COMPUTER NETWORKS

Lecture Notes
UNIT-I

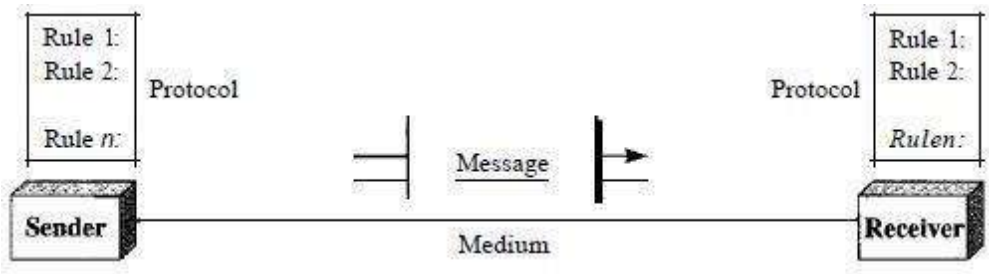
UNIT -I
Introduction to Computer Networks

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

Computer Network: A computer network is a set of computers connected together for the purpose of sharing resources. The most common resource shared today is connection to the Internet. Other shared resources can include a printer or a file server. The Internet itself can be considered a computer network.

1.1.1 Components:

A data communications system has five components.



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

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

3. Receiver. The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

4. Transmission medium. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves

5. Protocol. A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

1.1.2 Data Representation:

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

Text:

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

Numbers:

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

Images:

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

Audio:

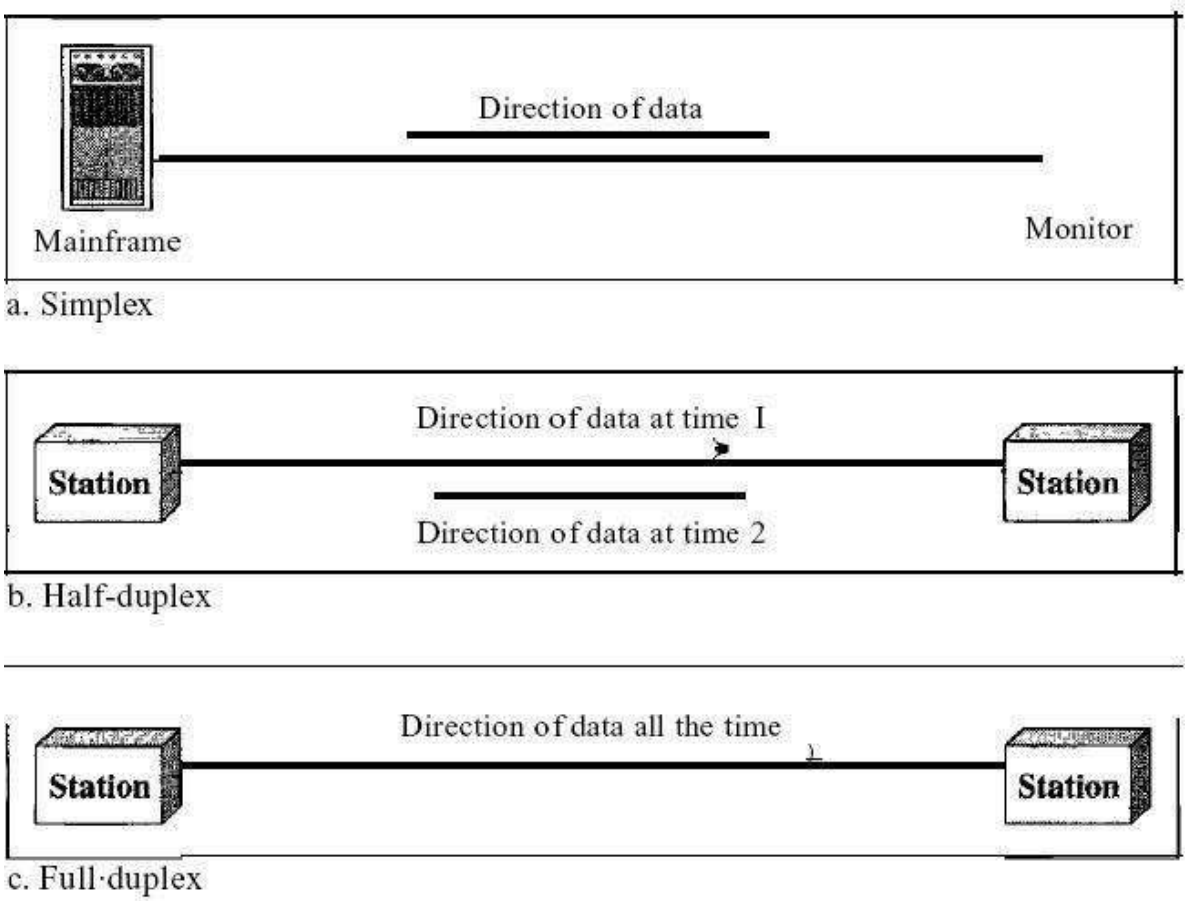
Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal. In Chapters 4 and 5, we learn how to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

Video:

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

1.1.3 Data Flow

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



Simplex:

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

Half-Duplex:

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

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

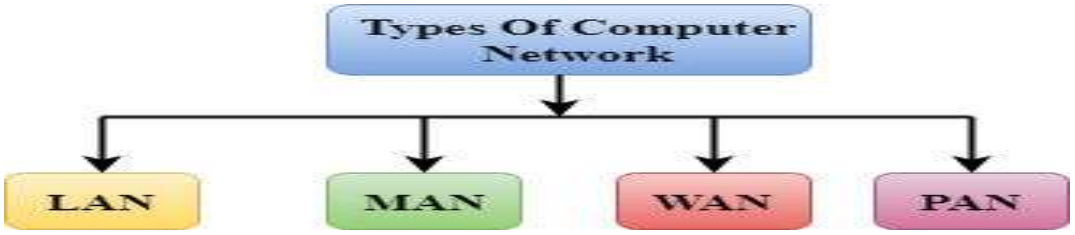
The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

Full-Duplex:

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

Types of Computer Networks:

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

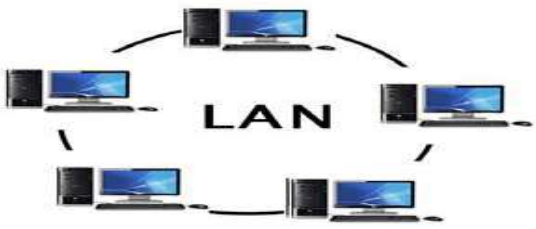


5

1. **Local Area Network (LAN).**
2. **Metropolitan Area Network (MAN).**
3. **Wide Area Network(WAN).**
4. **Personal Area Network.** A **Personal Area Network (PAN)** is the most basic type, usually used for homes or home offices. ...

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 a twisted pair, coaxial cable, etc.
- It is less costly as it is built with inexpensive hardware such as hubs, network adapters, and Ethernet cables.
- The data is transferred at an extremely faster rate in Local Area Network.
- Local Area Network provides higher security.



Metropolitan Area Network(MAN) : A metropolitan area network, or MAN, covers a city. The best-known example of a MAN is the cable television network available in many cities. This system grew from earlier community antenna systems used in areas with poor over-the-air television reception. In these early systems, large antenna was placed on top of a nearby hill and signal was then piped to the subscribers' houses.

* A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.
* Government agencies use MAN to connect to the citizens and private industries.
* In MAN, various LANs are connected to each other through a telephone exchange line.
* The most widely used protocols in MAN are RS-232, Frame Relay, ATM, ISDN, etc.
*It has a higher range than Local Area Network (LAN).

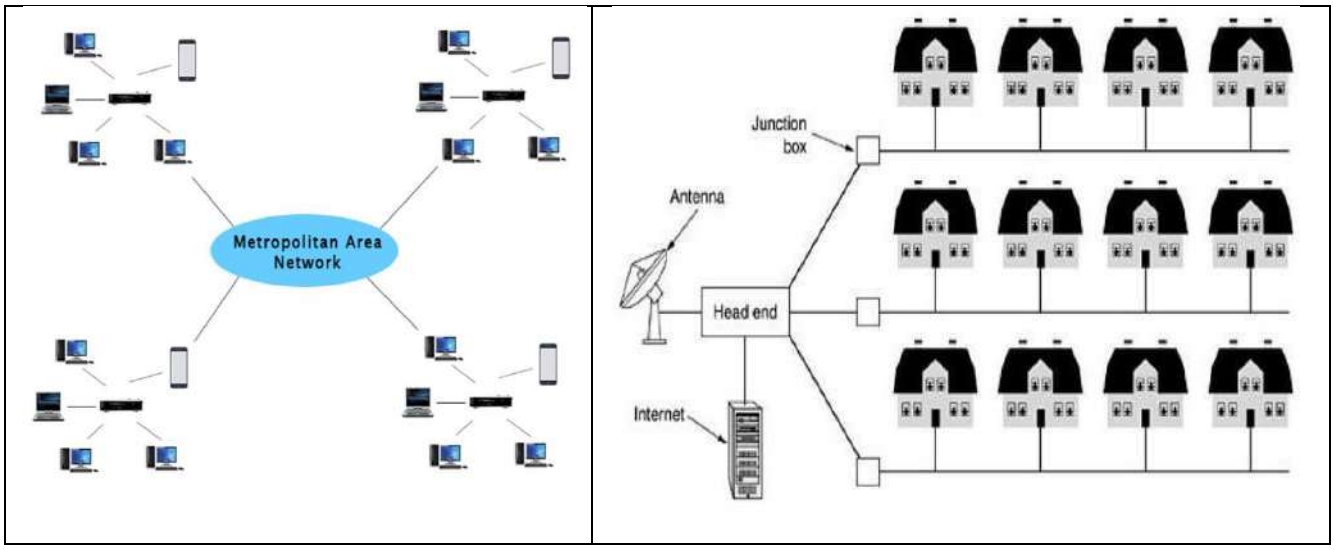


Fig.: Metropolitan area network based on cable TV.

Wide Area Network (WAN) :

A wide area network, or WAN, spans a large geographical area, often a country or continent. It contains a collection of machines intended for running user (i.e., application) programs. These machines are called as hosts. In most WANs, the network contains numerous transmission lines, each one connecting a pair of routers. If two routers that do not share a transmission line wish to communicate, they must do this indirectly, via other routers. When a packet is sent from one route to another via one or more intermediate routers, the packet is received at each intermediate router in its entirety, stored there until the required output line is free, and then forwarded. A subnet organized according to this principle is called a store-and-forward or packet-switched subnet.

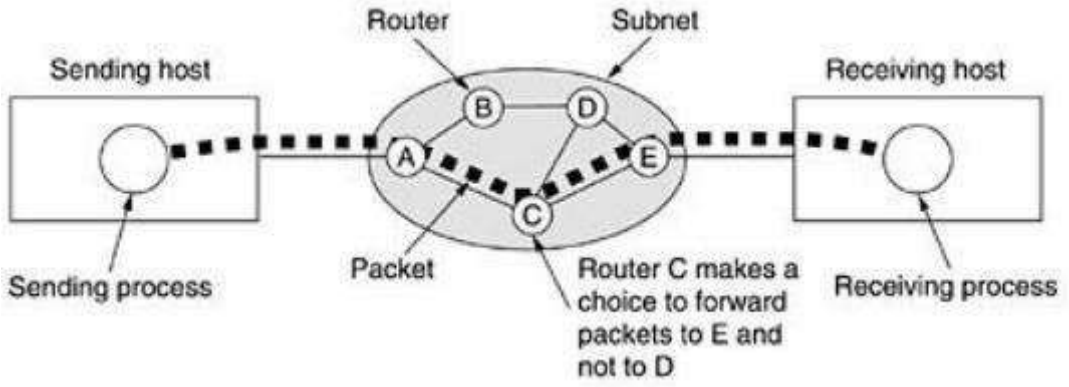
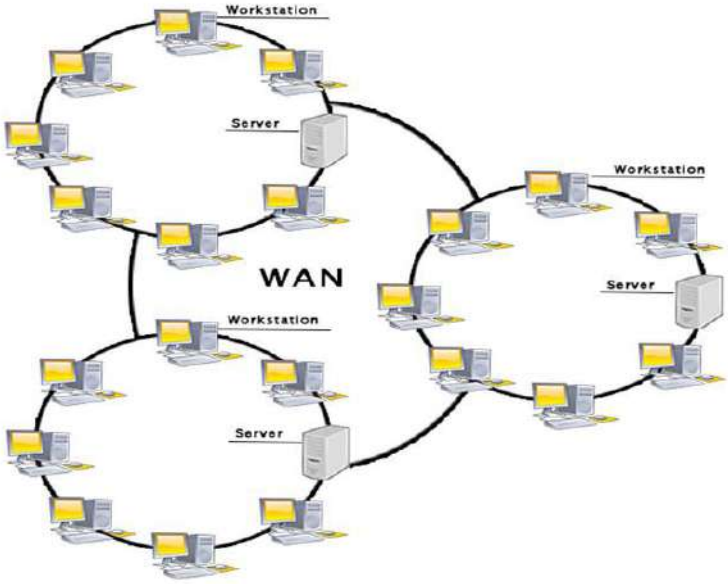


Fig.: A stream of packets from sender to receiver

* A Wide Area Network is a network that extends over a large geographical area such as states or countries.
* A Wide Area Network is quite bigger network than the LAN.
* A Wide Area Network is not limited to a single location, but it spans over a large geographical area through a telephone line, fibre 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.



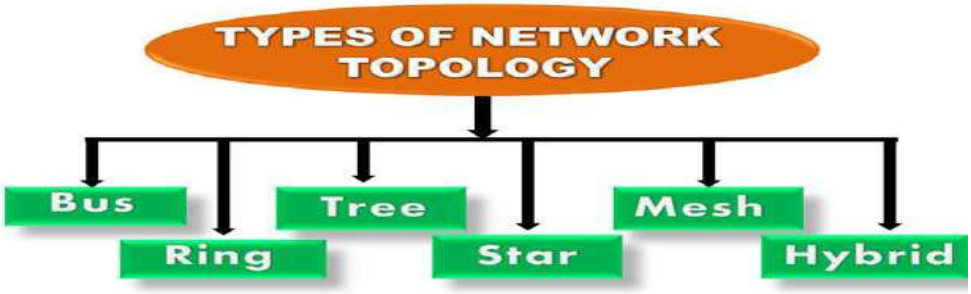
Personal Area Network (PAN):

- Personal Area Network is a network arranged within 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.
- Thomas Zimmerman was the first research scientist to bring the idea of the Personal Area Network.
- Personal Area Network covers an area of 30 feet.
- Personal computer devices that are used to develop the personal area network are the aptop, mobile phones, media player and play stations.

Topologies

What is Topology? :

- Topology defines the structure of the network of how all the components are interconnected to each other.
- There are two types of topology: **physical and logical topology**.
- Physical topology is the geometric representation of all the nodes in a network.



1. **Bus Topology:**

- The bus topology is designed in such a way that all the stations are connected through a single cable known as a backbone cable.
- Each node is either connected to the backbone cable by drop cable or directly connected to the backbone cable
- When a node wants to send a message over the network, it puts a message over the network. All the stations available in the network will receive the message whether it has been addressed or not.
- The bus topology is mainly used in 802.3 (Ethernet) and 802.4 standard networks.
- The configuration of a bus topology is quite simpler as compared to other topologies.

- The backbone cable is considered as a **"single lane"** through which the message is broadcast to all the stations.
- The most common access method of the bus topologies is **CSMA** (Carrier Sense Multiple Access).

Advantages and Disadvantages of Bus Topology:

Advantages of Bus Topology :

- It is cost effective.
- Cable required is least compared to other network topology.
- Used in small networks.
- It is easy to understand.
- Easy to expand joining two cables together.

Disadvantages of Bus Topology :

- Cables fails then whole network fails.
- If network traffic is heavy or nodes are more the performance of the network decreases.
- Cable has a limited length.
  It is slower than the ring topology.

2. Ring Topology :

- Ring topology is like a bus topology, but with connected ends.
- The node that receives the message from the previous computer will retransmit to the next node.
- The data flows in one direction, i.e., it is unidirectional.
- The data flows in a single loop continuously known as an endless loop.
- It has no terminated ends, i.e., each node is connected to other node and having no termination point.
- The data in a ring topology flow in a clockwise direction.
- The most common access method of the ring topology is **token passing**.

  Token passing: It is a network access method in which token is passed from one node to another node.

  Token: It is a frame that circulates around the network

Working of Token passing :

- A token moves around the network, and it is passed from computer to computer until it reaches the destination
- The sender modifies the token by putting the address along with the data.
- The data is passed from one device to another device until the destination address matches. Once the token received by the destination device, then it sends the acknowledgment to the sender.
- In a ring topology, a token is used as a carrier.

Advantages and Disadvantages of Ring topology :

Advantages of Ring topology:

- **Network Management:** Faulty devices can be removed from the network without bringing the network down.
- **Product availability:** Many hardware and software tools for network operation and monitoring are available.
- **Cost:** Twisted pair cabling is inexpensive and easily available. Therefore, the installation cost is very low.
- **Reliable:** It is a more reliable network because the communication system is not dependent on the single host computer.

Disadvantages of Ring topology :

- **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
- **Failure:** The breakdown in one station leads to the failure of the overall network.
- **Reconfiguration difficult:** Adding new devices to the network would slow down the network.
- **Delay:** Communication delay is directly proportional to the number of nodes. Adding new devices increases the communication delay.

3. Star Topology:

- Star topology is an arrangement of the network in which every node is connected to the central hub, switch or a central computer.
- The central computer is known as a **server**, and the peripheral devices attached to the server are known as **clients**.
- Coaxial cable or RJ-45 cables are used to connect the computers.
- Hubs or Switches are mainly used as connection devices in a **physical star topology**.
- Star topology is the most popular topology in network implementation.

Advantages and Disadvantages of Star topology of Star topology:

Advantages of Star topology :

- **Efficient troubleshooting:** Troubleshooting is quite efficient in a star topology as compared to bus topology.In a bus topology, the manager has to inspect the kilometers of cable. In a star topology, all the stations are connected to the centralized network.
- **Limited failure:** As each station is connected to the central hub with its own cable, therefore failure in one cable will not affect the entire network.
- **Familiar technology:** Star topology is a familiar technology as its tools are cost-effective.
- **Easily expandable:** It is easily expandable as new stations can be added to the open ports on the hub.

- **Cost effective:** Star topology networks are cost-effective as it uses inexpensive coaxial cable.
- **High data speeds:** It supports a bandwidth of approx 100Mbps. Ethernet 100BaseT is one of the most popular Star topology networks.

Disadvantages of Star topology :

- **A Central point of failure:** If the central hub or switch goes down, then all the connected nodes will not be able to communicate with each other.
- **Cable:** Sometimes cable routing becomes difficult when a significant amount of routing is required.

4. Tree topology

- Tree topology combines the characteristics of bus topology and star topology.
- A tree topology is a type of structure in which all the computers are connected with each other in hierarchical fashion.
- The top-most node in tree topology is known as a root node, and all other nodes are the descendants of the root node.
- There is only one path exists between two nodes for the data transmission. Thus, it forms a parent -child hierarch

Advantages and Disadvantages of Tree topology:

Advantages of Tree topology:

- **Support for broadband transmission:** Tree topology is mainly used to provide broadband transmission, i.e., signals are sent over long distances without being attenuated.
- **Easily expandable:** We can add the new device to the existing network. Therefore, we can say that tree topology is easily expandable.
- **Easily manageable:** In tree topology, the whole network is divided into segments known as star networks which can be easily managed and maintained.
- **Error detection:** Error detection and error correction are very easy in a tree topology.
- **Limited failure:** The breakdown in one station does not affect the entire network.
- **Point-to-point wiring:** It has point-to-point wiring for individual segments

Disadvantages of Tree topology

- **Difficult troubleshooting:** If any fault occurs in the node, then it becomes difficult to troubleshoot the problem.
- **High cost:** Devices required for broadband transmission are very costly.
- **Failure:** A tree topology mainly relies on main bus cable and failure in main bus cable will damage the overall network.
- **Reconfiguration difficult:** If new devices are added, then it becomes difficult to rec

5. Mesh topology

- Mesh technology is an arrangement of the network in which computers are interconnected with each other through various redundant connections.
- There are multiple paths from one computer to another computer.
- It does not contain the switch, hub or any central computer which acts as a central point of communication.
- The Internet is an example of the mesh topology.
- Mesh topology is mainly used for WAN implementations where communication failures are a critical concer.
- Mesh topology is mainly used for wireless networks.
- Mesh topology can be formed by using the formula:

Number of cables = (n*(n-1))/2;
Where n is the number of nodes that represents the network.

Advantages and Disadvantages of Mesh topology :

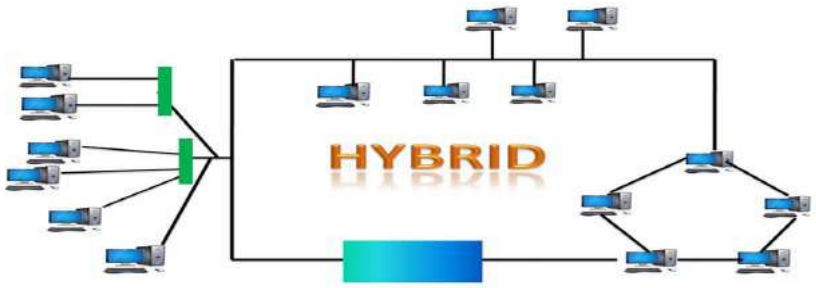
Advantages of Mesh topology:

- **Reliable:** The mesh topology networks are very reliable as if any link breakdown will not affect the communication between connected computers.
- **Fast Communication:** Communication is very fast between the nodes.
- **Easier Reconfiguration:** Adding new devices would not disrupt the communication between other Devices Disadvantages of Mesh topology
- **Cost:** A mesh topology contains a large number of connected devices such as a router and more transmission media than other topologies.

- **Management:** Mesh topology networks are very large and very difficult to maintain and manage. If the network is not monitored carefully, then the communication link failure goes undetected.
- **Efficiency:** In this topology, redundant connections are high that reduces the efficiency of the network.

6. Hybrid Topology

- The combination of various different topologies is known as **Hybrid topology**.
- A Hybrid topology is a connection between different links and nodes to transfer the data.
- When two or more different topologies are combined together is termed as Hybrid topology and if similar topologies are connected with each other will not result in Hybrid topology. For example, if there exist a ring topology in one branch of ICICI bank and bus topology in another branch of ICICI bank, connecting these two topologies will result in Hybrid topology.



THE INTERNET

The Internet has revolutionized many aspects of our daily lives. It has affected the way we do business as well as the way we spend our leisure time. Count the ways you've used the Internet recently. Perhaps you've sent electronic mail (e-mail) to a business associate, paid a utility bill, read a newspaper from a distant city, or looked up a local movie schedule-all by using the Internet. The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.

A Brief History :

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letter i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks. The Advanced Research Projects Agency (ARPA) in the Department of Defense (DoD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort. In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected

computers.

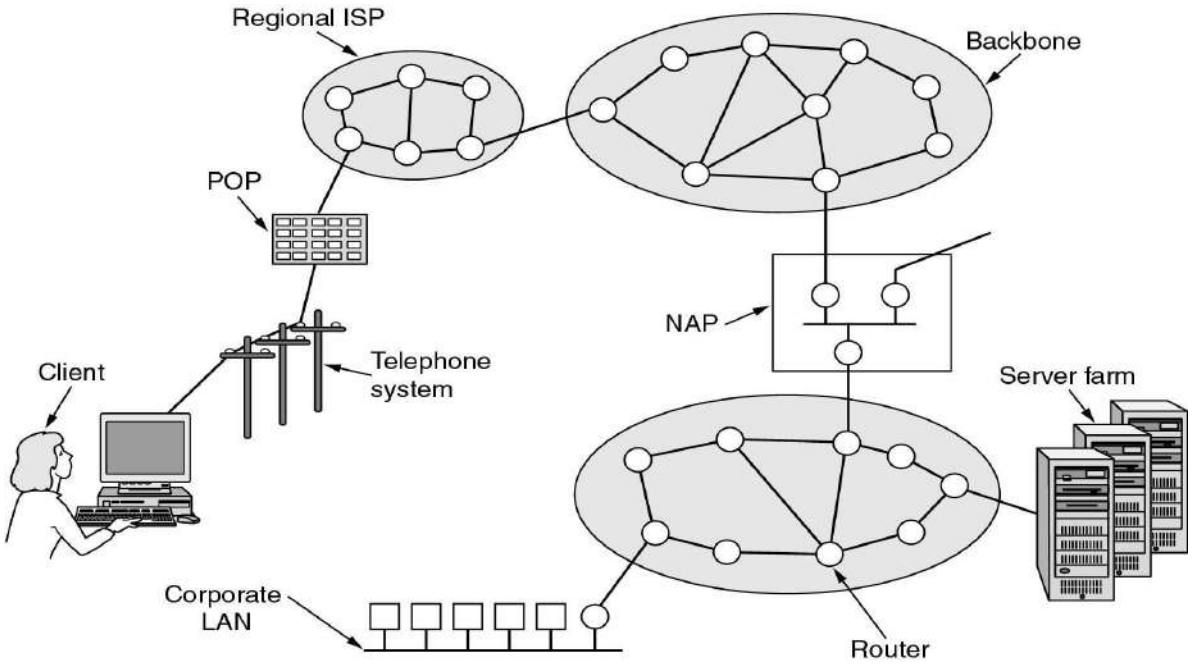
Transmission Control Protocol (TCP) and Internetworking Protocol (lP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The internetworking protocol became known as TCPIIP.

The Internet Today

The Internet has come a long way since the 1960s. The Internet today is not a simple hierarchical structure. It is made up of many wide- and local-area networks joined by connecting devices and switching stations. It is difficult to give an accurate representation of the Internet because it is continually changing-new networks are being added, existing networks are adding addresses, and networks of defunct companies are being removed.

Internet service provider (ISP), company that provides Internet connections and services to individuals and organizations. In addition to providing access to the Internet, **ISPs** may also provide software packages (such as browsers), e-mail accounts, and a personal Web site or home page.

POP protocol is used in the **application layer protocol**, and it delivers best ability to fetch and receive all email by users.



International Internet Service Providers:

At the top of the hierarchy are the international service providers that connect nations together.

National Internet Service Providers:

The national Internet service providers are backbone networks created and maintained by specialized companies. There are many national ISPs operating in North America; some of the most well known are SprintLink, PSINet, UUNet Technology, AGIS, and internet Mel. To provide connectivity between the end users, these backbone networks are connected by complex switching stations (normally run by a third party) called network access points (NAPs). Some national ISP networks are also connected to one another by private switching stations called *peering points.* These normally operate at a high data rate (up to 600 Mbps).

Regional Internet Service Providers:

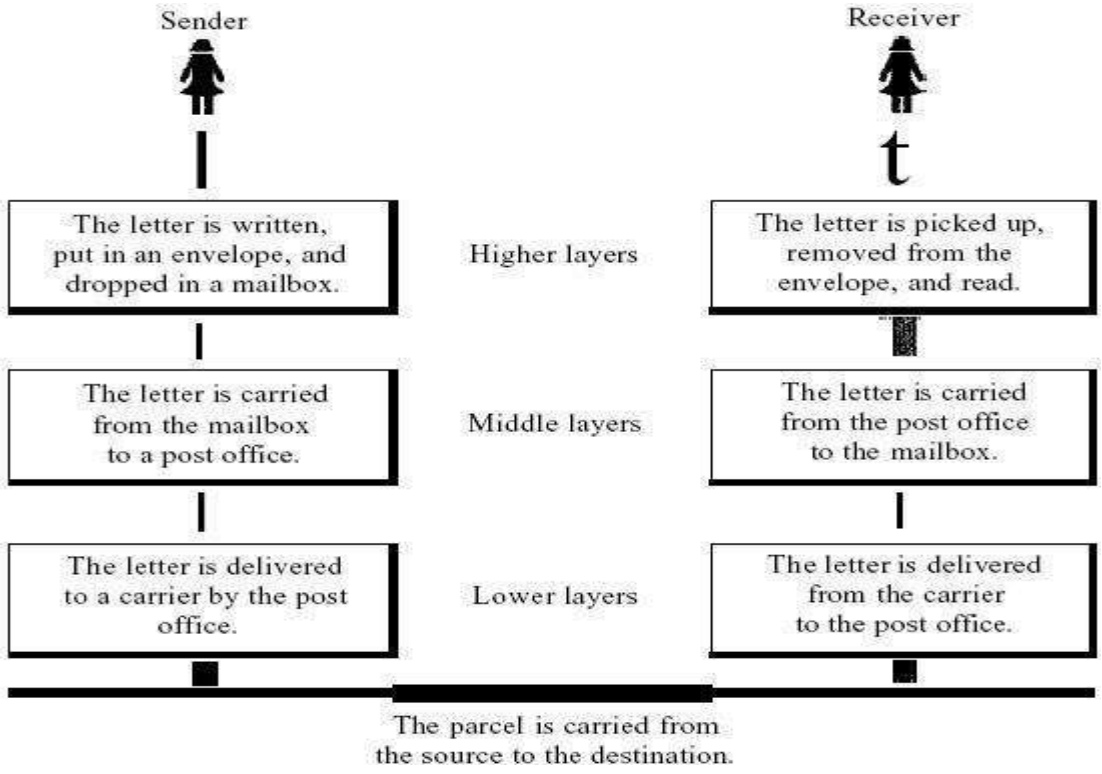
Regional internet service providers or regional ISPs are smaller ISPs that are connected to one or more national ISPs. They are at the third level of the hierarchy with a smaller data rate.

Local Internet Service Providers:

Local Internet service providers provide direct service to the end users. The local ISPs can be connected to regional ISPs or directly to national ISPs. Most end users are connected to the local ISPs. Note that in this sense, a local ISP can be a company that just provides Internet services, a corporation with a network that supplies services to its own employees, or a nonprofit organization, such as a college or a university, that runs its own network. Each of these local ISPs can be connected to a regional or national service provider.

LAYERED TASKS

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



Sender, Receiver, and Carrier

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

At the Sender Site

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

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

0n the Way: The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the Receiver Site

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

The OSI Reference Model:

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

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

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

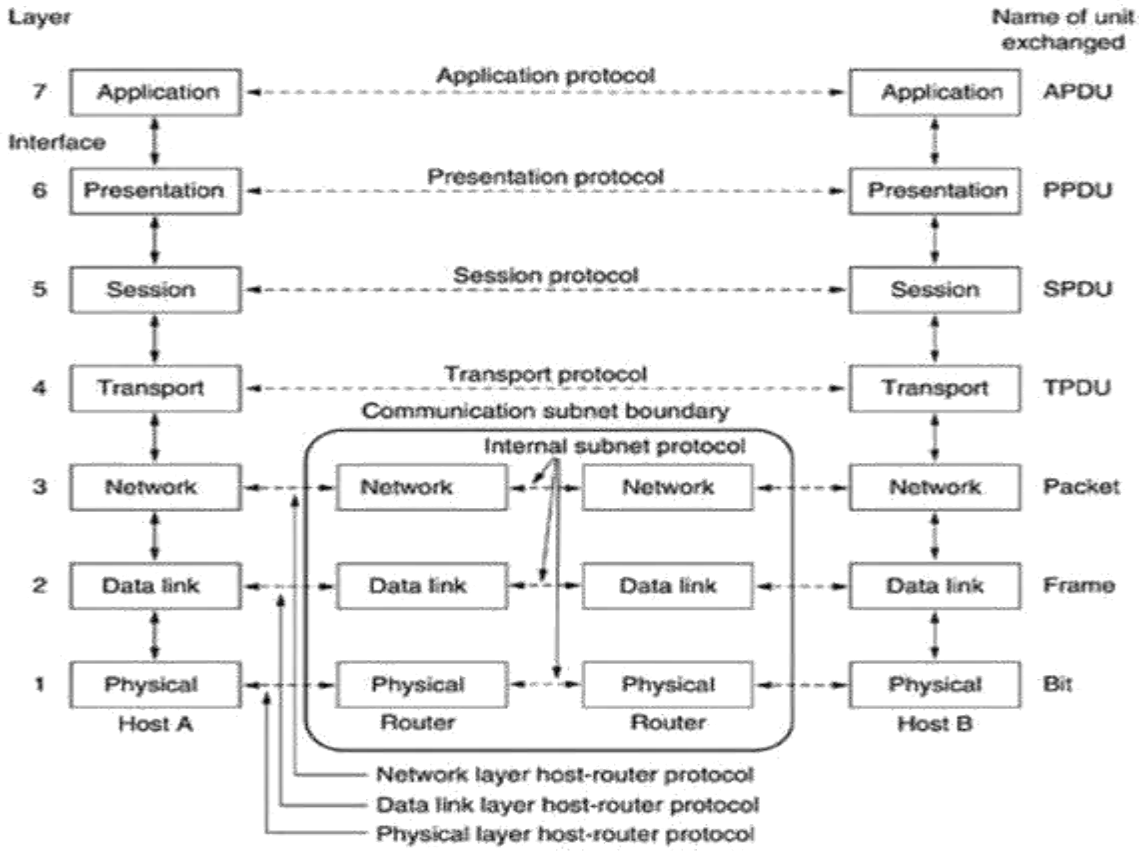


Fig.4: The OSI reference model

The Physical Layer:

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

The Data Link Layer:

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

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

The Network Layer:

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

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

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

The Transport Layer:

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

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

The Session Layer:

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

The Presentation Layer:

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

The Application Layer:

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

The TCP/IP Reference Model:

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

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

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

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

3. Transport Layer

4. Application Layer

Application Layer

Transport Layer

Internet Layer Host-to-

Network Layer

Host-to-Network Layer:

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

Internet Layer:

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

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

The Transport Layer:

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

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

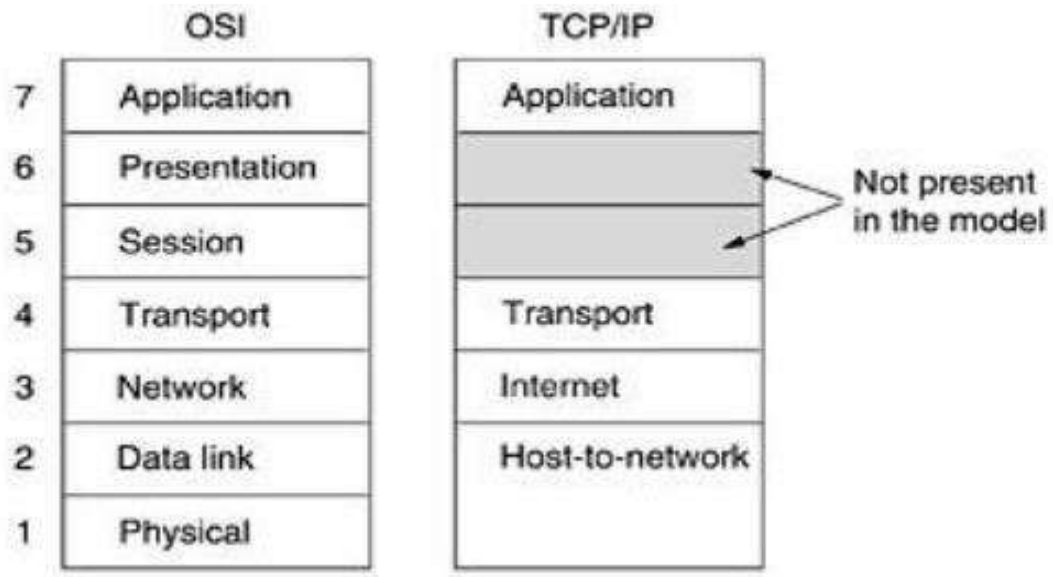


Fig.1: The TCP/IP reference model.

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

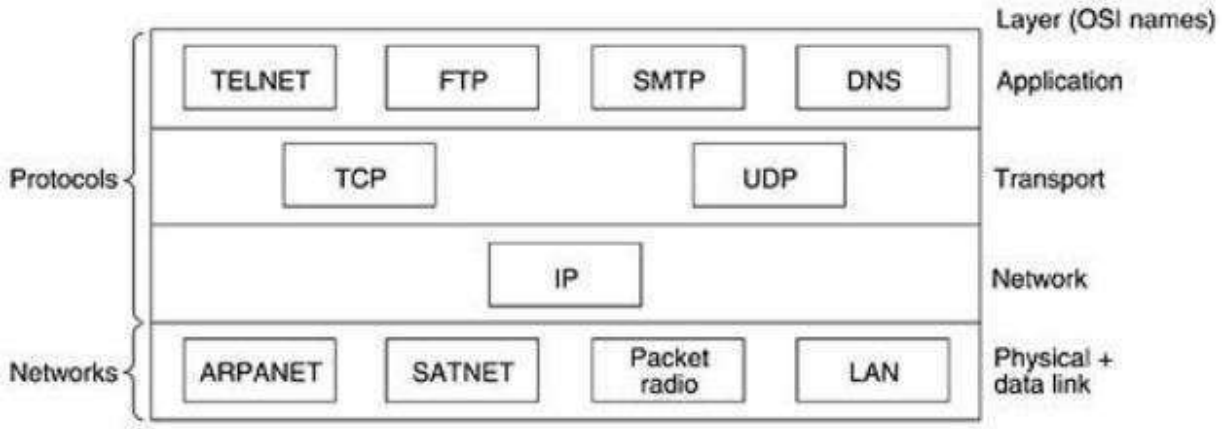


Fig.2: Protocols and networks in the TCP/IP model initially.

The Application Layer:

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

Comparison of the OSI and TCP/IP Reference Models:

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

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

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

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

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

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

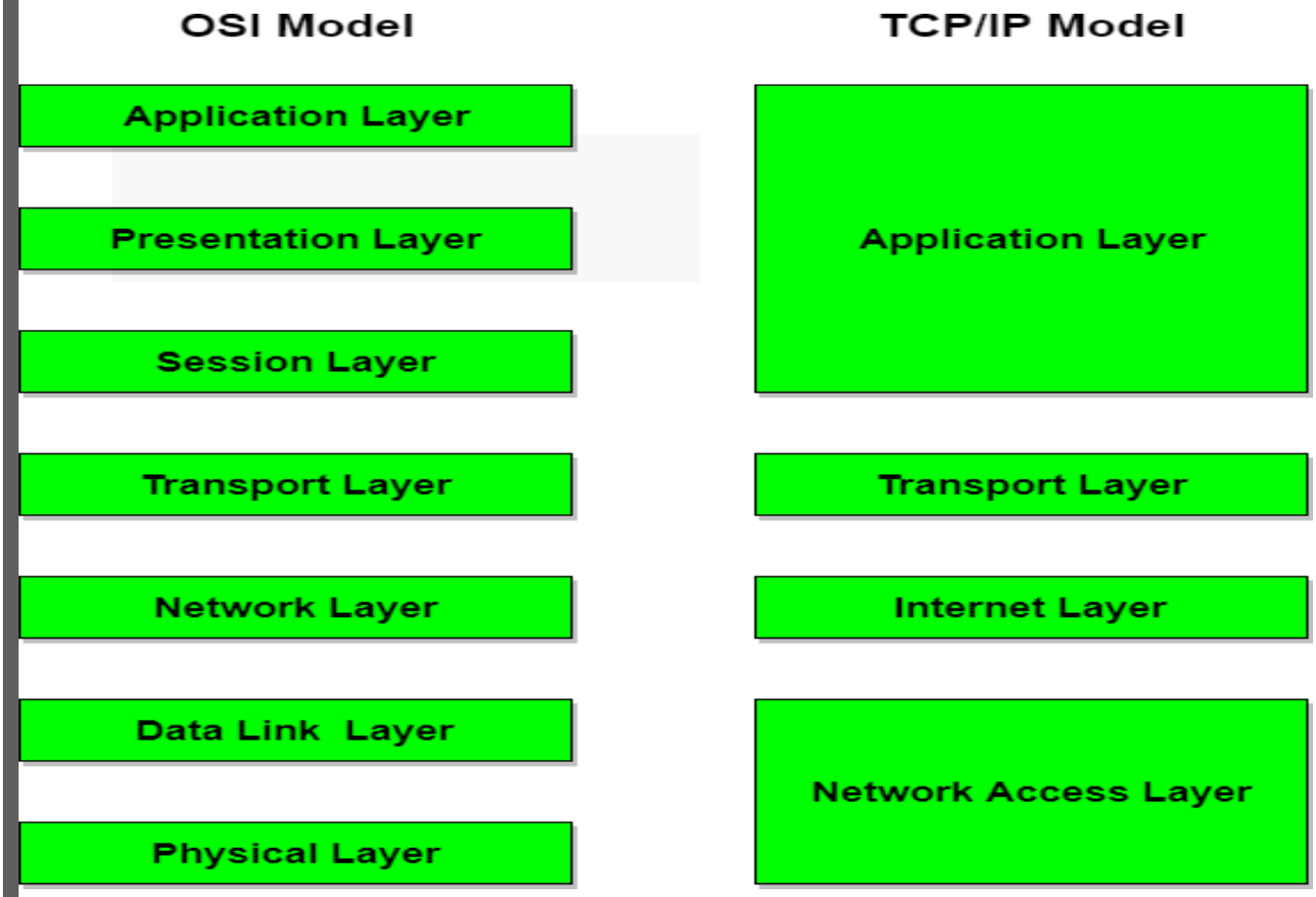
.

Comparison between OSI Reference Model and TCP/IP Reference Model

OSI(Open System Interconnection)	TCP/IP(Transmission Control Protocol / Internet Protocol)
1. OSI is a generic, protocol independent standard, acting as a communication gateway between the network and end user.	1. TCP/IP model is based on standard protocols around which the Internet has developed. It is a communication protocol, which allows connection of hosts over a network.
2. In OSI model the transport layer guarantees the delivery of packets.	2. In TCP/IP model the transport layer does not guarantees delivery of packets. Still the TCP/IP model is more reliable.
3. Follows vertical approach.	3. Follows horizontal approach.
4. OSI model has a separate Presentation layer and Session layer.	4. TCP/IP does not have a separate Presentation layer or Session layer.
5. Transport Layer is Connection Oriented.	5. Transport Layer is both Connection Oriented and Connection less.
6. Network Layer is both Connection Oriented and Connection less.	6. Network Layer is Connection less.

7. OSI is a reference model around which the networks are built. Generally it is used as a guidance tool.	7. TCP/IP model is, in a way implementation of the OSI model.
8. Network layer of OSI model provides both connection oriented and connectionless service.	8. The Network layer in TCP/IP model provides connectionless service.
9. OSI model has a problem of fitting the protocols into the model.	9. TCP/IP model does not fit any protocol
10. Protocols are hidden in OSI model and are easily replaced as the technology changes.	10. In TCP/IP replacing protocol is not easy.

Diagrammatic Comparison between OSI Reference Model and TCP/IP Reference Model



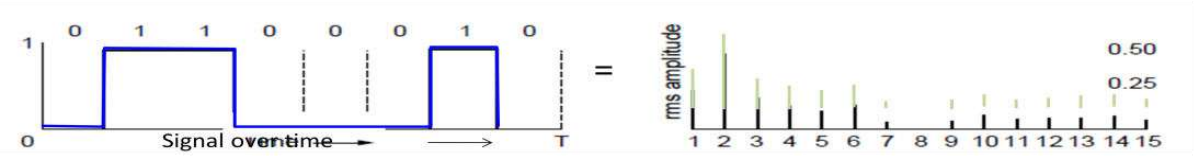
UNIT-II

Physical Layer – Fourier Analysis – Bandwidth Limited Signals – The Maximum Data Rate of a Channel - **Guided Transmission Media**, Digital Modulation and Multiplexing: **Frequency Division Multiplexing**, **Time Division Multiplexing**, Code Division Multiplexing **Data Link Layer Design Issues**, **Error Detection and Correction,** Elementary Data Link Protocols, **Sliding Window Protocols**

Fourier Analysis

- A time-varying signal can be equivalently represented as a series of frequency components (harmonics) or the sum of sines and cosines:

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft)$$

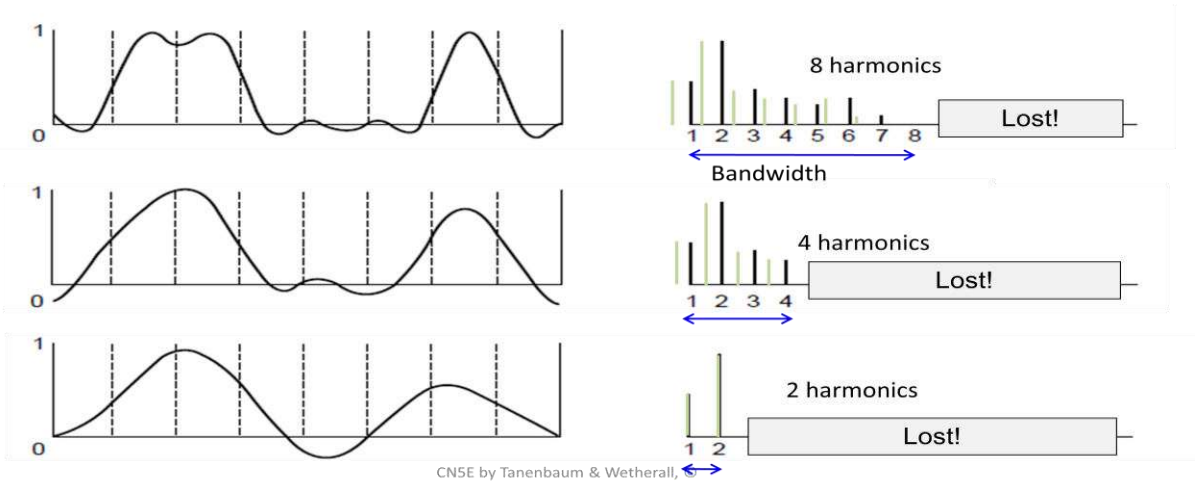


- Transmitted signals lose some power as they are transmitted.
- For a wire, amplitudes are transmitted mostly undiminished from 0 up to some frequency f. Frequencies above this frequency f are attenuated (reduced).
- The width of this frequency range is called the **bandwidth**.
    - Baseband run from 0 to some max frequency
    - Passband- shifted to occupy higher frequencies such as wireless

Bandwidth

- Bandwidth is a physical property of the transmission medium such as the construction, thickness and length of the wire or fiber.
- Limiting the bandwidth, limits the data rate.
- Goal for digital transmission is to receive a signal with enough fidelity to reconstruct the sequence of bits that was sent.
- Bandwidth – to Electrical Engineers –(analog) means – a quantity measured in Hz (cycles per second)
- Bandwidth – to Computer Scientists – (digital) means – the maximum data rate of a channel (bits/second)

Bandwidth-Limited Signals



Maximum Data Rate of a Channel

- Nyquist's theorem relates the data rate to the bandwidth (B) and number of signal levels (V):

$$\text{Max. data rate} = 2B \log_2 V \text{ bits/sec}$$

- Shannon's theorem relates the data rate to the bandwidth (B) and signal strength (S) relative to the noise (N):

$$\text{Max. data rate} = B \log_2(1 + S/N) \text{ bits/sec}$$

How fast signal can change

How many levels can be seen

Types of Transmission Medias

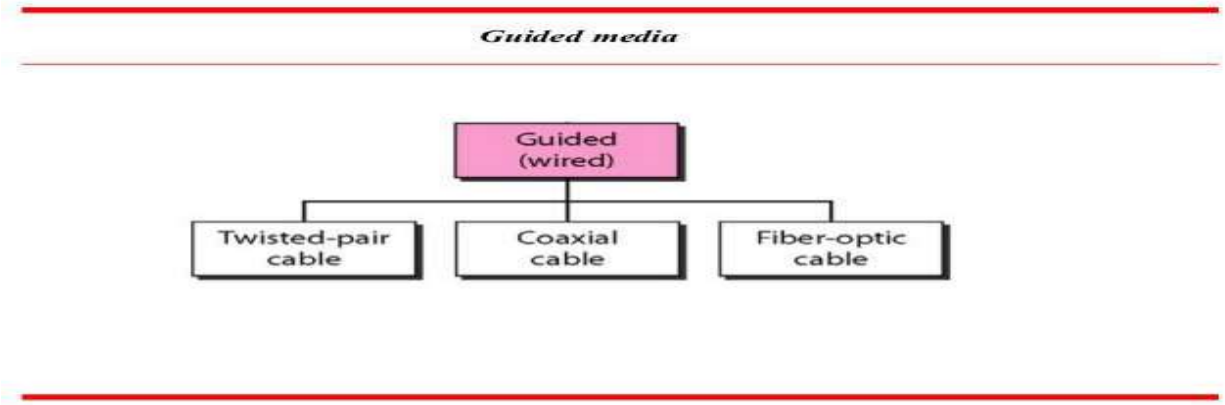
1. Guided Transmission Media

2. Unguided Transmission Media

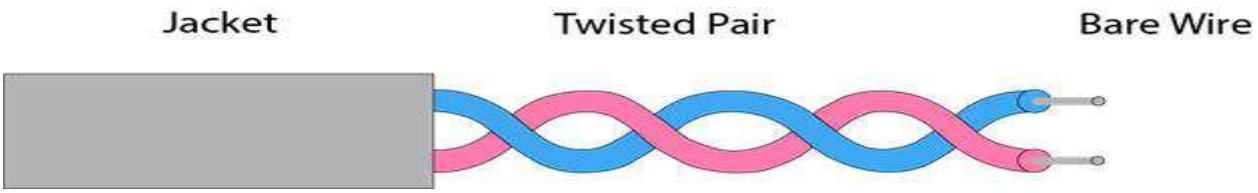
Guided Transmission Media :

It is defined as the physical medium through which the signals are transmitted. It is also known as Bounded media

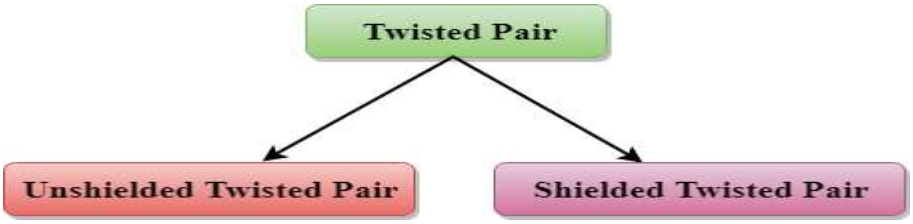
There are several types of cable which are commonly used with LANs. In some cases, a network will utilize only one type of cable, other networks will use a variety of cable types. The type of cable chosen for a network is related to the network's topology, protocol, and size. Understanding the characteristics of different types of cable and how they relate to other aspects of a network is necessary for the development of a successful network.



1. **Twisted Pair Cable :** Twisted pair is a physical media made up of a pair of cables twisted with each other. A twisted pair cable is cheap as compared to other transmission media. Installation of the twisted pair cable is easy, and it is a lightweight cable. The frequency range for twisted pair cable is from 0 to 3.5KHz.



Types of Twisted pair:



- Unshielded Twisted Pair (UTP) Cable
- Shielded Twisted Pair (STP) Cable

Unshielded Twisted Pair (UTP) Cable

Twisted pair cabling comes in two varieties: shielded and unshielded. Unshielded twisted pair (UTP) is the most popular and is generally the best option for school networks.

Figure 2.1 Unshielded Twisted Pair

The quality of UTP may vary from telephone-grade wire to extremely high-speed cable. The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other electrical devices. The tighter the twisting, the higher the supported transmission rate and the greater the cost per foot. The EIA/TIA (Electronic Industry Association / Telecommunication Industry Association) has established standards of UTP and rated five categories of wire.

Type	Use
Category 1	Voice Only (Telephone Wire)
Category 2	Data to 4 Mbps (LocalTalk)
Category 3	Data to 10 Mbps (Ethernet)
Category 4	Data to 20 Mbps (16 Mbps Token Ring)
Category 5	Data to 100 Mbps (Fast Ethernet)

Table Categories of Unshielded Twisted Pair

Advantages Of Unshielded Twisted Pair:

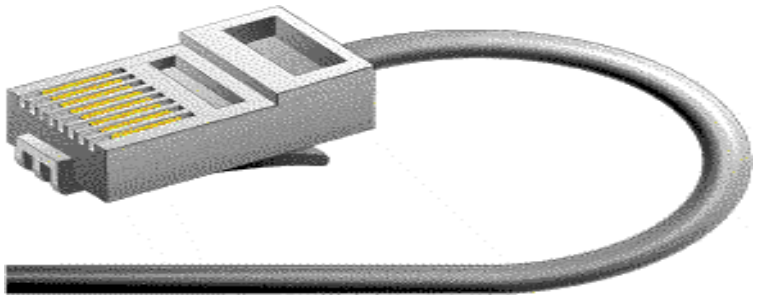
- It is cheap.
- Installation of the unshielded twisted pair is easy.
- It can be used for high-speed LAN.

Disadvantage:

- This cable can only be used for shorter distances because of attenuation.

Unshielded Twisted Pair Connector

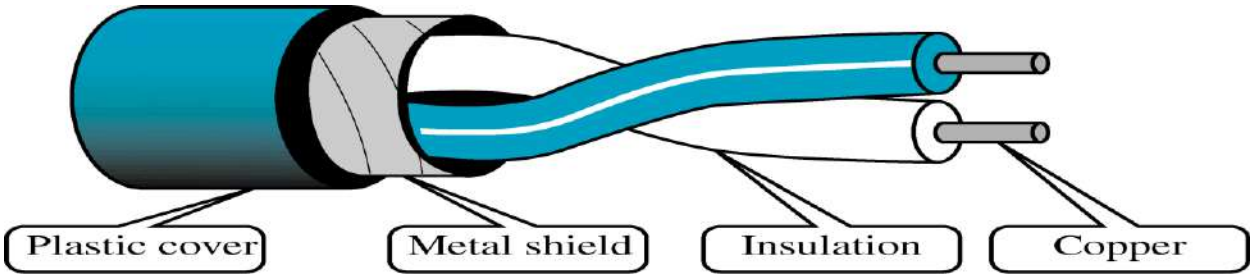
The standard connector for unshielded twisted pair cabling is an RJ-45 connector. This is a plastic connector that looks like a large telephone-style connector (See fig. 2.2). a slot allows the RJ-45 to be inserted only one way. RJ stands for Registered Jack, implying that the connector follows a standard borrowed from the telephone industry. This standard designates which wire goes with each pin inside the connector.



Shielded Twisted Pair (STP) Cable :

A shielded twisted pair is a cable that contains the mesh surrounding the wire that allows the higher transmission rate.

A disadvantage of UTP is that it may be susceptible to radio and electrical frequency interference. Shielded twisted pair |(STP) is suitable for environments with electrical interference; however, the extra shielding can make the cables quite bulky. Shielded twisted pair is often used on networks using Token Ring topology.



Characteristics Of Shielded Twisted Pair:

- **The cost of the shielded twisted pair cable is not very high and not very low.**
- **An installation of STP is easy.**
- **It has higher capacity as compared to unshielded twisted pair cable.**
- **It has a higher attenuation.**
- **It is shielded that provides the higher data transmission rate.**

Disadvantages

- **It is more expensive as compared to UTP and coaxial cable.**
- **It has a higher attenuation rate.**

2. Coaxial Cable

Coaxial cable is very commonly used transmission media, for example, TV wire is usually a coaxial cable.

Coaxial cabling has a single copper conductor at its center. A plastic layer provides insulation between the center conductor and the braided metal shield (See fig. 3). The metal shield helps to block any outside interference from fluorescent lights, motors, and other computers.

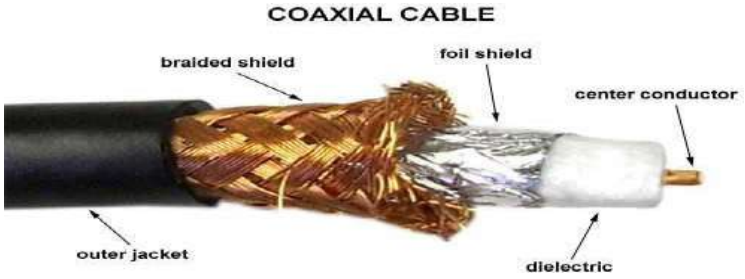
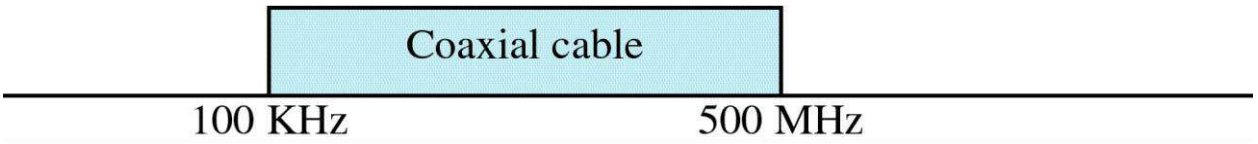


Fig. 2.3 Coaxial cable



- The name of the cable is coaxial as it contains two conductors parallel to each other.
- It has a higher frequency as compared to Twisted pair cable.
- The inner conductor of the coaxial cable is made up of copper, and the outer conductor is made up of copper mesh. The middle core is made up of non-conductive cover that separates the inner conductor from the outer conductor.

- Thin coaxial cable is also referred to as thinnet. 10base2 refers to the specifications for thin coaxial cable carrying Ethernet signals. The 2 refers to the approximate maximum segment length being 200 meters. In actual fact the maximum segment length is 185 meters. Thin coaxial cable is popular in school networks, especially linear bus networks.

- Thick coaxial cable is also referred to as thicknet. 10base refers to the specifications for thick coaxial cable carrying Ethernet signals. The 5 refers to the maximum segment length being 500 meters. Thick coaxial cable has an extra protective plastic cover that helps keep moisture away from the center conductor. This makes thick coaxial a great choice when running longer lengths in a linear bus network. One disadvantage of thick coaxial is that it does not bend easily and is difficult to install.

Types of Coaxial cable

Baseband transmission: It is defined as the process of transmitting a single signal at high speed.
Broadband transmission: It is defined as the process of transmitting multiple signals simultaneously.

Advantages Of Coaxial cable:
- The data can be transmitted at high speed.
- It has better shielding as compared to twisted pair cable.
- It provides higher bandwidth.

Disadvantages Of Coaxial cable:
- It is more expensive as compared to twisted pair cable.

If any fault occurs in the cable causes the failure in the entire network.

Fiber Optic Cable :

- Fiber optic cable is a cable that uses electrical signals for communication.

- Fiber optic is a cable that holds the optical fibers coated in plastic that are used to send the data by pulses of light.

- The plastic coating protects the optical fibers from heat, cold, electromagnetic interference from other types of wiring.

Fiber optics provide faster data transmission than copper wires

Fiber optic cabling consists of a center glass core surrounded by several layers of protective materials. It transmits light rather than electronic signals eliminating the problem of electrical interference.

This makes it ideal for certain environments that contain a large amount of electrical interference. It has also made it the standard for connecting networks between buildings, due to its immunity to the effects of moisture and lighting.

Fiber optic cable has the ability to transmit signals over much longer distances than coaxial and twisted pair. It also has made it the standard for connecting networks between buildings, due to its immunity to the effects of moisture and lighting.

Fiber optic cable has the ability to transmit signals over mush longer distances than coaxial and twisted pair. It also has the capability to carry information at vastly greater speeds. This capacity broadens communication possibilities to include services such as video conferencing and interactive services. The cost of fiber optic cabling is comparable to copper cabling; however it is more difficult to install and modify. 10BaseF refers to the specifications for fiber optic cable carrying Ethernet signals.

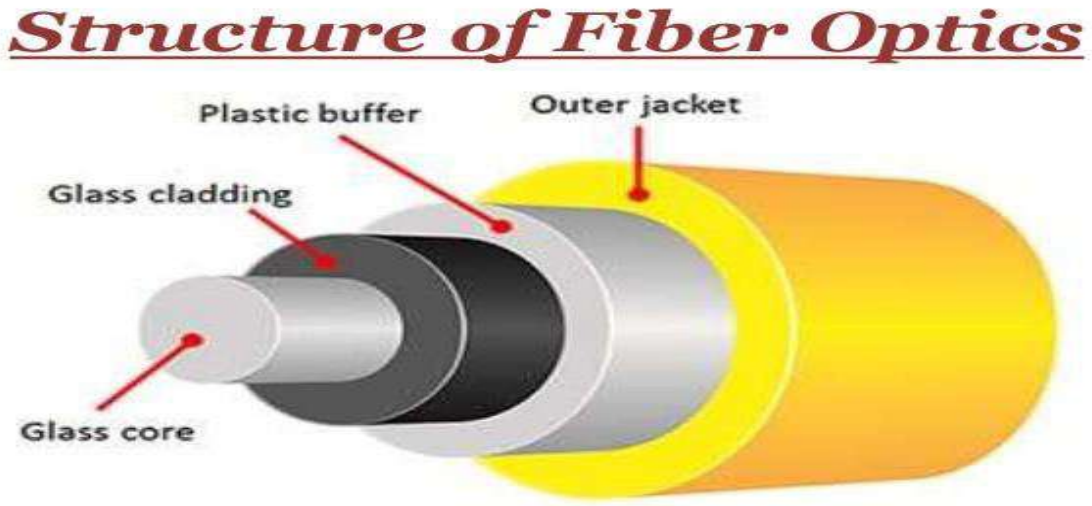


Fig-1. Fiber Optic Cable

Facts about fiber optic cables:

- Outer insulating jacket is made of Teflon or PVC.
- Kevlar fiber helps to strengthen the cable and prevent breakage.
- A plastic coating is used to cushion the fiber center.
- Center (core) is made of glass or plastic fibers.

Fiber Optic Connector

The most common connector used with fiber optic cable is an ST connector. It is barrel shaped, similar to a BNC connector. A newer connector, the SC, is becoming more popular. It has a squared face and is easier to connect in a confined space.

Specification	Cable Type	Maximum length
10BaseT	**Unshielded Twisted Pair**	**100 meters**
10Base2	**Thin Coaxial**	**185 meters**
10Base5	**Thick Coaxial**	**500 meters**
10BaseF	**Fiber Optic**	**2000 meters**
100BaseT	**Unshielded Twisted Pair**	**100 meters**
100BaseTX	**Unshielded Twisted Pair**	**220 meters**

Table : Ethernet Cable Summary

Unguided Transmission Media:

Unguided transmission media is data signals that flow through the air. They are not guided or bound to a channel to follow.

Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication. Signals are normally broadcast through free space and thus are available to anyone who has a device receiving them. Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation.

In **ground propagation**, radio waves travel through the lowest portion of the atmosphere, hugging the earth. These low-frequency signals emanate in all directions from the transmitting antenna and follow the curvature of the planet. Distance depends on the amount of power in the signal: *The greater the power, the greater the distance.* Ground waves have carrier frequencies up to 2 MHz. AM radio is an example of ground wave propagation.

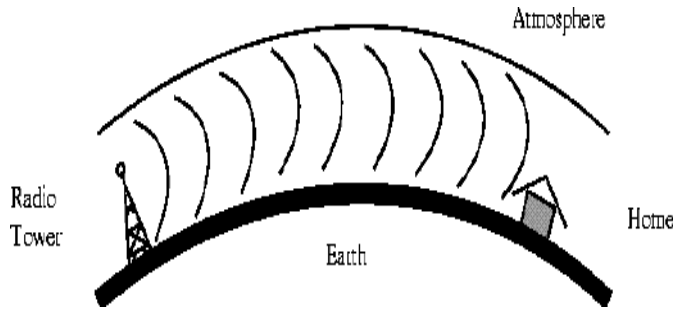


Fig. 2.6 Ground Wave Propagation

In **sky propagation**, higher frequency radio waves radiate upward into the ionosphere (the layer of atmosphere where the particles exist as ions) where they are reflected back to the earth. This type of transmission allows for greater distances with lower output power.

It is sometimes called double hop propagation. It operates in the frequency range of 30 – 85 MHz. Because it depends on the earth's ionosphere, it changes with the weather and time of day. The signal bounces off of the ionosphere and back to the earth. Ham radios operate in this range. Other books called this **Ionospheric propagation**.

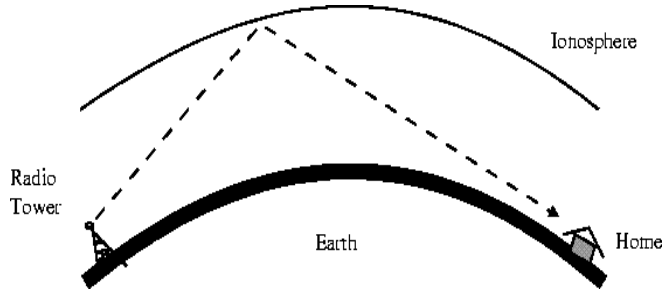


Fig. 2.7 Ionospheric Propagation

In **line-of-sight propagation**, very high-frequency signals are transmitted in straight lines directly from antenna to antenna. Antennas must be directional, facing each other and either tall enough or close enough together not to be affected by the curvature the earth. Line-of-sight propagation is tricky because radio transmission cannot be completely focused.

It is sometimes called space waves or tropospheric propagation. It is limited by the curvature of the earth for ground-based stations (100 km, from horizon to horizon). Reflected waves can cause promlems. Axamples are: FM radio, microwave and satellite.

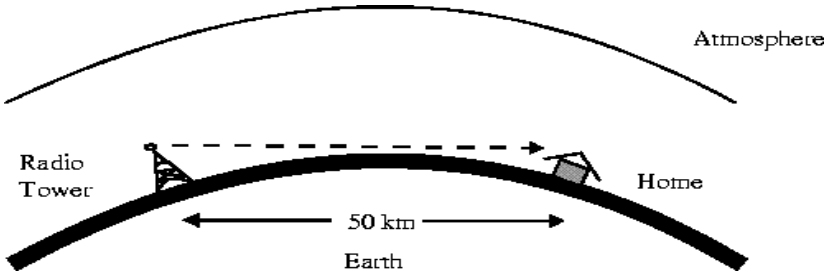
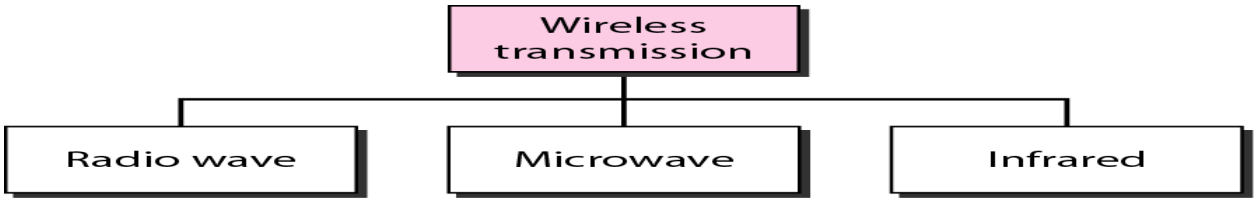


Fig. 2.8 Line-of-sight Propagation

We can divide wireless transmission into **three broad groups**: radio waves, microwaves, and infrared waves.



1. Radio Waves

Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves.

Radio waves are omnidirectional. When antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves that can be received by any receiving antenna.

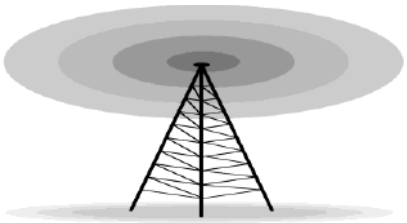


Fig: Omnidirectional antenna

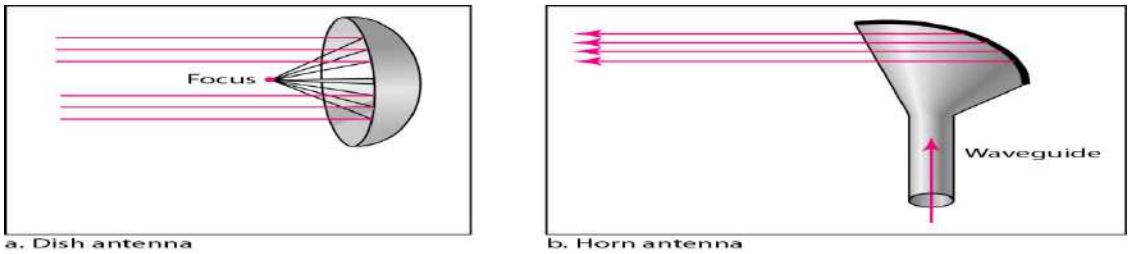
The omnidirectional property has a disadvantage too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.

Radio waves, particularly those of low and medium frequencies, can penetrate walls. This characteristic can be both an advantage and disadvantage. It is an advantage because, for example, an AM radio can receive signals inside a building. It is a disadvantage because we cannot isolate a communication to just inside or outside a building.

2. Microwaves

Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves.

Microwaves are unidirectional. When an antenna transmits microwave waves, they can be narrowly focused. This means that the sending and receiving antennas need to be aligned. The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas. The following describes some characteristics of microwave propagation:

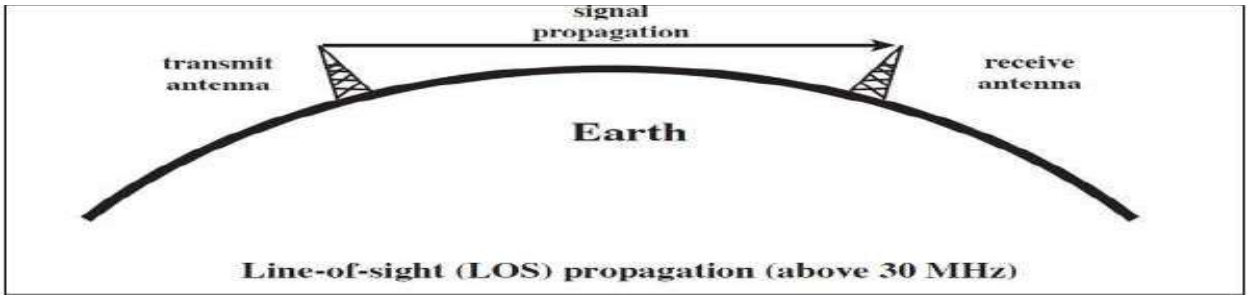


a. Dish antenna b. Horn antenna

- Microwave propagation is line-of-sight. Since towers with the mounted antennas need to be in direct sight of each other. This also set a limit on the distance between stations depending on the local geography. Towers that are far apart need to be very tall. The curvature of the earth as well as other blocking obstacles does not allow two short towers to communicate by using microwaves. Typically the line of sight due to the Earth's curvature is only 50 km to the horizon. Repeaters are often needed for long-distance communication.
- Very high frequency microwaves cannot penetrate walls. This characteristic can be a disadvantage if receivers are inside the buildings.
- The microwave band is relatively wide, almost 299 GHz. Therefore wider subbands can be assigned, and a high data rate is possible.
- Use of certain portions of the band requires permission from authorities.

3. Infrared Waves

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 mm), can be used for short-range communication. Infrared waves, having high frequencies, cannot penetrate walls. This advantageous characteristic prevents interference between one



system and another; a short-range communication system in one room cannot be affected by another system in the next room. When we use our infrared remote control, we do not

interfere with the use of the remote of our neighbors. However, this same characteristic makes infrared signals useless for long-range communication. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

Satellite

Satellites are transponders (units that receive on one frequency and retransmit on another) that are set in geostationary orbits directly over the equator. These geostationary orbits are 36, 000 km from the Earths's surface. At this point, the gravitational pull of the Earth and the centrifugal force of Earth's rotation are balanced and cancel each other out. Centrifugal force is the rotational force placed on the satellite that wants to fling it out into the space.

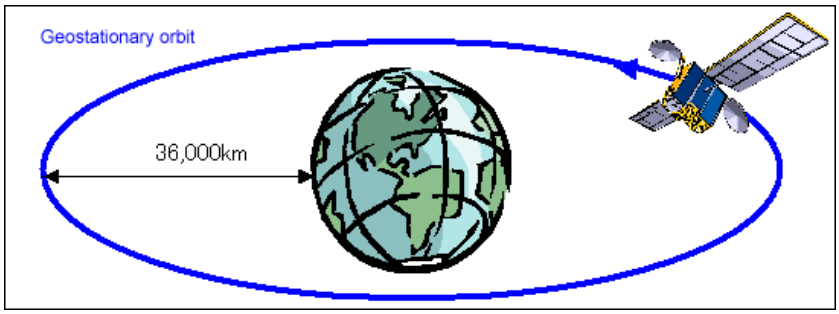


Fig. Satellite Communication

The uplink is the transmitter of data to the satellite. The downlink is the receiver of data. Uplinks and downlinks are also called Earth stations because they are located on the Earth. The footprint is the "shadow" that the satellite can transmit to, the shadow being the area that can receive the satellite's transmitted signal.

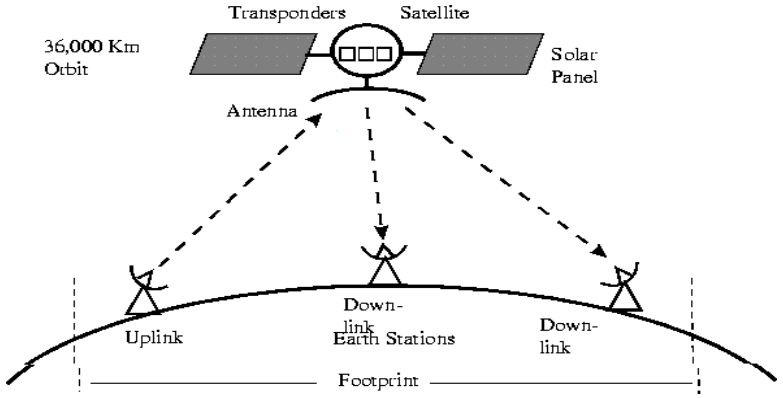
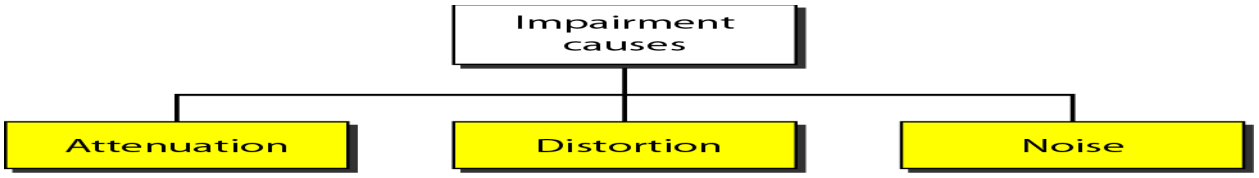


Fig. Uplink and Downlink

Impairments: It means that signals that are transmitted at the beginning of the medium are not the same as the signals that are received at the end of the medium that is what is sent is not what is received.



Attenuation:

- Means loss of energy -> weaker signal
- When a signal travels through a medium it loses energy overcoming the resistance of the medium
- Amplifiers are used to compensate for this loss of energy by amplifying the signal.

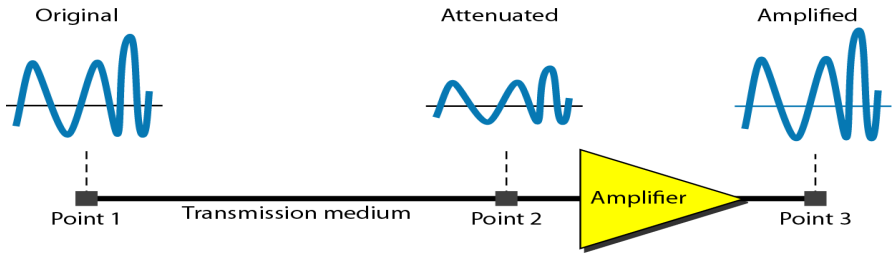


Fig. ***Example for Attenuation***

Distortion:

- Means that the signal changes its form or shape
- Distortion occurs in composite signals
- Each frequency component has its own propagation speed traveling through a medium.
- The different components therefore arrive with different delays at the receiver.
- That means that the signals have different phases at the receiver than they did at the source.

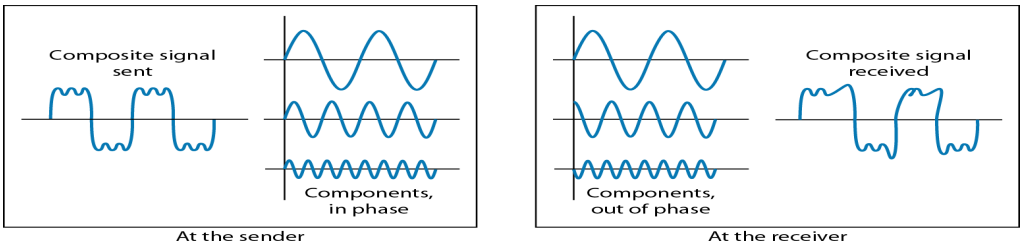
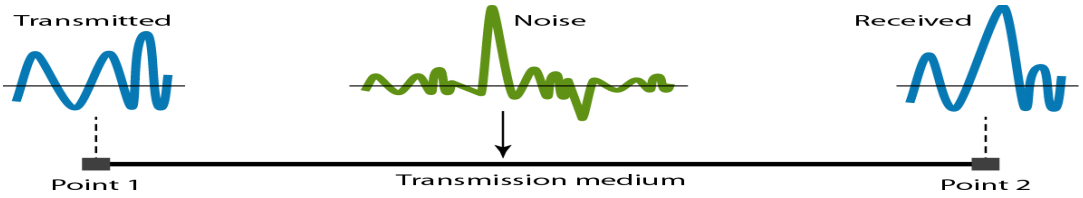


Fig: Distortion

Noise:

- Thermal - random noise of electrons in the wire creates an extra signal
- Crosstalk -Crosstalk is the transmission of signals and noise due to coupling between lines, and is also called interference.
- Impulse - Spikes that result from power lines etc.
-



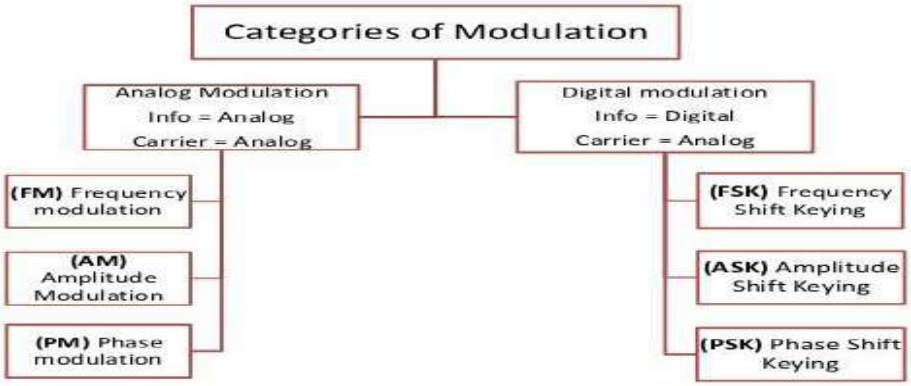
Modulation:

- Modulation is a process of suppressing low frequency information signal on a high frequency carrier signal.

OR

Modulation is a process of modifying the any of the characteristics (amplitude, frequency, phase) of high frequency carrier in accordance with low frequency information signal.

6.2 Categories of Modulation



Digital Modulation Techniques :

Digital-to-Analog signals is the next conversion we will discuss in this chapter. These techniques are also called as **Digital Modulation techniques**.

Digital Modulation Digital modulation is defined as the modulation process in which discrete signals are used for modulating carrier waves and it is used for removing noise from the waves.

There are many types of digital modulation techniques and also their combinations, depending upon the need. Of them all, we will discuss the prominent ones.

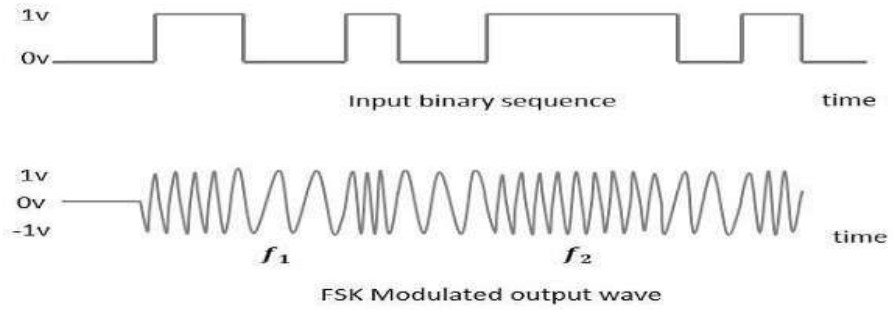
Types of **Digital Modulation**:

There are three types of digital modulation, and they are:

Amplitude shift key (ASK)

Frequency shift key (FSK)

Phase shift key (PSK)



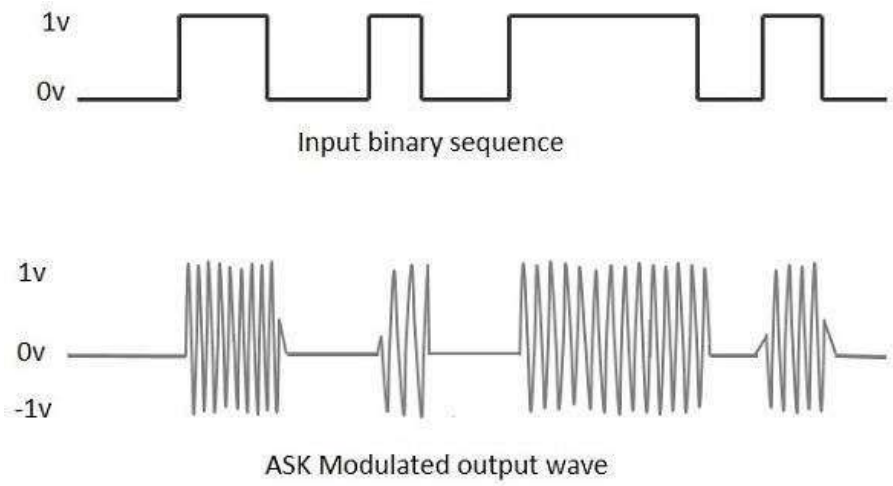
- ASK – Amplitude Shift Keying

The amplitude of the resultant output depends upon the input data whether it should be a zero level or a variation of positive and negative, depending upon the carrier frequency.

Amplitude Shift Keying (ASK) is a type of Amplitude Modulation which represents the binary data in the form of variations in the amplitude of a signal.

- Any modulated signal has a high frequency carrier. The binary signal when ASK modulated, gives a **zero** value for **Low** input while it gives the **carrier output** for **High** input.

- The following figure represents ASK modulated waveform along with its input.



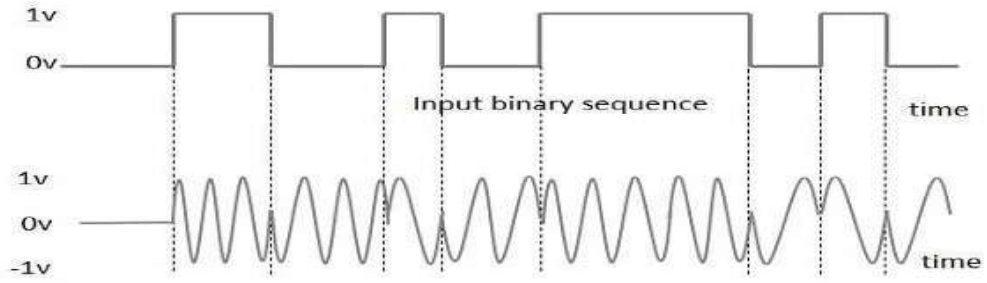
Frequency Shift Keying (FSK) is the digital modulation technique in which the frequency of the carrier signal varies according to the digital signal changes. FSK is a scheme of frequency modulation.

The output of a FSK modulated wave is high in frequency for a binary High input and is low in frequency for a binary Low input. The binary **1s** and **0s** are called Mark and Space frequencies.

The following image is the diagrammatic representation of FSK modulated waveform along with its input.

Phase Shift Keying (PSK) is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine and cosine inputs at a particular time. PSK technique is widely used for wireless LANs, bio-metric, contactless operations, along with RFID and Bluetooth communications.

Following is the diagrammatic representation of BPSK Modulated output wave along with its given input.



BPSK Modulated output wave

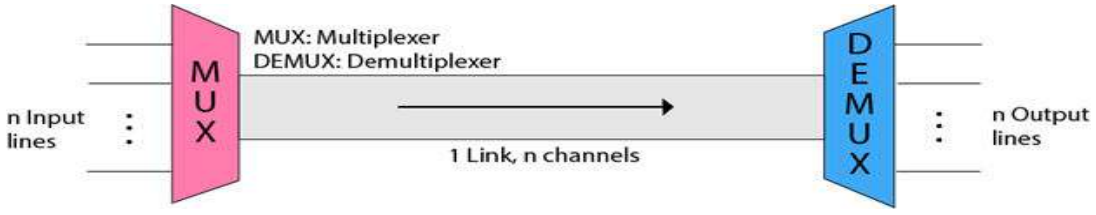
Multiplexing is a technique by which different analog and digital streams of transmission can be simultaneously processed over a shared link. Multiplexing divides the high capacity medium into low capacity logical medium which is then shared by different streams.

Communication is possible over the air (radio frequency), using a physical media (cable), and light (optical fiber). All mediums are capable of multiplexing.

When multiple senders try to send over a single medium, a device called Multiplexer divides the physical channel and allocates one to each. On the other end of communication, a De-multiplexer receives data from a single medium, identifies each, and sends to different receivers.

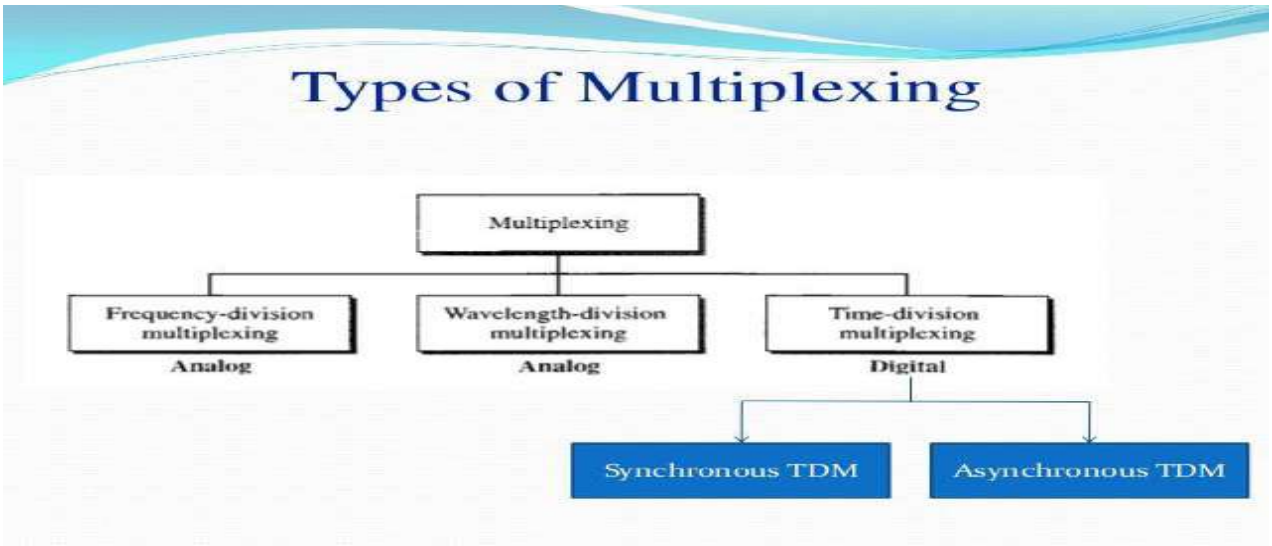
Multiplexing:

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



Advantages of Multiplexing:

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



1. Frequency Division Multiplexing:

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

When the carrier is frequency, FDM is used. FDM is an analog technology. FDM divides the spectrum or carrier bandwidth in logical channels and allocates one user to each channel. Each user can use the channel frequency independently and has exclusive access of it. All channels are divided in such a way that they do not overlap with each other. Channels are separated by guard bands. Guard band is a frequency which is not used by either channel.

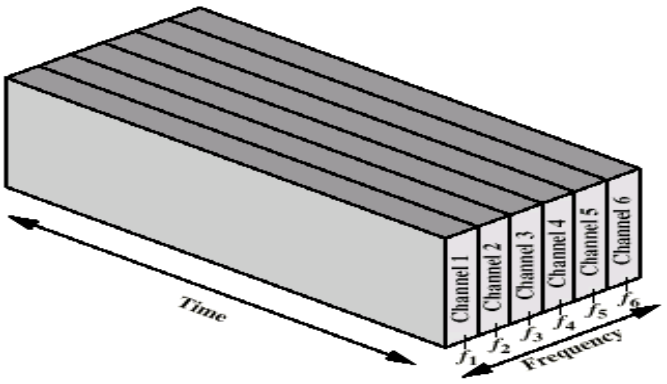
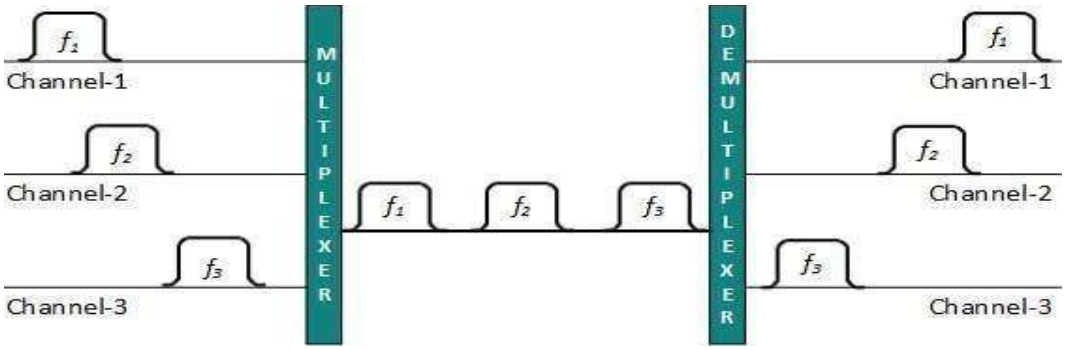
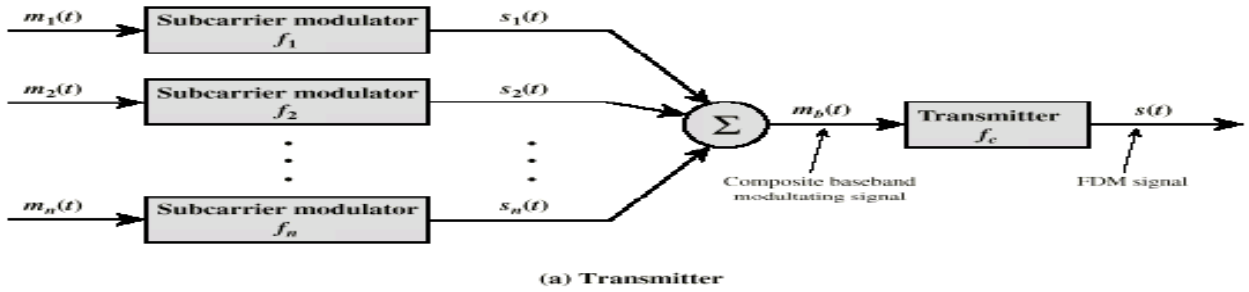
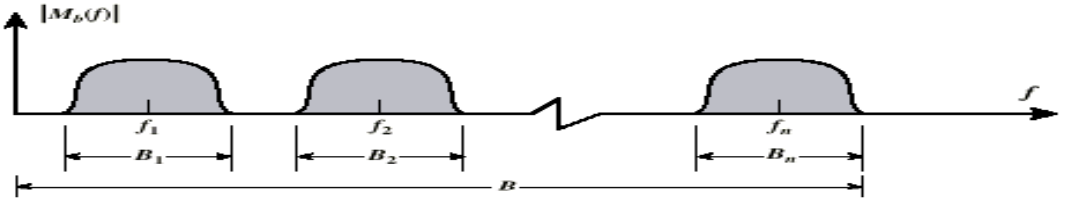


Fig: FDM System

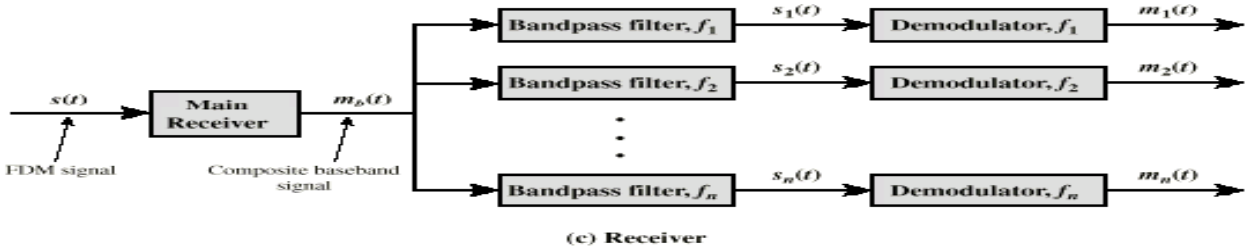




(a) Transmitter



(b) Spectrum of composite baseband modulating signal



(c) Receiver

2.Wavelength division multiplexing:

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

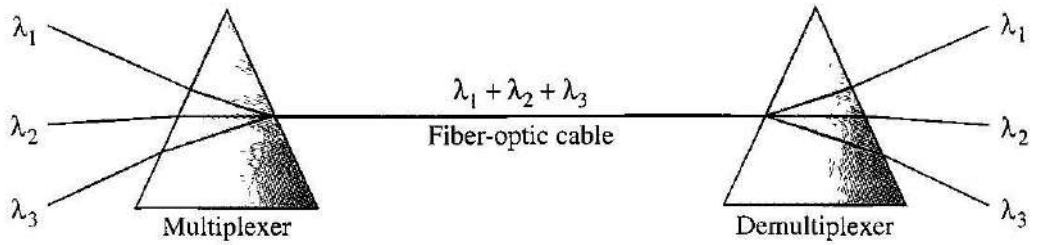
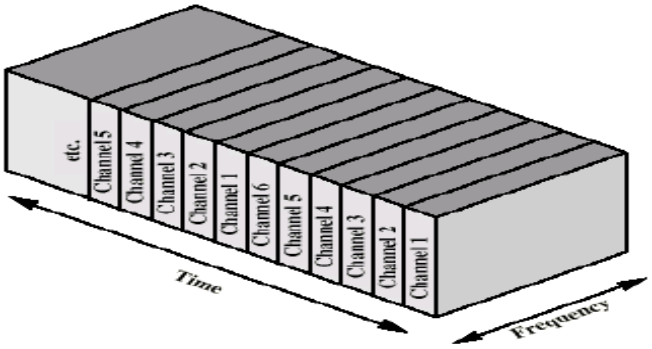


Fig : Wave Division Multiplexing

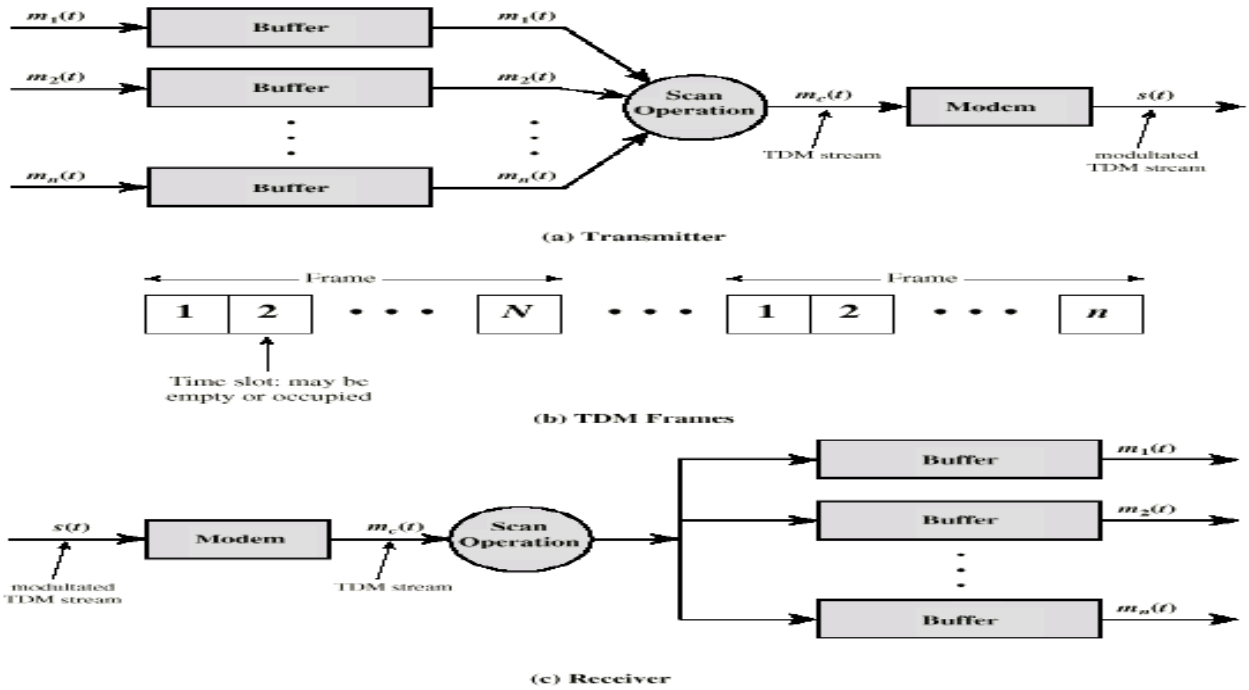
- Number of sources generating laser beams at different frequencies.
- Multiplexer consolidates sources for transmission over single fiber.
- Optical amplifiers amplify all wavelengths.
  -Typically tens of km apart
- Demux separates channels at the destination
- Mostly 1550nm wavelength range
- Same general architecture as other FDM
- Was 200MHz per channel
- Now 50GHz

Time Division Multiplexing:

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



TDM is applied primarily on digital signals but can be applied on analog signals as well. In TDM the shared channel is divided among its user by means of time slot. Each user can transmit data within the provided time slot only. Digital signals are divided in frames, equivalent to time slot i.e. frame of an optimal size which can be transmitted in given time slot.TDM works in synchronized mode. Both ends, i.e. Multiplexer and De-multiplexer are timely synchronized and both switch to next channel simultaneously.



Types of TDM :

- **Synchronous TDM**
- **Asynchronous TDM**

Synchronous TDM :

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

Asynchronous TDM (or) Statistical TDM:
The channel capacity cannot be fully utilized. Some of the slots go empty in certain frames

Statistical TDM:

- In Synchronous TDM many slots are wasted
- Statistical TDM allocates time slots dynamically based on demand
- Multiplexer scans input lines and collects data until frame full
- Data rate on line lower than aggregate rates of input lines

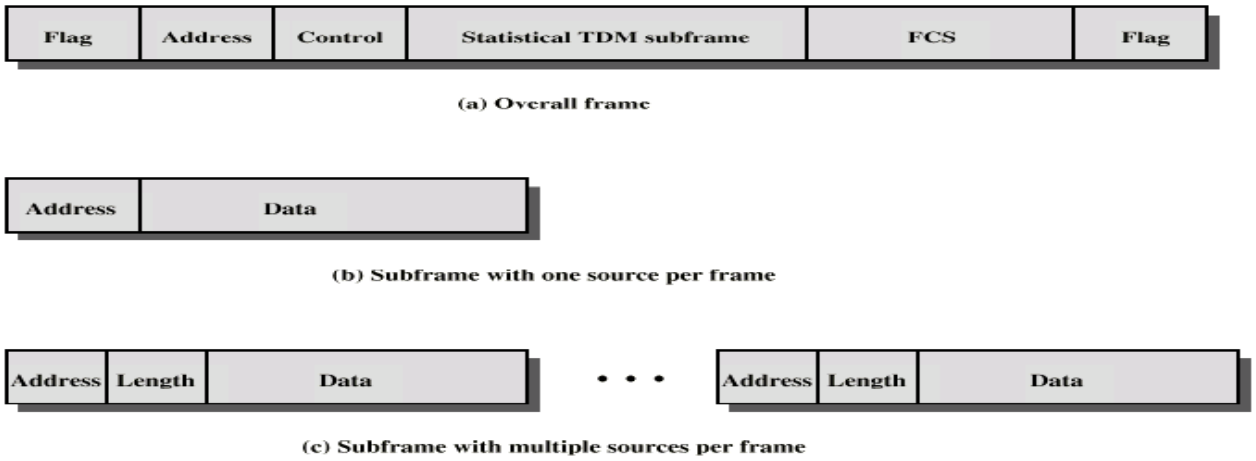
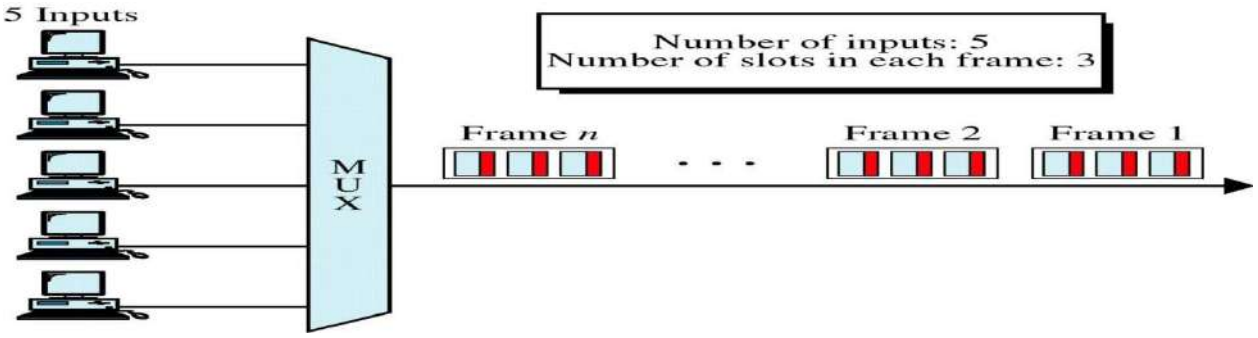


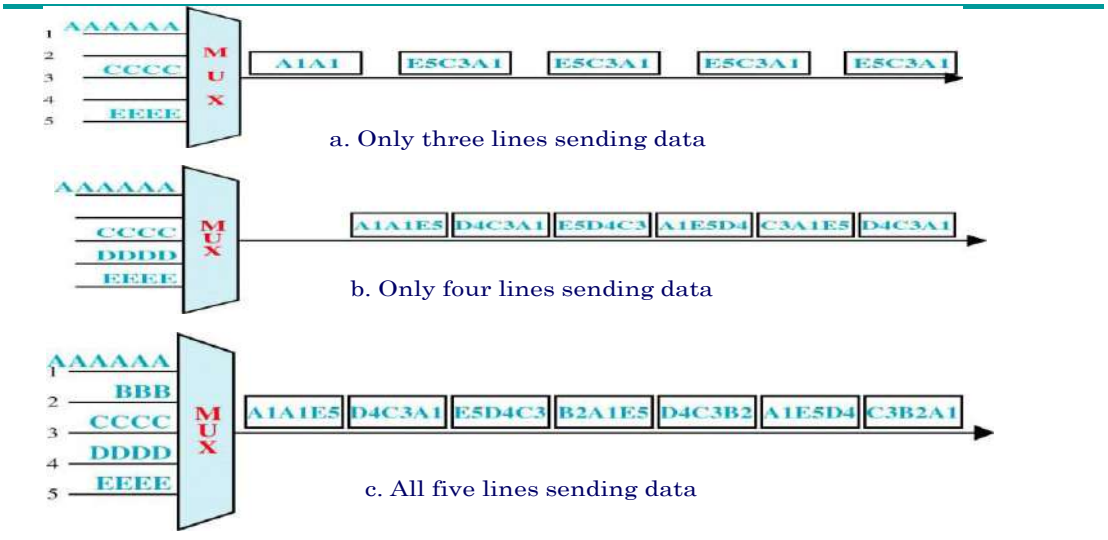
Fig: Statistical TDM Frame Formats



Asynchronous TDM

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

Frames and Addresses



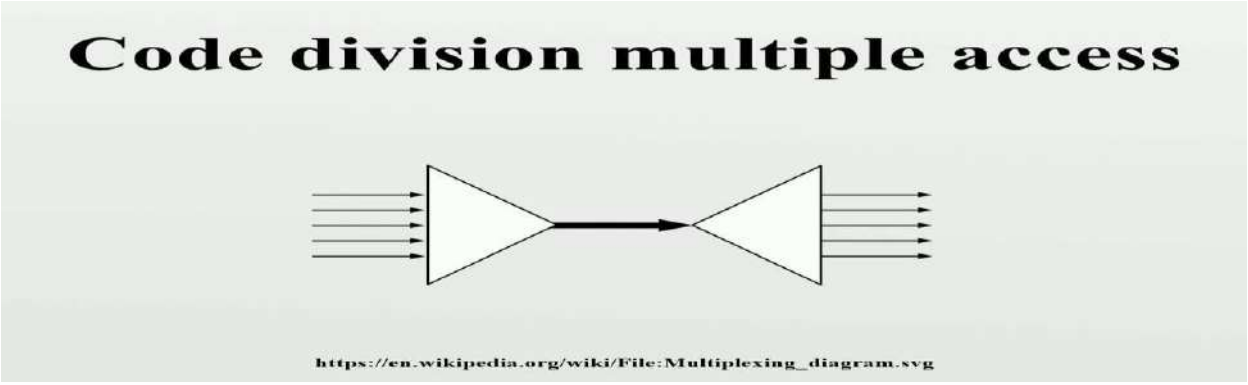
a. Only three lines sending data

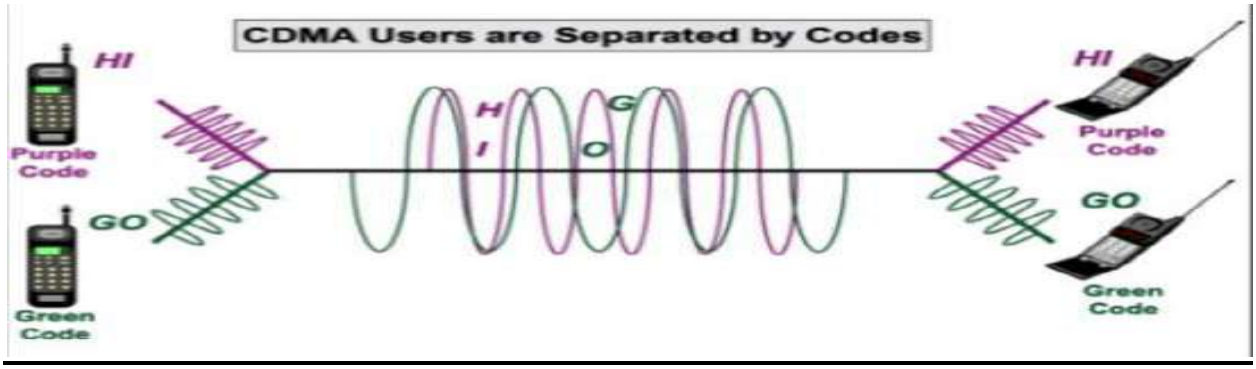
b. Only four lines sending data

c. All five lines sending data

Code Division Multiplexing (CDM) :

- Code division multiplexing (CDM) is a networking technique in which multiple data signals are combined for simultaneous transmission over a common frequency band.
- When CDM is used to allow multiple users to share a single communications channel, the technology is called code division multiple access (CDMA).





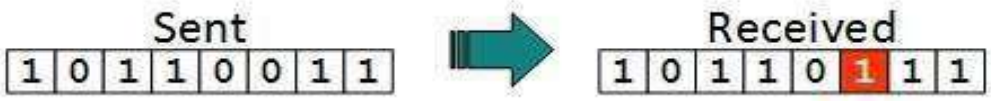
- CDM used in parts of the cellular telephone system and for some satellite communication
  - The specific version of CDM used in cell phones is known as Code Division Multi-Access (CDMA)

When CDM is used to allow multiple signals from multiple users to share a common communication channel, the technology is called Code Division Multiple Access (CDMA). Each group of users is given a shared code and individual conversations are encoded in a digital sequence. Data is available on the shared channel, but only those users associated with a particular code can access the data.

Error Detection and Correction :

Types of Errors : There may be three types of errors

- **Single bit error**



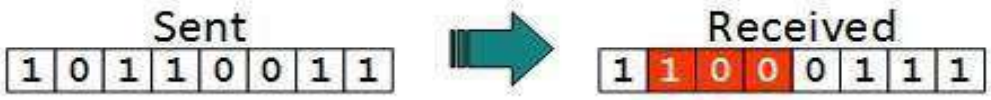
In a frame, there is only one bit, anywhere though, which is corrupt

.

- **Multiple bits error**

Frame is received with more than one bits in corrupted state.

- **Burst error**

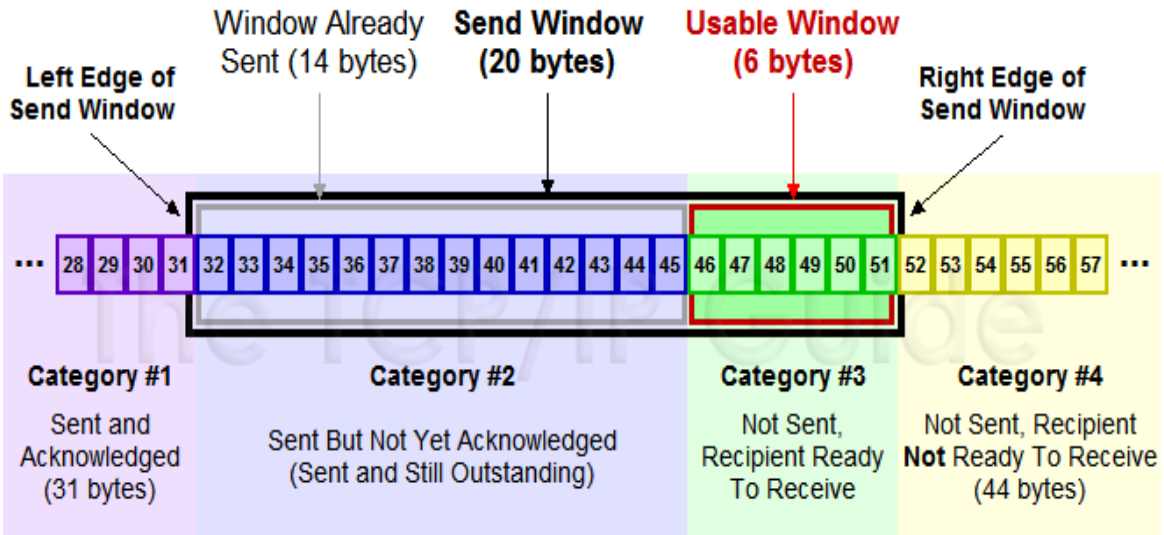


Frame contains more than1 consecutive bits corrupted.
Error control mechanism may involve two possible ways:

- **Error detection**
- **Error correction**

Sliding Window Protocols:

- **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



Computer Networks

UNIT – III

The Data Link Layer - Services Provided to the Network Layer – Framing – Error Control – Flow Control, Error Detection and Correction – Error-Correcting Codes – Error Detecting Codes, Elementary Data Link Protocols- A Utopian Simplex Protocol-A Simplex Stop and Wait Protocol for an Error free channel-A Simplex Stop and Wait Protocol for a Noisy Channel, Sliding Window Protocols-A One Bit Sliding Window Protocol-A Protocol Using Go-Back-NA Protocol Using Selective Repeat.

What is DLL (Data Link Layer)?

The Data Link Layer is the second layer in the OSI model, above the Physical Layer, which ensures that the error free data is transferred between the adjacent nodes in the network. It breaks the datagram passed down by above layers and converts them into frames ready for transfer. This is called **Framing.**

It provides two main functionalities

- Reliable data transfer service between two peer network layers
- Flow Control mechanism which regulates the flow of frames such that data congestion is not there at slow receivers due to fast senders.

THE DATA LINK LAYER DESIGN ISSUES

FUNCTIONS

- Providing a well-defined service interface to the network layer.
- Dealing with transmission errors.
- Regulating the flow of data so that slow receivers are not swamped by fast senders –flow control.

The two main functions of the data link layer are:

1. **Data Link Control (DLC)**: It deals with the design and procedures for communication b/w nodes: node-to-node communication.

2. **Media Access Control (MAC)**: It explains how to share the link.

1. DATA LINK CONTROL (DLC):

Data link control functions includes

(1) Framing.

(2) Error Control.

(3) Flow Control.

(1) FRAMING

The frame contains

1. Frame header
2. Payload field for holding packet
3. Frame trailer

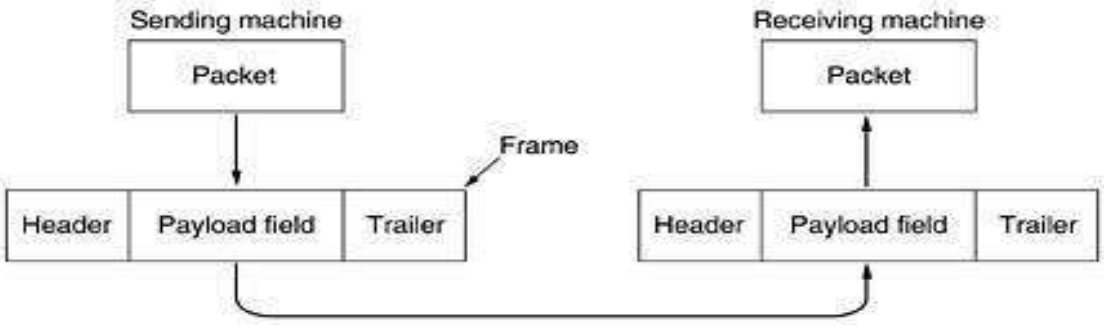


Figure 1.1 Relationships between Packets and Frames

Services provided to the network layer

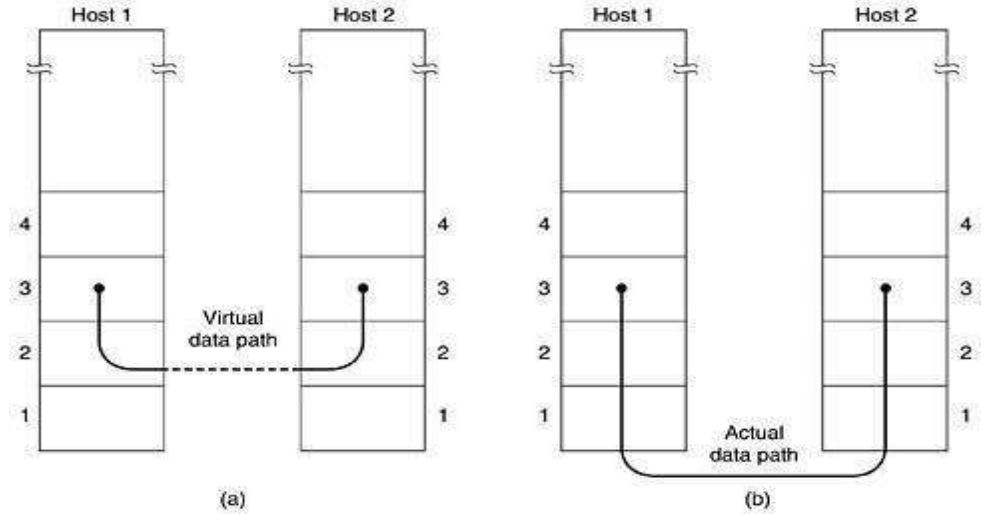


Figure 1.2 (a) Virtual communication. (b) Actual communication.

Transferring data from the network layer on the source machine to the network layer on the destination machine. 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**

   - Source machine sends independent frames to destination machine having destination machine acknowledge them
   - No logical connection
   - Used when error rate is very low
   - Good for real-time traffic (voice)

2. **Acknowledged connectionless service**

   - No logical connection
   - Each frame sent is individually acknowledged
   - Useful over unreliable channels (i.e. wireless systems)

3. **Acknowledged connection-oriented service**

   - Source and destination machines establish a connection before any data are transferred
   - Each frame is numbered
   - DLL guarantees that...
     - Each frame is received
     - Each frame is received exactly once
     - Each frame is received in the right order

3 PHASES

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

1. Connection established
2. Frames are transmitted
3. Connection released

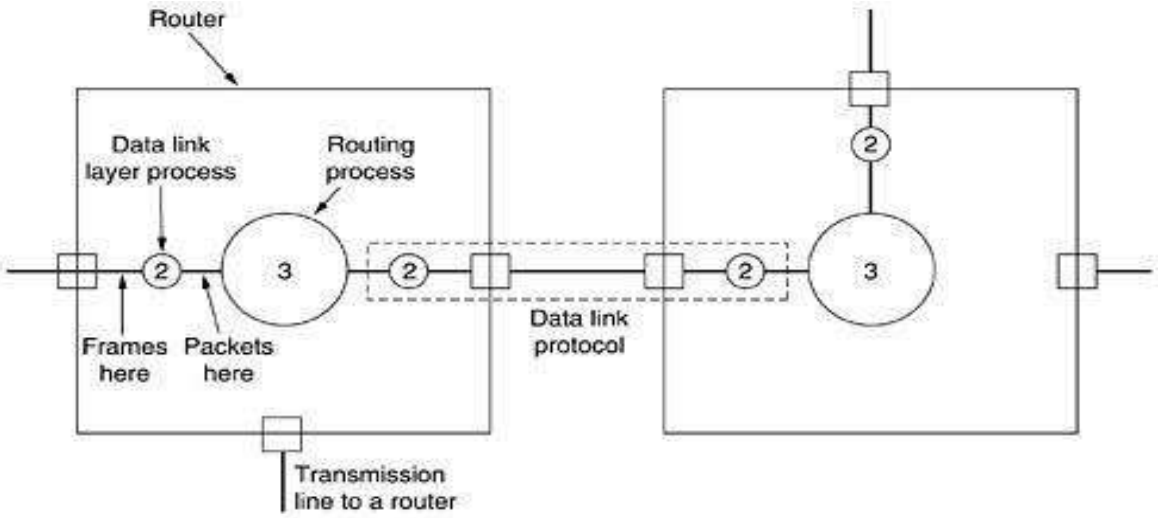


Figure 1.3 Placement of the data link Protocol

- Consider a typical example: a WAN subnet consisting of routers connected by point-to-point leased telephone lines.
- When a frame arrives at a router, the hardware checks it for errors, and then passes the frame to the data link layer software.
- The data link layer software checks to see if this is the frame expected, and if so, gives the packet contained in the payload field to the routing software.
- The routing software then chooses the appropriate outgoing line and passes the packet back down to the data link layer software, which then transmits it. The flow over two routers is shown in **Fig. 1-3.**

(1). FRAMING

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.

There are four methods:

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

1.Character count:

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.

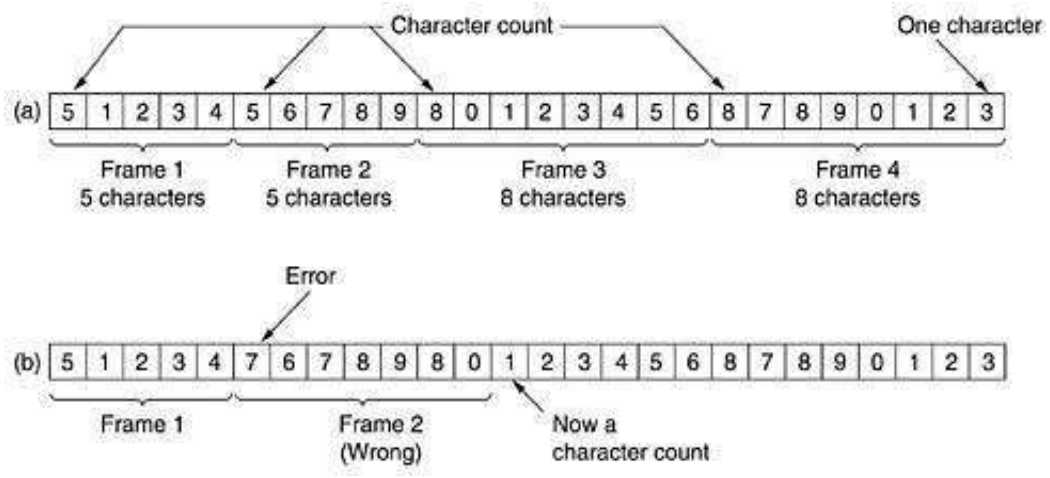


Figure 3-4. A character stream. (a) Without errors. (b) With one error.

Explanation (Figure 3-4.(a) A character stream Without errors.)

- 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.

Explanation (Figure 3-4.(b) A character stream with errors.)

- 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.

2.Flag bytes with byte stuffing:

Character-oriented framing approach

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

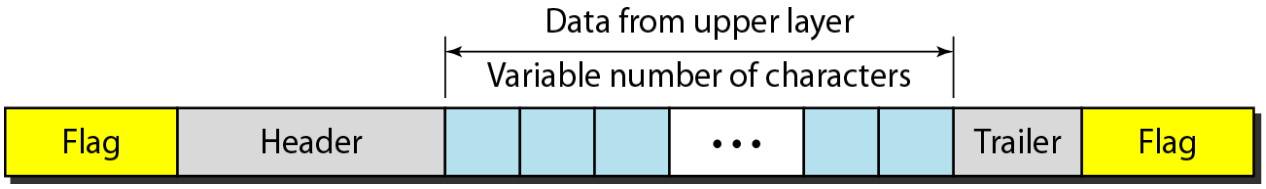


Figure: shows the format of a frame in a character-oriented protocol

Advantage:

1. Simple framing method.
2. Character-oriented framing was popular when only text was exchanged by the data Link layers.
3. The flag could be selected to be any character not used for text communication.

Disadvantage:

1. Even if with checksum, the receiver knows that the frame is bad there is no way to tell where the next frame starts.
2. Asking for retransmission doesn't help either because the start of the retransmitted frame is not known.
3. Hence No longer used.

3.Starting and ending character with byte stuffing

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

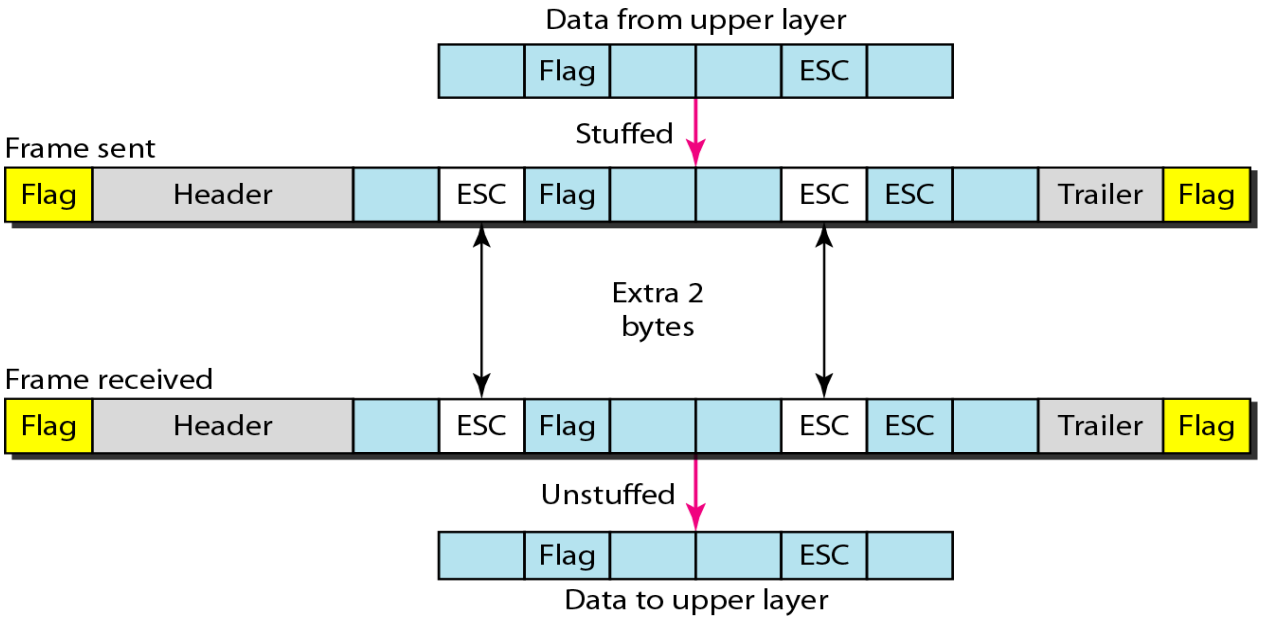


Figure : Byte stuffing and unstuffing

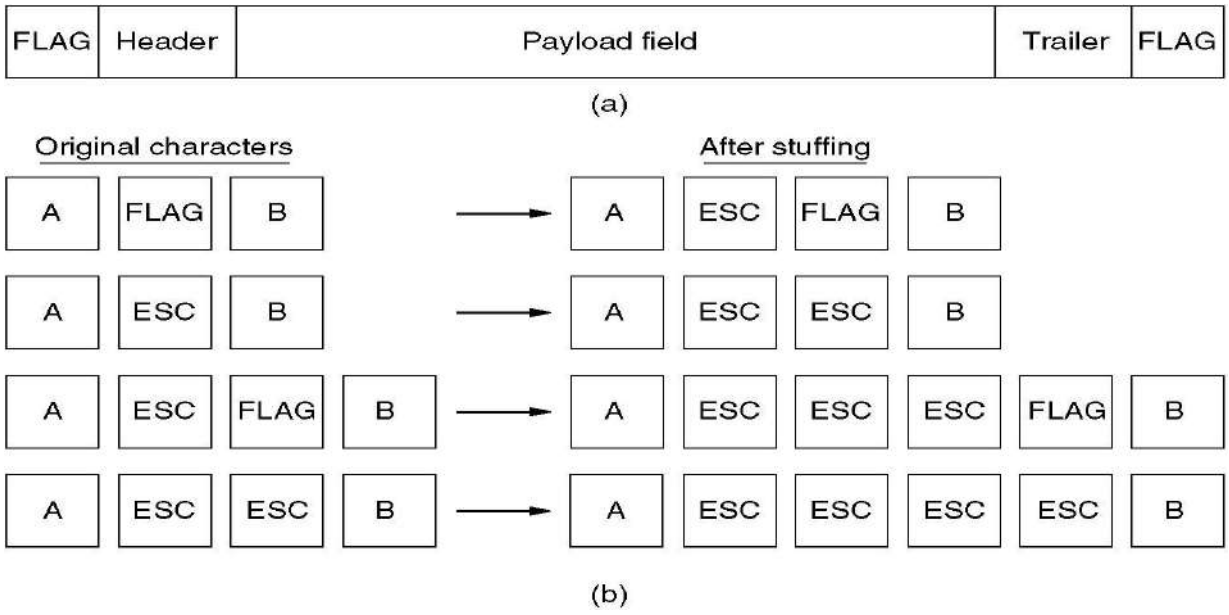


Fig: Framing with byte stuffing

Problem: fixed character size: assumes character size to be 8 bits: can't handle heterogeneous environment.

Bit-Oriented framing approach

- Bit stuffing is the process of adding one extra 0 whenever five consecutive 1's follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.
- Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in Figure below
- This flag can create the same type of problem. That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame.

➢ We do this by stuffing 1 single bit (instead of I byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing.

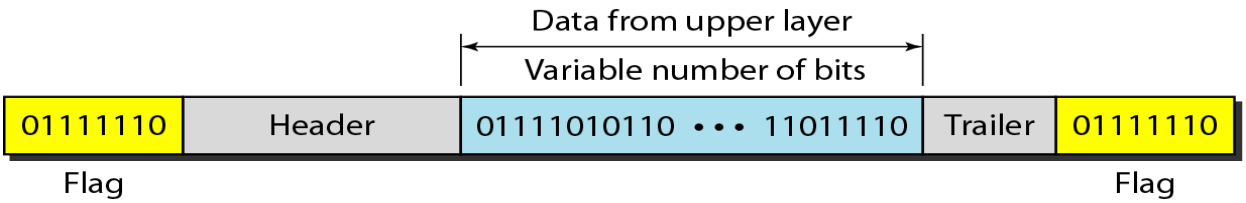


Figure (a)

Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

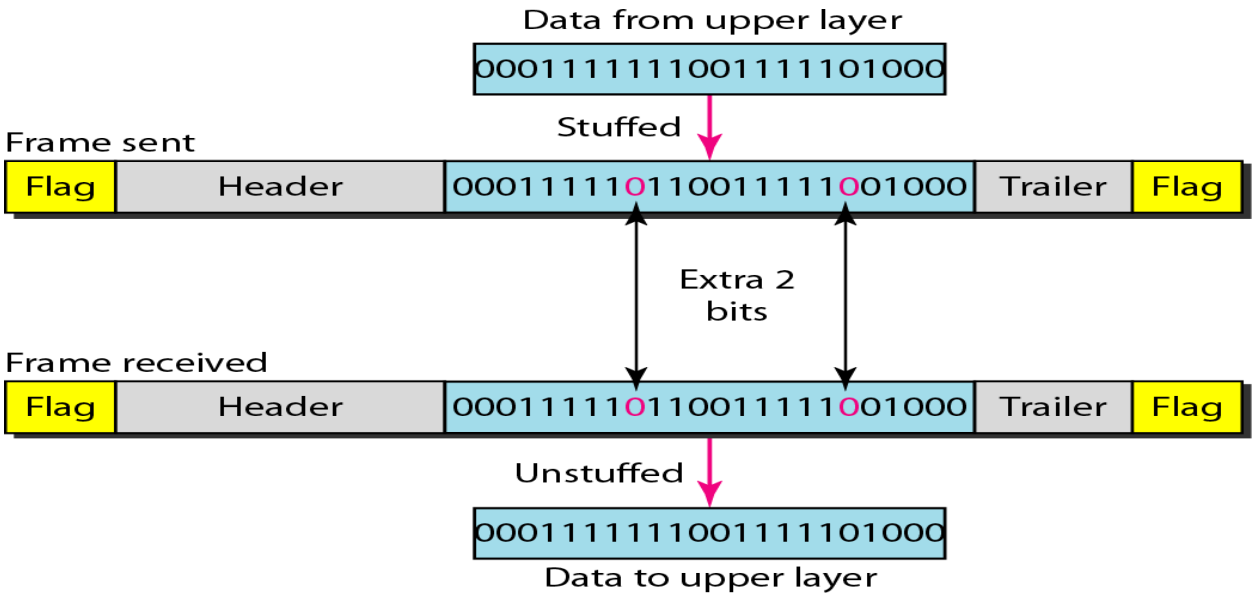


Figure (b)

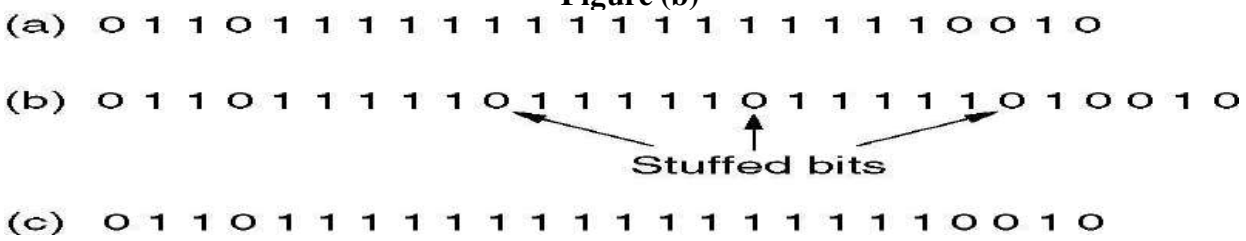


Figure (c)

(a) The original data.
(b) The data as they appear on the line.
(c) The data as they are stored in receiver's memory after destuffing.

4.Physical layer coding violation:

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

. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.

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.

(2) ERROR CONTROL

- How do we make sure that all frames are eventually delivered to the network layer at the destination and in the proper order?
- Provide sender with some acknowledgement about what is happening with the receiver
- Sender could wait for acknowledgement

Disadvantages

- If a frame vanishes, the receiver will not send an acknowledgement thus, sender will wait forever
- Dealt with by timers and sequence numbers – important part of DLL
- Sender transmits a frame, starts a timer.
- Timer set to expire after interval long enough for frame to reach destination, be processed, and have acknowledgement sent to sender
- Is a danger of frame being transmitted several times, however dealt with by assigning sequence numbers to outgoing frames, so that receiver can distinguish retransmissions from originals.

(3) FLOW CONTROL

What do we do when a sender transmits frames faster than the receiver can accept them?

- **Feedback-based flow control** – 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** – the protocol has a built-in mechanism that limits the rate at which the sender may transmit data, using feedback from the receiver.

ERROR DETECTION AND CORRECTION METHODS

- Because of Attenuation, distortion, noise and interferences, errors during transmission are inevitable, leading to corruption transmitted bits.
- Longer the frame size and higher the probability of single bit error, lower is the probability receiving a frame without error.

ERROR

- When data is being transmitted from one machine to another, it may possible that data become corrupted on its way. Some of the bits may be altered, damaged or lost during transmission. Such a condition is known as **error**.

TYPES OF ERRORS

- **Single bit error**: Only one bit gets corrupted. Common in Parallel transmission.
- **Burst error:** More than one bit gets corrupted very common in serial transmission of data occurs when the duration of noise is longer than the duration of one bit.

Single bit error:

- The term single-bit error means that only one bit of given data unit (such as a byte, character, or data unit) is changed from 1 to 0 or from 0 to 1 as shown in Fig. 3.2.1.
- Single bit errors are least likely type of errors in serial data transmission.
- For example, if 16 wires are used to send all 16 bits of a word at the same time and one of the wires is noisy, one bit is corrupted in each word.

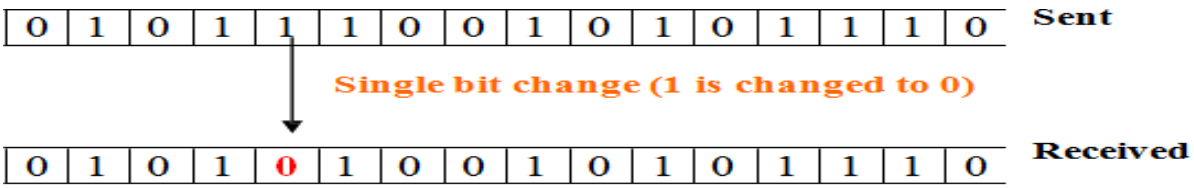


Figure 3.2.1 Single bit error

Burst error:

- More than one bit gets corrupted very common in serial transmission of data occurs when the duration of noise is longer than the duration of one bit.
- The noise affects data; it affects a set of bits.
- The number of bits affected depends on the data rate and duration of noise.

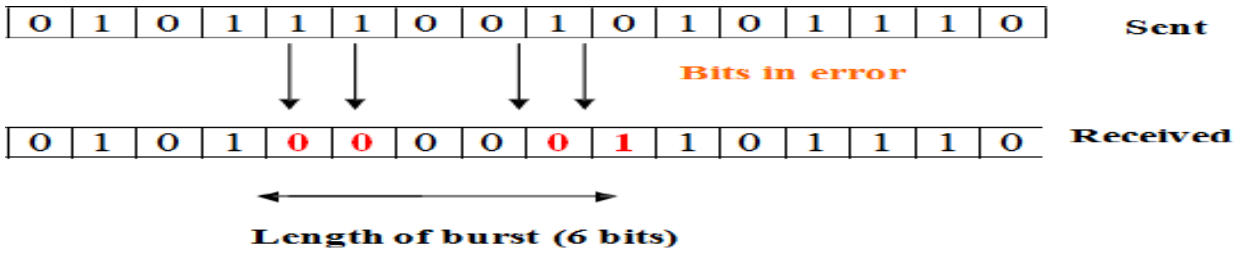


Figure 3.2.2 Burst Error

ERROR DETECTION TECHNIQUES

Basic approach used for error detection is the use of redundancy, where additional bits are added to facilitate detection and correction of errors. Popular techniques are

- Simple Parity check
- Two-dimensional Parity check
- Checksum
- Cyclic redundancy check

Redundancy is the method in which some extra bits are added to the data so as to check whether the data contain error or not.

m - data bits (i.e., message bits)
r - redundant bits (or check bits).
n - total number of bits
n= (m + r).

An n-bit unit containing data and check-bits is often referred to as an n-bit codeword.

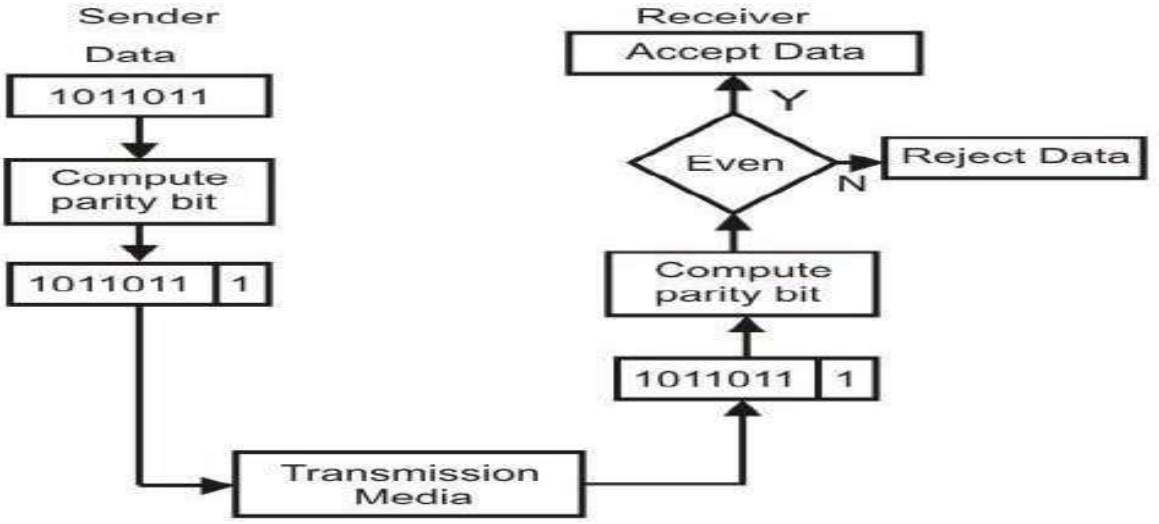
SIMPLE PARITY CHECK

The simplest and most popular error detection scheme. Appends a Parity bit to the end of the data.

Even Parity: **01000001** – Number of ones in the group of bits is even
Odd Parity: **11000001 -** Number of ones in the group of bits is odd

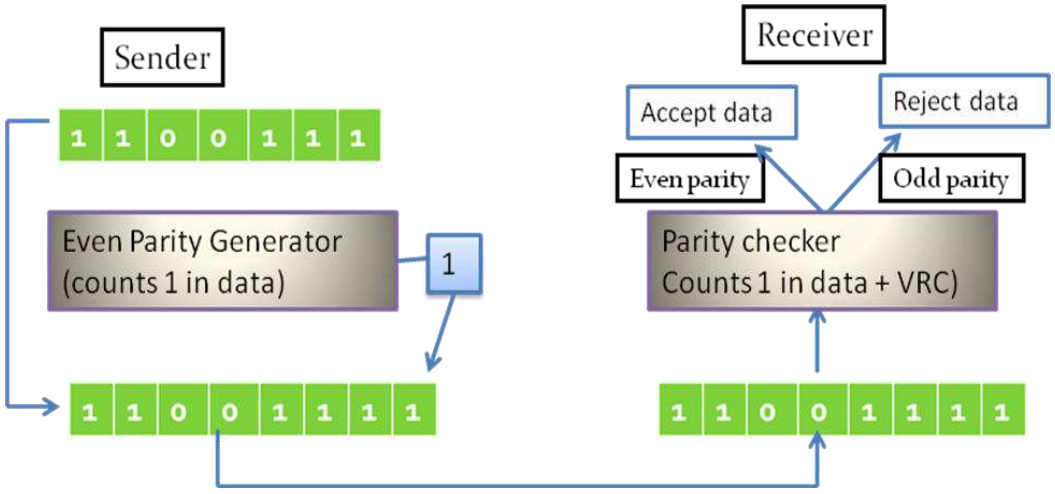
A parity of 1 is added to the block if it contains an odd number of 1's (ON bits) and 0 is added if it contains an even number of 1's. At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit.

This scheme makes the total number of 1's even, that is why it is called *even parity checking*. Considering a 4-bit word, different combinations of the data words and the corresponding code words are given in Table 3.2.1.



Decimal value	Data Block	Parity bit	Code word
0	0000	0	00000
1	0001	1	00011
2	0010	1	00101
3	0011	0	00110
4	0100	1	01001
5	0101	0	01010
6	0110	0	01100
7	0111	1	01111
8	1000	1	10001
9	1001	0	10010
10	1010	0	10100
11	1011	1	10111
12	1100	0	11000
13	1101	1	11011
14	1110	1	11101
15	1111	0	11110

Example:



PERFORMANCE OF SIMPLE PARITY CHECK

- Simple parity check can **detect all single-bit error**
- It can also detect burst error, if the number of bits in **even or odd.**
- The technique is not foolproof against burst errors that **invert more than one bit**. If an even number of bits is inverted due to error, the **error is not detected.**

TWO-DIMENSION PARITY CHECKING

- Performance can be improved by using two dimensional parity check, which **organizes the block of bits in the form of table.**
- Parity check bits are **calculated from each row**, which is equivalent to a simple parity check.
- Parity check bits are also **calculated for all columns.**
- Both are sent along with the data.
- At the receiving end these are compared with the parity bits calculated on the received data.

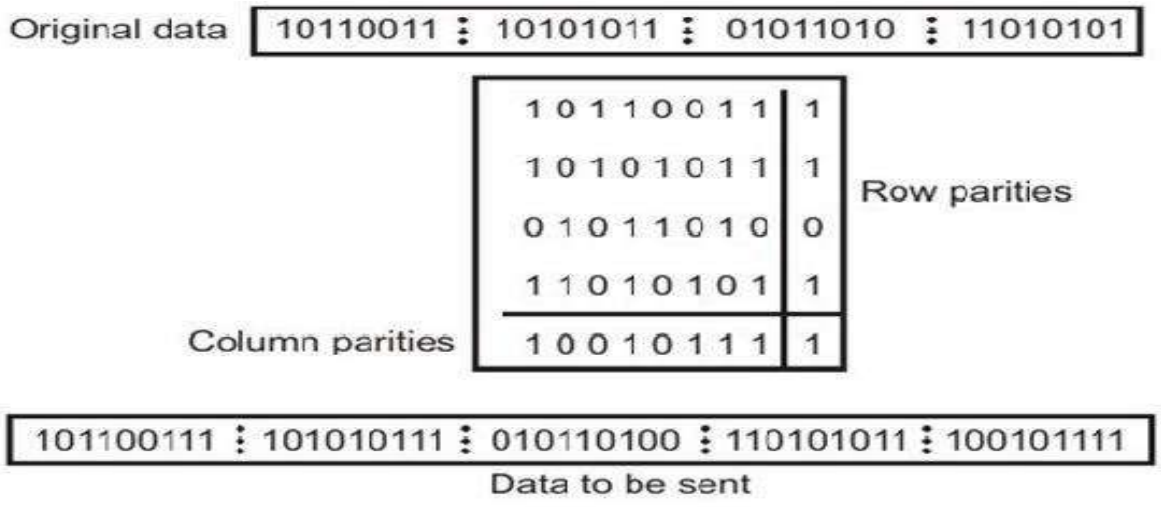


Figure 3.2.4 Two-dimension Parity Checking

Performance:

- If two bits in one data unit are damaged and two bits in exactly same position in another data unit are also damaged, The 2-D Parity check **checker will not detect an error**.
- For example, if two data units: **11001100 and 10101100.**
- If first and second from last bits in each of them is changed, making the data units as **01001110 and 00101110**, the error cannot be detected by 2-D Parity check.

CHECKSUM

- In checksum error detection scheme, the **data is divided into k segments each of m bits.**
- In the sender's end the segments are added using **1's complement arithmetic to get the sum.**
- The sum is complemented to get the checksum. The **checksum** segment is sent **along with the data segments**

Example 1:

Sender

10101001	subunit1
00111001	subunit 2
11100010	sum
00011101	Complement of sum

Reciever:

10101001	subunit1
00111001	subunit2
00011101	Checksum
11111111	sum
00000000	complement

Conclusion = Accept data.

10101001	00111001	00011101
Data		checksum

Example 2: K= 10110011, 10101011, 01011111, 11010101

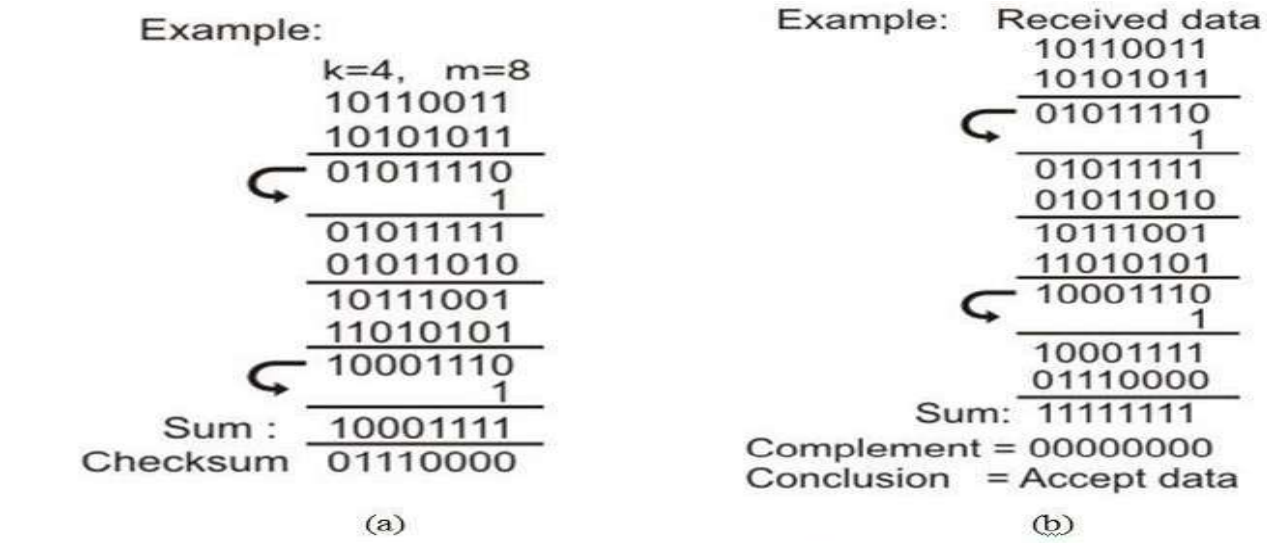


Figure 3.2.5 (a) Sender's end for the calculation of the checksum, (b) Receiving end for checking the checksum

CYCLIC REDUNDANCY CHECK

- One of the most powerful and commonly used error detecting codes.

Basic approach:

- Given a m-bit block of bit sequence, the sender generates an n-bit sequence known as **frame sequence check(FCS),** so that the resulting frame, consisting of m+n bits exactly divisible by **same predetermined number.**

- The receiver divides the incoming frame by that number and, if there is **no reminder, assumes there was no error.**

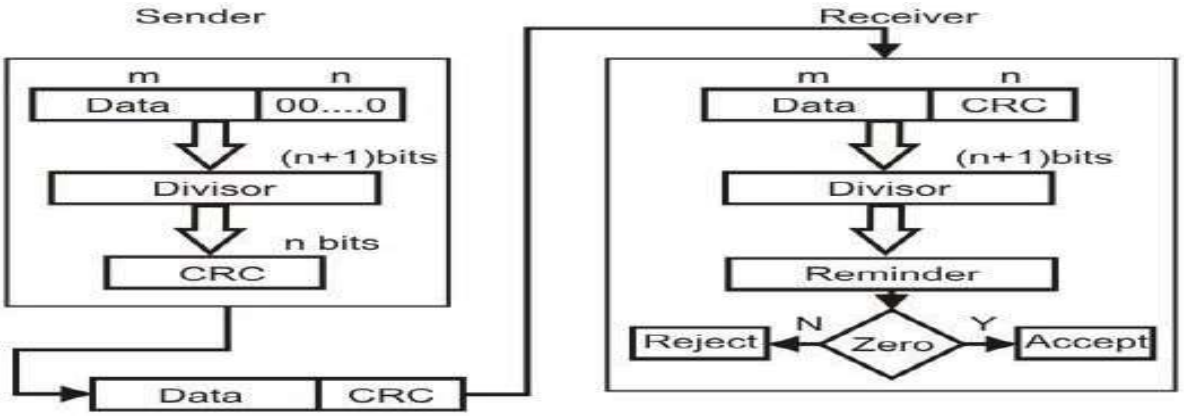


Fig. 3.2.7 by dividing a sample 4- bit number by the coefficient of the generator polynomial x^3+x+1, which is 1011, using the modulo-2 arithmetic.

Modulo-2 arithmetic is a binary addition process without any carry over, which is just the Exclusive-OR operation.

Consider the case where k=1101. Hence we have to divide 1101000 (i.e. *k* appended by 3 zeros) by 1011, which produces the remainder r=001, so that the bit frame *(k+r)* =1101001 is actually being transmitted through the communication channel.

At the receiving end, if the received number, i.e.,1101001 is divided by the same generator polynomial 1011 to get the remainder as 000, it can be assumed that the data is free of errors.

Sender: Sender transmit the data along with remainder (**CRC**)

```
      1111  Quotient          Data: 1101
1011 | 1101000  <------ K     Divisor: 1011
       1011
        1100
        1011
         1110
         1011
          1010                Data to be sent:   1 1 0 1 0 0 1
          1011                                   Data | CRC
           001 Reminder (r)
```

Receiver:

```
          1111   Quotient          Data: 1101001
     ┌──────────
1011 │ 1101001                     Divisor: 1011
       1011
       ----
        1100
        1011
        ----
        1110
        1011
        ----
         1011
         1011
         ----
          000   Reminder
```

Note: Remainder is zero, no error. Receiver can accept the data.

Performance of CRC

- CRC can detect all single-bit errors.
- CRC can detect all double-bit errors(three1's)
- CRC can detect any odd number of errors of less than the degree of the polynomial.
- CRC detects most of the larger burst errors with a high probability.

ERROR CORRECTING CODES

Concept of error-correction can be easily understood by examining the simplest case of single-bit errors. As we have already seen that a single-bit error can be detected by addition of a parity bit with the data, which needed to be send.

A single additional bit can detect error, but it's not sufficient enough to correct that error too. For correcting an error one has to know the exact position of error, i.e. exactly which bit is in error (to locate the invalid bits).

For example, to correct a single-bit error in an ASCII character, the error correction must determine which one of the seven bits is in error. To this, we have to add some additional redundant bits.

To calculate the numbers of redundant bits (r) required to correct d data bits, let us find out the relationship between the two. So we have (d+r) as the total number of bits, which are to be transmitted; then r must be able to indicate at least d+r+1 different values. Of these, one value means no error, and remaining d+r values indicate error location of error in each of d+r locations. So, d+r+1 states must be distinguishable by r bits, and r bits can indicates 2^r states. Hence, 2^r must be greater than d+r+1.

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

The value of r must be determined by putting in the value of d in the relation. For example, if d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d+r = 7+4 =11).

Now let us examine how we can manipulate these bits to discover which bit is in error. A technique developed by R.W. Hamming provides a practical solution. The solution or coding scheme he developed is commonly known as **Hamming Code**.

Hamming code can be applied to data units of any length and uses the relationship between the data bits and redundant bits as discussed.

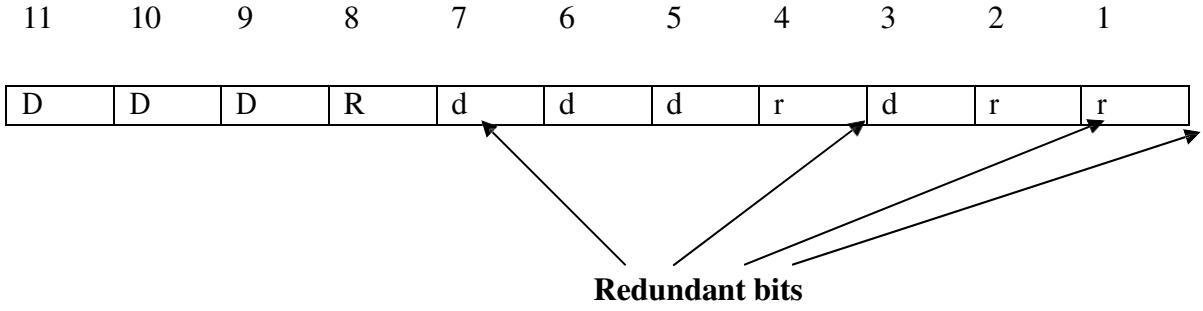


Figure 3.2.8 Positions of redundancy bits in hamming code

Basic approach for error detection by using Hamming code is as follows:

- To each group of m information bits k parity bits are added to form (m+k) bit code as shown in Fig. 3.2.8.
- Location of each of the (m+k) digits is assigned a decimal value.
- The k parity bits are placed in positions 1, 2, …, 2^{k-1} positions.–K parity checks are performed on selected digits of each codeword.
- At the receiving end the parity bits are recalculated. The decimal value of the k parity bits provides the bit-position in error, if any.

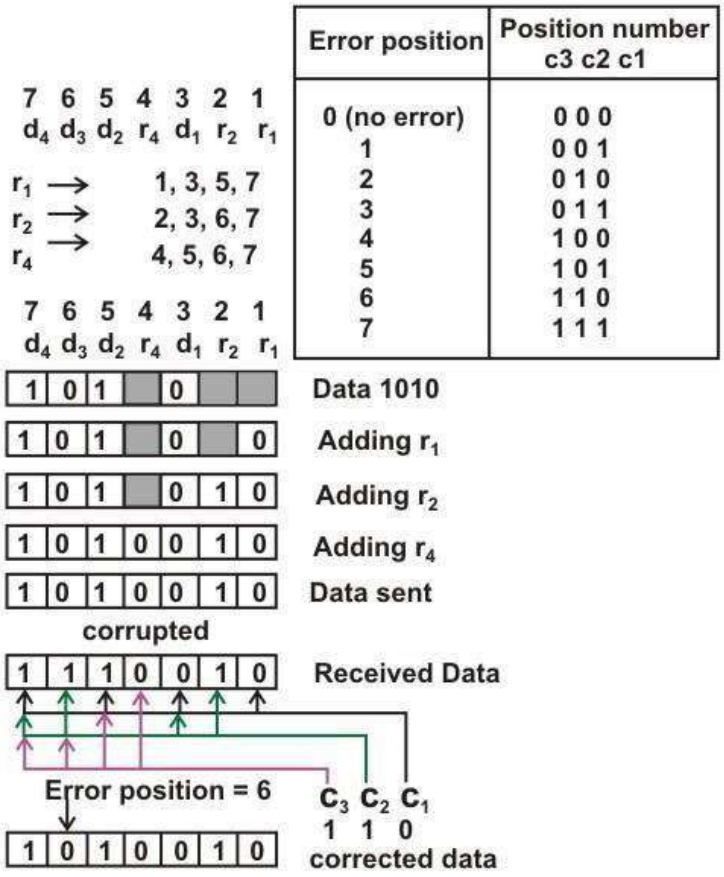


Figure 3.2.9 Use of Hamming code for error correction for a 4-bit data

Figure 3.2.9 shows how hamming code is used for correction for 4-bit numbers (d4d3d2d1) with the help of three redundant bits (r3r2r1).

For the example data 1010, first r1 (0) is calculated considering the parity of the bit positions, 1, 3, 5 and 7. Then the parity bits r2 is calculated considering bit positions 2, 3, 6 and 7.

Finally, the parity bits r4 is calculated considering bit positions 4, 5, 6 and 7 as shown. If any corruption occurs in any of the transmitted code 1010010, the bit position in error can be found out by calculating r3r2r1 at the receiving end.

For example, if the received code word is 1110010, the recalculated value of r3r2r1 is 110, which indicates that bit position in error is 6, the decimal value of 110.

DATA LINK LAYER PROTOCOLS

i) AUTOPIAN SIMPLEX PROTOCOL

The following assumption has been made for developing the (algorithm) simplex protocol.

- The channel is a perfect noiseless channel.
- Hence an ideal channel in which no frames are lost, duplicated, or corrupted.
- No flow control and error control used.
- It is a unidirectional protocol in which data frames are traveling in only one direction- from the sender to receiver.
- Both transmitting and receiving network layer are always ready.
- Processing time that is small enough to be negligible.
- Infinite buffer space is available.

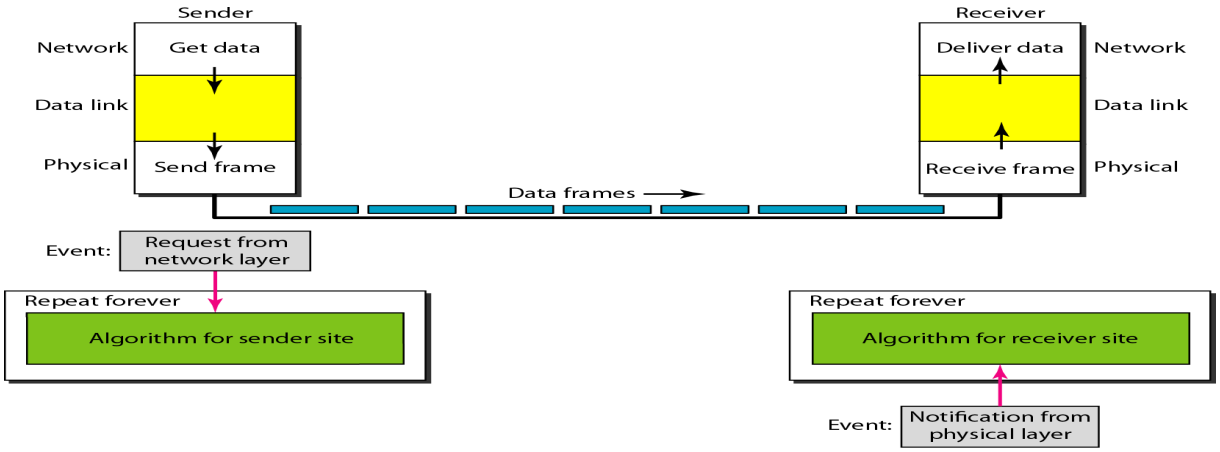


Figure3.1 shows The design of the simplest protocol with no flow or error control

Example for simplex protocol

- **Figure 3.2** below shows an example of communication using this protocol. It is very simple.
- The sender sends a sequence of frames without even thinking about the receiver.
- To send three frames, three events occur at the sender site and three events at the receiver site.

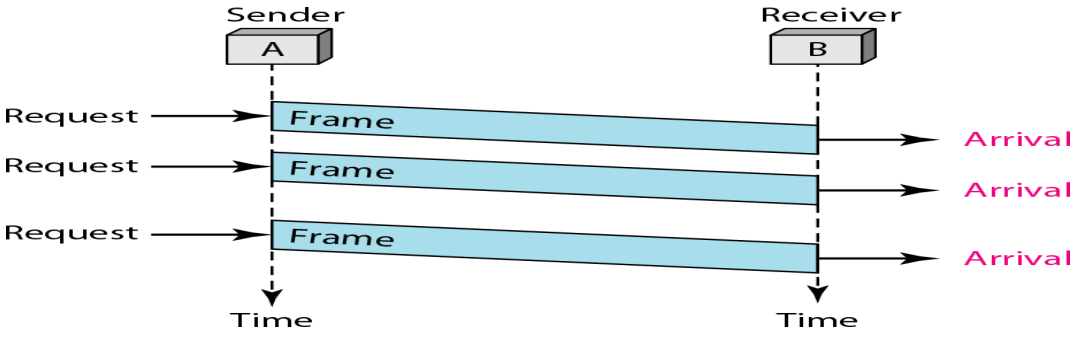


Figure3.2 Example for simplex protocol

Efficiency analysis

- Transmission in one direction
- The receiver is always ready to receive the next frame (has infinite buffer storage).
- Error-free communication channel.
- No acknowledgments or retransmissions used.
- If frame has *d* data bits and *h* overhead bits, channel bandwidth *b* bits/second:

Maximum Channel Utilization = data size / frame size = $d / (d + h)$

Maximum Data Throughput = $d / (d + h) *$ channel bandwidth = $d / (d + h) * b$

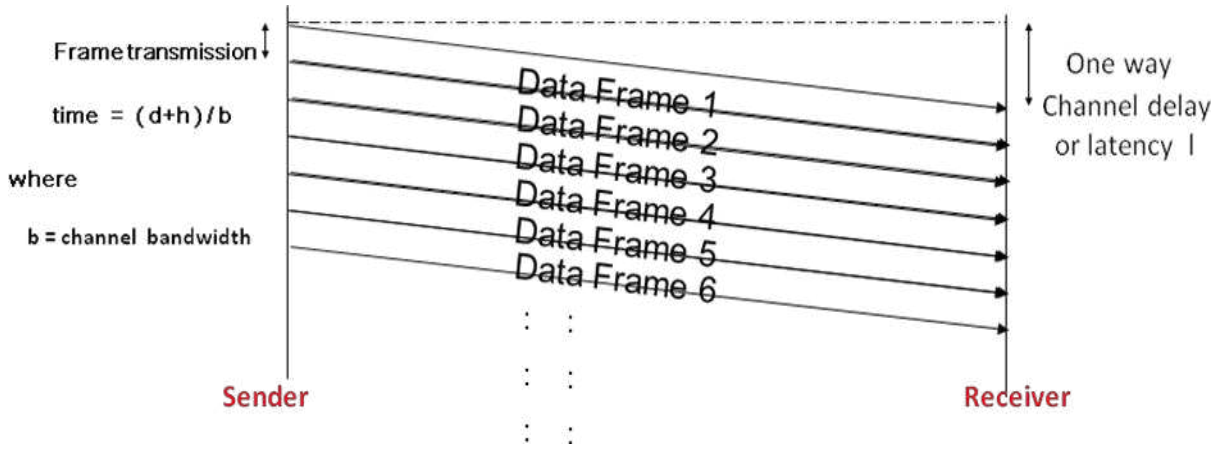


Figure 3.3. Efficiency analysis

ii) A SIMPLEX STOP-AND-WAIT PROTOCOL FOR AN ERROR-FREE CHANNEL

The following assumption has been made for developing the Stop-and-Wait Protocol

- The channel is a perfect noiseless channel.
- Flow control used
- It is a bidirectional protocol in which frames are traveling in both direction
- Both transmitting and receiving network layer are always not ready.
- Processing time considerable
- Finite buffer space is available
- The receiver may not be always ready to receive the next frame (finite buffer storage).
- Receiver sends a positive acknowledgment frame to sender to transmit the next data frame which showed in the below figure(3.4).
- Error-free communication channel assumed. No retransmissions used.
- Maximum channel utilization » (time to transmit frame /round trip time)* $d / (d + h)$

  » $d / (b * R)$

- Maximum data throughput » Channel Utilization * Channel Bandwidth

  » $d / (b * R) * b = d / R$

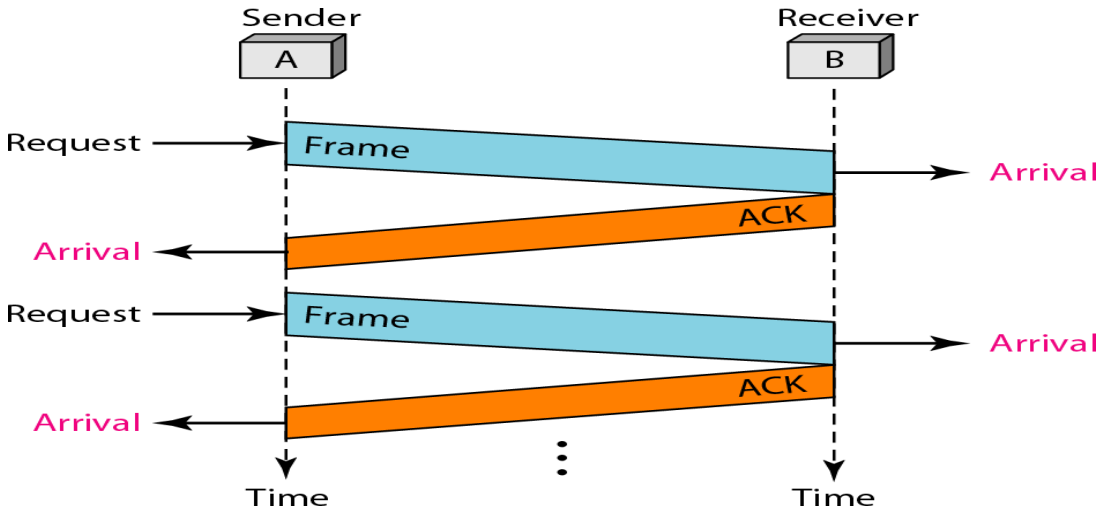


Figure 3.4. Stop-and-Wait protocol flow diagram

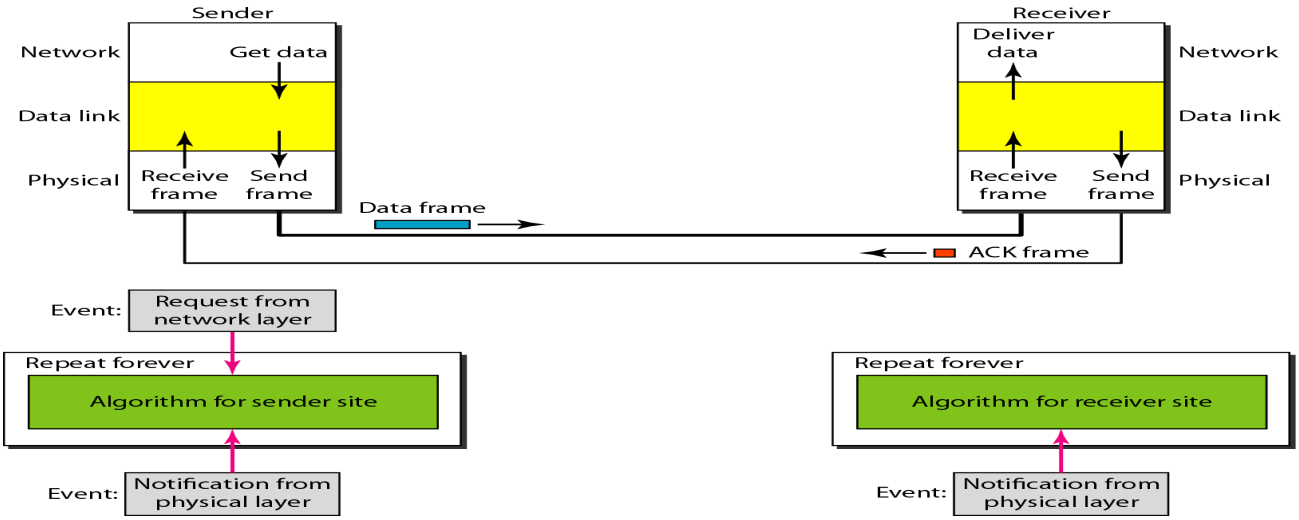


Figure 3.5 Design of Stop-and-Wait Protocol

iii) A SIMPLEX STOP-AND-WAIT PROTOCOL FOR A NOISY CHANNEL

Automatic Repeat Request (ARQ)

Purpose: To ensure a sequence of information packets is delivered in order and without errors or duplications despite transmission errors & losses.

1. STOP AND WAIT WITH ARQ

Automatic Repeat Request (ARQ), an error control method, is incorporated with stop and wait flow control protocol

- If error is detected by receiver, it discards the frame and send a negative ACK (NAK), causing sender to re-send the frame.
- In case a frame never got to receiver, sender has a timer: each time a frame is sent, timer is set ! If no ACK or NAK is received during timeout period, it re-sends the frame
- Timer introduces a problem: Suppose timeout and sender retransmits a frame but receiver actually received the previous transmission ! receiver has duplicated copies.
- To avoid receiving and accepting two copies of same frame, frames and ACKs are alternatively labeled 0 or 1: ACK0 for frame 1, ACK1 for frame 0.

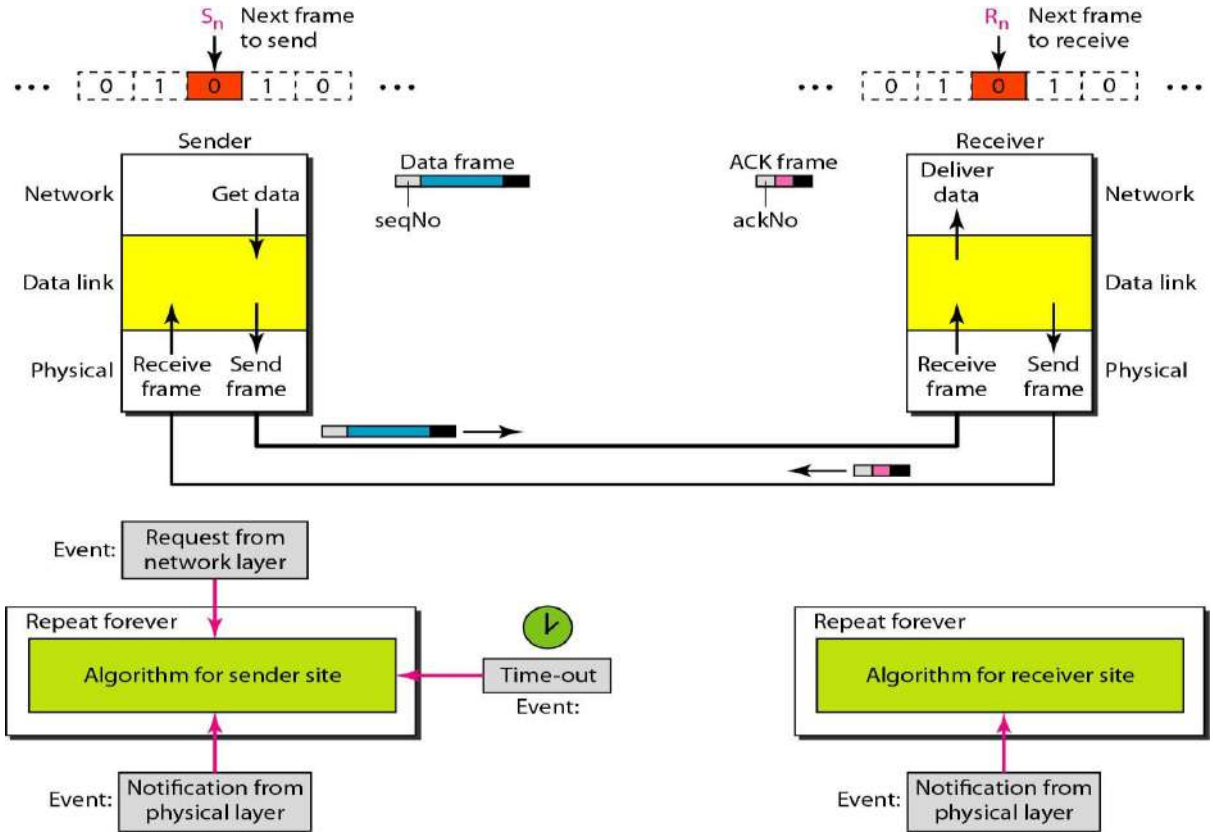


Figure 3.6 shows Design of the Stop-and-Wait ARQ Protocol

Example

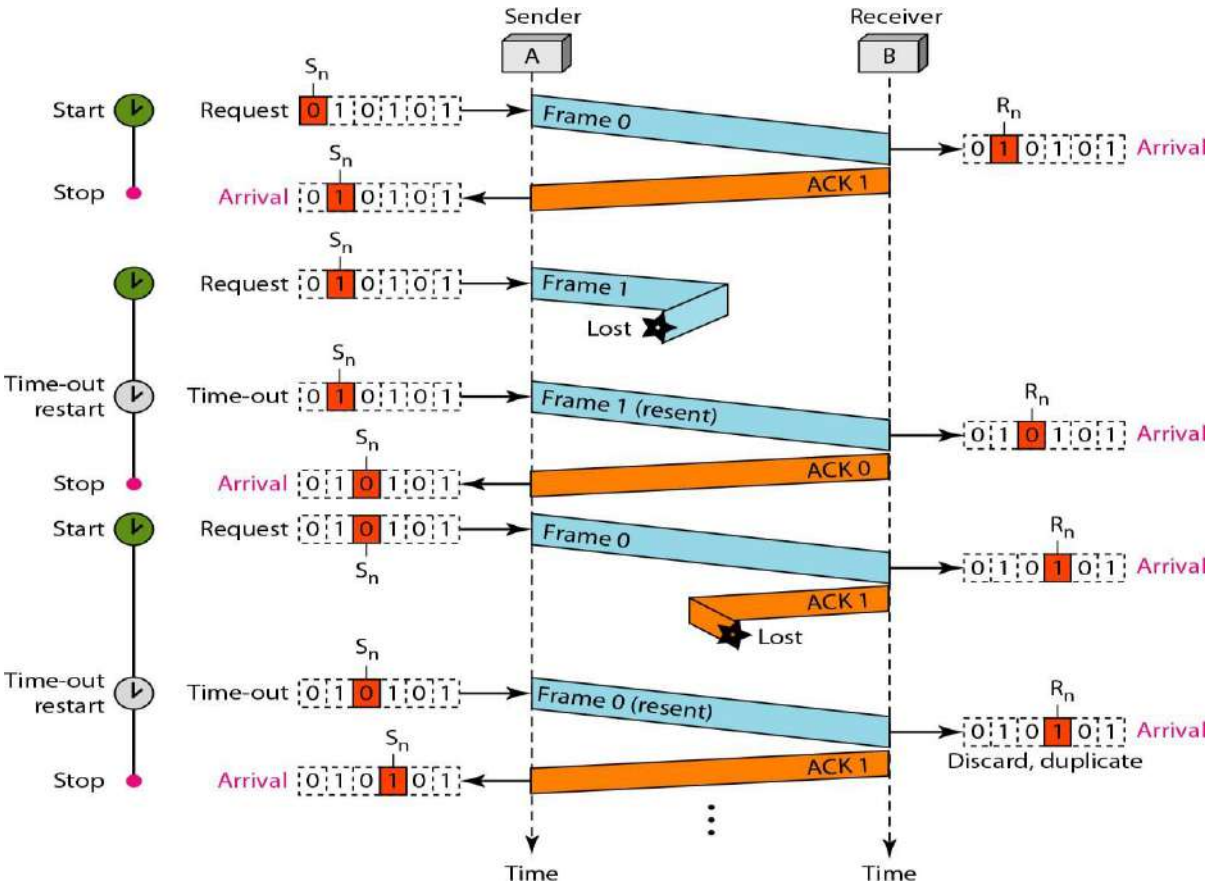


Figure 3.7 Shows an example of Stop-and-Wait ARQ.

Event :

- Frame 0 is sent and acknowledged.
- Frame 1 is lost and resent after the time-out.
- The resent frame 1 is acknowledged and the timer stops.
- Frame 0 is sent and acknowledged, but the acknowledgment is lost.
- The sender has no idea if the frame or the acknowledgment is lost.
- So after the time-out, it resends frame0, which is acknowledged.

ADVANTAGES OF STOP AND WAIT ARQ

- It can be used for noisy channels
- It has both error and flow control mechanism
- It has timer implementation

DISADVANTAGES OF STOP AND WAIT ARQ

- Efficiency is very less.
- Only 1 frame is sent at a time
- Timer should be set for each individual frame

- No pipelining
- sender window size is 1(disadvantage over Go back N ARQ)
- receiver window size is 1(disadvantage over selective repeat ARQ)

SLIDING WINDOW PROTOCOLS

1. **A One-Bit Sliding Window Protocol**
2. **A Protocol Using Go-Back-N**
3. **A Protocol Using Selective Repeat**

i) A ONE-BIT SLIDING WINDOW PROTOCOL

The starting machine fetches the first packet from its network layer, builds a frame from it, and sends it. When this (or any) frame arrives, the receiving data link layer checks to see if it is a duplicate, just as in protocol 3. If the frame is the one expected, it is passed to the network layer and the receiver's window is slid up.

The acknowledgement field contains the number of the last frame received without error. If this number agrees with the sequence number of the frame the sender is trying to send, the sender knows it is done with the frame stored in buffer and can fetch the next packet from its network layer. If the sequence number disagrees, it must continue trying to send the same frame. Whenever a frame is received, a frame is also sent back.

Assume that computer A is trying to send its frame 0 to computer B and that B is trying to send its frame 0 to A. Suppose that A sends a frame to B, but A's timeout interval is a little too short. Consequently, A may time out repeatedly, sending a series of identical frames, all with seq = 0 and ack = 1.

When the first valid frame arrives at computer B, it will be accepted and frame_expected will be set to 1. All the subsequent frames will be rejected because B is now expecting frames with sequence number 1, not 0. Furthermore, since all the duplicates have ack = 1 and B is still waiting for an acknowledgement of 0, B will not fetch a new packet from its network layer. After every rejected duplicate comes in, B sends A a frame containing seq = 0 and ack = 0. Eventually, one of these arrives correctly at A, causing A to begin sending the next packet. No combination of lost frames or premature timeouts can cause the protocol to deliver duplicate packets to either network layer, to skip a packet, or to deadlock.

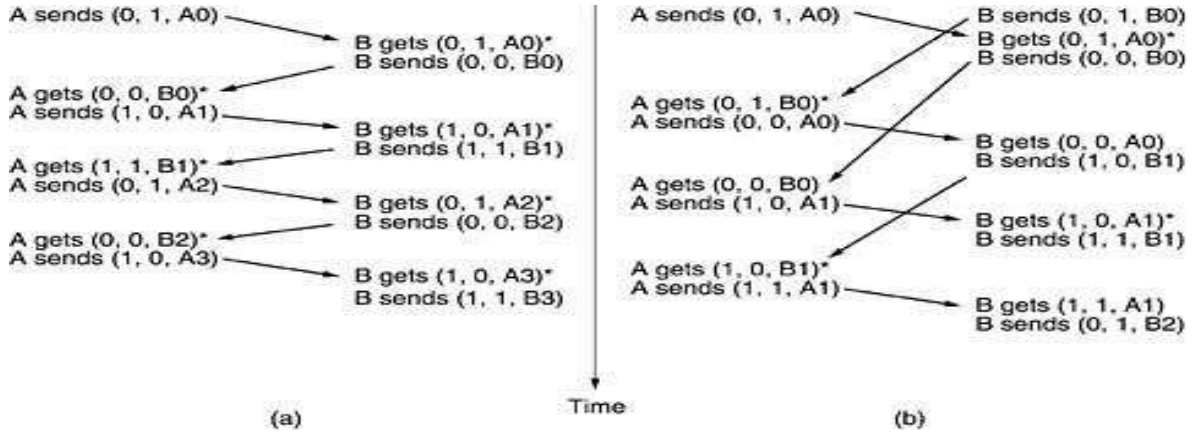
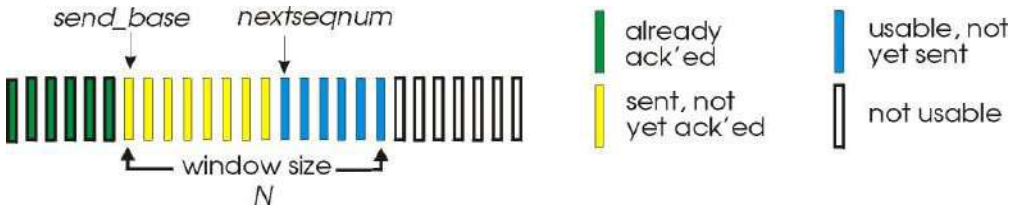


Fig 2.8: Two scenarios for protocol 4. (a) Normal case. (b) Abnormal case. The notation is (seq, ack, packet number). An asterisk indicates where a network layer accepts a packet

ii) A PROTOCOL USING GO-BACK-N

In a Go-Back-N (GBN) protocol, the sender is allowed to transmit multiple packets (when available) without waiting for an acknowledgment, but is constrained to have no more than some maximum allowable number, *N*, of unacknowledged packets in the pipeline.



Sender's view of sequence numbers in Go-Back-N

The above figure shows the sender's view of the range of sequence numbers in a GBN protocol. If we define *base* to be the sequence number of the oldest unacknowledged packet and *nextseqnum* to be the smallest unused sequence number (i.e., the sequence number of the next packet to be sent), then four intervals in the range of sequence numbers can be identified.

The range of permissible sequence numbers for transmitted but not-yet-acknowledged packets can be viewed as a ``window" of size *N* over the range of sequence numbers. *N* is often referred to as the **window size** and the GBN protocol itself as a **sliding window protocol.**

The GBN sender must respond to three types of events:

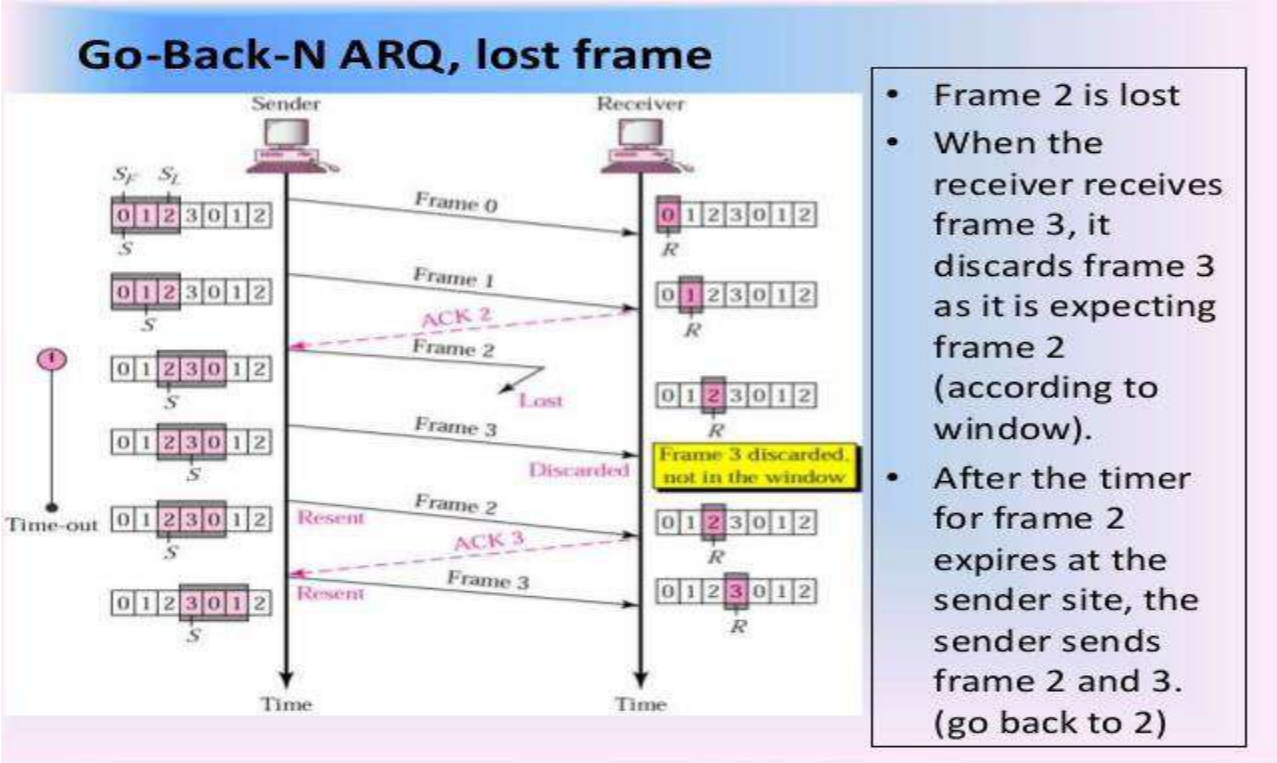
- **Invocation from above.** When rdt_send() is called from above, the sender first checks to see if the window is full, i.e., whether there are *N* outstanding, unacknowledged packets. If the window is not full, a packet is created and sent, and variables are appropriately updated. If the window is full, the sender simply returns the data back to the upper layer, an implicit indication that the window is full.
- **Receipt of an ACK.** In our GBN protocol, an acknowledgement for packet with sequence number *n* will be taken to be a **cumulative acknowledgement,** indicating that all packets with a sequence number up to and including n have been correctly received at the receiver. We'll come back to this

issue shortly when we examine the receiver side of GBN.

- **A timeout event.** The protocol's name, ``Go-Back-N,'' is derived from the sender's behavior in the presence of lost or overly delayed packets. As in the stop-and-wait protocol, a timer will again be used to recover from lost data or acknowledgement packets. If a timeout occurs, the sender resends *all* packets that have been previously sent but that have not yet been acknowledged. If an ACK is received but there are still additional transmitted-but-yet-to-be-acknowledged packets, the timer is restarted. If there are no outstanding unacknowledged packets, the timer is stopped.

In our GBN protocol, the receiver discards out-of-order packets. While it may seem silly and wasteful to discard a correctly received (but out-of-order) packet, there is some justification for doing so. Recall that the receiver must deliver data, in-order, to the upper layer.

The advantage of this approach is the simplicity of receiver buffering - the receiver need not buffer *any* out-of-order packets. Thus, while the sender must maintain the upper and lower bounds of its window and the position of *nextseqnum* within this window, the only piece of information the receiver need maintain is the sequence number of the next in-order packet. Of course, the disadvantage of throwing away a correctly received packet is that the subsequent retransmission of that packet might be lost or garbled and thus even more retransmissions would be required.



In the above figure, shows the operation of the GBN protocol for the case of a window size of four packets. Because of this window size limitation, the sender sends packets 0 through 3 but then must wait for one or more of these packets to be acknowledged before proceeding. As each successive ACK (e.g., ACK0 and ACK1) is received, the window slides forwards and the sender can transmit one new packet (pkt4 and pkt5, respectively).

On the receiver side, packet 2 is lost and thus packets 3, 4, and 5 are found to be out-of-order and are discarded.

The implementation would also likely be in the form of various procedures that implement the actions to be taken in response to the various events that can occur. In such **event-based programming,** the various procedures are called (invoked) either by other procedures in the protocol stack, or as the result of an interrupt.

iii) A PROTOCOL USING SELECTIVE REPEAT

Selective Repeat (SR) protocols avoid unnecessary retransmissions by having the sender retransmit only those packets that it suspects were received in error (i.e., were lost or corrupted) at the receiver. This individual, as-needed, retransmission will require that the receiver *individually* acknowledge correctly-received packets. A window size of *N* will again be used to limit the number of outstanding, unacknowledged packets in the pipeline.

The SR receiver will acknowledge a correctly received packet whether or not it is in-order. Out-of-order packets are buffered until any missing packets (i.e., packets with lower sequence numbers) are received, at which point a batch of packets can be delivered in-order to the upper layer. Figure receiver itemizes the various actions taken by the SR receiver.

Selective Repeat sender actions

- **Data received from above.** When data is received from above, the SR sender checks the next available sequence number for the packet. If the sequence number is within the sender's window, the data is packetized and sent; otherwise it is either buffered or returned to the upper layer for later transmission, as in GBN.

- **Timeout.** Timers are again used to protect against lost packets. However, each packet must now have its own logical timer, since only a single packet will be transmitted on timeout. A single hardware timer can be used to mimic the operation of multiple logical timers.
- **ACK received.** If an ACK is received, the SR sender marks that packet as having been received, provided it is in the window. If the packet's sequence number is equal to sendbase, the window base is moved forward to the unacknowledged packet with the smallest sequence number. If the window moves and there are untransmitted packets with sequence numbers that now fall within the window, these packets are transmitted.

Selective Repeat Receiver Actions

- **Packet with sequence number in [*rcvbase, rcvbase+N-1*] is correctly received.** In this case, the received packet falls within the receivers window and a selective ACK packet is returned to the sender. If the packet was not previously received, it is buffered. If this packet has a sequence number equal to the base of the receive window, then this packet, and any previously buffered and

consecutively numbered (beginning with *rcvbase)* packets are delivered to the upper layer. The receive window is then moved forward by the number of packets delivered to the upper layer.

- **Packet with sequence number in *[rcvbase-N,rcvbase-1*] is received.** In this case, an ACK must be generated, even though this is a packet that the receiver has previously acknowledged.
- **Otherwise.** Ignore the packet.

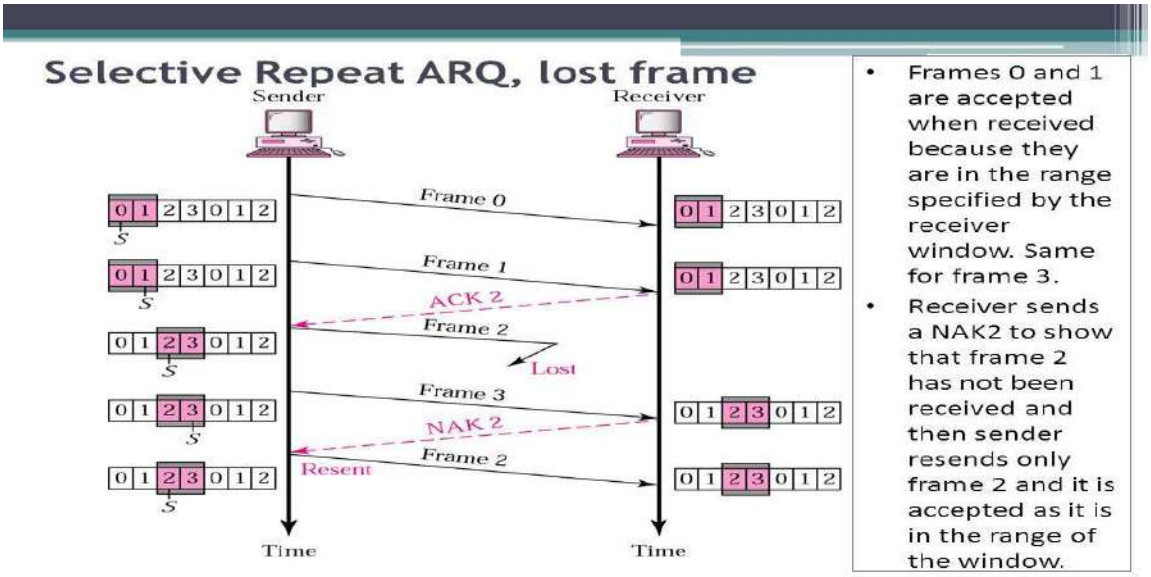
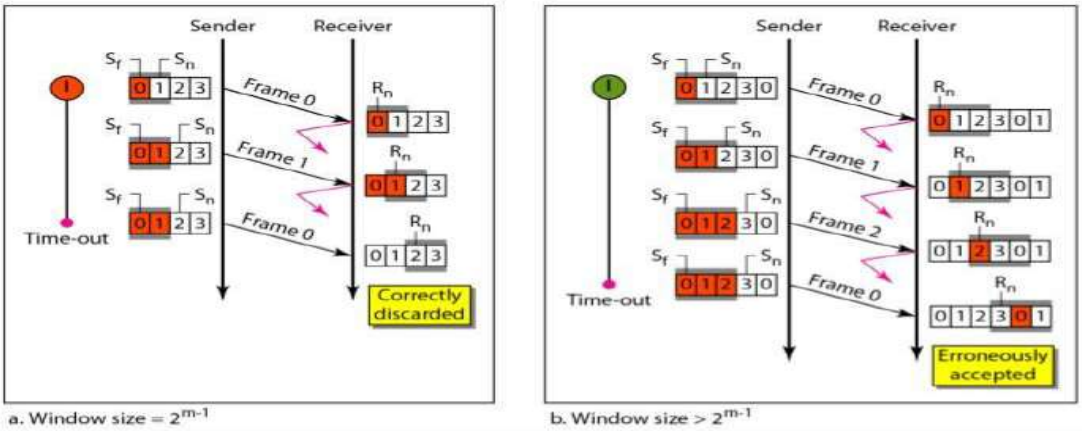


Figure 11.21 ***Selective Repeat ARQ, window size***



11.59

The lack of synchronization between sender and receiver windows has important consequences when we are faced with the reality of a finite range of sequence numbers.

Consider what could happen, for example, with a finite range of four packet sequence numbers,

0,1,2,3 and a window size of three. Suppose packets 0 through 2 are transmitted and correctly received and acknowledged at the receiver.

At this point, the receiver's window is over the fourth, fifth and sixth packets, which have sequence numbers 3, 0, and 1, respectively. Now consider two scenarios.

In the first scenario, shown in Figure (a), the ACKs for the first three packets are lost and the sender retransmits these packets. The receiver thus next receives a packet with sequence number 0 - a copy of the first packet sent.

In the second scenario, shown in Figure 3.4-19(b), the ACKs for the first three packets are all delivered correctly. The sender thus moves its window forward and sends the fourth, fifth and sixth packets, with sequence numbers 3, 0, 1, respectively. The packet with sequence number 3 is lost, but the packet with sequence number 0 arrives - a packet containing *new* data.

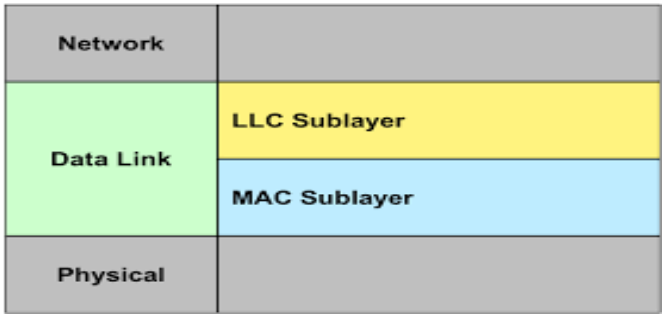
Computer Networks

UNIT- IV

The Medium Access Control Sub layer-The Channel Allocation Problem-Static Channel Allocation-Dynamic Channel Allocation, Multiple Access Protocols-Aloha- Carrier Sense Multiple Access Protocols-Collision-Free Protocols-Limited Contention Protocols- **Wireless LAN Protocols,** Ethernet-Classic Ethernet MAC Sub layer Protocol-Fast Ethernet Gigabit Ethernet-10- Gigabit Ethernet- Wireless LANs -The 802.11 Architecture and Protocol Stack-The 802.11 Physical Layer-The802.11 MAC Sublayer Protocol.

MEDIUM ACCESS CONTROL SUBLAYER

The MAC sublayer is the bottom part of the data link layer. The protocols used to determine who goes next on a multiaccess channel belong to a sublayer of the data link layer called the **MAC** (**Medium Access Control**) sublayer. The MAC sublayer is especially important in LANs, particularly wireless ones because wireless is naturally a broadcast channel. broadcast channels are sometimes referred to as **multi-access channels** or **random access channels**.



THE CHANNEL ALLOCATION PROBLEM

The channel allocation problem is how to allocate a single broadcast channel among competing users.

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 a user, 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.
- However, even assuming that the number of users could somehow be held constant at N, dividing the single available channel into static sub channels is inherently inefficient.
- The basic problem is that when some users are quiescent, their bandwidth is simply lost. They are not using it, and no one else is allowed to use it either.
- Furthermore, in most computer systems, data traffic is extremely bursty (peak traffic to mean traffic ratios of 1000:1 are common). Consequently, most of the channels will be idle most of the time.
- The poor performance of static FDM can easily be seen from a simple queuing 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 μC frames/sec. From queuing theory it can be shown that for Poisson arrival and service times,

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

- For example, if C is 100 Mbps, the mean frame length, $1/\mu$, is 10,000 bits, and the frame arrival rate, λ, is 5000 frames/sec, then $T = 200$ μ sec. Note that if we ignored the queuing delay and just asked how long it takes to send a 10,000 bit frame on a 100-Mbps network, we would get the (incorrect) answer of 100 μ sec. That result only holds when there is no contention for the channel.
- Now let us divide the single channel into N independent sub channels, each with capacity C/N bps. The mean input rate on each of the sub channels will now be λ/N. Recomputing T we get

Equation 4

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

- The mean delay using FDM is N times worse than if all the frames were somehow magically arranged orderly in a big central queue.
- Precisely the same arguments that apply to FDM also apply to time division multiplexing (TDM). Each user is statically allocated every Nth time slot. If a user does not use the allocated slot, it just lies fallow. The same holds if we split up the networks physically. Using our previous example again, if we were to replace the 100-Mbps network with 10 networks of 10 Mbps each and statically allocate each user to one of them, the mean delay would jump from 200 μ sec to 2 msec.

- Since none of the traditional static channel allocation methods work well with bursty traffic, we will now explore dynamic methods.

I. RANDOM ACCESS PROTOCOLS

- In a random access protocol, a transmitting node always transmits at the full rate of the channel, namely, R bps.
- When there is a collision, each node involved in the collision repeatedly retransmits its frame(that is ,packet) until the frame gets through without a collision. But when a node experiences a collision, it doesn't necessarily retransmitting the frame right away. Instead it waits a random delay before retransmitting the frame.
- Each node involved in a collision chooses independent random delays .Because the random delays are independently chosen, it is possible that one of the nodes will pick a delay that is sufficiently less than the delays of the other colliding nodes and will therefore be able to sneak its frame into the channel without a collision.

The most commonly used random access protocols

1. The ALOHA protocol ,
2. CSMA (carrier sense multiple access) protocol ,
3. CSMA/CD (carrier sense multiple access /collision detection) protocol and
4. Collision-Free Protocols
5. Limited-Contention Protocols

1. ALOHA

- In the 1970s, Norman Abramson and his colleagues at the University of Hawaii devised a new and elegant
  method to solve the channel allocation problem.
- Although Abramson's work, called the ALOHA system, used ground-based radio broadcasting, the basic idea is applicable to any system in which uncoordinated users are competing for the use of a single shared channel.
- The two versions of ALOHA here: **pure and slotted**.
- They differ with respect to whether time is divided into discrete slots into which all frames must fit.
- Pure ALOHA does not require global time synchronization; slotted ALOHA does.

PURE ALOHA

- If you have data to send, send the data
- If the message collides with another transmission, try resending "later"

Note that the first step implies that Pure ALOHA does not check whether the channel is busy before transmitting. The critical aspect is the "later" concept: the quality of the backoff scheme chosen significantly influences the efficiency of the protocol, the ultimate channel capacity, and the predictability of its behavior.

A sketch of frame generation in an ALOHA system is given in Fig. 4-1. We have made the frames all the same length because the throughput of ALOHA systems is maximized by having a uniform frame size rather than by allowing variable length frames.

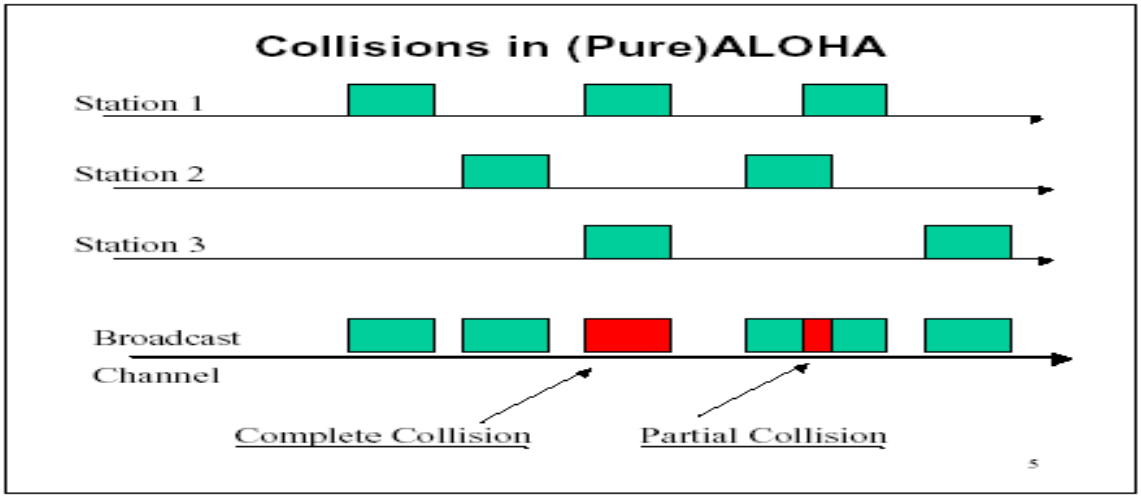
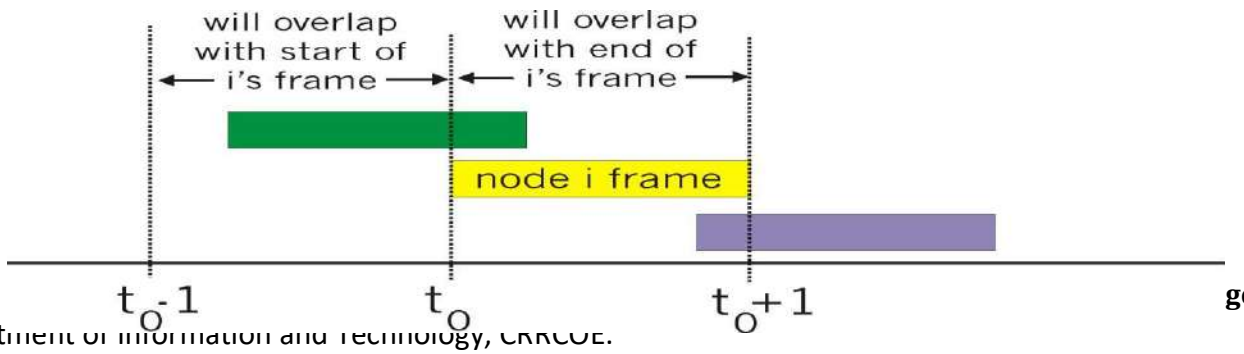


Figure 4-1. In pure ALOHA, frames are transmitted at completely arbitrary times.

To assess Pure ALOHA, we need to predict its throughput, the rate of (successful) transmission of frames. First, let's make a few simplifying assumptions:

- All frames have the same length.
- Stations cannot generate a frame while transmitting or trying to transmit.
- The population of stations attempts to transmit (both new frames and old frames that collided) according to a Poisson distribution.

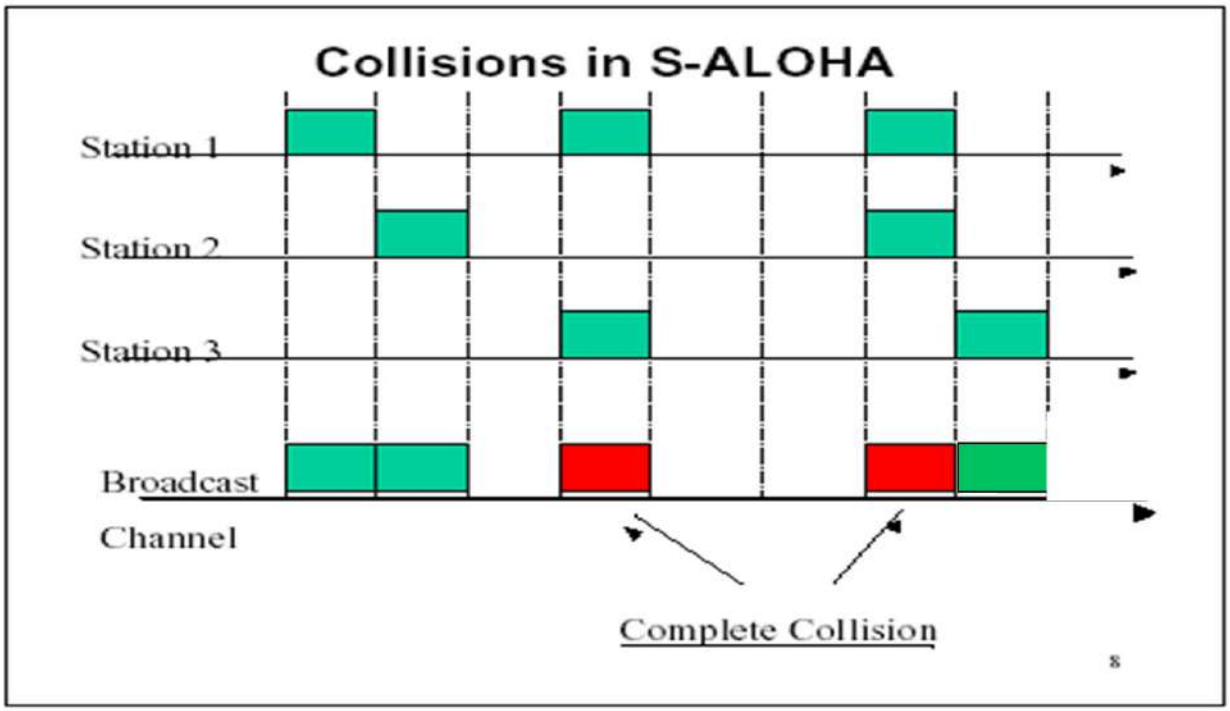
Let "*T*" refer to the time needed to transmit one frame on the channel, and let's define "frame-time" as a unit of time equal to *T*. Let "*G*" refer to the mean used in the Poisson distribution over transmission-attempt amounts: that is, on average, there are *G* transmission-attempts per frame-time.



.

SLOTTED ALOHA

- An improvement to the original ALOHA protocol was "Slotted ALOHA", which introduced discrete timeslots and increased the maximum throughput.
- A station can send only at the beginning of a timeslot, and thus collisions are reduced. In this case, we only need to worry about the transmission-attempts within 1 frame-time and not 2 consecutive frame-times, since collisions can only occur during each timeslot. Thus, the probability of there being zero transmission-attempts in a single timeslot is:



- Slotted ALOHA is used in low-data-rate tactical satellite communications networks by military forces, in subscriber-based satellite communications networks, mobile telephony call setup, and in the contactless RFID technologies.

Pros

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons

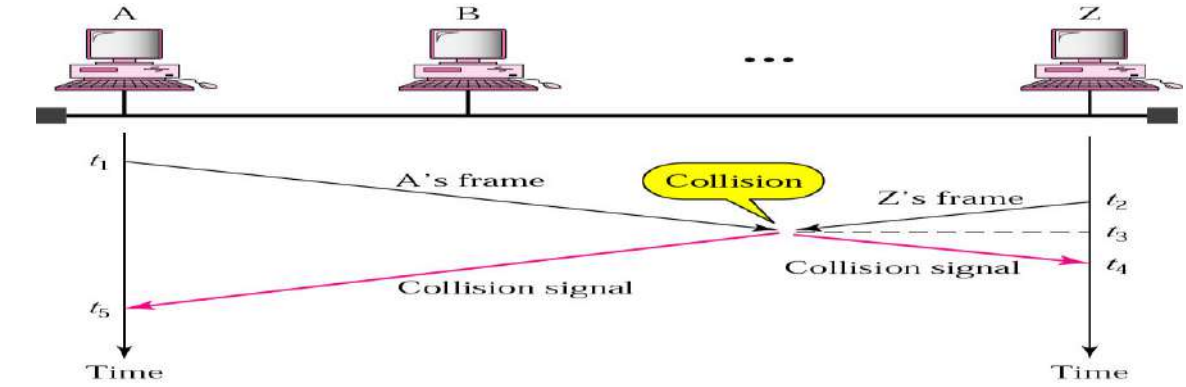
- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

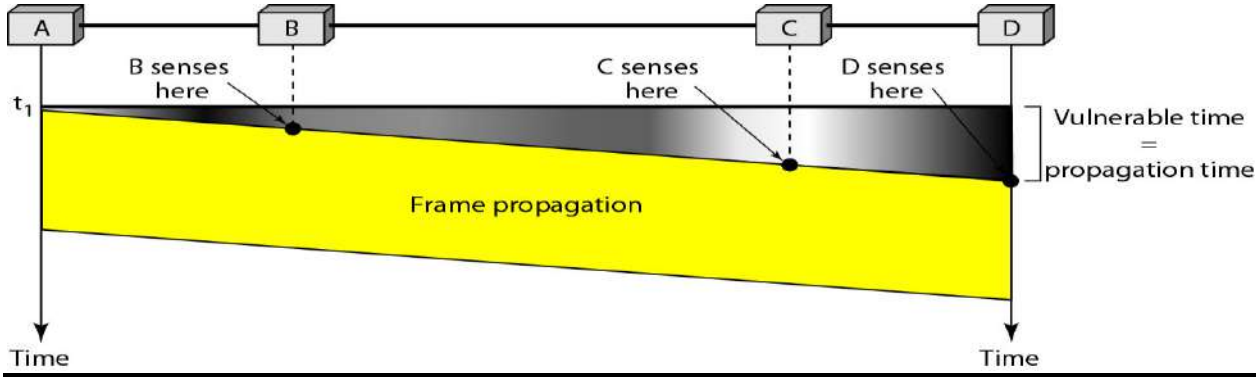
CARRIER SENSE MULTIPLE ACCESS

Carrier Sense Multiple Access (**CSMA**) is a probabilistic Media Access Control (MAC) protocol in which a node verifies the absence of other traffic before transmitting on a shared transmission medium, such as an electrical bus, or a band of the electromagnetic spectrum.

"**Carrier Sense**" describes the fact that a transmitter uses feedback from a receiver that detects a carrier wave before trying to send. That is, it tries to detect the presence of an encoded signal from another station before attempting to transmit. If a carrier is sensed, the station waits for the transmission in progress to finish before initiating its own transmission.

"**Multiple Access**" describes the fact that multiple stations send and receive on the medium. Transmissions by one node are generally received by all other stations using the medium.





ADVANTAGES

- Fairly simple to implement
- Functional scheme that works

DISADVANTAGES

- Cannot recover from a collision (inefficient waste of medium time)

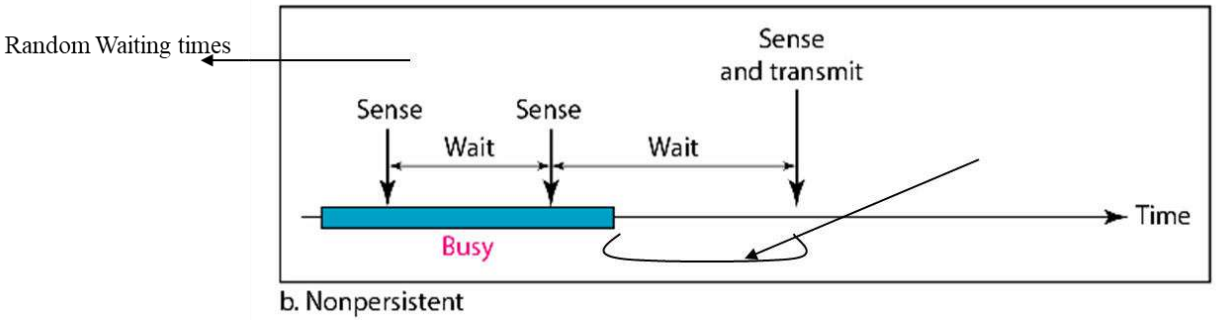
Types of CSMA Protocols :

Different CSMA protocols that determine:

- What a station should do when the medium is idle?
- What a station should do when the medium is busy?
    1. Non-Persistent CSMA
    2. 1-Persistent CSMA
    3. p-Persistent CSMA

1) Nonpersistent CSMA :

- A station with frames to be sent, should sense the medium
    1. If medium is idle, transmit; otherwise, go to 2
    2. If medium is busy, (backoff) wait a *random* amount of time and repeat 1
- Non-persistent Stations are deferential (respect others)
- Performance:
    1. Random delays reduce probability of collisions because two stations with data to be transmitted will wait for different amount of times.
    2. Bandwidth is wasted if waiting time (backoff) is large because medium will remain idle following end of transmission even if one or more stations have frames to send

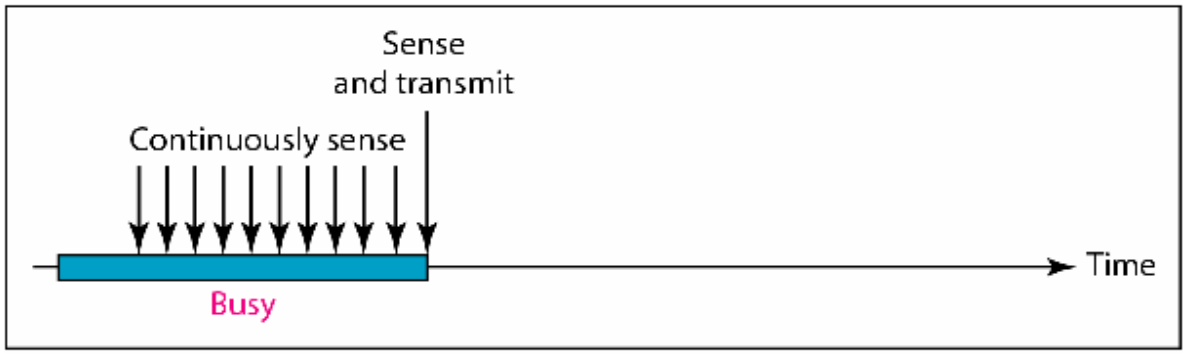


b. Nonpersistent

2) 1-persistent CSMA: To avoid idle channel time, 1-persistent protocol used

- Station wishing to transmit listens to the medium:
    1. If medium idle, **transmit** immediately;
    2. If medium busy, **continuously listen** until medium becomes idle; then transmit immediately with probability 1

- Performance
  - 1-persistent stations are **selfish**
  - If two or more stations becomes ready at the same time, **collision guaranteed**

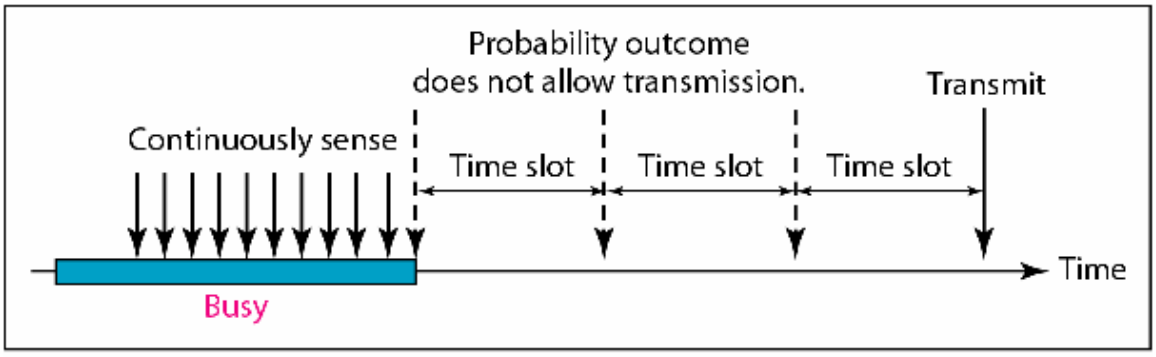


a. 1-persistent

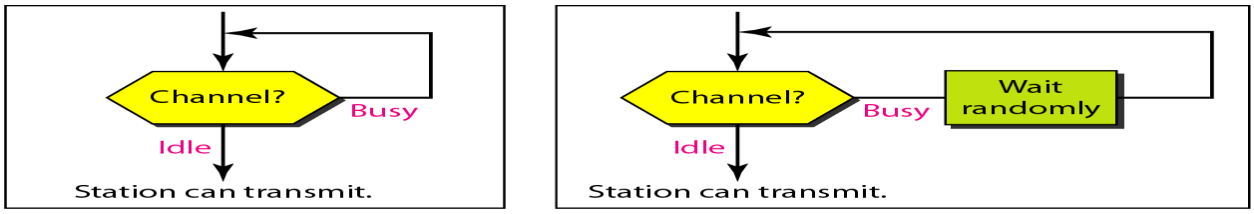
3).P-persistent CSMA: Time is divided to slots where each Time unit (slot) typically equals **maximum propagation delay**

- Station wishing to transmit listens to the medium:

1. If medium idle,
   - transmit with probability (**p**), OR
   - wait **one time unit (slot)** with probability (**1 – p**), then repeat 1.
2. If medium busy, **continuously listen until idle** and repeat step **1**
3. Performance
   - Reduces the possibility of collisions like **nonpersistent**
   - Reduces channel idle time like **1-persistent**

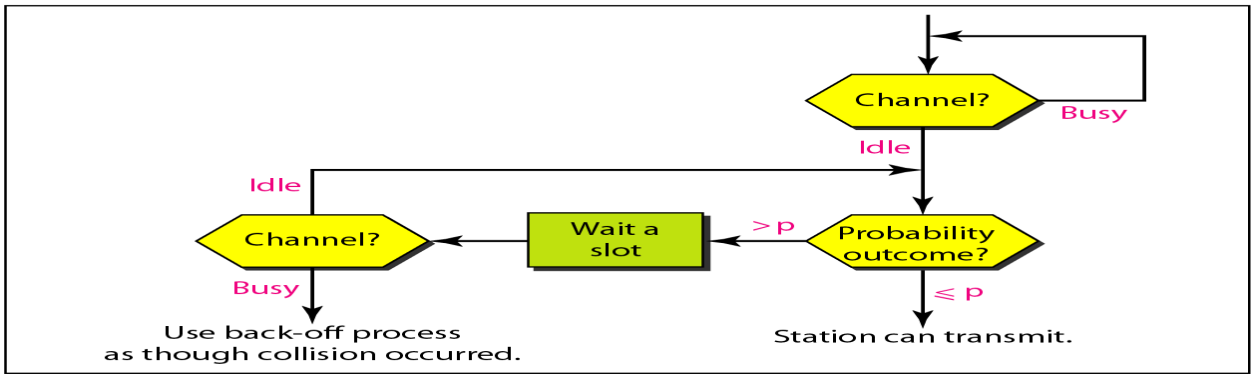


c. p-persistent



a. 1-persistent

b. Nonpersistent



c. p-persistent

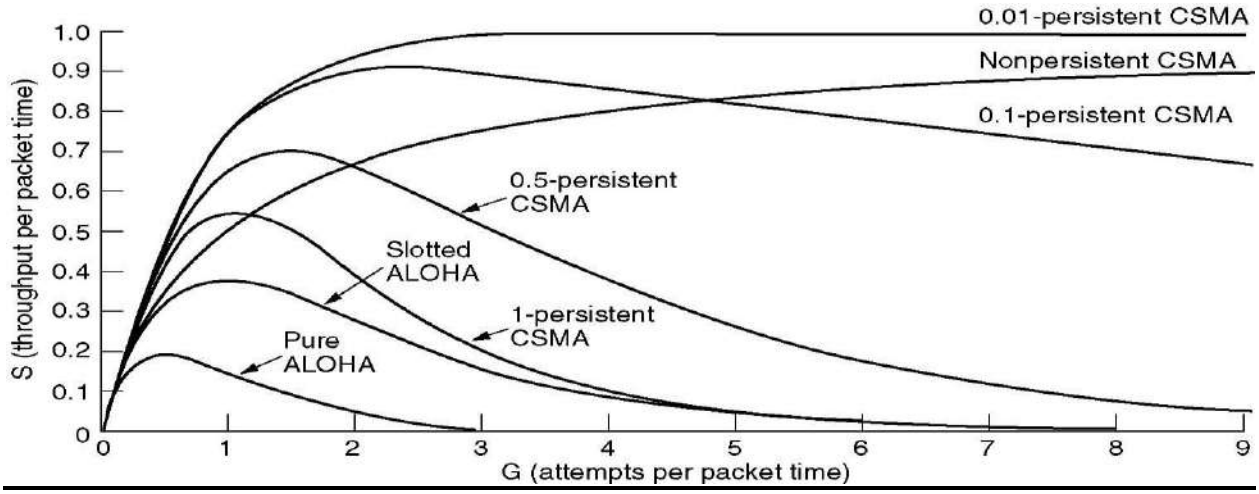


Fig: Comparison of the channel utilization versus load for various random access protocols.

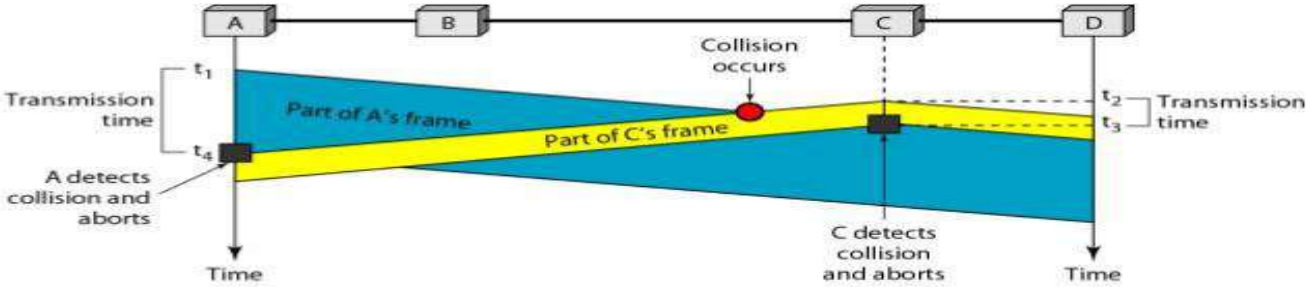
CSMA/CD (Collision Detection)

Carrier sense multiple access with collision detection (**CSMA/CD**) is a Media Access Control method in which.

- a carrier sensing scheme is used.
- a transmitting data station that detects another signal while transmitting a frame, stops transmitting that frame, transmits a **jam signal**, and then waits for a random time interval before trying to resend the frame.
- CSMA/CD is a modification of pure carrier sense multiple access (CSMA). CSMA/CD is used to improve CSMA performance by terminating transmission as soon as a collision is detected, thus shortening the time required before a retry can be attempted.

CSMA/CD

- Carrier Sense Multiple Access with Collision Detection
- Station monitors channel when sending a frame



ALGORITHM : The following procedure is used to initiate a transmission. The procedure is complete when the frame is transmitted successfully or a collision is detected during transmission

When a station wants to send some information, it uses the following algorithm.

Main procedure

1. Is a frame ready for transmission? If not, wait for a frame.
2. Is medium idle? If not, wait until it becomes ready
3. Start transmitting and monitor for collision during transmission.
4. Did a collision occur? If so, go to collision detected procedure.
5. Reset retransmission counters and end frame transmission.

Collision detected procedure:

The following procedure is used to resolve a detected collision. The procedure is complete when retransmission is initiated or the retransmission is aborted due to numerous collisions.

1. Continue transmission until minimum packet time is reached to ensure that all receivers detect the collision.
2. Increment retransmission counter.
3. Was the maximum number of transmission attempts reached? If so, abort transmission.
4. Calculate and wait random backoff period based on number of collisions.
5. Re-enter main procedure at stage 1.

Methods for collision detection are media dependent, but on an electrical bus such as 10BASE-5 or 10BASE-2, collisions can be detected by comparing transmitted data with received data or by recognizing a higher than normal signal amplitude on the bus.

JAM SIGNAL

The **jam signal** is a signal that carries a 32-bit binary pattern sent by a data station to inform the other stations that they must not transmit.

ADVANTAGES

More efficient than basic CSMA

DISADVANTAGES

Requires ability to detect collisions

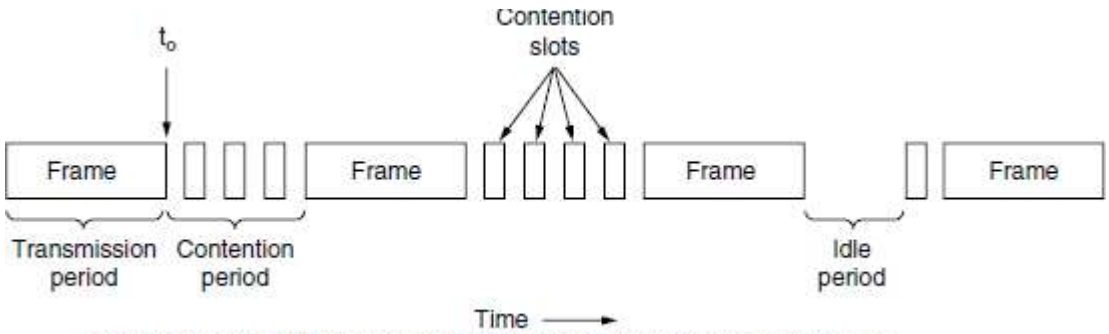


Figure 4-5. CSMA/CD can be in contention, transmission, or idle state.

Collision-Free Protocols

Binary Countdown

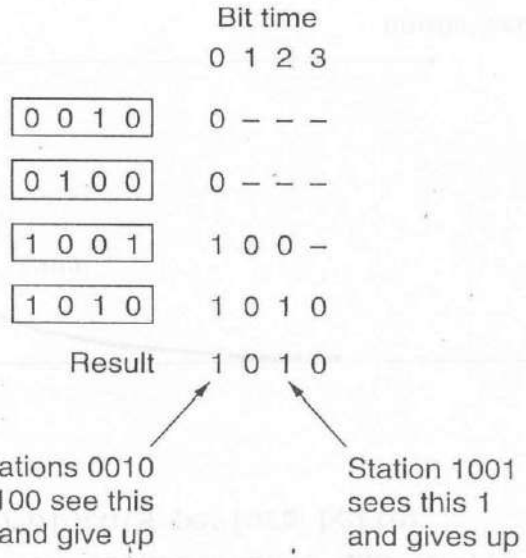


Fig. The binary countdown protocol. A dash indicates silence.

Binary Countdown Protocol

- Binary station addresses are used. The one with the highest address wins.
- The channel efficiency of this method is d/(d+lnN). The overhead per frame is N. The amount of data is d bits.
- If the frame format has been cleverly chosen so that the sender's address is the first field in the frame, even these ln N bits are not wasted, and the efficiency is 100 percent.
- Various binary countdown protocols have been proposed. Some use parallel rather than serial interface. Some use virtual station numbers.

Wireless LAN Protocols

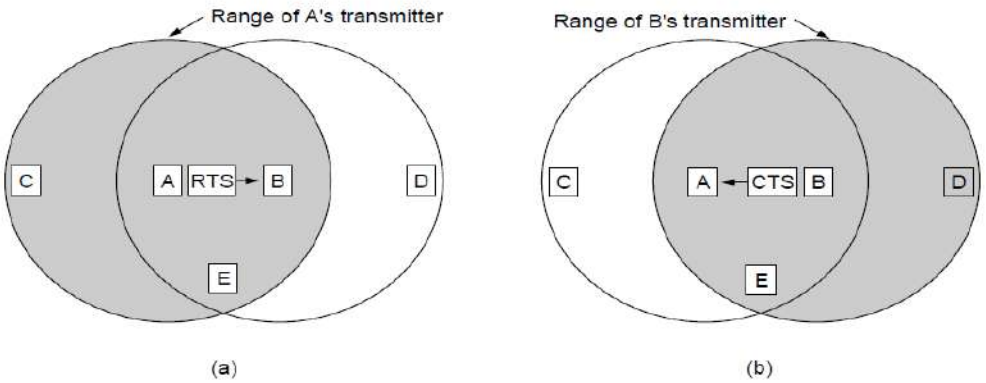
MACA and MACAW

An early protocol designed for wireless LANs is **MACA (Multiple Access with Collision Avoidance)** (Karn, 1990). It was used as the basis for the IEEE 802.11 wireless LAN standard. 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 themselves for the duration of the upcoming (large) data frame. MACA is illustrated in Fig.

Let us consider how *A* sends a frame to *B*. *A* starts 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.

Now let us see how stations overhearing either of these frames react. Any station hearing the RTS is clearly close to *A* and must remain silent long enough for the CTS to be transmitted back to *A* without conflict. Any station hearing the CTS is clearly close to *B* and must remain silent during the upcoming data transmission, whose length it can tell by examining the CTS frame.

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



(a) (b)

Ethernet

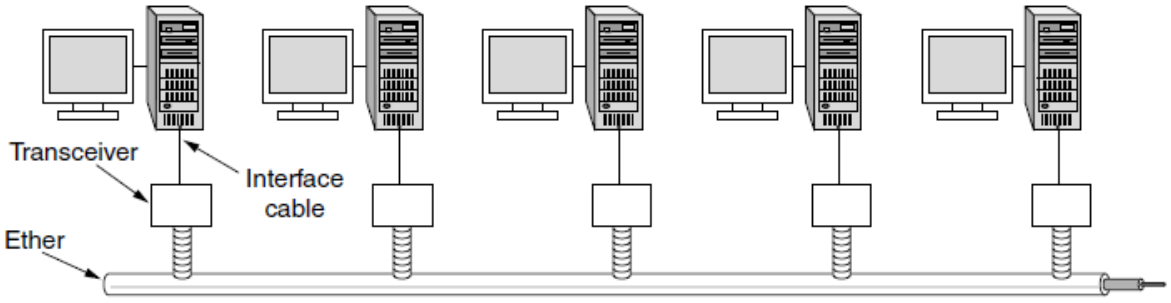


Fig : Architecture of classic Ethernet

Ethernet :

Ethernet is a family of computer networking technologies commonly used in local area networks (LAN), metropolitan area networks (MAN) and wide area networks (WAN). It was commercially introduced in 1980 and first standardized in 1983 as IEEE 802.3. Ethernet has since retained a good deal of backward compatibility and has been refined to support higher bit rates, a greater number of nodes, and longer link distances. Over time, Ethernet has largely replaced competing wired LAN technologies such as Token Ring, FDDI(**Fiber Distributed Data Interface**) (**FDDI**). and ARCNET. **(Attached Resource Computer NETwork** (**ARCNET** or **ARCnet**))

The original 10BASE5 Ethernet uses coaxial cable as a shared medium, while the newer Ethernet variants use twisted pair and fiber optic links in conjunction with switches. Over the course of its history, Ethernet data transfer rates have been increased from the original 2.94 megabits per second (Mbit/s) to the latest 400 gigabits per second (Gbit/s). The Ethernet standards comprise several wiring and signaling variants of the OSI physical layer in use with Ethernet.

Switched Ethernet : switched Ethernet. An Ethernet LAN that uses switches to connect individual hosts or segments. In the case of individual hosts, the switch replaces the repeater and effectively gives the device full 10 Mbps bandwidth (or 100 Mbps for Fast Ethernet) to the rest of the network

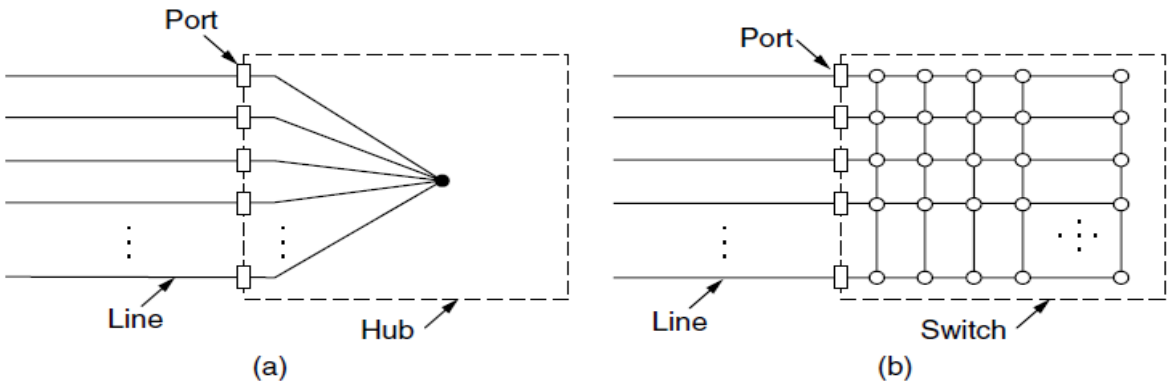


Fig : (a) Hub. (b) Switch.

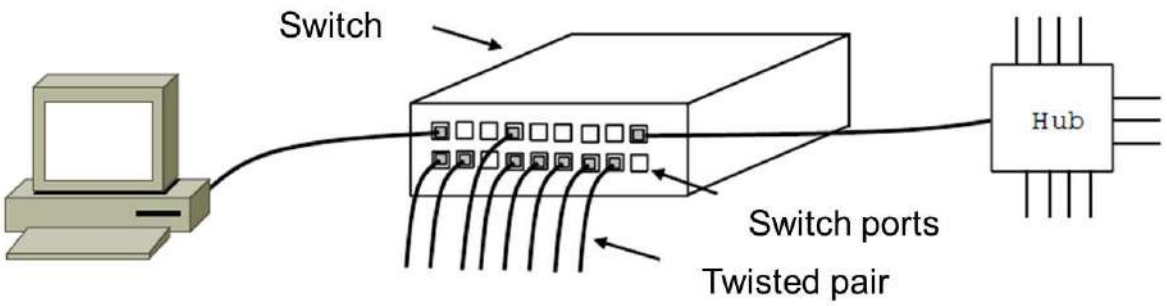


Fig : An Ethernet switch

Fast Ethernet : Fast Ethernet physical layers carry traffic at the nominal rate of 100 Mbit/s. The prior Ethernet speed was 10 Mbit/s. Of the Fast Ethernet physical layers, 100BASE-TX is by far the most common.

Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps (Cat 5 UTP)
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

Table: The original fast Ethernet cabling

Gigabit Ethernet : Gigabit Ethernet, a transmission technology based on the Ethernet frame format and protocol used in local area networks (LANs), provides a data rate of 1 billion bits per second (one gigabit). Gigabit Ethernet is defined in the IEEE 802.3 standard and is currently being used as the backbone in many enterprise networks

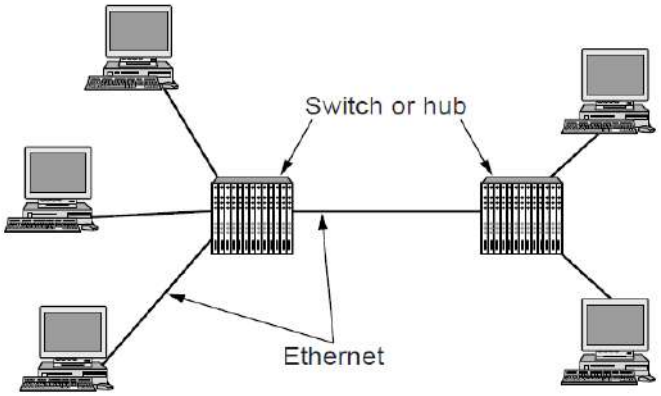


Fig :A two-station Ethernet

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

Fig: Gigabit Ethernet cabling

.

Wireless LANs : A wireless LAN (WLAN) is a wireless computer network that links two or more devices using wireless communication to form a local area network (LAN) within a limited area such as a home, school, computer laboratory, campus, or office building. This gives users the ability to move around within the area and remain connected to the network. Through a gateway, a WLAN can also provide a connection to the wider Internet. Most modern WLANs are based on IEEE 802.11 standards and are marketed under the Wi-Fi brand name.

802.11 Architecture and Protocol Stack :

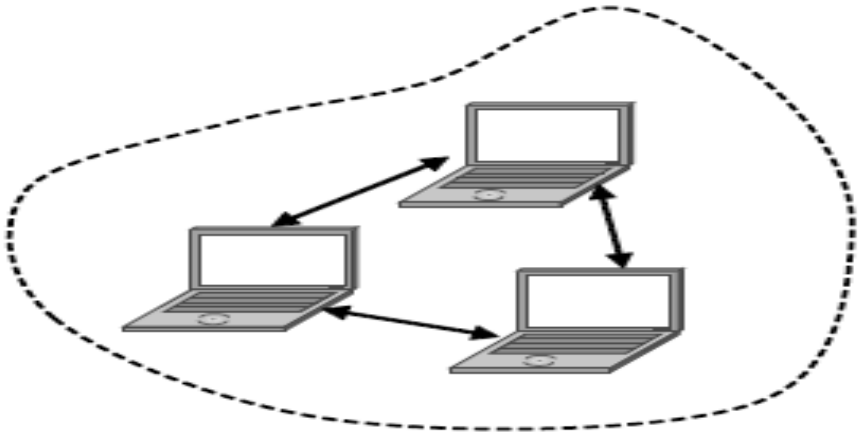


Fig : 802.11 architecture – ad-hoc mode

802.11 Architecture and Protocol Stack: The 802.11 Protocol Stack Part of the 802.11 protocol stack. 12. Wireless Physical Layer 802.11 Infrared – Two capacities 1 Mbps or 2 Mbps. Wireless Physical Layer 802.11 DSSS (Direct Sequence Spread Spectrum) – Spreads signal over entire spectrum using pseudo-random sequence .

orthogonal frequency-division multiplexing (OFDM)
first Wi-Fi standard that introduced MIMO (Multiple-Input and Multiple-Output)

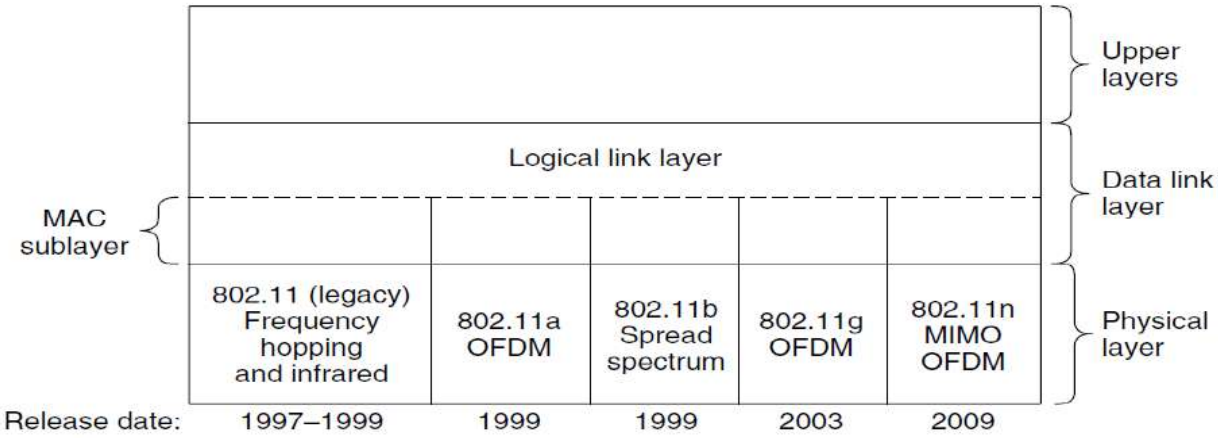


Fig: Part of the 802.11 protocol stack.

The 802.11 MAC Sublayer Protocol : The 802.11 MAC Sublayer Protocol. IEEE 802.11 standard, popularly known as WiFi, lays down the architecture and specifications of wireless LANs (WLANs). The 802.11 MAC sublayer provides an abstraction of the physical layer to the logical link control sublayer and upper layers of the

OSI network.

The hidden terminal problem : In wireless networking, the hidden node problem or hidden terminal problem occurs when a node can communicate with a wireless access point (AP), but cannot directly communicate with other nodes that are communicating with that AP. ... Practical protocol solutions exist to the hidden node problem.

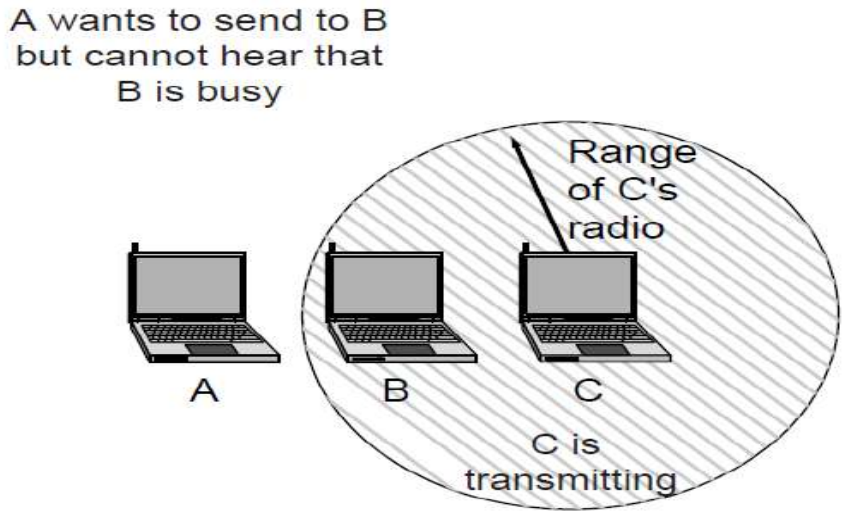


Fig : The hidden terminal problem

The exposed terminal problem: The Exposed Terminal Problem. In wireless LANs (wireless local area networks), the exposed terminal problem is a transmission problem that arises when a transmitting station is prevented from sending frames due to interference with another transmitting station.

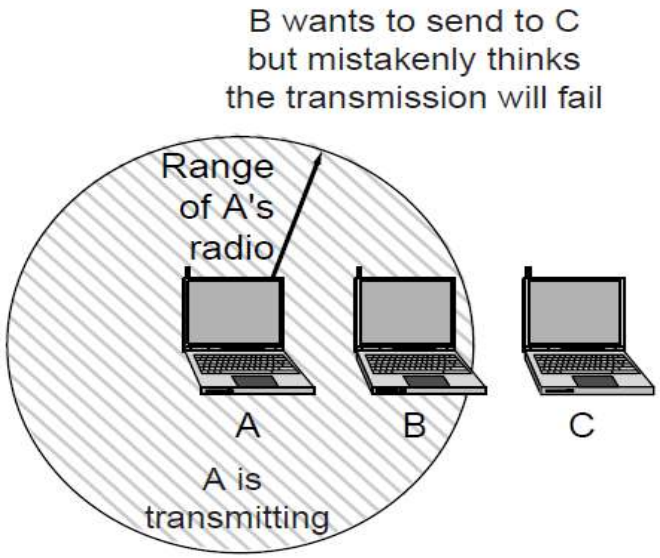


Fig: The exposed terminal problem

The use of virtual channel sensing using CSMA/CA : Virtual channel sense or virtual carrier sense is a mechanism to predict future traffic in wireless networks that uses carrier sense multiple access with collision

avoidance (CSMA/CA). It is implemented in wireless network protocols, IEEE 802.11 and IEEE 802.16, which operates in the medium access control (MAC) layer

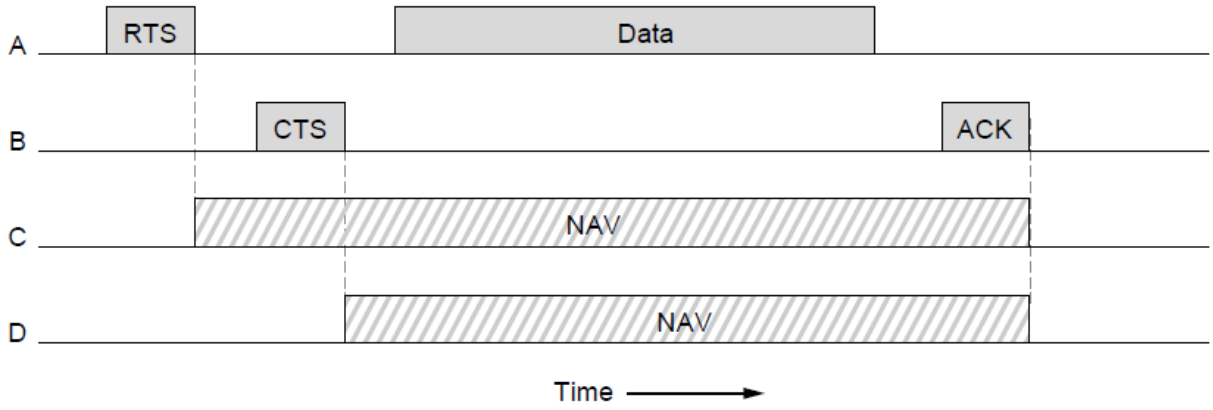


Fig: The use of virtual channel sensing using CSMA/CA

Interframe spacing in 802.11: In IEEE 802.11 spacing is used to separate frames. The length of interframe space determines when the channel can be accessed. Thus, the interframe spacing is used to set prioritized access to the channel.

Short Interframe Space (SISF), Extended inter-frame spacing (**EIFS**)

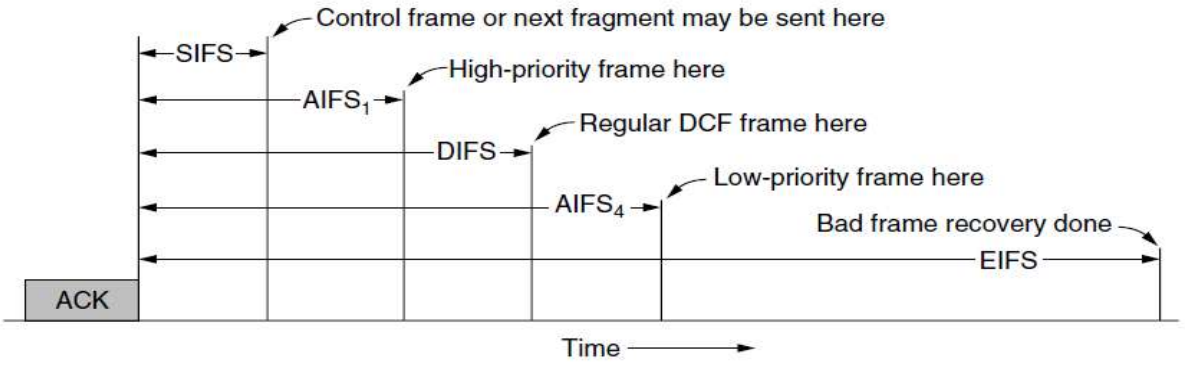
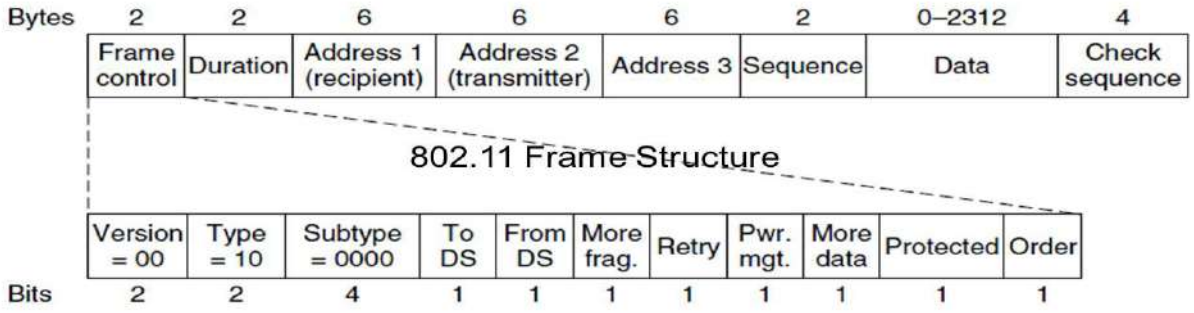


Fig: Interframe spacing in 802.11

802.11 Frame Structure : The 802.11 Frame Structure. The IEEE 802.11 standard, lays down the architecture and specifications of wireless local area networks (WLANs). WLAN or WiFi uses high frequency radio waves instead of cables for connecting the devices in LAN. Users connected by WLANs can move around within the area of network coverage



UNIT-V
NETWORK LAYER

Network layer – design issues – Routing algorithms - The Optimality Principle - Shortest Path Algorithm – Flooding - Distance Vector Routing - Link State Routing - Hierarchical Routing - Broadcast Routing - Multicast Routing Congestion Control – Approaches - Traffic-Aware Routing

1. Explain the design issues of the Network layer.

Discuss the services provided to the transport layer by the Network layer. /

How Connection Oriented and Connection Less Services are implemented? Explain. /

Compare the virtual circuits and datagram within the subnet.

2. Discuss about different routing algorithms in detail. /

Discuss shortest path routing. /

What is flooding? Discuss. /

Differentiate and explain adaptive and nonadaptive routing algorithms. /

Describe hierarchical Broadcast and Multicasting routing.

3. Discuss about different congestion control algorithms in detail /

Discuss the reasons for occurrence of congestion. Suggest some algorithms to control congestion.

1. Explain the design issues of the Network layer. /

Discuss the services provided to the transport layer by the Network layer. /

How Connection Oriented and Connection Less Services are implemented? Explain. /

Compare the virtual circuits and datagram within the subnet.

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.

STORE-AND-FORWARD PACKET SWITCHING

- The network layer protocols operation can be seen in Fig. 5-1.
- The major components of the system are the carrier's equipment (routers connected by transmission lines), shown inside the shaded oval.
- 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.

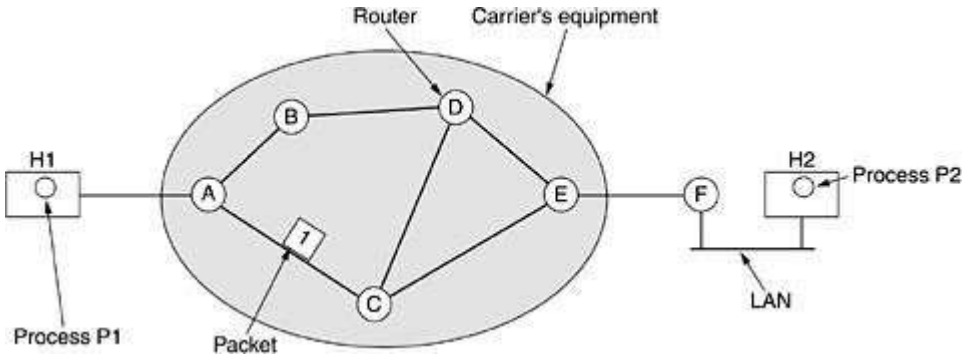


Figure 5-1. The environment of the network layer protocols.

- 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.
- This mechanism is **store-and-forward packet switching**.

SERVICES PROVIDED TO THE TRANSPORT LAYER

The network layer provides services to the transport layer at the network layer/transport layer interface. 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.

- There is a discussion centers on whether the network layer should provide connection-oriented service or connectionless service.
  - In their view (based on 30 years of actual experience with a real, working computer network), the subnet is inherently unreliable, no matter how it is designed. Therefore, the hosts should accept the fact that the network is unreliable and do error control (i.e., error detection and correction) and flow control themselves.
  - This viewpoint leads quickly to the conclusion that the network service should be connectionless, with primitives SEND PACKET and RECEIVE PACKET and little else.
  - In particular, no packet ordering and flow control should be done, because the hosts are going to do that anyway, and there is usually little to be gained by doing it twice.
  - Furthermore, each packet must carry the full destination address, because each packet sent is carried independently of its predecessors, if any.
- The other camp (represented by the telephone companies) argues that the subnet should provide a reliable, connection-oriented service.
- These two camps are best exemplified by the Internet and ATM. The Internet offers connectionless network-layer service; ATM networks offer connection-oriented network-layer service. However, it is interesting to note that as quality-of-service guarantees are becoming more and more important, the Internet is evolving. In particular, it is starting to acquire properties normally associated with connection-oriented service, as we will see later. Actually, we got an inkling of this evolution during our study of VLANs in Chap. 4.

IMPLEMENTATION OF CONNECTIONLESS SERVICE

- Two different organizations are possible, depending on the type of service offered.
- 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**.

- 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**.
- Let us now see how a datagram subnet works. Suppose that the process *P1* in **Fig. 5-2** 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.

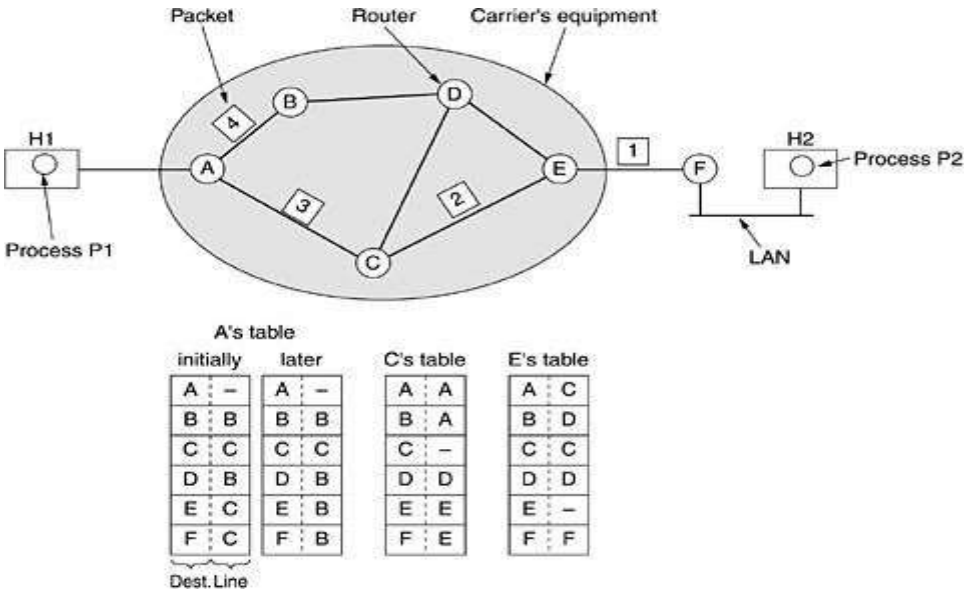


Figure 5-2. Routing within a datagram subnet

- 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,
- For example, PPP. 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. 5-2, *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**.

IMPLEMENTATION OF CONNECTION-ORIENTED SERVICE

- For connection-oriented service, we need a virtual-circuit subnet.
- Let us see how that works.
- The idea behind virtual circuits is to avoid having to choose a new route for every packet sent, as in Fig. 5-2. 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. 5-3**. 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.

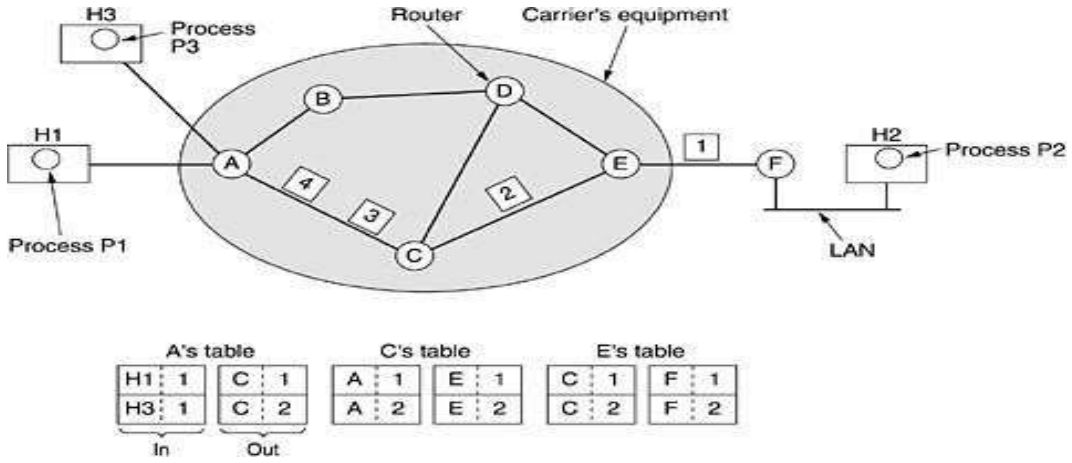


Figure 5-3. Routing within a virtual-circuit subnet.

- 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.

COMPARISON OF VIRTUAL-CIRCUIT AND DATAGRAM SUBNETS

The major issues are listed in Fig. 5-4, although purists could probably find a counter example for everything in the figure.

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

Figure 5-4. Comparison of datagram and virtual-circuit subnets.

2. Discuss about different routing algorithms in detail. (or)

Discuss shortest path routing. (Or)

What is flooding? Discuss. (Or)

Differentiate and explain adaptive and nonadaptive routing algorithms. (Or)

Describe hierarchical Broadcast and Multicasting routing.

(Nov'11, May'10, Dec'08, Nov'07, Dec'05, Dec'04)

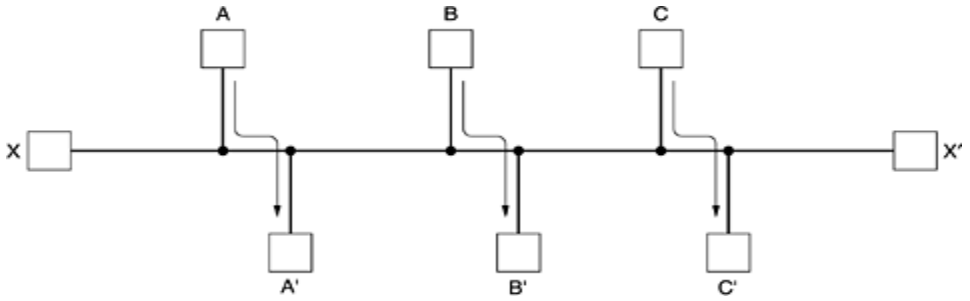
ROUTING ALGORITHMS

The **routing algorithm** is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.

PROPERTIES OF ROUTING ALGORITHM:

Correctness, simplicity, robustness, stability, fairness, and optimality

FAIRNESS AND OPTIMALITY.



Fairness and optimality may sound obvious, but as it turns out, they are often contradictory goals. There is enough traffic between A and A', between B and B', and between C and C' to saturate the horizontal links. To maximize the total flow, the X to X' traffic should be shut off altogether. Unfortunately, X and X' may not see it that way. Evidently, some compromise between global efficiency and fairness to individual connections is needed.

CATEGORY OF ALGORITHM

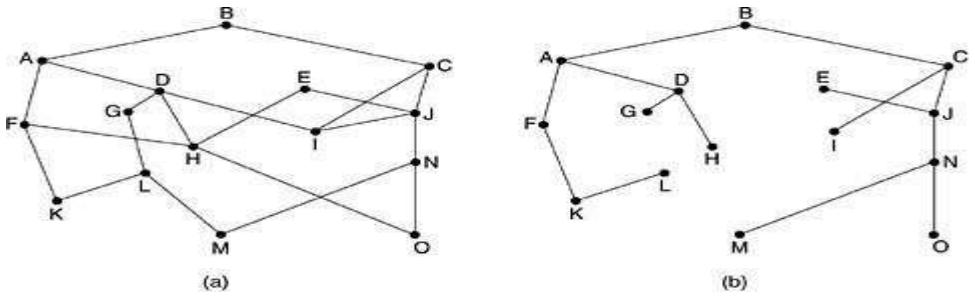
- Routing algorithms can be grouped into two major classes: **nonadaptive and adaptive.**
- **Nonadaptive algorithms** do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off-line, and downloaded to the routers when the network is booted.
- This procedure is sometimes called **Static routing**.

- **Adaptive algorithms**, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well
- This procedure is sometimes called **dynamic routing**

THE OPTIMALITY PRINCIPLE

(a) **I**f router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.

(b) The set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a sink tree.

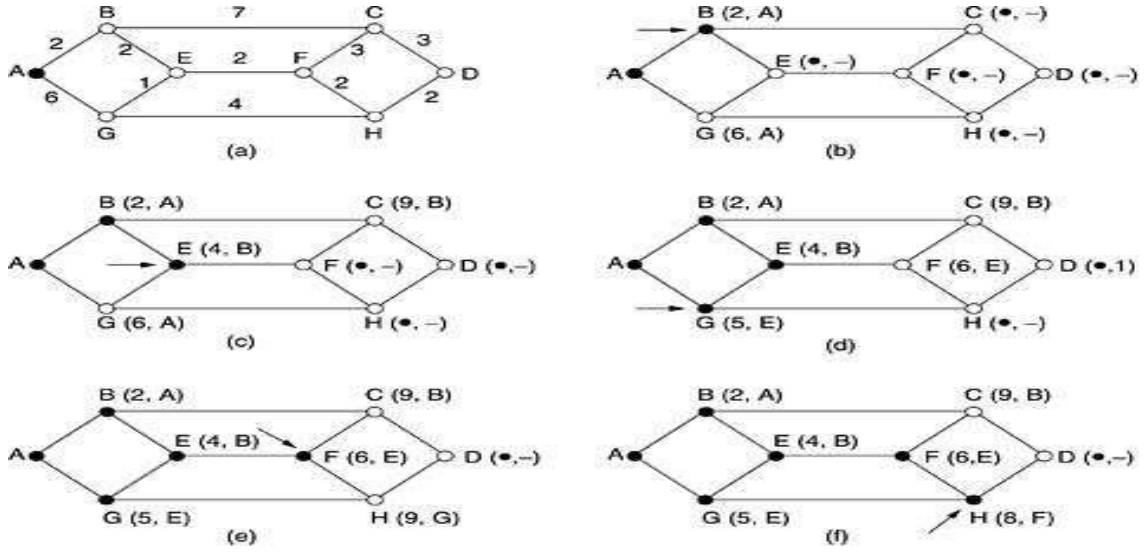


***(c) A subnet. (b) A sink tree for router* B.**

- As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination.
- Such a tree is called a **sink tree** where the distance metric is the number of hops. Note that a sink tree is not necessarily unique; other trees with the same path lengths may exist.
- The goal of all routing algorithms is to discover and use the sink trees for all routers.

SHORTEST PATH ROUTING

- A technique to study routing algorithms: The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line (often called a link).
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- One way of measuring path length is the number of hops. Another metric is the geographic distance in kilometers. Many other metrics are also possible. For example, each arc could be labeled with the mean queuing and transmission delay for some standard test packet as determined by hourly test runs.
- In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria.



The first five steps used in computing the shortest path from* A *to* D. *The arrows indicate the working node.

- To illustrate how the labelling algorithm works, look at the weighted, undirected graph of Fig. 5-7(a), where the weights represent, for example, distance.
- We want to find the shortest path from *A* to *D*. We start out by marking node *A* as permanent, indicated by a filled-in circle.
- Then we examine, in turn, each of the nodes adjacent to *A* (the working node), relabeling each one with the distance to *A*.

- Whenever a node is relabelled, we also label it with the node from which the probe was made so that we can reconstruct the final path later.
- Having examined each of the nodes adjacent to *A*, we examine all the tentatively labelled nodes in the whole graph and make the one with the smallest label permanent, as shown in Fig. 5-7(b).
- This one becomes the new working node.

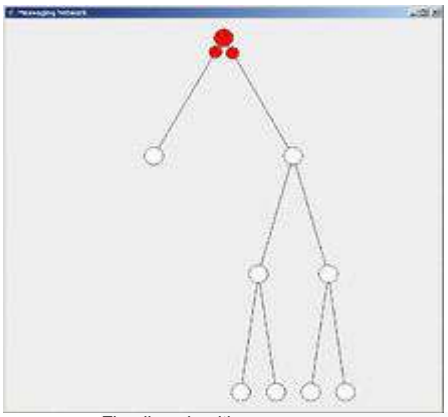
We now start at *B* and examine all nodes adjacent to it. If the sum of the label on *B* and the distance from *B* to the node being considered is less than the label on that node, we have a shorter path, so the node is relabelled.

After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively-labelled node with the smallest value. This node is made permanent and becomes the working node for the next round. Figure 5-7 shows the first five steps of the algorithm.

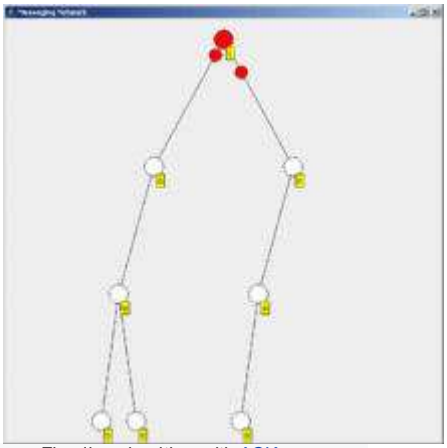
- To see why the algorithm works, look at Fig. 5-7(c). At that point we have just made *E* permanent. Suppose that there were a shorter path than *ABE*, say *AXYZE*. There are two possibilities: either node *Z* has already been made permanent, or it has not been. If it has, then *E* has already been probed (on the round following the one when *Z* was made permanent), so the *AXYZE* path has not escaped our attention and thus cannot be a shorter path.
- Now consider the case where *Z* is still tentatively labelled. Either the label at *Z* is greater than or equal to that at *E*, in which case *AXYZE* cannot be a shorter path than *ABE*, or it is less than that of *E*, in which case *Z* and not *E* will become permanent first, allowing *E* to be probed from *Z*.
- This algorithm is given in Fig. 5-8. The global variables *n* and *dist* describe the graph and are initialized before *shortest path* is called. The only difference between the program and the algorithm described above is that in Fig. 5-8, we compute the shortest path starting at the terminal node, *t*, rather than at the source node, *s*. Since the shortest path from *t* to *s* in an undirected graph is the same as the shortest path from *s* to *t*, it does not matter at which end we begin (unless there are several shortest paths, in which case reversing the search might discover a different one). The reason for searching backward is that each node is labelled with its predecessor rather than its successor. When the final path is copied into the output variable, *path*, the path is thus reversed. By reversing the search, the two effects cancel, and the answer is produced in the correct order.

FLOODING

- Another static algorithm is **flooding**, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.
- There are generally two types of flooding available, **uncontrolled flooding and controlled flooding.**
- In **uncontrolled flooding** each node unconditionally distributes packets to each of its neighbors. Without conditional logic to prevent indefinite recirculation of the same packet.
- **Controlled flooding** has its own two algorithms to make it reliable, SNCF (**Sequence Number Controlled Flooding**) and RPF (**Reverse Path Forwarding**). In SNCF, the node attaches its own address and sequence number to the packet, since every node has a memory of addresses and sequence numbers. If it receives a packet in memory, it drops it immediately while in RPF, the node will only send the packet forward. If it is received from the next node, it sends it back to the sender.



Flooding algorithm

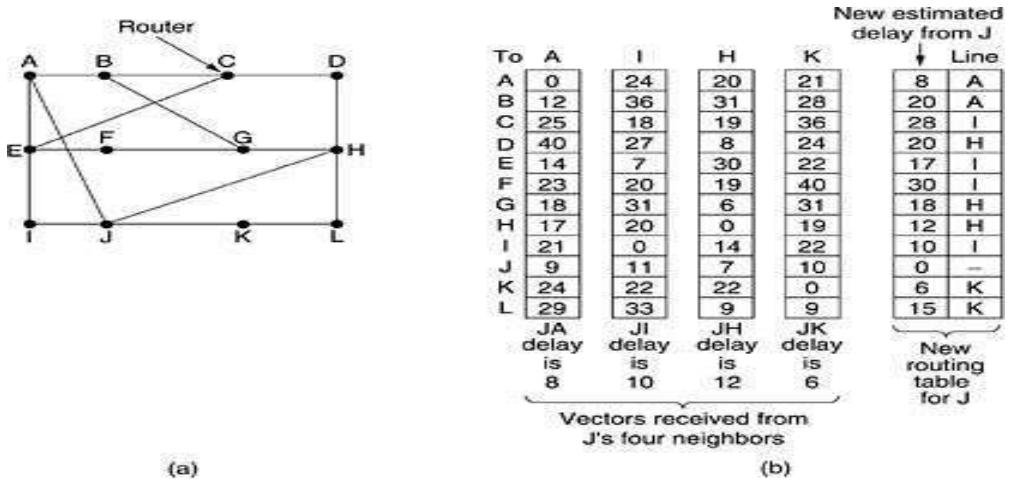


Flooding algorithm with ACK messages

DISTANCE VECTOR ROUTING

- **Distance vector routing** algorithms operate by having each router maintain a table (i.e, a vector) giving the best known distance to each destination and which line to use to get there.
- These tables are updated by exchanging information with the neighbors.
- The distance vector routing algorithm is sometimes called by other names, most commonly the distributed **Bellman-Ford** routing algorithm and the **Ford-Fulkerson** algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962).

- It was the original ARPANET routing algorithm and was also used in the Internet under the name RIP(**Routing Information Protocol**).



***(a) A subnet. (b) Input from* A, I, H, K, *and the new routing table for* J.**

- Part (a) shows a subnet. The first four columns of part (b) show the delay vectors received from the neighbours of router *J*.
- *A* claims to have a 12-msec delay to *B*, a 25-msec delay to *C*, a 40-msec delay to *D*, etc. Suppose that *J* has measured or estimated its delay to its neighbours, *A*, *I, H*, and *K* as 8, 10, 12, and 6 msec, respectively.

Each node constructs a one-dimensional array containing the "distances"(costs) to all other nodes and distributes that vector to its immediate neighbors.

1. The starting assumption for distance-vector routing is that each node knows the cost of the link to each of its directly connected neighbors.
2. A link that is down is assigned an infinite cost.

LINK STATE ROUTING

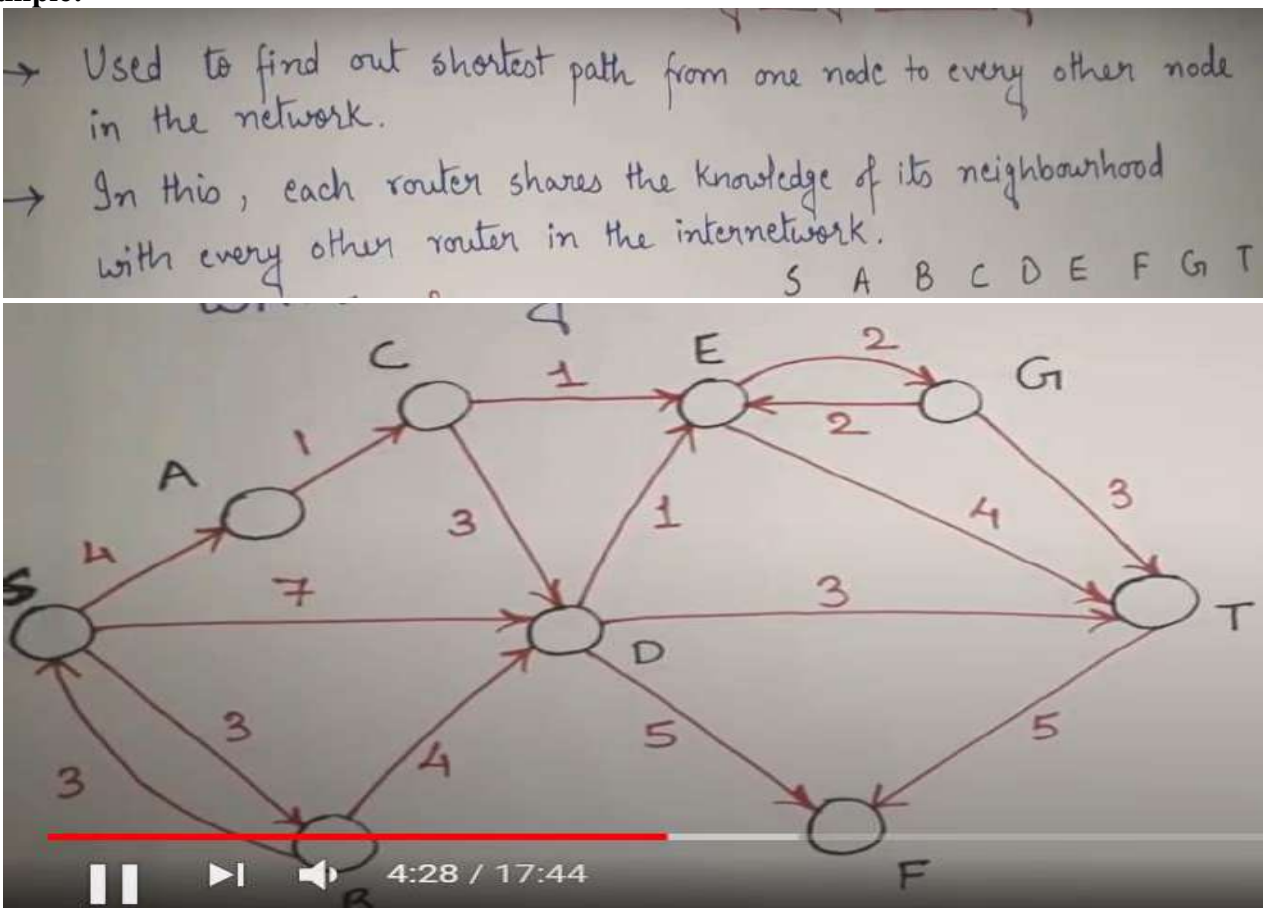
The idea behind link state routing is simple and can be stated as five parts. Each router must do the following:

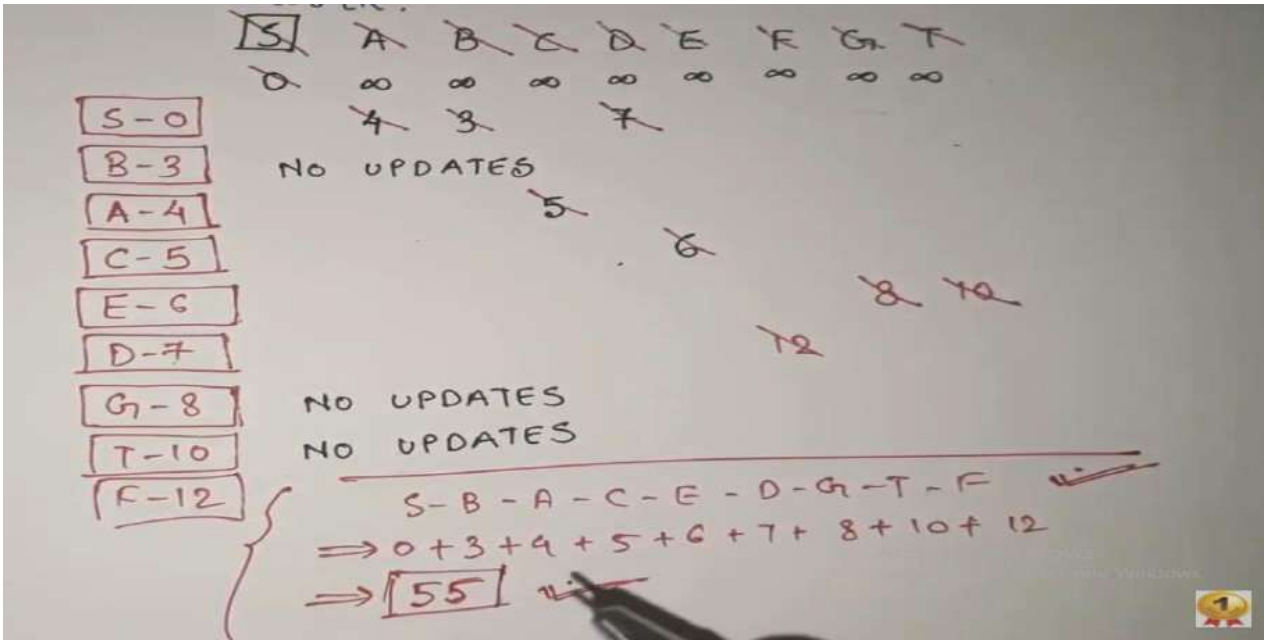
1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router

Learning about the Neighbours

When a router is booted, its first task is to learn who its neighbours are. It accomplishes this goal by sending a special HELLO packet on each point-to-point line. The router on the other end is expected to send back a reply telling who it is.

Example:



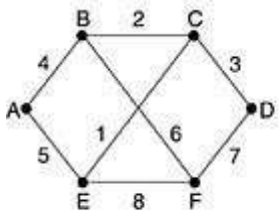


Measuring Line Cost

- The link state routing algorithm requires each router to know, or at least have a reasonable estimate of, the delay to each of its neighbors. The most direct way to determine this delay is to send over the line a special ECHO packet that the other side is required to send back immediately.

- By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay.
- For even better results, the test can be conducted several times, and the average used. Of course, this method implicitly assumes the delays are symmetric, which may not always be the case.

Building Link State Packets



(a)

Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

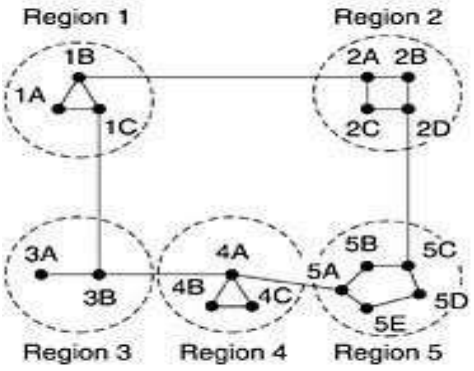
(b)

(a) A subnet. (b) The link state packets for this subnet.

- Once the information needed for the exchange has been collected, the next step is for each router to build a packet containing all the data.
- The packet starts with the identity of the sender, followed by a sequence number and age (to be described later), and a list of neighbours.
- For each neighbour, the delay to that neighbour is given.

HIERARCHICAL ROUTING

- The routers are divided into what we will call regions, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions.
- For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on, until we run out of names for aggregations.



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

- Figure 5-15 gives a quantitative example of routing in a two-level hierarchy with five regions.
- The full routing table for router 1*A* has 17 entries, as shown in Fig. 5-15(b).
- When routing is done hierarchically, as in Fig. 5-15(c), there are entries for all the local routers as before, but all other regions have been condensed into a single router, so all traffic for region 2 goes via the 1*B* -2*A* line, but the rest of the remote traffic goes via the 1*C* -3*B* line.
- Hierarchical routing has reduced the table from 17 to 7 entries. As the ratio of the number of regions to the number of routers per region grows, the savings in table space increase.

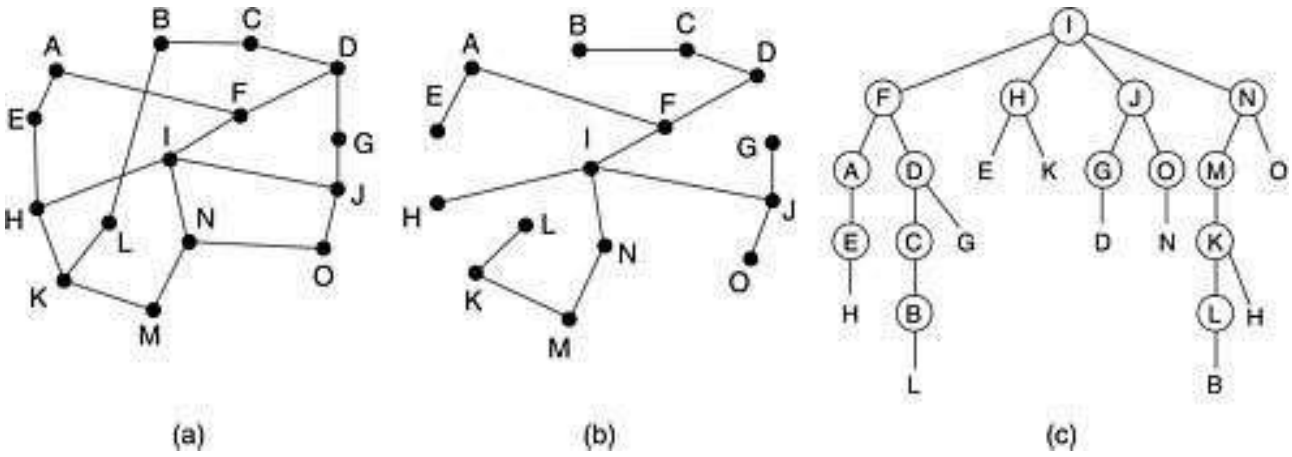
BROADCAST ROUTING

Sending a packet to all destinations simultaneously is called broadcasting.

1) The source simply sends a distinct packet to each destination. Not only is the method wasteful of bandwidth, but it also requires the source to have a complete list of all destinations.

2) Flooding.

The problem with flooding as a broadcast technique is that it generates too many packets and consumes too much bandwidth.



Reverse path forwarding. (a) A subnet. (b) A sink tree. (c) The tree built by reverse path forwarding.

Part (a) shows a subnet, part (b) shows a sink tree for router *I* of that subnet, and part (c) shows how the reverse path algorithm works.

- When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending packets to the source of the broadcast. If so, there is an excellent chance that the broadcast packet itself followed the best route from the router and is therefore the first copy to arrive at the router.

- This being the case, the router forwards copies of it onto all lines except the one it arrived on. If, however, the broadcast packet arrived on a line other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.

MULTICAST ROUTING

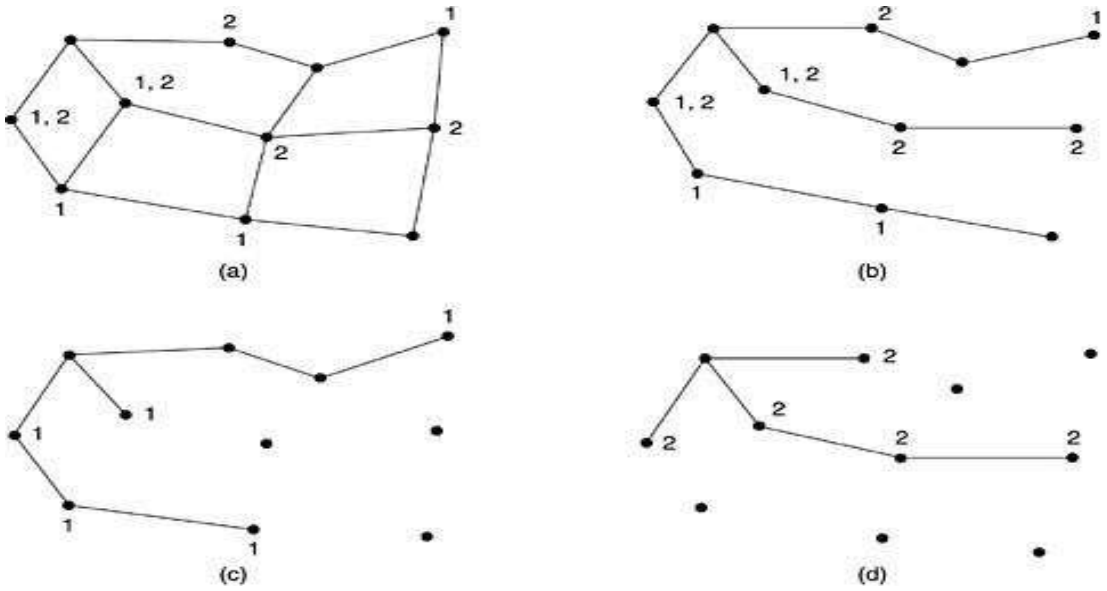
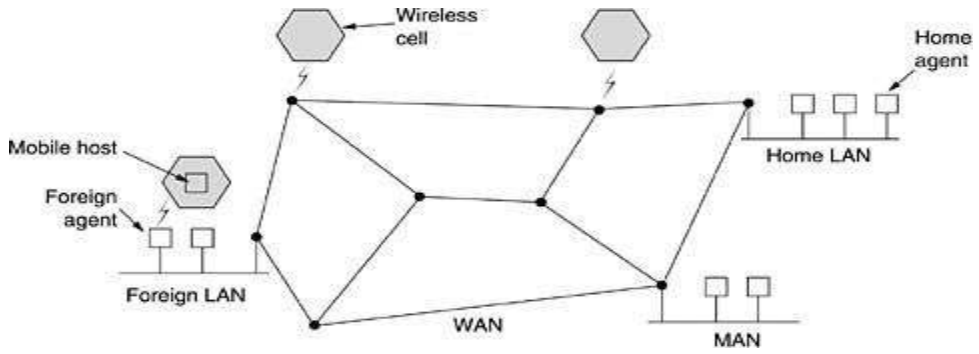


Fig. 5-17

- To do multicast routing, each router computes a spanning tree covering all other routers. For example, in Fig. 5-17(a) we have two groups, 1 and 2.
- Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure.
- A spanning tree for the leftmost router is shown in Fig. 5-17(b). When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group.
- In our example, Fig. 5-17(c) shows the pruned spanning tree for group 1. Similarly, Fig. 5-17(d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

ROUTING FOR MOBILE HOSTS



- Hosts that never move are said to be stationary.
- They are connected to the network by copper wires or fiber optics. In contrast, we can distinguish two other kinds of hosts.

- Migratory hosts are basically stationary hosts who move from one fixed site to another from time to time but use the network only when they are physically connected to it.
- Roaming hosts actually compute on the run and want to maintain their connections as they move around.
- We will use the term **mobile hosts** to mean either of the latter two categories, that is, all hosts that are away from home and still want to be connected

The registration procedure typically works like this:

1. Periodically, each foreign agent broadcasts a packet announcing its existence and address. A newly-arrived mobile host may wait for one of these messages, but if none arrives quickly enough, the mobile host can broadcast a packet saying: Are there any foreign agents around?
2. The mobile host registers with the foreign agent, giving its home address, current data link layer address, and some security information.
3. The foreign agent contacts the mobile host's home agent and says: One of your hosts is over here. The message from the foreign agent to the home agent contains the foreign agent's network address. It also includes the security information to convince the home agent that the mobile host is really there.
4. The home agent examines the security information, which contains a timestamp, to prove that it was generated within the past few seconds. If it is happy, it tells the foreign agent to proceed.
5. When the foreign agent gets the acknowledgement from the home agent, it makes an entry in its tables and informs the mobile host that it is now registered.

ROUTING IN AD HOC NETWORKS

We have now seen how to do routing when the hosts are mobile but the routers are fixed. An even more extreme case is one in which the routers themselves are mobile. Among the possibilities are:

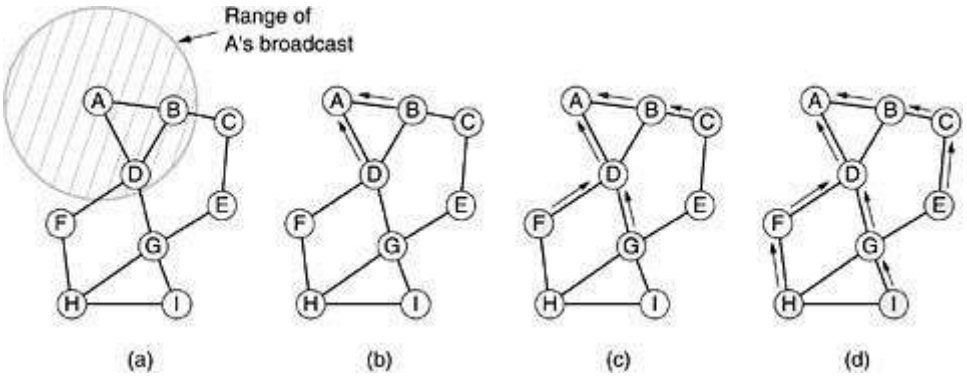
1. Military vehicles on a battlefield with no existing infrastructure.
2. A fleet of ships at sea.
3. Emergency workers at an earthquake that destroyed the infrastructure.
4. A gathering of people with notebook computers in an area lacking 802.11.

In all these cases, and others, each node consists of a router and a host, usually on the same computer. Networks of nodes that just happen to be near each other are called **ad hoc networks** or **MANETs** (**Mobile Ad hoc NETworks**).

- What makes ad hoc networks different from wired networks is that all the usual rules about fixed topologies, fixed and known neighbours, fixed relationship between IP address and location, and more are suddenly tossed out the window.

- Routers can come and go or appear in new places at the drop of a bit. With a wired network, if a router has a valid path to some destination, that path continues to be valid indefinitely (barring a failure somewhere in the system).
- With an ad hoc network, the topology may be changing all the time.
- A variety of routing algorithms for ad hoc networks have been proposed. One of the more interesting ones is the **AODV (Ad hoc On-demand Distance Vector**) routing algorithm (Perkins and Royer, 1999).
- It takes into account the limited bandwidth and low battery life found in environment. Another unusual characteristic is that it is an on-demand algorithm, that is, it determines a route to some destination only when somebody wants to send a packet to that destination. Let us now see what that means.

Route Discovery



(a) Range of* A*'s broadcast. (b) After* B *and* D *have received* A*'s broadcast. (c) After* C*,* F*, and* G *have received* A*'s broadcast. (d) After* E*,* H*, and* I *have received* A*'s broadcast. The shaded nodes are new recipients. The arrows show the possible reverse routes.

- To locate *I*, *A* constructs a special ROUTE REQUEST packet and broadcasts it. The packet reaches *B* and *D*, as illustrated in Fig. 5-20(a).
- The format of the ROUTE REQUEST packet is shown in Fig. 5-21

Format of a ROUTE REQUEST packet.

Source address	Request ID	Destination address	Source sequence #	Dest. sequence #	Hop count

The format of the ROUTE REQUEST packet is shown in Fig. 5-21. It contains the source and destination addresses, typically their IP addresses, which identify who is looking for whom. It also contains a *Request ID*, which is a local counter maintained separately by each node and incremented each time a ROUTE REQUEST is broadcast. Together, the *Source address* and *Request ID* fields uniquely identify the ROUTE REQUEST packet to allow nodes to discard any duplicates they may receive.

Format of a ROUTE REPLY packet

Source address	Destination address	Destination sequence #	Hop count	Lifetime

In addition to the *Request ID* counter, each node also maintains a second sequence counter incremented whenever a ROUTE REQUEST is sent (or a reply to someone else's ROUTE REQUEST). It functions a little bit like a clock and is used to tell new routes from old routes. The fourth field of Fig. 5-21 is *A*'s sequence counter; the fifth field is the most recent value of *I*'s sequence number that *A* has seen (0 if it has never seen it). The use of these fields will become clear shortly. The final field, *Hop count*, will keep track of how many hops the packet has made. It is initialized to 0.

1. No route to *I* is known.
2. The sequence number for *I* in the ROUTE REPLY packet is greater than the value in the routing table.
3. The sequence numbers are equal but the new route is shorter.

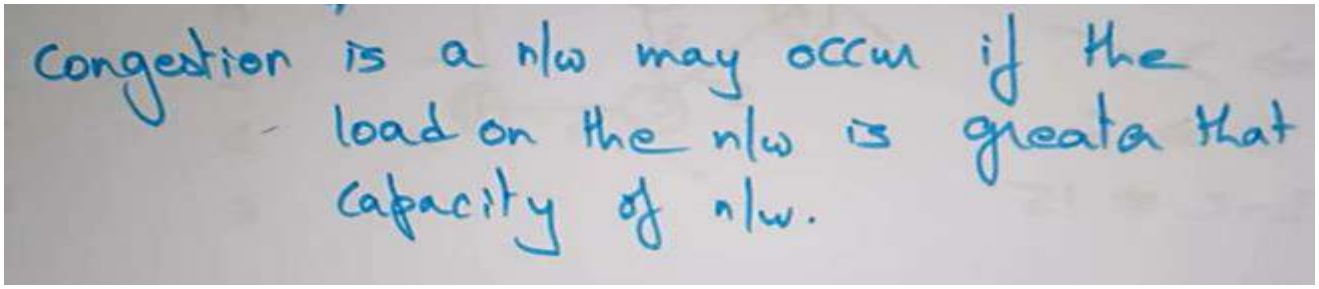
CONGESTION CONTROL ALGORITHMS

- When too many packets are present in (a part of) the subnet, performance degrades. This situation is called **congestion**. (or)
- In the network layer, before the network can make Quality of service guarantees, it must know what traffic is being guaranteed. One of the main causes of congestion is that traffic is often bursty.
- To understand this concept first we have to know little about traffic shaping. Traffic Shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Approach of congestion management is called Traffic shaping. Traffic shaping helps to regulate rate of data transmission and reduces congestion.

There are 2 types of traffic shaping algorithms:

1. Leaky Bucket
2. Token Bucket

- Suppose we have a bucket in which we are pouring water in a random order but we have to get water in a fixed rate, for this we will make a hole at the bottom of the bucket. It will ensure that water coming out is in a some fixed rate, and also if bucket will full we will stop pouring in it.
- The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.



- Figure 5-25 depicts the symptom. When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered (except for a few that are afflicted with transmission errors) and the number delivered is proportional to the number sent.
- However, as traffic increases too far, the routers are no longer able to cope and they begin losing packets. This tends to make matters worse. At very high traffic, performance collapses completely and almost no packets are delivered.

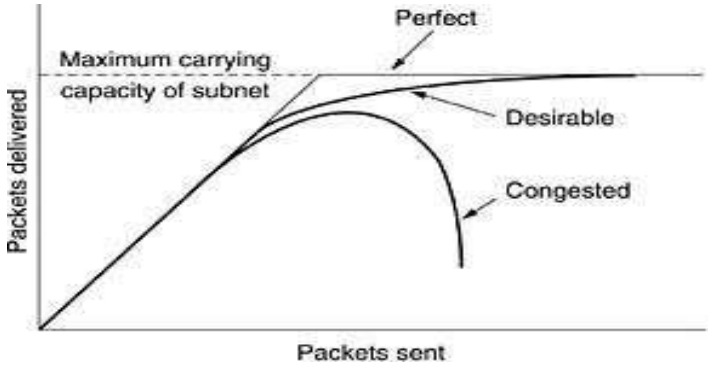
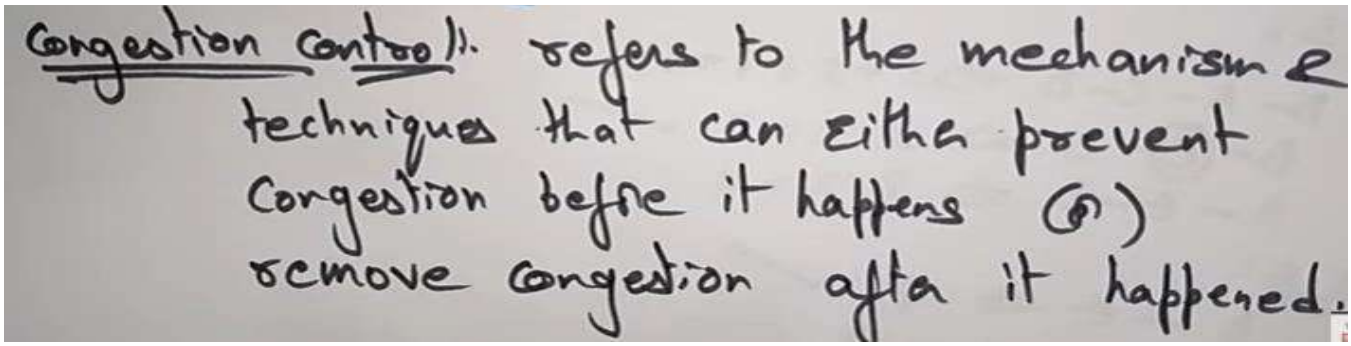
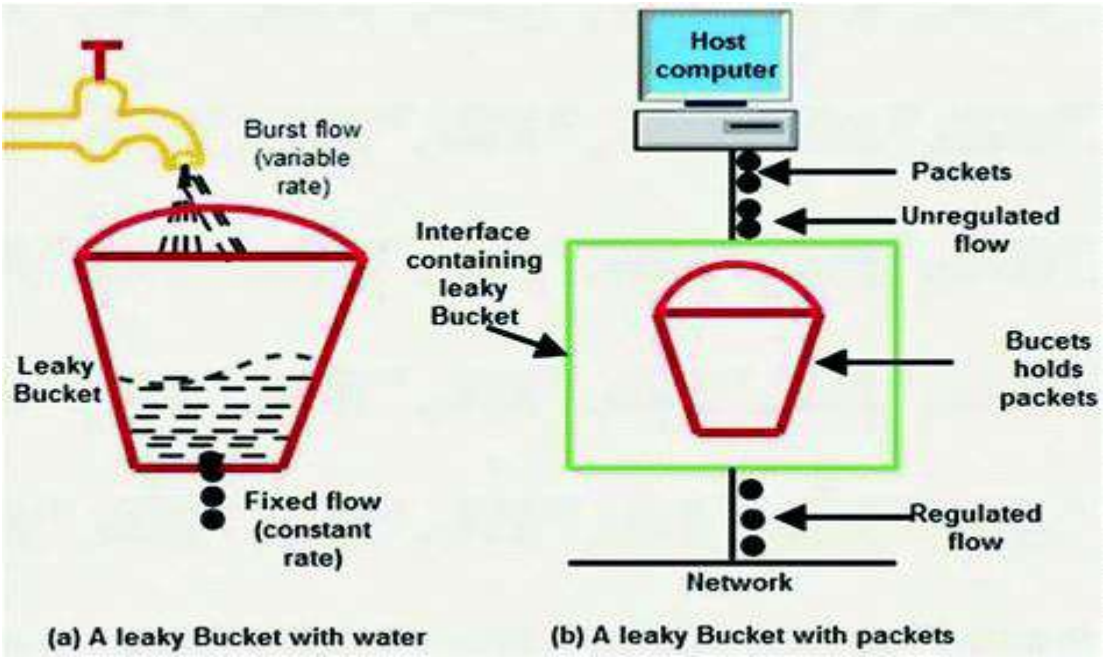
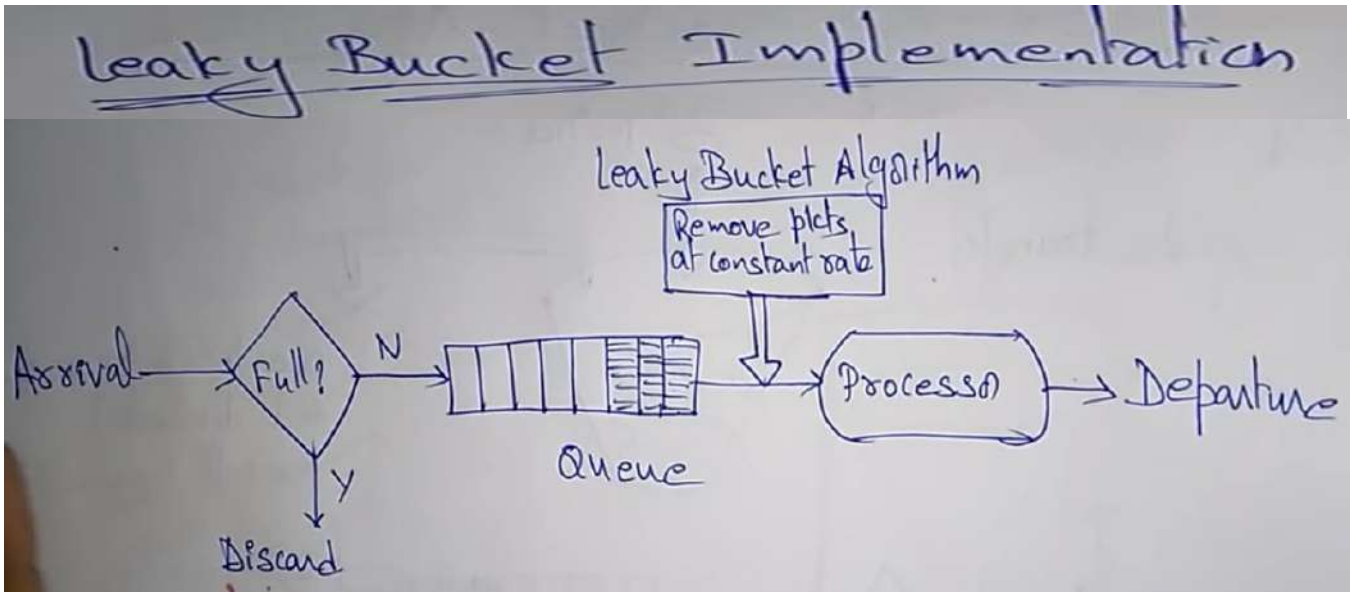


Figure 5-25. When too much traffic is offered, congestion sets in and performance degrades sharply.







- **Congestion can be brought on by several factors**. If all of a sudden, streams **of packets begin arriving on three or four input lines and all need the same output line, a queue will build up.**
- **If there is insufficient memory to hold all of them, packets will be lost.**
- **Slow processors can also cause congestion**. If the routers' CPUs are slow at performing the bookkeeping tasks required of them (queuing buffers, updating tables, etc.), queues can build up, even though there is excess line capacity. **Similarly, low-bandwidth lines can also cause congestion.**

APPROACHES TO CONGESTION CONTROL

- Many problems in complex systems, such as computer networks, can be viewed from a control theory point of view. **This approach leads to dividing all solutions into two groups: open loop and closed loop.**

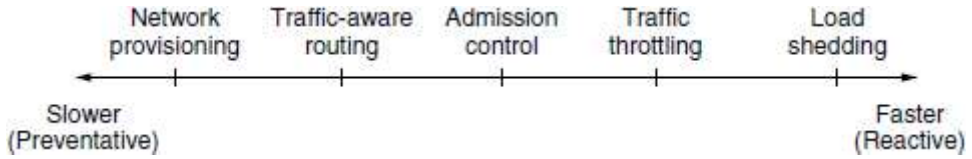


Figure: Time scales Of Approaches To Congestion Control

- **Open loop** solutions attempt to solve the problem **by good design.**
- Tools for doing open-loop control include **deciding when to accept new traffic**, **deciding when to discard packets** and **which ones**, and **making scheduling decisions at various points in the network.**
- **Closed loop** solutions are based on the concept of a **feedback loop.**

- This approach has **three parts** when applied to congestion control:
  1. Monitor the system to detect when and where congestion occurs.
  2. Pass this information to places where action can be taken.
  3. Adjust system operation to correct the problem.

- A variety of metrics can be used to monitor the subnet for congestion. Chief among these are the percentage of all packets discarded for lack of buffer space, the average queue lengths, the number of packets that time out and are retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion.
- The second step in the feedback loop is to transfer the information about the congestion from the point where it is detected to the point where something can be done about it.
- In all feedback schemes, the hope is that knowledge of congestion will cause the hosts to take appropriate action to reduce the congestion.
- The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle. Two solutions come to mind: increase the resources or decrease the load.

CONGESTION PREVENTION POLICIES

The methods to control congestion by looking at **open loop systems**. These systems are designed to minimize congestion in the first place, rather than letting it happen and reacting after the fact. They try to achieve their goal by using appropriate policies at various levels. In Fig. 5-26 we see different data link, network, and transport policies that can affect congestion (Jain, 1990).

Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Figure 5-26. Policies that affect congestion.

The **data link layer Policies.**

- The **retransmission policy** is concerned with how fast a sender times out and what it transmits upon timeout. A jumpy sender that times out quickly and retransmits all

outstanding packets using **go back n** will put a heavier load on the system than will a leisurely sender that uses **selective repeat**.

- Closely related to this is the **buffering policy**. If receivers routinely discard all out-of- order packets, these packets will have to be transmitted again later, creating extra load. With respect to congestion control, selective repeat is clearly better than go back n.
- **Acknowledgement policy** also affects congestion. If each packet is acknowledged immediately, the acknowledgement packets generate extra traffic. However, if acknowledgements are saved up to piggyback onto reverse traffic, extra timeouts and retransmissions may result. A tight flow control scheme (e.g., a small window) reduces the data rate and thus helps fight congestion.

The **network layer Policies.**

- The choice between using **virtual circuits and using datagrams** affects congestion since many congestion control algorithms work only with virtual-circuit subnets.
- **Packet queueing and service policy** relates to whether routers have one queue per input line, one queue per output line, or both. It also relates to the order in which packets are processed (e.g., round robin or priority based).
- **Discard policy** is the rule telling which packet is dropped when there is no space.
- A good **routing algorithm** can help avoid congestion by spreading the traffic over all the lines, whereas a bad one can send too much traffic over already congested lines.
- **Packet lifetime management** deals with how long a packet may live before being discarded. If it is too long, lost packets may clog up the works for a long time, but if it is too short, packets may sometimes time out before reaching their destination, thus inducing retransmissions.

The **transport layer Policies**,

- The **same issues occur as in the data link layer**, but in addition, determining the **timeout interval** is harder because the transit time across the network is less predictable than the transit time over a wire between two routers. If the timeout interval is too short, extra packets will be sent unnecessarily. If it is too long, congestion will be reduced but the response time will suffer whenever a packet is lost.

ADMISSION CONTROL

- One technique that is widely used to keep congestion that has already started from getting **Service Mechanism** is **admission control**.
- Once congestion has been signaled, no more virtual circuits are set up until the problem has gone away.
- An alternative approach is to allow new virtual circuits but carefully route all new virtual circuits around problem areas. For example, consider the subnet of Fig. 5-27(a), in which two routers are congested, as indicated.

.

CHOKE PACKETS

- In this approach, the router sends a **choke packet** back to the source host, giving it the destination found in the packet.
- The original packet is tagged (a header bit is turned on) so that it will not generate any more choke packets farther along the path and is then forwarded in the usual way.
- **When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination by *X* percent.** Since other packets aimed at the same destination are probably already under way and will generate yet more choke packets**, the host should ignore choke packets referring to that destination for a fixed time interval. After that period has expired, the host listens for more choke packets for another interval.**
- The feedback implicit in this protocol can help prevent congestion yet not throttle any flow unless trouble occurs.
- Hosts can reduce traffic by adjusting their policy parameters.
- Increases are done in smaller increments to prevent congestion from reoccurring quickly.
- Routers can maintain several thresholds. Depending on which threshold has been crossed, the choke packet can contain a mild warning, a stern warning, or an ultimatum.

HOP-BY-HOP BACK PRESSURE

- **At high speeds or over long distances, sending a choke packet to the source hosts does not work well because the reaction is so slow.**

Consider, for example, **a host in San Francisco** (router *A* in Fig. 5-28) that is sending traffic **to a host in New York** (router *D* in Fig. 5-28) at 155 Mbps. If the New York host begins to run out of buffers, it will take about 30 msec for a choke packet to get back to San Francisco to tell it to slow down. The choke packet propagation is shown as the second, third, and fourth steps in Fig. 5-28(a). In those 30 msec, another 4.6 megabits will have been sent. Even if the host in San Francisco completely shuts down immediately, the 4.6 megabits in the pipe will continue to pour in and have to be dealt with. Only in the seventh diagram in Fig. 5-28(a) will the New York router notice a slower flow.

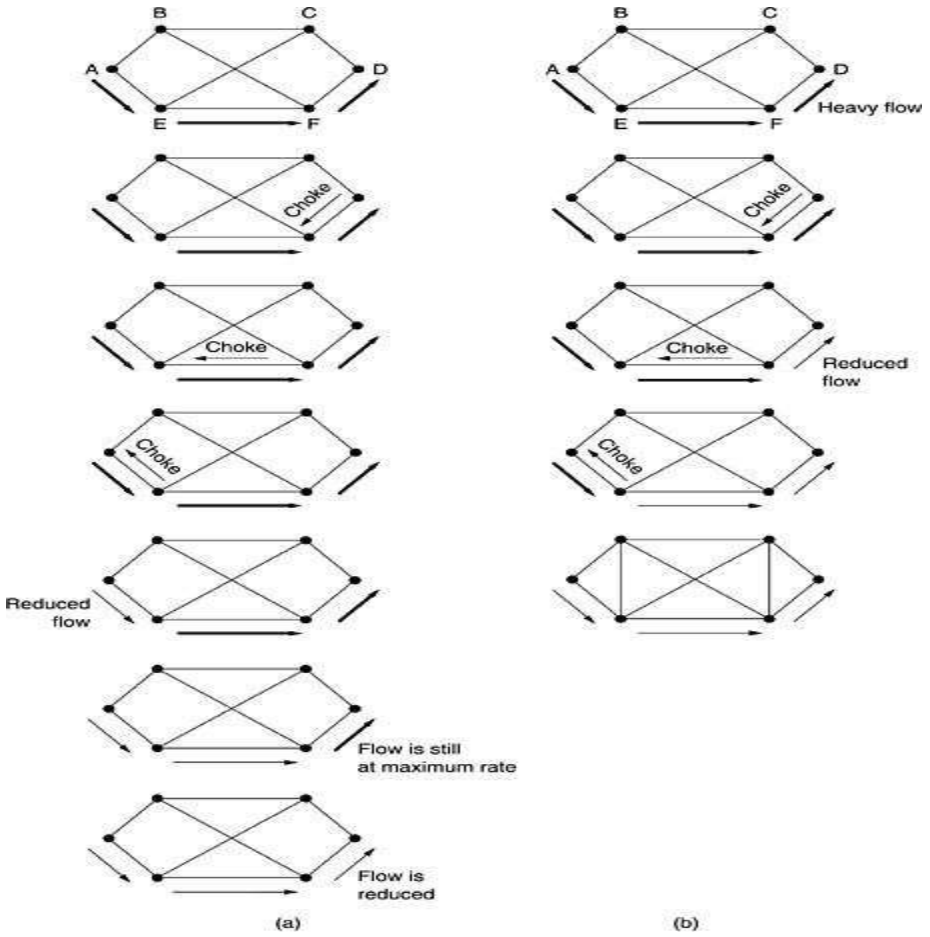


Figure 5-28. (a) A choke packet that affects only the source. (b) A choke packet that affects each hop it passes through.

An alternative approach is to have the choke packet take effect at every hop it passes through, as shown in the sequence of Fig. 5-28(b). Here, as soon as the choke packet reaches *F*, *F* is required to reduce the flow to *D*. Doing so will require *F* to devote more buffers to the flow, since the source is still sending away at full blast, but it gives *D* immediate relief, like a headache remedy in a television commercial. In the next step, the choke packet reaches *E*, which tells *E* to reduce the flow to *F*. This action puts a greater demand on *E*'s buffers but gives *F* immediate relief. Finally, the choke packet reaches *A* and the flow genuinely slows down.

The net effect of this hop-by-hop scheme is to provide quick relief at the point of congestion at the price of using up more buffers upstream. In this way, congestion can be nipped in the bud without losing any packets.

JITTER CONTROL

- **The variation (i.e., standard deviation) in the packet arrival times is called jitter.**
- High jitter, for example, having some packets taking 20 msec and others taking 30 msec to arrive will give an uneven quality to the sound or movie. Jitter is illustrated in Fig. 5-29. In contrast, an agreement that 99 percent of the packets be delivered with a delay in the range of 24.5 msec to 25.5 msec might be acceptable.

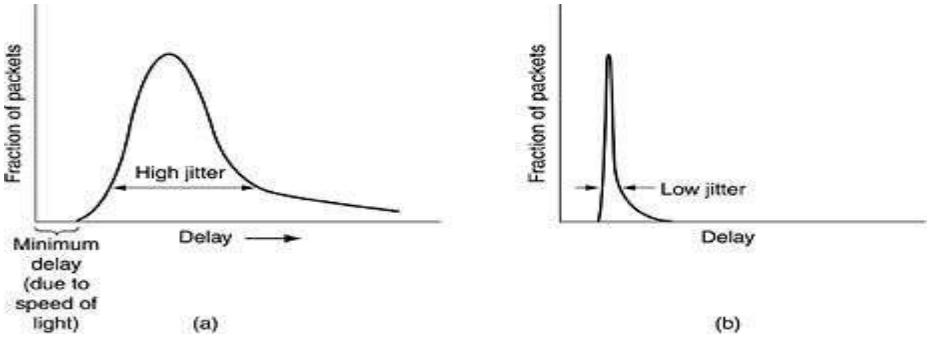
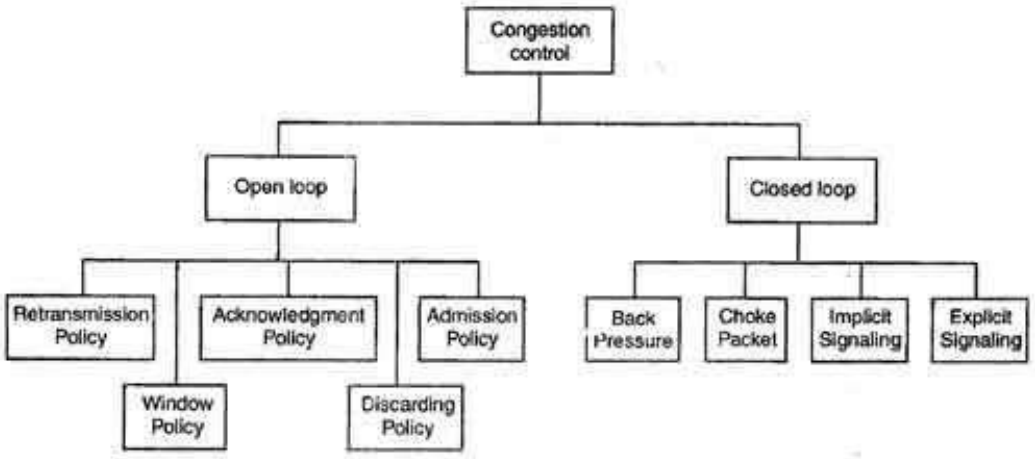


Figure 5-29. (a) High jitter. (b) Low jitter.

- The jitter can be bounded by computing the expected transit time for each hop along the path. When a packet arrives at a router, the router checks to see how much the packet is behind or ahead of its schedule. This information is stored in the packet and updated at each hop.
- In fact, the algorithm for determining which of several packets competing for an output line should go next can always choose the packet furthest behind in its schedule.
- In this way, packets that are ahead of schedule get slowed down and packets that are behind schedule get speeded up, in both cases reducing the amount of jitter.

How to correct the Congestion Problem:

Congestion Control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. Congestion control mechanisms are divided into two categories, one category prevents the congestion from happening and the other category removes congestion after it has taken place.



Types of Congestion Control Methods

These two categories are:

1. Open loop
2. Closed loop

Open Loop Congestion Control

• **In this method, policies are used to prevent the congestion before it happens**.

• Congestion control is handled either by the source or by the destination.

• The various methods used for open loop congestion control are:

1. Retransmission Policy

• The sender retransmits a packet, if it feels that the packet it has sent is lost or corrupted.

• However retransmission in general may increase the congestion in the network. But we need to implement good retransmission policy to prevent congestion.

• The retransmission policy and the retransmission timers need to be designed to optimize efficiency and at the same time prevent the congestion.

2. Window Policy

• To implement window policy, selective reject window method is used for congestion control.

• Selective Reject method is preferred over Go-back-n window as in Go-back-n method, when timer for a packet times out, several packets are resent, although some may have arrived safely at the receiver. Thus, this duplication may make congestion worse.

• Selective reject method sends only the specific lost or damaged packets.

3. Acknowledgement Policy

• The acknowledgement policy imposed by the receiver may also affect congestion.

• If the receiver does not acknowledge every packet it receives it may slow down the sender and help prevent congestion.

• Acknowledgments also add to the traffic load on the network. Thus, by sending fewer acknowledgements we can reduce load on the network.

• To implement it, several approaches can be used:

1. A receiver may send an acknowledgement only if it has a packet to be sent.
2. A receiver may send an acknowledgement when a timer expires.
3. A receiver may also decide to acknowledge only *N* packets at a time.

4. Discarding Policy

• A router may discard less sensitive packets when congestion is likely to happen.

• Such a discarding policy may prevent congestion and at the same time may not harm the integrity of the transmission.

5. Admission Policy

• An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.

• Switches in a flow first check the resource requirement of a flow before admitting it to the network.

• A router can deny establishing a virtual circuit connection if there is congestion in the "network or if there is a possibility of future congestion.

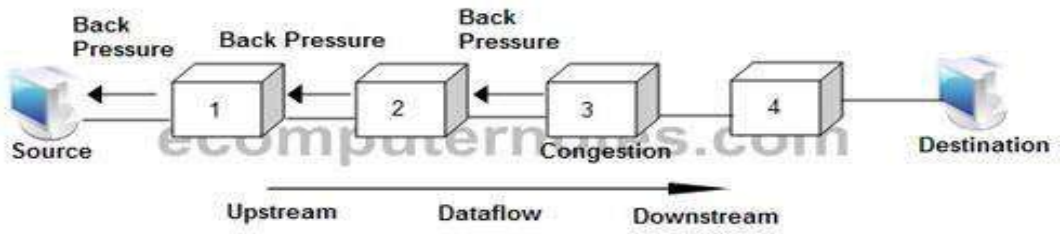
Closed Loop Congestion Control

• Closed loop congestion control mechanisms try to remove the congestion after it happens.

• The various methods used for closed loop congestion control are:

1. Backpressure

• Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow.



Backpressure Method

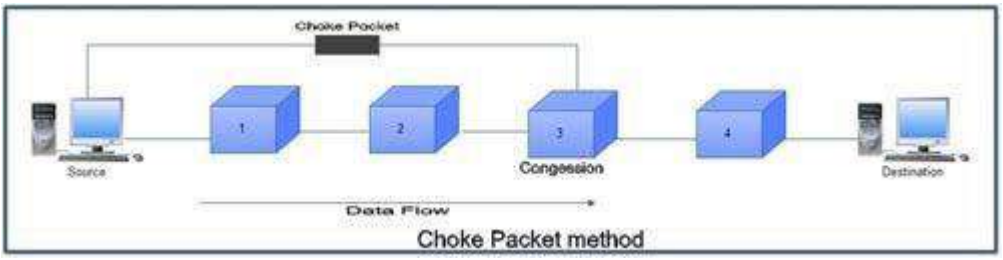
• The backpressure technique can be applied only to virtual circuit networks. In such virtual circuit each node knows the upstream node from which a data flow is coming.
• In this method of congestion control, the congested node stops receiving data from the immediate upstream node or nodes.
• This may cause the upstream node on nodes to become congested, and they, in turn, reject data from their upstream node or nodes.
• As shown in fig node 3 is congested and it stops receiving packets and informs its upstream node 2 to slow down. Node 2 in turns may be congested and informs node 1 to slow down. Now node 1 may create congestion and informs the source node to slow down. In this way the congestion is alleviated. Thus, the pressure on node 3 is moved backward to the source to remove the congestion.

2. Choke Packet

• In this method of congestion control, congested router or node sends a special type of packet called choke packet to the source to inform it about the congestion.

• Here, congested node does not inform its upstream node about the congestion as in backpressure method.

• In choke packet method, congested node sends a warning directly to the source station *i.e.* the intermediate nodes through which the packet has traveled are not warned.



3. Implicit Signaling

• In implicit signaling, there is no communication between the congested node or nodes and the source.

• The source guesses that there is congestion somewhere in the network when it does not receive any acknowledgment. Therefore the delay in receiving an acknowledgment is interpreted as congestion in the network.

• On sensing this congestion, the source slows down.
• This type of congestion control policy is used by TCP.

4. Explicit Signaling

• In this method, the congested nodes **explicitly send a signal to the source or destination to inform about the congestion.**

• Explicit signaling is different from the choke packet method. In choke packed method, a separate packet is used for this purpose whereas in explicit signaling method, the signal is included in the packets that carry data .

• Explicit signaling can occur in either the forward direction or the backward direction.

• In backward signaling, a bit is set in a packet moving in the direction opposite to the congestion. This bit warns the source about the congestion and **informs the source to slow down.**

• In forward signaling, a bit is set in a packet moving in the direction of congestion. This bit warns the destination about the congestion. The receiver in this case uses policies such as **slowing down the acknowledgements to remove the congestion**.

UNIT VI

TRANSPORT LAYER &

APPLICATION LAYER

Transport Layer – The Internet Transport Protocols: Udp, the Internet Transport Protocols: Tcp

Application Layer –The Domain Name System: The DNS Name Space, Resource Records, Name Servers, Electronic Mail: Architecture and Services, The User Agent, Message Formats, Message Transfer, Final Delivery

Introduction:

The transport layer provides a logical communication between application processes running on different hosts. ... For example, TCP and UDP are two transport layer protocols that provide a different set of services to the network layer. All transport layer protocols provide multiplexing/demultiplexing service.

The network layer provides end-to-end packet delivery using data-grams or virtual circuits. The transport layer builds on the network layer to provide data transport from a process on a source machine to a process on a destination machine with a desired level of reliability that is independent of the physical networks currently in use. It provides the abstractions that applications need to use the network.

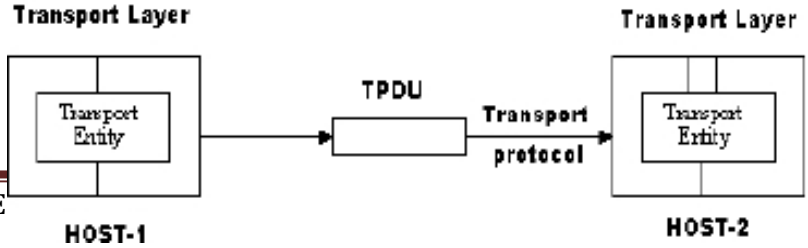
Transport Entity: The hardware and/or software which make use of services provided by the network layer, (within the transport layer) is called transport entity.

Transport Service Provider: Layers 1 to 4 are called Transport Service Provider.

Transport Service User: The upper layers i.e., layers 5 to 7 are called Transport Service User.

Transport Service Primitives: Which allow transport users (application programs) to access the transport service.

TPDU (Transport Protocol Data Unit): Transmissions of message between 2 transport entities are carried out by TPDU. The transport entity carries out the transport service primitives by blocking the caller and sending a packet the service. Encapsulated in the payload of this packet is a transport layer message for the server's transport entity. The task of the transport layer is to provide reliable, cost-effective data transport from the source machine to the destination machine, independent of physical network or networks currently in use.



TRANSPORT SERVICE

1. Services Provided to the Upper Layers

The ultimate goal of the transport layer is to provide efficient, **reliable, and cost-effective data transmission** service to its users, normally processes in the application layer. To achieve this, the transport layer makes use of the **services pro-vided by the network layer.** The software and/or hardware within the transport layer that does the work is called the **transport entity.** The transport entity can be located in the operating system kernel, in a library package bound into network applications, in a separate user process, or even on the network interface card.

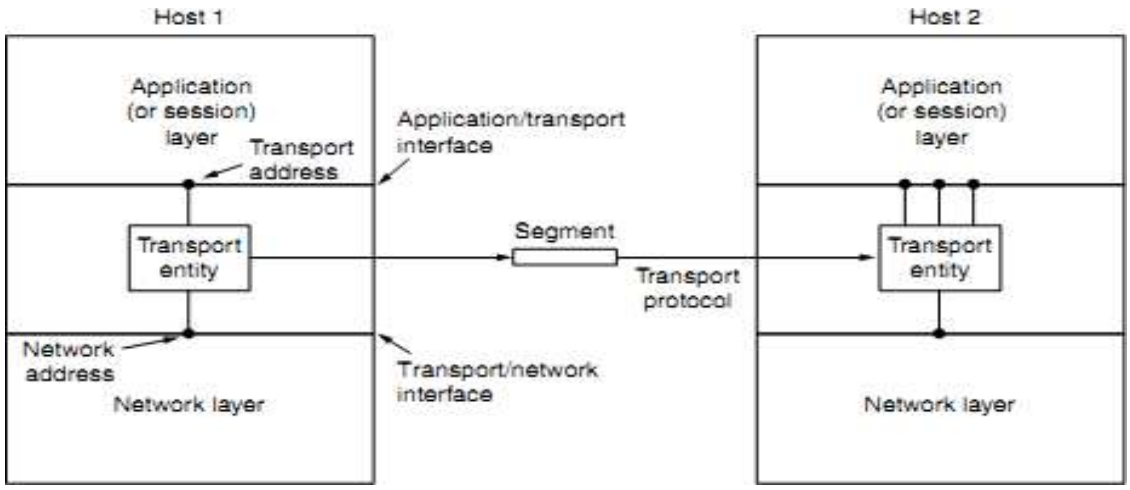


Fig 4.1: The network, Application and transport layer

There are two types of network service

- Connection-oriented
- Connectionless

Similarly, there are also two types of transport service. The connection-oriented transport service is similar to the connection-oriented network service in many ways.

In both cases, connections have three phases:

- Establishment
- Data transfer
- Release.

- Addressing and flow control are also similar in both layers. Furthermore, the connectionless transport service is also very similar to the connectionless network service.
- The bottom four layers can be seen as the transport service provider, whereas the upper layer(s) are the transport service user.

2. Transport Service Primitives

- To allow users to access the transport service, the transport layer must provide some operations to application programs, that is, a transport service interface. Each transport service has its own interface.
- The transport service is similar to the network service, but there are also some important differences.
- The **main difference** is that the network service is intended to model the service offered by real networks. Real networks can lose packets, so the network service is generally **unreliable.**
- The (connection-oriented) transport service, in contrast, is **reliable**

As an example, consider two processes connected by pipes in UNIX. They assume the connection between them is perfect. They do not want to know about acknowledgements, lost packets, congestion, or anything like that. What they want is a 100 percent reliable connection. Process A puts data into one end of the pipe, and process B takes it out of the other.

A **second difference** between the network service and transport service is **whom the services are intended for**. The network service is used only by the transport entities. Consequently, the transport service must be convenient and easy to use.

Table:4.1 - The primitives for a simple transport service.

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

Eg: Consider an application with a server and a number of remote clients.

1. The server executes a “LISTEN” primitive by calling a library procedure that makes a System call to block the server until a client turns up.
2. When a client wants to talk to the server, it executes a “CONNECT” primitive, with “CONNECTION REQUEST” TPDU sent to the server.
3. When it arrives, the TE unblocks the server and sends a “CONNECTION ACCEPTED” TPDU back to the client.
4. When it arrives, the client is unblocked and the connection is established. Data can now be exchanged using “SEND” and “RECEIVE” primitives.
5. When a connection is no longer needed, it must be released to free up table space within the 2 transport entries, which is done with “DISCONNECT” primitive by sending “DISCONNECTION REQUEST”

TPDU. This disconnection can b done either by asymmetric variant (connection is released, depending on other one) or by symmetric variant (connection is released, independent of other one).

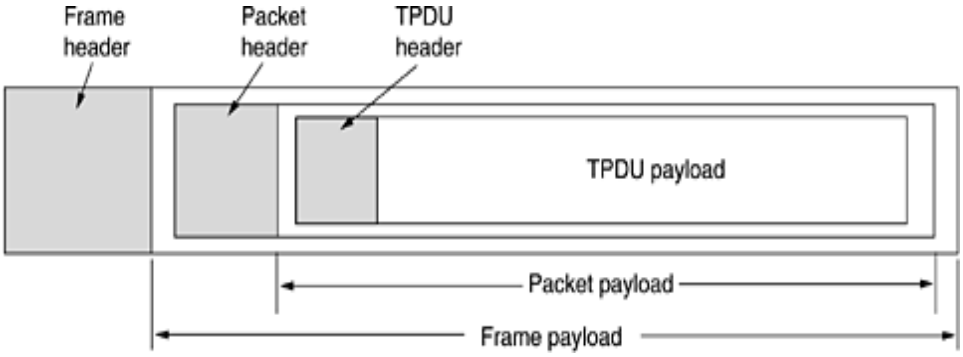


Figure 4.2 - Nesting of TPDUs, packets, and frames

- The term segment for messages sent from transport entity to transport entity.
- TCP, UDP and other Internet protocols use this term. Segments (exchanged by the transport layer) are contained in packets (exchanged by the network layer).
- These packets are contained in frames(exchanged by the data link layer).When a frame arrives, the data link layer processes the frame header and, if the destination address matches for local delivery, passes the contents of the frame payload field up to the network entity.
- The network entity similarly processes the packet header and then passes the contents of the packet payload up to the transport entity. This nesting is illustrated in Fig. 4.2.

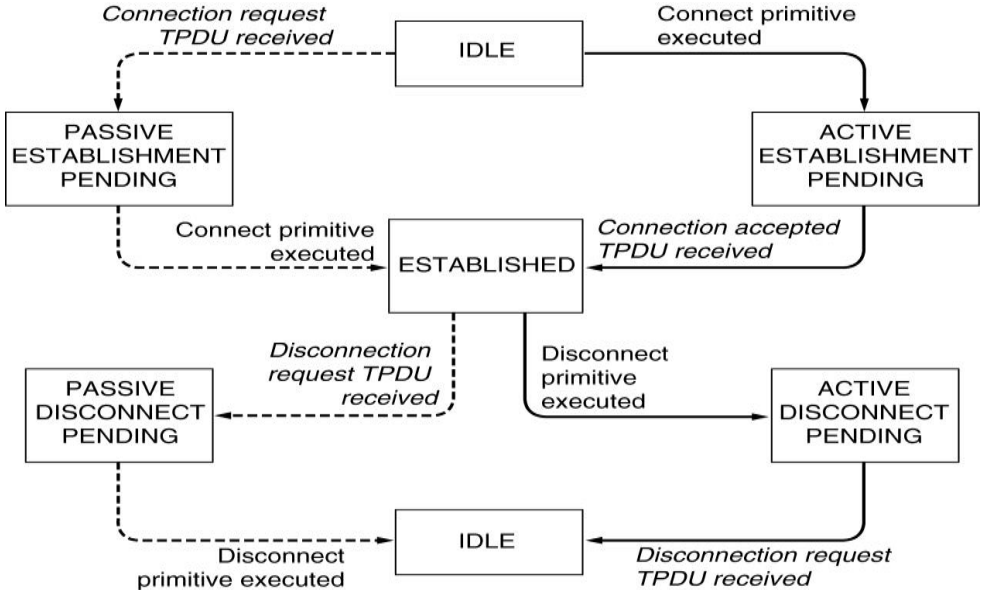


Figure 4.3 - A state diagram for a simple connection management scheme. Transitions labelled in italics are caused by packet arrivals. The solid lines show the client's state sequence. The dashed lines show the server's state sequence.

In fig. 4.3 each transition is triggered by some event, either a primitive executed by the local transport user or an incoming packet. For simplicity, we assume here that each TPDU is separately acknowledged. We also assume that a symmetric disconnection model is used, with the client going first. Please note that this model is quite unsophisticated. We will look at more realistic models later on.

BERKLEY SOCKETS

These primitives are socket primitives used in Berkley UNIX for TCP.

The socket primitives are mainly used for TCP. These sockets were first released as part of the Berkeley UNIX 4.2BSD software distribution in 1983. They quickly became popular. The primitives are now widely used for Internet programming on many operating systems, especially UNIX -based systems, and there is a socket-style API for Windows called ''**winsock**.''

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Figure 4.4 - The socket primitives for TCP.

The first four primitives in the list are executed in that order by servers.

The **SOCKET** primitive creates a new endpoint and allocates table space for it within the transport entity. The parameter includes the addressing format to be used, the type of service desired and the protocol. Newly created sockets do not have network addresses.

- The **BIND** primitive is used to connect the newly created sockets to an address. Once a server has bound an address to a socket, remote clients can connect to it.
- The **LISTEN** call, which allocates space to queue incoming calls for the case that several clients try to connect at the same time.
- The server executes an **ACCEPT** primitive to block waiting for an incoming connection.

Some of the client side primitives are. Here, too, a socket must first be created

- The **CONNECT** primitive blocks the caller and actively starts the connection process. When it completes, the client process is unblocked and the connection is established.
- Both sides can now use **SEND** and **RECEIVE** to transmit and receive data over the full-duplex connection.
- Connection release with sockets is symmetric. When both sides have executed a **CLOSE** primitive, the connection is released.

ELEMENTS OF TRANSPORT PROTOCOLS

The transport service is implemented by a transport protocol used between the two transport entities. The transport protocols resemble the data link protocols. Both have to deal with error control, sequencing, and flow control, among other issues. The difference transport protocol and data link protocol depends upon the environment in which they are operated.

These differences are due to major dissimilarities between the environments in which the two protocols operate, as shown in Fig.

At the data link layer, two routers communicate directly via a physical channel, whether wired or wireless, whereas at the transport layer, this physical channel is replaced by the entire network. This difference has many important implications for the protocols.

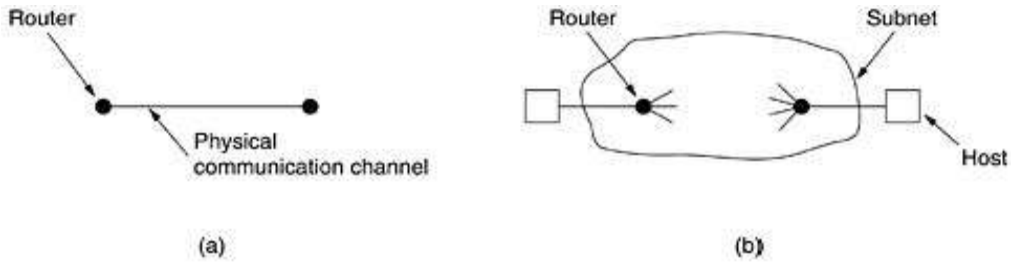


Figure (a) Environment of the data link layer. (b) Environment of the transport layer.

In the data link layer, it is not necessary for a router to specify which router it wants to talk to. In the transport layer, explicit addressing of destinations is required.

In the transport layer, initial connection establishment is more complicated, as we will see. Difference between the data link layer and the transport layer is the potential existence of storage capacity in the subnet

Buffering and flow control are needed in both layers, but the presence of a large and dynamically varying number of connections in the transport layer may require a different approach than we used in the data link layer.

The transport service is implemented by a transport protocol between the 2 transport entities.

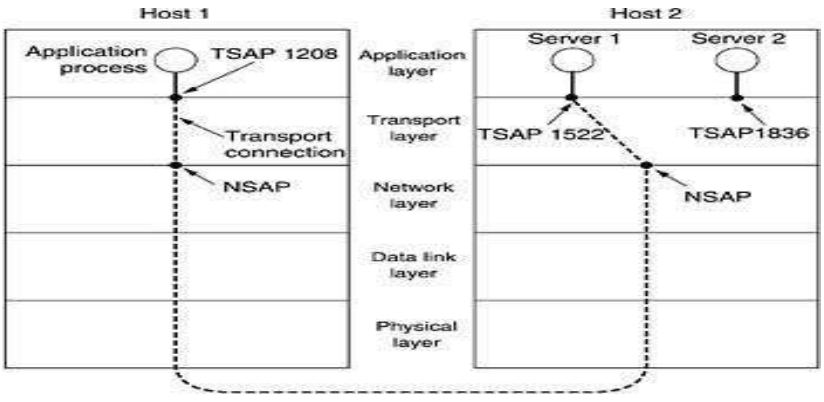


Figure 4.5 illustrates the relationship between the NSAP, TSAP and transport connection. Application processes, both clients and servers, can attach themselves to a TSAP to establish a connection to a remote

TSAP.

These connections run through NSAPs on each host, as shown. The purpose of having TSAPs is that in some networks, each computer has a single NSAP, so some way is needed to distinguish multiple transport end points that share that NSAP.

The elements of transport protocols are:

1. ADDRESSING
2. Connection Establishment.
3. Connection Release.
4. Error control and flow control
5. Multiplexing.

1. ADDRESSING

When an application (e.g., a user) process wishes to set up a connection to a remote application process, it must specify which one to connect to. The method normally used is to define transport addresses to which processes can listen for connection requests. In the Internet, these endpoints are called **ports**.

There are two types of access points.

TSAP (Transport Service Access Point) to mean a specific endpoint in the transport layer.

The analogous endpoints in the network layer (i.e., network layer addresses) are not surprisingly called **NSAPs (Network Service Access Points).** IP addresses are examples of NSAPs.

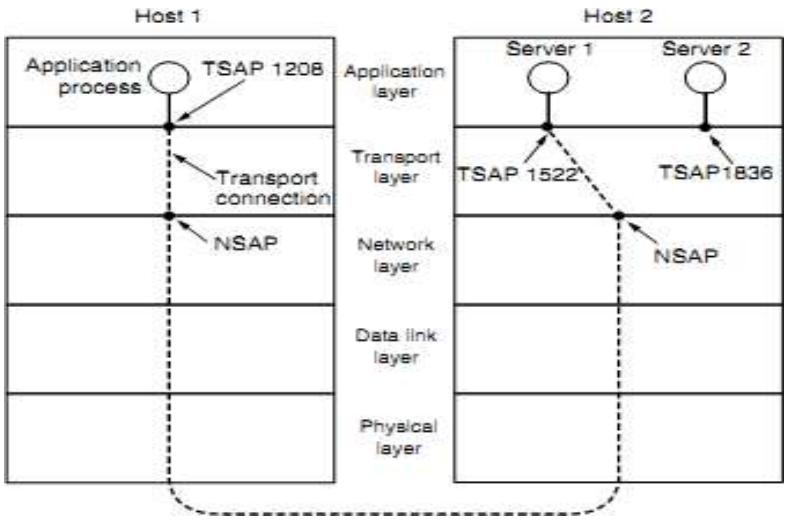


Fig 4.5: TSAP and NSAP network connections

Application processes, both clients and servers, can attach themselves to a local TSAP to establish a connection to a remote TSAP. These connections run through NSAPs on each host. The purpose of having TSAPs is that in some networks, each computer has a single NSAP, so some way is needed to distinguish multiple transport endpoints that share that NSAP.

A possible scenario for a transport connection is as follows:

1. A mail server process attaches itself to TSAP 1522 on host 2 to wait for an incoming call. How a process attaches itself to a TSAP is outside the networking model and depends entirely on the local operating system. A call such as our LISTEN might be used, for example.
2. An application process on host 1 wants to send an email message, so it attaches itself to TSAP 1208 and issues a CONNECT request. The request specifies TSAP 1208 on host 1 as the source and TSAP 1522 on host 2 as the destination. This action ultimately results in a transport connection being established between the application process and the server.
3. The application process sends over the mail message.
4. The mail server responds to say that it will deliver the message.
5. The transport connection is released.

2. CONNECTION ESTABLISHMENT:

With packet lifetimes bounded, it is possible to devise a fool proof way to establish connections safely. Packet lifetime can be bounded to a known maximum using one of the following techniques:

- Restricted subnet design
- Putting a hop counter in each packet
- Time stamping in each packet

Using a 3-way hand shake, a connection can be established. This establishment protocol doesn't require both sides to begin sending with the same sequence number.

- The **first technique** includes any method that prevents packets from looping, combined with some way of bounding delay including congestion over the longest possible path. It is difficult, given that internets may range from a single city to international in scope.

- The **second method** consists of having the hop count initialized to some appropriate value and decremented each time the packet is forwarded. **The network protocol simply discards any packet whose hop counter becomes zero.**
- The **third method** requires each packet to bear the time it was created, with the routers agreeing to discard any packet older than some agreed-upon time.

In **fig (A)** Tomlinson (1975) introduced the **three-way handshake**.

- This establishment protocol involves one peer checking with the other that the connection request is indeed current. Host 1 chooses a sequence number, x , and sends a CONNECTION REQUEST segment containing it to host 2. Host 2 replies with an ACK segment acknowledging x and announcing its own initial sequence number, y.
- Finally, host 1 acknowledges host 2's choice of an initial sequence number in the first data segment that it sends

In **fig (B)** the first segment is a delayed duplicate CONNECTION REQUEST from an old connection.

- This segment arrives at host 2 without host 1's knowledge. Host 2 reacts to this segment by sending host1an ACK segment, in effect asking for verification that host 1 was indeed trying to set up a new connection.
- When host 1 rejects host 2's attempt to establish a connection, host 2 realizes that it was tricked by a delayed duplicate and abandons the connection. In this way, a delayed duplicate does no damage.

- The worst case is when both a delayed CONNECTION REQUEST and an ACK are floating around in the subnet.

In **fig (C)** previous example, host 2 gets a delayed CONNECTION REQUEST and replies to it.

- At this point, it is crucial to realize that host 2 has proposed using y as the initial sequence number for host 2 to host 1 traffic, knowing full well that no segments containing sequence number y or acknowledgements to y are still in existence.
- When the second delayed segment arrives at host 2, the fact that z has been acknowledged rather than y tells host 2 that this, too, is an old duplicate.
- The important thing to realize here is that there is no combination of old segments that can cause the protocol to fail and have a connection set up by accident when no one wants it.

TRANSPORT PROTOCOLS - UDP

The Internet has two main protocols in the transport layer, **a connectionless protocol** and a **connection-oriented** one. The protocols complement each other. The connectionless protocol is **UDP.** It does almost nothing beyond sending packets between applications, letting applications build their own protocols on top as needed.

The connection-oriented protocol is **TCP.** It does almost everything. It makes connections and adds reliability with retransmissions, along with flow control and congestion control, all on behalf of the applications that use it. Since UDP is a transport layer protocol that typically runs in the operating system and protocols that use UDP typically run in user s pace, these uses might be considered applications.

INTROUCTION TO UDP

- The Internet protocol suite supports a connectionless transport protocol called UDP (User Datagram Protocol). UDP provides a way for applications to send encapsulated IP datagrams without having to establish a connection.
- UDP transmits segments consisting of an 8-byte header followed by the pay-load. The two ports serve to identify the end-points within the source and destination machines.

- When a UDP packet arrives, its payload is handed to the process attached to the destination port. This attachment occurs when the BIND primitive. Without the port fields, the transport layer would not know what to do with each incoming packet. With them, it delivers the embedded segment to the correct application.

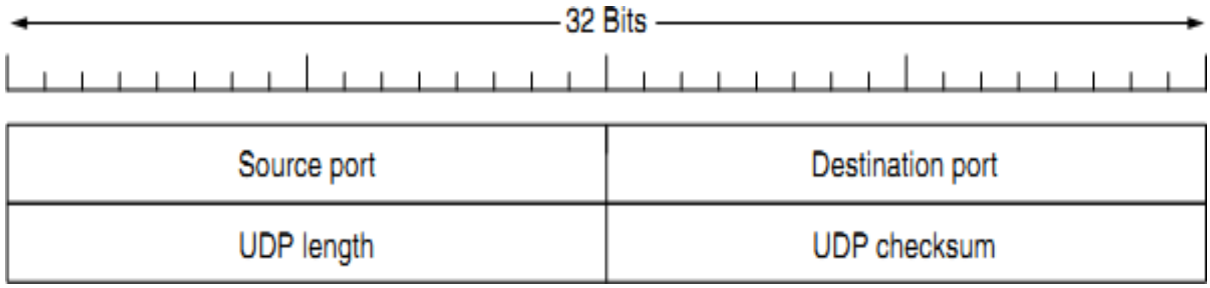


Fig 4.9: The UDP header

Source port, destination port: Identifies the end points within the source and destination machines.
UDP length: Includes 8-byte header and the data
UDP checksum: Includes the UDP header, the UDP data padded out to an even number of bytes if need be. It is an optional field

REMOTE PROCEDURE CALL

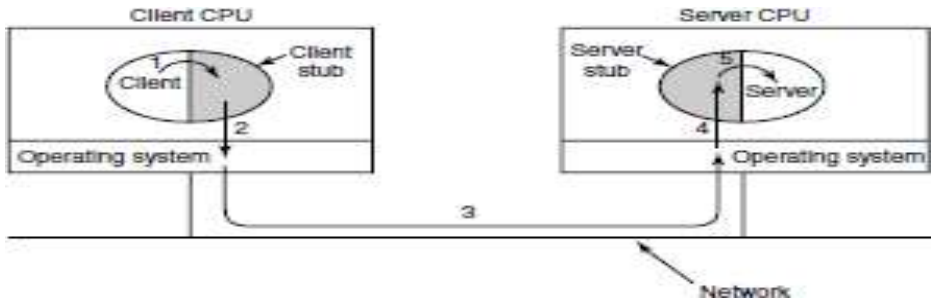


Fig 4.10: Steps in making a RPC

Step 1 is the client calling the client stub. This call is a local procedure call, with the parameters pushed onto the stack in the normal way.
Step 2 is the client stub packing the parameters into a message and making a system call to send the message. Packing the parameters is called **marshaling**.
Step 3 is the operating system sending the message from the client machine to the server machine.
Step 4 is the operating system passing the incoming packet to the server stub.
Step 5 is the server stub calling the server procedure with the **unmarshaled** parameters. The reply traces the same path in the other direction.

TCP (TRANSMISSION CONTROL PROTOCOL)

It was specifically designed to provide a reliable end-to end byte stream over an unreliable network. It was designed to adapt dynamically to properties of the inter network and to be robust in the face of many kinds of failures.

Each machine supporting TCP has a TCP transport entity, which accepts user data streams from local processes, breaks them up into pieces not exceeding 64kbytes and sends each piece as a separate IP datagram. When these datagrams arrive at a machine, they are given to TCP entity, which reconstructs the original byte streams. It is up to TCP to time out and retransmits them as needed, also to reassemble datagrams into messages in proper sequence.

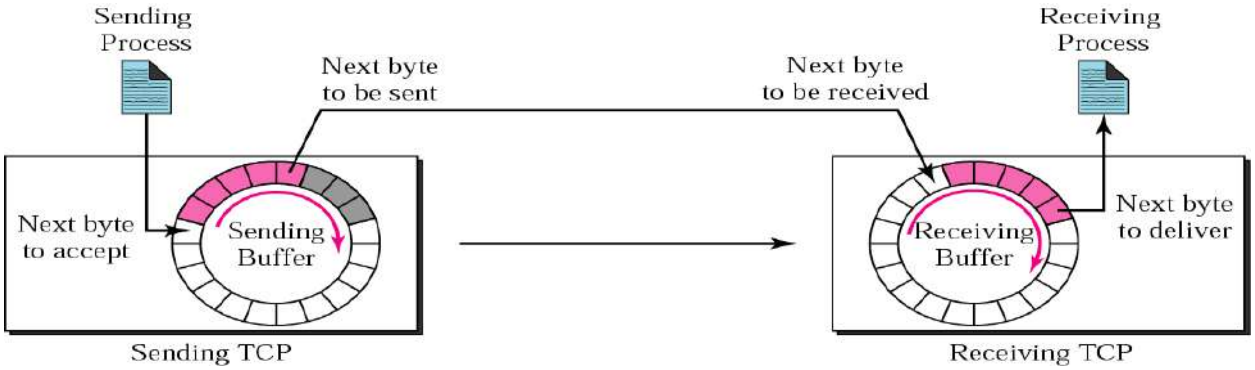
The different issues to be considered are:

1. The TCP Service Model
2. The TCP Protocol
3. The TCP Segment Header
4. The Connection Management
5. TCP Transmission Policy
6. TCP Congestion Control
7. TCP Timer Management.

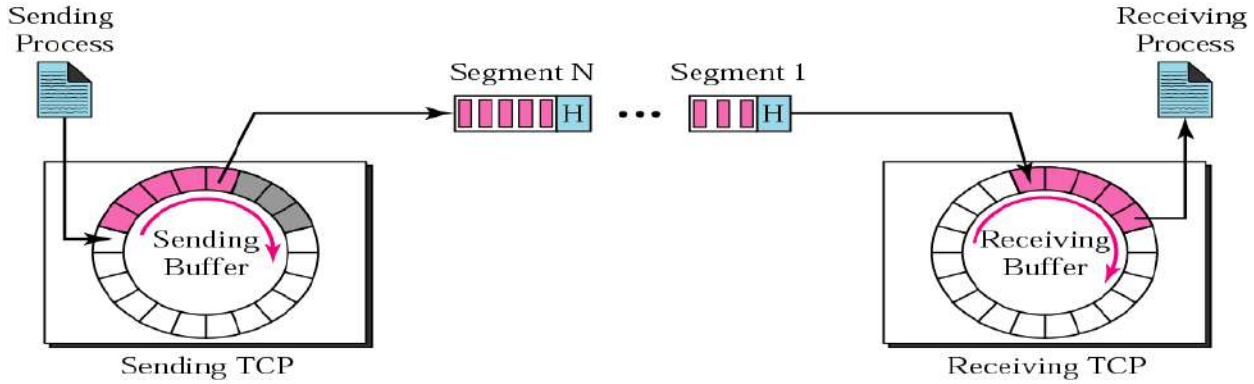
The TCP Service Model

- TCP service is obtained by having both the sender and receiver create end points called **SOCKETS**
- Each socket has a socket number(address)consisting of the IP address of the host, called a **"PORT"** (= TSAP)
- To obtain TCP service a connection must be explicitly established between a socket on the sending machine and a socket on the receiving machine
- All TCP connections are full duplex and point to point i.e., multicasting or broadcasting is not supported.
- A TCP connection is a byte stream, not a message stream i.e., the data is delivered as chunks

Sending and receiving buffers :



TCP segments :



Sockets:

A socket may be used for multiple connections at the same time. In other words, 2 or more connections may terminate at same socket. Connections are identified by socket identifiers at same socket. Connections are identified by socket identifiers at both ends. Some of the sockets are listed below:

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Ports: Port numbers below 256 are called Well- known ports and are reserved for standard services.

***Eg*:**

PORT-21	To establish a connection to a host to transfer a file using FTP
PORT-23	To establish a remote login session using TELNET

The TCP Protocol

- A key feature of TCP, and one which dominates the protocol design, is that every byte on a TCP connection has its own 32-bit sequence number.
- When the Internet began, the lines between routers were mostly 56-kbps leased lines, so a host blasting away at full speed took over 1 week to cycle through the sequence numbers.
- The basic protocol used by TCP entities is the **sliding window protocol**.
- When a sender transmits a segment, it also starts a timer.
- When the segment arrives at the destination, the receiving TCP entity sends back a segment (with data if any exist, otherwise without data) bearing an acknowledgement number equal to the next sequence number it expects to receive.

- If the sender's timer goes off before the acknowledgement is received, the sender transmits the segment again.

The TCP Segment Header

Every segment begins with a fixed-format, 20-byte header. The fixed header may be followed by header options. After the options, if any, up to 65,535 - 20 - 20 = 65,495 data bytes may follow, where the first 20 refer to the IP header and the second to the TCP header. Segments without any data are legal and are commonly used for acknowledgements and control messages.

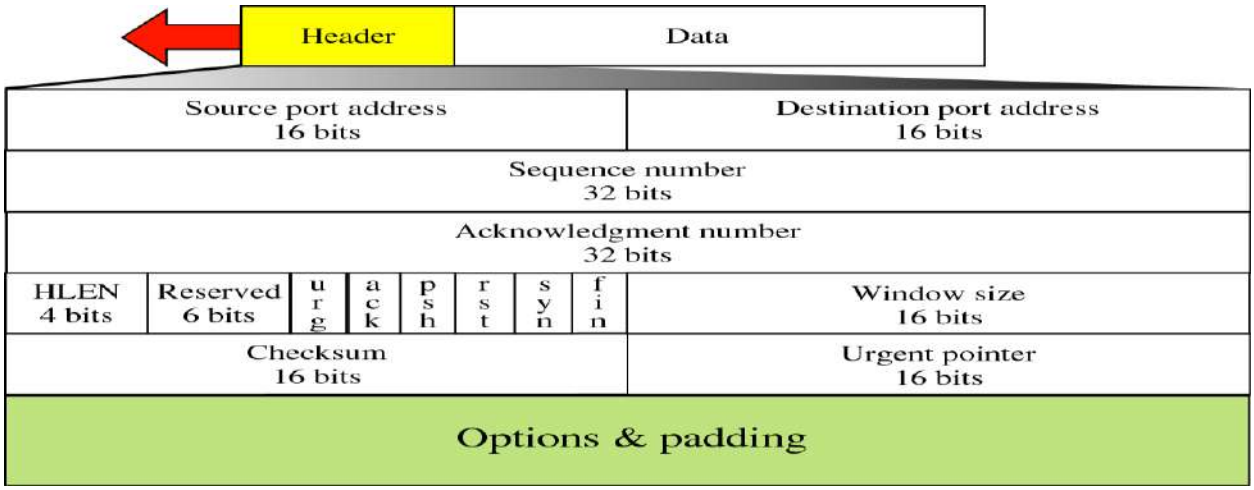


Fig 4.11: The TCP Header Format

Source Port, Destination Port : Identify local end points of the connections

Sequence number: Specifies the sequence number of the segment

Acknowledgement Number: Specifies the next byte expected.

TCP header length: Tells how many 32-bit words are contained in TCP header

URG: It is set to 1 if URGENT pointer is in use, which indicates start of urgent data.

ACK: It is set to 1 to indicate that the acknowledgement number is valid.

PSH: Indicates pushed data

RST: It is used to reset a connection that has become confused due to reject an invalid segment or refuse an attempt to open a connection.

FIN: Used to release a connection.

SYN: Used to establish connections.

1. Source Port-

Source Port is a 16 bit field.

It identifies the port of the sending application.

2. Destination Port-

- Destination Port is a 16 bit field.
- It identifies the port of the receiving application.

3.Sequence Number-

- Sequence number is a 32 bit field.
- TCP assigns a unique sequence number to each byte of data contained in the TCP segment.
- This field contains the sequence number of the first data byte.

4. Acknowledgement Number-

- Acknowledgment number is a 32 bit field.
- It contains sequence number of the data byte that receiver expects to receive next from the sender.
- It is always sequence number of the last received data byte incremented by 1.

5. Header Length-

- Header length is a 4 bit field.
- It contains the length of TCP header.
- It helps in knowing from where the actual data begins.

Minimum and Maximum Header length-

The length of TCP header always lies in the range-

[20 bytes , 60 bytes]

- The initial 5 rows of the TCP header are always used.
- So, minimum length of TCP header = 5 x 4 bytes = 20 bytes.
- The size of the 6th row representing the Options field vary.
- The size of Options field can go up to 40 bytes.

- So, maximum length of TCP header = 20 bytes + 40 bytes = 60 bytes.

Concept of Scaling Factor-

- Header length is a 4 bit field.
- So, the range of decimal values that can be represented is [0, 15].
- But the range of header length is [20, 60].
- So, to represent the header length, we use a scaling factor of 4.

In general,

Header length = Header length field value x 4 bytes

Examples-

- If header length field contains decimal value 5 (represented as 0101), then-
- Header length = 5 x 4 = 20 bytes
- If header length field contains decimal value 10 (represented as 1010), then-
- Header length = 10 x 4 = 40 bytes
- If header length field contains decimal value 15 (represented as 1111), then-
- Header length = 15 x 4 = 60 bytes

6. Reserved Bits-

- The 6 bits are reserved.
- These bits are not used.

7. URG Bit-

URG bit is used to treat certain data on an urgent basis.

When URG bit is set to 1,

- It indicates the receiver that certain amount of data within the current segment is urgent.
- Urgent data is pointed out by evaluating the urgent pointer field.
- The urgent data has be prioritized.
- Receiver forwards urgent data to the receiving application on a separate channel.

8. ACK Bit-

ACK bit indicates whether acknowledgement number field is valid or not.

- When ACK bit is set to 1, it indicates that acknowledgement number contained in the TCP header is valid.
- For all TCP segments except request segment, ACK bit is set to 1.
- Request segment is sent for connection establishment during **Three Way Handshake**.

9. PSH Bit-

PSH bit is used to push the entire buffer immediately to the receiving application.

When PSH bit is set to 1,

- All the segments in the buffer are immediately pushed to the receiving application.
- No wait is done for filling the entire buffer.
- This makes the entire buffer to free up immediately.

10. RST Bit-

RST bit is used to reset the TCP connection.

When RST bit is set to 1,

- It indicates the receiver to terminate the connection immediately.
- It causes both the sides to release the connection and all its resources abnormally.
- The transfer of data ceases in both the directions.
- It may result in the loss of data that is in transit.

This is used only when-

- There are unrecoverable errors.
- There is no chance of terminating the TCP connection normally.

11. SYN Bit-

SYN bit is used to synchronize the sequence numbers.

When SYN bit is set to 1,

- It indicates the receiver that the sequence number contained in the TCP header is the initial sequence number.
- Request segment sent for connection establishment during Three way handshake contains SYN bit set to 1.
- **12. FIN Bit-**

FIN bit is used to terminate the TCP connection.

When FIN bit is set to 1,

- It indicates the receiver that the sender wants to terminate the connection.
- FIN segment sent for TCP Connection Termination contains FIN bit set to 1.

13. Window Size-

- Window size is a 16 bit field.
- It contains the size of the receiving window of the sender.
- It advertises how much data (in bytes) the sender can receive without acknowledgement.
- Thus, window size is used for **Flow Control**.

14. Checksum-

- Checksum is a 16 bit field used for error control.
- It verifies the integrity of data in the TCP payload.
- Sender adds CRC checksum to the checksum field before sending the data.
- Receiver rejects the data that fails the CRC check.

Also Read- CRC | Checksum

15. Urgent Pointer-

- Urgent pointer is a 16 bit field.
- It indicates how much data in the current segment counting from the first data byte is urgent.
- Urgent pointer added to the sequence number indicates the end of urgent data byte.
- This field is considered valid and evaluated only if the URG bit is set to 1.

16. Options-

- Options field is used for several purposes.
- The size of options field vary from 0 bytes to 40 bytes.

Options field is generally used for the following purposes-

1. Time stamp
2. Window size extension
3. Parameter negotiation

4. Padding

A. Time Stamp-

When wrap around time is less than life time of a segment,

- Multiple segments having the same sequence number may appear at the receiver side.
- This makes it difficult for the receiver to identify the correct segment.
- If time stamp is used, it marks the age of TCP segments.
- Based on the time stamp, receiver can identify the correct segment.

B. Window Size Extension-

- Options field may be used to represent a window size greater than 16 bits.
- Using window size field of TCP header, window size of only 16 bits can be represented.
- If the receiver wants to receive more data, it can advertise its greater window size using this field.
- The extra bits are then appended in Options field.

C. Parameter Negotiation-

Options field is used for parameters negotiation.

Example- During connection establishment,

- Both sender and receiver have to specify their maximum segment size.
- To specify maximum segment size, there is no special field.
- So, they specify their maximum segment size using this field and negotiates.
-

D. Padding-

- Addition of dummy data to fill up unused space in the transmission unit and make it conform to the standard size is called as padding.
- Options field is used for padding.

Example-

- When header length is not a multiple of 4, extra zeroes are padded in the Options field.
- By doing so, header length becomes a multiple of 4.
- If header length = 30 bytes, 2 bytes of dummy data is added to the header.
- This makes header length = 32 bytes.
- Then, the value 32 / 4 = 8 is put in the header length field.
- In worst case, 3 bytes of dummy data might have to be padded to make the header length a multiple of 4.

TCP Connection Establishment

To establish a connection, one side, say, the server, passively waits for an incoming connection by executing the LISTEN and ACCEPT primitives, either specifying a specific source or nobody in particular.

The other side, say, the client, executes a CONNECT primitive, specifying the IP address and port to which it wants to connect, the maximum TCP segment size it is willing to accept, and optionally some user data (e.g., a password).

The CONNECT primitive sends a TCP segment with the *SYN* bit on and *ACK* bit off and waits for a response.

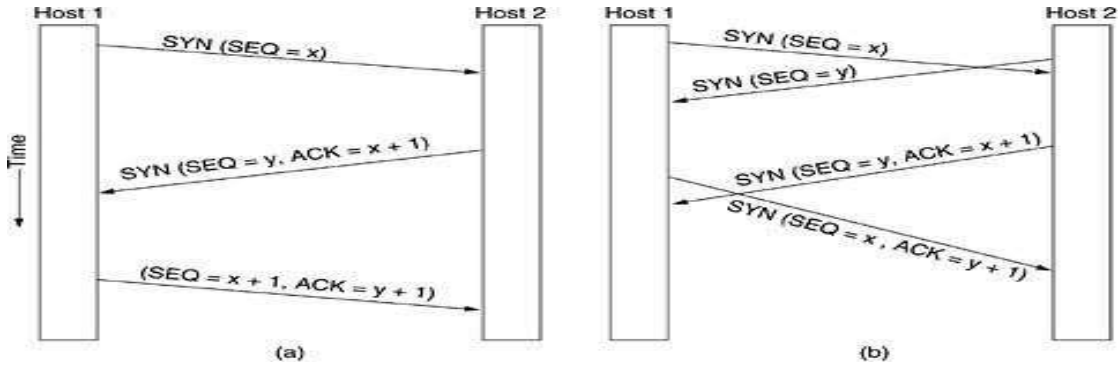


Fig 4.12: a) TCP Connection establishment in the normal case b) Call Collision

TCP Connection Release

- Although TCP connections are full duplex, to understand how connections are released it is best to think of them as a pair of simplex connections.
- Each simplex connection is released independently of its sibling. To release a connection, either party can send a TCP segment with the *FIN* bit set, which means that it has no more data to transmit.
- When the *FIN* is acknowledged, that direction is shut down for new data. Data may continue to flow indefinitely in the other direction, however.
- When both directions have been shut down, the connection is released.
- Normally, four TCP segments are needed to release a connection, one *FIN* and one *ACK* for each direction. However, it is possible for the first *ACK* and the second *FIN* to be contained in the same segment, reducing the total count to three.

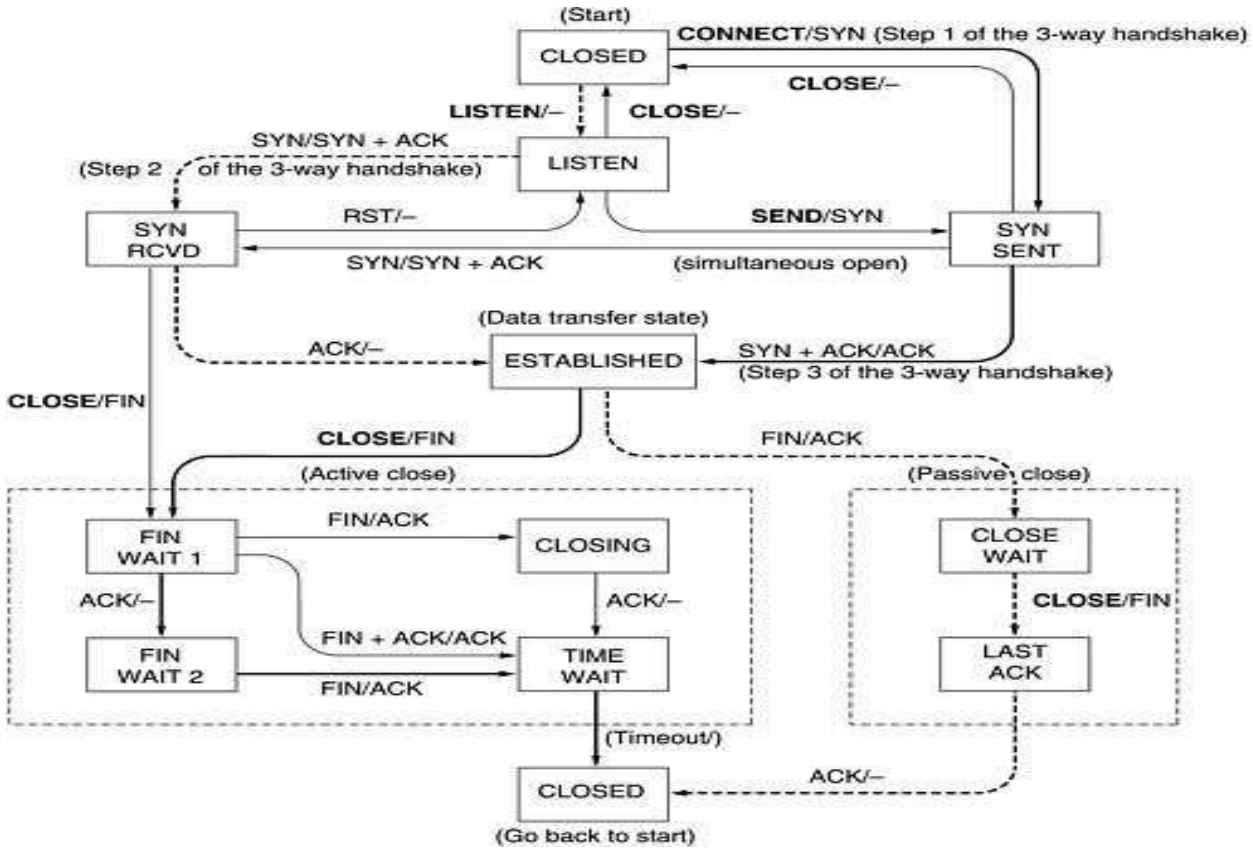


Figure 4.14 - TCP connection management finite state machine.

TCP Connection management from server's point of view:

1. The server does a **LISTEN** and settles down to see who turns up.
2. When a **SYN** comes in, the server acknowledges it and goes to the **SYNRCVD** state
3. When the servers **SYN** is itself acknowledged the 3-way handshake is complete and server goes to the **ESTABLISHED** state. Data transfer can now occur.
4. When the client has had enough, it does a close, which causes a **FIN** to arrive at the server [dashed box marked passive close].
5. The server is then signaled.
6. When it too, does a **CLOSE**, a **FIN** is sent to the client.
7. When the client's acknowledgement shows up, the server releases the connection and deletes the connection record.

TCP Transmission Policy

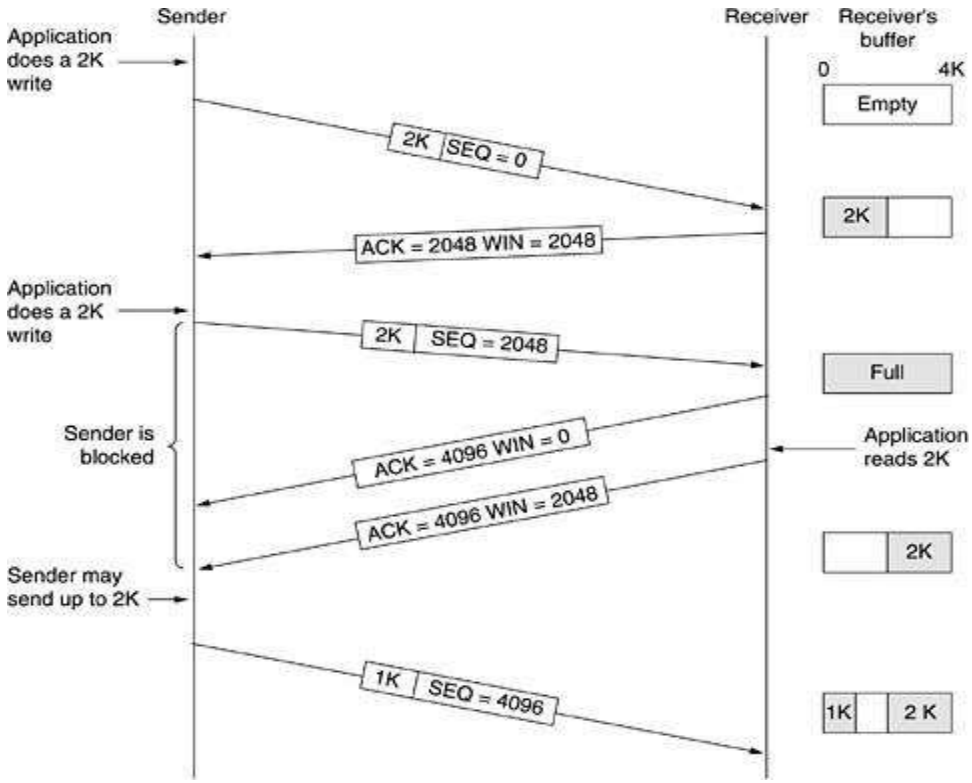
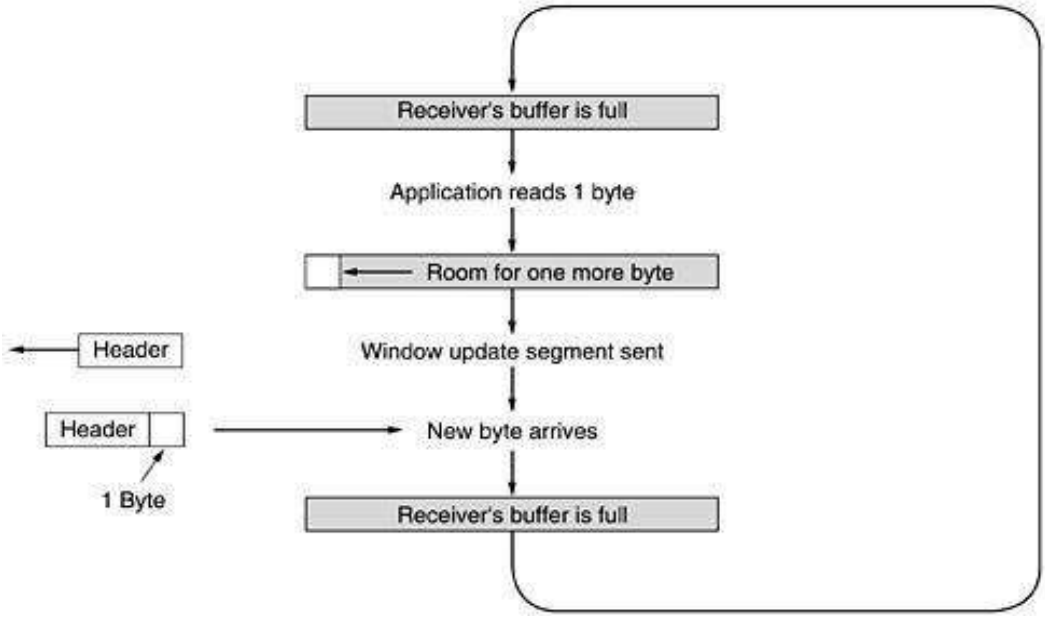


Figure 4.15 - Window management in TCP.

1. In the above example, the receiver has 4096-byte buffer.
2. If the sender transmits a 2048-byte segment that is correctly received, the receiver will acknowledge the segment.
3. Now the receiver will advertise a window of 2048 as it has only 2048 of buffer space, now.
4. Now the sender transmits another 2048 bytes which are acknowledged, but the advertised window is'0'.
5. The sender must stop until the application process on the receiving host has removed some data from the buffer, at which time TCP can advertise a layer window.

SILLY WINDOW SYNDROME:

This is one of the problems that ruin the TCP performance, which occurs when data are passed to the sending TCP entity in large blocks, but an interactive application on the receiving side reads 1 byte at a time.



- Initially the TCP buffer on the receiving side is full and the sender knows this(win=0)
- Then the interactive application reads 1 character from tcp stream.
- Now, the receiving TCP sends a window update to the sender saying that it is all right to send 1 byte.
- The sender obligates and sends 1 byte.
- The buffer is now full, and so the receiver acknowledges the 1 byte segment but sets window to zero. This behavior can go on forever.

TCP CONGESTION CONTROL:

TCP does to try to prevent the congestion from occurring in the first place in the following way:

When a connection is established, a suitable window size is chosen and the receiver specifies a window based on its buffer size. If the sender sticks to this window size, problems will not occur due to buffer overflow at the receiving end. But they may still occur due to internal congestion within the network. Let's see this problem occurs.

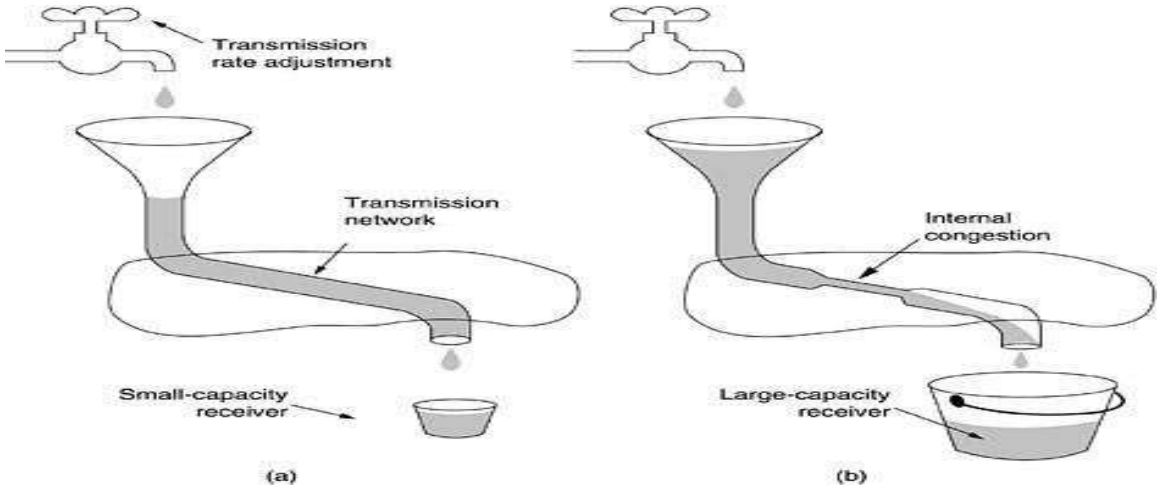


Figure 4.16. (a) A fast network feeding a low-capacity receiver. (b)A slow network feeding a high-capacity receiver.

In fig (a): We see a thick pipe leading to a small- capacity receiver. As long as the sender does not send more water than the bucket can contain, no water will be lost.

In fig (b): The limiting factor is not the bucket capacity, but the internal carrying capacity of the n/w. if too much water comes in too fast, it will backup and some will be lost.

- When a connection is established, the sender initializes the congestion window to the size of the max segment in use our connection.
- It then sends one max segment .if this max segment is acknowledged before the timer goes off, it adds one segment s worth of bytes to the congestion window to make it two maximum size segments and sends 2 segments.
- As each of these segments is acknowledged, the congestion window is increased by one max segment size.
- When the congestion window is 'n' segments, if all 'n' are acknowledged on time, the congestion window is increased by the byte count corresponding to 'n' segments.
- The congestion window keeps growing exponentially until either a time out occurs or the receiver's window is reached.
- The internet congestion control algorithm uses a third parameter, the **"threshold"** in addition to receiver and congestion windows.

TCP TIMER MANAGEMENT:

TCP uses 3 kinds of timers:

1. Retransmission timer
2. Persistence timer
3. Keep-Alive timer.

1. Retransmission timer: When a segment is sent, a timer is started. If the segment is acknowledged before the timer expires, the timer is stopped. If on the other hand, the timer goes off before the acknowledgement comes in, the segment is retransmitted and the timer is started again. The algorithm that constantly adjusts the time-out interval, based on continuous measurements of n/w performance was proposed by JACOBSON and works as follows:

- For each connection, TCP maintains a variable RTT, that is the best current estimate of the round trip time to the destination.
- When a segment is sent, a timer is started, both to see how long the acknowledgement takes and to trigger a retransmission if it takes too long.
- If the acknowledgement gets back before the timer expires, TCP measures how long the measurements took say M
- It then updates RTT according to the formula

$$\mathbf{RTT = \alpha RTT + (1-\alpha) M}$$

Where α = *a smoothing factor that determines how much weight is given to the old value. Typically,* α ***=7/8***
Retransmission timeout is calculated as

$$D = \alpha D + (1-\alpha) | RTT-M |$$

Where D = another smoothed variable, Mean RTT = expected acknowledgement value
M = observed acknowledgement value

$$Timeout = RTT+(4*D)$$

2. Persistence timer:
It is designed to prevent the following deadlock:

- The receiver sends an acknowledgement with a window size of '0' telling the sender to wait later, the receiver updates the window, but the packet with the update is lost now both the sender and receiver are waiting for each other to do something
- when the persistence timer goes off, the sender transmits a probe to the receiver the response to the probe gives the window size
- if it is still zero, the persistence timer is set again and the cycle repeats
- if it is non zero, data can now be sent

3. Keep-Alive timer: When a connection has been idle for a long time, this timer may go off to cause one side to check if other side is still there. If it fails to respond, the connection is terminated.

UNIT – VI.
APPLICATION LAYER

The Application Layer- DNS - Name Space - Resource Records - Name Servers- E-Mail - Architecture And Services – The User Agent –Message Format –Message Transfer –Final Delivery –

- Application Layer Protocols-

- Domain Name Service (DNS)
- Hyper Text Transfer Protocol (HTTP)
- Simple Mail Transfer Protocol (SMTP)
- Post Office Protocol (POP)
- File Transfer Protocol (FTP)

DOMAIN NAME SYSTEM

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. ... The Internet maintains two principal namespaces, the **domain name hierarchy and the Internet Protocol (IP) address spaces**.

This is primarily used for mapping host and e-mail destinations to IP addresses but can also be used other purposes.

Working:-

- To map a name onto an IP address, an application program calls a library procedure called Resolver, passing it the name as a parameter.
- The resolver sends a UDP packet to a local DNS server, which then looks up the name and returns the IP address to the resolver, which then returns it to the caller.
- Armed with the IP address, the program can then establish a TCP connection with the destination, or send it UDP packets.

1. **The DNS name space.**
2. **Resource Records.**
3. **Name Servers.**

1. THE DNS NAME SPACE:

The Internet is divided into several hundred top level domains, where each domain covers many hosts. Each domain is partitioned into sub domains, and these are further partitioned as so on. All these domains can be represented by a tree, in which the leaves represent domains that have no sub domains. A leaf domain may contain a single host, or it may represent a company and contains thousands of hosts. Each domain is named by the path upward from it to the root. The components are separated by periods (pronounced "dot")

Eg: Sun Microsystems Engg. Department = eng.sun.com.

The top domain comes in 2 flavours:-

- **Generic:** com(commercial), edu(educational instructions), mil(the U.S armed forces, government), int (certain international organizations), net(network providers), org (non profit organizations).
- **Country:** include 1 entry for every country. Domain names can be either absolute (ends with a period e.g. eng.sum.com) or relative (doesn't end with a period). Domain names are case sensitive and the component names can be up to 63 characters long and full path names must not exceed 255 characters.

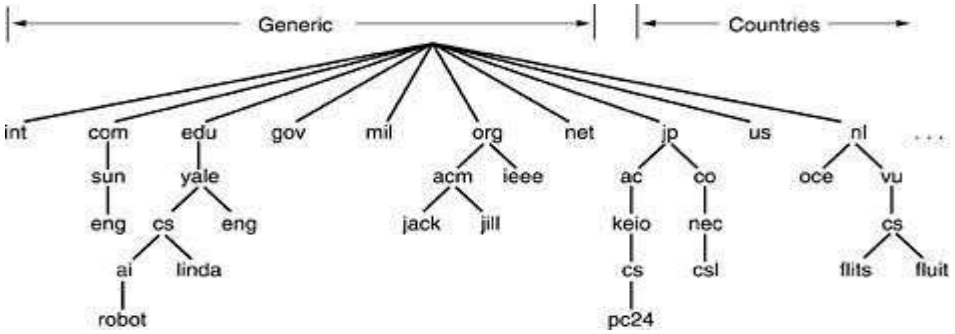


Figure 5-1. A portion of the Internet domain name space.

Insertions of a domain into the tree can be done in 2 days:-

• Under a generic domain (Eg: cs.yale.edu)

• Under the domain of their country (E.g: cs.yale.ct.us)

2. RESOURCE RECORDS:

Every domain can have a set of resource records associated with it. For a single host, the most common resource record is just its IP address. When a resolver gives a domain name to DNS, it gets both the resource records associated with that name i.e., the real function of DNS is to map domain names into resource records. A resource record is a 5-tuple and its format is as follows:

Domain	Name	Time to live	Type	Class	Value

Domain _name : Tells the domain to which this record applies.
Time- to- live : Gives an identification of how stable the record is (High Stable = 86400 i.e. no. of seconds /day) (High Volatile = 1 min)
Type: Tells what kind of record this is.
Class: It is IN for the internet information and codes for non internet information
Value: This field can be a number a domain name or an ASCII string

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text
A	**(Address)**	**IPv4 address corresponding to a host**
AAAA	**(Address)**	**IPv6 address corresponding to a host**.

NAME SERVERS:

It contains the entire database and responds to all queries about it. DNS name space is divided up into non- overlapping zones, in which each zone contains some part of the tree and also contains name servers holding the authoritative information about that zone.

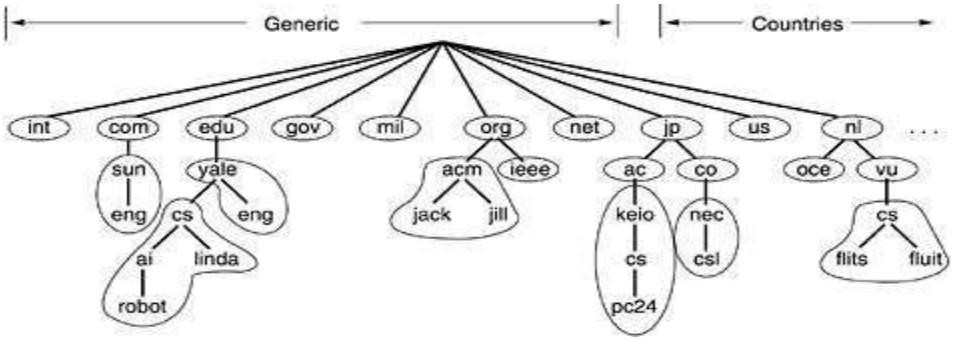


Figure 5-2. Part of the DNS name space showing the division into zones.

When a resolver has a query about a domain name, it passes the query to one of the local name servers:

1. If the domain being sought falls under the jurisdiction of name server, it returns the authoritative resource records(that comes from the authority that manages the record, and is always correct).
2. If the domain is remote and no information about the requested domain is available locally the name server sends a query message to the top level name server for the domain requested.

E.g.: A resolver of flits.cs.vle.nl wants to know the IP address of the host Linda.cs.yale.edu

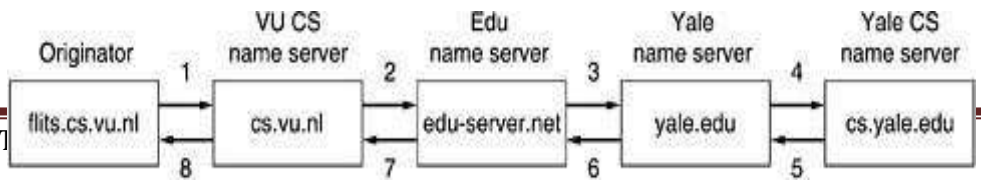


Figure 5-3. How a resolver looks up a remote name in eight steps.

Step 1: Resolver sends a query containing domain name sought the type and the class to local name server, cs.vu.nl.

Step 2: Suppose local name server knows nothing about it, it asks few others nearby name servers. If none of them know, it sends a UDP packet to the server for edu-server.net.

Step 3: This server knows nothing about Linda.cs.yale.edu or cs.yale.edu and so it forwards the request to the name server for yale.edu.

Step 4: This one forwards the request to cs.yale.edu which must have authoritative resource records.

Step 5 to 8: The resource record requested works its way back in steps 5-8 This query method is known as **Recursive Query**

ELECTRONIC MAIL

1. ARCHITECTURE AND SERVICES:

E-mail systems consist of two subsystems. They are:-

(1). User Agents, which allow people to read and send e-mail

(2). Message Transfer Agents, which move messages from source to destination E-mail systems support 5 basic functions:-

a. Composition
b. Transfer
c. Reporting
d. Displaying
e. Disposition

(a). Composition: It refers to the process of creating messages and answers. Any text editor is used for body of the message. While the system itself can provide assistance with addressing and numerous header fields attached to each message.

(b). Reporting: It has to do with telling the originator what happened to the message that is, whether it was delivered, rejected (or) lost.

(c). Transfer: It refers to moving messages from originator to the recipient.

(d). Displaying: Incoming messages are to be displayed so that people can read their email.

(e). Disposition: It concerns what the recipient dose with the message after receiving it. Possibilities include throwing it away before reading (or) after reading, saving it and so on.

Most systems allow users to create **mailboxes** to store incoming e-mail. Commands are needed to create and destroy mailboxes, inspect the contents of mailboxes, insert and delete messages from mailboxes, and so on.

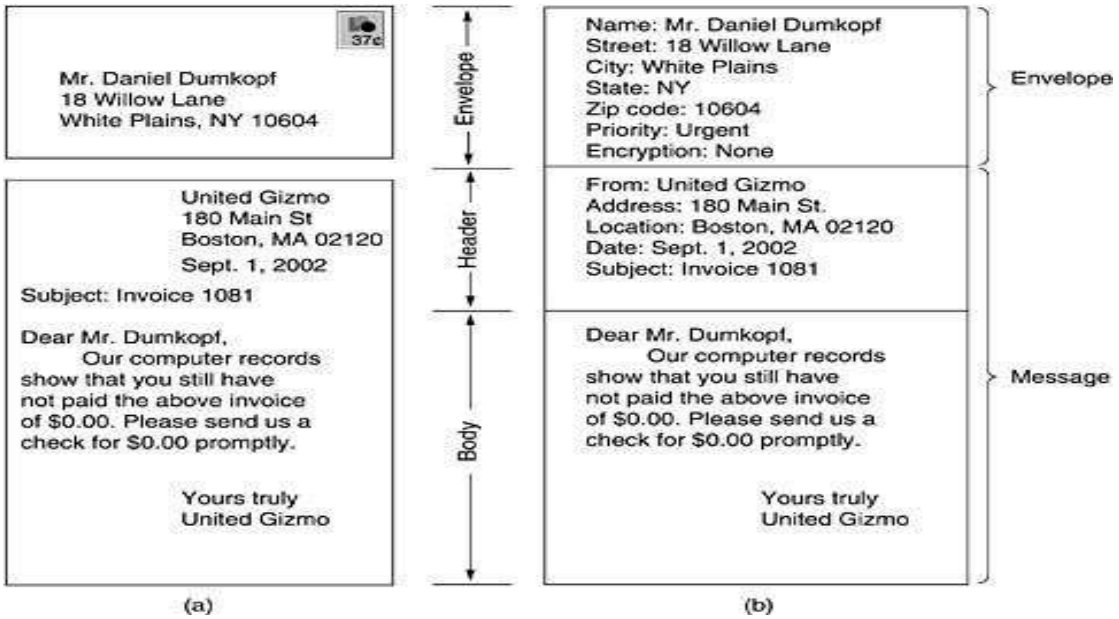


Figure 5-4: Envelopes and messages. (a) Paper mail. (b) Electronic mail.

(1) THE USER AGENT

A user agent is normally a program (sometimes called a mail reader) that accepts a variety of commands for composing, receiving, and replying to messages, as well as for manipulating mailboxes.

SENDING E-MAIL

To send an e-mail message, a user must provide the message, the destination address, and possibly some other parameters. The message can be produced with a free-standing text editor, a word processing program, or possibly with a specialized text editor built into the user agent. The destination address must be in a format that the user agent can deal with. Many user agents expect addresses of the form *user@dns-address*.

READING E-MAIL

When a user agent is started up, it looks at the user's mailbox for incoming e-mail before displaying anything on the screen. Then it may announce the number of messages in the mailbox or display a one-line summary of each one and wait for a command.

(2) MESSAGE

FORMATS RFC 822

Messages consist of a primitive envelope (described in RFC 821), some number of header fields, a blank line, and then the message body. Each header field (logically) consists of a single line of ASCII text containing the field name, a colon, and, for most fields, a value.

Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

Figure 5-5: RFC 822 header fields related to message transport

MIME — The Multipurpose Internet Mail Extensions

RFC 822 specified the headers but left the content entirely up to the users. Nowadays, on the worldwide Internet, this approach is no longer adequate. The problems include sending and receiving

1. Messages in languages with accents (e.g., French and German).
2. Messages in non-Latin alphabets (e.g., Hebrew and Russian).
3. Messages in languages without alphabets (e.g., Chinese and Japanese).
4. Messages not containing text at all (e.g., audio or images).

A solution was proposed in RFC 1341 called **MIME** (**Multipurpose Internet Mail Extensions**)

The basic idea of MIME is to continue to use the RFC 822 format, but to add structure to the message body and define encoding rules for non-ASCII messages. By not deviating from RFC 822, MIME messages can be sent using the existing mail programs and protocols. All that has to be changed are the sending and receiving programs, which users can do for themselves.

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

Figure 5-6: RFC 822 headers added by MIME

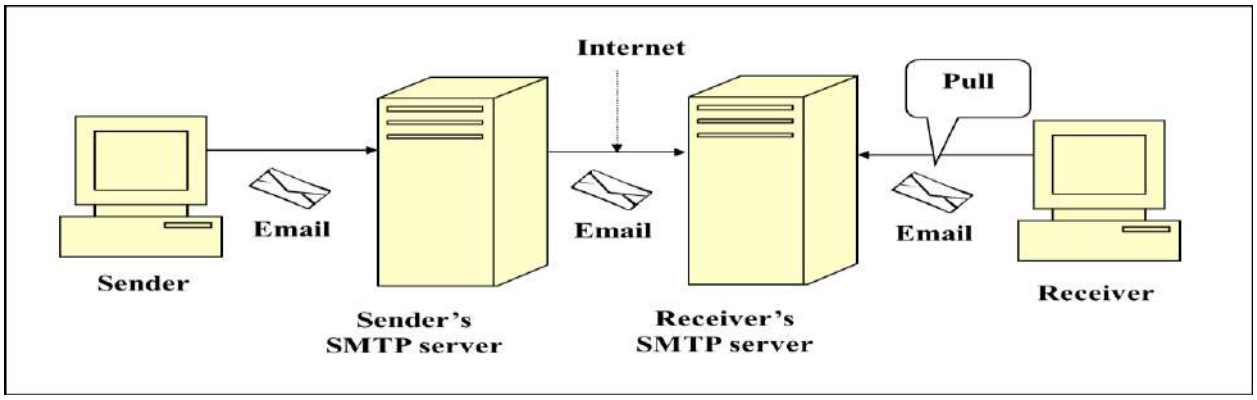
MESSAGE TRANSFER

The message transfer system is concerned with relaying messages from the originator to the recipient. The simplest way to do this is to establish a transport connection from the source machine to the destination machine and then just transfer the message.

SMTP—THE SIMPLE MAIL TRANSFER PROTOCOL

SMTP is a simple ASCII protocol. After establishing the TCP connection to port 25, the sending machine, operating as the client, waits for the receiving machine, operating as the server, to talk first. The server starts by sending a line of text giving its identity and telling whether it is prepared to receive mail. If it is not, the client releases the connection and tries again later.

Even though the SMTP protocol is completely well defined, a **few problems** can still arise.



One problem relates to message length. Some older implementations cannot handle messages exceeding 64 KB.

Another problem relates to timeouts. If the client and server have different timeouts, one of them may give up while the other is still busy, unexpectedly terminating the connection.

Finally, in rare situations, infinite mail storms can be triggered.

For example, if host 1 holds mailing list *A* and host 2 holds mailing list *B* and each list contains an entry for the other one, then a message sent to either list could generate a never-ending amount of e-mail traffic unless somebody checks for it.

FINAL DELIVERY

With the advent of people who access the Internet by calling their ISP over a modem, it breaks down.

One solution is to have a message transfer agent on an ISP machine accept e-mail for its customers and store it in their mailboxes on an ISP machine. Since this agent can be on-line all the time, e-mail can be sent to it 24 hours a day.

POP3

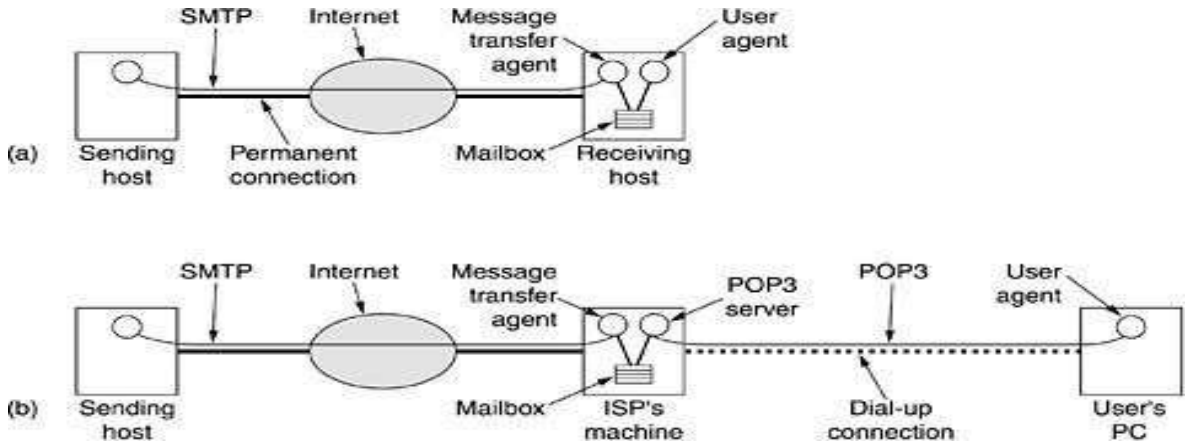


Figure:5-7

(a) Sending and reading mail when the receiver has a permanent Internet connection and the user agent runs on the same machine as the message transfer agent.

(b) Reading e-mail when the receiver has a dial-up connection to an ISP

POP3 begins when the user starts the mail reader. The mail reader calls up the ISP (unless there is already a connection) and establishes a TCP connection with the message transfer agent at port 110. Once the connection has been established, the POP3 protocol goes through three states in sequence:

1. Authorization.
2. Transactions.
3. Update.

The authorization state deals with having the user log in.

The transaction state deals with the user collecting the e-mails and marking them for deletion from the mailbox. The update state actually causes the e-mails to be deleted.

IMAP (**Internet Message Access Protocol**).

POP3 normally downloads all stored messages at each contact, the result is that the user's e-mail quickly gets spread over multiple machines, more or less at random; some of them not even the user's.
This disadvantage gave rise to an alternative final delivery protocol, **IMAP** (**Internet Message Access Protocol**).
IMAP assumes that all the e-mail will remain on the server indefinitely in multiple mailboxes. IMAP provides extensive mechanisms for reading messages or even parts of messages, a feature useful when using a slow modem to read the text part of a multipart message with large audio and video attachments.