** iCloud Mail offers ample storage, IMAP access, and an elegantly functional web application.
5.	**ATM Mail** ATM Mail is a free email service with good spam protection.
6.	**Mail.com** and **GMX Mail** Mail.com and GMX Mail offers reliable mail service with unlimited online storage.
7.	**Shortmail** Shortmail offers easy and fast email service but with limited 500 characters per message.
8.	**Inbox.com** Inbox.com offers 5 GB of free online storage. IMAP is not supported by Inbox.com
9.	**Facebook Messages** Facebook Messages includes the message conversation.
10.	**My Way Mail** My Way Mail offers clean and fast free email service but lacks in secure messaging.

3. SMTP/POP Protocols:

E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Here in this tutorial, we will discuss various protocols such as **SMTP, POP,** and **IMAP.**

SMPTP:

SMTP stands for **Simple Mail Transfer Protocol**. It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

Key Points:

- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.
- It handles exchange of messages between e-mail servers over TCP/IP network.
- Apart from transferring e-mail, SMPT also provides notification regarding incoming mail.
- When you send e-mail, your e-mail client sends it to your e-mail server which further contacts the recipient mail server using SMTP client.
- These SMTP commands specify the sender's and receiver's e-mail address, along with the message to be send.
- The exchange of commands between servers is carried out without intervention of any user.
- In case, message cannot be delivered, an error report is sent to the sender which makes SMTP a reliable protocol.

SMTP Commands:

The following table describes some of the SMTP commands:

S.N.	Command Description
1	**HELLO** This command initiates the SMTP conversation.
2	**EHELLO** This is an alternative command to initiate the conversation. ESMTP indicates that the sender server wants to use extended SMTP protocol.
3	**MAIL FROM** This indicates the sender's address.
4	**RCPT TO** It identifies the recipient of the mail. In order to deliver similar message to multiple users this command can be repeated multiple times.
5	**SIZE** This command let the server know the size of attached message in bytes.
6	**DATA** The **DATA** command signifies that a stream of data will follow. Here stream of data refers to the body of the message.
7	**QUIT** This commands is used to terminate the SMTP connection.
8	**VERFY** This command is used by the receiving server in order to verify whether the given username is valid or not.
9	**EXPN** It is same as VRFY, except it will list all the users name when it used with a distribution list.

IMAP:

IMAP stands for **Internet Message Access Protocol.** It was first proposed in 1986. There exist five versions of IMAP as follows:

1. Original IMAP
2. IMAP2
3. IMAP3
4. IMAP2bis
5. IMAP4

Key Points:

- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.
- It enables us to take any action such as downloading, delete the mail without reading the mail.It enables us to create, manipulate and delete remote message folders called mail boxes.
- IMAP enables the users to search the e-mails.
- It allows concurrent access to multiple mailboxes on multiple mail servers.

IMAP Commands

The following table describes some of the IMAP commands:

S.N.	Command Description
1	**IMAP_LOGIN** This command opens the connection.
2	**CAPABILITY** This command requests for listing the capabilities that the server supports.
3	**NOOP** This command is used as a periodic poll for new messages or message status updates during a period of inactivity.
4	**SELECT** This command helps to select a mailbox to access the messages.
5	**EXAMINE** It is same as SELECT command except no change to the mailbox is permitted.
6	**CREATE** It is used to create mailbox with a specified name.
7	**DELETE** It is used to permanently delete a mailbox with a given name.
8	**RENAME** It is used to change the name of a mailbox.
9	**LOGOUT** This command informs the server that client is done with the session. The server must send BYE untagged response before the OK response and then close the network connection.

POP:

POP stands for Post Office Protocol. It is generally used to support a single client. There are several versions of POP but the POP 3 is the current standard.

Key Points

- POP is an application layer internet standard protocol.
- Since POP supports offline access to the messages, thus requires less internet usage time.
- POP does not allow search facility.
- In order to access the messaged, it is necessary to download them.
- It allows only one mailbox to be created on server.
- It is not suitable for accessing non mail data.
- POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.

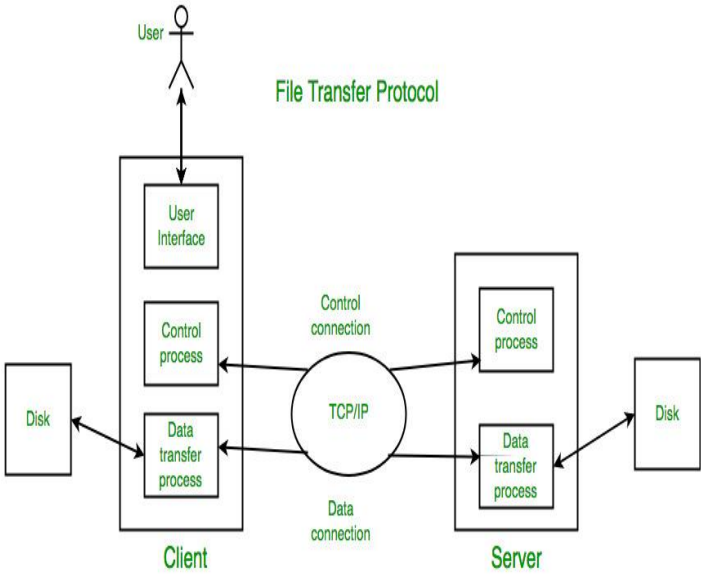
POP Commands

The following table describes some of the POP commands:

S.N.	Command Description
1	**LOGIN** This command opens the connection.
2	**STAT** It is used to display number of messages currently in the mailbox.
3	**LIST** It is used to get the summary of messages where each message summary is shown.
4	**RETR** This command helps to select a mailbox to access the messages.
5	**DELE** It is used to delete a message.
6	**RSET** It is used to reset the session to its initial state.
7	**QUIT** It is used to log off the session.

4. File transfer protocol (FTP):

File Transfer Protocol (FTP) is an application layer protocol which moves files between local and remote file systems. It runs on the top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.



What is control connection?

For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of control connection. The control connection is initiated on port number 21.

What is data connection?

For sending the actual file, FTP makes use of data connection. A data connection is initiated on port number 20.

FTP sends the control information out-of-band as it uses a separate control connection. Some protocols send their request and response header lines and the data in the same TCP connection. For this reason, they are said to send their control information in-band. HTTP and SMTP are such examples.

FTP Session:

When a FTP session is started between a client and a server, the client initiates a control TCP connection with the server side. The client sends control information over this. When the server receives this, it initiates a data connection to the client side. Only one file can be sent over one data connection. But the control connection remains active throughout the user session. As we know HTTP is stateless i.e. it does not have to keep track of any user state. But FTP needs to maintain a state about its user throughout the session.

Data Structures: FTP allows three types of data structures :

1. **File Structure –** In file-structure there is no internal structure and the file is considered to be a continuous sequence of data bytes.
2. **Record Structure –** In record-structure the file is made up of sequential records.
3. **Page Structure –** In page-structure the file is made up of independent indexed pages.

5. The WWW & HTTP:

Overview

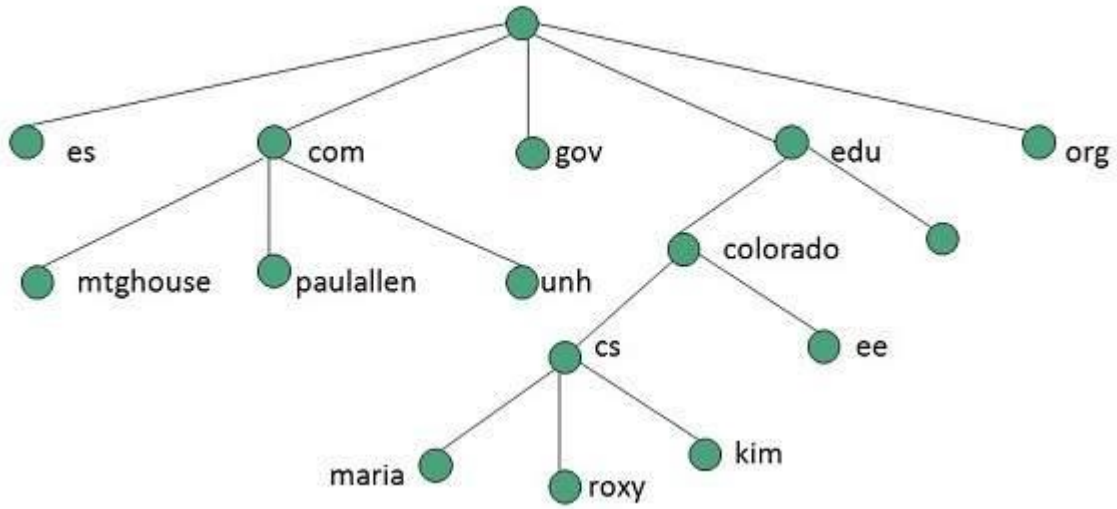
WWW stands for **World Wide Web.** A technical definition of the World Wide Web is : all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).

A broader definition comes from the organization that Web inventor **Tim Berners-Lee** helped found, the **World Wide Web Consortium (W3C).**

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

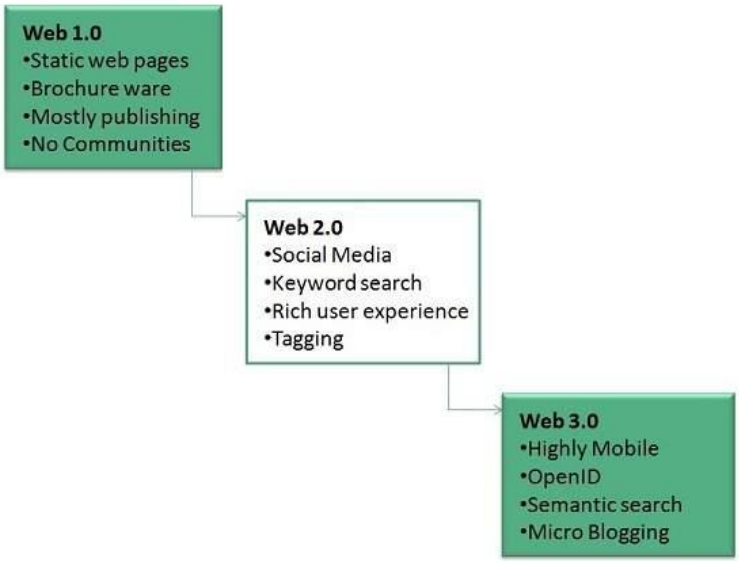
Internet and **Web** is not the same thing: Web uses internet to pass over the information.



Evolution

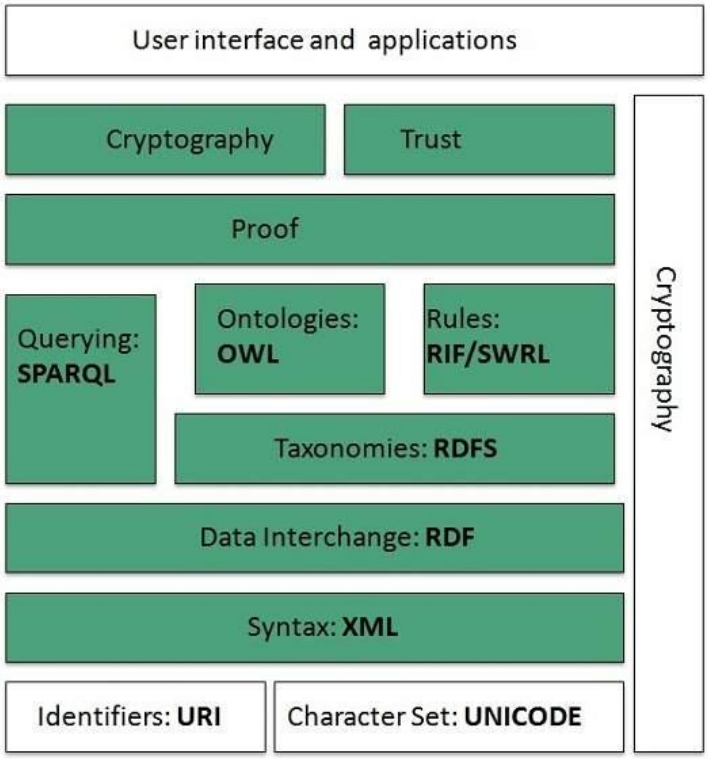
World Wide Web was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva.** World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN.** Eventually it became **World Wide Web.**

The following diagram briefly defines evolution of World Wide Web:



WWW Architecture:

WWW architecture is divided into several layers as shown in the following diagram:



Identifiers and Character Set:

Uniform Resource Identifier (URI) is used to uniquely identify resources on the web and **UNICODE** makes it possible to built web pages that can be read and write in human languages.

Syntax:

XML (Extensible Markup Language) helps to define common syntax in semantic web.

Data Interchange

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

Taxonomies

RDF Schema (RDFS) allows more standardized description of **taxonomies** and other **ontological** constructs.

Ontologies:

Web Ontology Language (OWL) offers more constructs over RDFS. It comes in following three versions:

- OWL Lite for taxonomies and simple constraints.
- OWL DL for full description logic support.
- OWL for more syntactic freedom of RDF

Rules

RIF and **SWRL** offers rules beyond the constructs that are available from **RDFs** and **OWL.** Simple Protocol and **RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

Proof

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

Cryptography

Cryptography means such as digital signature for verification of the origin of sources is used.

User Interface and Applications

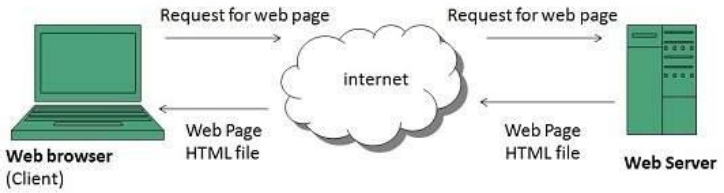
On the top of layer **User interface and Applications** layer is built for user interaction.

WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

1. User enters the URL (say, **http://www.tutorialspoint.com**) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.

5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.



Future:

There had been a rapid development in field of web. It has its impact in almost every area such as education, research, technology, commerce, marketing etc. So the future of web is almost unpredictable.

Apart from huge development in field of WWW, there are also some technical issues that W3 consortium has to cope up with.

User Interface

Work on higher quality presentation of 3-D information is under deveopment. The W3 Consortium is also looking forward to enhance the web to full fill requirements of global communities which would include all regional languages and writing systems.

Technology

Work on privacy and security is under way. This would include hiding information, accounting, access control, integrity and risk management.

Architecture

There has been huge growth in field of web which may lead to overload the internet and degrade its performance. Hence more better protocol are required to be developed.

Web Page:

web page is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.

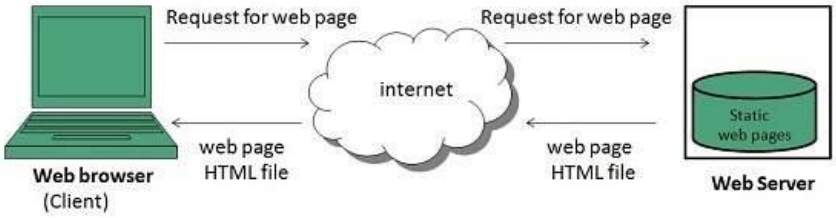
A web page can cotain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages.

Collection of linked web pages on a web server is known as **website.** There is unique **Uniform Resource Locator (URL)** is associated with each web page.

Static Web page

Static web pages are also known as flat or stationary web page. They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can't do any modification or interact with the information.

Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.



Dynamic Web page:

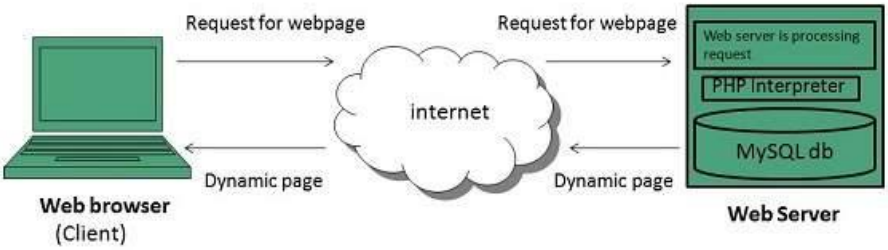
Dynamic web page shows different information at different point of time. It is possible to change a portaion of a web page without loading the entire web page. It has been made possible using **Ajax** technology.

Server-side dynamic web page

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

Client-side dynamic web page

It is processed using client side scripting such as JavaScript. And then passed in to **Document Object Model (DOM).**



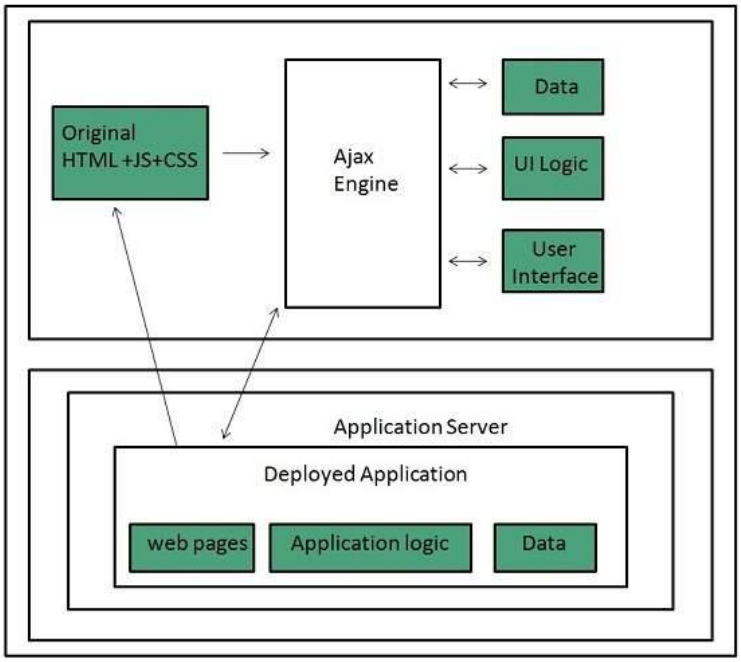
Scripting Languages:

Scripting languages are like programming languages that allow us to write programs in form of script. These scripts are interpreted not compiled and executed line by line. Scripting language is used to create dynamic web pages.

Client-side Scripting

Client-side scripting refers to the programs that are executed on client-side. Client-side scripts contains the instruction for the browser to be executed in response to certain user's action.

Client-side scripting programs can be embedded into HTML files or also can be kept as separate files.

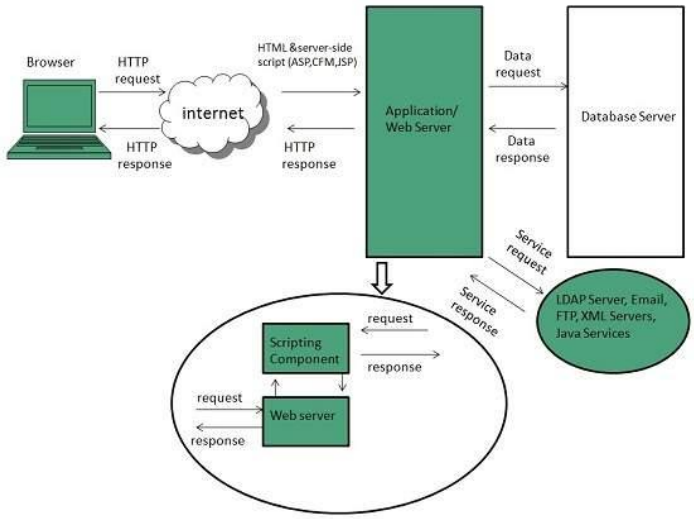


Following table describes commonly used Client-Side scripting languages:

S.N.	Scripting Language Description
1.	**JavaScript** It is a prototype based scripting language. It inherits its naming conventions from java. All java script files are stored in file having **.js** extension.
2.	**ActionScript**It is an object oriented programming language used for the development of websites and software targeting Adobe flash player.
3.	**Dart** It is an open source web programming language developed by Google. It relies on source-to-source compiler to JavaScript.
4.	**VBScript** It is an open source web programming language developed by Microsoft. It is superset of JavaScript and adds optional static typing class-based object oriented programming.

Server-side Scripting:

Sever-side scripting acts as an interface for the client and also limit the user access the resources on web server. It can also collects the user's characteristics in order to customize response.



Following table describes commonly used Server-Side scripting languages:

S.N.	Scripting Language Description
1.	**ASP** Active Server Pages (ASP)is server-side script engine to create dynamic web pages. It supports **Component Object Model (COM)** which enables ASP web sites to access functionality of libraries such as DLL.
2.	**ActiveVFP** It is similar to PHP and also used for creating dynamic web pages. It uses native **Visual Foxpro** language and database.
3.	**ASP.net** It is used to develop dynamic websites, web applications, and web services.
4.	**Java** Java Server Pages are used for creating dynamic web applications. The Java code is compiled into byte code and run by **Java Virtual Machine (JVM).**
5.	**Python** It supports multiple programming paradigms such as object-oriented, and functional programming. It can also be used as non-scripting language using third party tools such as **Py2exe** or **Pyinstaller.**
6.	**WebDNA** It is also a server-side scripting language with an embedded database system.

6. The SNMP Protocol:

Simple Network Management Protocol (SNMP):

If an organization has 1000 of devices then to check all devices, one by one every day, are working properly or not is a hectic task. To ease these up, Simple Network Management Protocol (SNMP) is used.

Simple Network Management Protocol (SNMP) –

SNMP is an application layer protocol which uses UDP port number 161/162.SNMP is

used to monitor the network, detect network faults and sometimes even used to configure remote devices.

SNMP components –

There are 3 components of SNMP:

1. **SNMP Manager –**
   It is a centralised system used to monitor network.It is also known as Network Management Station (NMS)
2. **SNMP agent –**
   It is a software management software module installed on a managed device. Managed devices can be network devices like PC, router, switches, servers etc.
3. **Management Information Base –**
   MIB consists of information of resources that are to be managed. These information is organised hierarchically. It consists of objects instances which are essentially variables.

Unit- 7

The Application Layer

Content:

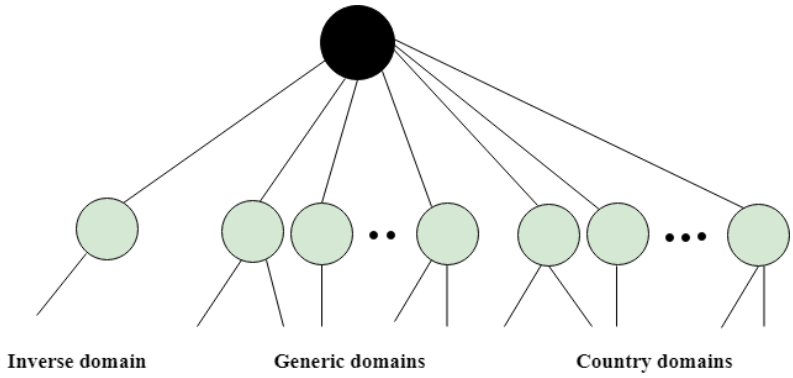
1. Domain Name System
2. E-mail Services
3. SMTP/POP protocols
4. File transfer protocols
5. WWW Service and HTTP
6. SNMP Protocol

1. Domain Name System (DNS):

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.

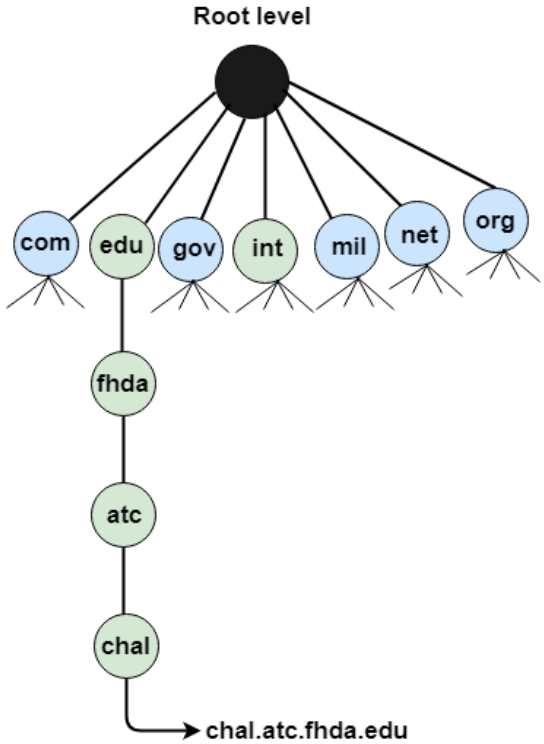


Generic Domains

- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers

int	International Organizations
mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations



Country Domain

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookup while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as DNS resolver sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

2. E-mail Services:

Email is a service which allows us to send the message in electronic mode over the internet. It offers an efficient, inexpensive and real time mean of distributing information among people.

E-Mail Address

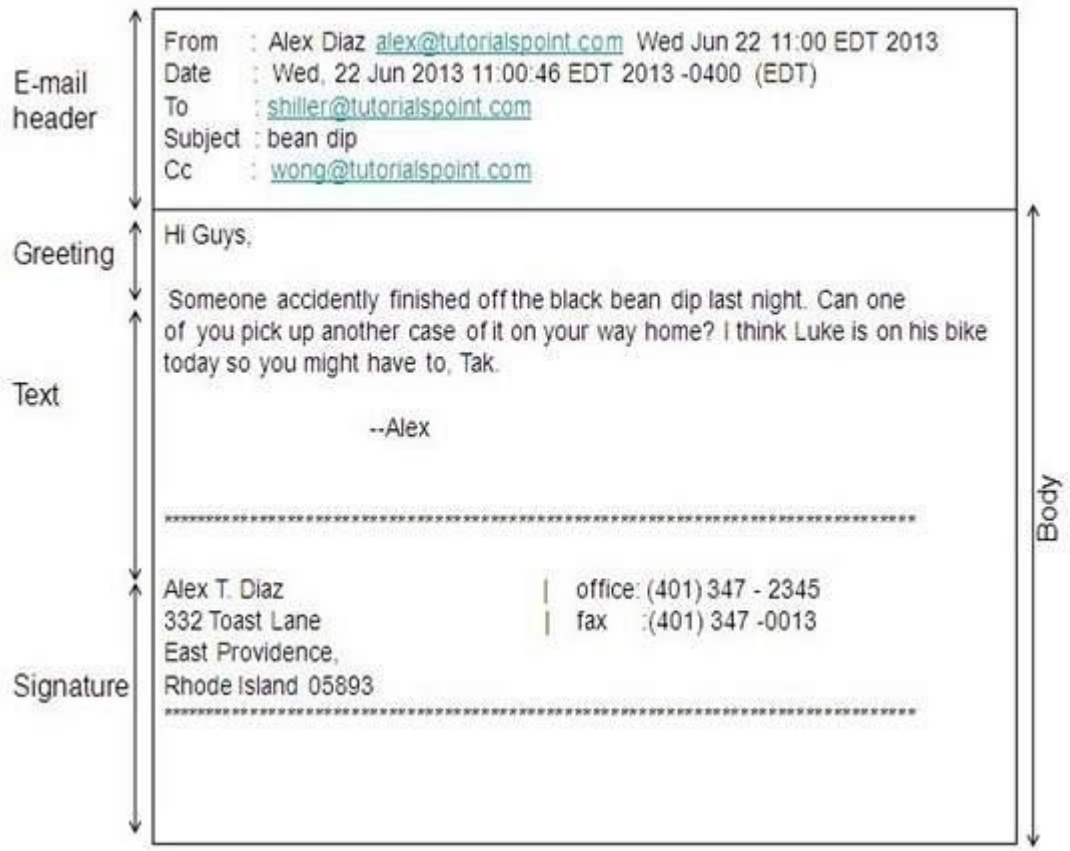
Each user of email is assigned a unique name for his email account. This name is known as E-mail address. Different users can send and receive messages according to the e-mail address.

E-mail is generally of the form username@domainname. For example, webmaster@tutorialspoint.com is an e-mail address where webmaster is username and tutorialspoint.com is domain name.

- The username and the domain name are separated by **@ (at)** symbol.
- E-mail addresses are not case sensitive.
- Spaces are not allowed in e-mail address.

E-mail Message Components:

E-mail message comprises of different components: E-mail Header, Greeting, Text, and Signature. These components are described in the following diagram:



E-mail Header:

The first five lines of an E-mail message is called E-mail header. The header part comprises of following fields:

- From
- Date
- To
- Subject
- CC
- BCC

- From-
  The **From** field indicates the sender's address i.e. who sent the e-mail.
- Date-
  The **Date** field indicates the date when the e-mail was sent.
- To-
  The **To** field indicates the recipient's address i.e. to whom the e-mail is sent.
- Subject-
  The **Subject** field indicates the purpose of e-mail. It should be precise and to the point.
- CC-
  CC stands for Carbon copy. It includes those recipient addresses whom we want to keep informed but not exactly the intended recipient.
- BCC-
  BCC stands for Black Carbon Copy. It is used when we do not want one or more of the recipients to know that someone else was copied on the message.
- Greeting-
  Greeting is the opening of the actual message. Eg. Hi Sir or Hi Guys etc.
- Text-
  It represents the actual content of the message.
- Signature-
  This is the final part of an e-mail message. It includes Name of Sender, Address, and Contact Number.

Advantages:

E-mail has prooved to be powerful and reliable medium of commmunication. Here are the benefits of **E-mail:**

- Reliable
  Many of the mail systems notify the sender if e-mail message was undeliverable.
- Convenience
  There is no requirement of stationary and stamps. One does not have to go to post office. But all these things are not required for sending or receiving an mail.
- Speed
  E-mail is very fast. However, the speed also depends upon the underlying network.
- Inexpensive
  The cost of sending e-mail is very low.
- Printable
  It is easy to obtain a hardcopy of an e-mail. Also an electronic copy of an e-mail can also be saved for records.
- Global

E-mail can be sent and received by a person sitting across the globe.

- Generality

  It is also possible to send graphics, programs and sounds with an e-mail.

Disadvantages:

Apart from several benefits of E-mail, there also exists some disadvantages as discussed below:

- Forgery

  E-mail doesn't prevent from forgery, that is, someone impersonating the sender, since sender is usually not authenticated in any way.

- Overload

  Convenience of E-mail may result in a flood of mail.

- Misdirection

  It is possible that you may send e-mail to an unintended recipient.

- Junk

  Junk emails are undesirable and inappropriate emails. Junk emails are sometimes referred to as spam.

- No Response

  It may be frustrating when the recipient does not read the e-mail and respond on a regular basis.

  E-mail System

  E-mail system comprises of the following three components:

- **Mailer:**

  It is also called **mail program, mail application** or **mail client.** It allows us to manage, read and compose e-mail.

- **Mail Server:**

  The function of mail server is to receive, store and deliver the email. It is must for mail servers to be Running all the time because if it crashes or is down, email can be lost.

- **Mailboxes:**

  Mailbox is generally a folder that contains emails and information about them.

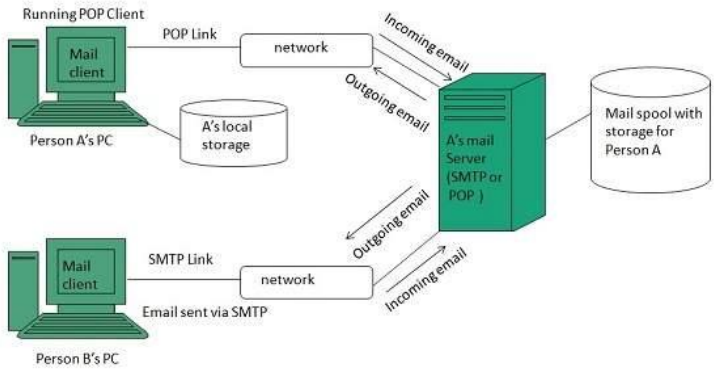
Working of E-mail:

Email working follows the client server approach. In this client is the mailer i.e. the mail application or mail program and server is a device that manages emails.

Following example will take you through the basic steps involved in sending and receiving emails and will give you a better understanding of working of email system:

- Suppose person A wants to send an email message to person B.
- Person A composes the messages using a mailer program i.e. mail client and then select Send option.
- The message is routed to **Simple Mail Transfer Protocol** to person B's mail server.
- The mail server stores the email message on disk in an area designated for person B.
- The disk space area on mail server is called mail spool.
- Now, suppose person B is running a POP client and knows how to communicate with B's mail server.
- It will periodically poll the POP server to check if any new email has arrived for B.As in this case, person B has sent an email for person B, so email is forwarded over the network to B's PC. This is message is now stored on person B's PC.

The following diagram gives pictorial representation of the steps discussed above:



Composing and Sending Email:

Before sending an email, we need to compose a message. When we are composing an email message, we specify the following things:

- Sender's address in To field
- Cc (if required)
- Bcc (if required)
- Subject of email message
- Text
- Signature

You should specify the correct email address; otherwise it will send an error back to the sender.

Once you have specified all the above parameters, It's time to send the email. The mailer program provides a Send button to send email, when you click Send, it is sent to the mail server and a message **mail sent successfully** is shown at the above.

Reading Email:

Every email program offers you an interface to access email messages. Like in Gmail, emails are stored under different tabs such as primary, social, and promotion. When you click one of tab, it displays a list of emails under that tab.

In order to read an email, you just have to click on that email. Once you click a particular email, it gets opened. The opened email may have some file attached with it. The attachments are shown at the bottom of the opened email with an option called **download attachment.**

Replying Email:

After reading an email, you may have to reply that email. To reply an email, click **Reply** option shown at the bottom of the opened email.

Once you click on Reply, it will automatically copy the sender's address in to the To field. Below the To field, there is a text box where you can type the message.

Once you are done with entering message, click Send button. It's that easy. Your email is sent.

Forwarding Email:

It is also possible to send a copy of the message that you have received along with your own comments if you want. This can be done using **forward** button available in mail client software.

The difference between replying and forwarding an email is that when you reply a message to a person who has send the mail but while forwarding you can send it to anyone.

When you receive a forwarded message, the message is marked with a > character in front of each line and **Subject:** field is prefixed with **Fw.**

Deleting Email:

If you don't want to keep email into your inbox, you can delete it by simply selecting the message from the message list and clicking **delete** or pressing the appropriate command.

Some mail clients offers the deleted mails to be stored in a folder called deleted items or trash from where you can recover a deleted email.

Email Features:

Now a day, the mail client comes with enhanced features such as attachment, address book, and MIME support. Here in this chapter we will discuss all of these features which will give you a better understanding of added feature of a mail client program.

Attachment:

Ability to attach file(s) along with the message is one of the most useful features of email. The attachment may be a word document, PowerPoint presentation, audio/video files, or images.

- In order to attach file(s) to an email, click the attach button. As a result, a dialog box appears asking for specifying the name and location of the file you want to attach.
- Once you have selected the appropriate file, it is attached to the mail.
- Usually a paper clip icon appears in the email which indicates that it has an attachment.
- When adding an attachment it is better to compress the attached files so as to reduce the file size and save transmission time as sending and downloading large files consumes a lot of space and time.

Address Book:

Address book feature of a mail program allows the users to store information about the people whom they communicate regularly by sending emails. Here are some of the key features of an Address book:

- Address book includes the nick names, email addresses, phone number etc. of the people.
- Using address book allows us not to memorize email of address of a person, you just have to select recipient name from the list.
- When you select a particular name from the list, the corresponding email address link automatically get inserted in to the To: field.
- Address book also allows creating a group so that you can send a email to very member of the group at once instead of giving each person email address one by one.

MIME Types:

MIME is acronym of Multipurpose Internet Mail Extensions. MIME compliant mailer allows us to send files other than simple text i.e. It allows us to send audio, video, images, document, and pdf files as an attachment to an email.

Suppose if you want to send a word processor document that has a group of tabular columns with complex formatting.

If we transfer the file as text, all the formatting may be lost. MIME compliant mailer takes care of messy details and the message arrives as desired.

The following table describes commonly used MIME Types:

1.	Type	Subtype	Description	File extension(s)
2.	Application	postscript tex troff	Printable postscript document TEX document Printable troff document	.eps, .ps .tex .t, .tr, .roff
3.	Audio	aiff au midi real audio	Apple sound Sun Microsystems sound Musical Instrument Digital Interface Progressive Network sound	.aif, .aiff,.aifc .au, .snd .midi, .mid .ra, .ram
4.	image	gif jpeg png triff	Graphics Interchange Format Joint Photographic Experts Group Portable Network Graphics Tagged Image Modeling Language	.gif .jpeg, .jpg, .jpe .png .tiff, .tif
5.	Model	vrml	Virual reality Modelling Language	.wrl
6.	Text plain sgml	html	Hyper Text Markup Language Unformatted text Standard Generalized Markup language	.html, .htm .txt .sgml
7.	Video	avi mpeg quicktime sgi-movie	Microsoft Audio Video Interleaved Moving Pictures Expert Group Apple QuickTime movie silicon graphic movie	.avi .mpeg, .mpg .qt, .mov .movie

The Email service providers:

There are several email service providers available in the market with their enabled features such as sending, receiving, drafting, storing an email and much more.

The following table shows the popular email service providers:

S.N.	Service and Description
1.	**Gmail** Gmail is an email service that allows users to collect all the messages. It also offers approx 7 GB of free storage.
2.	**Hotmail** Hotmail offers free email and practically unlimited storage accessible on web.
3.	**Yahoo Mail** Yahoo Mail offers unlimited storage, SMS texting, social networking and instant messaging to boot.
4.	**iCloud Mail** iCloud Mail offers ample storage, IMAP access, and an elegantly functional web application.
5.	**ATM Mail** ATM Mail is a free email service with good spam protection.
6.	**Mail.com** and **GMX Mail** Mail.com and GMX Mail offers reliable mail service with unlimited online storage.
7.	**Shortmail** Shortmail offers easy and fast email service but with limited 500 characters per message.
8.	**Inbox.com** Inbox.com offers 5 GB of free online storage. IMAP is not supported by Inbox.com
9.	**Facebook Messages** Facebook Messages includes the message conversation.
10.	**My Way Mail** My Way Mail offers clean and fast free email service but lacks in secure messaging.

3. SMTP/POP Protocols:

E-mail Protocols are set of rules that help the client to properly transmit the information to or from the mail server. Here in this tutorial, we will discuss various protocols such as **SMTP, POP,** and **IMAP.**

SMPTP:

SMTP stands for **Simple Mail Transfer Protocol**. It was first proposed in 1982. It is a standard protocol used for sending e-mail efficiently and reliably over the internet.

Key Points:

- SMTP is application level protocol.
- SMTP is connection oriented protocol.
- SMTP is text based protocol.
- It handles exchange of messages between e-mail servers over TCP/IP network.
- Apart from transferring e-mail, SMPT also provides notification regarding incoming mail.
- When you send e-mail, your e-mail client sends it to your e-mail server which further contacts the recipient mail server using SMTP client.
- These SMTP commands specify the sender's and receiver's e-mail address, along with the message to be send.
- The exchange of commands between servers is carried out without intervention of any user.
- In case, message cannot be delivered, an error report is sent to the sender which makes SMTP a reliable protocol.

SMTP Commands:

The following table describes some of the SMTP commands:

S.N.	Command Description
1	**HELLO** This command initiates the SMTP conversation.
2	**EHELLO** This is an alternative command to initiate the conversation. ESMTP indicates that the sender server wants to use extended SMTP protocol.
3	**MAIL FROM** This indicates the sender's address.
4	**RCPT TO** It identifies the recipient of the mail. In order to deliver similar message to multiple users this command can be repeated multiple times.
5	**SIZE** This command let the server know the size of attached message in bytes.
6	**DATA** The **DATA** command signifies that a stream of data will follow. Here stream of data refers to the body of the message.
7	**QUIT** This commands is used to terminate the SMTP connection.
8	**VERFY** This command is used by the receiving server in order to verify whether the given username is valid or not.
9	**EXPN** It is same as VRFY, except it will list all the users name when it used with a distribution list.

IMAP:

IMAP stands for **Internet Message Access Protocol.** It was first proposed in 1986. There exist five versions of IMAP as follows:

1. Original IMAP
2. IMAP2
3. IMAP3
4. IMAP2bis
5. IMAP4

Key Points:

- IMAP allows the client program to manipulate the e-mail message on the server without downloading them on the local computer.
- The e-mail is hold and maintained by the remote server.
- It enables us to take any action such as downloading, delete the mail without reading the mail.It enables us to create, manipulate and delete remote message folders called mail boxes.
- IMAP enables the users to search the e-mails.
- It allows concurrent access to multiple mailboxes on multiple mail servers.

IMAP Commands

The following table describes some of the IMAP commands:

S.N.	Command Description
1	**IMAP_LOGIN** This command opens the connection.
2	**CAPABILITY** This command requests for listing the capabilities that the server supports.
3	**NOOP** This command is used as a periodic poll for new messages or message status updates during a period of inactivity.
4	**SELECT** This command helps to select a mailbox to access the messages.
5	**EXAMINE** It is same as SELECT command except no change to the mailbox is permitted.
6	**CREATE** It is used to create mailbox with a specified name.
7	**DELETE** It is used to permanently delete a mailbox with a given name.
8	**RENAME** It is used to change the name of a mailbox.
9	**LOGOUT** This command informs the server that client is done with the session. The server must send BYE untagged response before the OK response and then close the network connection.

POP:

POP stands for Post Office Protocol. It is generally used to support a single client. There are several versions of POP but the POP 3 is the current standard.

Key Points

- POP is an application layer internet standard protocol.
- Since POP supports offline access to the messages, thus requires less internet usage time.
- POP does not allow search facility.
- In order to access the messaged, it is necessary to download them.
- It allows only one mailbox to be created on server.
- It is not suitable for accessing non mail data.
- POP commands are generally abbreviated into codes of three or four letters. Eg. STAT.

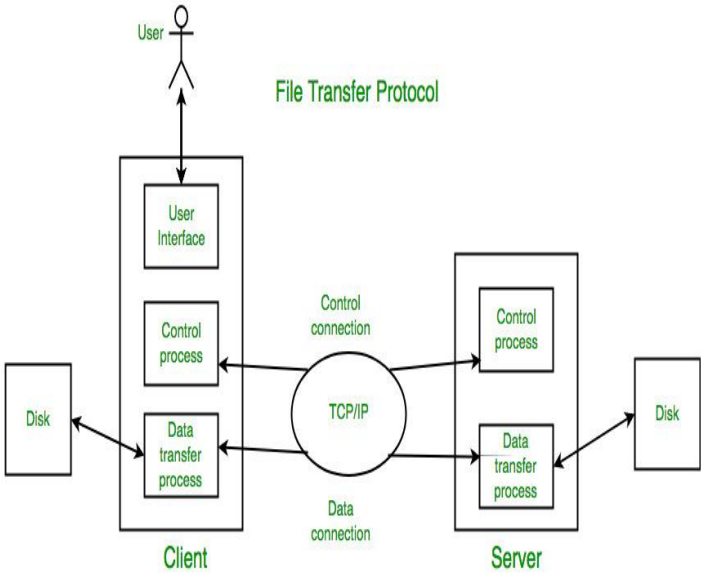
POP Commands

The following table describes some of the POP commands:

S.N.	Command Description
1	**LOGIN** This command opens the connection.
2	**STAT** It is used to display number of messages currently in the mailbox.
3	**LIST** It is used to get the summary of messages where each message summary is shown.
4	**RETR** This command helps to select a mailbox to access the messages.
5	**DELE** It is used to delete a message.
6	**RSET** It is used to reset the session to its initial state.
7	**QUIT** It is used to log off the session.

4. File transfer protocol (FTP):

File Transfer Protocol (FTP) is an application layer protocol which moves files between local and remote file systems. It runs on the top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.



What is control connection?

For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of control connection. The control connection is initiated on port number 21.

What is data connection?

For sending the actual file, FTP makes use of data connection. A data connection is initiated on port number 20.

FTP sends the control information out-of-band as it uses a separate control connection. Some protocols send their request and response header lines and the data in the same TCP connection. For this reason, they are said to send their control information in-band. HTTP and SMTP are such examples.

FTP Session:

When a FTP session is started between a client and a server, the client initiates a control TCP connection with the server side. The client sends control information over this. When the server receives this, it initiates a data connection to the client side. Only one file can be sent over one data connection. But the control connection remains active throughout the user session. As we know HTTP is stateless i.e. it does not have to keep track of any user state. But FTP needs to maintain a state about its user throughout the session.

Data Structures: FTP allows three types of data structures :

1. **File Structure –** In file-structure there is no internal structure and the file is considered to be a continuous sequence of data bytes.
2. **Record Structure –** In record-structure the file is made up of sequential records.
3. **Page Structure –** In page-structure the file is made up of independent indexed pages.

5. The WWW & HTTP:

Overview

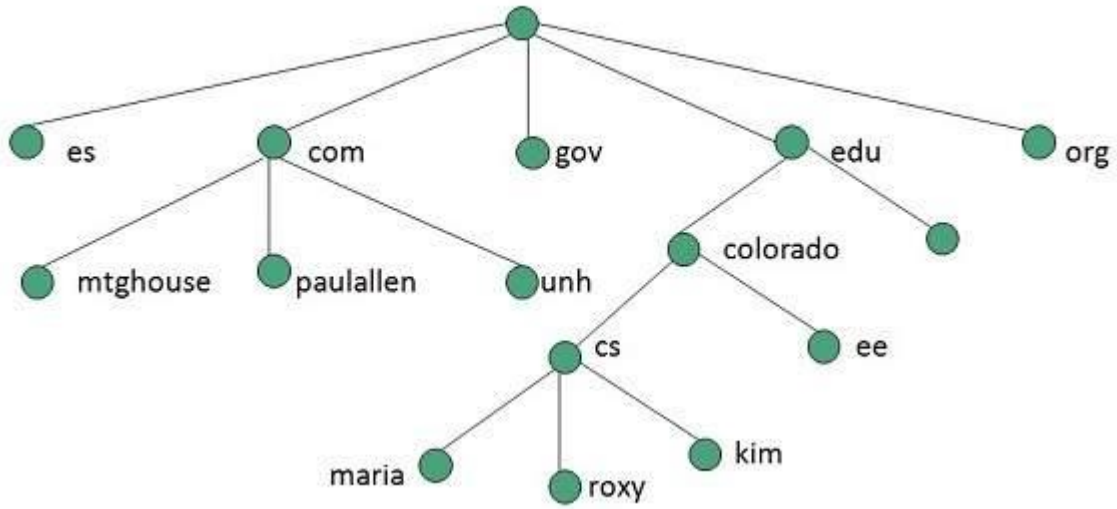
WWW stands for **World Wide Web.** A technical definition of the World Wide Web is : all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).

A broader definition comes from the organization that Web inventor **Tim Berners-Lee** helped found, the **World Wide Web Consortium (W3C).**

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

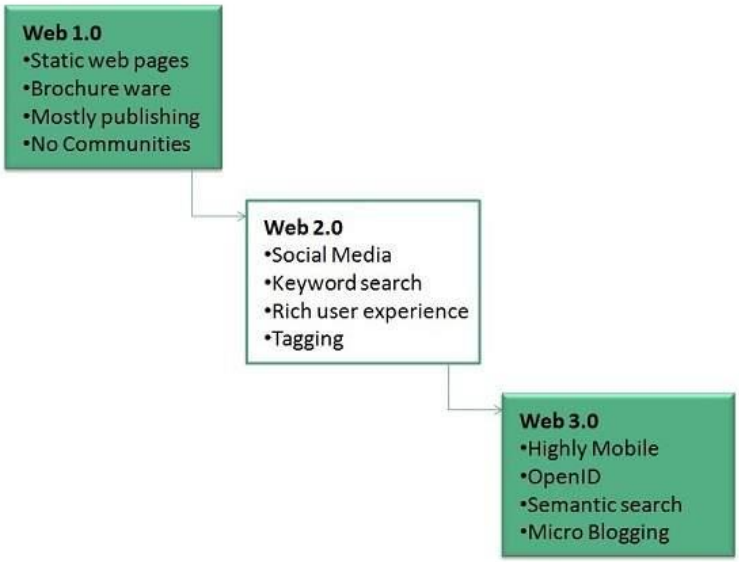
Internet and **Web** is not the same thing: Web uses internet to pass over the information.



Evolution

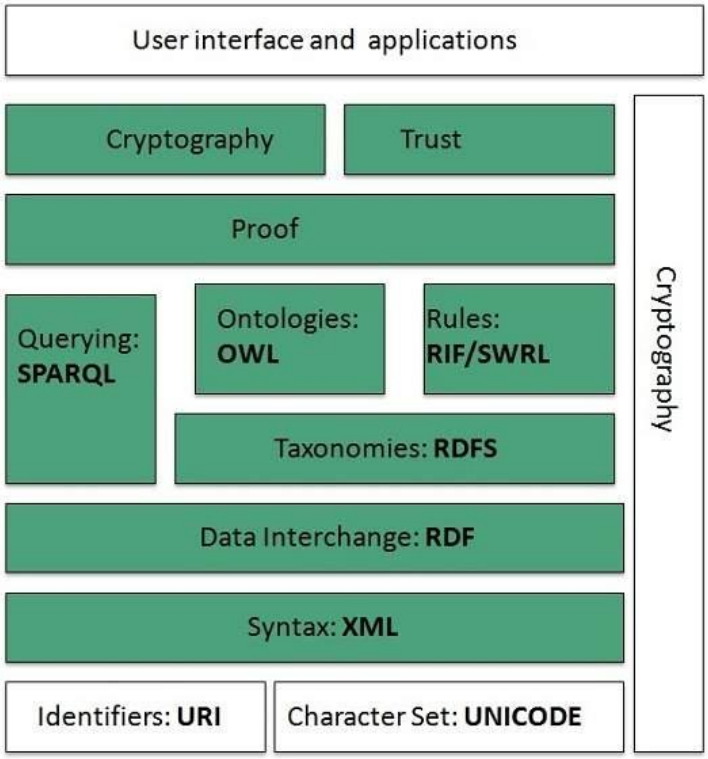
World Wide Web was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva.** World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN.** Eventually it became **World Wide Web.**

The following diagram briefly defines evolution of World Wide Web:



WWW Architecture:

WWW architecture is divided into several layers as shown in the following diagram:



Identifiers and Character Set:

Uniform Resource Identifier (URI) is used to uniquely identify resources on the web and **UNICODE** makes it possible to built web pages that can be read and write in human languages.

Syntax:

XML (Extensible Markup Language) helps to define common syntax in semantic web.

Data Interchange

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

Taxonomies

RDF Schema (RDFS) allows more standardized description of **taxonomies** and other **ontological** constructs.

Ontologies:

Web Ontology Language (OWL) offers more constructs over RDFS. It comes in following three versions:

- OWL Lite for taxonomies and simple constraints.
- OWL DL for full description logic support.
- OWL for more syntactic freedom of RDF

Rules

RIF and **SWRL** offers rules beyond the constructs that are available from **RDFs** and **OWL.** Simple Protocol and **RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

Proof

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

Cryptography

Cryptography means such as digital signature for verification of the origin of sources is used.

User Interface and Applications

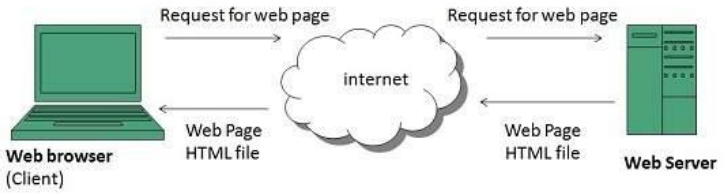
On the top of layer **User interface and Applications** layer is built for user interaction.

WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

1. User enters the URL (say, **http://www.tutorialspoint.com**) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.

5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.



Future:

There had been a rapid development in field of web. It has its impact in almost every area such as education, research, technology, commerce, marketing etc. So the future of web is almost unpredictable.

Apart from huge development in field of WWW, there are also some technical issues that W3 consortium has to cope up with.

User Interface

Work on higher quality presentation of 3-D information is under deveopment. The W3 Consortium is also looking forward to enhance the web to full fill requirements of global communities which would include all regional languages and writing systems.

Technology

Work on privacy and security is under way. This would include hiding information, accounting, access control, integrity and risk management.

Architecture

There has been huge growth in field of web which may lead to overload the internet and degrade its performance. Hence more better protocol are required to be developed.

Web Page:

web page is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.

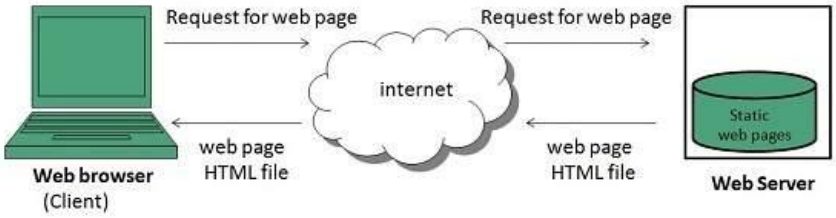
A web page can cotain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages.

Collection of linked web pages on a web server is known as **website.** There is unique **Uniform Resource Locator (URL)** is associated with each web page.

Static Web page

Static web pages are also known as flat or stationary web page. They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can't do any modification or interact with the information.

Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.



Dynamic Web page:

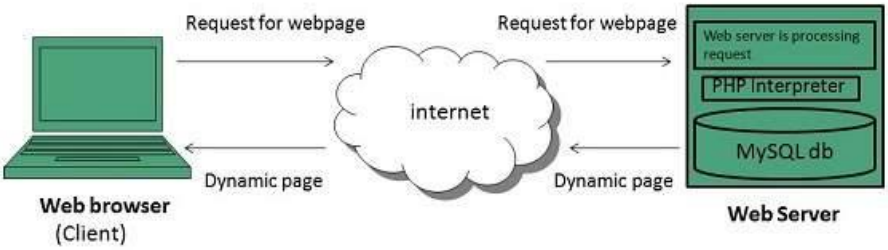
Dynamic web page shows different information at different point of time. It is possible to change a portaion of a web page without loading the entire web page. It has been made possible using **Ajax** technology.

Server-side dynamic web page

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

Client-side dynamic web page

It is processed using client side scripting such as JavaScript. And then passed in to **Document Object Model (DOM).**



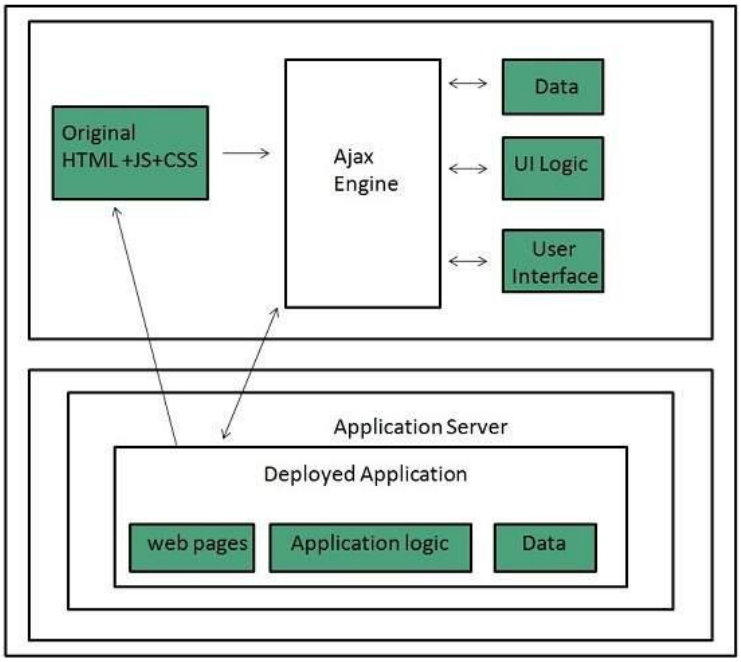
Scripting Languages:

Scripting languages are like programming languages that allow us to write programs in form of script. These scripts are interpreted not compiled and executed line by line. Scripting language is used to create dynamic web pages.

Client-side Scripting

Client-side scripting refers to the programs that are executed on client-side. Client-side scripts contains the instruction for the browser to be executed in response to certain user's action.

Client-side scripting programs can be embedded into HTML files or also can be kept as separate files.

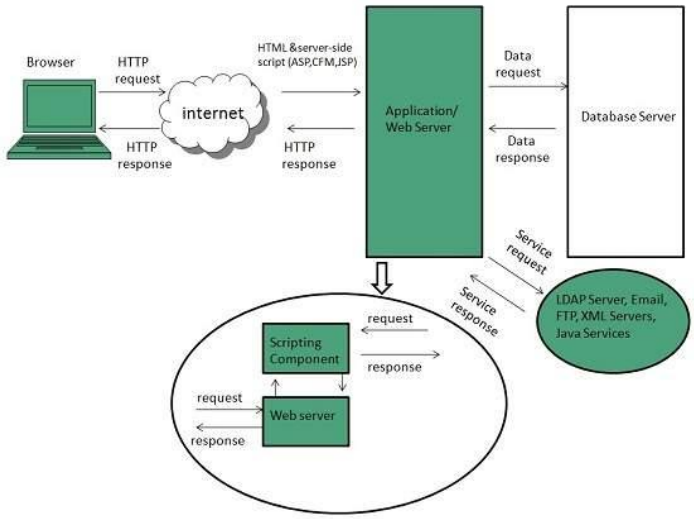


Following table describes commonly used Client-Side scripting languages:

S.N.	Scripting Language Description
1.	**JavaScript** It is a prototype based scripting language. It inherits its naming conventions from java. All java script files are stored in file having **.js** extension.
2.	**ActionScript**It is an object oriented programming language used for the development of websites and software targeting Adobe flash player.
3.	**Dart** It is an open source web programming language developed by Google. It relies on source-to-source compiler to JavaScript.
4.	**VBScript** It is an open source web programming language developed by Microsoft. It is superset of JavaScript and adds optional static typing class-based object oriented programming.

Server-side Scripting:

Sever-side scripting acts as an interface for the client and also limit the user access the resources on web server. It can also collects the user's characteristics in order to customize response.



Following table describes commonly used Server-Side scripting languages:

S.N.	Scripting Language Description
1.	**ASP** Active Server Pages (ASP)is server-side script engine to create dynamic web pages. It supports **Component Object Model (COM)** which enables ASP web sites to access functionality of libraries such as DLL.
2.	**ActiveVFP** It is similar to PHP and also used for creating dynamic web pages. It uses native **Visual Foxpro** language and database.
3.	**ASP.net** It is used to develop dynamic websites, web applications, and web services.
4.	**Java** Java Server Pages are used for creating dynamic web applications. The Java code is compiled into byte code and run by **Java Virtual Machine (JVM).**
5.	**Python** It supports multiple programming paradigms such as object-oriented, and functional programming. It can also be used as non-scripting language using third party tools such as **Py2exe** or **Pyinstaller.**
6.	**WebDNA** It is also a server-side scripting language with an embedded database system.

6. The SNMP Protocol:

Simple Network Management Protocol (SNMP):

If an organization has 1000 of devices then to check all devices, one by one every day, are working properly or not is a hectic task. To ease these up, Simple Network Management Protocol (SNMP) is used.

Simple Network Management Protocol (SNMP) –

SNMP is an application layer protocol which uses UDP port number 161/162.SNMP is

used to monitor the network, detect network faults and sometimes even used to configure remote devices.

SNMP components –

There are 3 components of SNMP:

1. **SNMP Manager –**
   It is a centralised system used to monitor network.It is also known as Network Management Station (NMS)
2. **SNMP agent –**
   It is a software management software module installed on a managed device. Managed devices can be network devices like PC, router, switches, servers etc.
3. **Management Information Base –**
   MIB consists of information of resources that are to be managed. These information is organised hierarchically. It consists of objects instances which are essentially variables.