Contents

UNIT 1 - INTRODUCTION TO PHYSICAL LAYER

Definition: Computer Network is a group of computers connected with each other through wires, optical fibers or optical links so that various devices can interact with each through a network.

The aim of the computer network is the sharing of resources among various devices.

1.1. USES OF COMPUTER NETWORK:

1.1.1 Business Application:

Most companies have a substantial number of computers. For example

1. **Resource Sharing:** The goal is to make all programs, equipment and especially data available to anyone on the network without regard to the physical location of the resource or the user.
2. **Sharing Physical Resources:** Such as printers and tape backup systems, is sharing information. Companies small and large are vitally dependent on computerized information.
3. **Network called VPNs: (Virtual Private Networks)** may be used to join the individual networks at different sites into one extended network.
4. **Client Server Model:**

   The data are stored on powerful computer called **servers.**

   Employees have simpler machines, called **client.**

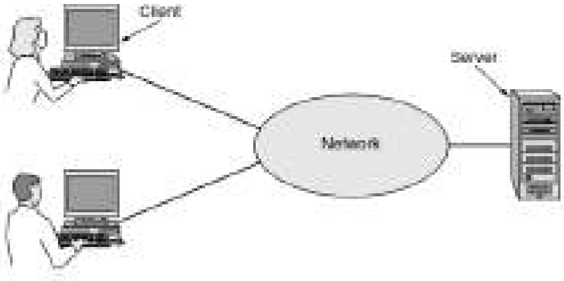


Figure 1-1. A network with two clients and one server.

5. **Client Server Model involves request and reply:**

   If we look at the client-server model in details, we see that two process (i.e., running programs) are involved, one on the client machine and one on the server machine. Communication takes the form of the client process sending a message over the network to the server process. The client process then waits for a reply message. When the server process gets the request, it performs the requested work or looks up the requested data and sends back a reply. These messages are shown in figure

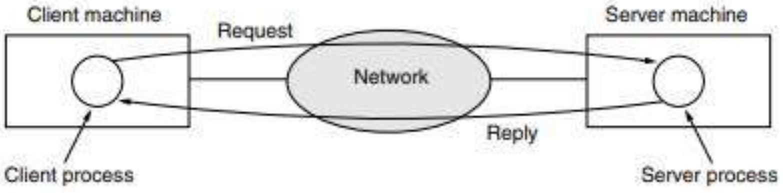


Figure 1-2. The client-server model involves requests and replies.

6. **Communication Medium:** A second goal of setting up a computer network has to do with people rather than information or even computer. A computer network can provide a powerful **communication medium** among employees.
7. **Electronic Mail (E-mail):** virtually every company that has two or more computers now has **email,** which employees generally use for a deal of daily communication.
8. **Desktop Sharing:** Video can be added to audio so that employees at distant location can see and hear each other as they hold a meeting. This technique is a powerful tool for eliminating the cost and time previously devoted to travel. **Desktop sharing** lets remote workers see and interact with a graphical computer screen.
9. A third goal for many companies is doing business electronically with customers and suppliers. This new model is called **e-commerce (electronic commerce).**

1.1.2. Home Applications:

- Recently, the biggest reason to buy a home computer was probably for **Internet access.**
- Internet access provides home users with connectivity to remote computers.
- Access to remote information comes in many forms. It can be **surfing the World Wide Web** for information or just for fun.
- Many **newspapers** have gone online and can be personalized.
- The next step beyond newspapers (plus magazines and scientific journals) is the **online digital library.** Many professional organization, such as the **ACM (www.acm.org)** and the **IEEE Computer Society (www.computer.org),** already have all their journals and conference proceedings online.

Peer- to- Peer Communication:

- In this form, individuals who from a loose group can communicate with others in the group, as shown in figure. Every person can in principal, communicate with one or more other people; there is no fixed division into client and server.

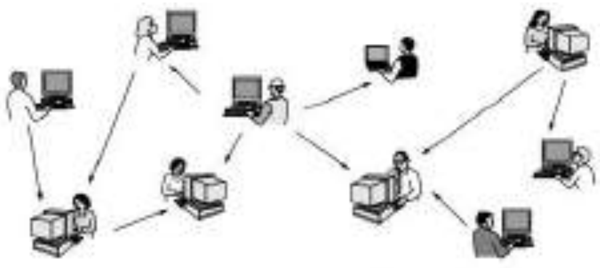


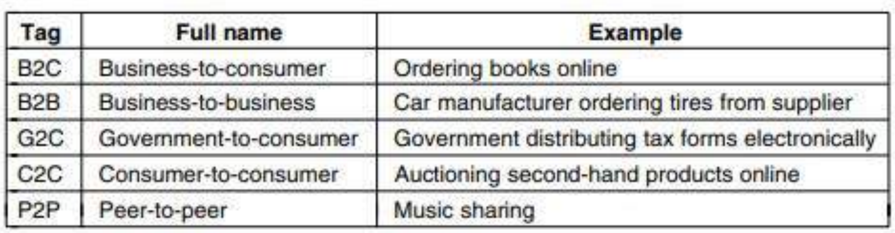
Figure 1-3. In a peer-to-peer system there are no fixed clients and servers.

- Peer-to Peer communication is often used to share music and video.

- **Twitter** service that lets people send short text messages called "tweets" to their circle of friends or other willing audiences.
- One of the most popular social networking sites is **Facebook.**
- The most famous wiki is the **Wikipedia,** an encyclopedia can edit, but there are thousands of other wikis.

Home Application- E-Commerce:

- Another in which e-commerce is widely used to financial institutions. Many people already pay their bills, manages their bank account, and handle their investments electronically. This trend will surely continue as networks become more secure.



Tag	Full name	Example
B2C	Business-to-consumer	Ordering books online
B2B	Business-to-business	Car manufacturer ordering tires from supplier
G2C	Government-to-consumer	Government distributing tax forms electronically
C2C	Consumer-to-consumer	Auctioning second-hand products online
P2P	Peer-to-peer	Music sharing

Figure 1-4. Some forms of e-commerce.

- Users can find, buy, and download MP3 songs and DVD quality movies and them to their personal collection. TV shows now reach many homes via **IPTV (IP Television)** system that are based on IP technology instead of cable TV or radio transmissions.
- Another form of entertainment is **game playing.** Already we have multi person real-time simulation games, like hide and seek in a virtual dungeon and fights simulators with the players on one team trying to shoot down the players on the opposite team.
- Device such as televisions that plug into wall can use **power-line network** to send information throughout the house over the wires that carry electricity.
- A technology called **RFID (Radio Frequency IDentification)** will push this idea even further in the future. This lets RFID readers locate and communicate with the items over a distance of up to several meters, depending on the kind of RFID.

1.1.3 Mobile users:

1. **Connectivity** to the internet enables many of these mobile uses.

2. Wireless **hotspots** based on the 802.11 standard are another kind of wireless network for mobile computers
3. Wireless networks are of great value to fleets of trucks, taxis, delivery vehicles, and repairpersons for keeping in contact with their home base.
4. Wireless networks are also important to the military. Here we see a distinction between **fixed wireless and mobile wireless** network. Even notebook computer are sometimes wired.
5. For example, if a traveler plug a notebook computer into wired network jack in a hotel room, he has mobility without a wireless network.
6. **Text messaging or texting** is tremendously popular. It lets a mobile phone user type a short message that is then delivered by the cellular network to another mobile subscriber.
7. Since mobile phones know their locations, often because they are equipped with **GPS (Global Positioning System)** receivers, some services are intentionally location dependent.
8. An area in which mobile phones are now starting to be used is **m-commerce (mobile commerce)**

1.1.4. Social Issues: - 1.

1. **Network neutrality:**

   If you are a big company and pay well then you get good service, but if you are a small-time player, you get poor service. Opponents of this practice argue that peer-to-peer and other content should be treated in the same way because they are all just bits to the network. This argument for communications that are not differentiated by their content or source or who is providing the content is known as network neutrality (Wu, 2003). It is probably safe to say that this debate will go on for a while.

2. **Profiling (Summarizing) users:**

   The government does not have a monopoly on threatening people's privacy. The private sector does its bit too by profiling users. For example, small files called cookies that Web browsers store on users' computers allow companies to track users' activities in cyberspace and may also allow credit card numbers, social security numbers, and other confidential information to leak all over the Internet (Berghel, 2001). Companies that provide Web-based services may maintain large amounts of personal information about their users that allows them to study user activities directly.

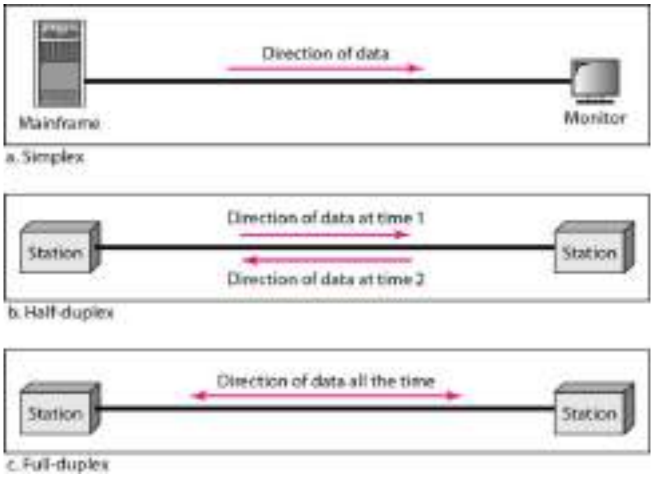
3. **Phishing:**

Phishing messages masquerade as originating from a trustworthy party, for example, your bank, to try to trick you into revealing sensitive information, for example, credit card numbers. Identity theft is becoming a serious problem as thieves collect enough information about a victim to obtain credit cards and other documents in the victim's name.

1.2. NETWORK HARDWARE

1.2.1. *Transmission Modes:*

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 (Figure a). Keyboards and traditional monitors are examples of simplex devices.

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 (Figure b). Walkie-talkies and CB (citizens band) radios are both half- duplex systems.

Full-Duplex

In full-duplex, both stations can transmit and receive simultaneously (Figure c). 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.

Simplex	**Half-duplex**	**Full-duplex**
It provides one-way communication.	It provides two-way communication but one way at a time.	It provides two-way communication at the same time.
A device can only send data, but it cannot receive data.	A device can send and receive data but one at a time.	A device can send and receive data at the same time.
It utilizes less bandwidth than half-duplex and full-duplex.	It utilizes more bandwidth than simplex but less than full-duplex.	It utilizes more bandwidth than simplex and half-duplex.
It uses one channel to transmit data.	It also uses one channel to transmit data.	It uses two separate channels to transmit data.
Keyboards and scanners are examples of simplex.	Hubs and old NICs are examples of half-duplex.	Switches and modern NICs are examples of full-duplex.

Network Criteria

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

Performance

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

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

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

1.2.2. *Type of Connection*

A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another.

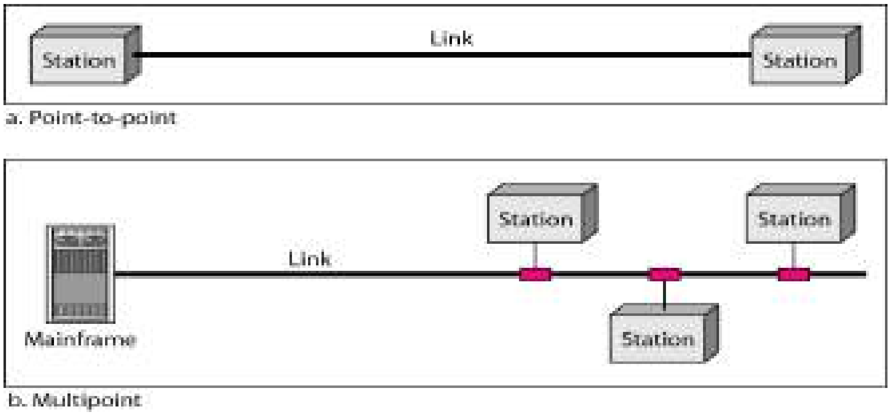
There are two possible types of connections: point-to-point and multipoint.

Point-to-Point A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible

When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

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

In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a spatially shared connection. If users must take turns, it is a timeshared connection.

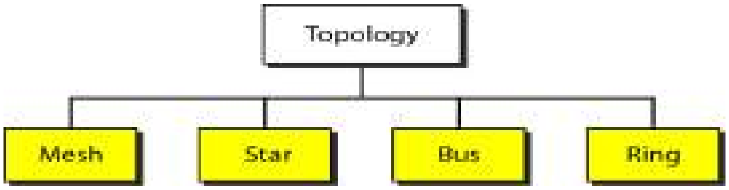


1.2.3. *Transmission topology*

The term physical topology refers to the way in which a network is laid out physically.

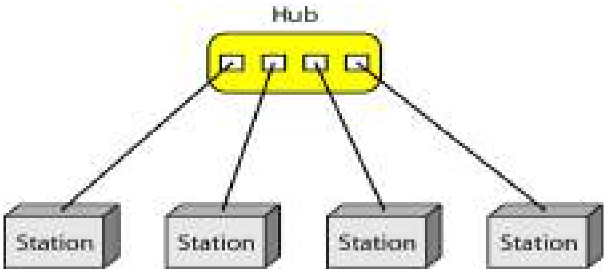
Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another.

There are four basic topologies possible: mesh, star, bus, and ring



STAR:

A **star network, star topology** is one of the most common network setups. In this configuration, every node connects to a central network device, like a hub, switch, or computer. The central network device acts as a server and the peripheral devices act as clients. Depending on the type of network card used in each computer of the star topology, a coaxial cable or a RJ-45 network cable is used to connect computers together.



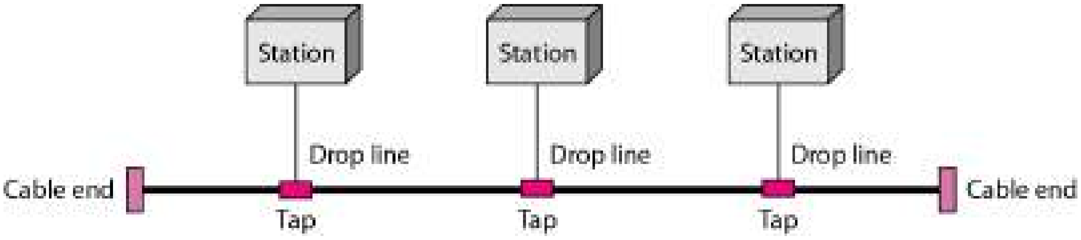
Advantages of star topology

- Centralized management of the network, through the use of the central computer, hub, or switch.
- Easy to add another computer to the network.
- If one computer on the network fails, the rest of the network continues to function normally.
- The star topology is used in local-area networks (LANs), High-speed LANs
- often use a star topology with a central hub.

Disadvantages of star topology

- Can have a higher cost to implement, especially when using a switch or router as the central network device.
- The central network device determines the performance and number of nodes the network can handle.
- If the central computer, hub, or switch fails, the entire network goes down and all computers are disconnected from the network

BUS:



a **line topology, a bus topology** is a network setup in which each computer and network device are connected to a single cable or backbone.

Advantages of bus topology

- It works well when you have a small network.
- It's the easiest network topology for connecting computers or peripherals in a linear fashion.
- It requires less cable length than a star topology.

Disadvantages of bus topology

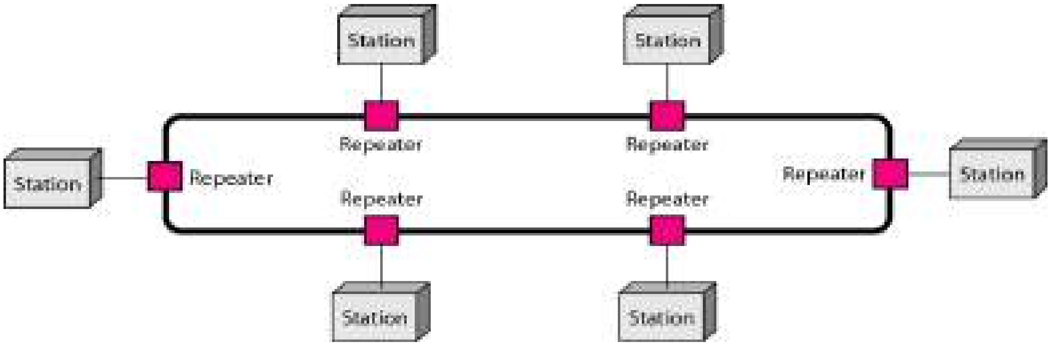
- It can be difficult to identify the problems if the whole network goes down.
- It can be hard to troubleshoot individual device issues.
- Bus topology is not great for large networks.
- Terminators are required for both ends of the main cable.
- Additional devices slow the network down.
- If a main cable is damaged, the network fails or splits into two.

RING:

A **ring topology** is a network configuration in which device connections create a circular data path. In a ring network, packets of data travel from one device to the next until they reach their destination. Most ring topologies allow packets to travel only in one direction, called **a unidirectional ring** network. Others permit data to move in either direction, called **bidirectional.**

The major disadvantage of a ring topology is that if any individual connection in the ring is broken, the entire network is affected.

Ring topologies may be used in either local area networks (LANs) or wide area networks (WANs).



Advantages of ring topology

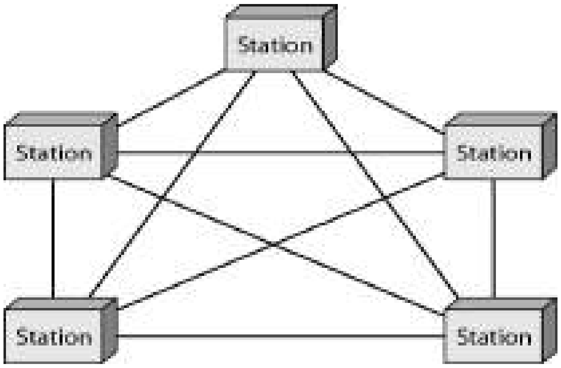
- All data flows in one direction, reducing the chance of packet collisions.
- A network server is not needed to control network connectivity between each workstation.
- Data can transfer between workstations at high speeds.
- Additional workstations can be added without impacting performance of the network.

Disadvantages of ring topology

- All data being transferred over the network must pass through each workstation on the network, which can make it slower than a star topology.
- The entire network will be impacted if one workstation shuts down.
- The hardware needed to connect each workstation to the network is more expensive than Ethernet cards and hubs/switches.

MESH:

A mesh topology is the one where every node is connected to every other node in the network. A mesh topology can be a **full mesh topology** or a **partially connected mesh topology.** In a ***full mesh topology***, every computer in the network has a connection to each of the other computers in that network. The number of connections in this network can be calculated using the following formula (n is the number of computers in the network): **n(n-1)/2**



In a partially connected mesh topology, at least two of the computers in the network have connections to multiple other computers in that network. It is an inexpensive way to implement redundancy in a network. In the event that one of the primary computers or connections in the network fails, the rest of the network continues to operate normally.

Advantages of a mesh topology

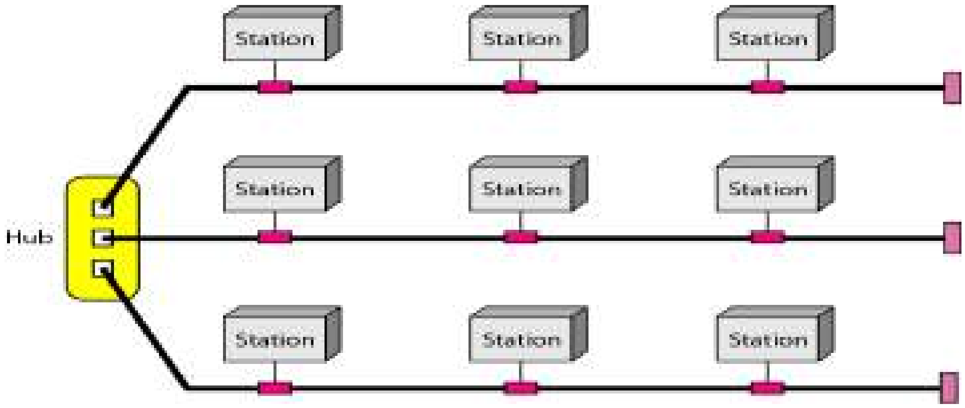
- Can handle high amounts of traffic, because multiple devices can transmit data simultaneously.
- A failure of one device does not cause a break in the network or transmission of data.
- Adding additional devices does not disrupt data transmission between other devices.

Disadvantages of a mesh topology

- The cost to implement is higher than other network topologies, making it a less desirable option.
- Building and maintaining the topology is difficult and time consuming.
- The chance of redundant connections is high, which adds to the high costs and potential for reduced efficiency.

Hybrid Topology

A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology as shown in Figure.



1.2.4. Transmission technology

The transmission technology can be categorized broadly into two types: **Broadcast networks** and **Point-to-point networks**

Broadcast networks:

- Broadcast networks have a single communication channel that is shared or used by all the machines on the network. Short messages called packets sent by any machine are received by all the others.

- Broadcast systems generally use a special code in the address field for addressing a packet to all the concerned computers. This mode of operation is called broadcasting.
- Some broadcast systems also support transmission to a subset of the machines known as **multicasting.**
- Upon receiving a packet, a machine checks the address field. If the packet is addressed to it then the packet is processed, otherwise the packet is ignored.

Point-to-point networks

- Point to point networks consists of many connections between individual pairs of machines. To go from the source to the destination a packet on these types of network may have to go through intermediate computers before they reach the desired computer.
- Often the packets have to follow multiple routes, of different lengths.
- Point-to-point transmission with exactly one sender and exactly one receiver is sometimes called unicasting.
- In contrast, on a broadcast network, the communication channel is shared by all the machines on the network; packets sent by any machine are received by all the others. A wireless network is a common example of a broadcast link, with communication shared over a coverage region that depends on the wireless channel and the transmitting machine.

1.2.5. Types of 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.

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.

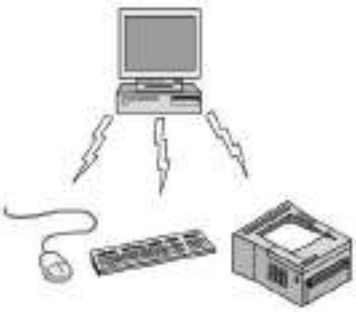


Figure 1-7. Bluetooth PAN configuration.

Local Area Network (LAN):

- Local Area Network is a group of computers connected to each other in a small area such as building, office.
- LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc.
- 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.
- LAN can be connected using a common cable or a Switch.

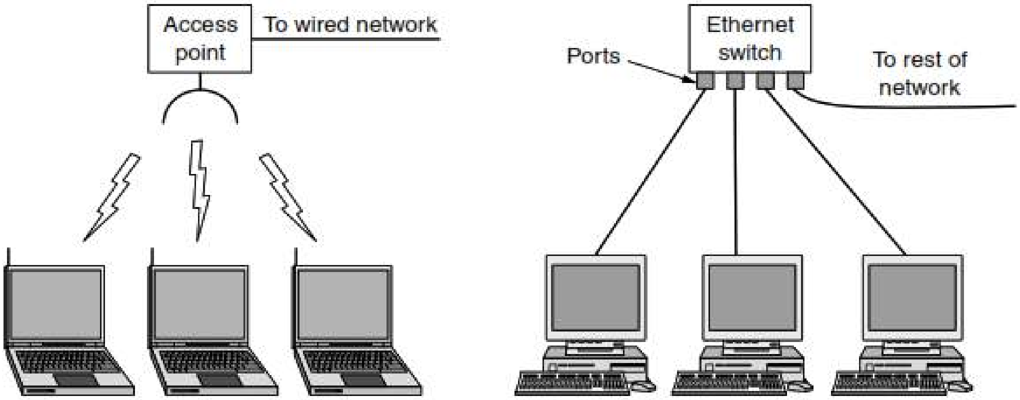


Figure 1-8. Wireless and wired LANs. (a) 802.11. (b) Switched Ethernet.

Metropolitan Area Network(MAN):

- A metropolitan area network is a network that covers a larger geographic area by interconnecting a different LAN to form a larger network.
- It generally covers towns and cities (50 km).
- In MAN, various LANs are connected to each other through a telephone exchange line.
- Communication medium used for MAN are optical fibers, cables etc.

- It has a higher range than Local Area Network(LAN).It is adequate for distributed computing applications.

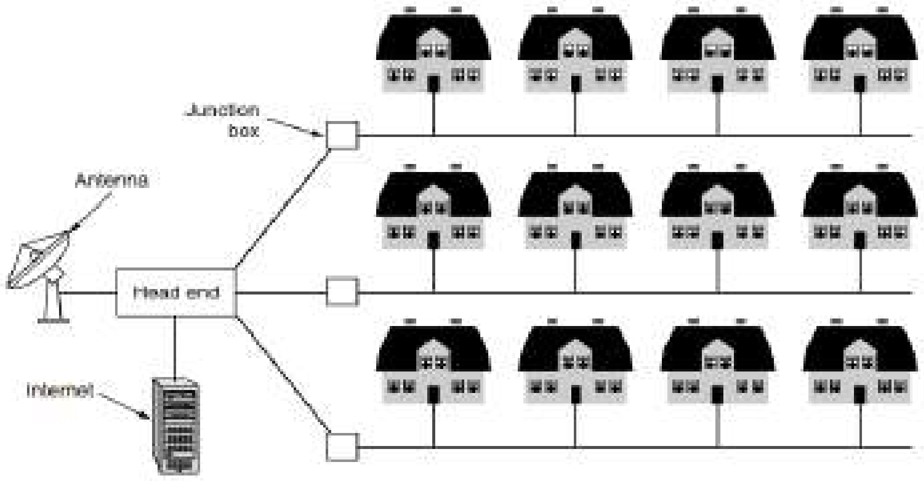
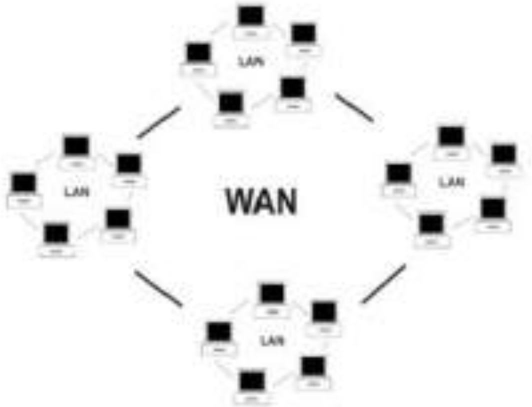


Figure 1-9. A metropolitan area network based on cable TV.

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.

1.3 NETWORK SOFTWARE

1.3.1 Protocol Hierarchies

A network protocol is a set of rules and standards which must be followed by network devices for proper communication among them.

WHY Layers? To reduce the design complexity of computer communications, hardware and software, the functionalities needed is organized as series of layers each built on its predecessor • The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented • Layer n on one machine carries on a conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer n protocol • A five-layer network is illustrated in fig

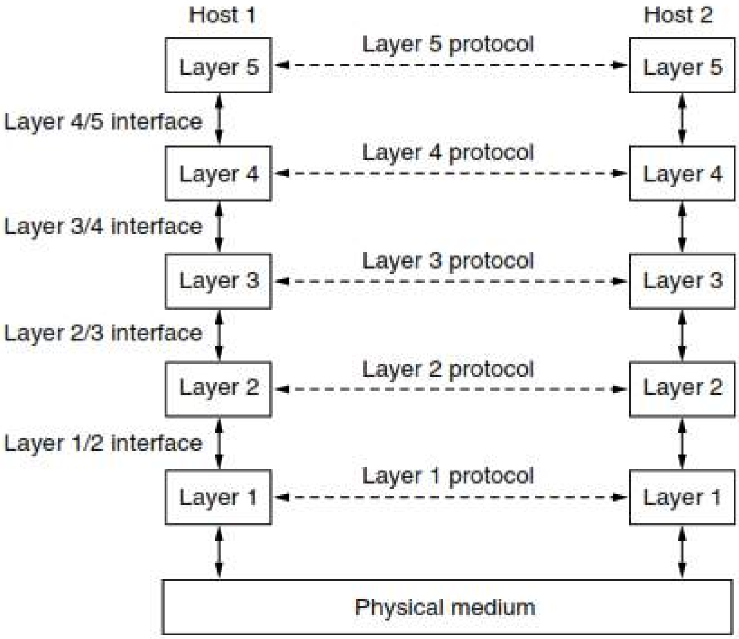


Figure 1-13. Layers, protocols, and interfaces.

In reality, no data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. In the figure 1.13, virtual communication is shown by dotted lines and physical communication by solid lines.

Interfaces and services

- Between each pair of adjacent layers is an interface. The interface defines which primitive operations and services the lower layer makes available to the upper one.
- A set of layers and protocols is called a network architecture
- A list of protocols used by a certain system, one protocol per layer, is called a protocol stack
- A message, M, is produced by an application process running in layer 5 and given to layer 4 for transmission. Layer 4 puts a header in front of the message to identify the message and passes the result to layer 3. The header includes control information, such as sequence numbers, to allow layer 4 on the destination machine to deliver messages in the right order if the lower layers do not maintain sequence.

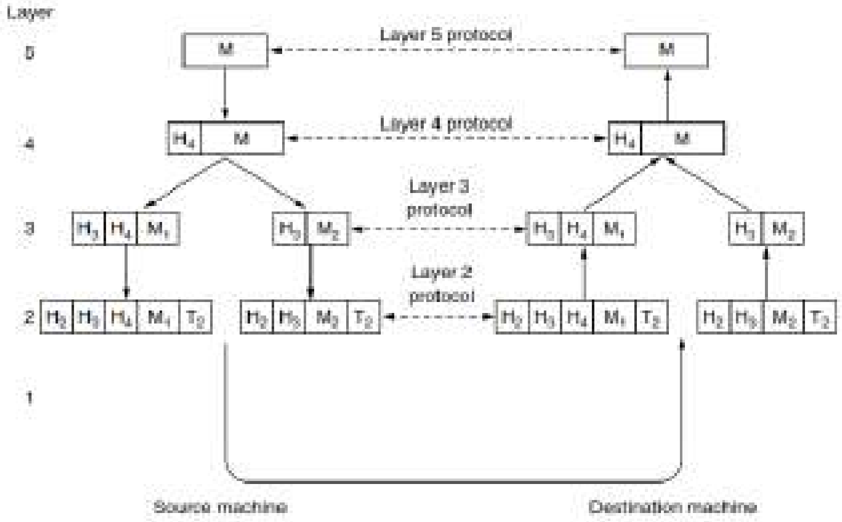


Figure 1-15. Example information flow supporting virtual communication in layer 5.

Consequently, layer 3 must break up the incoming messages into smaller units, packets, prepending a layer 3 headers to each packet. In this example, M is split into two parts, M1 and M2. Layer 3 decides which of the outgoing lines to use and passes the packets to layer 2. Layer 2 adds not only a header to each piece, but also a trailer, and gives the resulting unit to layer 1 for physical transmission. At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses.

1.3.2 Design Issues for the Layers

1. Reliability:

Think about the bits of a packet traveling through the network. There is a chance that some of these bits will be received damaged. How is it possible that we find and fix these errors?

One mechanism for finding errors in received information uses codes for **error detection**. Information that is incorrectly received can then be retransmitted until it is received correctly. More powerful codes allow for **error correction**, where the correct message is recovered from the possibly incorrect bits that were originally received. Both of these mechanisms work by adding redundant information. They are used at low layers, to protect packets sent over individual links, and high layers, to check that the right contents were received.

Another reliability issue is finding a working path through a network. Often there are multiple paths between a source and destination, and in a large network, there may be some links or routers that are broken. The network should automatically make this decision. This is called **routing.**

2. The evolution of the network:

Over time, networks grow larger and new designs emerge that need to be connected to the existing network. Since there are many computers on the network, every layer needs a mechanism for identifying the senders and receivers that are involved in a particular message. This mechanism is called **addressing or naming**, in the low and high layers, respectively.

An aspect of growth is that different network technologies often have different limitations. For example, not all communication channels preserve the order of messages sent on them, leading to solutions that number messages. Another example is differences in the maximum size of a message that the networks can transmit. This leads to mechanisms for **disassembling, transmitting, and then reassembling messages.** This overall topic is called internetworking. When networks get large, new problems arise. Cities can have traffic jams, a shortage of telephone numbers, and it is easy to get lost. Not many people have these problems in their own neighbourhood, but citywide they may be a big issue. Designs that continue to work well when the network gets large are said to be **scalable.**

3. Resource allocation.

An allocation problem that occurs at every level is how to keep a fast sender from swamping a slow receiver with data. Feedback from the receiver to the sender is often used. This is called **flow control.** Sometimes the problem is that the network is oversubscribed because too many computers want to send too much traffic, and the network cannot deliver it all. This overloading of the network is called **congestion.**

One strategy is for each computer to reduce its demand when it experiences congestion. It is interesting to observe that the network has more resources to offer than simply bandwidth. For uses such as carrying live video, the timeliness of delivery matters a great deal. Most networks must provide service to applications that want this real-time delivery at the same time that they provide service to applications that want high throughput.

Quality of service is the name given to mechanisms that reconcile these competing demands.

4. **Security:**

The last major design issue is to secure the network by defending it against different kinds of threats. One of the threats we have mentioned previously is that of eavesdropping on communications. Mechanisms that provide **confidentiality** defend against this threat, and they are used in multiple layers. Mechanisms for **authentication** prevent someone from impersonating someone else.

They might be used to tell fake banking Web sites from the real one, or to let the cellular network check that a call is really coming from your phone so that you will pay the bill. Other mechanisms for **integrity** prevent surreptitious changes to messages, such as altering ‘‘debit my account $10’’ to ‘‘debit my account $1000.’’

1.3.3 Connection-Oriented Versus Connectionless Service

Layers can offer two different types of service to the layers above them: connection-oriented and connectionless. Connection-oriented service is modelled after the telephone system. To talk to someone, you pick up the phone, dial the number, talk, and then hang up. Similarly, to use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at one end, and the receiver takes them out at the other end. In most cases the order is preserved so that the bits arrive in the order they were sent. In some cases, when a connection is established, the sender, receiver, and subnet conduct a negotiation about parameters to be used, such as maximum message size, quality of service required, and other issues. Typically, one side makes a proposal and the other side can accept it, reject it, or make a counterproposal.

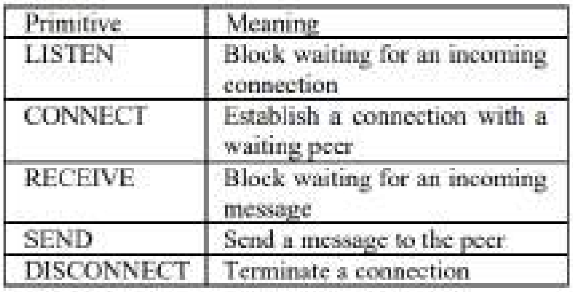
In contrast, connectionless service is modelled after the postal system. Each message (letter) carries the full destination address, and each one is routed through the system independent of all the others. Normally, when two messages are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent can be delayed so that the second one arrives first.

Each service can be characterized by a quality of service. Some services are reliable in the sense that they never lose data. Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message so the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable. A typical situation in which a reliable connection-oriented service is appropriate is file transfer. The owner of the file wants to be sure that all the bits arrive, correctly and in the same order they were sent. Very few file transfer customers would prefer a service that occasionally scrambles or loses a few bits, even if it is much faster.

1.3.4 Service Primitives

A service is formally specified by a set of primitives (operations) available to user processes to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets.

The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connectionless service.



Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Figure 1-12. Five service primitives for implementing a simple connection oriented service.

First, the server executes LISTEN to indicate that it is prepared to accept incoming connections. A common way to implement LISTEN is to make it a blocking system call. After executing the primitive. the server process is blocked until a request for connection appears.

Next, the client process executes CONNECT to establish a connection with the server. The CONNECT call needs to specify who to connect to, so it might have a parameter giving the server's address. The operating system then typically sends a packet to the peer asking it to connect, as shown by (1) in Fig. 1-13. The client process is suspended until there is a response. When the packet arrives at the server, it is processed by the operating system sees that the packet is requesting a connection, it checks to see if there is a listener.

If so, it does two things: unblocks the listener and sends back an acknowledgement (2). The arrival of this acknowledgement then releases the client. At this point the client and server are both running and they have a connection established.

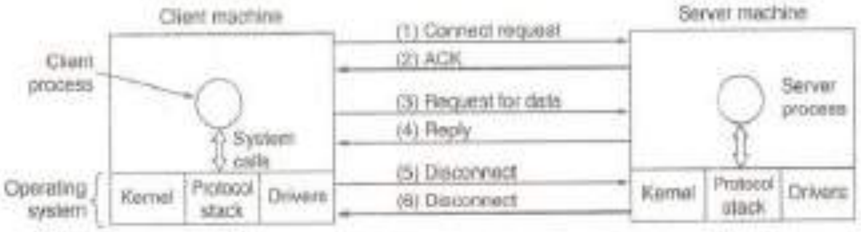


Figure 1-14. Packets sent in a simple client-server interaction on a connection oriented network.

The server process can then establish the connection with the ACCEPT call. This sends a response (2) back to the client process to accept the connection. The arrival of this response then releases the client. At this point the client and server are both running and they have a connection established. The obvious analogy between this protocol and real life is a customer (client) calling a company's customer service manager. At the start of the day, the service manager sits next to his telephone in case it rings. Later, a client places a call. When the manager picks up the phone, the connection is established.

The next step is for the server to execute RECEIVE to prepare to accept the first request. Normally, the server does this immediately upon being released from the LISTEN, before the acknowledgement can get back to the client. The RECEIVE call blocks the server.

Then the client executes SEND to transmit its request (3) followed by the execution of RECEIVE to get the reply. The arrival of the request packet at the server machine unblocks the server so it can handle the request. After it has done the work, the serve r uses SEND to return the answer to the client (4). The arrival of this packet unblocks the client, which can now inspect the answer. If the client has additional requests, it can make them now. When the client is done, it executes DISCONNECT to terminate the connection (5). Usually, an initial DISCONNECT is a blocking call, suspending the client and sending a packet to the server saying that the connection is no longer needed. When the server gets the packet, it also issues a DISCONNECT of its own, acknowledging the client and releasing the connection (6). When the server's packet gets back to the client machine, the client process is released and the connection is broken. In a nutshell, this is how connection-oriented communication works.

1.3.5. The Relationship of Services to Protocols

A service is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

A protocol, in contrast, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols to implement their service definitions. They are free to change their protocols at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled. This is a key concept that any network designer should understand well. To repeat this crucial point, services relate to the interfaces between layers.

In contrast, protocols relate to the packets sent between peer entities on different machines. It is very important not to confuse the two concepts.

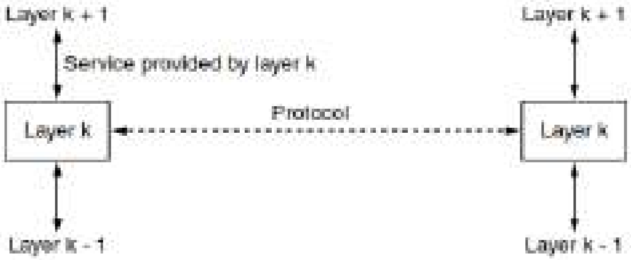


Figure 1-19. The relationship between a service and a protocol.

1.4. REFERENCE MODELS

Reference models give a conceptual framework that standardizes communication between heterogeneous networks. We will discuss two important network architectures: **the OSI reference model** and the **TCP/IP reference model**. Although the protocols associated with the OSI model are not used any more, the model itself is actually quite general and still valid, and the features discussed at each layer are still very important.

The TCP/IP model has the opposite properties: the model itself is not of much use but the protocols are widely used. For this reason we will look at both of them in detail.

1.4.1. The OSI Model

International Standards Organization (ISO) made a first step towards 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. Each layer has specific functions. All layers work together in the correct order to move data around a network.

The open systems interconnection (OSI) model refers to a standard model used to describe the flow of information from one computing device to another operating in a networking environment. The model defines a set of rules and requirements for data communication and interoperability between different devices, products, and software in a network infrastructure. The OSI model is split into seven fundamental layers (bottom to top): Physical, Data Link, Network, Transport, Session, Presentation, and Application**.**

In 1984, the International Organization for Standardization (ISO) published the OSI framework to standardize network design and equipment manufacturing principles. Until OSI emerged, network architecture lacked the standard protocols necessary for effective data communication and design infrastructure.

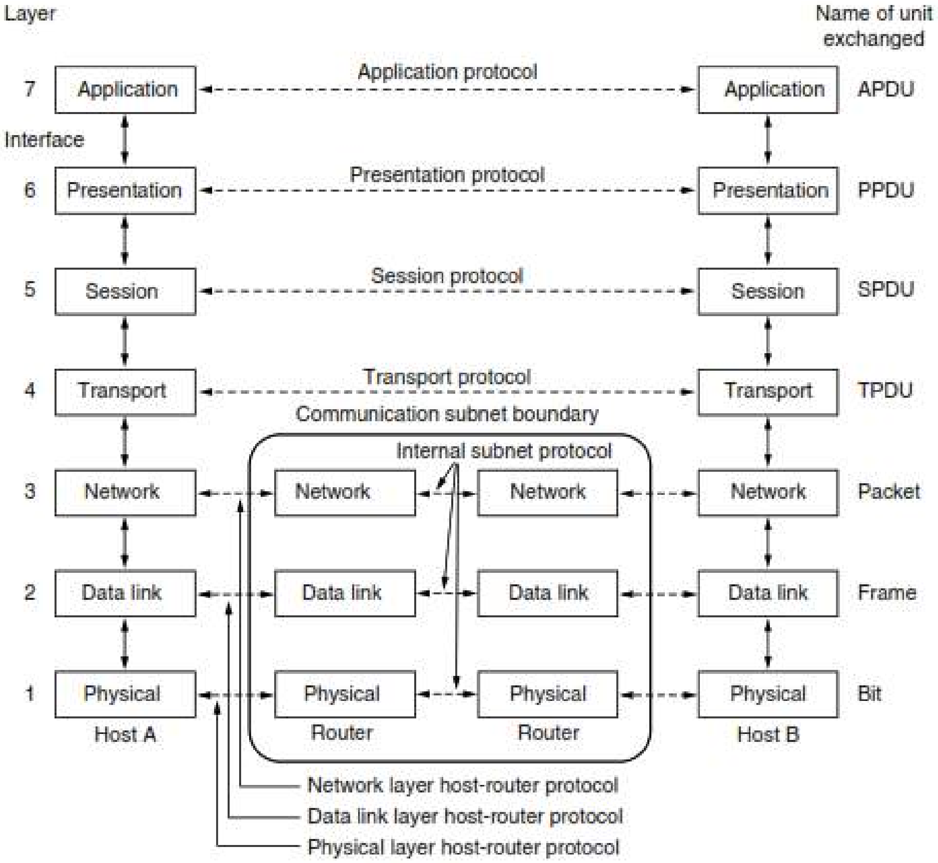


Figure 1-20. The OSI reference model.

As such, network administrators found installing, configuring, and setting up new equipment in existing networks challenging. Moreover, integrating such devices with outside networks seemed even more complex. With the arrival of the OSI reference model, administrators were able to design efficient network infrastructure where the equipment was capable of communicating with other universal networks.

Why does the OSI model matter?

Establishing standard communication protocols is key to formalizing communication across internal and external networks such as the cloud or the internet. Such standardization facilitates faster communication between devices and networks, irrespective of the data resides, where it is sent, or from where it is received.

The OSI model allows equipment manufacturers to define their standards and protocols while maintaining interconnectivity with other manufacturers. Moreover, the OSI model is key to troubleshooting network devices. When a networking unit fails, or an application goes down and cannot communicate with the rest of the network, the OSI model is handy as it allows administrators to zero in on the specific OSI layer and troubleshoot the failed component. Thus, the conceptual OSI framework is crucial for designing, manufacturing, and troubleshooting network technology.

How does the OSI model facilitate data communication?

Let’s take an example of an email application to understand better how data flows through the OSI model. You must note that the data flows from layer 7 to layer 1 at the sender side, while from layer 1 to layer 7 at the recipient’s side.

When an individual sends an email, it is sent to the presentation layer (layer 6) with the help of a standard outgoing protocol (SMTP protocol). At layer 6, the email message is compressed and forwarded to the session layer (layer 5). The session layer then establishes communication between the sender and the outgoing server, thereby starting the communication session.

The session layer then sends the message to the transport layer (layer 4), which performs data segmentation. The segmented message is further sent to the network layer (layer 3), which breaks down the segments into data packets. These data packets are forwarded to the data link layer (layer 2) where data packets are further divided into frames. These frames are then sent to the physical layer (layer 1) where the received data is converted into zeros and ones. These bit streams are then sent through the network via cables or network connections.

As the email message reaches the recipient’s device, the above process is reversed, implying that data flow starts at the physical layer and ends at the application layer. Thus, the above bit streams of zeros and ones are converted into the original email message that is finally available on the recipient’s email. When the recipient responds to this mail thread, the process is repeated with the data flow starting from layer 7 to layer 1. In this way, the OSI model facilitates communication between network components. Let’s understand each layer in greater detail. We start with the uppermost layer 7 and move to layer 1.

Application layer

The application layer is the topmost layer in the OSI model. The layer establishes communication between the application on the network and the end user using it by defining the protocols for successful user interaction. An excellent example of this layer is that of web browsers.

Application layer protocols allow the software to direct data flow and present it to the user. Some of the known protocols include Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP).

Key functions:

- The application layer provides user interfaces (UI) that are key to user interaction
- Supports a variety of applications such as e-mail and remote file transfer

In summary, layer 7 ensures effective communication between applications on different computing systems and networks.

Presentation layer

The presentation layer is often referred to as a syntax or translation layer as it translates the application data into a network format. This layer also encrypts and decrypts data before transmitting it over the network. For instance, layer 6 encrypts data from the application and decrypts it at the recipient’s end, ensuring secure data transmission. Moreover, this layer is known to compress data received from layer 7 to reduce the overall size of the data transferred.

Key functions:

- Performs data **translation** based on the application’s data semantics
- **Encrypts** and decrypts sensitive data transferred over communication channels
- Performs data **compression** to reduce the number of bits in exchanged data

In summary, layer 6 ensures that the communicated information is in the desired format as required by the receiving application.

Session layer

The session layer establishes a communication session between communicating entities. The session is maintained at a sufficient time interval to ensure efficient data transmission and avoid wasting computing resources.

This OSI layer is also responsible for data synchronization to maintain smooth data flow. This implies that in situations where large volumes of data are sent at once, layer 5 can break down the data into smaller chunks by adding checkpoints.

For example, let's say you want to send a 500-page document to another person. In this case, this layer can add checkpoints at 50 or 100 pages. This is in case a document transfer is interrupted due to network or system failure. Once the system failure issue is resolved, the document transfer resumes from the last checkpoint. Such a system saves time by not restarting the file transfer from the beginning.

Key functions:

- Opens maintains, and closes communication sessions
- Enables **data synchronization** by adding checkpoints to data streams

In summary, layer 5 **establishes, maintains, synchronizes, and terminates sessions** between end-user applications.

Transport layer

The transport layer allows safe message transfer between the sender and the receiver. It divides the data received from the layer above into smaller segments. It also reassembles the data at the receiver side to allow the session layer to read it.

Layer 4 performs two critical functions: flow control and error control. Flow control implies regulating data transfer speeds. It ensures that the communicating device with a good network connection does not send data at higher rates, which is difficult for devices with slower connections to handle. Error control refers to the error-checking functionality to ensure the completeness of data. In incomplete data cases, this layer requests the system to resend the incomplete data.

Examples of transport layer protocols include transmission control protocol (TCP) and user datagram protocol (UDP).

Key functions:

- The transport layer enables communication between two applications running on different computers.
- Maintains proper data transmission through **flow control and error control**
- Performs data **segmentation and reassembling** of data

In summary, layer 4 is responsible for transmitting an entire message from a sender application to a receiver application.

Network layer

The network layer enables the communication between multiple networks. It receives data segments from the layer above, further broken down into smaller packets at the sender side. On the receiver side, this layer reassembles the data together.

The network layer also handles routing functionality, wherein the data transmission is accomplished by choosing the best possible route or path that connects different networks and ensures efficient data transfer. This network layer uses **internet protocol (IP)** for data delivery.

Key functions:

- Handles **routing** to recognize suitable routes from sender to receiver
- Performs **logical addressing** that assigns unique names to each device operating over the network

In summary, layer 3 is responsible for dividing segmented data into network packets, reassembling them at the recipient's side, and identifying the shortest yet most suitable and secure path for transmitting data packets.

Data link layer

The data link layer transmits data between two nodes that are directly connected or are operating over the same network architecture. Typically, this layer takes data packets from layer 3 and breaks them down into frames before sending them to the destination.

Layer 2 is divided into two sub-layers: media access control (MAC) and logical link control (LLC). The MAC layer encapsulates data frames transmitted through the network connecting media such as wires or cables. In situations where such data transmission fails, LLC helps manage packet retransmission.

The well-known data link layer protocol includes the Address Resolution Protocol (ARP) that translates IP addresses to MAC addresses to establish communication between systems whose addresses vary in bit length (32 bits vs. 48 bits).

Key functions:

- Detects damaged or lost frames and retransmits them
- Performs **framing** where data received from layer 3 is further subdivided into smaller units called frames
- Updates headers of created frames by adding the MAC address of the sending device and receiving device

In summary, layer 2 is responsible for setting up and terminating physical connections between participating network nodes.

Physical layer

The last OSI layer is the physical layer that manages physical hardware and network components such as cables, switches, or routers that transmit data.

In the context of data, layer 1 transmits data in the form of ones and zeros. Technically, this layer picks up bits from the sender end, encodes them into a signal, sends the signal over the network, and decodes the signal at the receiver end. Thus, without layer 1, communicating data bits across network devices through physical media is not possible.

Key functions:

- Synchronizes data bits
- Enables modulation (conversion of a signal from one form to another for data transmission)
- Defines data transmission rate (bits/sec)
- Outlines the arrangement of network devices across different network topologies such as bus, tree, star, or mesh topology
- Defines transmission modes such as simple or half-duplex mode

In summary, layer 1 is responsible for transmitting data bits of 0s and 1s between network systems via electrical, mechanical, or procedural interfaces.

What is the TCP/IP Model?

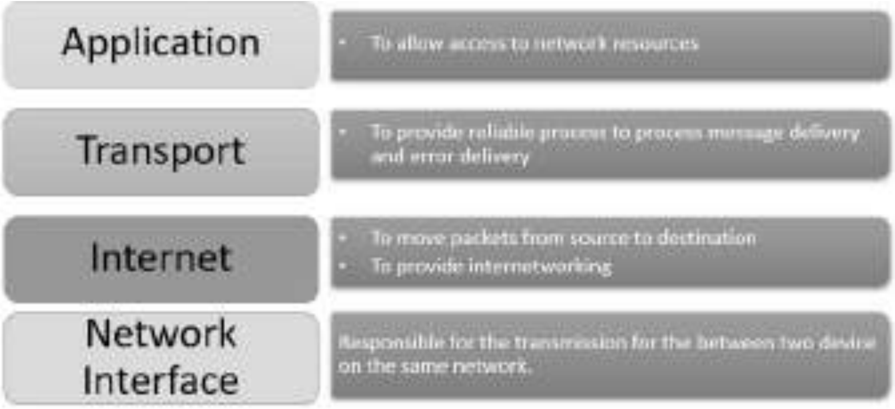
TCP/IP stands for Transmission Control Protocol/ Internet Protocol. This model helps you to determine how a specific computer should be connected to the internet and how data should be transmitted between them. It helps you to create a virtual network when multiple computer networks are connected together. The purpose of TCP/IP model is to allow communication over large distances.

1.4.2. TCP/IP Conceptual Layers

The functionality of the TCP IP model is divided into four layers, and each includes specific protocols.

TCP/IP is a layered server architecture system in which each layer is defined according to a specific function to perform. All these four TCP IP layers work collaboratively to transmit the data from one layer to another.

- Application Layer
- Transport Layer
- Internet Layer
- Network Interface



Application
- To allow access to network resources

Transport
- To provide reliable process to process message delivery and error delivery

Internet
- To move packets from source to destination
- To provide internetworking

Network Interface
Responsible for the transmission for the between two devine on the same network.

Application Layer

Application layer interacts with an application program, which is the highest level of OSI model. The application layer is the OSI layer, which is closest to the end-user. It means the OSI application layer allows users to interact with other software application.

Application layer interacts with software applications to implement a communicating component. The interpretation of data by the application program is always outside the scope of the OSI model. Example of the application layer is an application such as file transfer, email, remote login, etc.

Function of the Application Layers:

- Application-layer helps you to identify communication partners, determining resource availability, and synchronizing communication.
- It allows users to log on to a remote host
- This layer provides various e-mail services
- This application offers distributed database sources and access for global information about various objects and services.

Transport Layer

Transport layer builds on the network layer in order to provide data transport from a process on a source system machine to a process on a destination system. It is hosted using single or multiple networks, and also maintains the quality of service functions.

It determines how much data should be sent where and at what rate. This layer builds on the message which are received from the application layer. It helps ensure that data units are delivered error-free and in sequence.

Transport layer helps you to control the reliability of a link through flow control, error control, and segmentation or de-segmentation.

The transport layer also offers an acknowledgment of the successful data transmission and sends the next data in case no errors occurred. TCP is the best-known example of the transport layer.

Important functions of Transport Layers

- It divides the message received from the session layer into segments and numbers them to make a sequence.
- Transport layer makes sure that the message is delivered to the correct process on the destination machine.
- It also makes sure that the entire message arrives without any error else it should be retransmitted.

Internet Layer

An internet layer is a second layer of TCP/IP layes of the TCP/IP model. It is also known as a network layer. The main work of this layer is to send the packets from any network, and any computer still they reach the destination irrespective of the route they take.

The Internet layer offers the functional and procedural method for transferring variable length data sequences from one node to another with the help of various networks.

Message delivery at the network layer does not give any guaranteed to be reliable network layer protocol.

Layer-management protocols that belong to the network layer are:

1. Routing protocols
2. Multicast group management
3. Network-layer address assignment.

The Network Interface Layer

Network Interface Layer is this layer of the four-layer TCP/IP model. This layer is also called a network access layer. It helps you to defines details of how data should be sent using the network.

It also includes how bits should optically be signaled by hardware devices which directly interfaces with a network medium, like coaxial, optical, coaxial, fiber, or twisted-pair cables.

A network layer is a combination of the data line and defined in the article of OSI reference model. This layer defines how the data should be sent physically through the network. This layer is responsible for the transmission of the data between two devices on the same network.

Advantages of the TCP/IP model

Here, are pros/benefits of using the TCP/IP model:

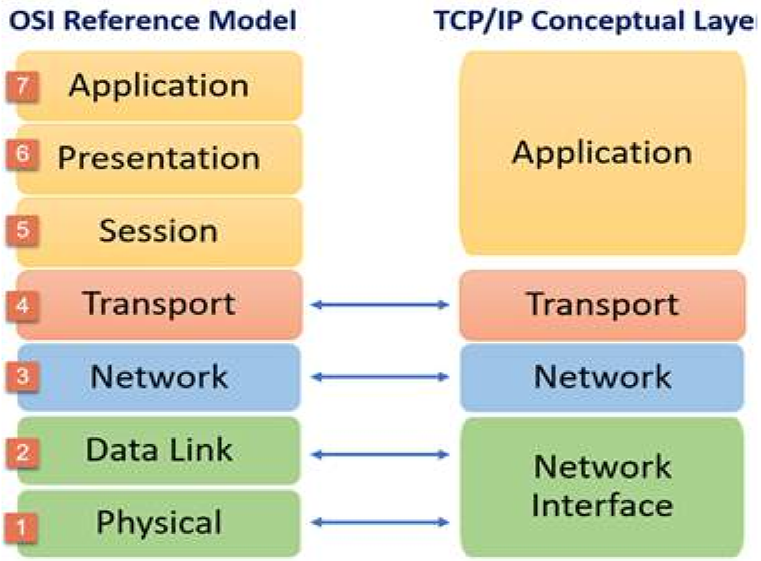
- It helps you to establish/set up a connection between different types of computers.
- It operates independently of the operating system.
- It supports many routing-protocols.
- It enables the internetworking between the organizations.
- TCP/IP model has a highly scalable client-server architecture.
- It can be operated independently.
- Supports a number of routing protocols.
- It can be used to establish a connection between two computers.

Disadvantages of the TCP/IP model

Here, are few drawbacks of using the TCP/IP model:

- TCP/IP is a complicated model to set up and manage.
- The shallow/overhead of TCP/IP is higher-than IPX (Internetwork Packet Exchange).
- In this, model the transport layer does not guarantee delivery of packets.
- Replacing protocol in TCP/IP is not easy.
- It has no clear separation from its services, interfaces, and protocols.

Differences between OSI and TCP/IP models



Difference between OSI and TCP/IP model

Here, are some important differences between the OSI and TCP/IP model:

OSI Model	**TCP/IP model**
It is developed by ISO (International Standard Organization)	It is developed by ARPANET (Advanced Research Project Agency Network).
OSI model provides a clear distinction between interfaces, services, and protocols.	TCP/IP doesn’t have any clear distinguishing points between services, interfaces, and protocols.
OSI refers to Open Systems Interconnection.	TCP refers to Transmission Control Protocol.
OSI uses the network layer to define routing standards and protocols.	TCP/IP uses only the Internet layer.
OSI follows a vertical approach.	TCP/IP follows a horizontal approach.
OSI model use two separate layers physical and data link to define the functionality of the bottom layers.	TCP/IP uses only one layer (link).

OSI Model	**TCP/IP model**
OSI layers have seven layers.	TCP/IP has four layers.
OSI model, the transport layer is only connection-oriented.	A layer of the TCP/IP model is both connection-oriented and connectionless.
In the OSI model, the data link layer and physical are separate layers.	In TCP, physical and data link are both combined as a single host-to-network layer.
Session and presentation layers are not a part of the TCP model.	There is no session and presentation layer in TCP model.
It is defined after the advent of the Internet.	It is defined before the advent of the internet.
The minimum size of the OSI header is 5 bytes.	Minimum header size is 20 bytes.

1.5. TRANSMISSION MEDIA

Transmission media are the physical infrastructure components that carry data from one computer to another. They are at the basis of data communications. Examples of simple forms of transmission media are telephone wires that connect telephones to the central offices (i.e. telephone exchanges), and coaxial cables that carry the cable television transmission to homes. Transmission media need not always be in the form of a physical wire-they can be invisible as well We shall study these different types of transmission media in this chapter.

Broadly, all transmission media can be divided into two main categories: Guided and Unguided, as shown in Fig. 2.6.

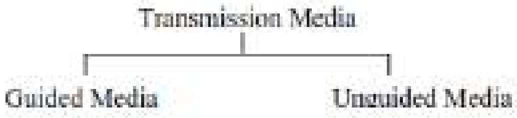


Fig2.6 Categories of Transmission Media

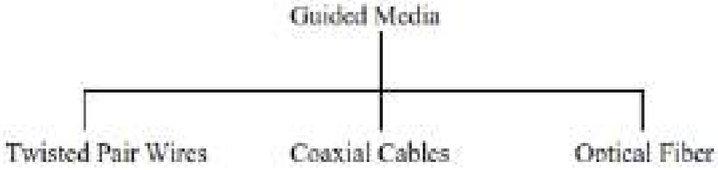


Fig 2.7 Types of Guided Media

1.5.1. GUIDED MEDIA

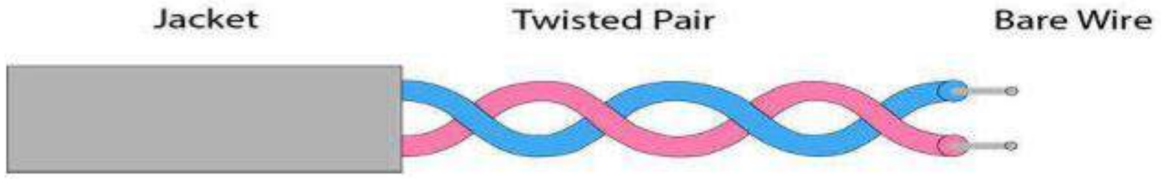
Guided Media can be further subdivided into three main types as shown in Fig.2.7 Guided media are typically based on some physical cable. Twisted pair and coaxial cables carry signals in the form of electrical current, whereas optical fiber carries signals in the form of light.

Twisted Pair Cable

There are two classes of twisted pair cables

a. Unshielded Twisted Pair (UTP)

This is the most commonly used medium today, because of its usage in the telephone system. This cable can carry both voice as well as data. It consists of two conductors (usually copper). In the beginning, the wires used to be kept parallel. However, this results in far greater levels of noise. Hence, the wires are normally twisted as shown Fig. 2.9. This results in the reduction of noise to a great extent, although it is not eliminated completely. The copper conductors are covered by the below figure. PVC. or other insulator.



UTP is flexible, cheap and easy to install. The Electronic Industries Association (EIA) has developed standards for UTP cables. They are given in Fig. 2.10. Each one is manufactured differently for a specific purpose.

Ethernet Cable Categories

There are many Ethernet cable options available, and each one of them has its unique purpose and use. Therefore, it is important to understand each cable and its application if you want to acquire in-depth knowledge about the types of Ethernet cables. You need to choose the higher quality cable, which will be stronger, faster, and a better fit for your specific needs. However, depending on your hardware, you can select your below-given ethernet cabling category.

Category-3

Cat3 cable is an earlier generation of cable, which supports a maximum frequency of 16 MHz. This cable may have 2, 3, or 4 copper pairs. Cat3 type of Ethernet cable is still used for two-line telephone systems and 10BASE-T networks. It is also used for alarm system installation or similar kinds of applications.

Category-5

These cables are slower compared to modern-day hardware requirements. So, you should use this type of Cable only if you have older hardware that demands outdated hardware.

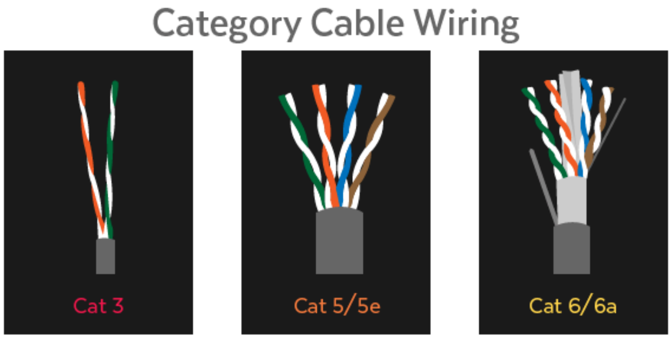
Category-5e

Cat5e is one of the most popular cabling types of an ethernet cable used for deployments because of its ability to support Gigabit speeds at a cost-effective price.

Cat 5e can support up to 1000 Mbps speeds, which is flexible enough for small space installations. Therefore, it is widely used in residential areas. Cat5e is one of the least expensive cabling options available in the market.

Category-6

Cat6 cabling support up to 10 Gbps and frequencies of up to 250 MHz. These types of cables are more tightly twisted and feature two or more twists per centimeter. It only supports 37-55 meters when transmitting 10 Gbps speeds.



Category-6a

Cat6a ethernet cable supports bandwidth frequencies of up to 500 MHz. Cat6a cabling is thicker compared to Cat6, making it less flexible. That is why it is more suited for industrial environments at a lower price point.

Today, UTPs of higher categories are also used in computer networks given the higher speeds and reliability.

b. Shielded Twisted Pair (STP)

In this case, apart from the insulator, the twisted wire pair itself is covered by metal shield and finally by the plastic cover as shown in Fig. 2.11. The metal shield prevents penetration of electromagnetic noise. It also helps eliminate crosstalk, an effect wherein one wire picks up some of the signals travelling on the other wire. This effect can be felt sometimes during telephone conversations, when we hear other conversations in the background during our call. The shield prevents such unwanted sounds.

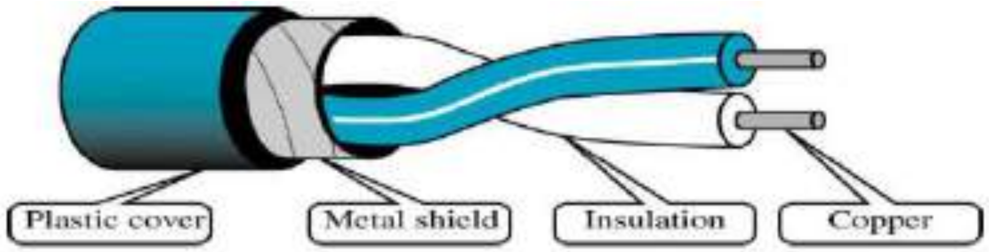


Fig 2.11. Shielded Twisted Pair (STP)

As we have seen, the shield reduces the noise and cross talk (noticed in the telephone calls sometimes1'Substantially. However, STP is more expensive than UTP.

Coaxial Cable

Coaxial cable (also called as coax) has an inner central conductor surrounded by an insulating sheath, which in turn is enclosed in the outer conductor (shield). This acts not only as a second conductor for completing the circuit but also acts as a shield against noise. This outer conductor is covered by a plastic cover. This whole arrangement is shown in Fig. 2.12

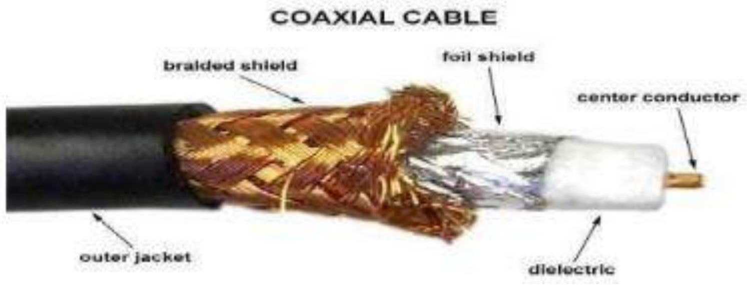


Fig. 2.12 Coaxial Cable

Compared to UTP or STP, coaxial cable is more expensive, less flexible and more difficult to install in a building where a number of twists and turns are required. However, it is much more reliable and can carry far higher data rates.

Coaxial cables are divided into various categories depending upon the thickness and size of the shields, insulators and the outer coating and other considerations. They are commonly used by cable companies to carry cable transmissions.

Optical Fiber

Optical fibers use light instead of electrical signals as a means of signal propagation. They are made of glass fibers that are enclosed in a plastic jacket. This allows the fibers to bend and not break. A transmitter at the sender's end of the optical fiber sends a light emitting diode (LED) or laser to send pulses of light across the fiber. A receiver at the other end makes use of a light-sensitive transistor to detect the absence or presence of light to indicate a 0 or a 1.

In the fiber, the cladding covers the core (fiber) depending upon the size. Commonly, the fiber is covered by a buffer layer, which protects it from moisture. Finally, an outer jacket surrounds the entire cable. This is shown in Fig. 2.13

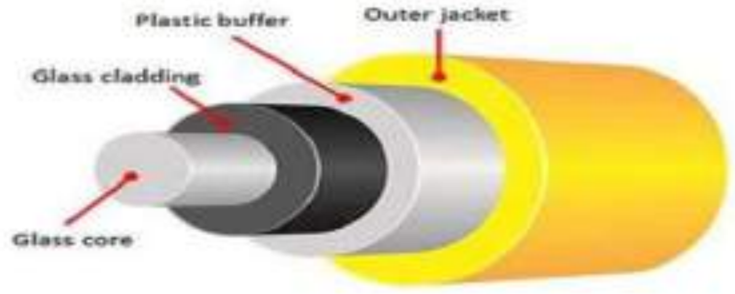


Fig 2.13. Optical Fiber

The core and cladding must be extremely pure, and should not vary in size or shape. The outer jacket, however, can be made of Teflon, plastic or metal. Teflon is more expensive than plastic, but both do not provide the structural strength to the fiber. Metal can provide that, but is very heavy and expensive. The choice depends upon many of these factors.

Advantages of Optical Fiber

Optical fiber provides the following advantages over twisted-pair and coaxial cable:

- **Resistance to noise** Optical fiber uses light rays for communication, rather than electricity. Therefore, noise is not an issue here. The only possible source of interference is external light, which is also blocked by the outer jacket of the optical fiber, thus providing resistance to interference.

- **Huge bandwidth** The bandwidth offered by optical fiber is huge as compared to twisted-pair and coaxial cable.
- **Higher signal** carrying capacity Unlike twisted-pair and coaxial cable, the signals carried by optical fiber travel longer distances (several kilometers) without needing regeneration.

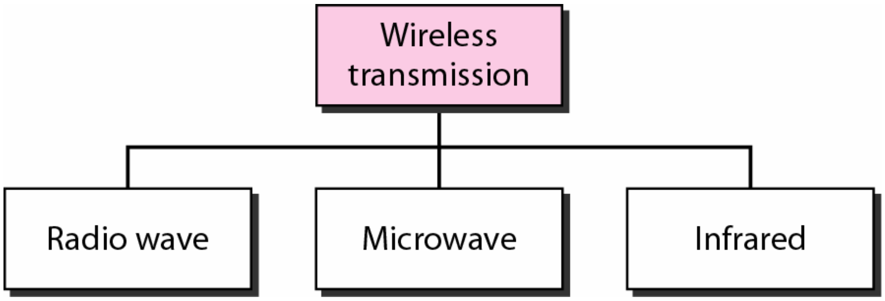
Disadvantages of Optical Fiber

Along with all the advantages that it offers, optical fiber also has some disadvantages, as summarized

- Fragility Optical fiber is more delicate than twisted-pair and coaxial cable, so the chances of it breaking are higher.
- Cost Optical fiber is quite expensive because it has to be extremely pure to be able to carry signals without damage over longer distances. The laser light source itself is also very costly, as compared to the electrical signal generators.
- Maintenance overhead Optical fibers need more maintenance overheads as compared to twisted-pair and coaxial cable.

1.5.2 UNGUIDED MEDIA (Wireless Communication)

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



Unguided media, also called as wireless communication, which transport electromagnetic waves without using a physical conductor. The signals propagate through air (or sometimes water). The communication band for unguided media is as shown in Fig. 2.18. We shall concentrate on the radio communications.

Radio Communication

Radio waves are easy to generate, can travel long distances, and can penetrate buildings easily, so they are widely used for communication, both indoors and outdoors. Radio waves also are omnidirectional, meaning that they travel in all directions from the source, so the transmitter and receiver do not have to be carefully aligned physically.

Sometimes omnidirectional radio is good, but sometimes it is bad.

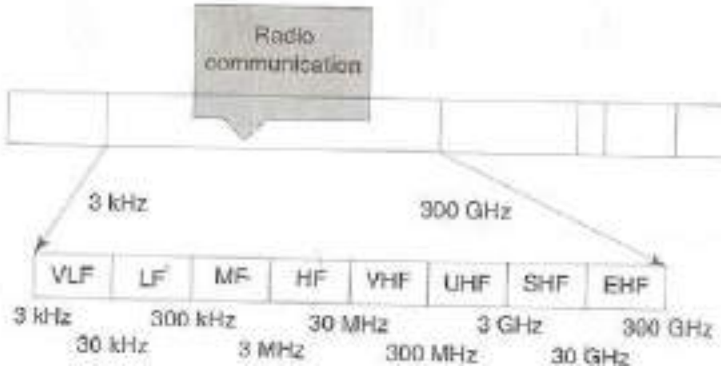
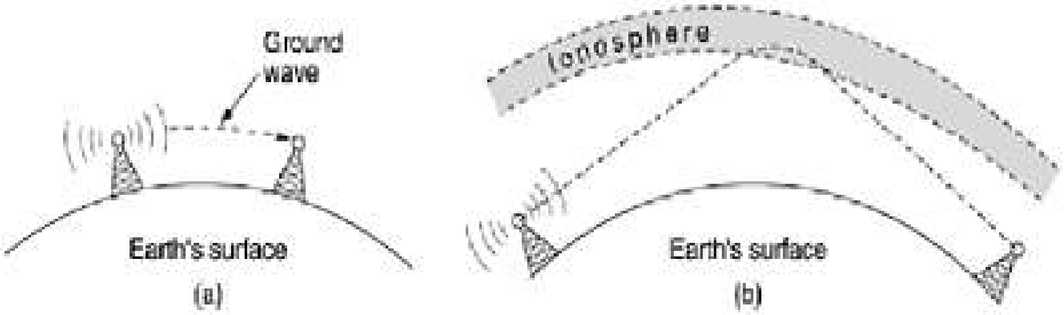


Fig 2.18. Radio Communications Band

In the VLF, LF, and MF bands, radio waves follow the ground, These waves can be detected for perhaps 1000 km at the lower frequencies, less at the higher ones. AM radio broadcasting uses the MF band, which is why the ground waves from Boston AM radio stations cannot be heard easily in New York. Radio waves in these bands pass through buildings easily, which is why portable radios work indoors. The main problem with using these bands for data communication is their low bandwidth. In the HF and VHF bands, the ground waves tend to be absorbed by the earth. However, the waves that reach the ionosphere, a layer of charged particles circling the earth at a height of 100 to 500 km, are refracted by it and sent back to earth). Under certain atmospheric conditions, the signals can bounce several times. Amateur radio operators (hams) use these bands to talk long distance. The military also communicate in the HF and VHF bands.



(b) Microwave Communication

As shown in Fig.2.19, microwaves use the line of sight method of propagation. as the signals do not travel along the surface of the earth. Therefore, the two antennas must be in a straight line, able to look at each other without any obstacle in between. The taller the antennas, the more is the distance that these waves travel. Usually, the antennas are positioned on mountain tops to avoid obstacles. Microwave signals travel only in one direction at a time. This means that for two-wave communication such as in telephony, two frequencies need to be allocated. At both ends, a transceiver is used which is a combination of a transmitter and a receiver operating at the two respective frequencies. Therefore, only one antenna can serve both the functions and cover both the frequencies. Repeaters are used along with. The antennas to enhance the signa1. The data rates offered are 1 Mbps -10 Gbps.

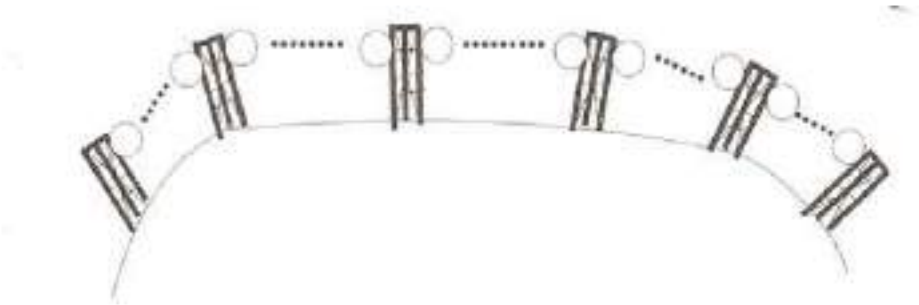


Fig.2.19 Terrestrial Microwave Communication

(c) Satellite Communication

Satellite communications is similar to the terrestrial microwave, except that the satellite acts as one of the stations. Figure 2.20 illustrates this. The satellite does the functions of an antenna and the repeater together. For instance, as Fig. 2.20 illustrates, ground station A can send the information to ground station B via the satellite.

This, however, poses a problem. If the earth along with its ground stations is revolving and the satellite is stationery, the sending and receiving earth stations and the satellite can be out of sync over time. Therefore, normally Geosynchronous satellites are used, which move at the same Revolutions Per Minute (RPM) as that of the earth in the same direction, exactly like the earth. Thus, both the earth and the satellite complete one

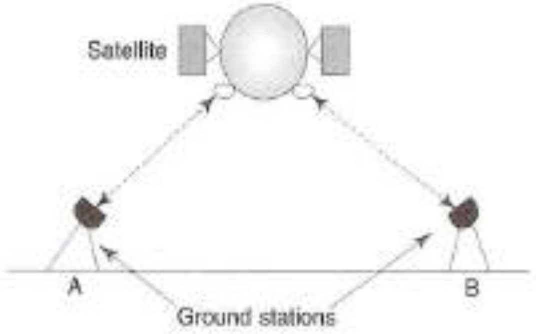


Fig.2.20 Satellite Microwave Communication

revolution exactly in the same time: the relative position of the ground station with respect to the-satellite never changes. Therefore, the movement of the earth does not matter for the communicating nodes based on the earth. Using satellites, we can communicate from any part of the world to any other. However, a minimum of three satellite needed to cover the earth's surface entirely as shown in the Fig. 2.21.

Normally, SHF, which covers the frequency range of 3 GHz to 30 GHz used for satellite communications. Two frequency bands are used for the signals from the earth to the satellite (called uplink), and from the satellite to the earth (called downlink).

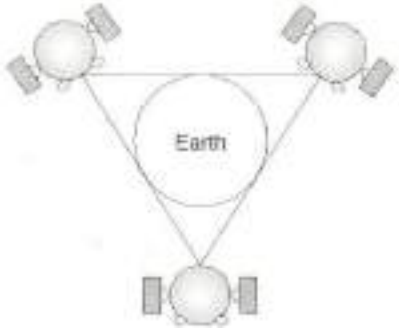


Fig.2.21 Three Satellites to Cover the planet

1.5.3. Multiplexing Methods

There are three methods for communication using satellites. These three methods use principles that are similar in concept to normal wired communication. Like the wired world, satellite communication is also based on modulation techniques. The three primary modulation techniques are: (a) Frequency Division Multiple Access (FDMA), (b) Time Division Multiple Access (TDMA) and (c) Code Division Multiple Access (CDMA).

These technologies can be explained at a broad level by breaking down the title of each one.

(a) The first word tells us what the access method is and the second word, division, lets you know that it splits transmissions based on that access method.

- FDMA puts each transmission on a separate frequency

- TDMA assigns each transmission a certain portion of time on a designated frequency
- CDMA gives a unique code to each transmission and spreads it over the available set of frequencies.

(b) The last words of each category are multiple access. This simply means that more than one user (multiple) can use (access) each cell. We shall now discuss these technologies in brief.

Frequency Division Multiple Access (FDMA)

FDMA splits the total bandwidth into multiple channels. Each ground station on the earth is allocated a particular frequency group (or a range of frequencies). Within each group, the ground station can allocate different frequencies to individual channels, which are used by different stations connected to that ground station. Before the transmission begins, the transmitting ground station looks for an empty channel within the frequency range that is allocated to it and once it finds an empty channel, it allocates it to the particular transmitting station.

This is the most popular method for communication using satellites. The transmission station on earth combines multiple signals to be sent into a single carrier signal of a unique frequency by multiplexing them, similar to the way a TV transmission works. The satellite receives this single multiplexed signal. The satellite, in turn, broadcasts the received signal to the receiving earth station. The receiving station is also supposed to agree to this carrier frequency, so that it can demultiplex the received signals. The basic idea of FDMA is shown in Fig. 2.25

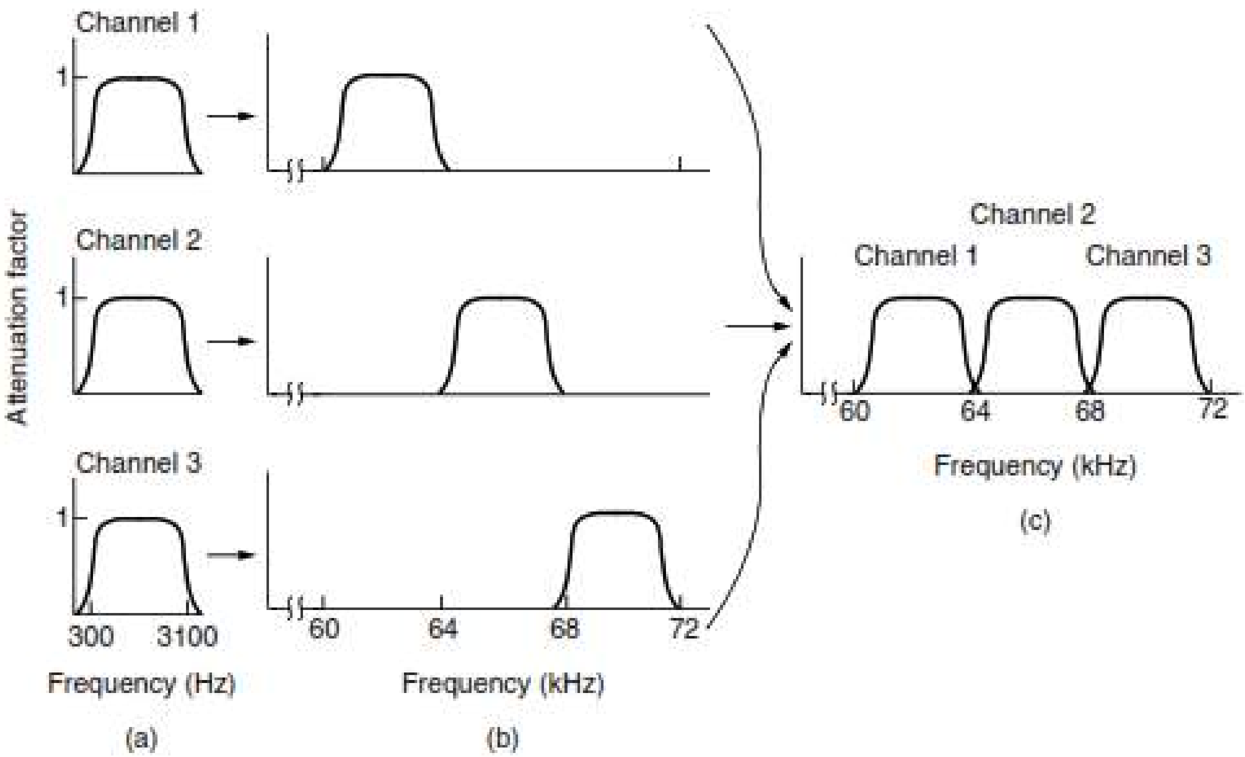


Figure 2-25. Frequency division multiplexing. (a) The original bandwidths. (b) The bandwidths raised in frequency. (c) The multiplexed channel.

Time Division Multiplexing (TDMA)

Unlike FDMA, TDMA allows access to the full bandwidth of the frequency spectrum. In TDMA, each transmitter is allocated a predefined time slot. Each transmitter receives the time slot in turn and it is allowed to transmit data for the duration of the time slot. This is shown in Fig. 2.27.

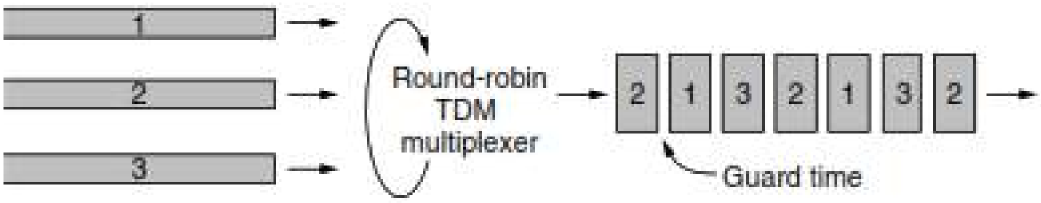


Figure 2-27. Time Division Multiplexing (TDM).

After FDMA, TDMA is the second most popular mechanism for communication using satellites. In case of TDMA, there is no modulation of frequencies (unlike FDMA). Instead, the transmitting earth station transmits data in the form of packets of data. These data packets arrive at the satellite one by one (hence the name TDMA). By the same logic that was discussed during TDM, TDMA is also a digital form of data transmission; TDMA operates in time domain, rather than frequency domain. Bit rates of 10-100 Mbps are common for TDMA transmissions. This can be translated into roughly 1800 simultaneous voice calls using 64 Kbps PCM.

Code Division Multiple Access (CDMA)

As we have seen, in FDMA, a fixed frequency channel is allocated to a transmission pair (i.e. the transmitter and the receiver). In the case of TDMA, transmission happens by using predefined time slots. Nothing of this sort happens in the case of CDMA. CDMA allows any transmitter to transmit in any frequency, and at any time. Thus, it is different from both FDMA and TDMA.

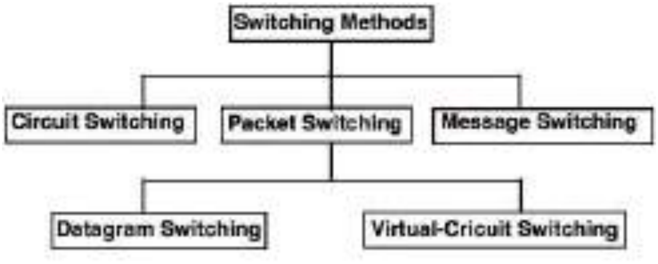
An obvious question is: Would this not lead to confusion and mixing of different signals? It would. However, to counter this, the transmitter and the receiver agree upon a unique coding scheme (similar to encryption) before the start of the transmission. The analogy often used to explain CDMA is an international party, where there are dozens of people. All are talking at once, and all talking in different languages that you do not understand. Suddenly, from across the room, you hear a voice speaking in your own familiar language, and your brain tunes out all the background gibberish and locks onto to that one person. Your brain understands the code being used by the other person, and vice versa.

Similarly, CDMA uses coding that is unique to a particular transmission and allows the receiver to disregard other transmissions on the same frequency. In this case, the coding scheme is a unique frequency with which the original signal is modulated to form the codified signal that is to be transmitted. The transmitter then codifies all its transmissions in this manner, before they are sent. Since only the receiver knows how to decode this (as agreed upon before the start of the transmission), there is no scope for confusion. CDMA is the newest and the least used access method using satellites. It is popular in military installations, but not so much in the case of commercial applications.

1.6. SWITCHING TECHNIQUES

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

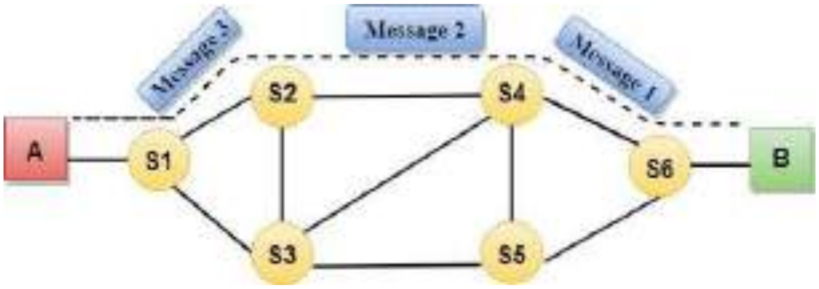
Classification of Switching Techniques



Types of Switching

1.6.1. Circuit Switching

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



- Fixed data can be transferred at a time in circuit switching technology. Communication through circuit switching has 3 phases:
    - Circuit establishment
    - Data transfer
    - Circuit Disconnect

Advantages:

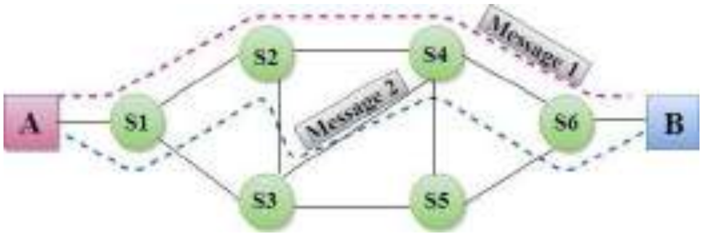
- In the case of Circuit Switching technique, the communication channel is dedicated.
- It has fixed bandwidth.

Disadvantages:

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

1.6.2. Message Switching

- Message Switching is a switching technique in which a message is transferred as a complete unit and routed through intermediate nodes at which it is stored and forwarded.
- In Message Switching technique, there is no establishment of a dedicated path between the sender and receiver.
- The destination address is appended to the message. Message Switching provides a dynamic routing as the message is routed through the intermediate nodes based on the information available in the message.
- Each and every node stores the entire message and then forward it to the next node.
- This type of network is known as store and forward network.
- Message switching treats each message as an independent entity.



Advantages of Message Switching:

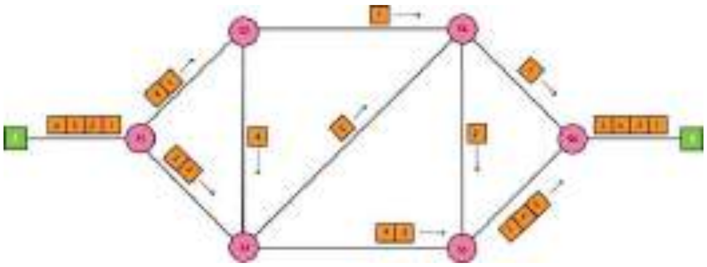
- Data channels are shared among the communicating devices that improve the efficiency of using available bandwidth.
- Traffic congestion can be reduced because the message is temporarily stored in the nodes.
- Message priority can be used to manage the network.
- The size of the message which is sent over the network can be varied. Therefore, it supports the data of unlimited size.

Disadvantages of Message Switching:

- The message switches must be equipped with sufficient storage to enable them to store the messages until the message is forwarded.
- The long delay can occur due to the storing and forwarding facility provided by the message switching technique.

1.6.3. Packet Switching

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



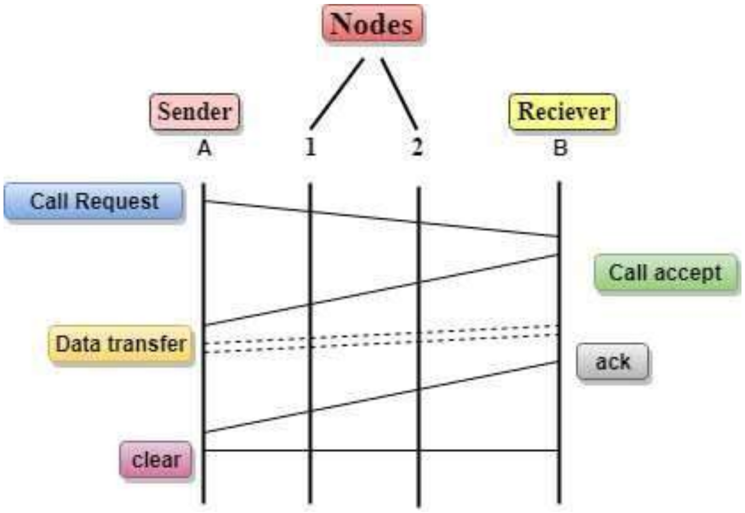
There are two approaches to Packet Switching:

Datagram Packet switching:

- It is a packet switching technology in which packet is known as a datagram, is considered as an independent entity. Each packet contains the information about the destination and switch uses this information to forward the packet to the correct destination.
- The packets are reassembled at the receiving end in correct order.
- In Datagram Packet Switching technique, the path is not fixed.
- Intermediate nodes take the routing decisions to forward the packets.
- Datagram Packet Switching is also known as connectionless switching.

Virtual Circuit Switching

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



- In the above diagram, A and B are the sender and receiver respectively. 1 and 2 are the nodes.

- Call request and call accept packets are used to establish a connection between the sender and receiver.
- When a route is established, data will be transferred.
- After transmission of data, an acknowledgment signal is sent by the receiver that the message has been received.
- If the user wants to terminate the connection, a clear signal is sent for the termination.

Advantages of Packet Switching:

- Cost-effective: In packet switching technique, switching devices do not require massive secondary storage to store the packets, so cost is minimized to some extent. Therefore, we can say that the packet switching technique is a cost-effective technique.
- Reliable: If any node is busy, then the packets can be rerouted. This ensures that the Packet Switching technique provides reliable communication.
- Efficient: Packet Switching is an efficient technique. It does not require any established path prior to the transmission, and many users can use the same communication channel simultaneously, hence makes use of available bandwidth very efficiently.

Disadvantages of Packet Switching:

- Packet Switching technique cannot be implemented in those applications that require low delay and high-quality services.
- The protocols used in a packet switching technique are very complex and requires high implementation cost.
- If the network is overloaded or corrupted, then it requires retransmission of lost packets. It can also lead to the loss of critical information if errors are nor recovered.

UNIT 2 DATA LINK LAYER

In this lesson we will study the design principles for layer 2, the data link layer. This study deals with the algorithms for achieving reliable, efficient communication between two adjacent machines at the data link layer. machine X just puts the bits on the wire, and machine Y just takes them off. Unfortunately, communication circuits make errors occasionally. Furthermore, they have only a finite data rate, and there is a nonzero propagation delay between the time a bit is sent and the time it is received. These limitations have important implications for the efficiency of the data transfer. The protocols used for communications must take all these factors into consideration.

The main task of the data link layer is to take a raw transmission facility and transform it into a line that appears free of transmission errors in the network layer. Here the sender breaks the input data into data frames, transmit the frames sequentially, and process the acknowledgment frames sent back by the receiver. Since the physical layer merely accepts and transmits a stream of bits without any regard to meaning of structure, it is up to the data link layer to create and recognize frame boundaries. This can be accomplished by attaching special bitpatterns to the beginning and end of the frame. If there is a chance that these bit patterns might occur in the data, special care must be taken to avoid confusion.

The data link layer should provide error control between adjacent nodes. Another issue that arises in the data link layer is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism must be employed in order to let the transmitter know how much buffer space the receiver has at the moment. Frequently, flow regulation and error handling are integrated, for convenience. If the line can be used to transmit data in both directions, this introduces a new complication that the data link layer software must deal with.

2.1 DATA LINK LAYER DESIGN ISSUES

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

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

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

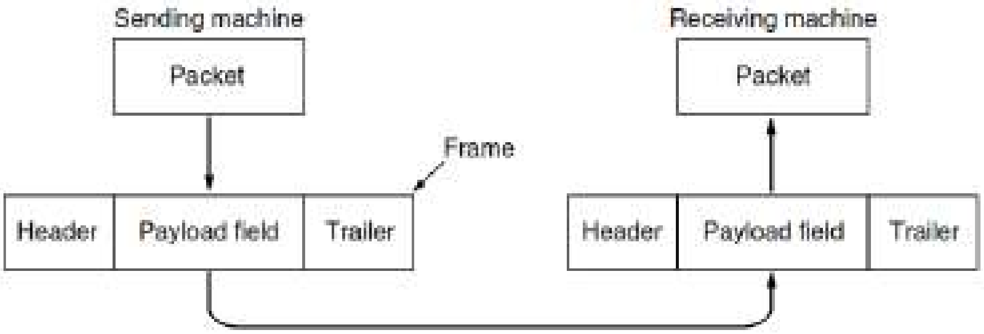


Figure 3-1. Relationship between packets and frames.

In data link layer these are the requirements and objectives for effective data communication between two directly connected transmitting-receiving stations:

Frame Synchronization-Data are sent in blocks called frames. The beginning and end of each frame must be recognizable.

Flow control-The sending station must not send frames at a rate faster than the receiving station can absorb them.

Error control-Any bit errors introduced by the transmission system must be corrected. Addressing-On a multipoint line, such as a local area network (LAN), the identity of the two stations involved in a transmission must be specified.

Control and data on same link- It is usually not desirable to have a physically separate communications path for control information. Accordingly, the receiver must be able to distinguish control information from the data being transmitted.

Link management-The initiation, maintenance, and termination of a sustained data exchange requires a fair amount of coordination and cooperation among stations. Procedures for the management of this exchange are required.

2.1.1 Services Provided to the Network Layer

The function of the data link layer is to provide services to the network layer. The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine. On the source machine is an entity, call it a process, in the network layer that hands some bits to the data link layer for transmission to the destination. The job of the data link layer is to transmit the bits to the destination machine so they can be handed over to the network layer there.

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

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

Unacknowledged connectionless service consists of having the source machine send independent frames to the destination machine without the destination machine acknowledge them. No logical connection is established beforehand or released afterward.

Acknowledged connectionless service, there are still no logical connections used, but each frame sent is individually acknowledged. In this way, the sender knows whether a frame has arrived correctly. Each frame sent is individually acknowledged. In this way, the sender knows whether a frame has arrived correctly or been lost. If it has not arrived within a specified time interval, it can be sent again.

Connection-oriented service, the source and destination machines establish a connection before any data is transferred. Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received. Furthermore, it guarantees that each frame is received exactly once and that all frames are received in the right order.

With connectionless service, in contrast, it is conceivable that a lost acknowledgement causes a packet to be sent several times and thus received several times. Connection-oriented service, in contrast, provides the network layer processes with the equivalent of a reliable bit stream.

When connection-oriented service is used, transfers go through three distinct phases. In the first phase, the connection is established by having both sides initialize variables and counters needed to keep track of which frames have been received and which ones have not. In the second phase, one or more frames are actually transmitted. In the third and final phase, the connection is released, freeing up the variables, buffers, and other resources used to maintain the connection.

2.1.2 Framing

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to detect error and, if necessary, correct errors. The usual approach is for the data link layer is to break the bit stream into discrete frames and compute the checksum for each frame. When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it. Breaking the bit stream up into frames is more difficult than it at first appears. In this section we will look at four methods:

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

In this method, a field in the header is used to specify the number of characters in the frame. This number helps the receiver to know the number of characters in the fame following this. This technique is shown in Fig. 3-3(a) for four small example frames of sizes 5, 5, 8, and 8 bytes, respectively.

The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the byte count of 5 in the second frame of Fig. 3-3(b) becomes a 7 due to a single bit flip, the destination will get out of synchronization. It will then be unable to locate the correct 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 bytes to skip over to get to the start of the retransmission. For this reason, the byte count method is rarely used by itself.

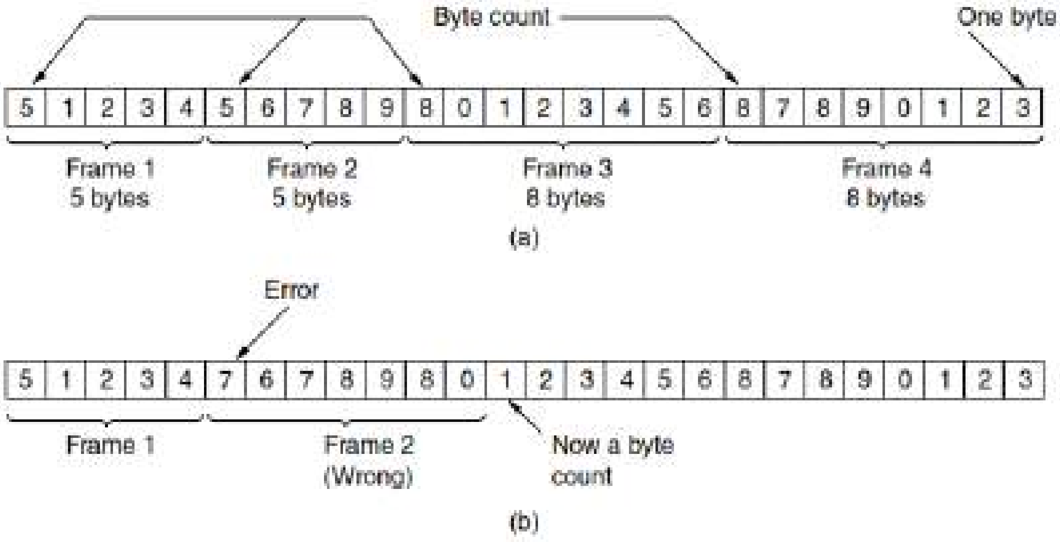


Figure 3-3. A byte stream. (a) Without errors. (b) With one error.

The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. Often the same byte, called a flag byte, is used as both the starting and ending delimiter. This byte is shown in Fig. 3-4(a) as FLAG. Two consecutive flag bytes indicate the end of one frame and the start of the next. However, there is a still a problem we have to solve. It may happen that the flag byte occurs in the data, especially when binary data such as photographs or songs are being transmitted. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each ‘‘accidental’’ flag byte in the data. Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it. The data link layer on the receiving end removes the escape bytes before giving the data to the network layer. This technique is called **byte stuffing.**

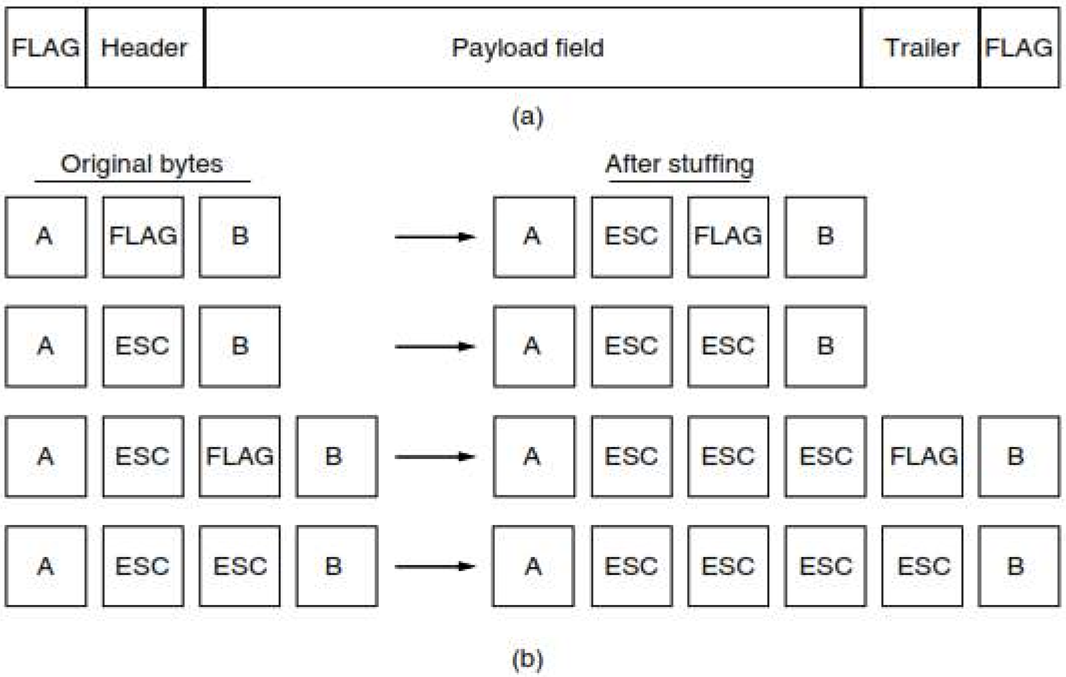


Figure 3-4. (a) A frame delimited by flag bytes. (b) Four examples of byte sequences before and after byte stuffing.

Of course, the next question is: what happens if an escape byte occurs in the middle of the data? The answer is that it, too, is stuffed with an escape byte. At the receiver, the first escape byte is removed, leaving the data byte that follows it (which might be another escape byte or the flag byte). Some examples are shown in Fig. 3-4(b). In all cases, the byte sequence delivered after destuffing is exactly the same as the original byte sequence. We can still search for a frame boundary by looking for two flag bytes in a row, without bothering to undo escapes.

The third method of delimiting the bit stream gets around a disadvantage of byte stuffing, which is that it is tied to the use of 8-bit bytes. Framing can be also be done at the bit level. Each frame begins and ends with a special bit pattern, 01111110 or 0x7E in hexadecimal. This pattern is a flag byte. Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 3-5 gives an example of bit stuffing.

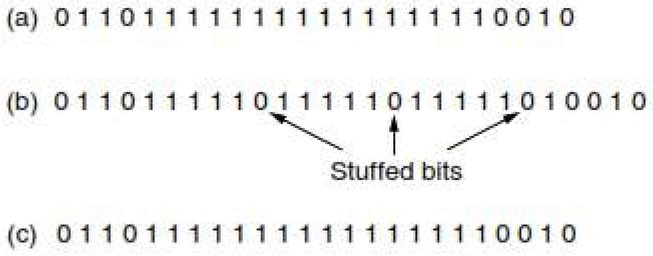

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0
(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0
Stuffed bits
(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Figure 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

A common pattern used for Ethernet and 802.11 is to have a frame begin with a well-defined pattern called a preamble. This pattern might be quite long (72 bits is typical for 802.11) to allow the receiver to prepare for an incoming packet. The preamble is then followed by a length (i.e., count) field in the header that is used to locate the end of the frame.

2.1.3 Error Control

The next problem is how to make sure that all frames are eventually delivered to the network layer at the destination and in the proper order. The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line. Typically, the protocol calls for the receiver to send back special control frames bearing positive or negative acknowledgements about the incoming frames. If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely. On the other hand, a negative acknowledgement means that something has gone wrong, and the frame must be transmitted again. The obvious solution is to just transmit the frame again. However, when frames may be transmitted multiple times there is a danger that the receiver will accept the same frame two or more times and pass it to the network layer more than once. To prevent this from happening, it is generally necessary to assign sequence numbers to outgoing frames, so that the receiver can distinguish retransmissions from originals. The whole issue of managing the timers and sequence numbers so as to ensure that each frame is ultimately passed to the network layer at the destination exactly once, no more and no less, is an important part of the data link layer's duties.

2.1.4 Flow Control

Another important design issue that occurs in the data link layer is what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them. This situation can easily occur when the sender is running on a fast computer and the receiver is running on a slow machine. The sender keeps pumping the frames out at a high rate until the receiver is completely swamped. Even if the transmission is error free, at a certain point the receiver will simply be unable to handle the frames as they arrive and will start to lose some. Clearly, something has to be done to prevent this situation. Two approaches are commonly used. In the first one, feedback-based flow control, the receiver sends back information to the sender giving it permission to send more data or at least telling the sender how the receiver is doing. In the second one, rate-based flow control, the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver.

2.2. ERROR DETECTION AND CORRECTION:

Errors are introduced into the binary data transmitted from the sender to the receiver due to noise during transmission. The error can be a single-bit error, multi-bit error, or burst error. Error detection methods are used to check whether the receiver has received correct data or corrupted data. Error correction is used to correct the detected errors during the transmission of data from sender to receiver.

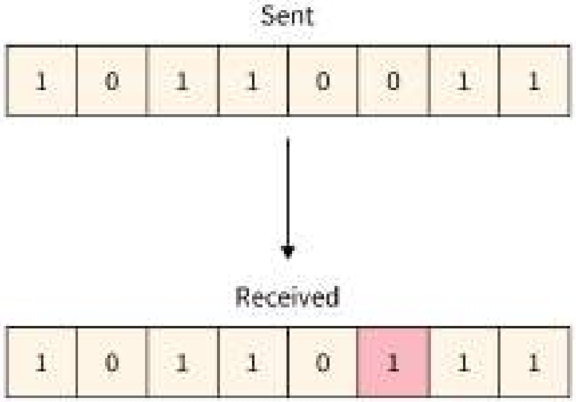
When the information received at the receiver end does not match the sent data. At the time of transmission, errors are introduced into the binary data sent from the sender to the receiver due to noise during transmission. This means that a bit having a **0** value can change to **1** and a bit having a **1** value can change to **0**.

2.2.1. Types of Errors

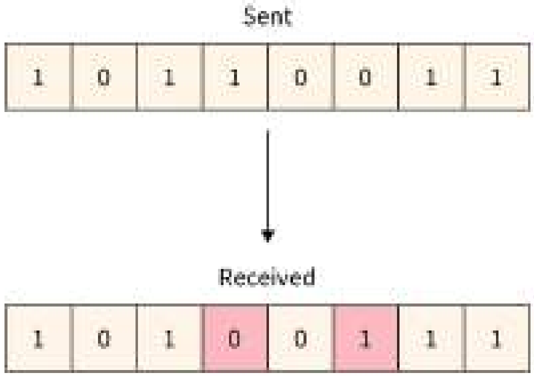
Some errors that occur during communication are given below:

Single-bit Error

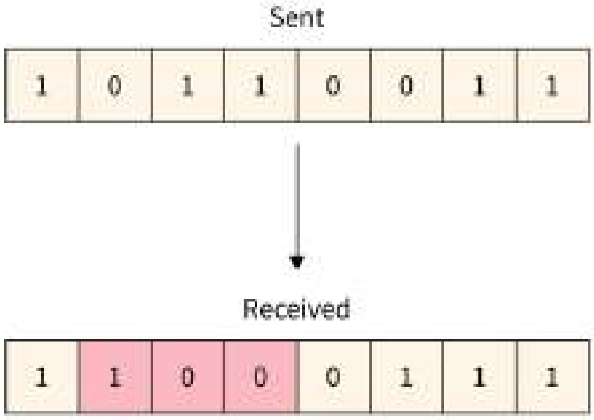
Typically, only one bit of the frame received is corrupt, and the corrupted bit can be located anywhere in the frame.



Multiple-bit Error: More than **one bit received** in the frame is found to be corrupted. Refer to the below image for the multiple-bit error



Burst Error: More than one consecutive bit is corrupted in the received frame.



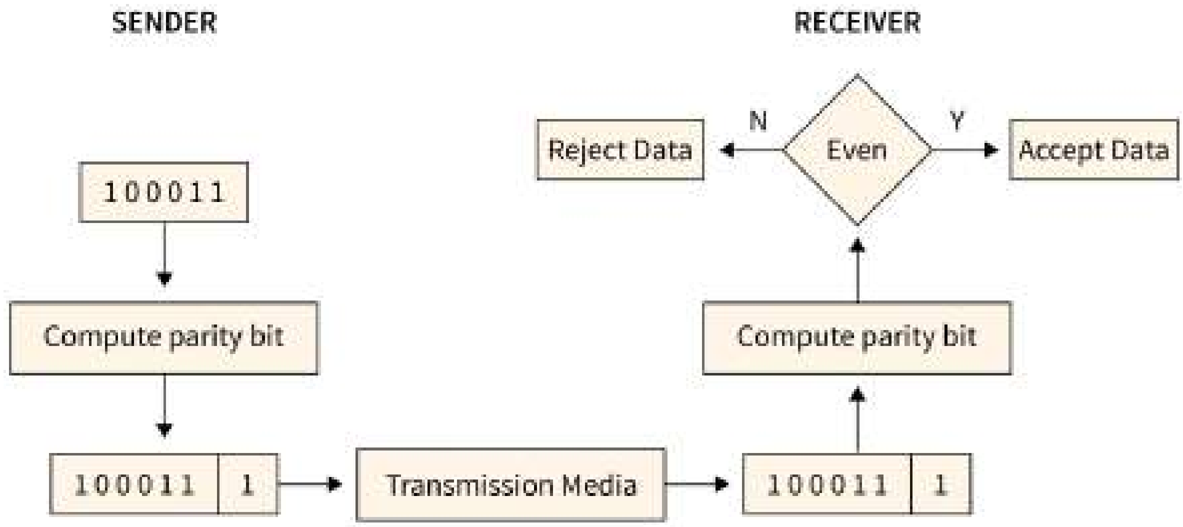
2.2.2. Error Detection:

2.2.2.1. Simple Parity Check

Data sent from the sender undergoes parity check :

- 1 is added as a parity bit to the data block if the data block has an **odd number of 1's**.
- 0 is added as a parity bit to the data block if the data block has an **even number of 1's**.

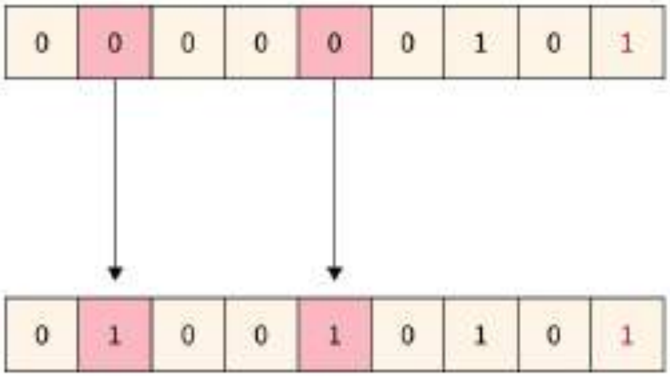
This procedure is used for making the **number of 1's even**. This is commonly known as even parity checking.



Disadvantage:

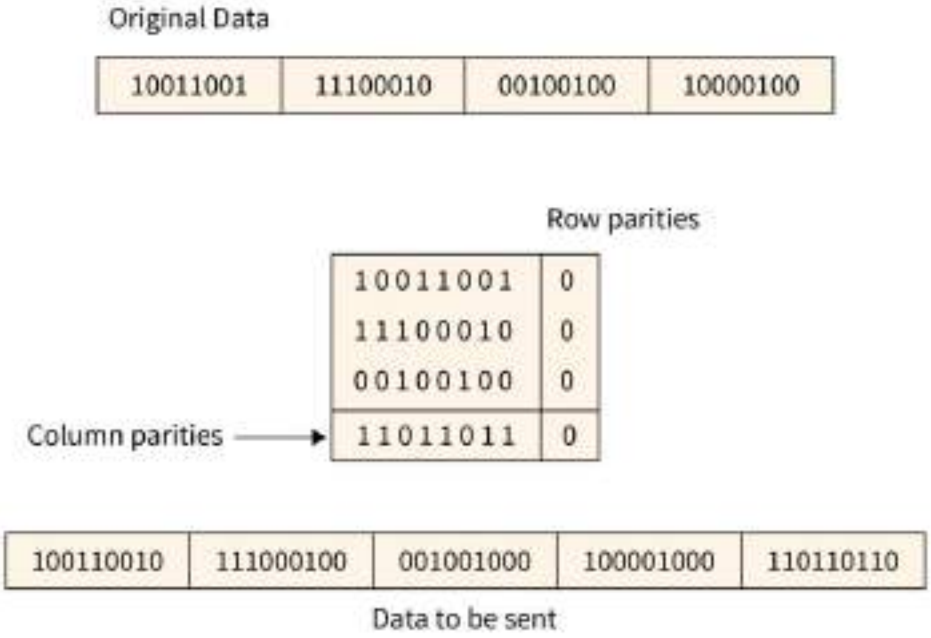
- Only **single-bit error** is detected by this method, it fails in multi-bit error detection.
- It cannot detect an error in case of an error in **two bits**.

Refer to the below image for the disadvantage simple parity checking method



Two-Dimensional Parity Check

For every **row and column**, parity check bits are calculated by a simple method of parity check. Parity for both rows and columns is transmitted with the data sent from sender to receiver. At the receiver’s side, parity bits are compared with the calculated parity of the data received.



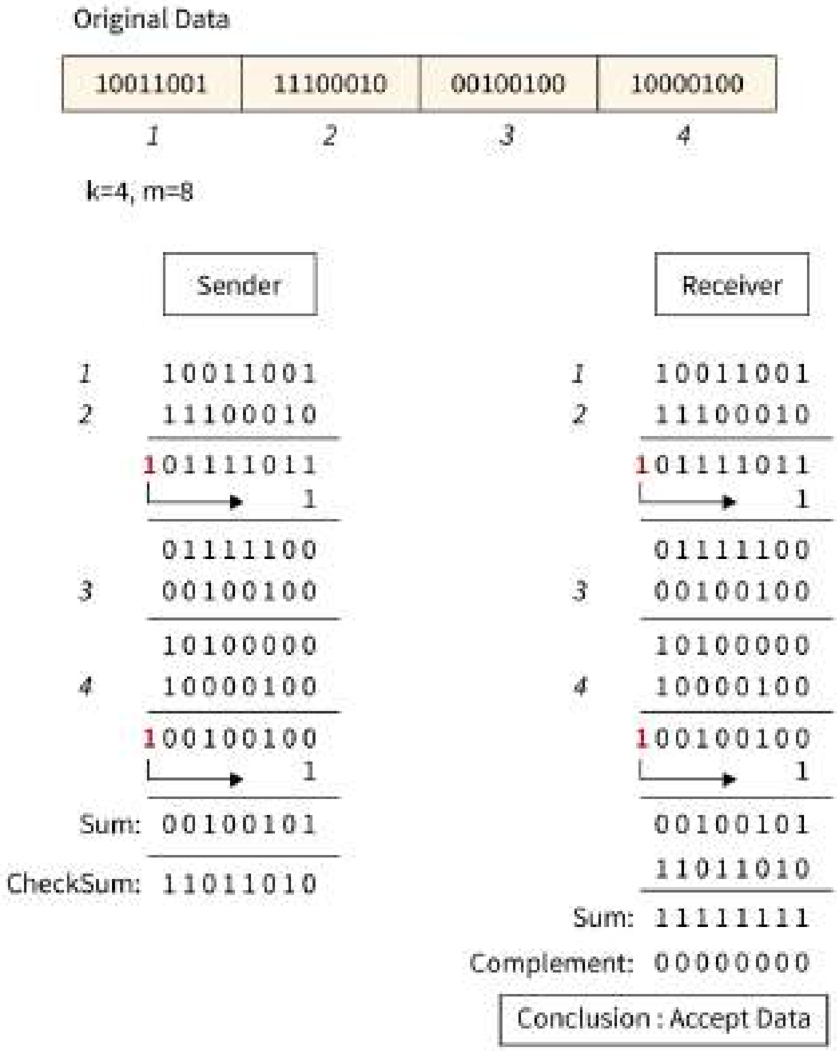
Disadvantages:

- If **2 bits are corrupted in 1 data unit** and another data unit exactly at the same position is corrupted then this method is not able to detect the error.
- Sometimes this method is not used for **detecting 4-bit **errors or more than 4-bit errors.

2.2.2.2. Checksum

Checksum is an error detection which detects the error by dividing the data into segments of equal size and then **use 1's complement** to find the sum of the segments and then the sum is transmitted with the data to the receiver and same process is done by the receiver and at the receiver side, all **zeros** in the sum indicates the correctness of the data.

1. First of all data is divided into **k segments** in a checksum error detection scheme and each segment has m bits.
2. For finding out the sum at the sender's side, all segments are added through **1's complement arithmetic**. And for determining the checksum we complement the sum.
3. Along with data segments, the checksum segments are also transferred.
4. All the segments that are received on the receiver's side are added through 1S complement arithmetic to determine the sum. Then complement the sum.
5. The received data is accepted only on the condition that the result is found to be **0**. And if the result is not 0 then it will be discarded.

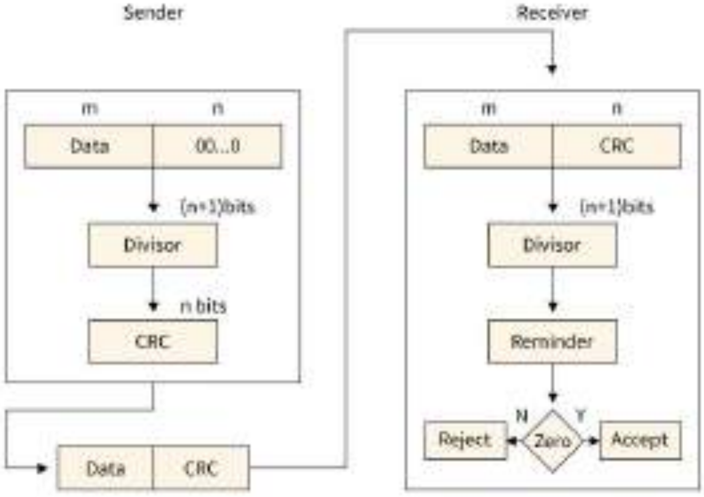


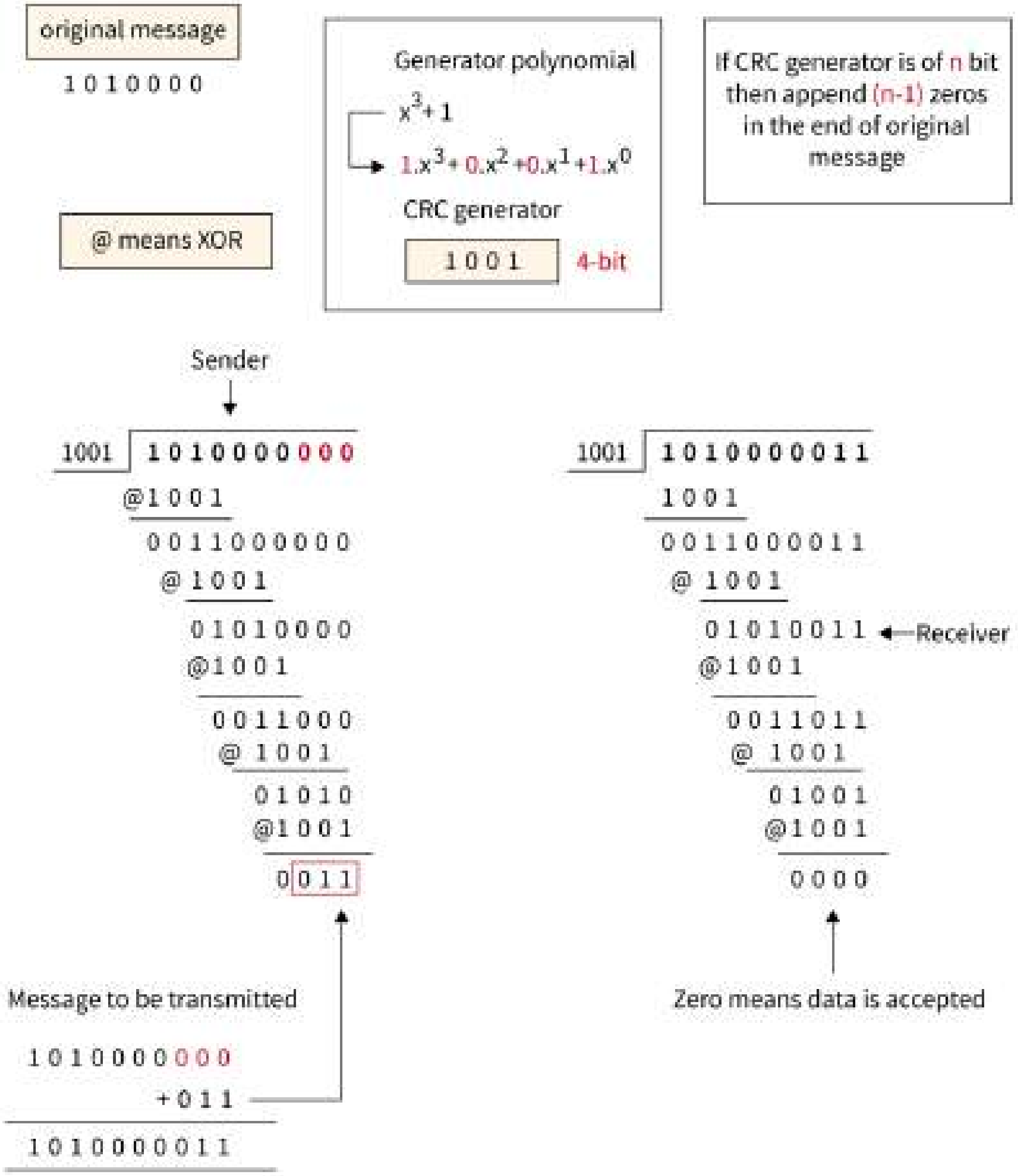
Disadvantages: In **checksum error** is not detected, if one sub-unit of the data has one or more corrupted bits and corresponding bits of the opposite value are also corrupted in another sub-unit. Error is not detected in this situation because in this case the sum of columns is not affected by corrupted bits.

2.2.2.3. Cyclic Redundancy Check

- The checksum scheme uses the addition method but **CRC** uses binary division.
- A bit sequence commonly known as cyclic redundancy check is added to the end of the bits in CRC. This is done so that the resulting data unit will be divisible by the second binary number that is predetermined.
- The receiving data units on the receiver's side need to be divided by the same number. These data units are accepted and found to be correct only on the condition that the remainder of this division is zero. The remainder shows that the data is not correct. So, they need to be discarded.

Disadvantages: Cyclic Redundancy Check may lead to **overflow of data**.





ERROR CORRECTION

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

A single additional bit can detect the error, but cannot correct it. For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.

Suppose r is the number of redundant bits and d is the total number of the data bits. The number of redundant bits r can be calculated by using the formula:

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

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

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

Hamming Code

Parity bits: The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

Even parity: To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

Odd Parity: To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

Algorithm of Hamming code:

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

Relationship b/w Error position & binary number.

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

Total number of data bits 'd' = 4

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

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

Apply the value of r from 1,2,3…. until it satisfy the above condition. Therefore, the value of r is 3 that satisfies the above relation.

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

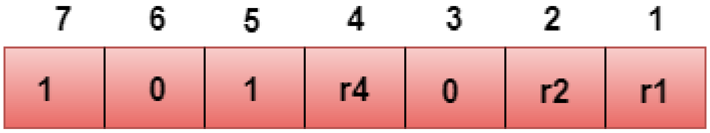
Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Determining the position of the redundant bits

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

1. The position of r1 = 1
2. The position of r2 = 2
3. The position of r4 = 4

Representation of Data on the addition of parity bits:

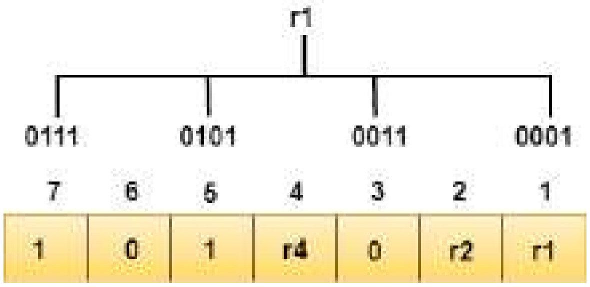


7	6	5	4	3	2	1
1	0	1	r4	0	r2	r1

Determining the Parity bits

Determining the r1 bit

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

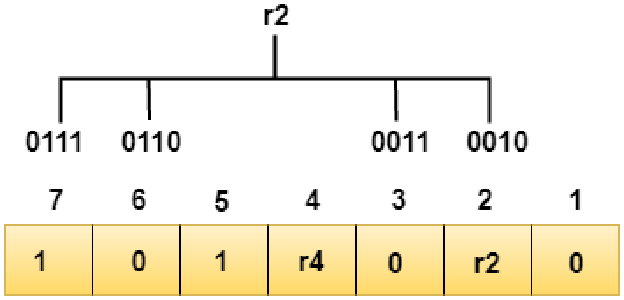


7	6	5	4	3	2	1
1	0	1	r4	0	r2	r1

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

Determining r2 bit

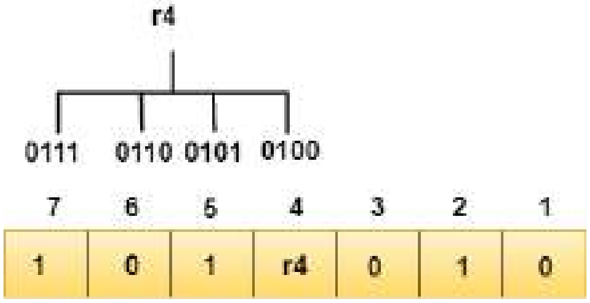
The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



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

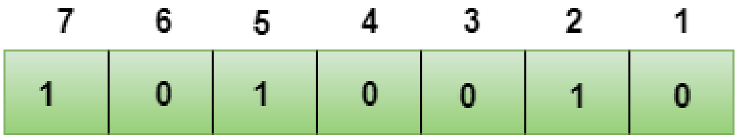
Determining r4 bit

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



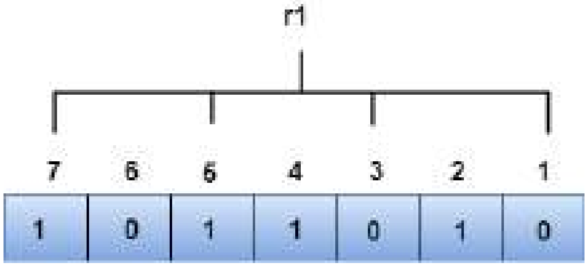
We observe from the above figure that the bit positions that includes 1 in the third position are **4, 5, 6, 7**. Now, we perform the even-parity check at these bit positions. The total number of 1 at these bit positions corresponding to r4 is **even, therefore, the value of the r4 bit is 0**.

Data transferred is given below:



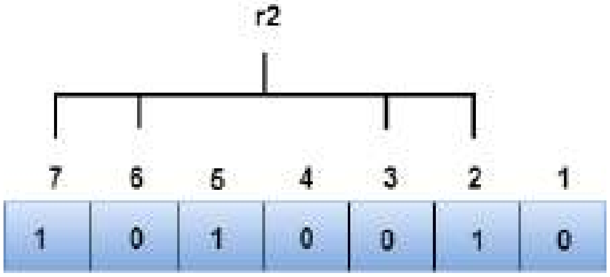
Suppose the 4th bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

R1 **bit** : The bit positions of the r1 bit are 1,3,5,7



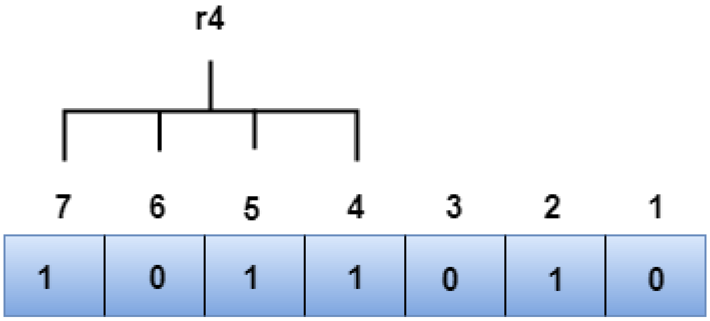
We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

R2 bit: The bit positions of r2 bit are 2,3,6,7.



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

R4 bit: The bit positions of r4 bit are 4,5,6,7.



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

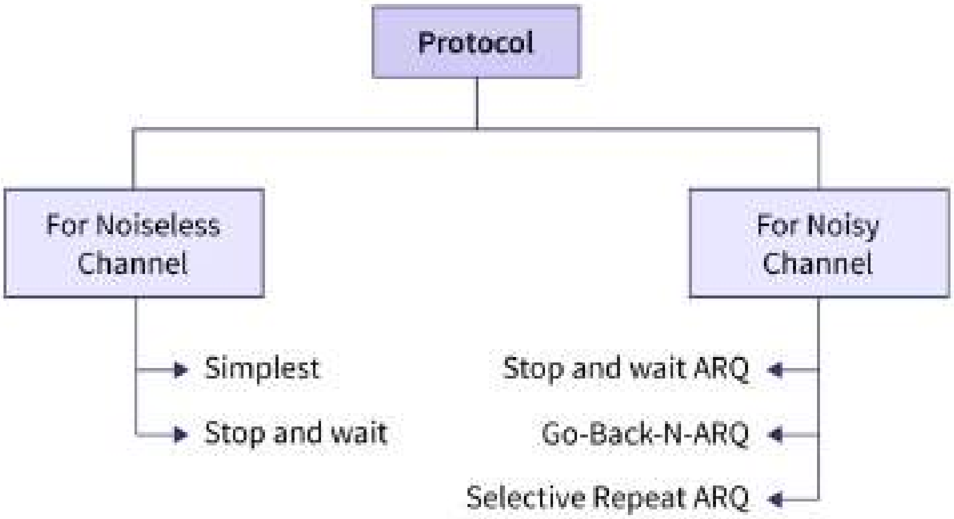
- *The binary representation of redundant bits, i.e., $r_4\, r_2\, r_1$ is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4th bit position. The bit value must be changed from 1 to 0 to correct the error.*

Exercise Problems:

1. Construct the even parity Hamming code word for a data byte 10101101. The resultant hamming code is transmitted to the receiver end. At the receiver side, the data the fourth bit from leftmost data is corrupted. identify the corrupt data and correct it.
2. we have message that we want to transmit: 111011. Construct the even parity Hamming code word. Assuming that there is an error during transmission in the above hamming code example, and the message received is 1110010011. Locate the position of the error bit and reverse it.
3. Suppose we have 10011001 11100010 00100100 10000100. calculate the checksum and validate the checksum at the receiver end.
4. Assume that g(x) is CRC-4, which equals X4 + X2 + 1, and the source data M is 101011011. calculate CRC value.

2.3. ELEMENTARY DATA LINK LAYER PROTOCOLS:

Data communication requires at least two devices working together, one to send and the other to receive. Even such a basic arrangement requires a great deal of coordination for an intelligible exchange to occur. The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.



A protocol refers to a defined set of guidelines and regulations that control the communication between different devices in a network. These guidelines specify the way in which data is formatted, timed, sequenced, and checked for errors during transmission.

Protocols play a crucial role in ensuring accurate, efficient, and clear communication between devices. They make it possible for devices of different types and brands to communicate with each other, and they provide a consistent method for transmitting data across a network.

Headers are responsible for selecting protocols, which are rules governing communication between devices in a network. These headers contain information describing the message's nature and the handling it will receive. To detect any issues, the headers must include information such as the message's checksum, destination, and source addresses, and other necessary header addresses.

2.3.1 Noiseless Channel protocols

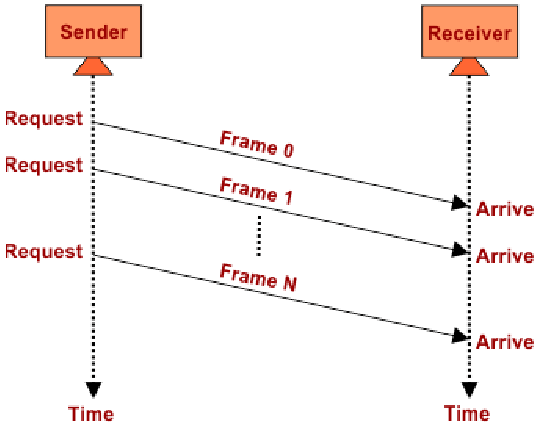
The noiseless channel has the following two protocols:

Simplex Protocol:

A simple protocol for a noiseless channel would be one that involves the direct transfer of data from the source to the destination without any intermediate processing. In this scenario, the channel is assumed to be noise-free, which means that the data does not get corrupted. In a noiseless channel, a simple protocol could consist of a straightforward method for transmitting data, such as sending one bit at a time, with no error correction or flow control mechanisms in place. This type of protocol is ideal for a noiseless channel as the lack of noise ensures that the data transmitted is received accurately, making the implementation of additional measures unnecessary.

Limitations:

The simplex protocol does not have any mechanisms for controlling the flow of data or detecting and correcting errors, as it is only used in noise-free channels. The protocol assumes that the receiver is always ready to process any data frames sent by the sender immediately. This type of protocol, which only allows data to flow from the sender to the receiver, is known as a one-way protocol. Since this protocol is unidirectional, there is no need for an acknowledgment or confirmation of receipt. Furthermore, because there is no data loss during transmission, there is no need to resend or retransmit the data.



Simplest Protocol

Stop-and-Wait Protocol

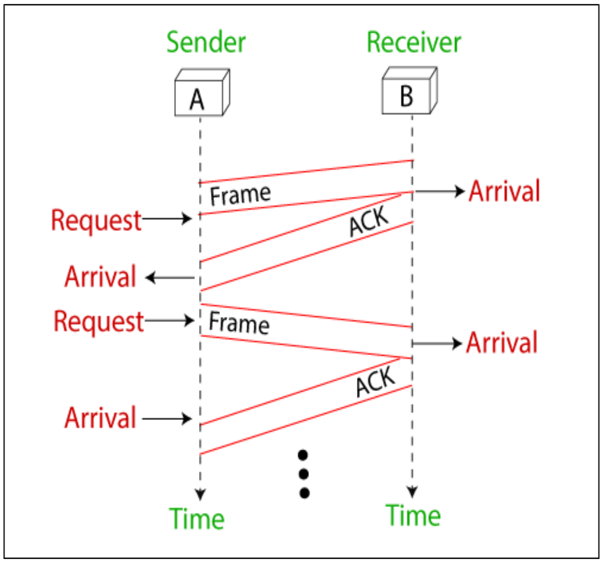
Stop and wait is a protocol that is used for reliable data transmission in a noiseless channel. In this protocol, the sender sends a single packet at a time and waits for an acknowledgment (ACK) from the receiver before sending the next packet. This way, the sender can ensure that each packet is received by the receiver and has been successfully processed. If the sender does not receive an ACK within a certain time frame, the packet is considered lost and must be retransmitted.

The stop and wait protocol is simple and efficient, but it has **one major drawback.** Because only one packet can be transmitted at a time, the overall data transmission rate is relatively slow. To overcome this limitation, the sliding window protocol was developed. In the sliding window protocol, multiple packets can be transmitted at the same time, allowing for faster data transmission. Despite this limitation, the stop and wait protocol is still widely used in many applications due to its simplicity and reliability.

Flow Diagram

The flow diagram of the Stop-and-wait protocol in a noiseless channel involves the following steps:

1. The sender transmits a data frame to the receiver.
2. The sender waits for an acknowledgment (ACK) from the receiver.
3. The receiver processes the received data frame.
4. The receiver sends an ACK to the sender to confirm receipt of the data frame.
5. The sender continues to transmit the next data frame, repeating the process from step i.



In this protocol, the sender sends one frame at a time and stops until it receives an ACK from the receiver. This prevents the receiver from becoming overwhelmed with incoming frames and ensures reliable data transmission. Additionally, the ACK frames are used for flow control, allowing the sender to adjust the transmission rate based on the receiver's processing capacity.

The main difference between the two protocols is that the Simplex Protocol has no flow control and error control mechanisms, while the Stop-and-Wait Protocol employs a flow control mechanism through the use of ACK frames.

In the Simplex Protocol, the recipient is always expected to be ready to receive any frames sent by the sender, so no acknowledgment is needed. In the Stop-and-Wait Protocol, the sender must wait for an acknowledgment from the receiver before sending the next frame.

Another difference between the two protocols is that the Simplex Protocol is unidirectional, while the Stop-and-Wait Protocol is bi-directional. In the Simplex Protocol, data frames can only move in one direction, from sender to receiver, while in the Stop-and-Wait Protocol, both data frames and ACK frames can travel in both directions.

2.3.2. Noisy Channel protocols

A Noisy Channel Protocol is a type of communication protocol that is used in communication systems where the transmission channel may introduce errors into the transmitted data. This type of protocol is designed to deal with errors in the communication channel and ensure that the data being transmitted is received accurately at the receiver end. Some examples of Noisy Channel Protocols include the Stop-and-Wait Protocol, the Sliding Window Protocol, and the Automatic Repeat Request (ARQ) Protocol.

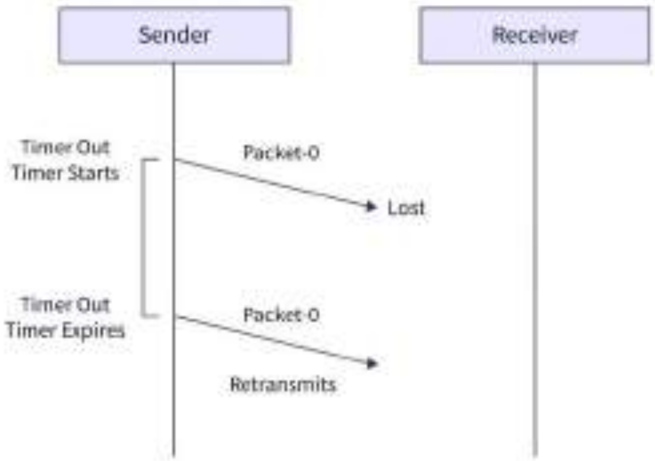
1.Stop-and-Wait ARQ Protocol

The Stop and Wait protocol is a protocol used for reliable data transmission over a noisy channel. In this protocol, the sender only sends one frame at a time and waits for an acknowledgment (ACK) from the receiver before sending the next frame. This helps to ensure that the receiver receives the data correctly and eliminates the need for retransmission in the case of errors caused by the noisy channel. The sender continuously monitors the channel for errors, and if an error is detected, it waits for the next ACK before resending the frame. This protocol adds error control to the basic unidirectional communication of data frames and ACK frames in the opposite direction.

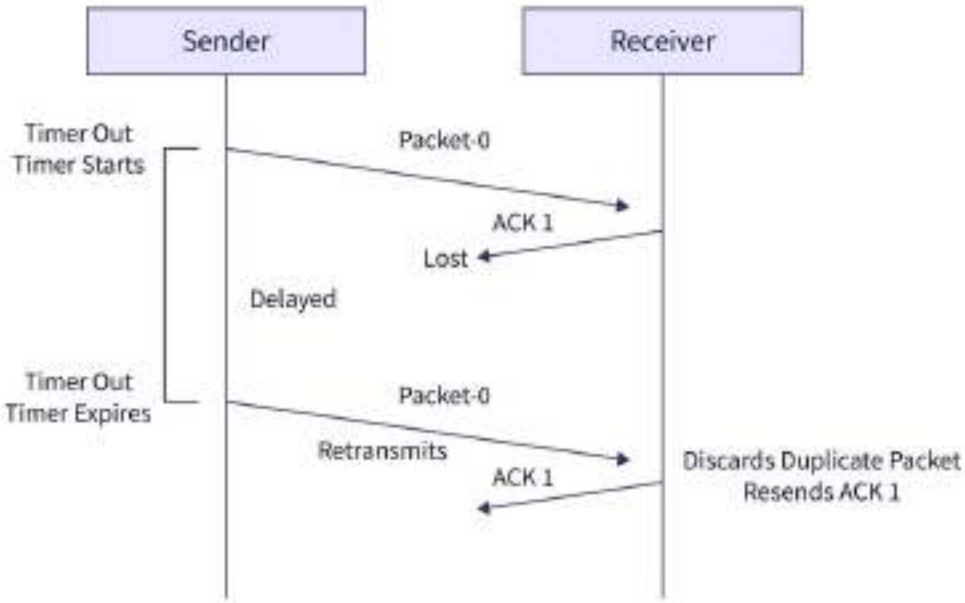
Let us discuss the various problems of the Stop and Wait ARQ -

1. **Problem of Lost Data Packet** When the sender sends the data packet and the receiver does not receive the data packet, it means that the data is lost in between the transmission. So, to overcome this type of problem, the sender uses a timer. When the sender sends the data packet, it starts a

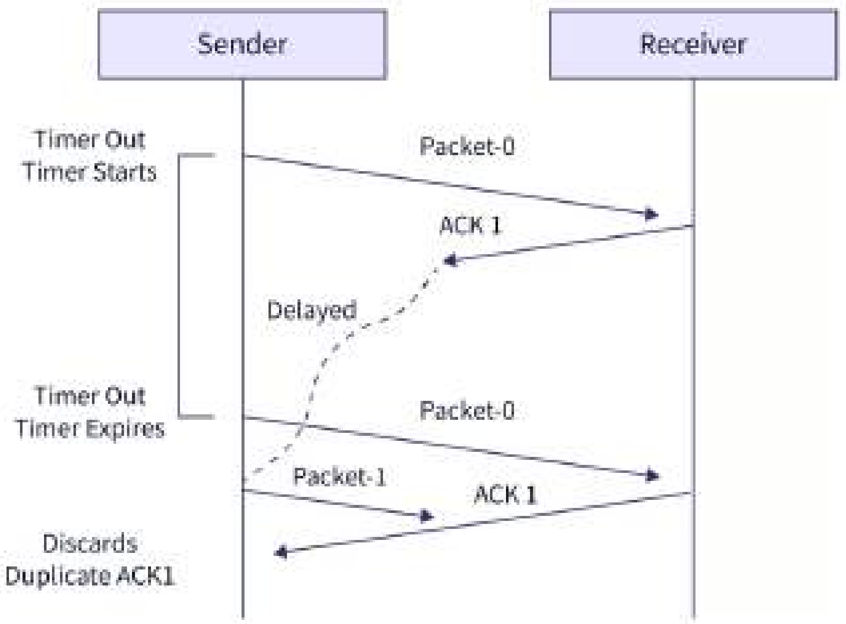
timer. If the timer goes off before receiving the acknowledgment from the receiver, the sender retransmits the same data packet. Refer to the image below for better visualization.



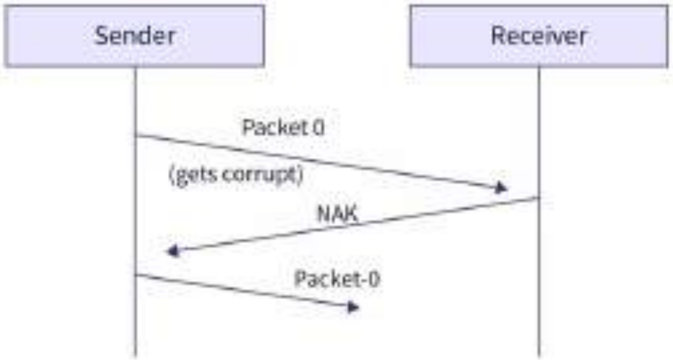
2. **Problem of Lost Acknowledgement** When the sender sends the data packet and the receiver receives the data packet but the acknowledgment from the receiver is not received. It means that the acknowledgment is lost in between the transmission. So, to overcome this type of problem, the sender uses sequence numbering. When the sender sends the data packet, it attaches a certain sequence number which helps the receiver identify the data packet. If the timer goes off before receiving the acknowledgment from the receiver, the sender retransmits the same data packet. But in this case, the receiver already has the data packet, so it discards the data and sends it back an acknowledgment. This tells the sender that the certain data packet is now received correctly.



3. **Problem of Delayed Acknowledgement** When the sender sends the data packet and the receiver receives the data packet but the acknowledgment from the receiver is not received. It means that the acknowledgment is lost or delayed in between the transmission. So, to overcome this type of problem, the sender uses sequence numbering. When the sender sends the data packet, it attaches a certain sequence number which helps the receiver identify the data packet. If the timer goes off before receiving the acknowledgment from the receiver, the sender retransmits the same data packet. But in this case, the receiver already has the data packet, so it discards the data and sends it back the acknowledgment again. Now, if the sender has received the previously sent acknowledgment then the newer acknowledgment is discarded by the sender. Refer to the image below for better visualization.



4. **Problem of Damaged Packet** When the sender sends the data packet and the receiver receives the data packet the received data packet is corrupted. So, to overcome this type of problem, the receiver uses negative acknowledgment (NACK). So, if the sender receives a negative acknowledgment then the sender retransmits the data packet.



In this protocol, the sender only sends one data frame at a time and waits for a response from the receiver before sending the next frame. This ensures that the receiver has enough time to process each frame before receiving the next one. The Stop-and-Wait protocol is reliable, but has low throughput compared to other protocols.

Disadvantages of Stop and Wait Protocol

Suppose the sender sends the data packet, but the data packet is lost due to some reason. Since the receiver has not received the packet for a long time, the sender does not receive any acknowledgment from the receiver, so it will not send the next packet. In this case, two problems occur in data transmission:

- For an acknowledgment, the sender has to wait for an infinite amount of time.
- The receiver will also have to wait for an infinite amount of time to receive the data.

Sliding Window Protocol

The sliding window protocol is a data link layer protocol that is useful in the sequential and reliable delivery of data frames. Using the sliding window protocol, the sender can send multiple frames at a time. When the receiver receives the frame, it sends back an ACK (acknowledgment) to the sender. The ACK lets the sender know that a particular frame is received by the receiver correctly.

The sliding window protocol uses a mechanism of sequence numbers. The sender associates a sequence number to the data frames so that the receiver can use this sequence number to arrange the frames in order if any frame was re-transmitted. The sequence number also helps the receiver identify the lost or damaged packets.

In the sliding window protocol, we use a buffer space called a window to hold the frames at the sender's end and the receiver's end. The window on the sender's side covers the sequence of data packets that are sent (or to be sent). On the other hand, the window on the receiver's side covers the sequence of data packets that are received (or to be received).

Types of Sliding Window Protocols

There are two types of sliding window protocols namely - Go-Back-N ARQ, and Selective Repeat ARQ.

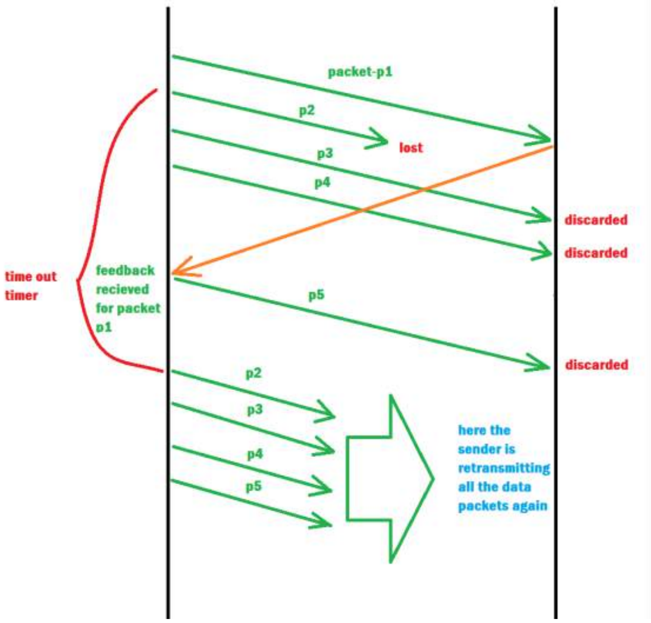
2.Go-Back-N ARQ Protocol

The Go-Back-N Automatic Repeat Request (ARQ) protocol is a type of error-control protocol used in data communication to ensure reliable delivery of data over a noisy channel. In a noisy channel, the probability of errors in the received packets is high, and hence, there is a need for a mechanism to detect and correct these errors.

The Go-Back-N ARQ protocol is a type of sliding window protocol where the sender transmits a window of packets to the receiver, and the receiver sends back an acknowledgment (ACK) to the sender indicating successful receipt of the packets. In case the sender does not receive an ACK within a specified timeout period, it retransmits the entire window of packets.

Flow Diagram

The flow diagram that illustrates the operation of the Go-Back-N ARQ protocol in a noisy channel:



Sender Side:

a. The sender transmits a window of packets to the receiver, starting with sequence number i and ending with sequence number i + N - 1, where N is the window size.
b. The sender sets a timer for each packet in the window.
c. The sender waits for an acknowledgment (ACK) from the receiver.

Receiver Side:

a. The receiver receives the packets and checks for errors.
b. If a packet is received correctly, the receiver sends an ACK back to the sender with the sequence number of the next expected packet.
c. If a packet is received with errors, the receiver discards the packet and sends a negative acknowledgment (NAK) to the sender with the sequence number of the next expected packet.

Sender Side (in case of no ACK received):

1. If the sender does not receive an ACK before the timer for a packet expires, the sender retransmits the entire window of packets starting with the packet whose timer expired.
2. The sender resets the timer for each packet in the window.
3. The sender waits for an ACK from the receiver.

Sender Side (in case of NAK received):

a. If the sender receives a NAK from the receiver, the sender retransmits only the packets that were not correctly received by the receiver.
b. The sender resets the timer for each packet that was retransmitted.
c. The sender waits for an ACK from the receiver.

The above steps are repeated until all packets have been successfully received by the receiver. The Go-Back-N ARQ protocol provides a reliable mechanism for transmitting data over a noisy channel while minimizing the number of retransmissions required.

3.Selective Repeat ARQ Protocol

The Selective Repeat ARQ protocol is a type of error-control protocol used in data communication to ensure reliable delivery of data over a noisy channel. Unlike the Go-Back-N ARQ protocol which retransmits the entire window of packets, the Selective Repeat ARQ protocol retransmits only the packets that were not correctly received.

In the Selective Repeat ARQ protocol, the sender transmits a window of packets to the receiver, and the receiver sends back an acknowledgment (ACK) to the sender indicating successful receipt of the packets. If the receiver detects an error in a packet, it sends a negative acknowledgment (NAK) to the sender requesting retransmission of that packet.

In the Selective Repeat ARQ protocol, the sender maintains a timer for each packet in the window. If the sender does not receive an ACK for a packet before its timer expires, the sender retransmits only that packet.

At the receiver side, if a packet is received correctly, the receiver sends back an ACK with the sequence number of the next expected packet. However, if a packet is received with errors, the receiver discards the packet and sends back a NAK with the sequence number of the packet that needs to be retransmitted.

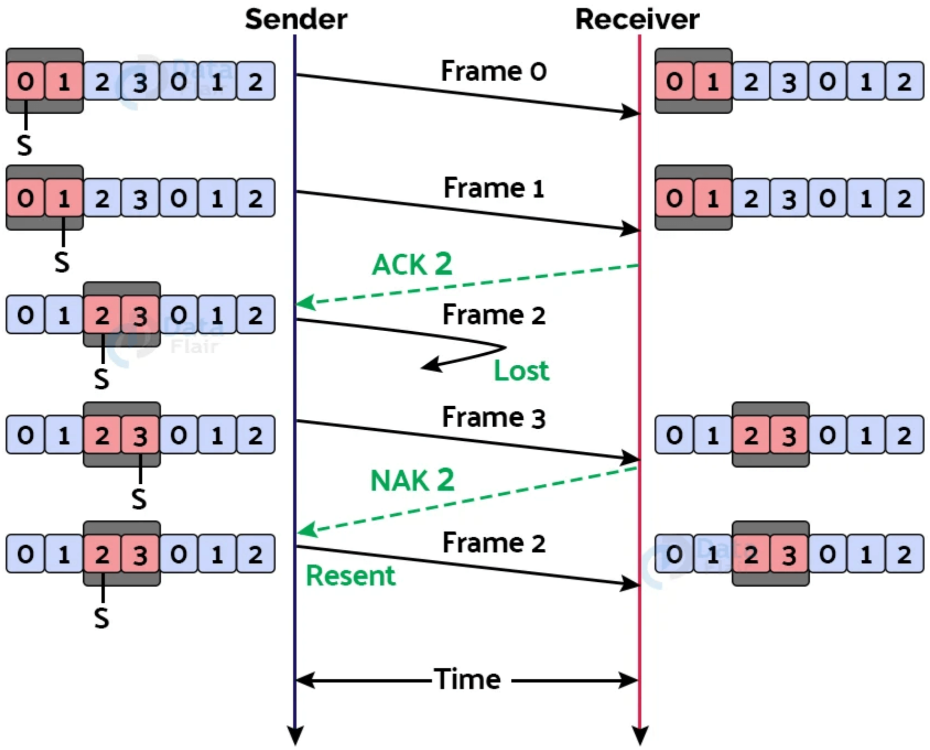
Unlike Go-Back-N ARQ, in Selective Repeat ARQ, the receiver buffer is maintained for all packets that are not in sequence. When a packet with a sequence number different from the expected sequence number arrives at the receiver, it is buffered, and the receiver sends an ACK for the last in-order packet it has received.

If a packet with a sequence number that the receiver has already buffered arrives, it is discarded, and the receiver sends an ACK for the last in-order packet it has received.

In summary, the Selective Repeat ARQ protocol provides a reliable mechanism for transmitting data over a noisy channel while minimizing the number of retransmissions required. It retransmits only the packets that were not correctly received and buffers packets that arrive out of order to reduce the number of retransmissions required.

Flow Diagram

The flow diagram that illustrates the operation of the Selective Repeat ARQ protocol in a noisy channel:



Sender Side:

a. The sender transmits a window of packets to the receiver, starting with sequence number i and ending with sequence number i + N - 1, where N is the window size.

b. The sender sets a timer for each packet in the window.

c. The sender waits for an acknowledgment (ACK) from the receiver.

Receiver Side:

a. The receiver receives the packets and checks for errors.

b. If a packet is received correctly and is in order, the receiver sends an ACK back to the sender with the sequence number of the next expected packet.

c. If a packet is received with errors or is out of order, the receiver discards the packet and sends a negative acknowledgment (NAK) to the sender with the sequence number of the packet that needs to be retransmitted.

d. The receiver buffers out-of-order packets and sends an ACK for the last in-order packet it has received.

Sender Side (in case of no ACK received):

1. If the sender does not receive an ACK before the timer for a packet expires, the sender retransmits only that packet.
2. The sender resets the timer for the retransmitted packet.
3. The sender waits for an ACK from the receiver.

Sender Side (in case of NAK received):

1. If the sender receives a NAK from the receiver, the sender retransmits only the packets that were not correctly received.
2. The sender resets the timer for each packet that was retransmitted.
3. The sender waits for an ACK from the receiver.

The above steps are repeated until all packets have been successfully received by the receiver. The Selective Repeat ARQ protocol provides a reliable mechanism for transmitting data over a noisy channel while minimizing the number of retransmissions required. It retransmits only the packets that were not correctly received, and buffers out-of-order packets to reduce the number of retransmissions required.

CONCLUSION:

A. Stop-and-Wait ARQ:

- The simplest of the three protocols.
- The sender transmits one packet at a time and waits for an acknowledgment (ACK) from the receiver before sending the next packet.
- If the ACK is not received within a certain time, the sender retransmits the packet.
- Suitable for channels with low error rates, low data rates, and short transmission distances.

B. Go-Back-N ARQ:

- The sender transmits a window of packets to the receiver.
- If the receiver detects an error in a packet, it sends a negative acknowledgment (NAK) to the sender requesting retransmission of that packet, as well as all subsequent packets in the window.
- The sender retransmits the entire window of packets that were not correctly received.
- Suitable for channels with moderate to high error rates and moderate data rates.

C. Selective Repeat ARQ:

- The sender transmits a window of packets to the receiver.
- If the receiver detects an error in a packet, it sends a NAK to the sender requesting retransmission of that packet only.
- The sender retransmits only the packets that were not correctly received.
- The receiver buffers out-of-order packets to reduce the number of retransmissions required.
- Suitable for channels with moderate to high error rates, high data rates, and long transmission distances.

In summary, Stop-and-Wait ARQ is the simplest and most reliable protocol but not suitable for high data rates. Go-Back-N ARQ is suitable for channels with moderate to high error rates, and Selective Repeat ARQ is suitable for high data rates and long transmission distances. The choice of protocol depends on the characteristics of the communication channel and the requirements of the application.

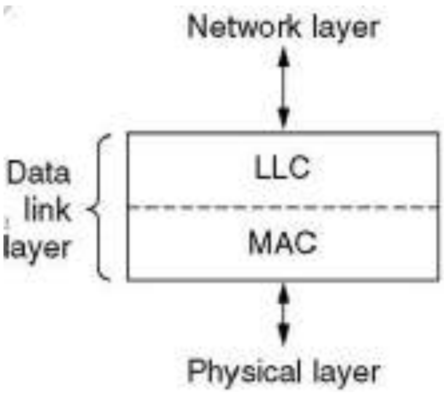
2.4. THE MEDIUM ACCESS SUB LAYER:

In any broadcast network, the key issue is how to determine who gets to use the channel when there is competition for it. To make this point, consider a conference call in which six people, on six different telephones, are all connected so that each one can hear and talk to all the others. It is very likely that when one of them stops speaking, two or more will start talking at once, leading to chaos. In a face-to-face meeting, chaos is avoided by external means. For example, at a meeting, people raise their hands to request permission to speak. When only a single channel is available, it is much harder to determine who should go next. Many protocols for solving the problem are known. In the literature, broadcast channels are sometimes referred to as multi access channels or random access channels.

To coordinate the access to the channel, multiple access protocols are requiring. All these protocols belong to the MAC sub layer. Data Link layer is divided into two sub layers:

1.**Logical Link Control (LLC) -** is responsible for error control & flow control.

2.**Medium Access Control (MAC) -** MAC is responsible for multiple access resolutions



2.4.1. THE CHANNEL ALLOCATION PROBLEM

In broadcast networks, single channel is shared by several stations. This channel can be allocated to only one transmitting user at a time. There are two different methods of channel allocations:

1. **Static Channel Allocation-** a single channel is divided among various users either on the basis of frequency (FDM) or on the basis of time (TDM). In FDM, fixed frequency is assigned to each user, whereas, in TDM, fixed time slot is assigned to each user.
2. **Dynamic Channel Allocation-** no user is assigned fixed frequency or fixed time slot. All users are dynamically assigned frequency or time slot, depending upon the requirements of the user

2.4.2. MULTIPLE ACCESS PROTOCOLS

Many protocols have been defined to handle the access to shared link. These protocols are organized in three different groups:

- Random Access Protocols
- Controlled Access Protocols
- Channelization Protocols

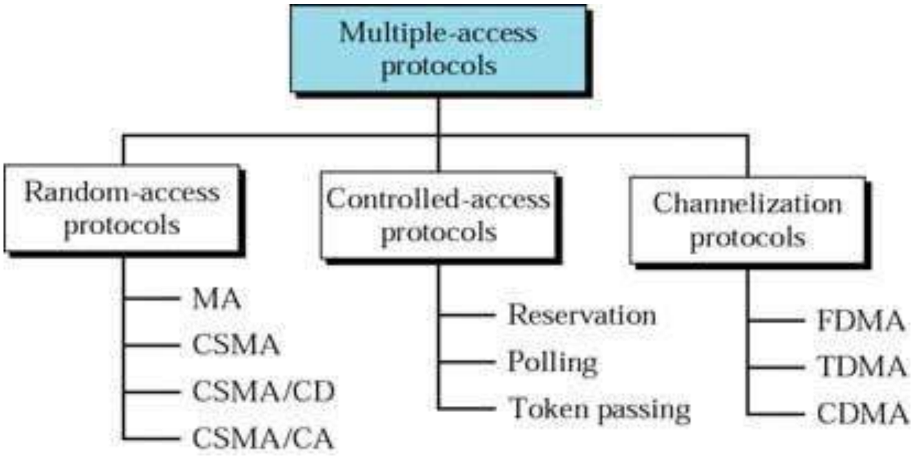


Fig 3.1 types of Multiple acess protocols

2.4.2.1 Random Access Protocols

There is no rule that decides which station should send next. If two stations transmit at the same time, there is collision and the frames are lost. The various random access methods are:

1.ALOHA

2.CSMA (Carrier Sense Multiple Access)

3.CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

4.CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

ALOHA

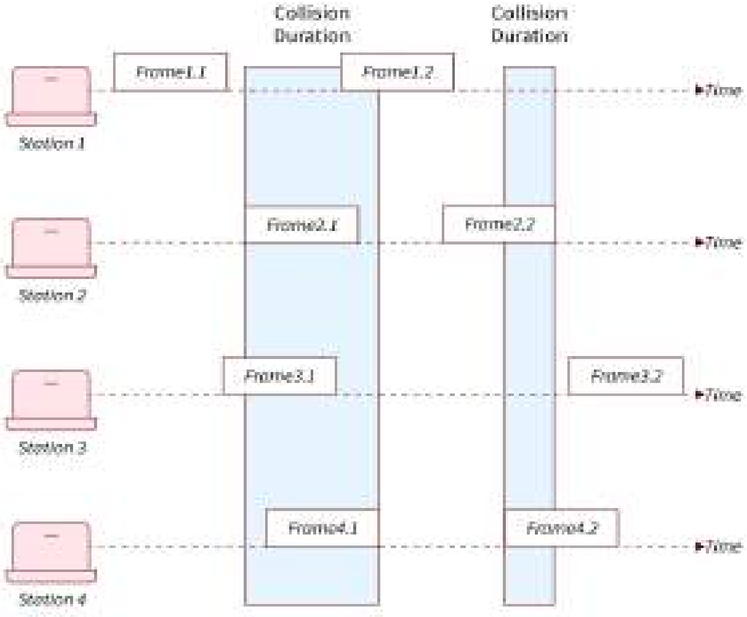
ALOHA is a multiple-access protocol that allows data to be transmitted via a shared network channel. Several data streams from multiple nodes are transmitted across a multi-point transmission channel using this protocol.

Each node or station in ALOHA transmits a frame without attempting to determine whether the transmission channel is idle or busy. If the channel is idle, the frames will be successfully transmitted. If two frames try to occupy the channel at the same time, they will collide and be discarded. These stations may opt to retransmit the corrupted frames until successful transmission occurs. There are two different versions of ALOHA:

- Pure ALOHA
- Slotted ALOHA

Pure ALOHA

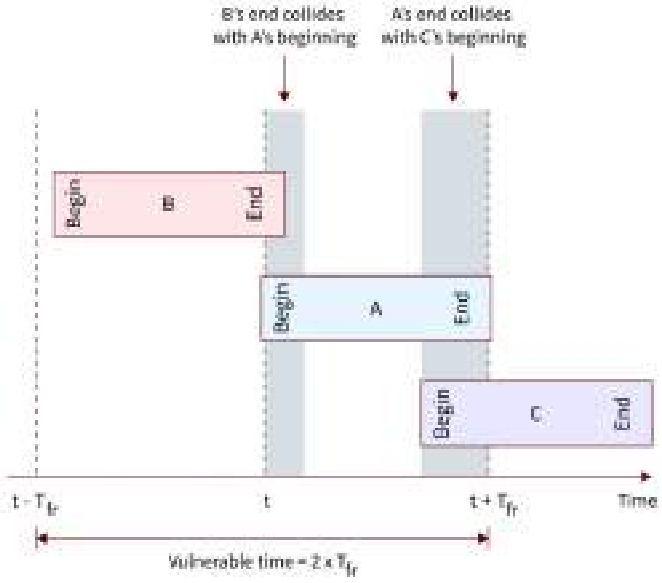
In pure ALOHA, stations transmit frames whenever they have data to send. If each station transmits data to a channel without determining if the channel is idle or not, a collision may occur, and the data frame may be lost. When any station sends a data frame to a channel, the pure Aloha waits for the receiver's acknowledgment. If the receiver's acknowledgment is not received within the given time, the station waits for a random amount of time, known as the **backoff time (Tb)**. Furthermore, the station may believe the frame has been lost or destroyed. As a result, it retransmits the frame until all of the data is properly delivered to the receiver.



Four stations (an unreasonable example) compete with one another for access to the shared channel. The diagram demonstrates that each station sends two frames for a total of eight frames on the shared medium. Due to several frames competing for the same shared channel, some of these frames collide. The picture above reveals that just two frames survive: frame 1.1 from station 1 and frame 3.2 from station 3. Even if one bit from one frame coexists on the channel with one bit from another frame, a collision occurs, and both are destroyed.

Vulnerable Time for Pure Aloha

We assume that the stations send fixed-length frames, which takes **Tt time** (Transmission Time) to transmit the frames completely. The diagram given below shows the vulnerable time for station A. The Vulnerable time for any station must be twice the transmission time because to stop the collision of frames, we must make sure that no other station starts its transmission between Tt time before and Tt time after the transmitting station.


B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
t - T_fr
t
t + T_fr
Time
Vulnerable time = 2 x T_fr

At the time t, Station A sends a frame. Assume station B has previously transmitted a frame between $t - T_t$ and t. As a result, the frames from station A and station B collide. The end of B's frame intersects with the start of A's frame. Assume, on the other hand, that station C sends a frame between t and $t+T_t$. There is a collision between frames from stations A and C here. The start of the C's frame intersects with the conclusion of A's frame.

$$\boldsymbol{V_t=2*T_t}$$

Where,

Vt is a Vulnerable Time for Pure Aloha, ***Tt*** is the Transmission Time of the station.

Note: - The Transmission Time or propagation time is the total time taken by the station to transmit the message completely. It is calculated by:

Transmission Time(*Tt*) = Message length/Bandwidth of the Channel

Throughput of Pure Aloha

Suppose G is the average number of frames created by the system during a single-frame transmission period. Then it may be demonstrated that the average number of successful ALOHA transmissions is given by S.

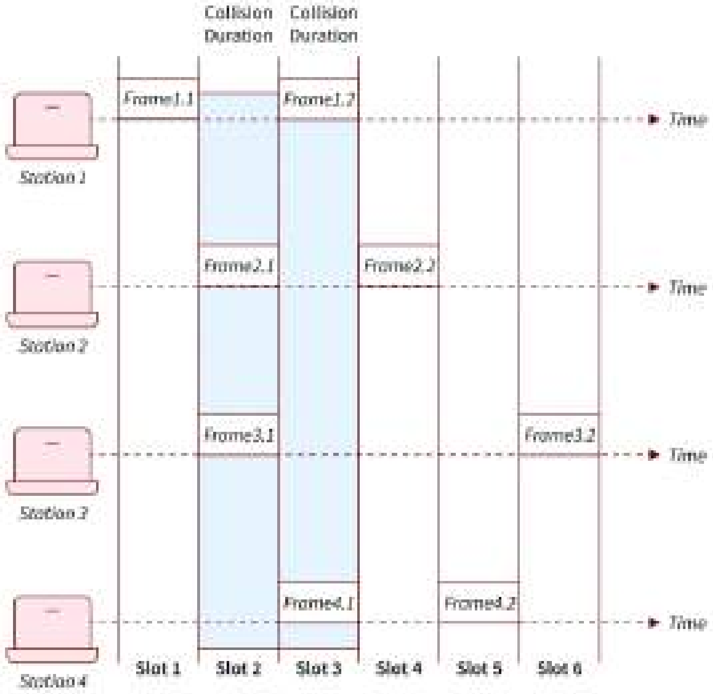
$$S=G*e^{-2G}$$

The value of S will be maxed if G=1/2. So the S max is 0.184. In other words, if one-half a frame is generated during one frame transmission time, 18.4 % of these frames will reach their destination successfully.

The efficiency of Pure Aloha is 18.4%.

Slotted ALOHA

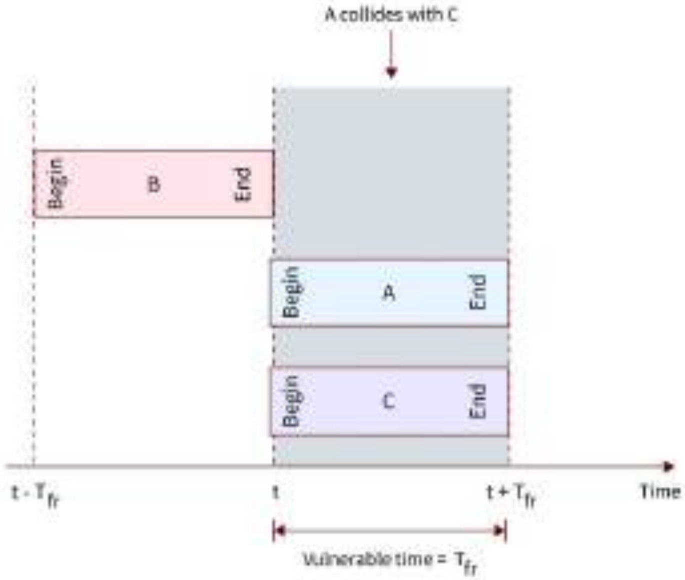
Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA, time of the channel is divided into intervals called slots. The station can send a frame only at the beginning of the slot and only one frame is sent in each slot. If any station is not able to place the frame onto the channel at the beginning of the slot, it has to wait until the next time slot. There is still a possibility of collision if two stations try to send at the beginning of the same time slot.



The figure shows how the channel is divided into slots. A station can start its transmission only at the start of the slot. So the only possible condition for collision is if two or more stations start transmission in the same slot. This condition is shown in Frame 1.2 of Station 1 and Frame 4.1 of Station 4.

Vulnerable Time for Slotted Aloha

The **Vulnerable Time** in the case of Slotted Aloha is equal to the transmission time of the station. It is because we bound the transmission of stations with the slots. The figure given below shows the Vulnerable time for Slotted Aloha.



Where,

Vt is a Vulnerable Time for Slotted Aloha, ***Tt*** is the Transmission Time of the station.

Throughput of Slotted Aloha

Suppose G is the average number of frames created by the system during a single-frame transmission period. Then it may be demonstrated that the average number of successful ALOHA transmissions is given by S.

$$S = G * e^{-G}$$

The value of S will be max if G=1. So the S_{max} is 0.368. In other words, 36.8 percent of these frames will reach their destination successfully if a frame is generated during one frame transmission time. The efficiency of Slotted Aloha is 36.8%.

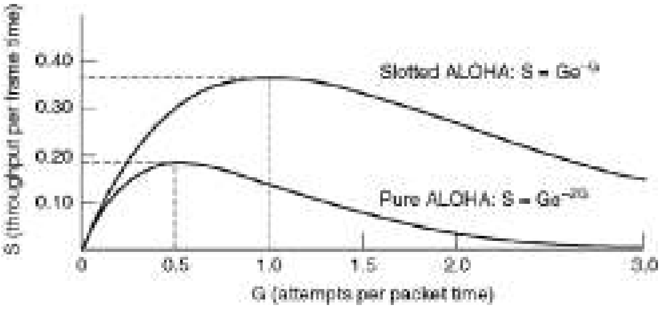


Fig 3.5 Throughput versus offered traffic for ALOHA systems.

Differences between Pure and Slotted ALOHA

Pure Aloha	**Slotted Aloha**
Any station can start transmission at any time.	Any station is allowed to start transmission only at the beginning of the slot.
Time is continuous and not globally synchronized.	Time is discrete and globally synchronized.
Vulnerable Time is equal to twice the transmission time of the station. $Vt=2*Tt$	Vulnerable time is equal to the transmission time and given by: $Vt=Tt$
The average number of successful ALOHA transmissions in the case of Pure Aloha is given by: $S=G*e^{-2G}$	The average number of successful ALOHA transmissions in the case of Slotted Aloha is given by: $S=G*e^{-G}$
The maximum efficiency is 18.4%.	The maximum efficiency is 36.8%
The chance of collision is high.	The chance of collision is relatively less as compared to Pure Aloha.

Carrier Sense Multiple Access (CSMA)

CSMA stands for Carrier Sense Multiple Access (CSMA). CSMA is one of the network protocols which works on the principle of ‘carrier sense’. CSMA is a protocol developed to increase the performance of the network and reduce the chance of collision in the network.

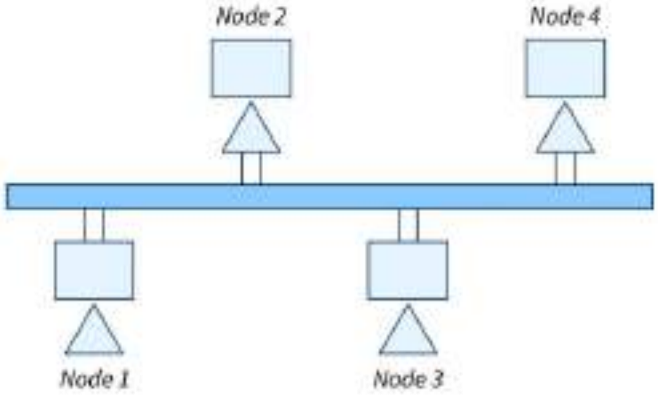
- If any device wants to send data then the device first senses or listens to the network medium to check whether the shared network is free or not. If the channel is found idle then the device will transmit its data.
- This sense reduces the chance of collision in the network but this method is not able to eliminate the collision.
- Carrier Sense Multiple Access (CSMA) is a protocol that senses or listens to the medium before any transmission of data in the medium.
- CSMA is used in Ethernet networks where two or more network devices are connected.

Working Principle of CSMA

- CSMA works on the principle of **"Listen before Talking" or "Sense before Transmit".** When the device on the shared medium wants to transmit a data frame, then the device first detects the channel to check the presence of any carrier signal from other connected devices on the network.

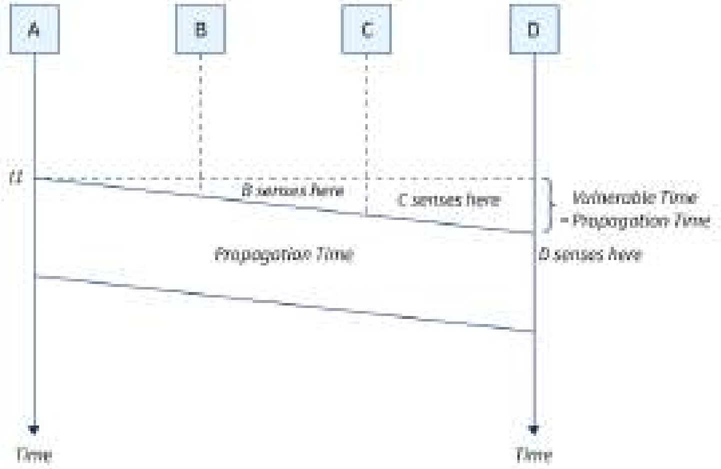
- In this situation, if the device senses any carrier signal on the shared medium, then this means that there is another transmission on the channel. The device will wait until the channel becomes idle and the transmission that is in progress is currently completed.
- When the channel becomes idle the station starts its transmission. All other stations connected in the network receive the transmission of the station.
- In CSMA, the station senses or detects the channel before the transmission of data so it reduces the chance of collision in the transmission.
- But there may be a situation where two stations detected the channel idle at the same time and they both start data transmission simultaneously so in this, there is a chance of collision.
- So CSMA reduces the chance of collision in data transmission but it does not eliminate the collision.

Example: In the network given below in the diagram if node 1 wants to transmit the data in the network then, first of all, it will sense the network if data from any other device is available on the network then it will not send the data. When node 1 finds the channel idle then it will transmit data on the channel.



Vulnerable Time in CSMA

In the CSMA vulnerable time is considered as the propagation time and **Tp** is used to denote it. Generally, it is the time taken by the data frame to reach from one end of the channel to another end. When two stations send the data simultaneously then it will result in a collision in the network. In a situation, if the first bit of the frame sent by the station reaches the end of the shared medium then every station connected in a network hears that bit and every device in the network refrains from sending it.



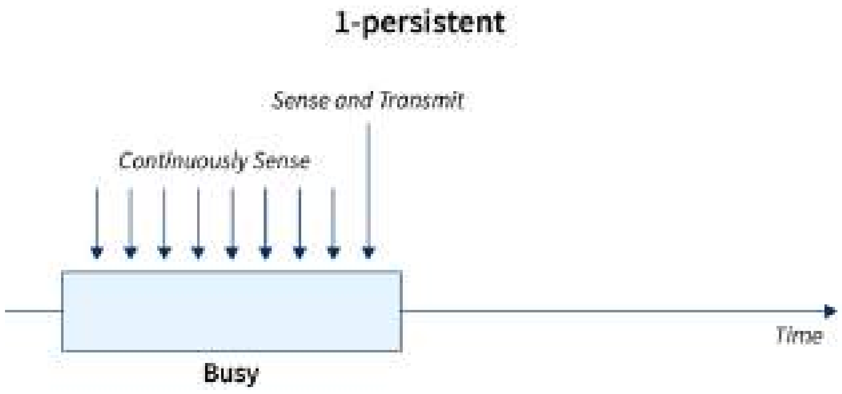
Types of CSMA Access Modes

1-Persistent

This method is considered the straightforward and simplest method of CSMA. In this method, if the station finds the medium idle then the station will immediately send the data frame with 1- probability.

- In this, if the station wants to transmit the data. Then the station first senses the medium.
- If the medium is busy then the station waits until the channel becomes idle. And the station continuously senses the channel until the medium becomes idle.
- If the station detects the channel as idle then the station will immediately send the data frame with 1 probability that’s why the name of this method is 1-persistent.

In this method, once the station finds that the medium is idle then it immediately sends the frame. By using this method there are higher chances for collision because it is possible that two or more stations find the shared medium idle at the same time and then they send their frames immediately.

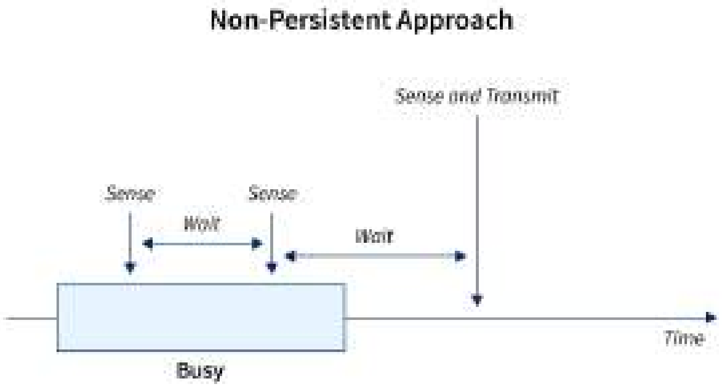


Non-Persistent

In this method of CSMA, if the station finds the channel busy then it will wait for a random amount of time before sensing the channel again.

- If the station wants to transmit the data then first of all it will sense the medium.
- If the medium is idle then the station will immediately send the data.
- Otherwise, if the medium is busy then the station waits for a random amount of time and then again senses the channel after waiting for a random amount of time.
- In P-persistent there is less chance of collision in comparison to the 1-persistent method as this station will not continuously sense the channel but since the channel after waiting for a random amount of time.

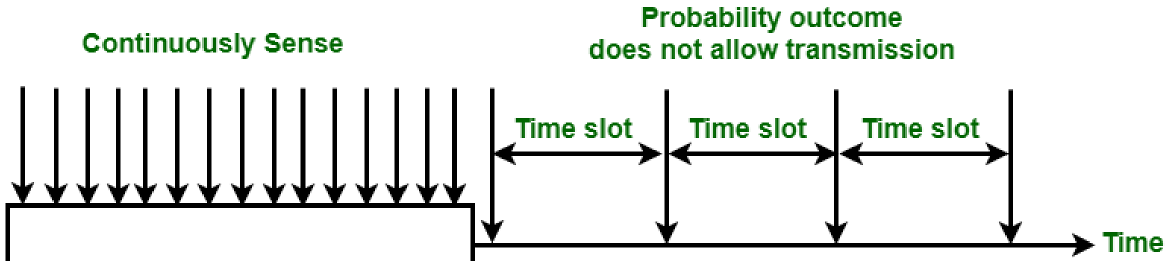
So the random amount of time is unlikely to be the same for two stations that’s why this method reduces the chance of collision.



P-Persistent

The p-persistent method of CSMA is used when the channel is divided into multiple time slots and the duration of time slots is greater than or equal to the maximum propagation time. This method is designed as a combination of the advantages of 1-Persistent and Non-Persistent CSMA. The p-persistent method of CSMA reduces the chance of collision in the network and there is an increment in the efficiency of the network. When any station wants to transmit the data firstly it will sense the channel If the channel is busy then the station continuously senses the channel until the channel becomes idle. If the channel is idle then the station does the following steps.

1. The station transmits its data frame in the network by p probability.
2. The station waits for the start of the next time slot with probability q=1-p and after waiting again senses the channel.
3. If the channel is again idle, then it again performs step 1. If the channel is busy, then it thinks that there is a collision in the network and now this station will follow the back-off procedure.



Variations of CSMA Protocol

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

Carrier sense multiple access/ collision detection is one of the network protocols for transmission. CSMA/CD protocol works with the medium access control layer of the network. That's why the station senses the channel before transmission of data and if the station finds the channel idle then the station transmits its data frames to check whether data transmission is successful in the network or not. If the station successfully the data frame sent then it will again send the next frame. If the station detects a collision in the network, then in CSMA/CD the station will send the stop/jam signal to all the stations connected in the network to terminate their transmission of data. Then the station waits for a random amount of time for the transmission of data.

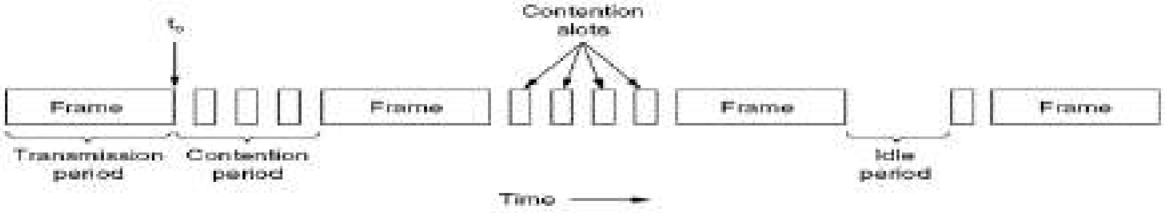


Figure 4-5. CSMA/CD can be in contention, transmission, or idle state.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

Carrier sense multiple access/collision avoidance is one of the network protocols for data frame transmission. When the station sends the data frame on the channel it receives the acknowledgment in response to the sent data frame to test whether the channel is idle or not. When the station receives a single signal i.e. its signal this means that there is no collision and data has been successfully received by the receiver. But in case of collision, the station receives two signals: its signal and the second signal sent by the other station. In CSMA/CA collision is avoided by using the following three strategies. Following are the methods used in the CSMA/ CA to avoid the collision:

- Interframe space
- Contention window
- Acknowledgement

Interframe Space or IFS: If the station wants to transmit the data then it waits until the channel becomes idle and when the channel becomes idle station does not immediately send the data but waits for some time. This period is known as the Interframe Space or IFS. IFS can also define the priority of the frame or station.

Contention window: The contention window is a time that is divided into time slots. When the station is ready for data transmission after waiting for IFS then it chooses the random amount of slots for waiting. After waiting for the random number of slots if the channel is still busy then the station does not initiate the whole process again, the station stops its timer and restarts again when the channel is sensed idle.

Acknowledgement: There may be a chance of collision or data may be corrupted during the transmission. Positive acknowledgment and time-out are used in addition to ensuring that the receiver has successfully received the data.

2.4.2.2. Controlled Access Protocols

In a controlled access protocol, the stations obtain info from each other to seek out what the station has the authority to send. It permits just one node to send at a time, to control the collision of messages on a shared medium. The 3 controlled-access methods are given below:

Reservation:

In the reservation methodology, a station has to create a reservation prior to data.

The schedule has 2 kinds of periods:

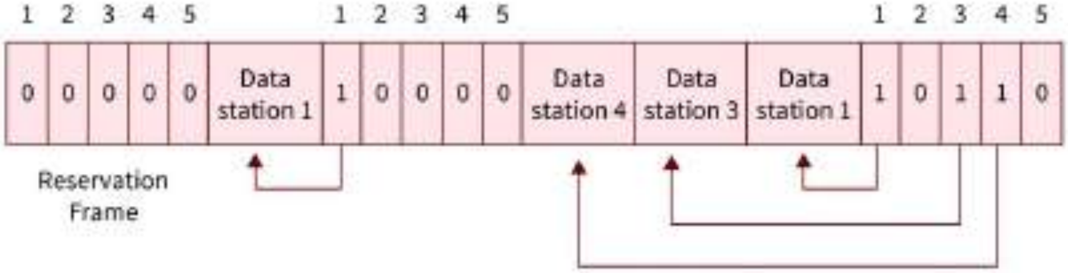
1. Reservation interval of agreed time length

2. Data transmission duration of variable frames.

If there are M stations, the reservation interval is split into M slots, and every station has only one slot.

Assume that station one encompasses a frame to send, it transmits one bit throughout slot one. No alternative station is permitted to transmit throughout this slot.

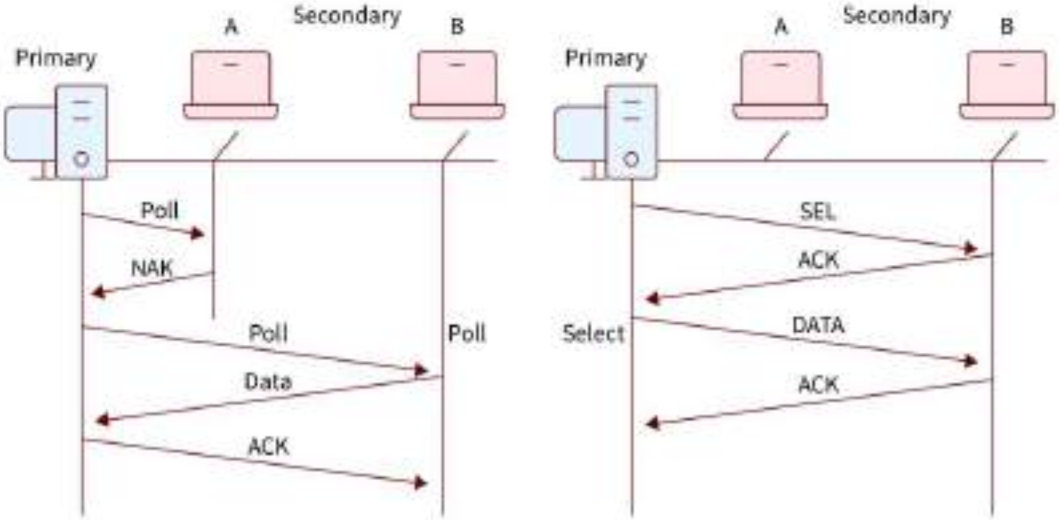
Commonly, i the station could announce that it's a frame to send by inserting one bit into i the slot. Finally, all N slots have been checked, every station is aware of which stations want to transmit.



In this figure a condition with 5 stations and 5 slot reservations available. However in this given first interval only 1,3 and 4 are able to make the reservation and in another second interval, only 1 could make it.

Polling:

The polling method is analogous to the roll call performed in school. rather like the teacher, a controller sends a message to every node successively.



In this, one acts as a primary station(controller) and therefore the others are secondary stations. All data exchanges should be created through the controller.

The message sent by the controller holds the address of the node being hand-picked for granting access.

Although all nodes receive the message however the addressed one responds to that and sends data if any. If no data is available, normally a “poll rejection”(NAK) message is dispatched back.

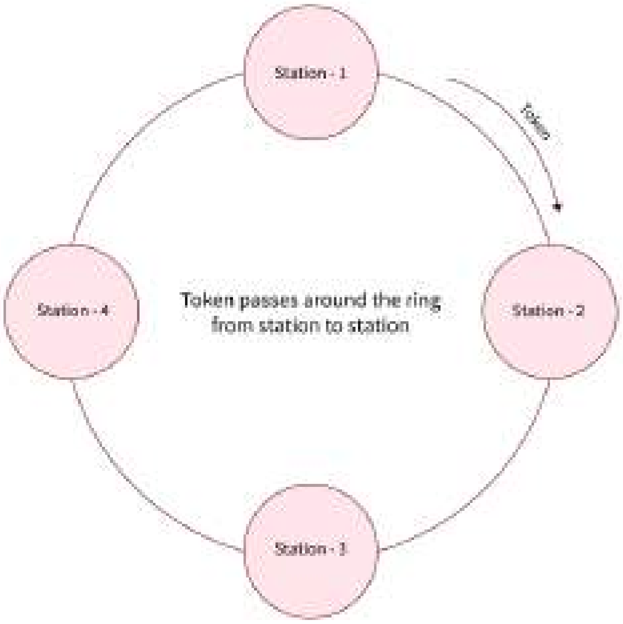
Token Passing:

In the token-passing process, the stations are linked logically to every alternative in the form of a ring, and access to stations is ruled by tokens.

A token could be a special bit pattern or a little message, that flows from 1 station to the consecutive in some predefined order.

In a Token ring, the token is passed from 1 station to a different adjacent station within the ring whereas, in the case of a Token bus, every station uses the bus to send the token to the consecutive station in some predefined order.

Basically, in both the given cases, the token represents permission to send. If a station incorporates a frame queued for transmission once it receives the token, it will redirect that frame before it passes the token to the consecutive station. If it has no queued frame, it passes the token merely.



2.4.2.3. Channelization Protocols

In the channelization protocol, the accessible bandwidth of the link is split in time, frequency, and code to numerous stations to access the channel at the same time.

Frequency Division Multiple Access (FDMA)

The convenient bandwidth is split into equal bands for every station that will be allotted its band. Guard bands are attached in such an order that no 2 bands overlap to avoid debate and noise.

Time Division Multiple Access (TDMA)

In this Time Division of Multiple Access, the bandwidth is divided between multiple stations. To circumvent collision time is split into slots and stations are allocated these slots to transmit data. but there is an overhead of synchronization as every station must grasp its time slot. This is often resolved by adding synchronization bits to every slot. Another issue with TDMA is propagation delay which is determined by the supplementary guard bands.

Code Division Multiple Access (CDMA)

One channel carries every transmission at the same time. There is neither a splitting of bandwidth nor a splitting of time. Let's Assume, that there are plenty of people in a hall all speaking at an identical time, then in addition to that excellent reception of data is feasible if only 2 people speak an identical language. Similarly, data from completely different stations will be transmitted at the same time in numerous code languages.