2nd Sem - Computer Networking with TCP/IP

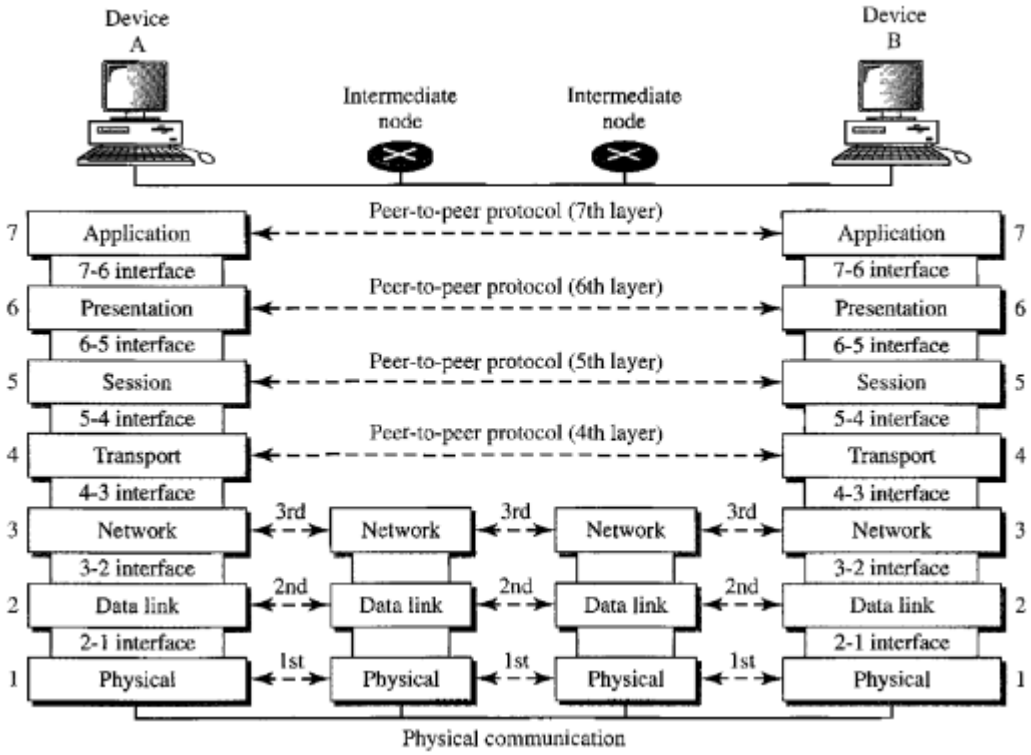
Module-1

#Networking Models:-A network is a combination of hardware and software that sends data from one location to another.
-The hardware consists of the physical equipment that carries signals from one point of the network to another.
-The software consists of instruction sets that make possible the services that we expect from a network.
-There are so many problems around the networking are so, to overcome such problem, the ISO has developed a layered approach.
-In a layered approach, networking concept is divided into several layers, and each layer is assigned a particular task.
-Therefore, we can say that networking tasks depend upon the layers.

***The OSI Model:**-OSI stands for Open Systems Interconnection.
-It has been developed by ISO – 'International Organization for Standardization',
-The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.
-The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems.
-It consists of seven separate but related layers.
-The Seven layers of the OSI model are;

7. **Application**
6. **Presentation**
5. **Session**
4. **Transport**
3. **Network**
2. **Data link**
1. **Physical**

-The OSI model is a collection of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7).



-The above figure shows the layers that involved when a message is sent from device A to device B.

-As the message travels from A to B, it may pass through many intermediate nodes.

-These intermediate nodes usually involve only the first three layers of the OSI model.

-For data to travel from the source to the destination, each layer of the OSI model at the source must communicate with its peer layer at the destination.

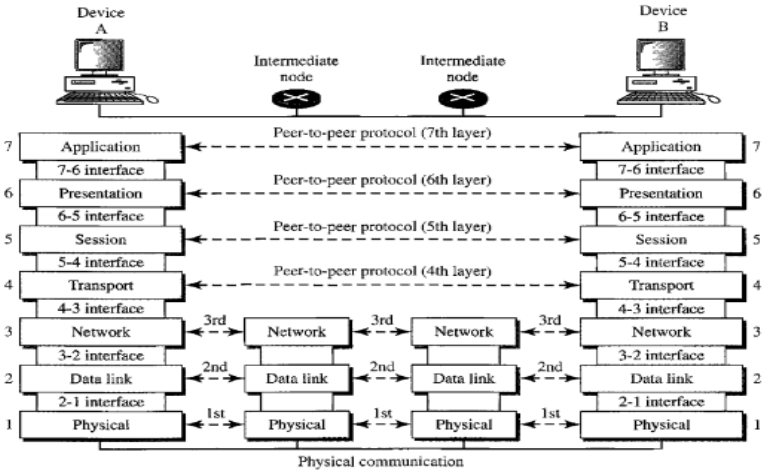
-This form of communication is referred to as peer-to-peer communication.

→Peer-to-Peer Processes:- At the physical layer, communication is direct ,device A sends a stream of bits to device B (through intermediate nodes).

-At the higher layers of device A communication must move down through the layers on device A and go to device B.

-Each layer in the sending device A adds its information to the messages it receives from the layer just above it and passes to the layer just below it.

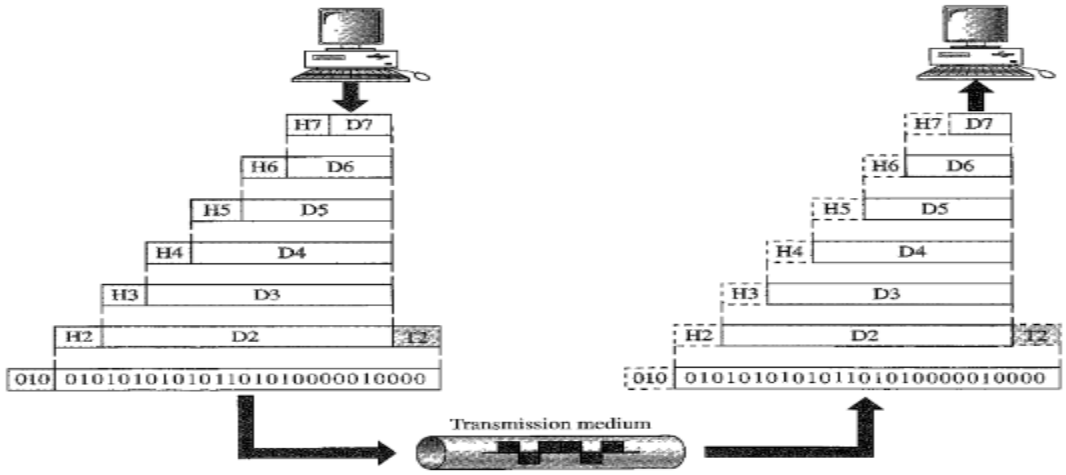
-At layer 1 the entire package is converted to a form that can be transmitted to the receiving device.



-At the receiving machine, the message is unwrapped layer by layer,with each process receiving and removing the data meant for it.

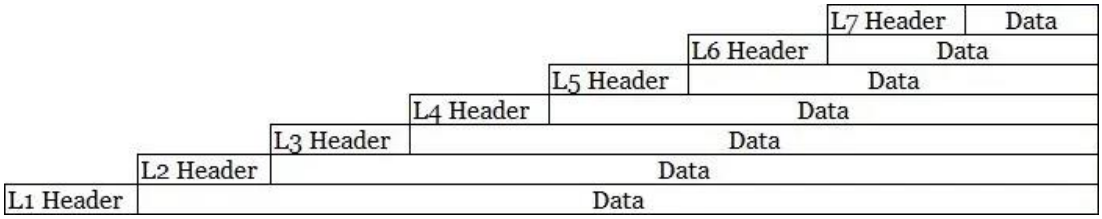
-For example, layer 2 removes the data meant for it, then passes the rest to layer 3.
-Layer 3 then removes the data meant for it and passes the rest to layer 4, and so on.

→overall view of the OSI layers:-In the below Figure gives an overall view of the OSI layers.
-D7 means the data unit at layer 7, D6 means the data unit at layer 6, and so on.
-The process starts at layer7 (the application layer), then moves from layer to layer in descending, sequential order.
-At each layer, a header, or possibly a trailer, can be added to the data unit.
-Commonly, the trailer is added only at layer 2.
-When the data unit passes through the physical layer (layer 1), it is changed into an electromagnetic signal and transported through a transmission medium9.



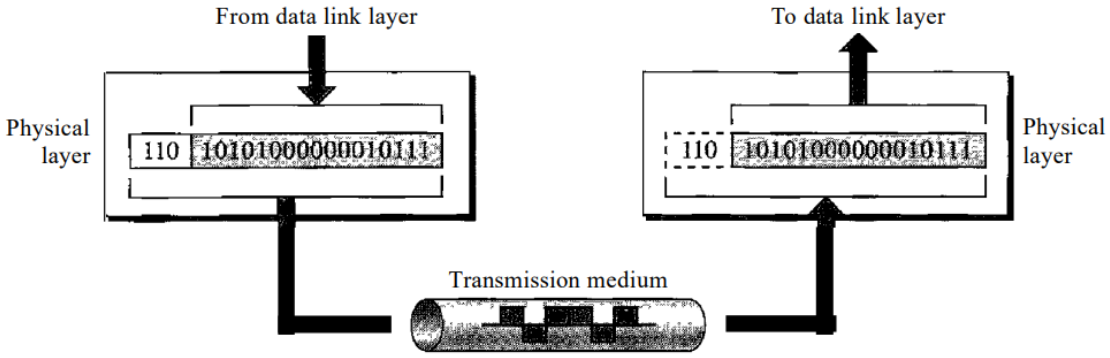
-When it reaches its destination, the signal passes into layer 1 and is transformed back into digital form.
-The data units then move back up through the OSI layers.
-As each block of data reaches the next higher layer, the headers and trailers attached to it at
the corresponding sending layer are removed, and appropriate data to that layer are taken.

→Encapsulation:-The term encapsulation is used to describe a process of adding headers and trailers around some data.
-A packet (header and data) at level 7 is encapsulated in a packet at level 6. The whole packet at level 6 is encapsulated in a packet at level 5, and so on.
-The lower layer encapsulates the higher layer's data between a header.



*Layers in the OSI Model (7 layers)

1. **Physical Layer :**-The lowest layer of the OSI reference model is the physical layer.
   -It is responsible for the actual physical connection between the devices.
   -The physical layer contains information in the form of bits.
   -It is responsible for transmitting individual bits from one node to the next.
   -Or The main functionality of the physical layer is to transmit the individual bits from one node to another node.
   -When receiving data, this layer will get the signal received and convert it into 0s and 1s and send them to the Data Link layer.

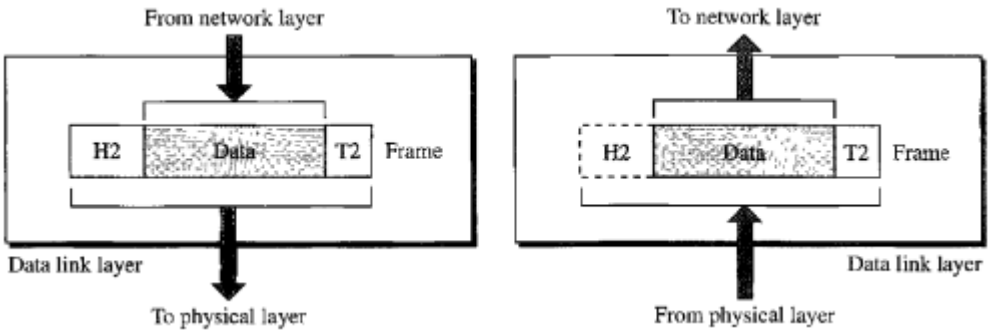


   -The transmission media defined by the physical layer include metallic cable, optical fiber, and the wireless radio-wave.
   -Encoding and decoding are done in the physical layer.

   >Functions of a Physical layer:

   - **Line Configuration**: It defines the way how two or more devices can be connected physically.
   - **Data Transmission:** It defines the transmission mode whether it is simplex, half-duplex or full-duplex mode between the two devices on the network.
   - **Topology:** It defines the way how network devices are arranged.
   - **Signals:** It determines the type of the signal used for transmitting the information.

2. **Data Link Layer :**-The data link layer is responsible for the node-to-node delivery of the message. (if we post a letter it passes through a lot o post office (node) to get the receiver)
   -The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer.
   -It defines the format of the data on the network.
   -It provides reliable and efficient communication between two or more devices.



-head portion is used to identify the local address (local address is like a pet name).
-header has 2 parts. The 1st part has the sender address and the 2nd part has the receiver address.

>Other responsibilities of the data link layer/The Functions of the Data Link Layer are;

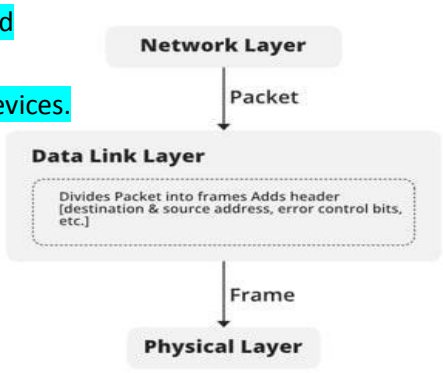
- **Framing:** The data link layer translates the physical's raw bit stream into packets known as Frames.
  -The Data link layer adds the header and trailer to the frame.
  - The header which is added to the frame contains the hardware destination and source address.

  -Header portion is used to identify the local address (local address is like a pet name).
  -Header has 2 parts. The 1st part has the sender address and the 2nd part has the receiver address.
- **Physical addressing:** After creating frames, the Data link layer adds physical addresses (MAC addresses) of the sender and/or receiver in the header of each frame.
- **Flow Control:** It is the technique through which the constant data rate is maintained on both the sides so that no data get corrupted.
  Or
  -The data link layer applies a flow control mechanism to prevent overloading the receiver if the rate at which data are produced in the sender is slower than the rate at which data is used by the receiver.
- **Error Control:** Error control is achieved by adding a calculated value CRC (Cyclic Redundancy Check) that is placed to the Data link layer's trailer which is added to the message frame before it is sent to the physical layer.

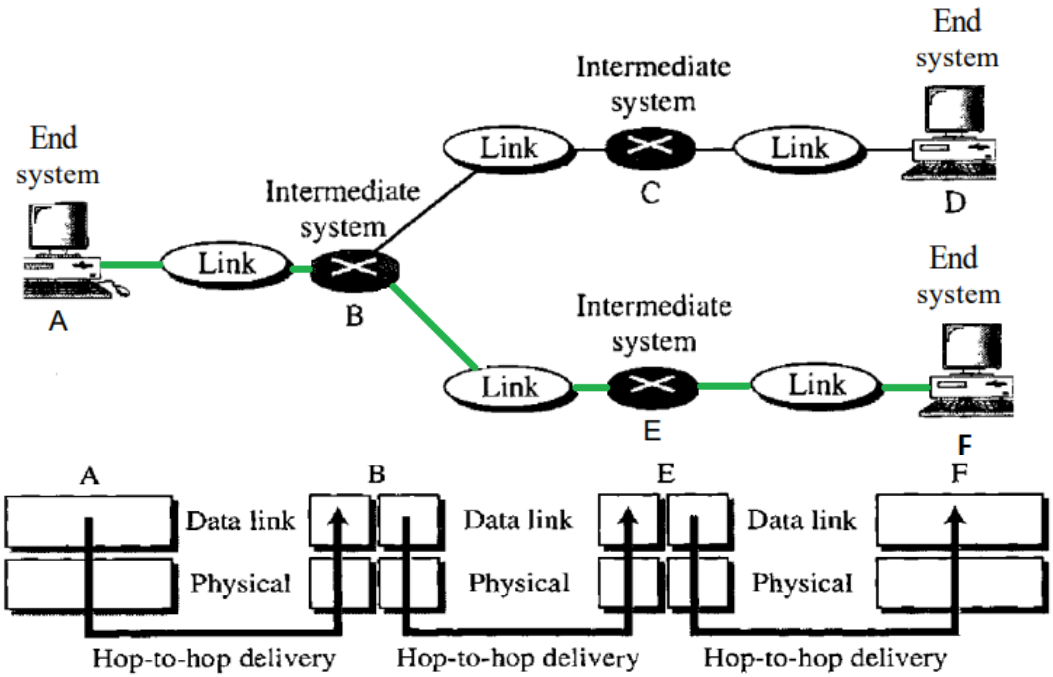
-If any error seems to occurr, then the receiver sends the acknowledgment for the retransmission of the corrupted frames.

- **Access Control:** When two or more devices are connected to the same communication channel, then the data link layer protocols are used to determine which device has control over the link at a given time.

-Packet in the Data Link layer is referred to as Frame.
-Switch & Bridge are Data Link Layer devices.



-Example of hop-to-hop (node-to-node) delivery of data link layer

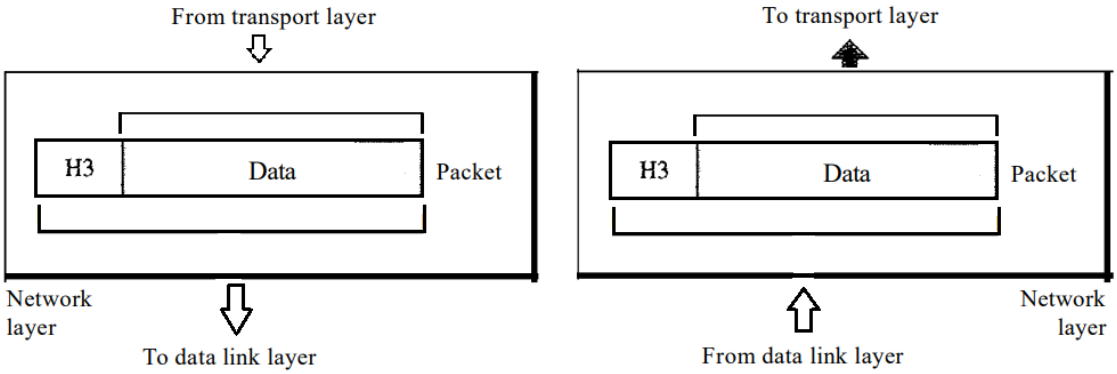


-To send data from A to F, three partial deliveries are made.
- First, the data link layer at A sends a frame to the data link layer at B (a router). -Second, the data link layer at B sends a new frame to the data link layer at E.
-Finally, the data link layer at E sends a new frame to the data link layer at F.
-Note that the frames that are exchanged between the three nodes have different values in the headers.
-The frame from A to B has B as the destination address and A as the source address.
-The frame from B to E has E as the destination address and B as the source address.

-The frame from E to F has F as the destination address and E as the source address.
-The values of the trailers can also be different if error checking includes the header of the frame.

3. **Network Layer :**-The network layer is responsible for the source-to-destination delivery of a packet.
   -It determines the best path to move data from source to the destination based on the network conditions, the priority of service, and other factors.
   -If the two devices communicating are on the same network, then the network layer is unnecessary.
   -The network layer breaks up segments from the transport layer into smaller units, called **packets**, on the sender's device, and reassembling these packets on the receiving device.

   -The network layer also finds the best physical path for the data to reach its destination; this is known as **routing.**

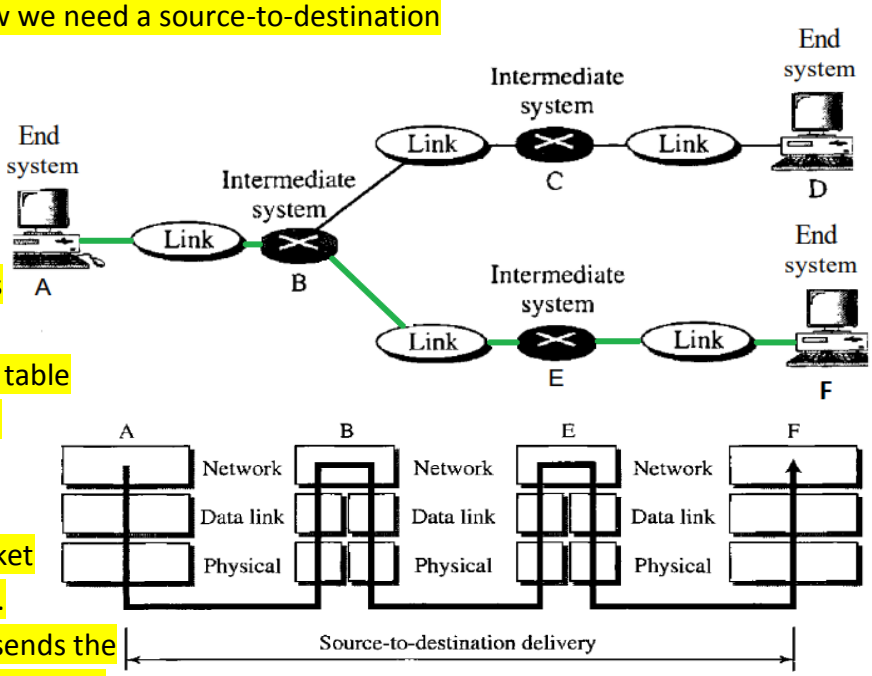


>**Other responsibilities of the network layer/functions are;**

- **Logical Addressing**: When packet is sent outside the network, network layer adds Logical address of the sender & receiver to each packet.
  -Such address determines each device uniquely.
- **Routing**: Routing is the major component of the network layer, and it determines the best optimal path out of the multiple paths from source to the destination.

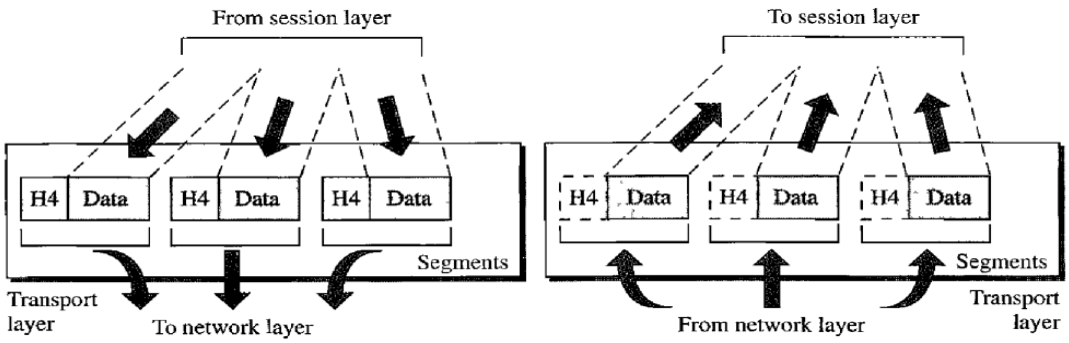
-Example end-to-end delivery by the network layer.

-As the figure shows, now we need a source-to-destination delivery.
-The network layer at A sends the packet to the network layer at B.
-When the packet arrives at router B.
-router B uses its routing table to find that the next hop is router E.
-The network layer at B, therefore, sends the packet to the network layer at E.
-The network layer at E sends the packet to the network layer at F.



4. **Transport Layer :**-The transport layer is responsible for process-to-process delivery of the entire message.
-The Transport layer ensures that messages are transmitted in the order in which they are sent and there is no duplication of data.
-The main responsibility of the transport layer is to transfer the data completely.
-It receives the data from the upper layer and converts them into smaller units known as segments.
-Or It takes data from the session layer and breaking it up into chunks called segments before sending it to layer 3.
-The transport layer on the receiving device is responsible for reassembling the segments into data the session layer can consume.

-The transport layer ensures that the whole message arrived complete and organised
-The transport layer is also responsible for **flow control and error control.**
-Flow control determines an optimal speed of transmission to ensure that a sender with a fast connection does not overwhelm a receiver with a slow connection.
- The transport layer performs error control on the receiving end by ensuring that the data received is complete, and requesting a retransmission if it isn't.



>Other responsibilities of the transport layer/function are;

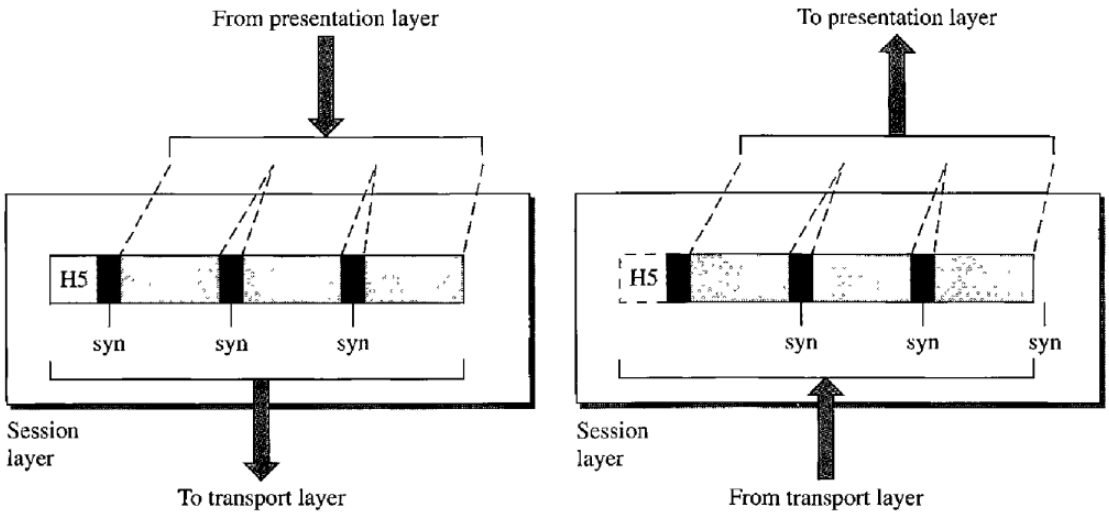
- **Service Point Addressing**: To deliver the message to the correct process, the transport layer header includes a type of address called s**ervice point address or port address.**
  -Thus by specifying this address, the transport layer makes sure that the message is delivered to the correct process.
- **Segmentation and Reassembly:** This layer accepts the message from the (session) layer, and breaks the message into smaller units.
  -Each of the segments produced has a header associated with it.
  -The transport layer at the destination station reassembles the message.
- **Connection control:** Transport layer provides two services **Connection-oriented service and connectionless service.**
  -A connectionless service treats each segment as an individual packet, and they all travel in different routes to reach the destination.
  -A connection-oriented service makes a connection with the transport layer at the destination machine before delivering the packets.
  -In connection-oriented service, all the packets travel in the single route.
- **Flow control:** Like the data link layer, the transport layer is responsible for flow control.
  -However, flow control at this layer is performed end to end rather than across a single link.
- **Error control:** Like the data link layer, the transport layer is responsible for error control.
  -However, error control at this layer is performed process-to-process rather than across a single link.
  -The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication).
  Error correction is usually achieved through retransmission.

5. **Session Layer :**-This layer is responsible for the establishment of connection, maintenance of sessions, and authentication, and also ensures security.

Or

-The Session layer is used to establish, maintain and synchronizes the interaction between communicating devices.
-This is the layer responsible for opening and closing communication between the two devices.
-The time between when the communication is opened and closed is known as the session.
-In opening sessions, it ensures they remain open and functional while data is being transferred, and closes them when communication ends.
-The session layer also synchronizes data transfer with checkpoints.

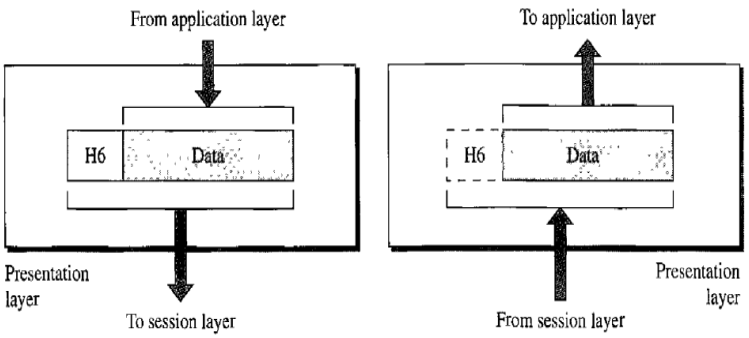
- For example, if a 100 megabyte file is being transferred, the session layer could set a checkpoint every 5 megabytes. In the case of a disconnect or a crash after 52 megabytes have been transferred, the session could be resumed from the last checkpoint, meaning only 50 more megabytes of data need to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



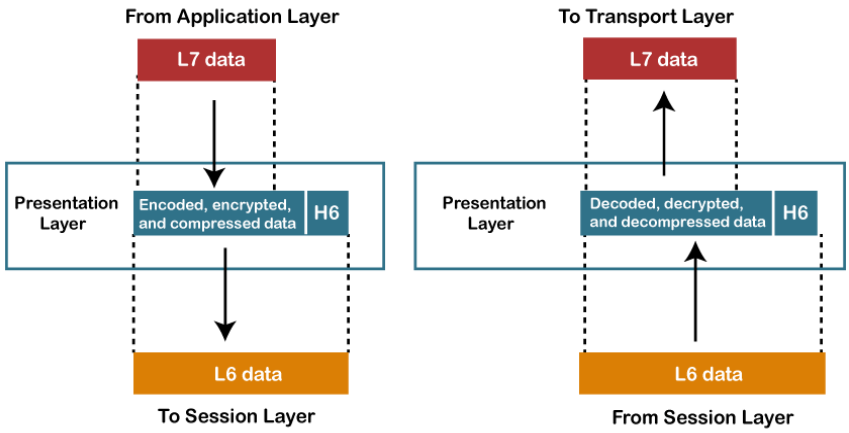
>Other responsibilities of the session layer/functions are;

- **Dialog control:** it allows the communication between two processes which can be either half-duplex or full-duplex.
- **Synchronization**: Session layer adds some checkpoints when transmitting the data in a sequence.
  -If some error occurs in the middle of the transmission of data, then the transmission will take place again from the checkpoint.
  -This process is known as Synchronization and recovery.

6. **Presentation Layer :**-The presentation layer prepares data for the application layer.
   - It defines how two devices should encode, encrypt, and compress data so it is received correctly on the other end.
   -The presentation layer takes any data transmitted by the application layer and prepares it for transmission over the session layer.



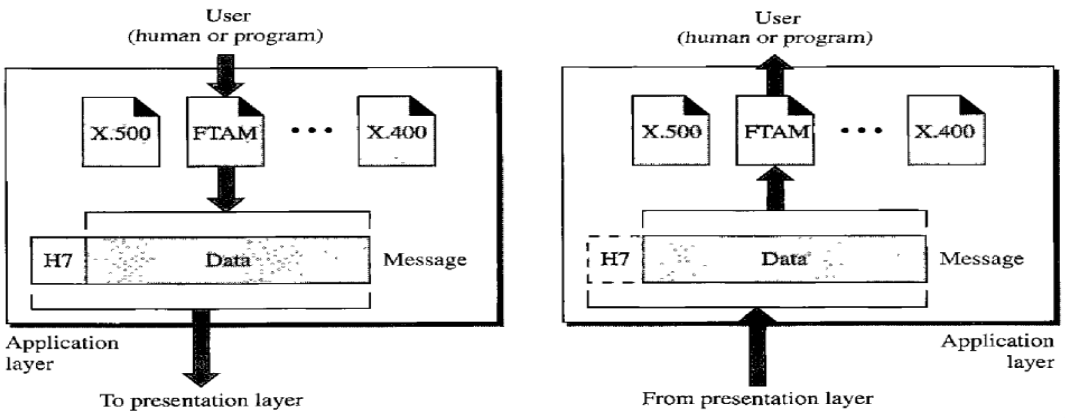
Or



>Other responsibilities of the presentation layer/functions are;

- **Translation:** The processes in two systems exchange the information in the form of character strings, numbers and so on.
  -Different computers use different encoding methods, the presentation layer handles the interoperability between the different encoding methods.
  -It converts the data from sender-dependent format into a common format and changes the common format into receiver-dependent format at the receiving end.
- **Encryption:** Encryption is needed to maintain privacy.
  - Encryption is a process of converting the sender-transmitted information into another form and sends the resulting message over the network.
- **Compression**: It is a process of compressing the data.
  -It shrinks large amount of data into smaller pieces i.e. it reduces the size of data.
  -Data compression is very important in multimedia such as text, audio, video.

7. **Application Layer :**-The application layer enables the user, whether human or software, to access the network.
   -It provides user interfaces and support for services such as electronic mail,remote file access and transfer, shared database management, and other types of distributed information services.
   -These applications produce the data, which has to be transferred over the network.
   -Example: Application – Browsers, Skype Messenger, etc.



>Other services provided by the application layer/function are;

- **File transfer, access, and management (FTAM):** An application layer allows a user to access the files in a remote computer, to retrieve the files from a computer and to manage the files in a remote computer.
- **Mail services:** An application layer provides the facility for email forwarding and storage.

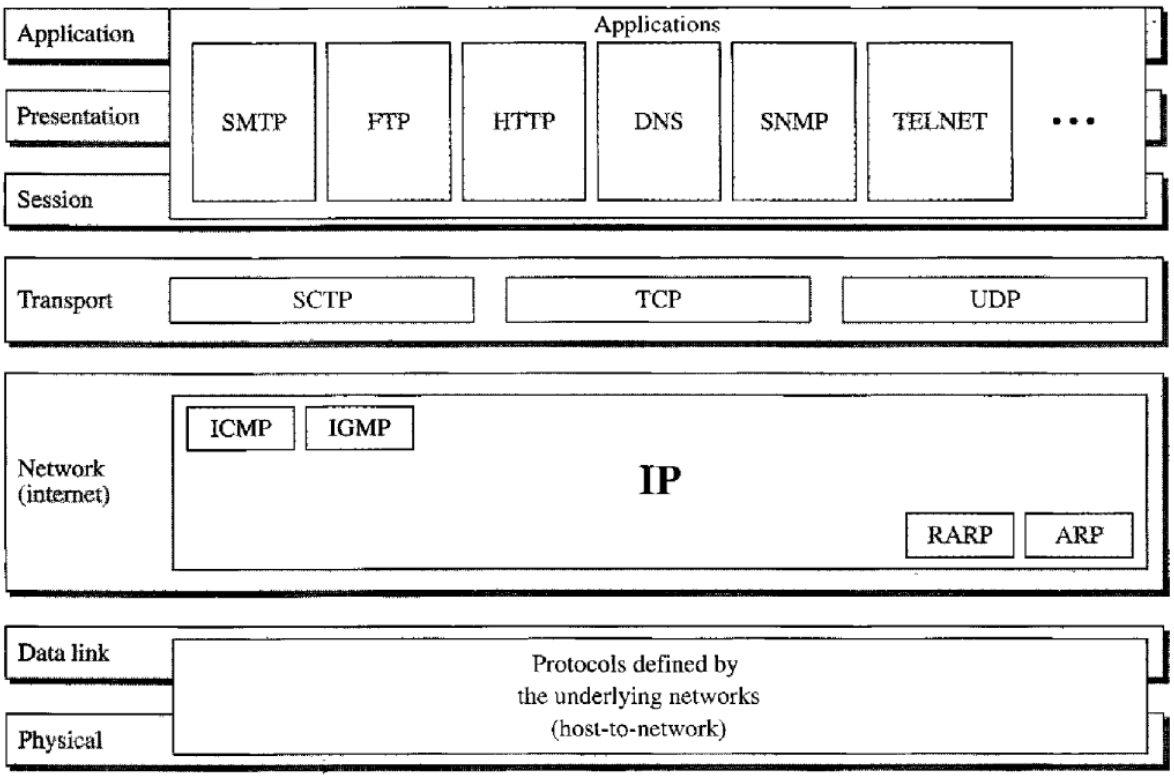
Layer No	Layer Name	Responsibility	Information Form(Data Unit)	Device
7	Application Layer	Helps in identifying the client and synchronizing communication.	Message	–
6	Presentation Layer	Data from the application layer is extracted and manipulated in the required format for transmission.	Message	–
5	Session Layer	Establishes Connection, Maintenance, EnsuresAuthentication, and Ensures security.	Message	Gateway
4	Transport Layer	Take Service from Network Layer and provide it to the Application Layer.	Segment	Firewall
3	Network Layer	Transmission of data from one host to another, located in different networks.	Packet	Router
2	Data Link Layer	Node to Node Delivery of Message.	Frame	Switch, Bridge
1	Physical Layer	Establishing Physical Connections between Devices.	Bits	Hub, Repeater, Modem, Cables

***TCP/IP protocol Suite:-**The TCP/IP protocol suite came before the OSI model.
-Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model.
-TCP/IP was designed and developed by the Department of Defense (DoD) in the 1960s and is based on standard protocols.

-It stands for **Transmission Control Protocol/Internet Protocol**.
-TCP/IP is a suite of protocols used for the communication of devices on a network.
-The network can be of any type: Internet or personal networks like the intranet, extranet, etc.
-The TCP/IP model consists of five layers they are;

- Application layer.
- Transport layer.
- Network layer.
- Data link layer.
- Physical layer.

-The three topmost layers in the OSI model are represented in TCP/IP by a single layer called the application layer.



-At the transport layer, TCP/IP defines three protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP).
-At the network layer, the main protocol defined by TCP/IP is the Internetworking Protocol (IP); there are also some other protocols that support data movement in this layer.

→Layers of TCP/IP are;

1. **Physical and Data Link Layers:**-At the physical and data link layers, TCP/IP does not define any specific protocol.
   -this layer works in the link between different devices in the network.
   -Error prevention and “framing” are also provided by the data-link layer.

2. **Network Layer:**-At the network layer TCP/IP supports the Internetworking Protocol (IP).
   -It also has four supporting protocols they are; ARP, RARP, ICMP, and IGMP.
   -IP takes care of the destination and host addresses and makes sure the connection is maintained.
   -The network layer also called the internet layer.
   -ARP stands for Address Resolution Protocol.
   ARP is a network layer protocol which is used to find the physical address from the IP address
   -The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address.
   -It is used when a computer is connected to a network for the first time.
   - Internet Control Message Protocol (ICMP) reports errors in case the connection is not proper.
   -The Internet Group Message Protocol (IGMP) is used to clear the way to send messages simultaneous to a group of receivers.

3. **Transport Layer:**- The transfer of data is done in this layer. It is responsible for maintaining the communication between the sender and receiver.
   -UDP and TCP are transport level protocols responsible for delivery of a message from a process (running program) to another process.
   -A new transport layer protocol, SCTP, has been devised to meet the needs of some newer applications.
   -User Datagram Protocol (UDP) provides connectionless service and end-to-end delivery of transmission.
   - it discovers the errors but not specify the error.
   -Transmission Control Protocol (TCP) creates a virtual circuit between the sender and receiver, and it is active for the duration of the transmission.
   -TCP is a reliable protocol as it detects the error and retransmits the damaged frames.
   -At the sending end, TCP divides the whole message into smaller units known as segment.
   -And each segment contains a sequence number which is required for reordering the frames to form an original message.
   -At the receiving end, TCP collects all the segments and reorders them based on sequence numbers.
   -The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet.
   - It is a transport layer protocol that combines the best features of UDP and TCP.

4. **Application Layer:**-An application layer is the topmost layer in the TCP/IP model.
   - The application layer in TCP/IP is equivalent to the combination of session, presentation, and application layers in the OSI model
   -It is responsible for end-to-end communication and error-free delivery of data.
   -This layer allows the user to interact with the application.
   -When one application layer protocol wants to communicate with another application layer, it forwards its data to the transport layer and so on.

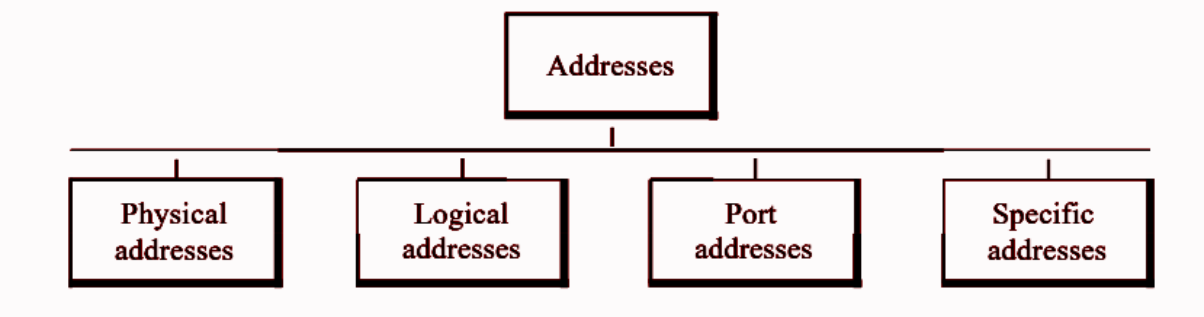
   >Following are the main protocols used in the application layer:

   - **HTTP:** It stands for Hypertext transfer protocol.
     -This protocol allows us to access the data over the world wide web.
     -It transfers the data in the form of plain text, audio, video.
   - **SMTP:** It stands for Simple mail transfer protocol. The TCP/IP protocol that supports the e-mail is known as a Simple mail transfer protocol.
     -This protocol is used to send the data to another e-mail address.
   - **FTP:** It stands for File Transfer Protocol.
     -FTP is a standard internet protocol used for transmitting the files from one computer to another computer.
     -bakki unde google ill nokkuvaa

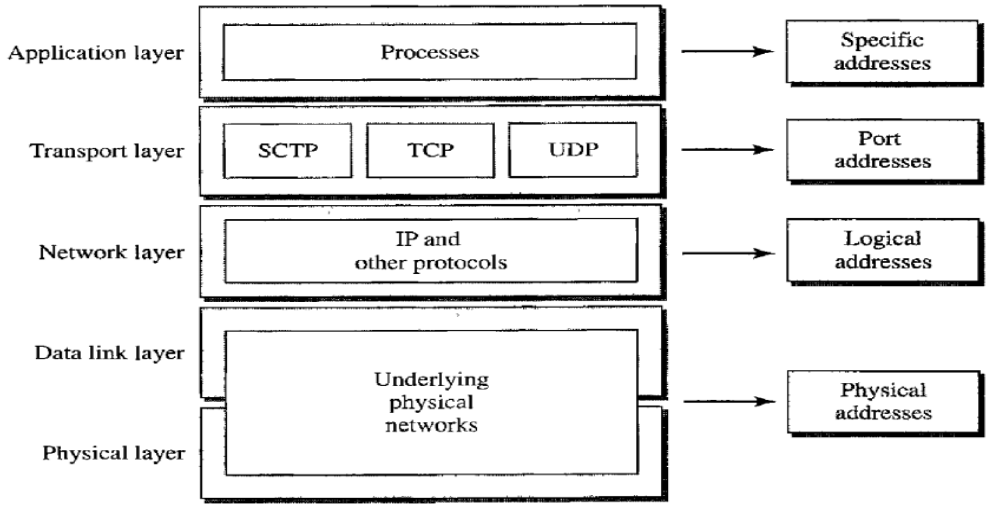
*Comparison of the OSI and TCP/IP Models

OSI Model	TCP/IP Model
OSI stands for Open Systems Interconnection.	TCP/IP stands for Transmission Control Protocol/Internet Protocol.
It has 7 layers.	It has 4 layers.
It is low in usage.	It is mostly used.
It is vertically approached.	It is horizontally approached.
Delivery of the package is guaranteed in OSI Model.	Delivery of the package is not guaranteed in TCP/IP Model.
Replacement of tools and changes can easily be done in this model.	Replacing the tools is not easy as it is in OSI Model.
It is less reliable than TCP/IP Model.	It is more reliable than OSI Model.

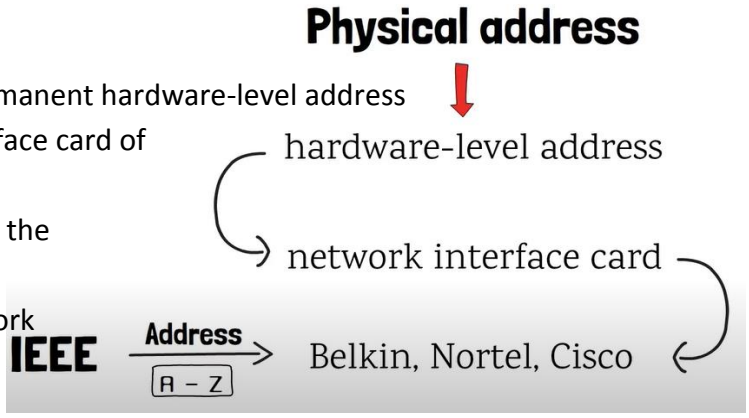
***ADDRESSING:**-Four levels of addresses are used in the TCP/IP protocols.



-Each address is related to a specific layer in the TCP/IP architecture ,fig in below.



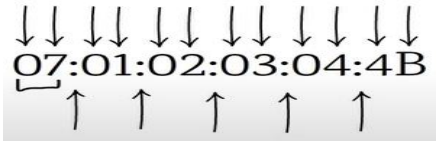
1. **Physical Addresses :**-The physical address also known as the link address/LAN address /MAC address.



-It is the lowest-level address.
-The physical address is the permanent hardware-level address embedded in the network interface card of a device by its manufacturer.
-IEEE gives a block of address to the manufacturer.
-The manufacturer of the Network interface card (NIC) takes an address from the address pool and embeds a unique physical address.
-Most Ethernet uses a 48-bit physical address written in the form of 12 hexadecimal numbers ,where each byte is separated by colons.

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address.



-For Example:Consider 4 computer are connected to the bus topology local area network (LAN).

- suppose there physical address as a two digit number .

-Host A is the sender with physical address 10 and Host P is the receiver with physical address 87.

-With LAN connection physical addresses are enough to transfer a data from Host A to Host P.

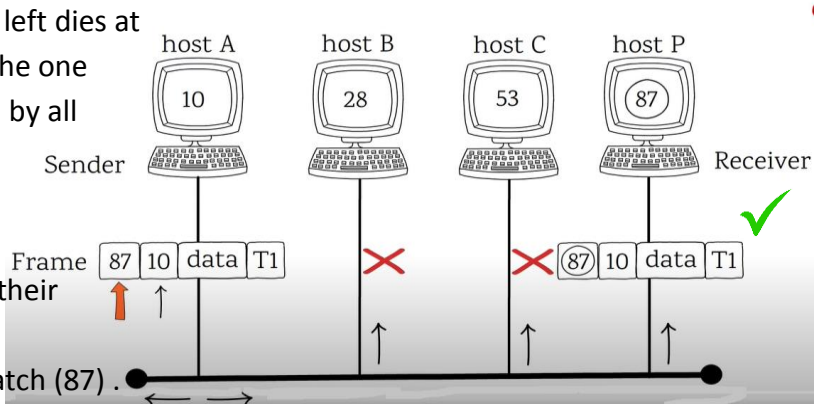
-The data from Host A have destination physical address ,source physical address and a tailor for error detection.

-Note that in a frame ,the destination physical address comes before the source physical address.

-The transmitted frame propagates in both directions.

-when the frame move to the left dies at the cable termination ,while the one moved to the right is received by all connected hosts.

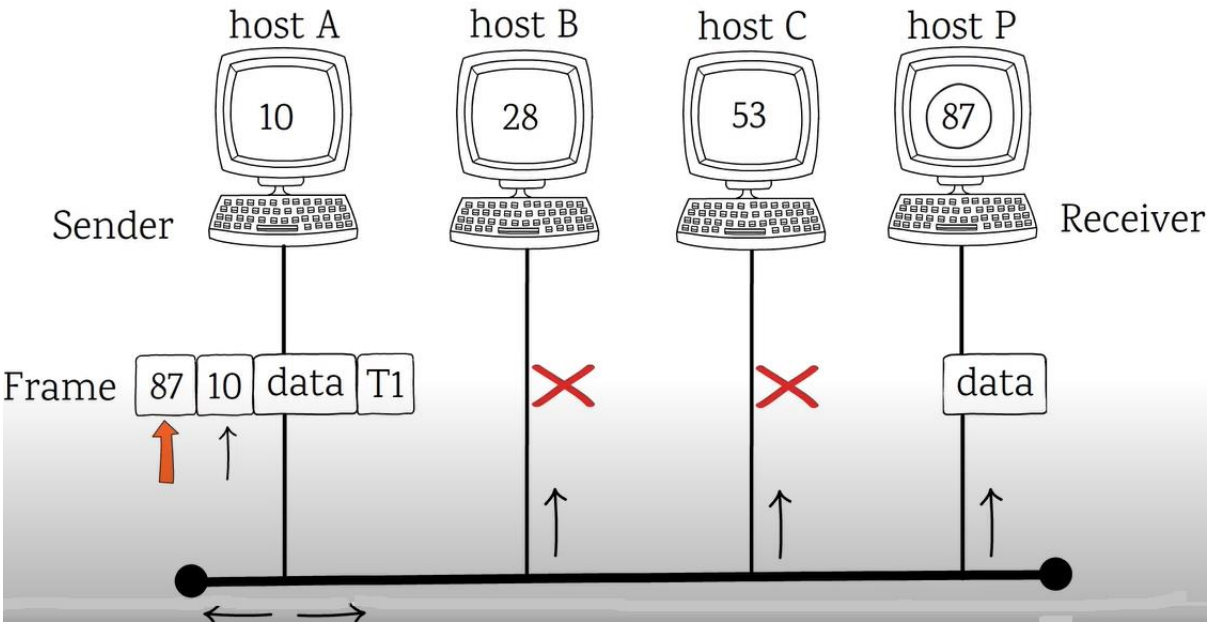
-Host B and C drop the frame because the destination address does not match with their physical addresses.



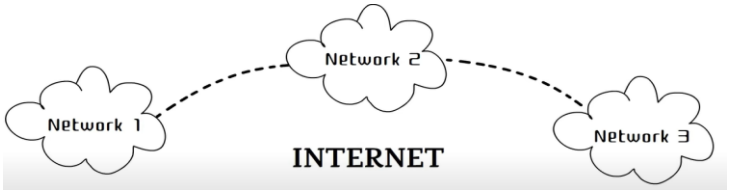
-however ,Host P finds the match (87) .

-So,it drops the header and trailer from the frame and delivers data to the upper layers.

-Therefore, the physical address helps to deliver data to the correct destination devices.



2. **Logical Addresses :**-Multiple networks link to each other to form an internetwork or the internet.



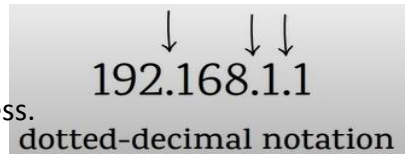
-On the internet ,the devices are identified with an address called logical address.

-It is a 32 bit address written in the form of decimal numbers ,separated with doys.

-It is called the dotted-decimal notation.

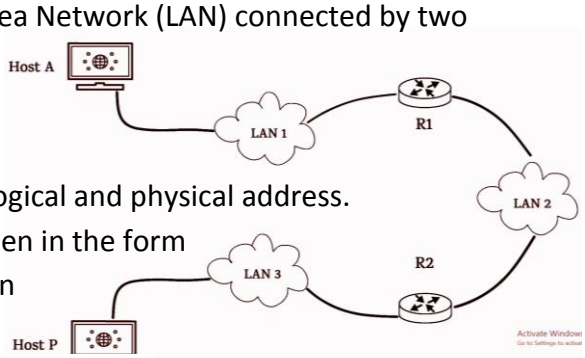
-Decimal number range from 0-255.

-There is no devices with same logical address.



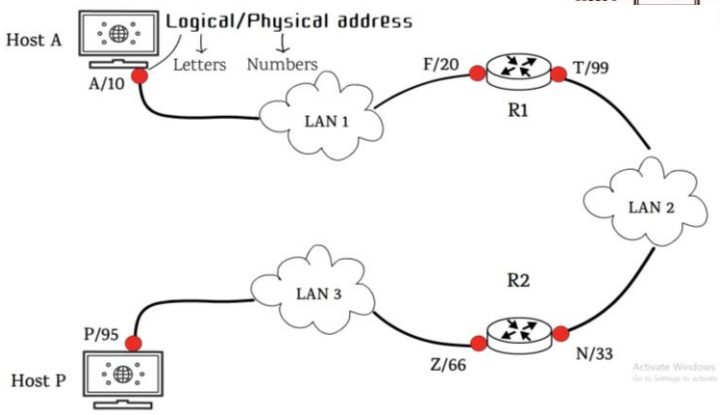
-Example: Consider a network of 3 Local Area Network (LAN) connected by two routers.

-Host A is present in LAN 1 and P is present inLAN 3.



-Each interface of the device has a pair of logical and physical address.

-In this example the logical address is written in the form of letters,and the physical address is written as numbers.



-Now the Host A needs to send data to Host P .

-Here data will move from one network to another so the sender and receiver's logical address is A and P.

-It will encapsulated with the data to form a packet in the network layer.

Packet | A | P | data | T1 |

-Note that the logical source address is written before the logical destination address.

-Since the destination Host P is present in different LAN .

-So,the data first must direct to router 1.

-The network layer finds the logical address of the router 1 using the routing table.

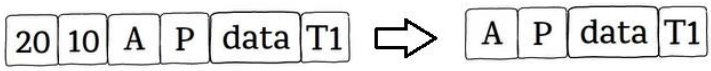
-Address Resolution protocol gives its physical address 20.

-Now the network layer passes this address to the data link layer which encapsulates the packet with destination physical address 20 and the source physical address 10 to the fram

Packet | A | P | data | T1 | ⇨ | 20 | 10 | A | P | data | T1 |

-All devices in LAN 1 receives this frame but only router 1 accepts it and others drop it.

-Router 1 decapsulates the frame and read the logical address destination address P.



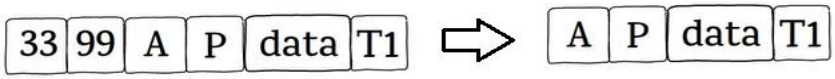
-Since it is different from its logical address,the router 1 knows the packet needs to be forwarded.

-The router finds the logical address of the next-hop that is Router 2,with its router table.

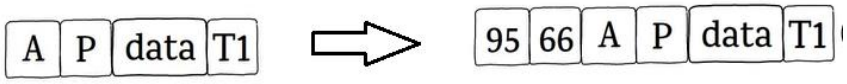
-As the Address Resolution Protocol provides the physical address of router 2 (33).

-Frame with physical destination address 33 and physical source address 99 is created and transmitted in LAN 2.

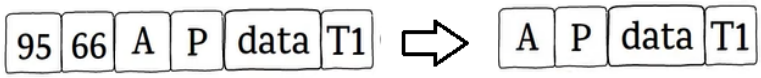
-Router 2 accepts the frame ,decapsulates it and check the logical destination address P and repeat the same scenario.



-Finally the frame with physical address destination address 95 and physical source address 66 is transmitted .



-Host P accepts the frame ,decapsulates it.

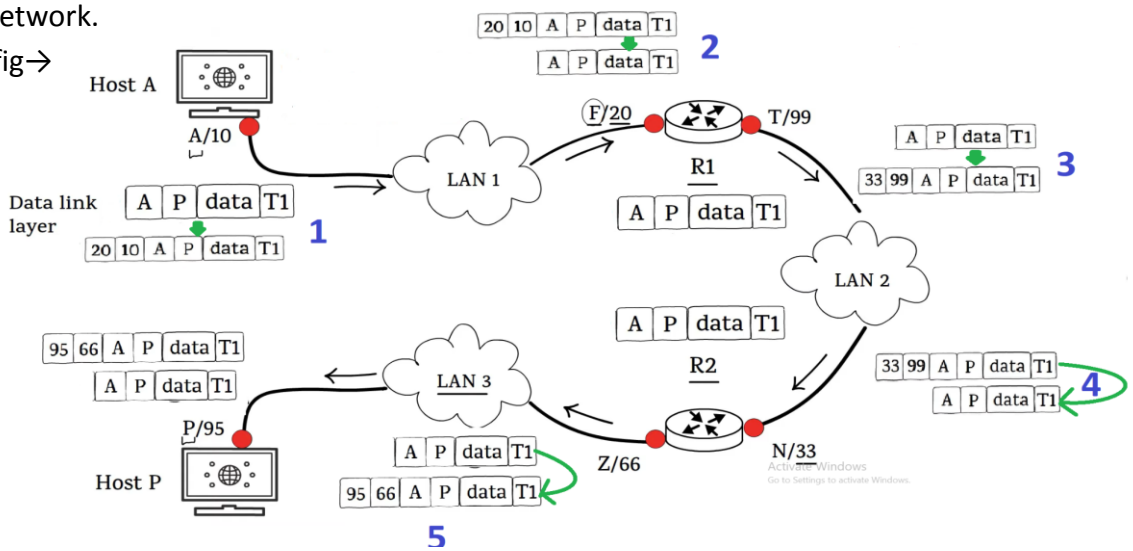


-And sends data to the upper layers.

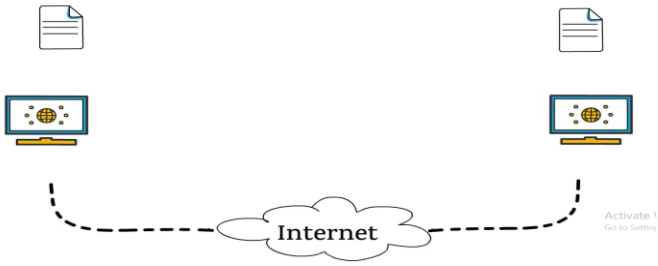
-Note that with every hop ,the physical address in the frame changes and on the other hand the logical address remains the same.

-The logical address should remain the same otherwise the packets will be lost in the network.

-fig→



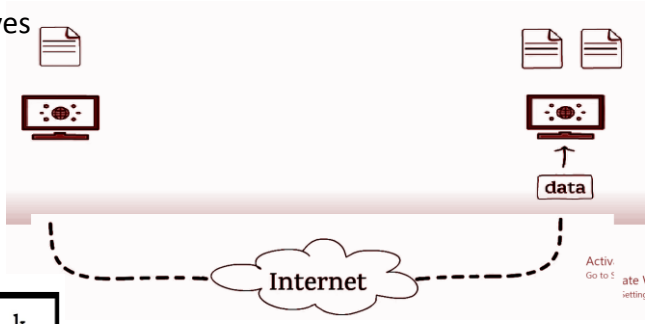
3. **Port Addresses :-** In a network a process running on one computer sends data to other process running in another computer.



-The destination Host can have multiple processes running simultaneously.

-So, once the destination host receives data using physical and logical addresses,it should be delivered to the right process.

-For this each process is assigned a label called **port address**.

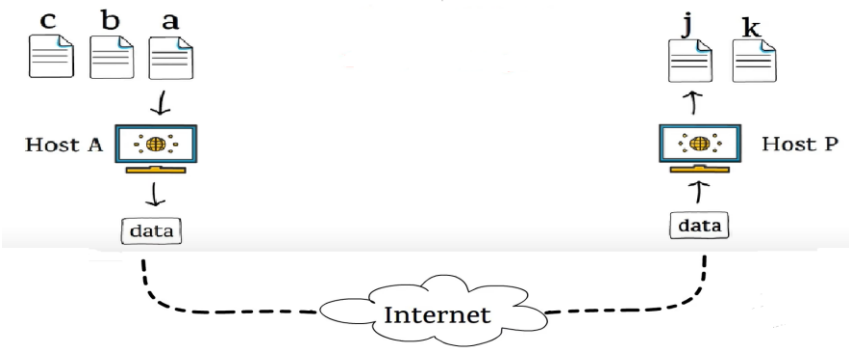


-Port address in TCP/IP is 16 bits in length.

-Example: Consider Host A is running 3 processes with port address a,b and c.

-Host P is running 2 processes with port address j and k .

-Host a generates some data which should be delivered to process j in Host P.



-To ensure the correct delivery of data to the right process, the transport layer encapsulates data from the application layer with source and destination port address a and j. | a | j | data |

The network layer adds the logical source address A and the Logical destination address P to the segment. | A | P | a | j | data |

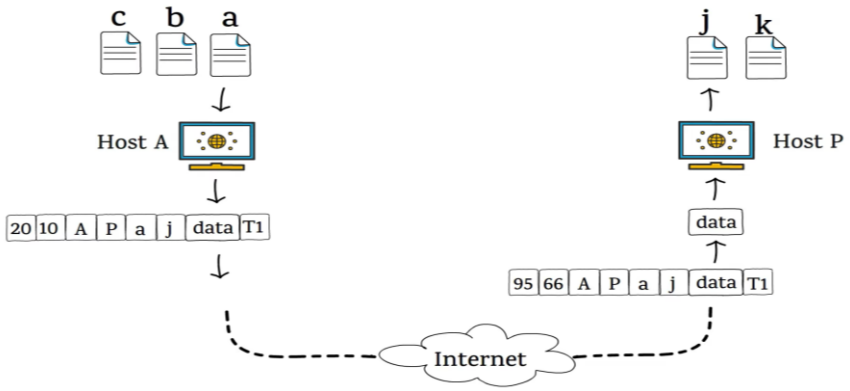
-The physical address is then added to the packet to form a frame which is then transmitted in the network. | 20 | 10 | A | P | a | j | data | T1 |

-The frame is received by the Host P which after decapsulation provides data to process J.

-Since the physical address changes with each hop, so the frame which Host P receives has a different physical address.

-Logical and port addresses remain the same.

-Fig→



Note:

- The logical address delivers data to the right Network.
- The Physical address delivers data to the right Host.
- The Port address delivers data to the right process.

Logical address ⟶ Network
Physical address ⟶ Host
Port address ⟶ Process

4. **Specific Addresses :**-The user-friendly addresses such as email address,URL or Universal Resource Address are referred to as specific addresses.
   -For example: Albert444@gmail.com and www.era.in .
   -These addresses get changed to the port address and logical address at the sender Using DNS.

#Data Link Layer:

-The Data-link layer is the second layer from the bottom.

-It is responsible for the node-to-node delivery of data. Its major role is to ensure error-free transmission of information.

-Specific responsibilities of the data link layer include framing, addressing, flow control, error control, and media access control.

-The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

-The data link layer adds a header to the frame to define the addresses of the sender and receiver of the frame.

-If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

-The data link layer also adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged, duplicate, or lost frames.
-When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

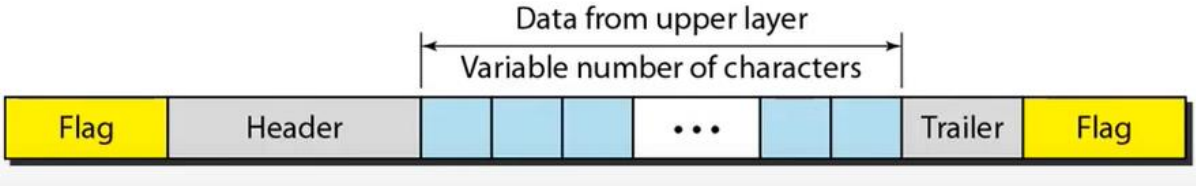
***Data Link Control:**-The two main functions of the data link layer are **data link control and media access control.**
-The first, data link control, **deals with the design and procedures for communication** between two adjacent nodes: node-to-node communication.
-The second function of the data link layer is media access control or how to share the link.
-Data link control functions include **framing, flow control and error control, and software implemented protocols** that provide smooth and reliable transmission of frames between nodes.
-To implement data link control, we need protocols.
- protocol is a set of rules that need to be implemented in software and **run by the two nodes involved in data exchange at the data link layer**.
-Here we have five protocols: two for noiseless (ideal) channels and three for noisy (real) channels.

***Framing:**-Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination.
-The data link layer, on the other hand, needs to pack bits into frames.
-Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address.
-Although the whole message could be packed in one frame, that is not normally done.
-One reason is that a frame can be very large, making flow and error control very Difficult because of the large size.
-When a single-bit error occurs in a very large frame, we need to retransmit the whole message again.
-When a message is divided into smaller frames, a single-bit error affects only that small frame and we need only resend that particular frame only.

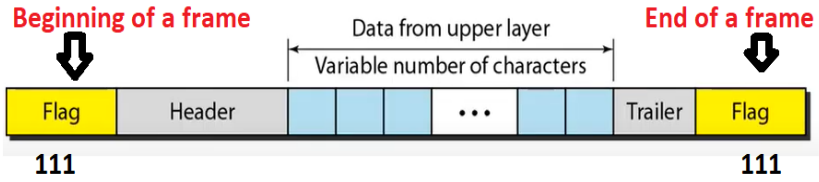
-There are 2 types of framing they are;

1. **Fixed-Size Framing:**-The frame is of fixed size and there is no need to provide boundaries to the frame.
   -The length of the frame itself acts as a delimiter.
   -it does not require additional boundary bits to identify the start and end of the frame.
   -Example − ATM cells.

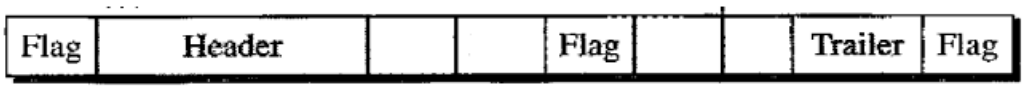
2. **Variable-Size Framing:**-In variable-size framing, we need a way to define the end of the frame and the beginning of the next.
   -There are two approaches were used for this purpose they are;:
   - **character-oriented approach/ :**-In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII.
     -The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits.
     -To separate one frame from the next, an 8-bit (I-byte) flag is added at the beginning and the end of a frame.



-Character-oriented framing was popular when only text was exchanged by the data link layers.
-Example: When 111 occur receiver knows that its a beginning of a frame and next it have Header,data, and tail .
-then 111 occur and receiver knows that its the end of the frame.



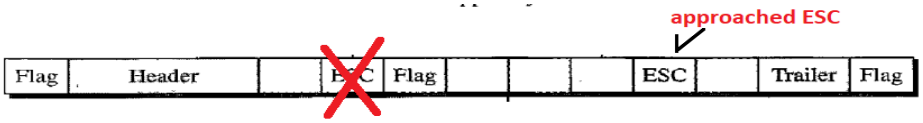
-suppose if the beginning flag and end flag and there is also a middle flag.
- basically the middle flag is a data section and it is not the end flag.
-but the receiver thinks that the middle flag is the end flag .so it will stop and it will look for another frame.



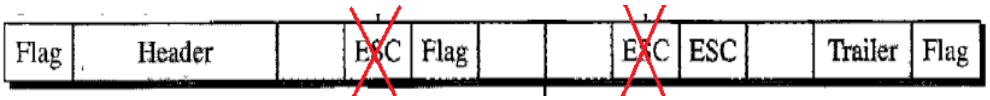
-to avoid this we add a special byte called ESC (escape character) before the middle flag.

-This method is called byte stuffing.
-And the receiver used to identify that the flag close to ESC is not the end flag
-And the receiver will remove the ESC and treat the middle flag as data.

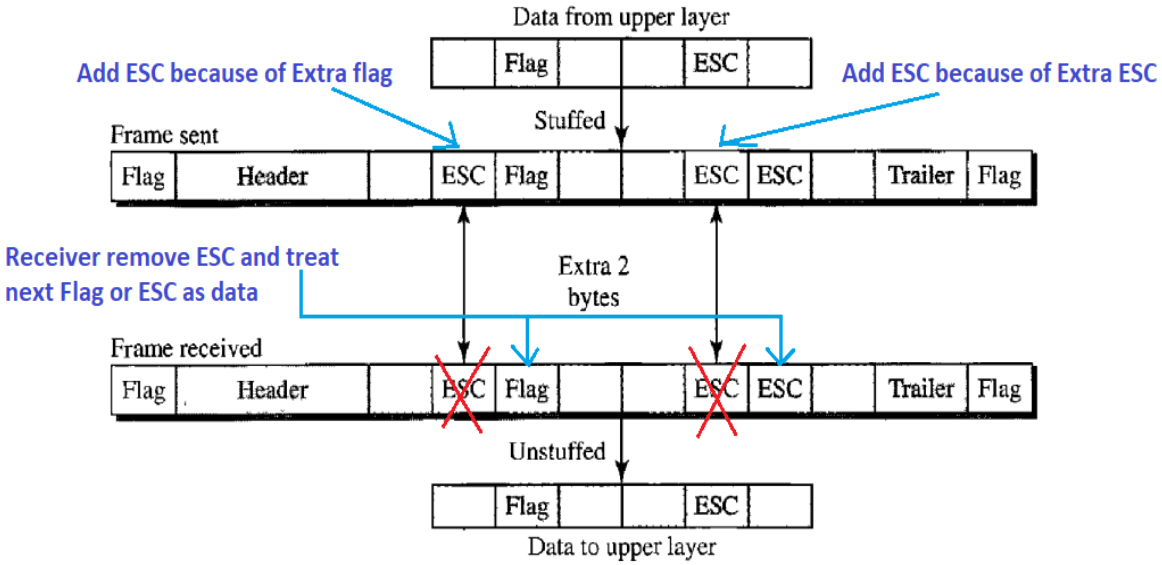
-If the introduced ESC (escape character) also approached in some other places in the data then we need to introduce on more ESC before approached ESC.



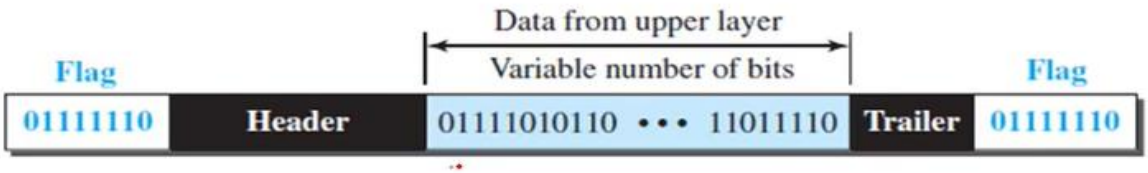
-Then the receiver will remove the ESC and treat the next ESC as data .



-Byte stuffing is the process of adding one Extra byte whenever there is a flag other than beginning and end flag or escape character ESC in the text.



- **bit-oriented approach/Bit Stuffing :**-In a bit-oriented approach, the name itself says that, it is concerned with bits.
  -So, it views the frame as a collection of bits.



-99% times both beginning and end flags have the same value “01111110”.
-Here data are in the form of bits.
-And also the flags are in the form of bits.
-so Here is a problem that when the data have similar bit pattern to flags bit pattern.
-
The receiver thinks that the sequence of bit part in the data part is the end flag which is similar to the end flag bit “01111110”.

-That will cause an error.

-So ,to overcome this issue we introduce / stuffed 0 after every five 1's.

Eg: →

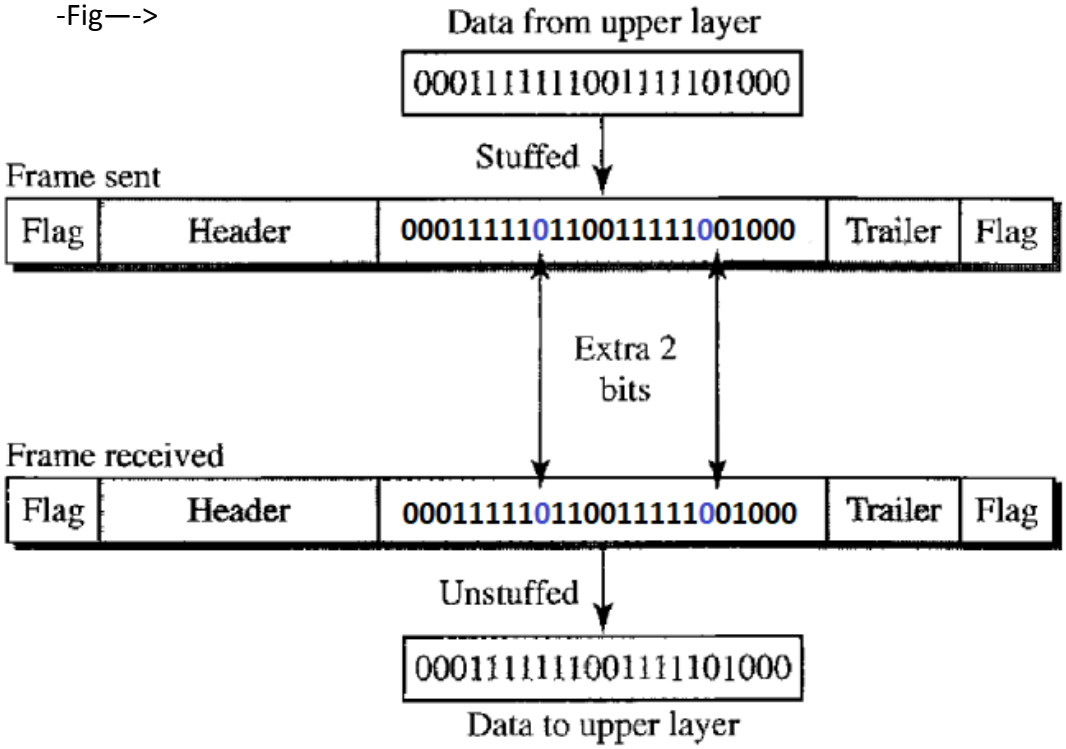
01111111110111111

⇩

01111101110111110 1

-And the receiver will unstuffed the added '0'.

-Fig—->



***Flow Control:**-The most important responsibilities of the data link layer are **flow control and error control**.

-Collectively, these functions are known as **data link control**.

-Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer.

-In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.

-The flow of data must not be allowed to overload the receiver.

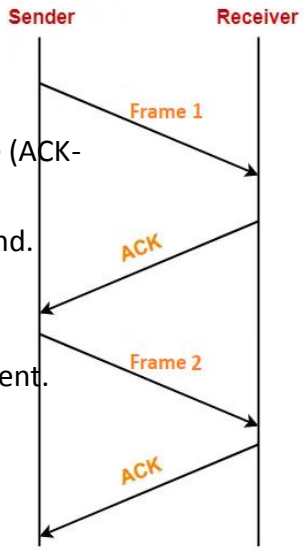
-Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.

-The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily.

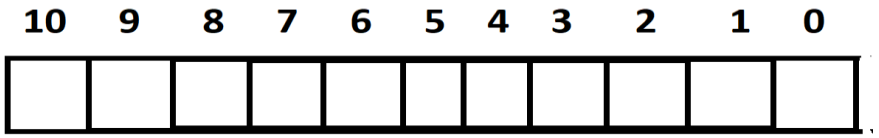
-Incoming data must be checked and processed before they can be used. Often, the processing speed takes longer than the transmission speed.

-For this reason,each receiving device has a block of memory, called a **buffer,** reserved for storing incoming data until they are processed.
-If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.
-Two methods have been developed to control the flow of data they are;

Stop-and-wait:-the stop and wait protocol is a flow control protocol where flow control is one of the services of data link layer.
-This protocol will not focus on any error control facilities.
-Sender send the frame one by one .
When the receiver get the frame 1 and send back the response (ACK-acknowledgement).
-sender only receives the ACK and then only next frame will send.
-In the Stop-and-wait method, the sender waits for an acknowledgement after every frame it sends.
-When acknowledgement is received, then only next frame is sent.



- **Sliding window protocol:**-In stop-and-wait protocol we can send one frame at a time ,where in Sliding window protocol we can send multiple frames at a time.
-The number of frames to be sent at a time is based on **Windows size.**
-each frame is numbers is called sequence number.
-Suppose we have a sender and a receiver ,the sender has 11 frames to send and each frames are numbered as 0 to 10.



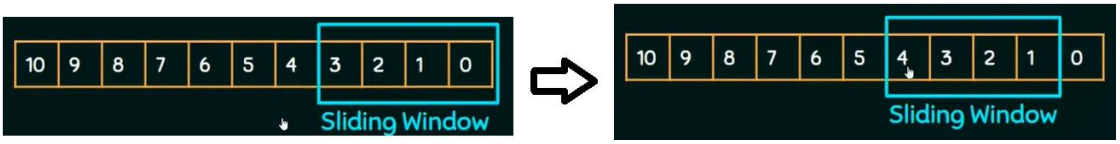
-How many frames can be send at a time is decided by Window size.
-suppose we have Windows size is 4 ,it means that 4 frames can be sent at a time .
-frame 0 is the first frame .
-Let's assume that the sender is sending the first frame that is frame 0 .
-Once the frame 0 is send there is a sliding window which is present in the frame and it says that frame 0 is send .

-after that frame 1,2,3 is sended before receiving and Acknowledgement.

-after the sending of frame 3 , now acknowledgement will send back to sender.
-first receiver send acknowledgement for Frame 0 to the sender .

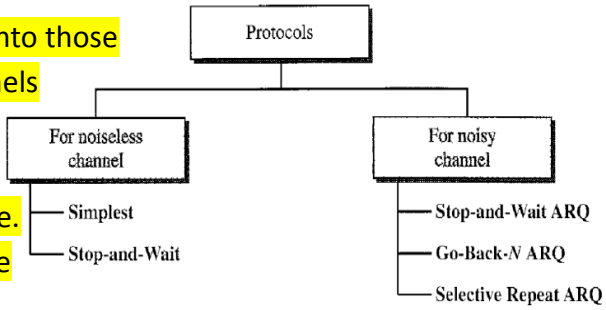
-when the acknowledgement received by sender ,then the sender now can send next frame that is frame number 4 .

-when the frame 4 send then the sliding window starts to slide frame 4 ,it means that frame 0 is acknowledged and frame 4 is send.



-How we assume that the receiver send acknowledge of frame 1 to sender.
-Then the sender can send the next frame 5 to the receiver.
-Once frame 5 is send the sliding window it moves to frame 5 and it says that frame 2 to 5 are not acknowledged,but frame 0 and frame 1 are acknowledged.

-In Flow control we have 2 protocols they are;
-Protocol and types ;(We divide the protocols into those that can be used for noiseless(error-free) channels and those that can be used for noisy (error-creating) channels)
→**simplest protocol:-**sending frames one by one.
-send one frame and after 10 second next frame is send.
-This is called simplest protocol
-bakkki explain cheythittunde.



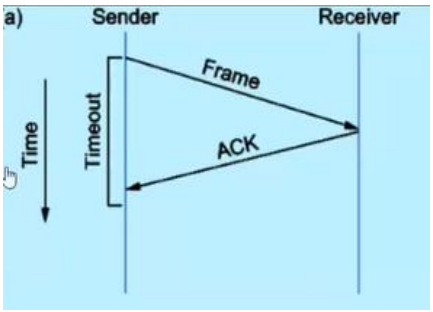
***Error control**:- is both error detection and error correction.
-It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.
-Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted.
-This process is called **automatic repeat request** (ARQ).
-Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.
-Categories of Error Control are;

- **Stop and wait ARQ** (Automatic Repeat Request):-It is a very simple protocol .
  -After transmitting one frame the sender waits for an acknowledgement before transmitting the next frame.
  -suppose the sender is sending one frame and it won't send the next frame before receiving an acknowledgement from the receiver.
  -But there are chances that the frame to be lost or acknowledgement is lost.
  -If that happens the sender will wait for an infinite amount of time and the receiver will also wait for an infinite amount of time.
  -And there will be no progress in the stop and wait protocol .
  -To avoid this we use Stop and wait ARQ.
  -In Stop and wait ARQ the acknowledgement does not arrive after a certain period of time the sender times out and retransmits the previous sended frame (original frame).
  -if the sender is sending a frame and that frame is received by the receiver and the receiver is sending an acknowledgment.
  - if the acknowledgment is not reaching the sender on time so there will be a timer that is running in the sender side once the timer expires or the sender times out immediately the sender will retransmit the frame again.
  -so this retransmission is automatic and that is why we call this as stop and wait automatic repeat request protocol (Stop and wait ARQ).
  -Here we have 4 types of scenarios

1. **Scenario A** —---------->

-In Scenario A is a sender and there is a receiver and sender has a timer.

-In Scenario A the ACK (acknowledgement) is received by the Sender before the time expires.



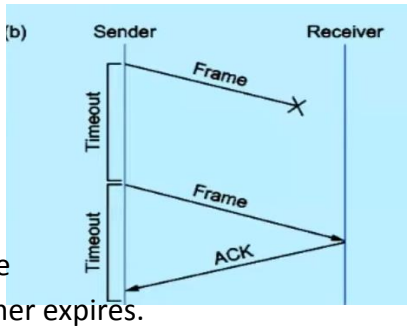
2. **Scenario B**

-In Scenario B the sender is sending a frame but this frame is lost .

-So in this Scenario the original frame is lost .

-Therefore the receiver did not send any ACK .

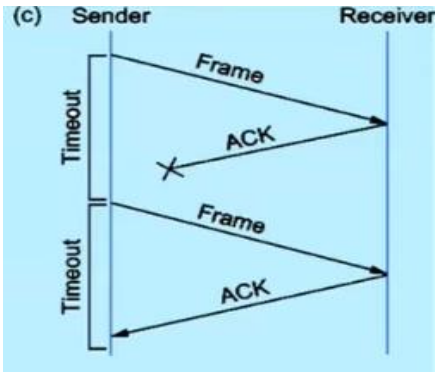
-So the timeout expires here ,And the sender retransmit this frame again and the ACK for the retransmitted frame is received before the timer expires.



3. **Scenario C**

In this Scenario the ACK is lost

-Bakki thanne ezhuthukaa

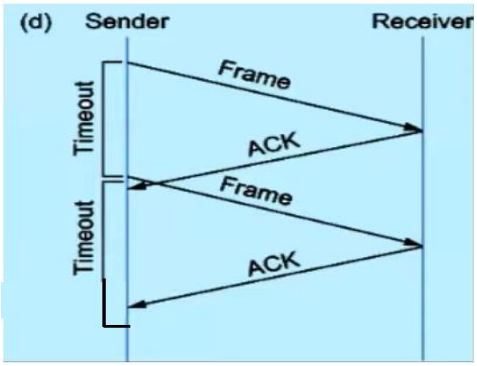


4. **Scenario D**

-In this Scenario The frame is not Lost and the ACK is also not lost.

-But sender get the ACK after The timer expires.

-so The sender retransmit the same frame to receiver and wait to get the ACK from the receiver.



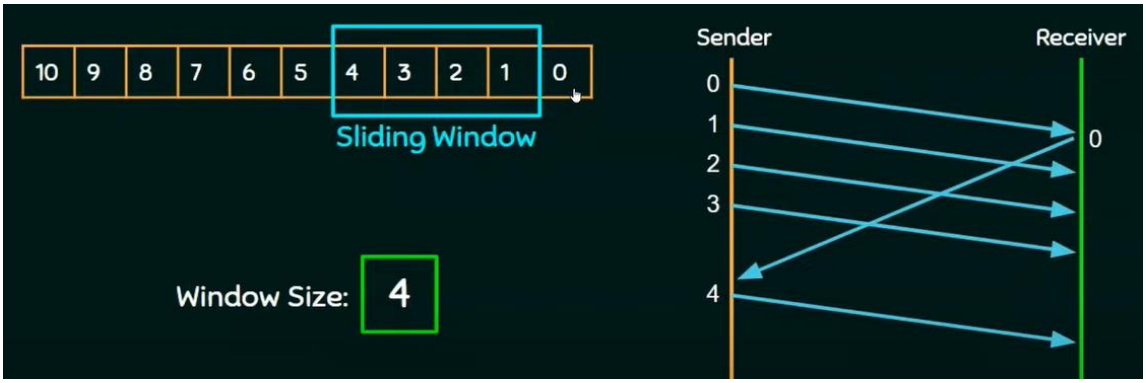
- **Go back-N ARQ:-**In Go back-N ARQ the 'N' means the sender window size.

-Suppose we have Go back 5 ARQ ,it means that we can send 5 frames can be send by the sender to the receiver before the getting the ACK from the receiver.

-In Go back-N ARQ uses the concept of protocol pipeline.
-That means the sender can send multiple frames before receiving the ACK for the first frame.
-If there are finite number of frames and each frames are numbered in a sequence manner.
-The number of frames that can be sent depends on the window size of the sender.
-**Suppose the ACK of the frame is nor received within a time period then all the frames in the current window are transmitted.**
-Suppose if we have a sender and we have a receiver and let's assume that there are 11 frames to be sent and the frames are numbered as 0 1 2 3 4 5 6 7 8 9 and 10

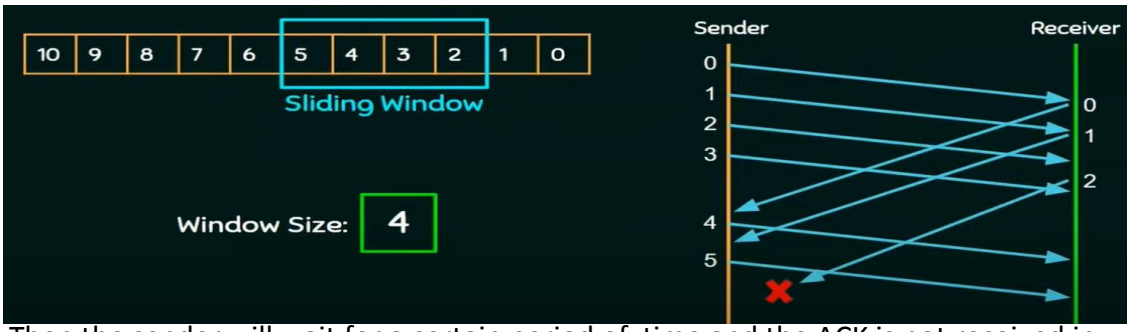
-And suppose the window size is 4.
-The sender will send frame 0 to frame 3 one by one.

-Now the sender is expected to receives an ACK for frame 0.
-When the sender receives an ACK for frame 0 ,it means that it has been successfully received by the receiver.
-Then the sender will send the next frame 4 and the window slides to frame 4.

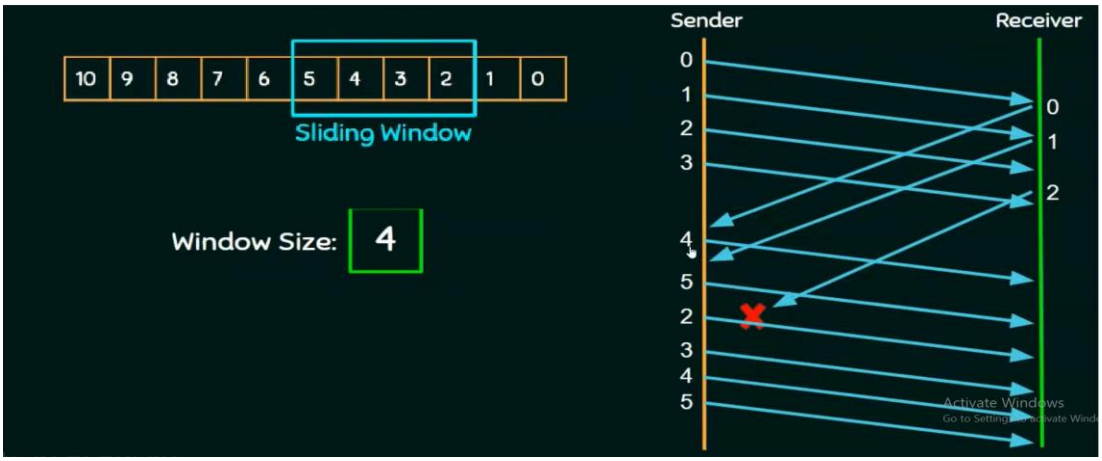


-Now the receiver will send ACK for frame 1 ,then the sender send next frame 5 and the window will slide to frame 5.

-Now let's assume that the sender not receives the ACK for frame 2 , because either the frame is lost or ACK is lost.



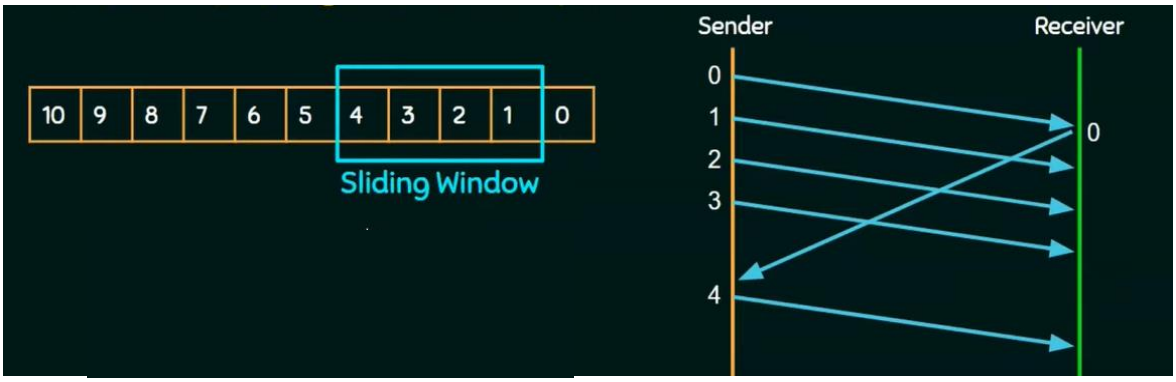
-Then the sender will wait for a certain period of time and the ACK is not received in time ,then the sender retransmits not only frame 2 but also retransmits all the frames in the current window are frame 2,3,4,5.



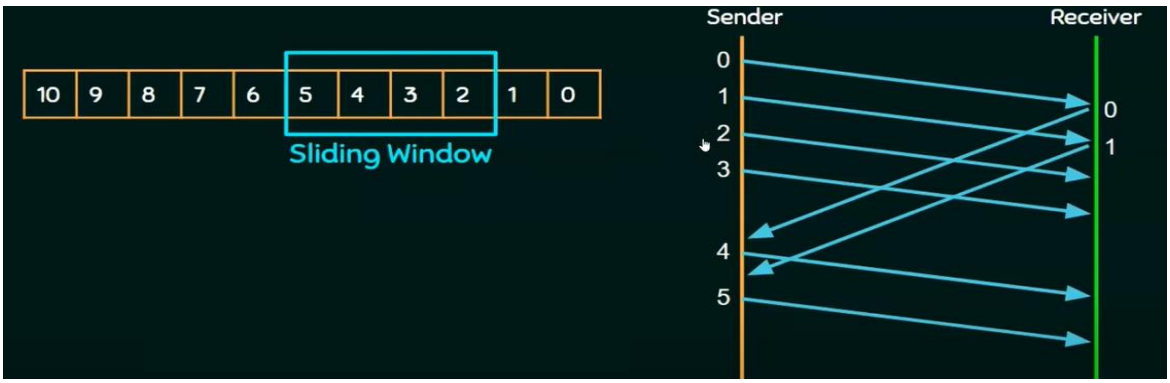
-The frame 3,4 and 5 are already send by sender but the receiver discard .
-because the receiver not get the frame 2 therefore it discards the all frame that send after frame 2.
-This is the working of Go back-N ARQ

- **Selective Repeat ARQ**:-In Selective Repeat ARQ only lost frames are retransmitted, while correct frames are received and not all frames are sended.
  -In Go back -N ARQ either the frame is lost or the ACK is lost all the frames in the current window are retransmitted.
  -Where in Selective Repeat ARQ only the lost frames or the error frames will be transmitted.
  -So The receiver will be keep track of all the frames that received using sequence number ,if any sequence number is missing it will send either a negative acknowledgement (NACK) or it won't send any ACK so that act as an indication to the sender to retransmit only that frame.
  -Suppose we have a sender and a receiver and there are 11 frames to send and the frames are numbered as 0 to 10.
  -and we have windows size 4 .
  -Now the sender is now going to send frame 0 and frame 1,2 and 3.

-And assume that frame 0 is ACK and the sender will send the next frame 4 and the window will slide to frame 4.



-And assume that receiver is now send ACK for frame 1 and sender will send the next frame 5 and window will slide to frame 5.

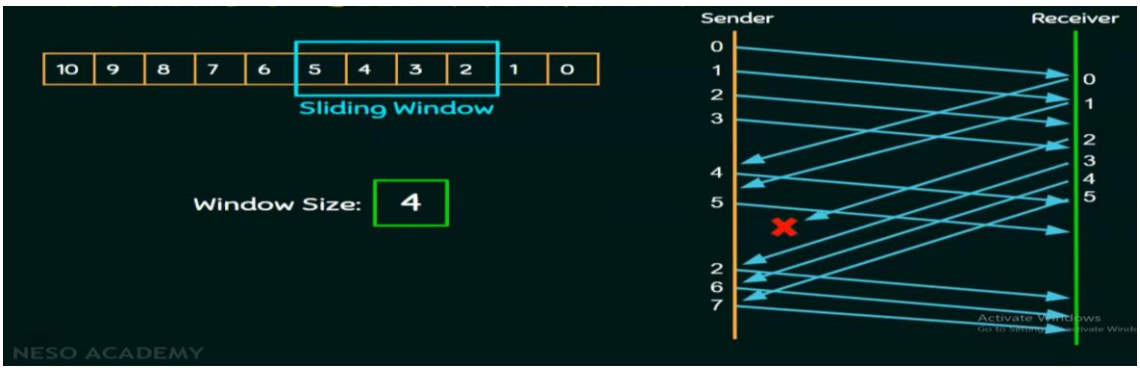


-And assume frame 2 is lost ,So obviously the receiver will not send ACK for frame 2 ,either the frame is lost or ACK is lost .

-In this case the receiver might have ACK frame 3 and the sender will not send frame 4 and 5 again and it knows that frame 2 is missing because the receiver would have send a negative ACK for frame 2 .

-so the sender will retransmit frame 2 alone.

-And so on



*** Error:**- When Data are transmitted in the network ,then the data can be corrupted during transmission .

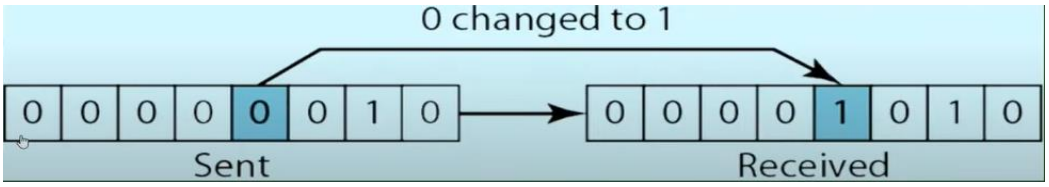
-So the errors that are caused because of this transmission are called as transmission error.

-For effective communication, error must be detected and corrected.

-There are two types of error they are;

1. **Bit error/Single bit error:**- In Bit error only one bit is going to be changed.

   -Example



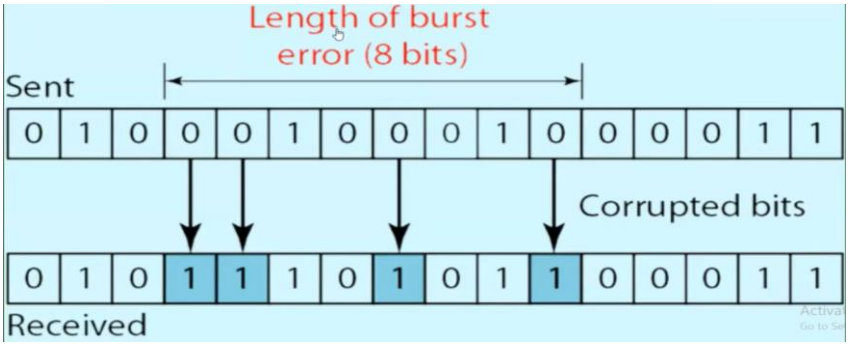
   -In this example the sender is sending 00000010 ,but the receiver received 00001010.

   -Here the sender is sending some information and the receiver is receiving a different information.

   -Here change is in only one bit ,this is called Bit error.

2. **Burst error:**- In Burst error two or more bits in the data unit have been changed.

   -Example



   -In this example some bits are corrupted.

   -The length of the burst error is 8 here .

->Error detection:-Generally error detection will be done by the receiver ,it means that the receiver has to decide whether the received data is correct or not ,without having a copy of the original message.

-When we send only the original message, it's very difficult for the receiver to understand or to know whether there is an error or not.

-So, with the message some additional information have to be sent by the sender.

-With this additional information it helps the receiver to understand or to find out whether there is an error or not in the data that is being transmitted.

-To detect or correct errors, we need to send some extra bits with the data.

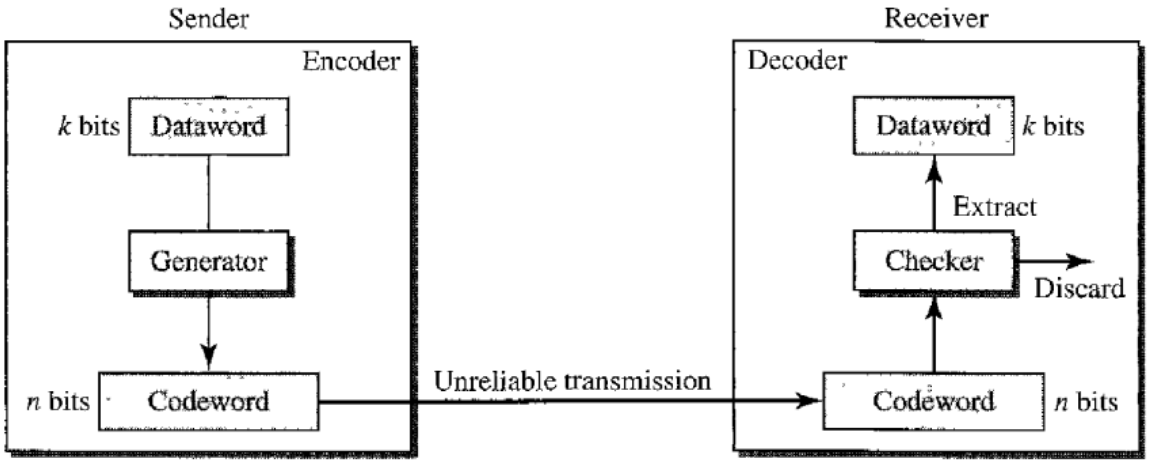
-This Extra bits are called **redundant bits.**

-Suppose the sender send a data to the r

00001 + 011 --> 00001011

(Original data) (Add extra bit)

-First the sender gives the data to the generator ,then the **generator** generates the code which is the redundancy bit .



-Then this redundancy bit will appends to the original data which is called **code word.**

Code word = original data (data word) + extra bit (redundancy bit)

-And this code word (whole message) is sended to the receiver .
-Then the receiver takes the message and check the message using **checker.**
-If there is no error then it accepts it otherwise it rejects it.
-There are many different methods of error detection they are;

1. **parity check:**-It is the simplest technique for detecting and correcting errors.
   -The parity check is a network method designed to detect errors and check the integrity of the data received at the receiver side by the sender side.
   -The parity check method adds a bit to the original data for checking errors at the receiver end.

-Example: data word is 1011

-Here generator create a redundant bit (parity bit) .

-Here parity bit is 1 because total number of words in the data word is odd .

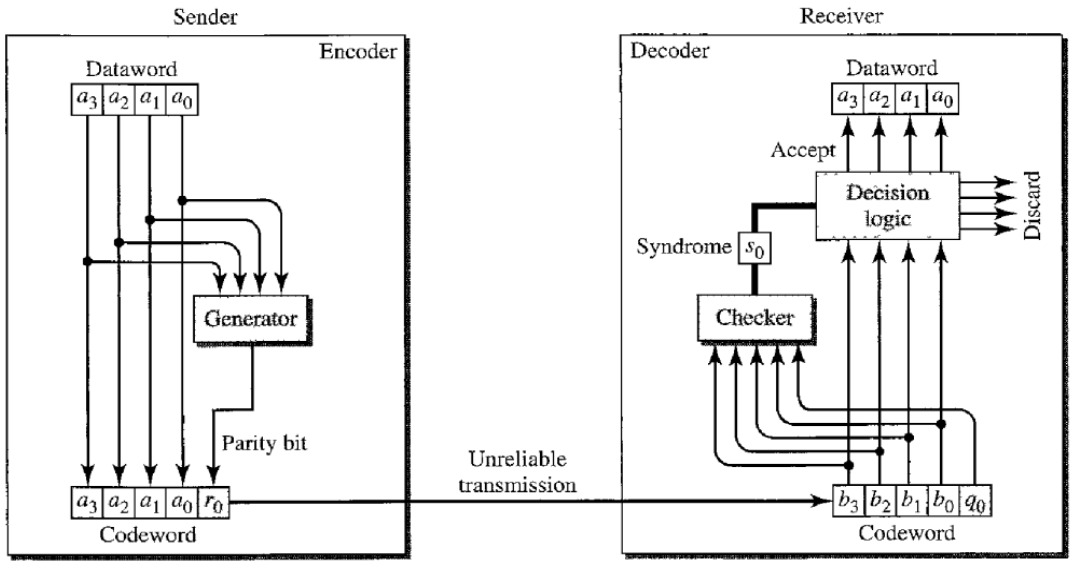
-and it transmit to the receiver side.

-receiver check the error using checker .

-The checker add the number of 1's , the added sum is called **syndrome**

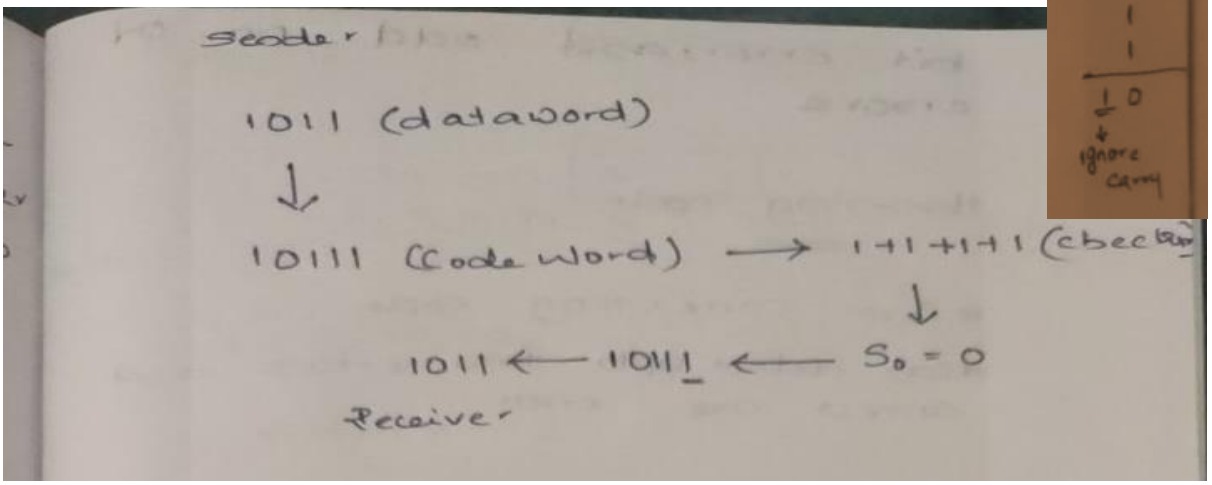
-if Syndrome 0 then the receiver accept data without the parity bit.

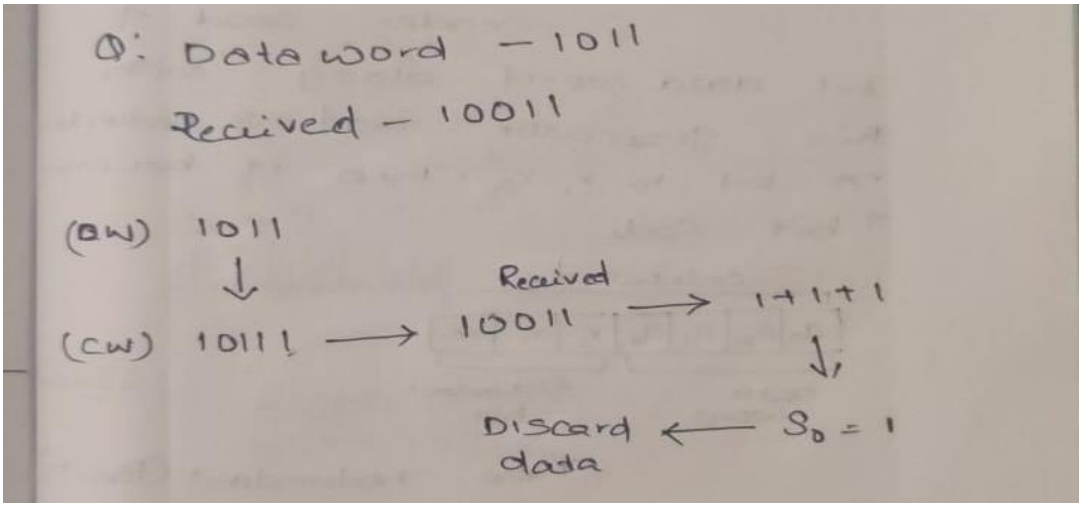
-If syndrome is 1 receiver will discard data.



-Parity check can't identify even no of errors.

It can identify only single bit error and odd no of errors.





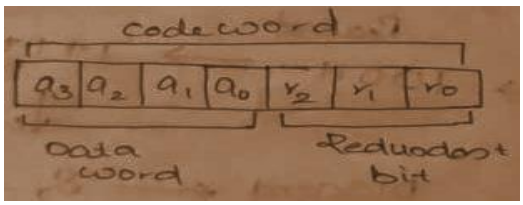
→hamming code:-It is used to detect and correct error.

-It can detect single bit and two bit errors.

-But only a single bit error can fix.

-Suppose Sender send 4 bit dataword and the generator add 3 redundant bit **r_0, r_1, r_2.**

-Then it become 7 bit code.

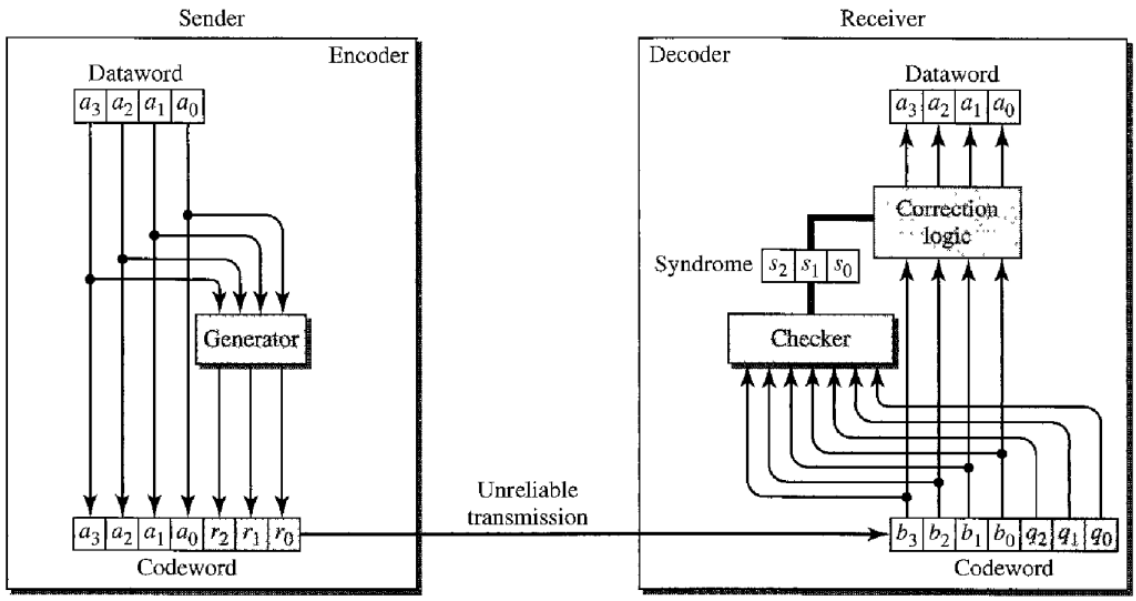


-And send to the receiver

-And the Syndrome check code word to find error.

-If Syndrome **s_2, s_1, s_0** is 0 0 0 then there is no error in the code word .

-Then the receiver accept this codeword and convert into dataword.

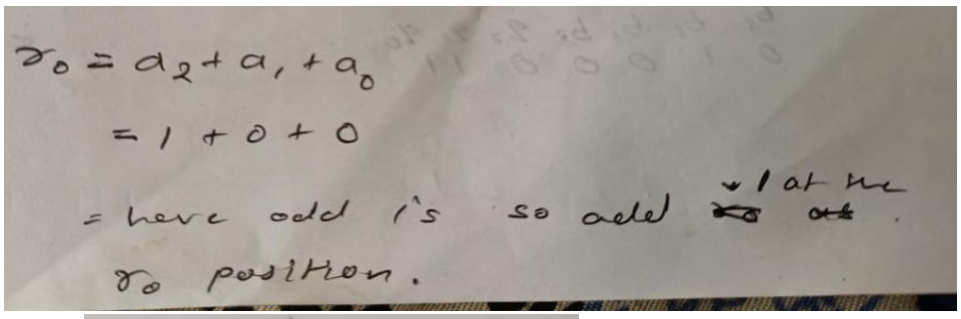


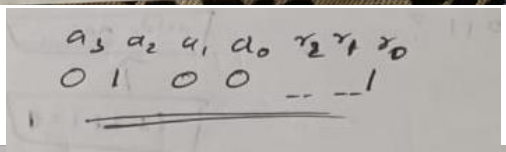
>Example:dataword is 0100.

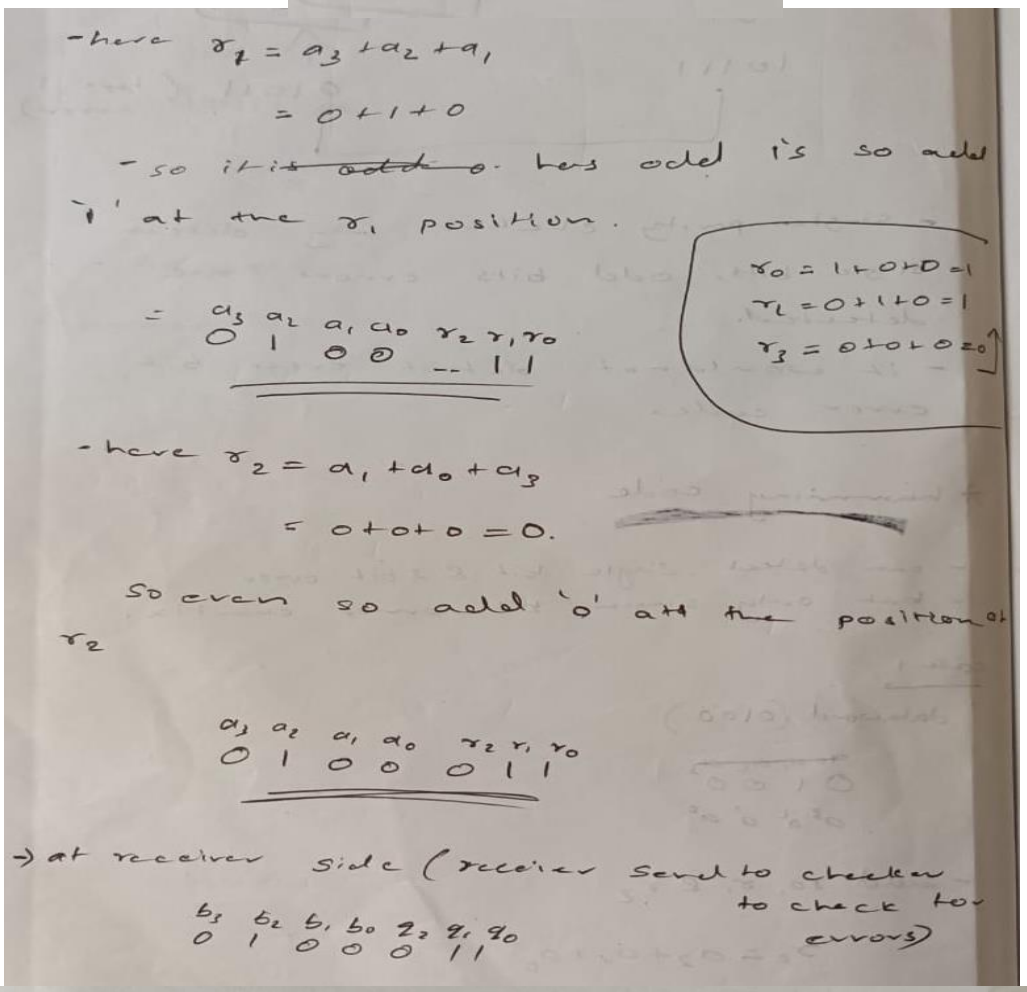
0 1 0 0

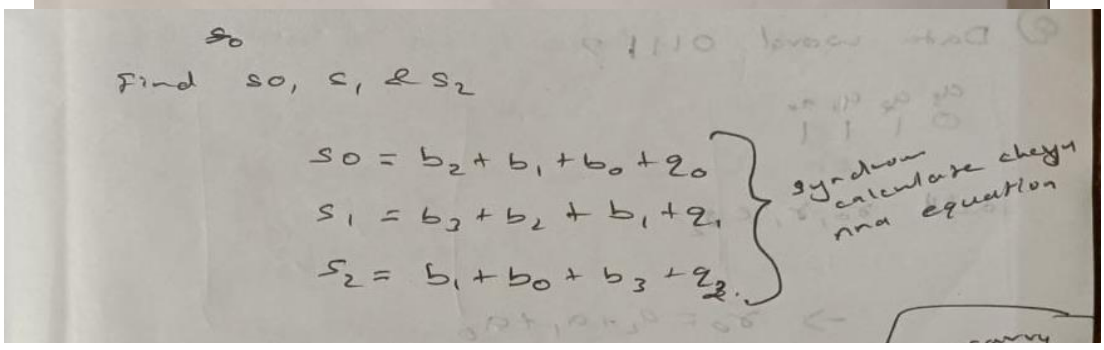
a_3 a_2 a_1 a_0

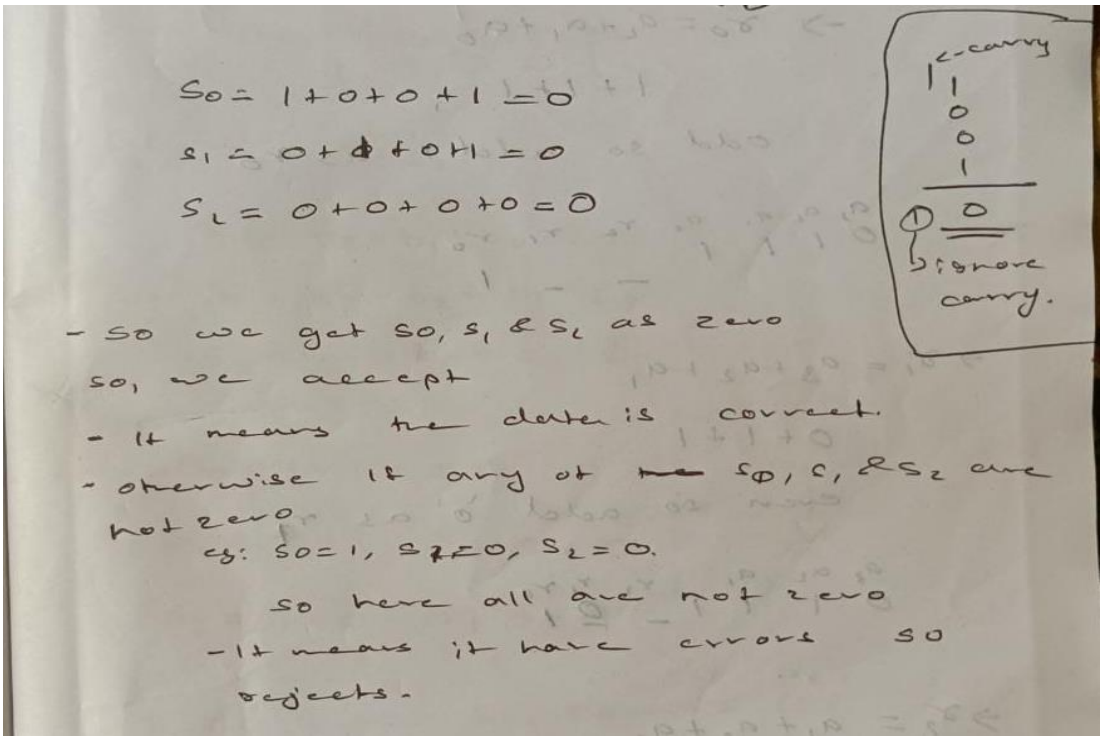
-add **r_0 ,r_1,r_2,** before adding we need to find the values of **r_0 ,r_1,r_2.**





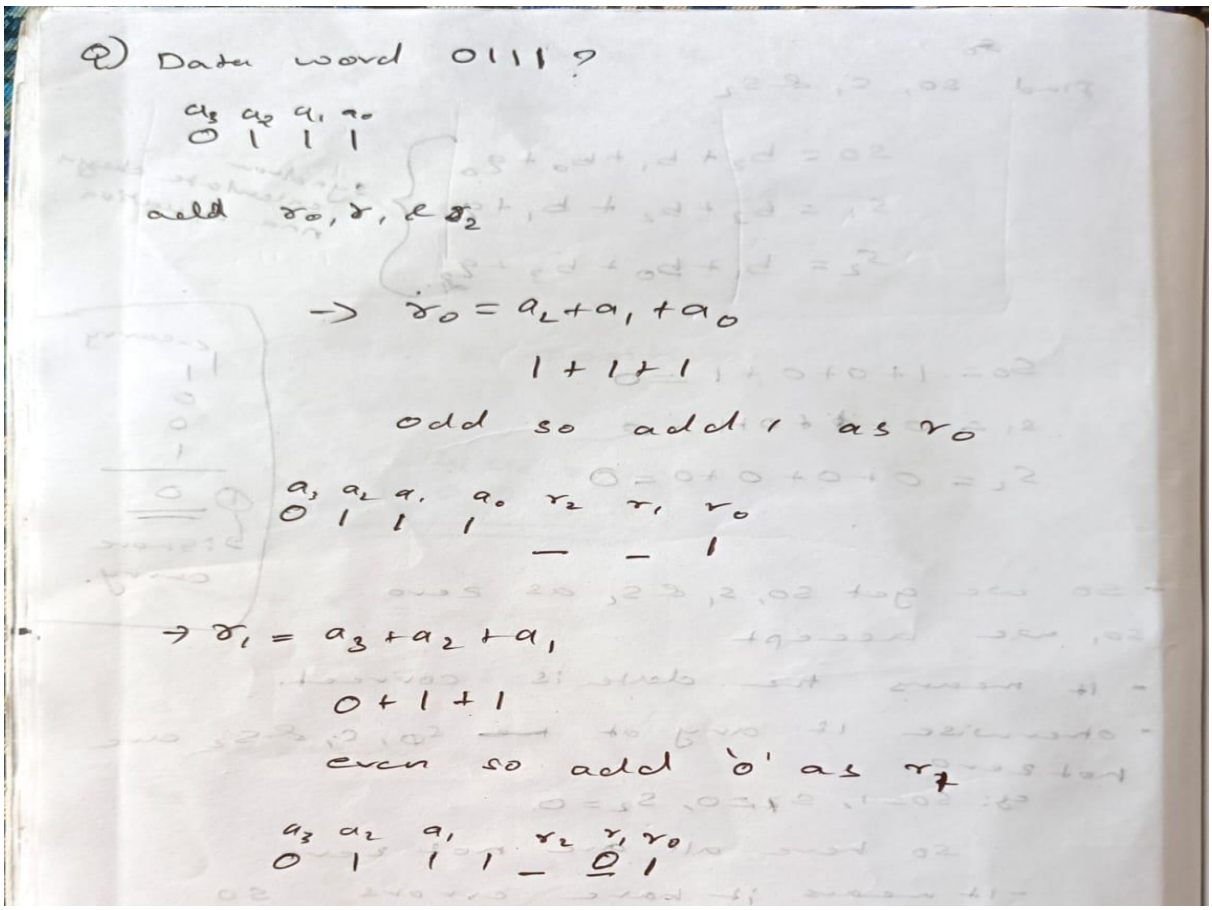






->To correct errors using hamming code. (table important annu)

Syndrome	000	001	010	011	100	101	110	111
Error	none	q0	q1	b2	q2	b0	b3	b1



$\Rightarrow r_2 = a_1 + a_0 + a_3$

$= 1 + 1 + 0$

even so add '0' as r_2

$a_3\ a_2\ a_1\ r_2\ r_1\ r_0$

$= 0\ 1\ 1\ 1\ 0\ 0\ 1$

codword = 0111001

→ at receiver side (

- receiver get codword as

0110001

$b_3\ b_2\ b_1\ b_0\ r_2\ r_1\ r_0$

0 1 1 0 0 0 1

Find s_0, s_1 & s_2

$s_0 = b_2 + b_1 + b_0 + r_0$

$s_1 = b_3 + b_2 + b_1 + r_1$

$s_2 = b_1 + b_0 + b_3 + r_2$

$s_0 = 1 + 1 + 0 + 1 = 1$

$s_1 = 0 + 1 + 1 + 0 = 0$

$s_2 = 1 + 0 + 0 + 0 = 1$

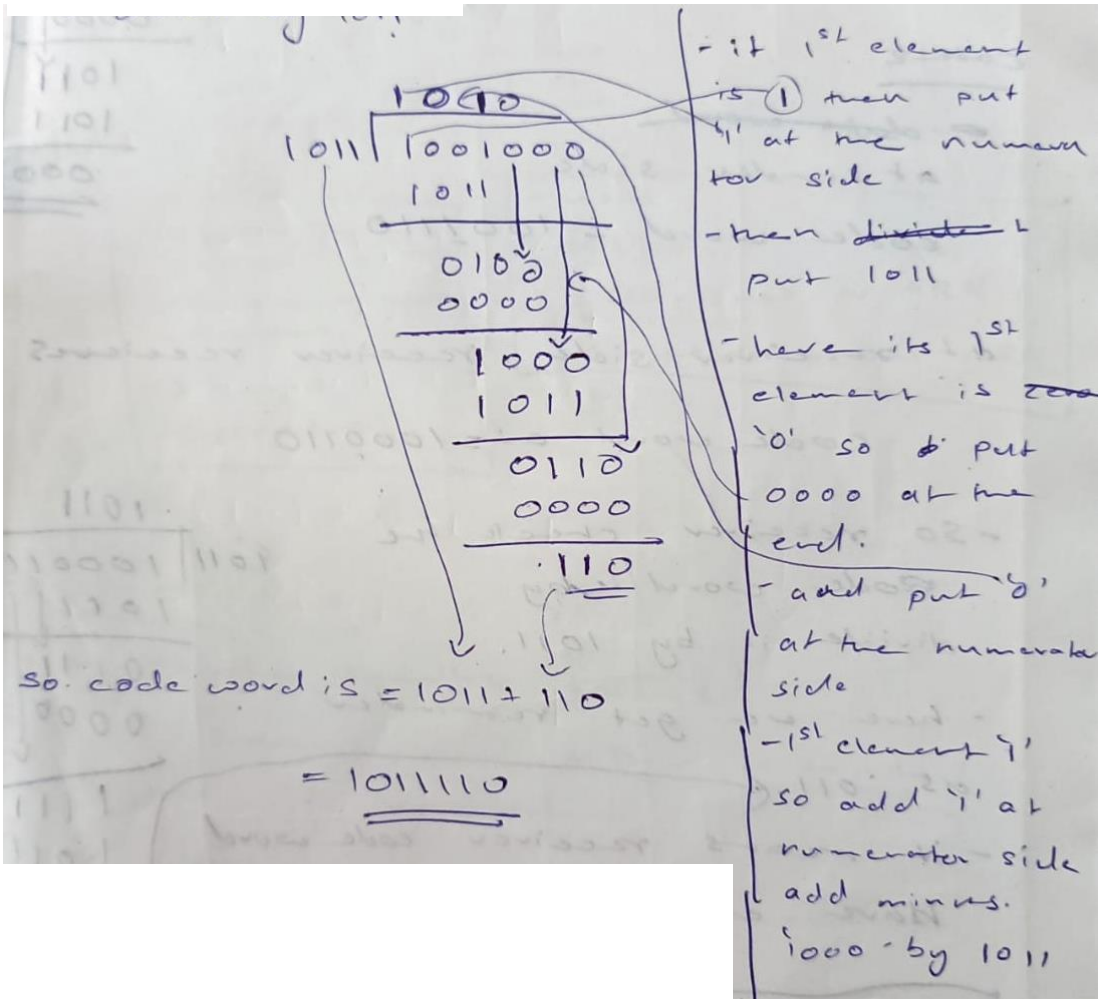
so, error position is 101 ($s_0 s_1 s_2$)

101 is b_0 position so take the inverse of b_0

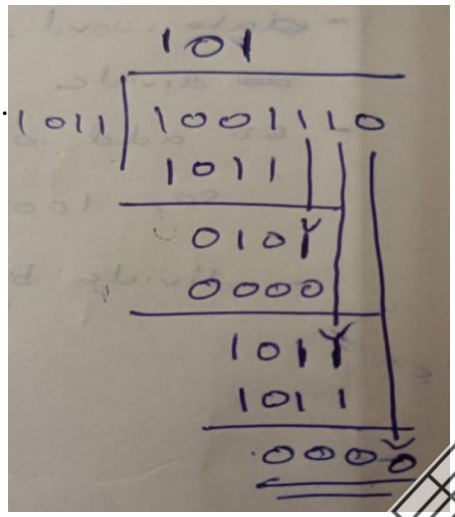
so corrected answer is

0111001

2. **CRC (cyclic redundancy check):**-The Cyclic Redundancy Checks (CRC) is the most powerful method for Error-Detection and Correction.
   -Example dataword is 1001 and divisor 1011.
   -The sender and receiver Knows the divisor value .
   -So based on the divisor the CRC is generated in the sender side and verified in the receiver side.
   -Add 000 at the end of the data word ie,1001000.
   -Then divide it with 1011.



   -Here we have codeword =1011110
   -And we transmit it to the receiver side.
   -Then the Checker divide the cordword with divisor 1011.
   1011110/1011
   -Here we get 0000 at the reminder side .
   -it means we get correct code.

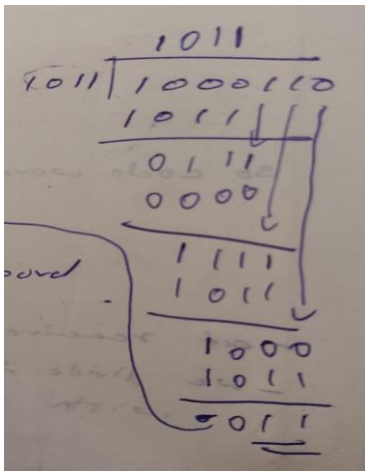


-There is another case ,that sender send codewode as 1001110 and at receiver sider receiver received codeword as 1000110.

-So receiver check the codeword by divisor 1011.

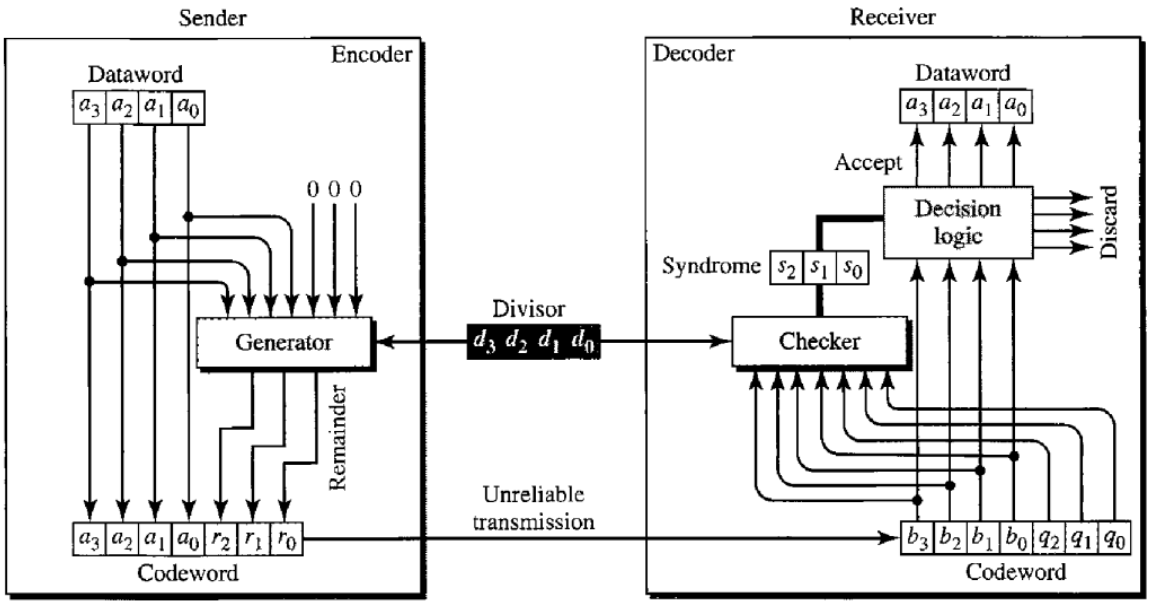
-Here we get reminder as 011 ,it means that the Receiver side codeword have error.

-so we reject it.



S_2,S_1,S_0 =000
We accept otherwise we reject.

-Fig :CRC encoder and decoder



Module-2

#Wired LAN:-a local area network (LAN) is a computer network that is designed for a limited geographic area such as a building or a campus.
-Most LANs today are also linked to a wide area network (WAN) or the Internet.
-In the LAN market, there are a number of technologies, They are **Ethernet, Token Ring, Token Bus, FDDI, and ATM LAN.**
-Some of these technologies survived for a while, However Ethernet is by far the most popular technology.
-Although Ethernet has gone through a four-generation evolution during the last few decades, the main concept has remained.

***IEEE Standards:**-In 1985, the Computer Society of the IEEE started a project, called **Project 802**, to create guidelines that will allow equipment from various manufacturers to communicate with one another.
-In Project 802 it does not try to replace any part of the OSI or the Internet model.
-Instead It specifies a function for the physical layer and the data link layer.
-The standard was adopted by the American National Standards Institute (ANSI). In 1987, And the International Organization for Standardization (ISO) also approved it as an international standard under the heading of ISO 8802.
-The relationship of the 802 Standard to the traditional OSI model is shown in the below Figure.

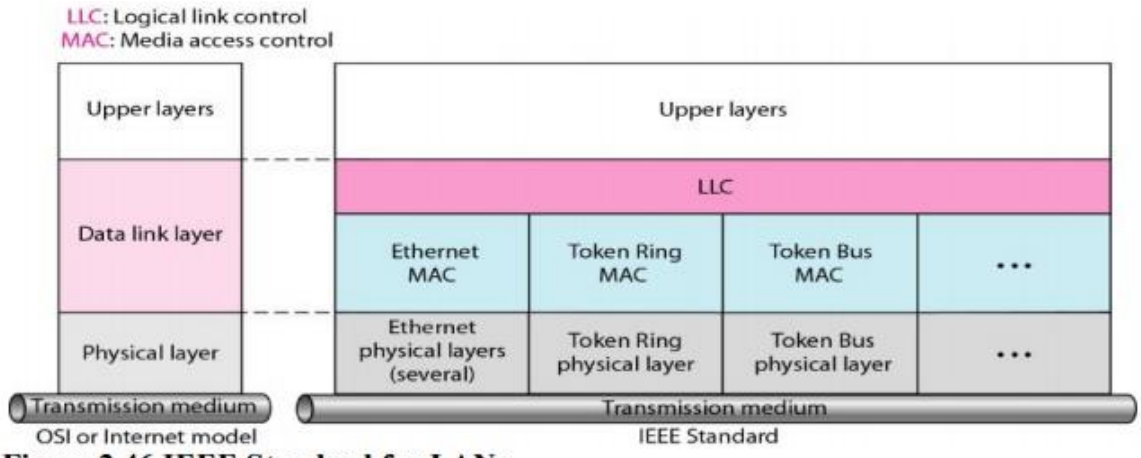


Figure 2.46 IEEE Standard for LANs.

-The IEEE has subdivided the data link layer into two sublayers: logical link control (LLC) and media access control (MAC).
-IEEE has also created several physical layer standards for different LAN protocols.

- **Data link layer:**-the data link layer in the IEEE standard is divided into two sublayers: **LLC** and **MAC.**

1. Logical Link Control (LLC):-The data link control handles framing, flow control, and error control.
   -But In IEEE Project 802, flow control, error control, and part of the framing duties are collected into one sublayer called the logical link control.
   -And Framing is handled in both the LLC sublayer and the MAC sublayer.
   -The LLC provides one single data link control protocol for all IEEE LANs.
   -But the media access control sublayer, which provides different protocols for different LANs.
   -The above figure shows one single LLC protocol serving several MAC protocols.

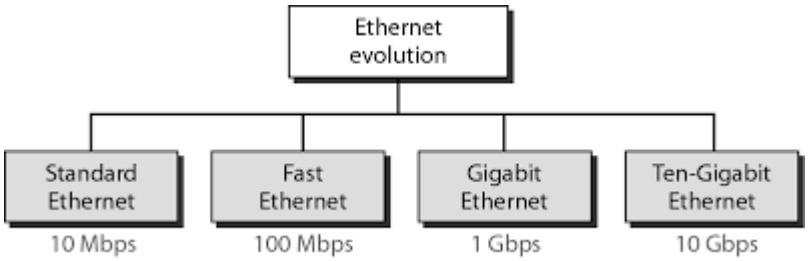
2. Media Access Control (MAC):-IEEE Project 802 has created a sublayer called media access control that defines the specific access method for each LAN.
   -And a part of the framing function is also handled by the MAC layer.
   -The MAC sublayer contains a number of different modules; each defines the access method and the framing format for the corresponding LAN protocol.

- **Physical Layer:**-The physical layer is dependent on the implementation and type of physical media used.

***Ethernet :**-One of the most widely used wired LAN technologies.
-Ethernet operates in the **data link layer** and the **physical layer.**
-Ethernet belongs to the family of networking technologies that are defined in the IEEE 802.2 and IEEE 802.3 standards.
-And it supports data bandwidth of 10,100,1000,10000,40000 and 100000 megabits per second (100 gigabits per second).
-initially when Ethernet was evolved it was in megabits per second now it ranges up to 100 gigabits per second and more.

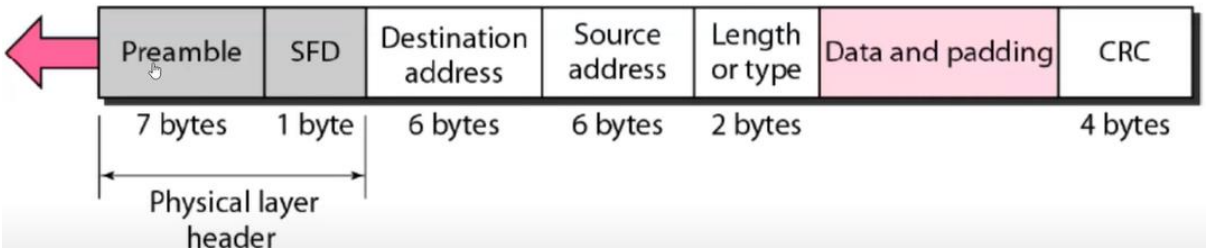
→Ethernet Standards:-The Ethernet define **layer 2 protocols** and **layer 1 technologies**.
-In layer 2 we will call it as an **Ethernet protocol** in layer 1 it is an **Ethernet technology** where this Ethernet technology deals with the physical layer.
-Ethernet works in layer 2 that is the data link layer Ethernet two separate sub layers of the data link layer to operate one is the logical link control LLC and the other one is the MAC.
-Explain LLC and MAC

→Evolution of Ethernet



-It started with standard Ethernet then came the Fast Ethernet then came the Gigabit Ethernet now we have 10 Gigabit Ethernet.
-The standard ethernet speeds up to 10 megabits per second fast ethernet speeds up to 100 megabits per second gigabit ethernet speeds up to 1 gigabit per second and 10 gigabit ethernet speeds up to 10 gigabits per second.

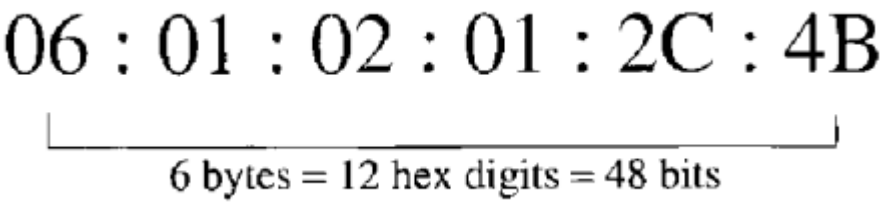
→Frame format of Ethernet



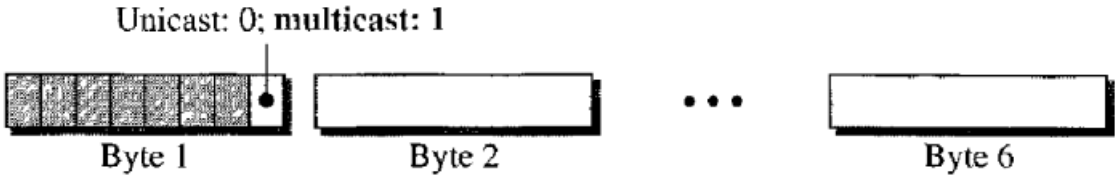
-Preamble and SFD, which operate at the physical layer, begin an Ethernet frame.
-CRC, the final field, is utilized to find errors.
-The **preamble** is of 7 bytes means it have 56 bits of alternating 1's and 0's .
-The **preamble** enables bit synchronization between the sender and receiver.
-The **SFD** stands for Start frame delimiter.
-The **SFS** is also used for synchronization purpose and its field is always set to 10101011.
-the last two bits of SFS (1010101**11**)that two ones indicates the receiver that the upcoming field is the destination address.
-The **Destination Address** contains the **MAC address** of the device for which the data is intended.
-The **Source Address** contains the source machine's MAC address.
-The **Source Address** is always a unique address (Unicast), and the least significant bit of the first byte is always 0.
-The **Length** indicates the length of the entire Ethernet frame.
-This 16-bit field can hold a length value between 0 to 65534, but length cannot be larger than 1500 Bytes because of some limitations of Ethernet.
-And describe the **type** of data.
-The **Data** where actual data is inserted, also known as P**ayload.**

-Both IP header and data will be inserted here if Internet Protocol is used over Ethernet.
-The maximum data present may be as long as 1500 Bytes.
-If the data length is less than the minimum length, which is 46 bytes, padding 0's are appended to make up the difference.
-The **Cyclic Redundancy Check (CRC)** it is mainly used for error detection purpose.

→ Ethernet Address



-The Ethernet address is 6 bytes (48 bits), normally written in hexadecimal notation, with a colon between the bytes.
-The least significant bit of the first byte defines the type of address.
- If The bit is 0, the address is unicast; otherwise, it is multicast.



-In byte 1 the last bit is 0 then the address is unicast ,otherwise it is multicase.
-A unicast destination address defines only one recipient; the relationship between the sender and the receiver is one-to-one.
-A multicast destination address defines a group of addresses; the relationship between the sender and the receivers is one-to-many.
-If all the bits are 1 then it is Broadcast Address.
Eg; ff:ff:ff:ff:ff:ff

>Example:Define the type of the following destination addresses:

a. 4A:30:10:21:1O:1A
b. 47:20:1B:2E:08:EE
c. FF:FF:FF:FF:FF:FF

-To find the type of the address, we need to look at the second hexadecimal digit from the left.
-If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast.
-Therefore, we have the following:

a. This is a unicast address because A in binary is 1010 (even).
b. This is a multicast address because 7 in binary is 0111 (odd).

c. This is a broadcast address because all digits are F's.

→Advantages of Ethernet:-It is not much costly to form an Ethernet network. As compared to other systems of connecting computers, it is relatively inexpensive.
-Ethernet network provides high security for data .
-In this network, the quality of the data transfer is maintained.

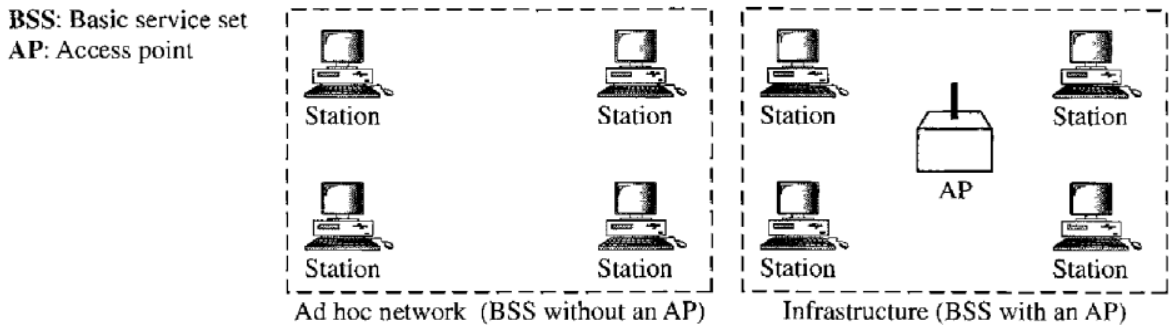
→Disadvantages of Ethernet:-The wired Ethernet network restricts you in terms of distances, and it is best for using in short distances.
-If you create a wired ethernet network that needs cables, hubs, switches, routers, they increase the cost of installation.
-Additionally, finding a problem is very difficult in an Ethernet network (if has), as it is not easy to determine which node or cable is causing the problem.

#Wireless LANS:-Wireless communication is one of the fastest-growing technologies. -The demand for connecting devices without the use of cables is increasing everywhere. -Wireless LANs can be found on college campuses, in office buildings, and in many public areas.
-wireless LAN technologies are **IEEE 802.11** wireless LANs, sometimes called wireless Ethernet and **Bluetooth** (a technology for small wireless LANs).

***IEEE 802.11:**-IEEE 802.11 standard, popularly known as WiFi or Wireless ethernet.
- WiFi or WLAN uses high-frequency radio waves instead of cables for connecting the devices in LAN.
-Users connected by WLANs can move around within the area of network coverage.

→802.11 Architecture:-There are two types of Services they are;

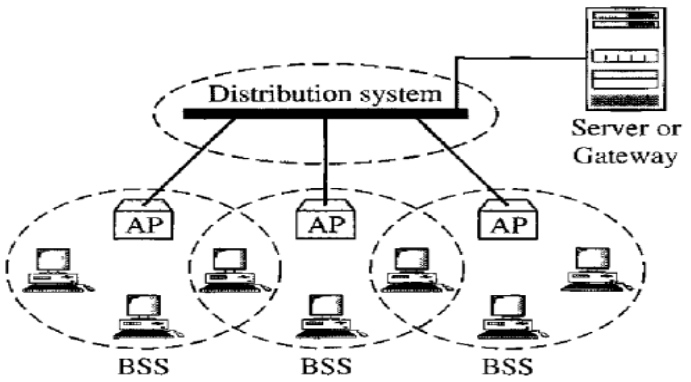
1. **Basic services set (BSS):**-IEEE 802.11 defines the basic service set (BSS) as the building block of a wireless LAN.
   -A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the **access point (AP).**



-The use of access point is optional.

-If the access point is not present, it is known as **stand-alone network** and cannot send data to other BSSs.
-This type of architecture is known as **ad hoc architecture.**
-The BSS in which an access point is present is known as an **infrastructure network**.

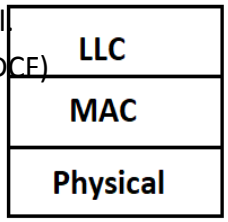
2. **Extended Service Set (ESS):**-An extended service set (ESS) is made up of two or more BSSs with APs.
   -In this case, the BSSs are connected through a distribution system, which is usually a wired LAN.
   -Or These extended networks are created by joining the access points of basic services sets through a wired LAN known as distribution system.
   -The distribution system connects the APs in the BSSs.



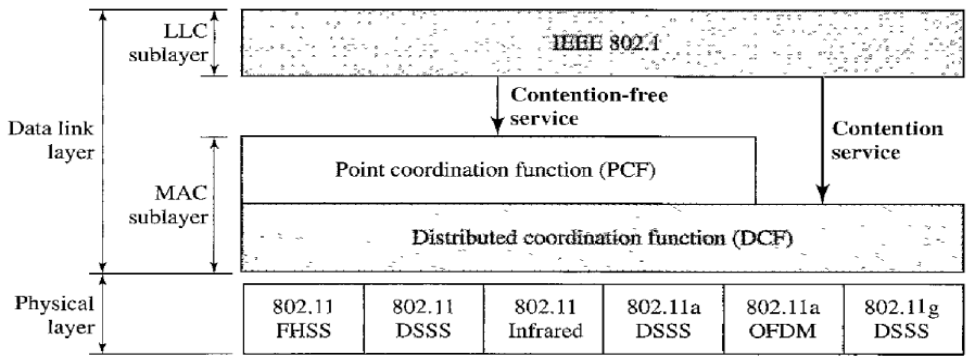
→802.11 Station Types:-IEEE 802.11 defines three types of stations based on their mobility (movement) in a wireless LAN they are;

1. **No-transition Mobility:**These types of stations are either stationary (not moving) or moving only inside a BSS.
2. **BSS-transition Mobility:**-These types of stations can move from one BSS to another and cannot change ESS.
3. **ESS-transition Mobility:**-These types of stations can move from one ESS to another.

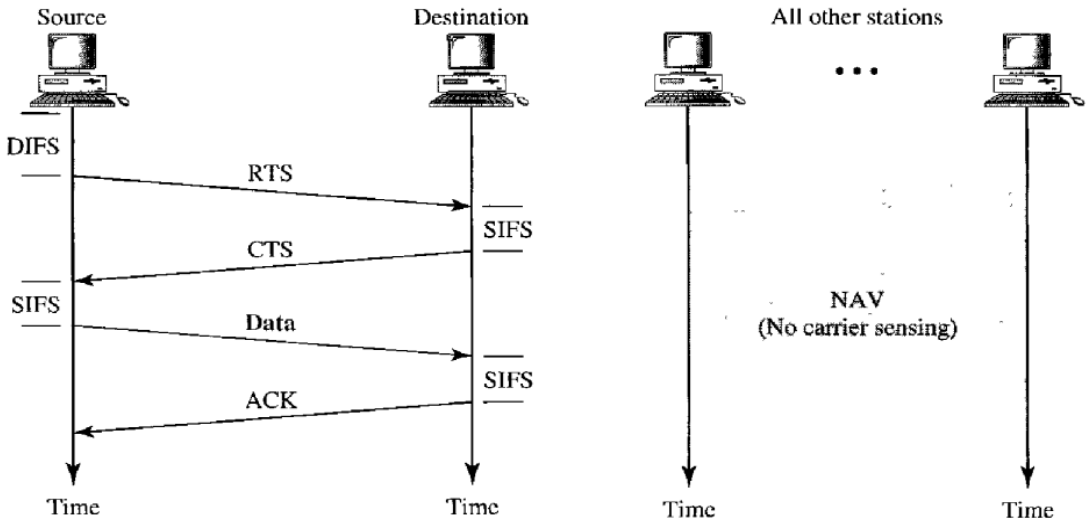
→MAC Sub layer:-The IEEE model have 3 layers instead of 7 layers in OSI model.



-IEEE 802.11 defines two MAC sublayers: the distributed coordination function (DCF) and point coordination function (PCF).
-Figure below shows the relationship between the two MAC sublayers, the LLC sublayer, and the physical layer.



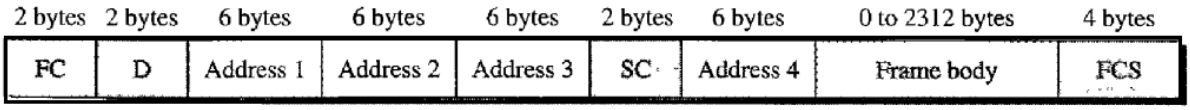
- **Distributed coordination function (DCF):-**The DCF is used in BSS having no access point.
  -DCF uses **CSMA/CA protocol for transmission.**



  -When a station wants to transmit, it senses the channel to see whether it is free or not.
  -If the channel is not free the station waits.
  -If the station finds a channel to be idle, the station waits for a period of time called **distributed interframe space (DIFS).**
  -The station then sends control frame called **request to send (RTS).**
  -The destination station receives the frame and waits for a short period of time called **short interframe space (SIFS).**
  -The destination station then sends a control frame called **clear to send (CTS)** to the source station.
  -This frame indicates that the destination station is ready to receive data.
  -The sender then waits for SIFS time and sends data.
  -The destination waits for SIFS time and sends **acknowledgement (ACK)** for the received frame.
  -802.11 standard uses **Network Allocation Vector (NAV)** for collision avoidance.

- **Point coordination function (PCF):**-The point coordination function (PCF) is an optional access method that can be implemented in an infrastructure network (not in an ad hoc network).
  -It is implemented on top of the DCF and IS used for time sensitive transmissions.
  -It is used to prevent collisions.
  -bakki google

→**Frame Format of 802.11:-**The MAC layer frame consists of nine fields.



1. Frame Control (FC). This is 2 byte field and defines the type of frame and some control information. This field contains several different subfields.
2. D. It stands for duration and is of 2 bytes. This field defines the duration for which the frame and its acknowledgement will occupy the channel.
3. Addresses. There are 4 address fields of 6 bytes length. These four addresses represent source, destination, source base station and destination base station.
4. Sequence Control (SC). This 2 byte field defines the sequence number of frame to be used in flow control.
5. Frame body. This field can be between 0 and 2312 bytes. It contains the information. FCS. This field is 4 bytes long and contains 'CRC -error detection sequence.

→IEEE 802.11 Frame types:-Frame Types A wireless LAN defined by IEEE 802.11 has three categories of frames they are:

1. **Management frames**:-It is used for initial communication between stations and access points.
2. **control frames** :-These are used for accessing the channel and acknowledging frames.
   -The control frames are RTS and CTS.
3. **data frames**:-These are used for carrying data and control information.

***IEEE 802.11 addressing mechanism:**-There are four different addressing cases depending upon the value of **To DS** And **from DS** it is the subfields of FC field.

To Ds-it indicate frame is going to distributed system.
From DS-It indicate frame is coming from distributed system.

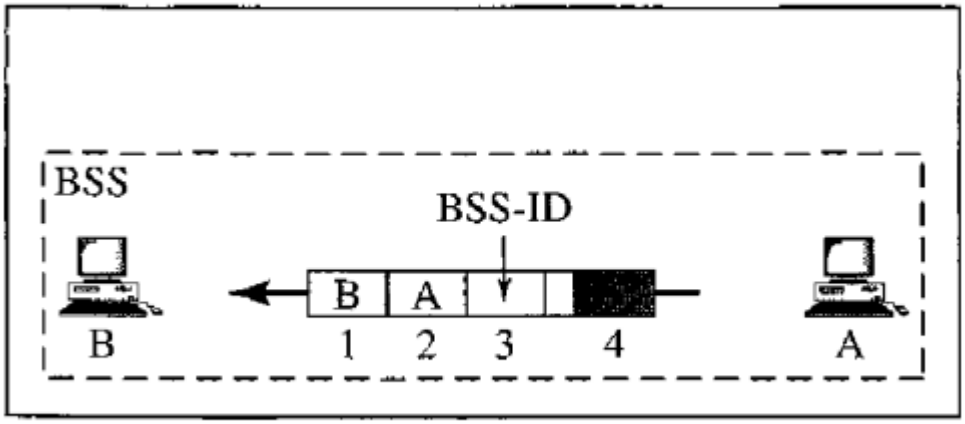
-Each flag can be either 0 or 1, resulting in four different situations.
-The table below specifies the addresses of all four cases.

To DS	*From* DS	*Address* 1	*Address* 2	*Address* 3	*Address* 4
0	0	Destination	Source	BSS ID	N/A
0	1	Destination	SendingAP	Source	N/A
1	0	Receiving AP	Source	Destination	N/A
1	1	Receiving AP	SendingAP	Destination	Source

-Address 1 is always the address of the next device.

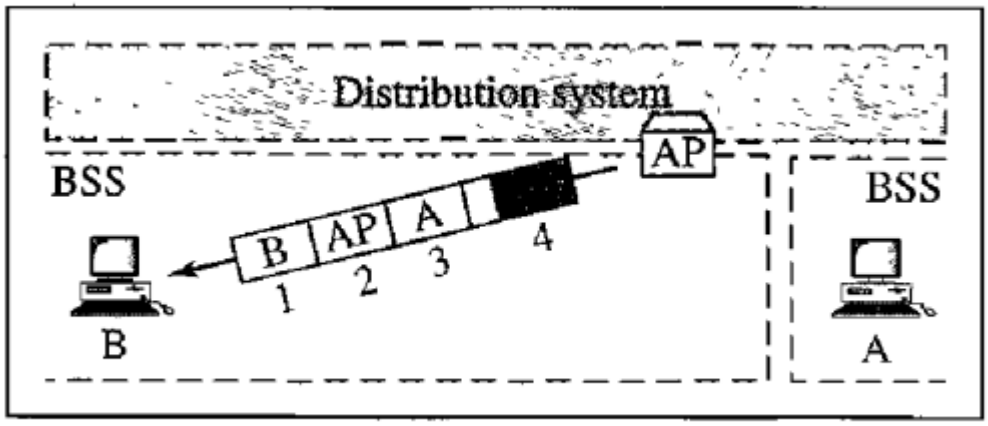
-Address 2 is always the address of the previous device.
-Address 3 is the address of the final destination station if it is not defined by address 1. -Address 4 is the address of the original source station if it is not the same as address 2.

- If To DS = 0 and From DS = 0, it indicates that frame is not going to distribution system and is not coming from a distribution system.
  -The frame is going from one station in a BSS to another.



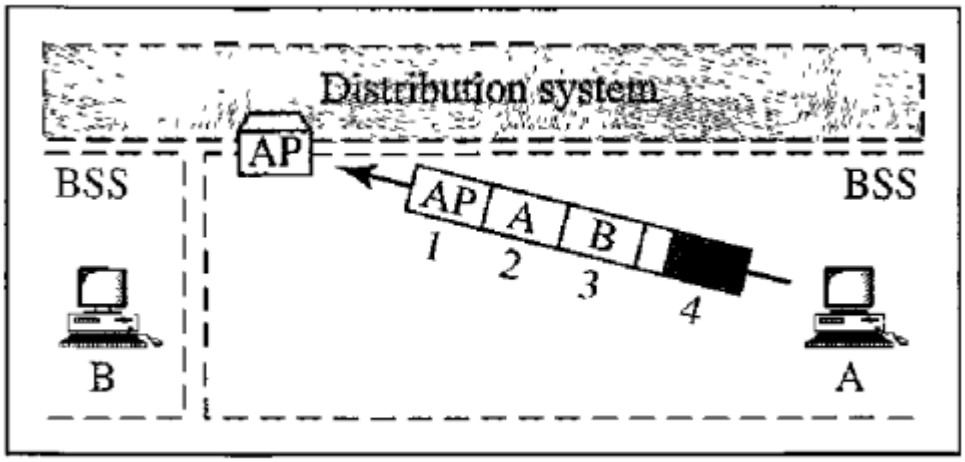
a. Case 1

- If To DS = 0 and From DS = 1, it indicates that the frame is coming from a distribution system.
  -The frame is coming from an AP and is going to a station.
  - The address 3 contains original sender of the frame (in another BSS).



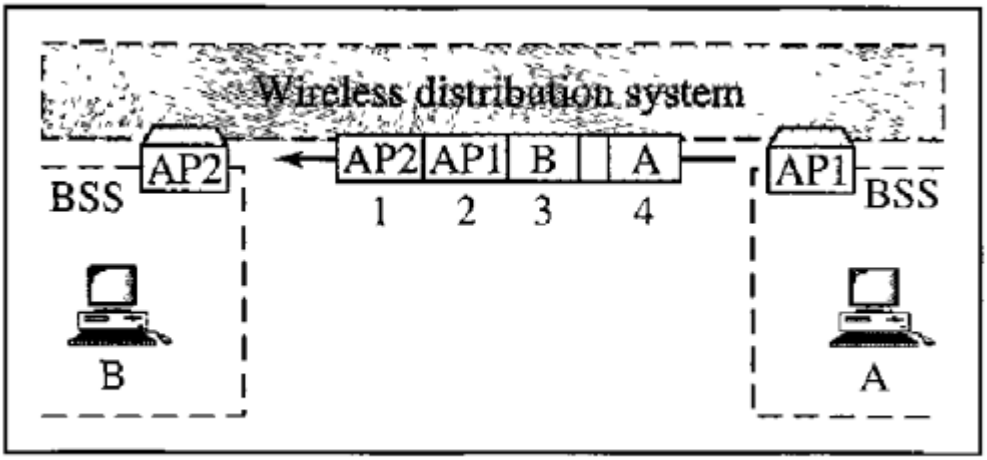
b. Case 2

- If To DS = 1 and From DS = 0, it indicates that the frame is going to a distribution system.
  -The frame is going from a station to an AP. The address 3 field contains the final destination of the frame.



c. Case 3

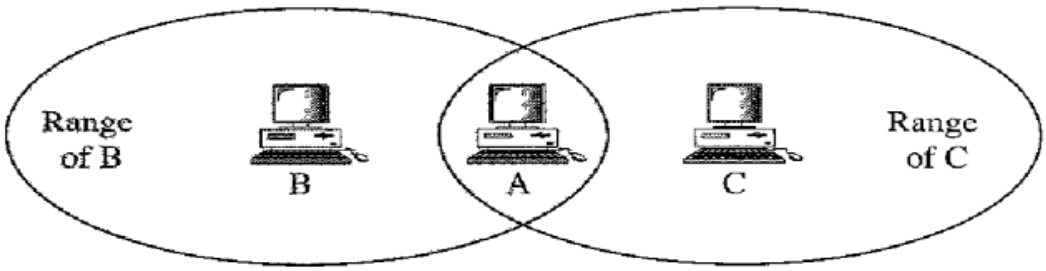
- If To DS = 1 and From DS = 1,it indicates that frame is going from one AP to another AP in a wireless distributed system.



d. Case 4

→Hidden and Exposed Station Problems

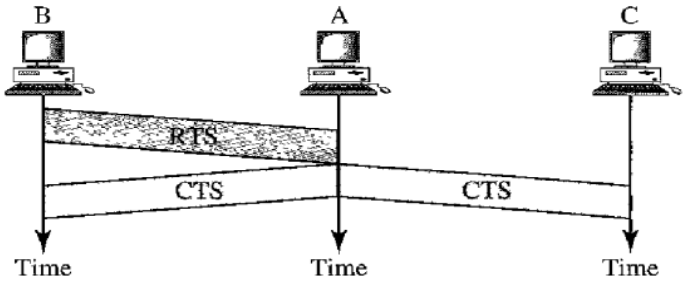
- **Hidden Station Problem:**-In hidden station problem theStation B has a transmission range shown by the left oval every station in this range can hear any signal transmitted by station B.
  -Station C has a transmission range shown by the right oval every station located in this range can hear any signal transmitted by C.
  -Station C is outside the transmission range of B; likewise, station B is outside the transmission range of C.
  -Station A,however, is in the area covered by both Band C; it can hear any signal transmitted by B or C.



B and C are hiddent from each other with respect to A.

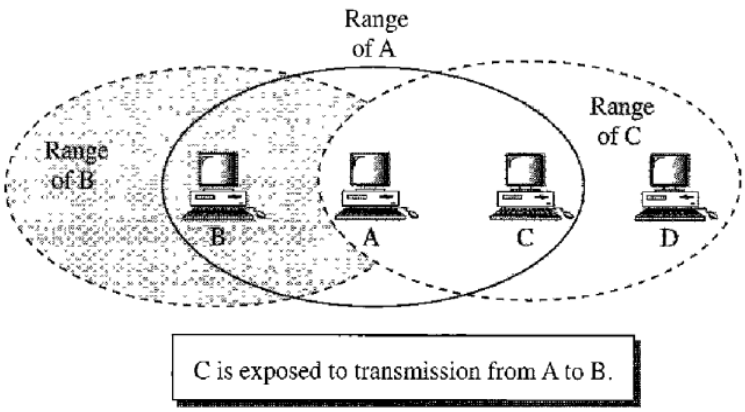
-Assume that station B is sending data to station A In middle of this transmission B and A, At that time station C also has data to send to station A.
-However, station C is out of B's range and transmissions from B cannot reach C. -Therefore C thinks the medium is free.
-Station C sends its data to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

-We can avoid this problem by using the handshake frames **(RTS and CTS)**



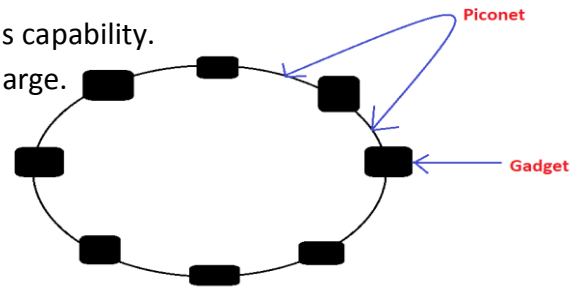
-Station B send a request (RST) then Station A send a clear (CTS) signal to Station B and Station C at same time.
-Then Station c knows that some hidden Station is occupying Station A.

- **Exposed station problem:**-The Exposed Station Problem is the inverse of Hidden Station Problem.



-Station A is transmitting to station B. Station C has some data to send to station D, - which can be sent without interfering with the transmission from A to B.
-However, station C is exposed to transmission from A.
-The handshaking messages RTS and CTS cannot help in this case,

***Bluetooth:**-Bluetooth is a wireless LAN technology designed to connect devices of different functions such as computers (desktop and laptop), cameras, printers, and so on.
-A Bluetooth LAN is an ad hoc network.
-Bluetooth simply follows the principle of transmitting and receiving data using radio waves.
-It can be paired with the other device which has also Bluetooth but it should be within the estimated communication range to connect.
-When two devices start to share data, they form a network called **piconet**
-Or different connection of devices is called **piconet.**
-Each device in piconet is called **gadget**.
-A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability.
-A Bluetooth LAN, by nature, cannot be large.
-Bluetooth technology has several Applications they are;

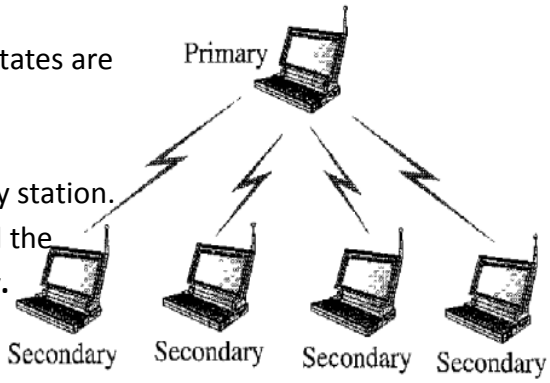


- Peripheral devices such as a wireless mouse or keyboard can communicate with the computer through this technology.
- Home security devices can use this technology to connect different sensors to the main security controller.
- Health care etc…

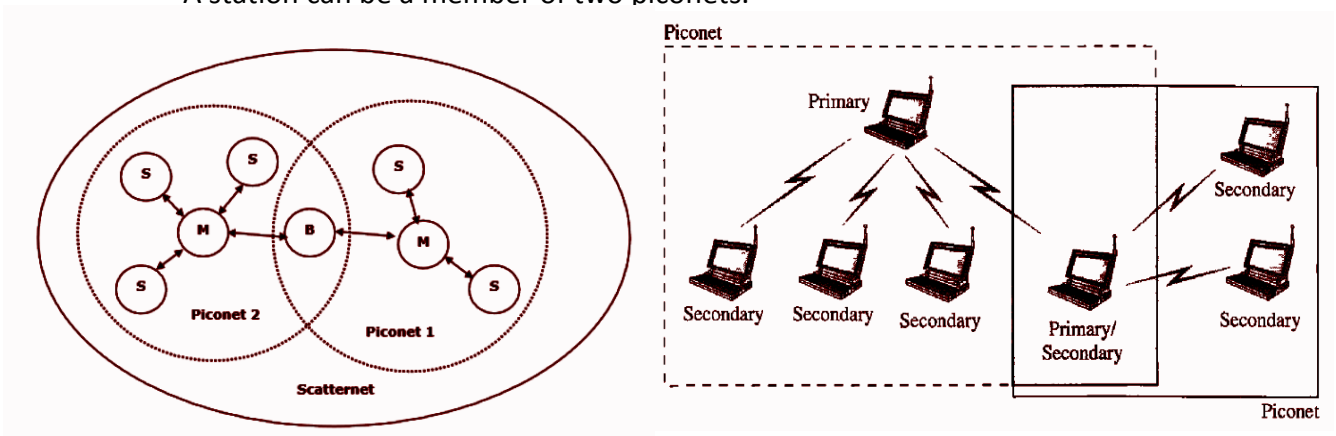
→Architecture of Bluetooth
-Bluetooth defines two types of networks: piconet and scatternet.

1. **piconet :**-A Bluetooth network is called a piconet, or a small net.
   -A piconet can have up to eight stations.

-One station act as primary state and other 7 states are secondary states.
-It is half duplex (walkie-talkie).
-Note that a piconet can have only one primary station.
-The communication between the primary and the secondary can be **one-to-one** or **one-to-many.**
-Possible communication is only between the primary and secondary ;
secondary-secondary communication is not possible until it is moved from parked state to active state.

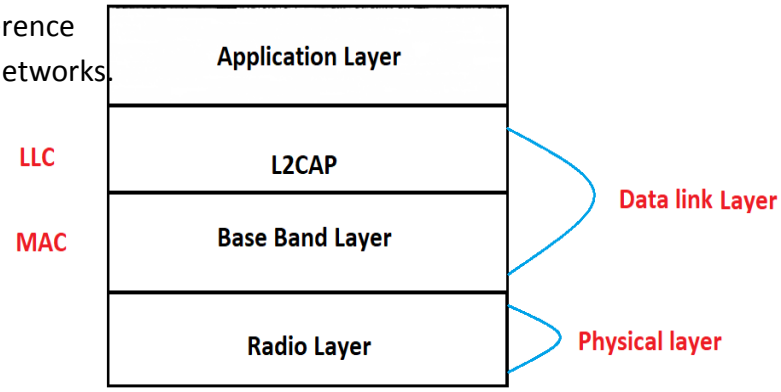


2. **Scatternet :**-Piconets can be combined to form what is called a scatternet. A secondary station in one piconet can be the primary in another piconet.
-This station can receive messages from the primary in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in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

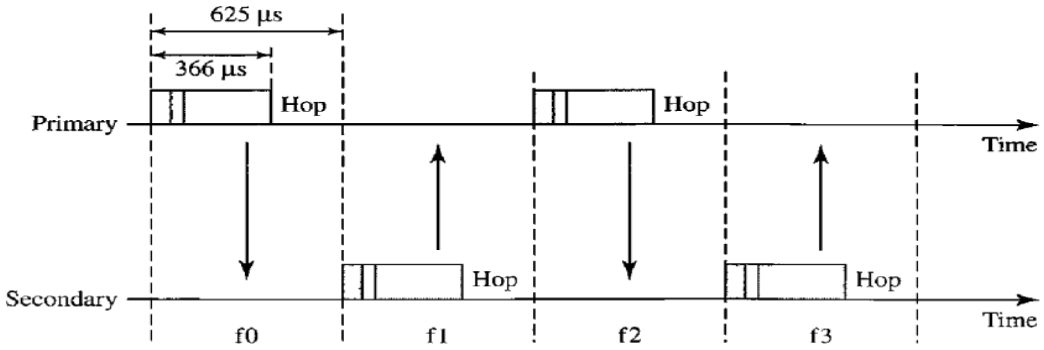


→Bluetooth Layers:-Bluetooth uses several layers they are;

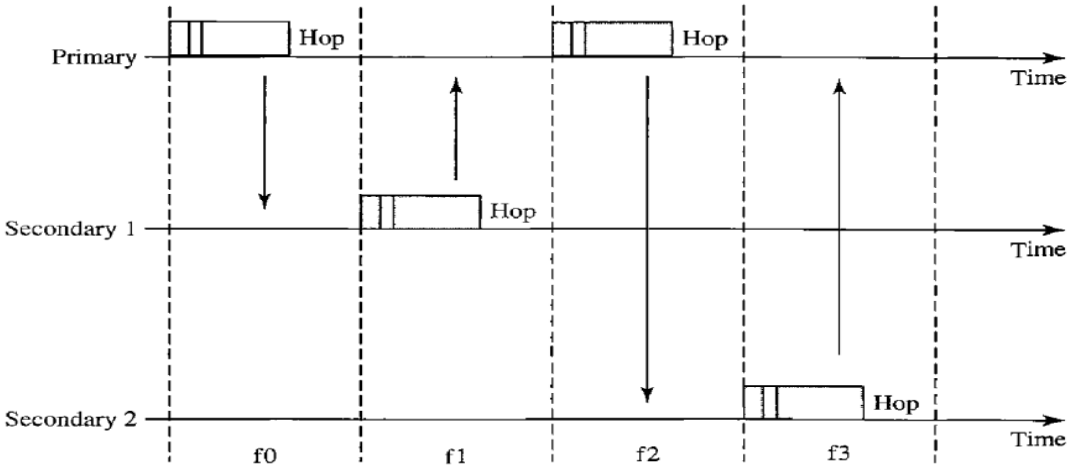
1. **Radio layer:**-The radio layer is roughly equivalent to the physical layer.
-Bluetooth devices are low-power and have a range of 10 m.
-Bluetooth uses the **frequency-hopping spread spectrum (FHSS)** method in the physical layer to avoid interference from other devices or other networks.
-To transform bits to a signal, Called **GFSK.**



2. **Base Band Layer:**-The baseband layer is roughly equivalent to the MAC sublayer in LANs.
   -Duration of one communication is 625 µs.
   -If we have a primary and a secondary state then we place primary as 0,2,4 as even numbers and secondary as odd numbers is called **Single secondary communication.**
   -Single secondary communication fig below.



   -Suppose we have multiple secondary state.
   -Then primary time slot '0'.
   -When primary communicate with secondary then it goes to primary again then to next secondary This is called **Multiple secondary state.**
   -Multiple secondary state fig below



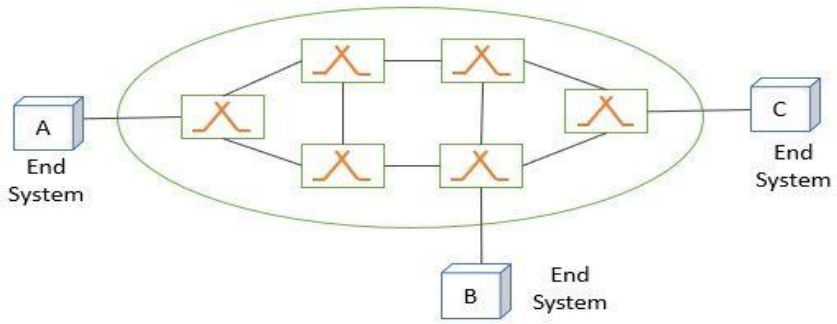
   -**Two types of links** can be created between a primary and a secondary:
   - **SCO links** :-A synchronous connection-oriented (SeQ) link
     -It is used when avoiding latency (delay in data delivery) is more important than integrity (error-free delivery).
     -If error occur then it have no retransmission.
   - **ACL links**:-An asynchronous connectionless link (ACL)
     -It is used when data integrity is more important than avoiding latency.

3. **L2CAP:**-The Logical Link Control and Adaptation Protocol, or L2CAP

-It is roughly equivalent to the LLC sublayer in LANs.
-It is used for data exchange on an ACL link; SCQ channels do not use L2CAP.
-The L2CAP can do multiplexing.
-At the sender site, it accepts data from one of the upper-layer protocols, frames them, and delivers them to the baseband layer.
-At the receiver site, it accepts a frame from the baseband layer, extracts the data, and delivers them to the appropriate protocol layer.

→Frame format of Bluetooth:- google

#Switched WANS:-Switched WAN network is a network that has several end nodes.
-In the switched WAN network, the data sent from a source node is routed to the destination node by being switched from one node to another in the network.
-Switched WAN networks are suitable for long-distance transmission.



Switched WAN

-Switched WAN network uses a connection-oriented technology where the switches are used to establish the path between source and destination node.
-If a source node wants to send data to the destination node at first a path is established between them and then the data is transmitted over the established path.
-Once the communication is over the path between the sender and receiver is terminated.
-It used Packet switching technology.
-There are 2 types;

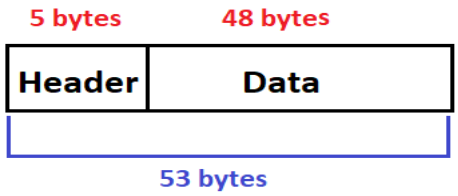
1. **X.25:**-X.25 is the first switched WAN which was introduced in the year 1970.
   -It can only send 64kbps data in one second. //drawback
   -It is expensive.
   -It works on network layer.
   -It is implemented using virtual circuit switching.
   -It uses **Frame relay protocol.**
   Flow control and error control is in minimum level.

2. **ATM:**-Asynchronous Transfer Mode (ATM).

-It is the cell relay protocol designed by the ATM Forum and adopted by the ITU-T. -The combination of ATM and SONET will allow high-speed interconnection of all the world's networks.

-The size of the cell is 53 bytes.

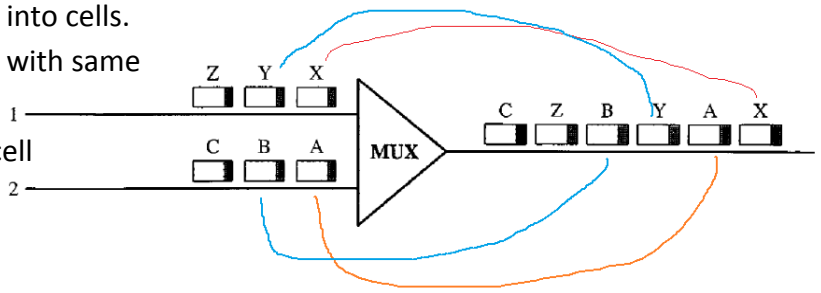
-And 48 bytes are used by data and 5 bytes are used by header section.

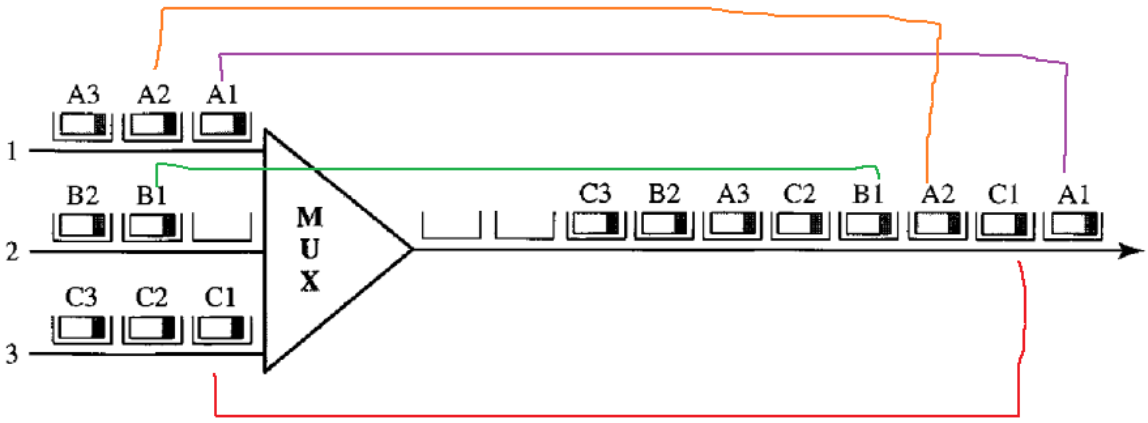


-First divide message into cells.

-Then pass each cells with same path.

-And again pass the cell with the same path.

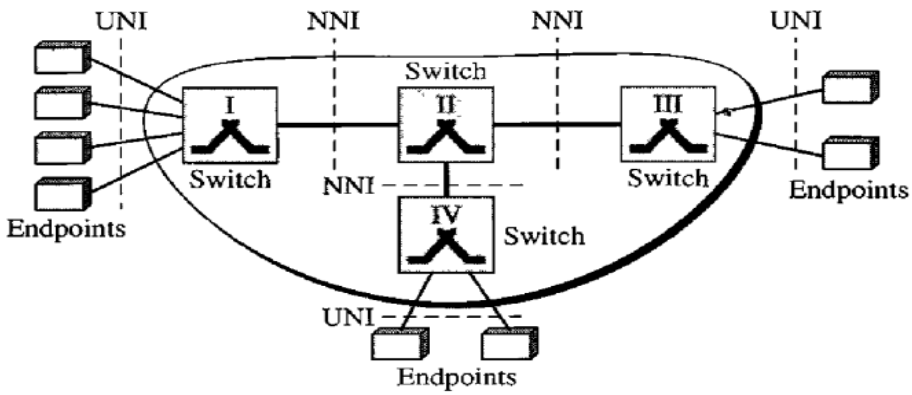


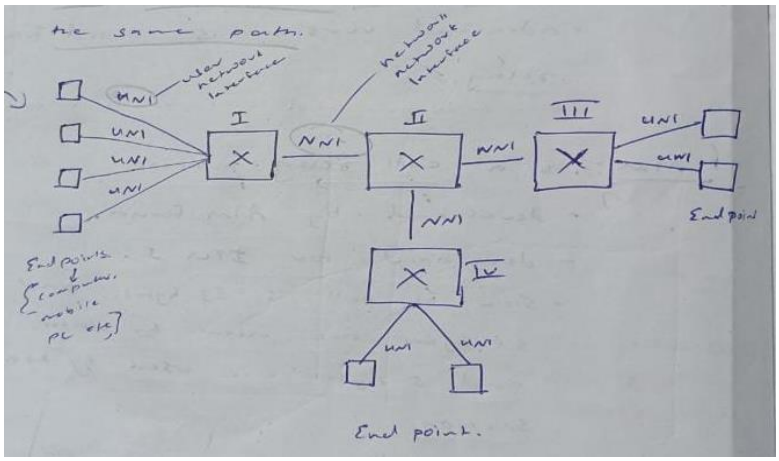


→ATM Architecture :-ATM is a cell-switched network.

-The user is called the endpoints And user to switch connected by UNI (user-to-network interface).

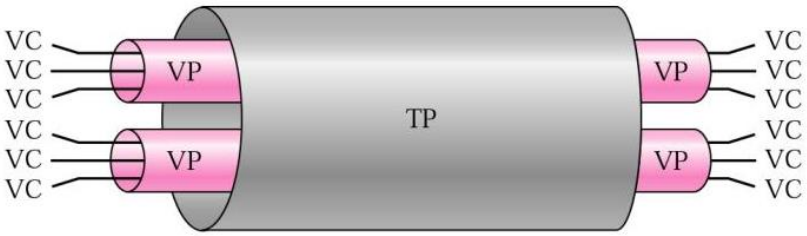
-The switch to switch connected by NNI (network-to-network interfaces).



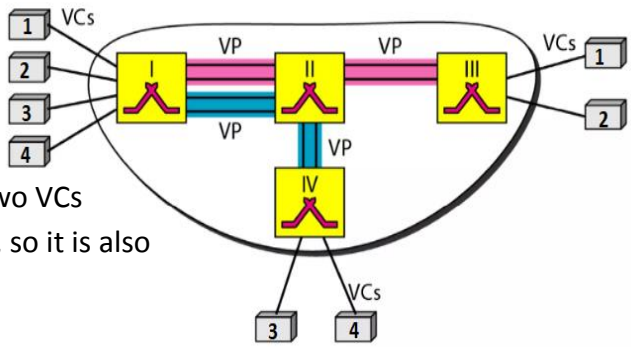


-Connection between two endpoints is accomplished through

- **Transmission paths (TP):**-A transmission path (TP) is the physical connection (wire, cable, satellite, and so on) between an endpoint and a switch or between two switches.
  -It is the collection of all path .
  -suppose we have p1 ,p2,p3,p4 paths to travel from A to B.
  -Then p1,p2,p3,p4 are called TP.
- **Virtual paths (VP)**:-A transmission path is divided into several virtual paths.
  -A single path for data translation called VP.
  -A virtual path (VP) provides a connection or a set of connections between two switches.
- **Virtual circuits (VC):**-A single message path between source and destination.

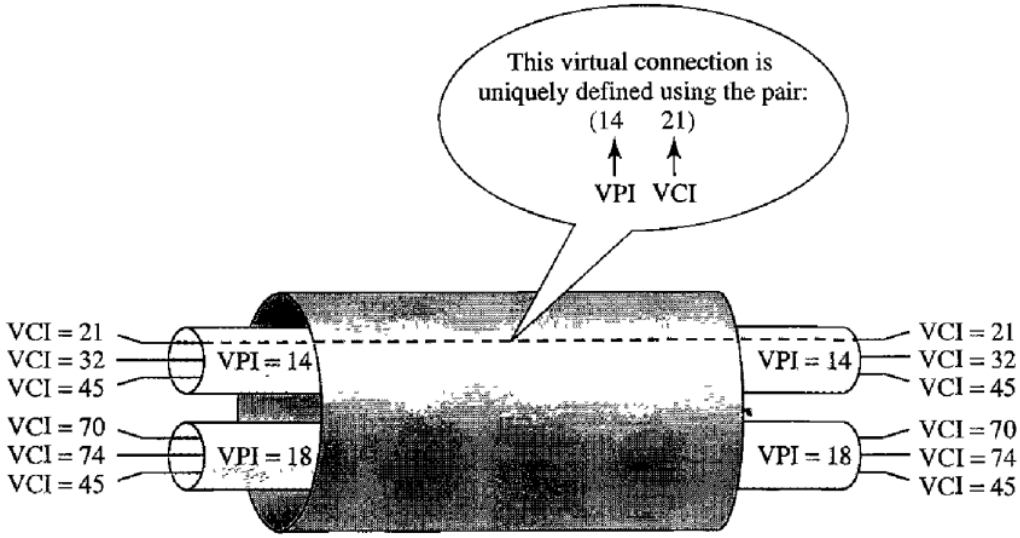


>Example:-In the below figure, There are four VCs.
-However, the first two VCs seem to share the same virtual path from switch I to switch III, so it is reasonable to bundle (combine) these two VCs together to form one **VP**.
- On the other hand, it is clear that the other two VCs share the same path from switch I to switch IV, so it is also reasonable to combine them to form one **VP**.

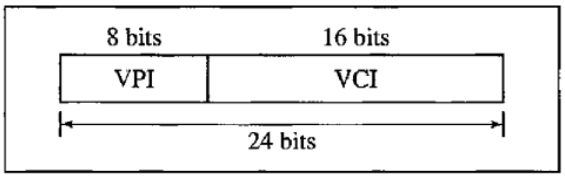


-In a virtual circuit network, to route data from one endpoint to another, the virtual connections need to be identified.

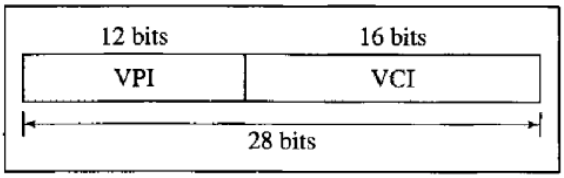
-VP and VC identifier called VPI and VCI.
-The below Figureshows the VPIs and VCIs for a transmission path.



-The lengths of the VPIs for UNIs and NNIs are different.
-In a UNI, the VPI is 8 bits, whereas in an NNI, the VPI is 12 bits.
-The length of the VCI is the same in both UNI and NNI.



a. VPI and VCI in a UNI

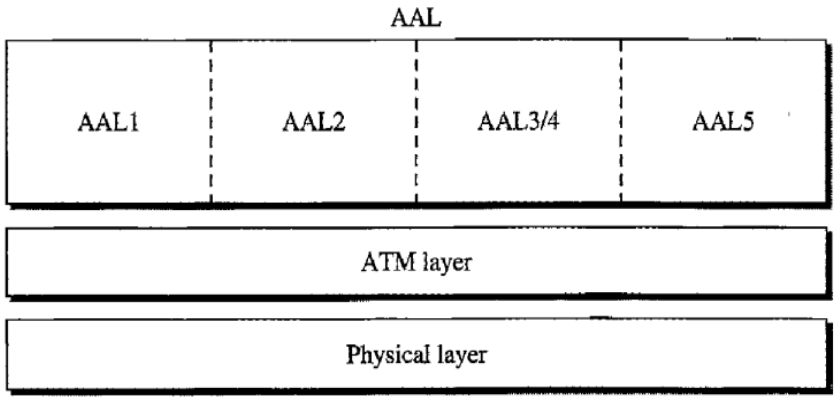


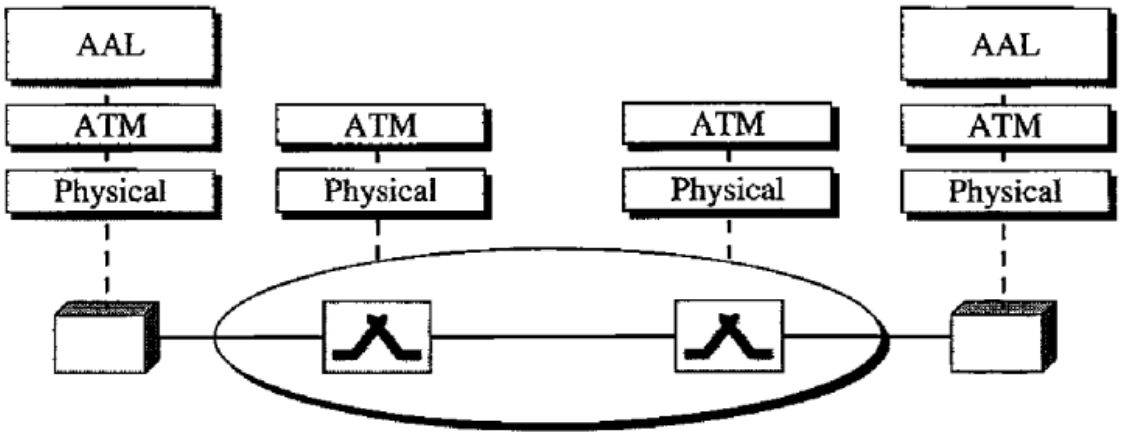
b. VPI and VCI in an NNI

→ATM Layers:-The ATM standard defines three layers.

1. **Application adaptation layer(AAL) :-** AAL1, AAL2, AAL3/4, AAL5.
2. **ATM layer:-**Layer int a explain google ill nokkanam
3. **physical layer:-**

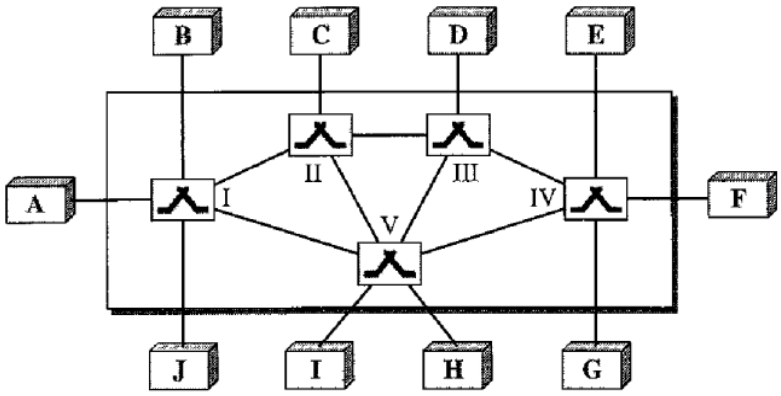
-The endpoints use all three layers while the switches use only the two bottom layers.
-Fig below.



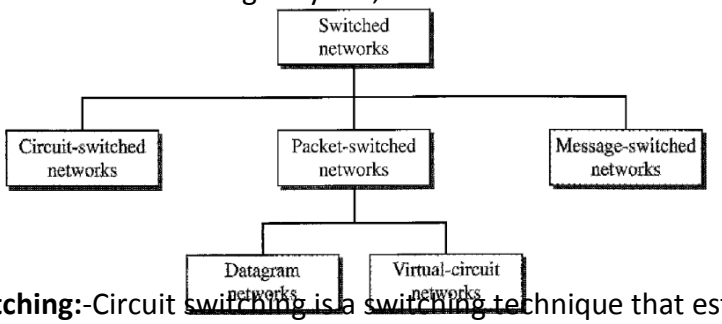


#Network Layer:-mukalill explain cheythittunde.

***Switching:**-. A switched network consists of a series of interlinked nodes, called switches.
-Switches are devices capable of creating temporary connections between two or more devices linked to the switch.
-In a switched network, some of these nodes are connected to the end systems (computers for example),Others are used only for routing.

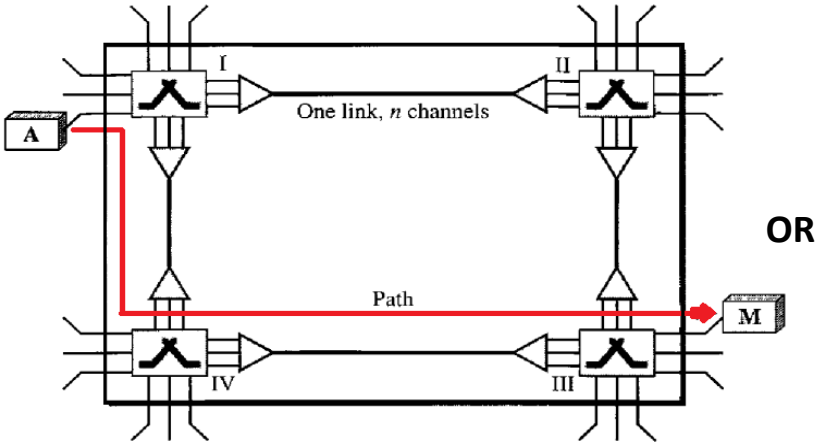


-In large networks, there can be multiple paths from sender to receiver.
-The switching technique will decide the best route for data transmission.
-Switching technique is used to connect the systems for making one-to-one communication.
-The end systems (communicating devices) are labelled A, B, C, D, and so on, and the switches are labelled I, II, III, IV, and V. Each switch is connected to multiple links.
-There are three methods of switching they are;

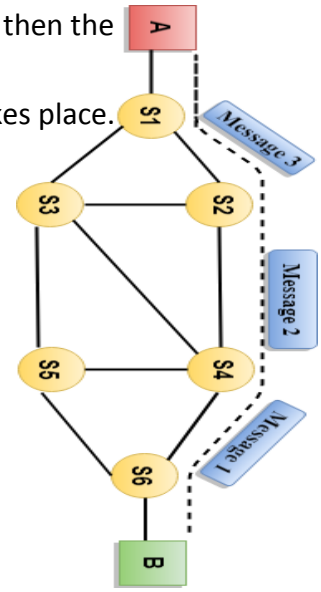


1. **Circuit switching:**-Circuit switching is a switching technique that establishes a dedicated path between sender and receiver.

-In the Circuit Switching Technique, once the connection is established then the dedicated path will remain to exist until the connection is terminated.
-A complete end-to-end path must exist before the communication takes place.



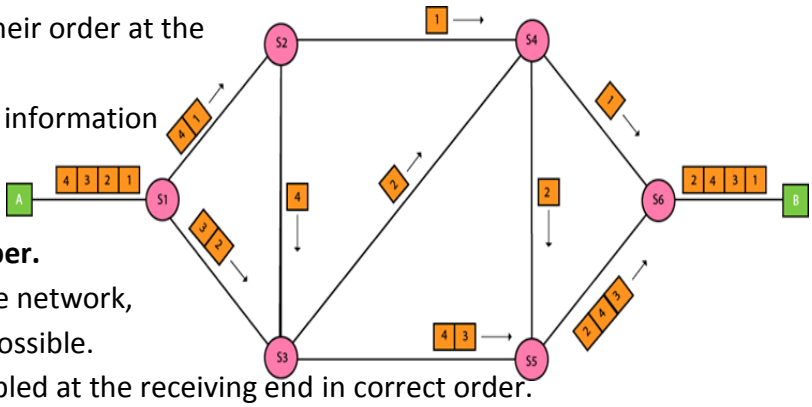
OR



-The circuit switching has 3 phases for communication they are;

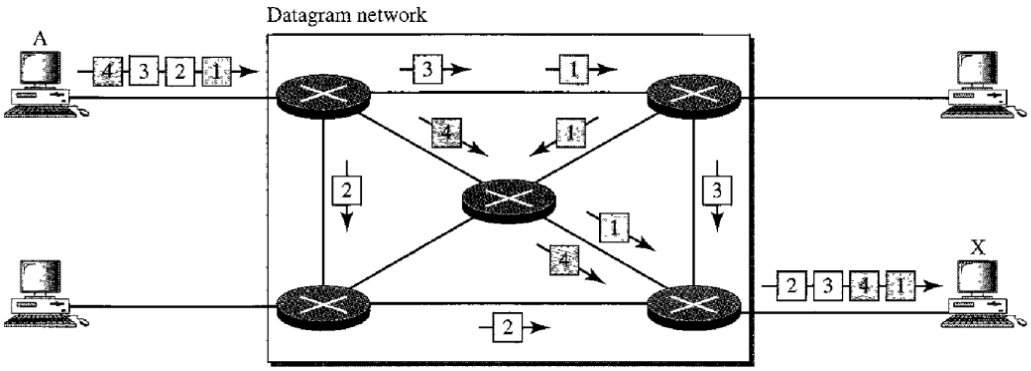
- Step up :-First made a connection between A and M (I→IV→III).
- Data transfer :-Then transfer the data.
- Tear down :- After translation of data then remove the connection.

2. **Packet switching**:-Packet switching is a method of transferring data to a network in form of packets.
-The packet switching is a switching technique in which the message is sent by dividing 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.**

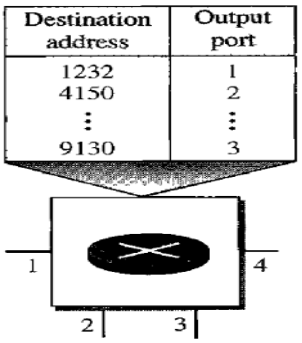


-Packets will travel across the network, taking the shortest path as possible.
-All the packets are reassembled at the receiving end in correct order.
-If any packet is missing or corrupted, then the message will be sent to resend the message.
-If the correct order of the packets is reached, then the acknowledgment message will be sent.
-There are two approaches/ further divide to Packet Switching they are;

- **Datagram networks:**-we need to send messages from one end system to another.
  -If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size.
  -Each packet is known as a **datagram**, and is considered as an independent.
  -Each packet contains the information about the destination and switch uses 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.**
  -Fig of datagram network with four switches (routers)



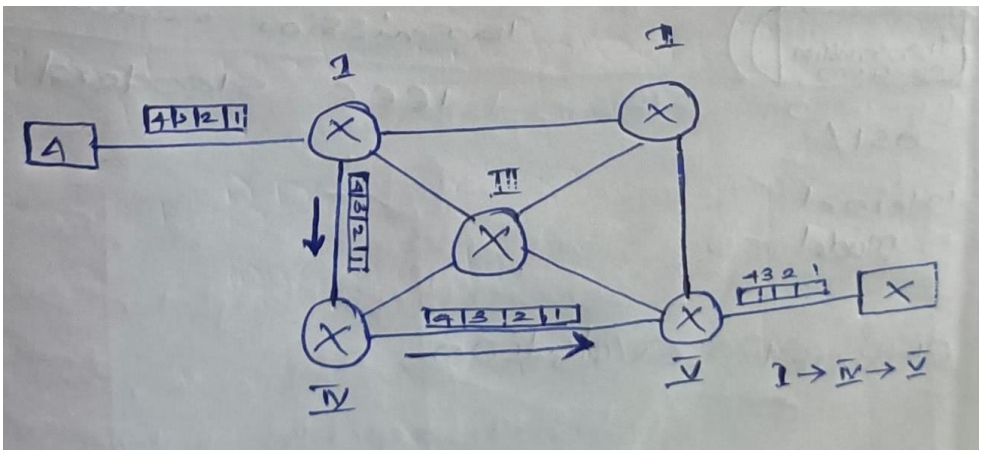
Note:
-If there are no setup or teardown phases, how are the packets routed to their destinations in a datagram network?
-In this type of network, each switch (or packet switch) has a routing table which is based on the destination address.
-The routing tables are dynamic and are updated periodically.
-The destination addresses and the corresponding forwarding output ports are recorded in the tables.



- **Virtual-circuit networks:**- A virtual-circuit network is a combination of circuit-switched network and a datagram network.
  -It has some characteristics of both.

-in circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
-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.
-It also have 3 phases they are;

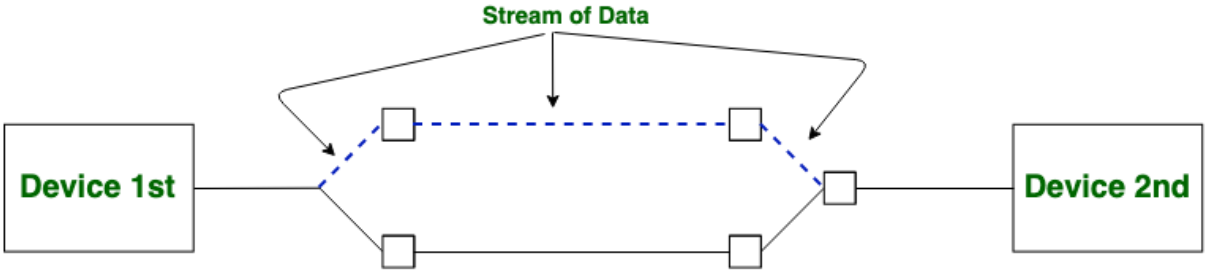
- Setup:-In setup phase first establish a connection / select a path to send the packages.
- data transfer:-After setup phase transfer the data.
- Teardown:-after completion of data transfer then remove the connection.



3. **Message switchin**g:-Ethu explain cheyandaa

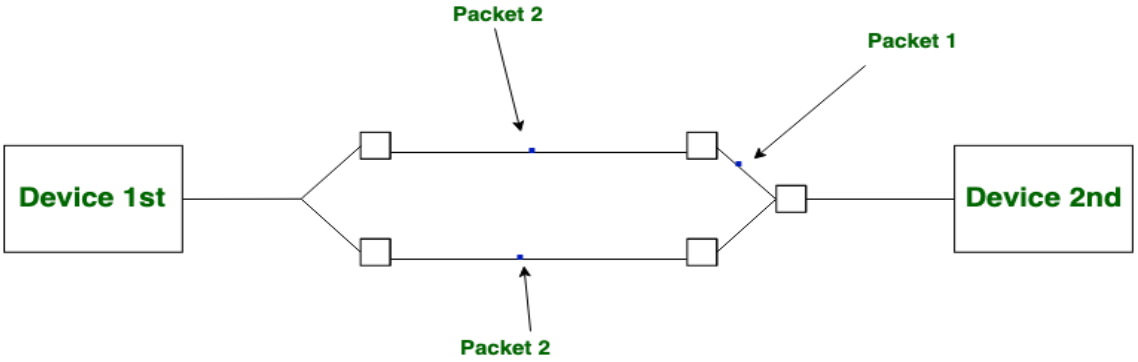
***connection oriented and connectionless service:**-Both Connection-oriented service and Connection-less service are used for the connection establishment between two or more two devices.
-In Connection-oriented service involves the creation and termination of the connection for sending the data between two or more devices.
- In connectionless service does not require establishing any connection and termination process for transferring the data over a network.

→Connection-oriented service:- In is related to the telephone system.
-It includes connection establishment and connection termination.
-In a connection-oriented service, the **Handshake method** is used to establish the connection between sender and receiver.
-Or It uses a handshake method that creates a connection between the user and sender for transmitting the data over the network.
- In connection-oriented service, packets are transmitted to the receiver in the same order the sender has sent them.



-After successfully exchanging or transmitting data, a sender can terminate the connection by sending a signal to the receiver.

→Connection-less service:-In is related to the postal system, where each letter takes along different route paths from the source to the destination address.
-It does not include any connection establishment and connection termination. -Connection-less Service does not give a guarantee of reliability.
-In this, Packets do not follow the same path to reach their destination.
-So it does not require establishing a connection before sending the data from the sender to the receiver.
-It is not a reliable network service because it does not guarantee the transfer of data packets to the receiver, and data packets can be received in any order to the receiver.
-In connectionless service, the transmitted data packet is not received by the receiver due to network congestion, and the data may be lost.



-For example, a sender can directly send any data to the receiver without establishing any connection because it is a connectionless service.
-Data sent by the sender will be in the packet.
-In connectionless service, the data can be travelled and received in any order.
-However, it does not guarantee to transfer of the packets to the right destination.

→Difference between Connection-oriented Service and Connection Less Service

Connection-oriented Service	**Connection Less Service**
It is designed and developed based on the telephone system.	It is service based on the postal system
It is used to create an end to end connection between the senders to the receiver before transmitting the data	It is used to transfer the data packets between senders to the receiver without creating any connection.
It creates a virtual path between the sender and the receiver.	It does not create any virtual connection or path between the sender and the receiver.
It requires authentication before transmitting the data packets to the receiver.	It does not require authentication before transferring data packets.
All data packets are received in the same order as those sent by the sender.	Not all data packets are received in the same order as those sent by the sender.
It is a more reliable connection service because it guarantees data packets transfer from one end to the other end with a connection.	It is not a reliable connection service because it does not guarantee the transfer of data packets from one end to another for establishing a connection.

Module-3

#Network Layer:-The network layer is responsible for the source-to-destination delivery of a packet.
-The network layer is responsible for the delivery of individual packets from the source to the destination host.
-The network layer adds a header that includes the logical addresses of the sender and receiver to the packet that came from the upper layer.

***IP addressing:**-An IP stands for internet protocol. An IP address is assigned to each device connected to a network.
-Each device uses an IP address for communication.
-It also behaves as an identifier as this address is used to identify the device on a network.
-An IP address is assigned to each device so that the device on a network can be identified uniquely.
-An IP address consists of two parts, i.e The first one is a network address, and the other one is a host address.
-The main types of devices that required an IP address included network devices, such as computers, servers, routers, and printers.
-There are two types of IP addresses they are;

1. IPv4
2. IPv6

1.IPV4 addressing:-IP stands for Internet Protocol and v4 stands for Version Four (IPv4).
-An IPv4 address is a **32-bit address**.
-Its addresses are 32-bit integers which will be expressed in decimal notation.
Example- 192.0.2.126
-IPv4 addresses are **unique.**
-They are unique in the sense that each address defines one, and only one, connection to the Internet.
-Two devices on the Internet can never have the same address at the same time.
-An address may be assigned to a device for a time period and then taken away and assigned to another device.
-The IPv4 addresses are **universal** in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

→IPV4 addressing notations:-There are two prevalent notations to show an IPv4 address:

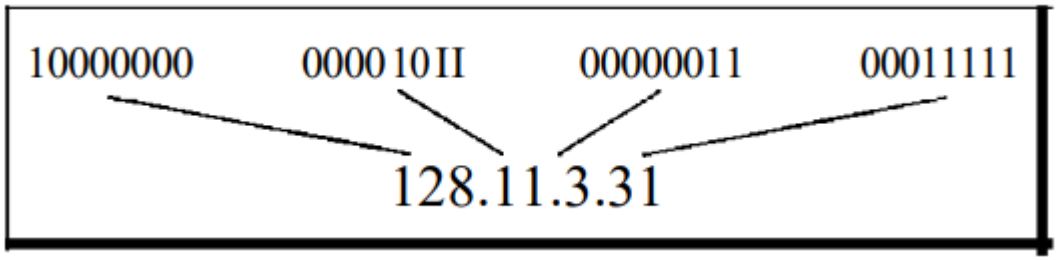
- Binary notation :-In binary notation, the IPv4 address is displayed as 32 bits.
  -it have 8 bits of 4 sections.
  -Each 8 bit is separated by space not use any dot or special operators.

  01110101 10010101 00011101 00000010

- Dotted decimal notation :-To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form.
  -Here we use dots to separate the bytes.
  -The following is the dotted decimal notation of the above address:

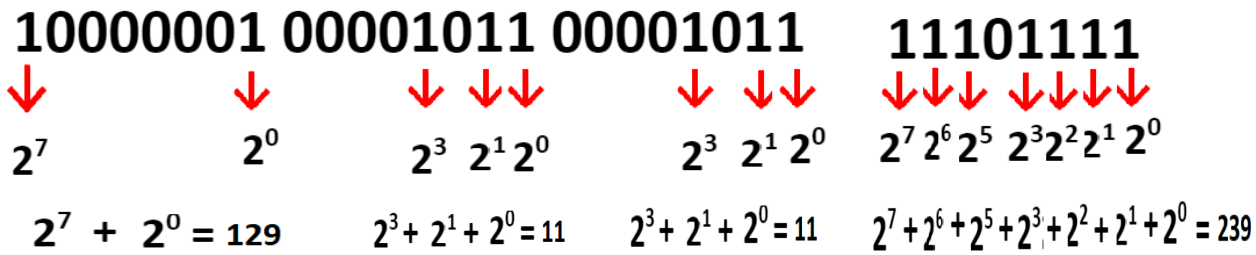
  117.149.29.2

  -Each number in dotted-decimal notation is a value ranging from **0 to 255**.



Q)Convert the following IPv4 addresses from binary notation to dotted-decimal notation ?

a. 10000001 00001011 00001011 11101111

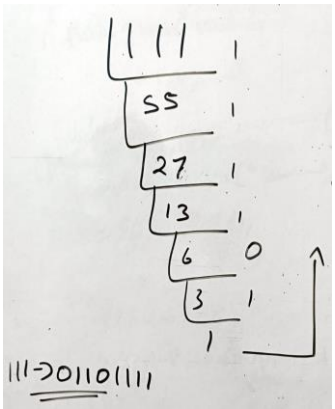


-We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.
-so final answer is **129.11.11.239**

b. 11000001 10000011 00011011 11111111
-its answer is 193.131.27.255

Q)Convert the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78 → 01101111 00111000 00101101 01001110
b. 221.34.7.82 → 11011101 00100010 00000111 01010010



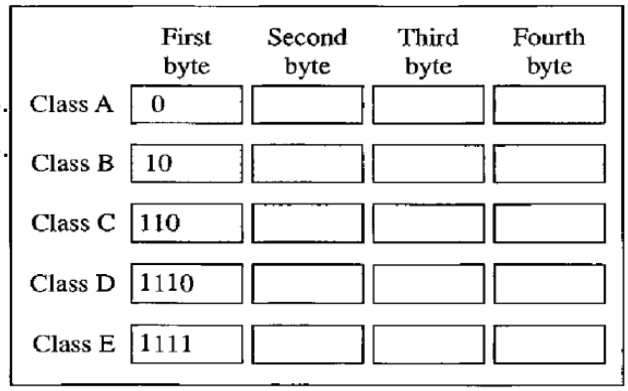
Q)Find the error, if any, in the following IPv4 addresses.
a. 111.56.**045**.78
-There must be no leading zero (045).
b. 221.34.7.8.20
-There can be no more than four numbers in an IPv4 address.

c. 75.45.**301**.14

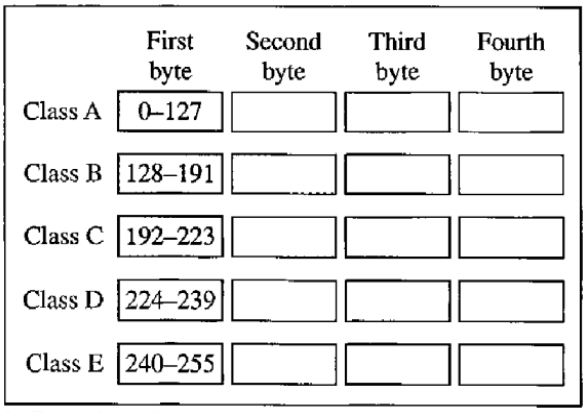
-Each number needs to be less than or equal to 255 (301 is outside this range)
d. 11100010.23.14.67
-A mixture of binary notation and dotted-decimal notation is not allowed.

→**Classful Addressing in IPV4 addressing**:-In classful addressing, the address space is divided into five classes:A, B, C, D, and E.
-We can find the class of an address when given the address in binary notation or dotted-decimal notation.
-If the address is given in binary notation, the first few bits can immediately tell us the class of the address.
-If the address is given in decimal-dotted notation, the first byte defines the class.
-eg:

-1st byte have '0' then it is in class A.
-when 1st byte have '10' then it is class B.
-when 1st byte have '10' then it is class B.
-when 1st byte have '110' then it is class C.
-when 1st byte have '1110' then it is class D.
-when 1st byte have '1111' then it is class B.



a. Binary notation



b. Dotted-decimal notation

-when its first byte starts from 0 to 127 then it is in class A.
-when its first byte starts from 128 to 191 then it is in class B.

Q)Find the class of each address.
a. 00000001 00001011 00001011 11101111
-The first bit is O. This is a class A address.
b. 11000001 10000011 00011011 11111111
-The first 2 bits are 1; the third bit is O. This is a class C address.

c. 14.23.120.8

-The first byte is 14 (between 0 and 127); the class is A.
d. **252**.5.15.111
-The first byte is 252 (between 240 and 255); the class is E.

-In classful addressing, a large part of the available addresses were wasted.
-In classful addressing, an IP address in class A, B, or C is divided into **netid** and **hostid**.
-The netid is in color, the hostid is in white.
-Note that the concept does not apply to classes D and E.
-In class A, one byte defines the netid and three bytes define the hostid.
-In class B, two bytes define the netid and two bytes define the hostid.
-In class C, three bytes define the netid and one byte defines the hostid.

Class	*Binary*	*Dotted-Decimal*
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

> Mask or Default mask :-Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous Is followed by contiguous 0s.
-The concept does not apply to classes D and E.
-The mask can help us to find the netid and the hostid.
-For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

-The last column of above Table shows the mask in the form **/n** where n can be 8, 16, or 24 in classful addressing.
-This notation is also called **slash notation or Classless Interdomain Routing (CIDR) notation**.

>**Subnetting**:- During the era of classful addressing, subnetting was introduced.
-If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets)
-or, in rare cases, share part of the addresses with neighbors.

→**Classless Addressing in IPV4 addressing**:-To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented.

-In this scheme, there are no classes.

-This technique assigns a block of IP addresses based on specified conditions when the user demands a specific amount of IP addresses.

-This block is known as a "**CIDR block**", and it contains the necessary number of IP addresses.

-When allocating a block, classless addressing is concerned with the following three rules.

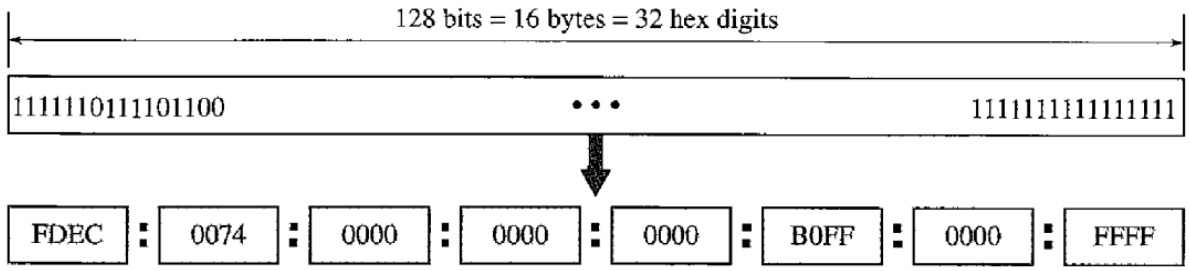
- The addresses in a block must be contiguous, one after another.
- The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).
- The first address must be evenly divisible by the number of addresses

2.IPv6 addressing:-IPv6 address consists of 16 bytes (octets); it is 128 bits long.

-To make addresses more readable, IPv6 specifies hexadecimal colon notation.

-In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits.

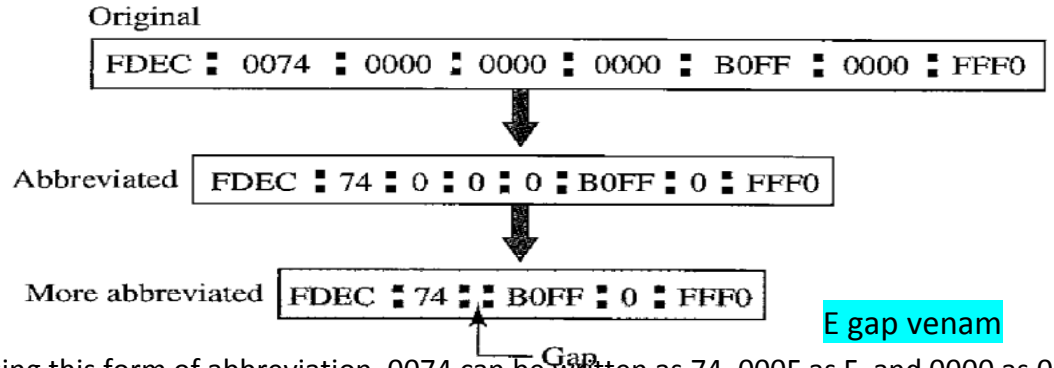
-Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.



-Although the IP address, even in hexadecimal format, is very long, many of the digits are zero's.

-In this case, we can abbreviate the address.

-Then the leading zeros of a section (four digits between two colons) can be omitted. -Only the leading zeros can be dropped, not the trailing zeros.



E gap venam

-Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0.

-Note that 3210 cannot be abbreviated.

→**Address Space in Ipv6**:-IPv6 has a much larger address space; 2^{128} addresses are available.

-The designers of IPv6 divided the address into several categories.

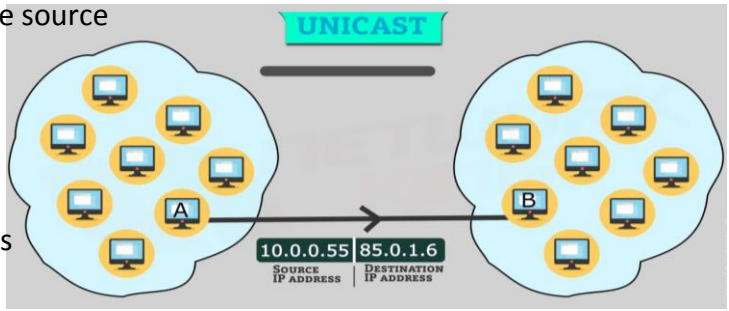
- A few leftmost bits, called the **type prefix.**

-The type prefix can easily be determined.

Type Prefix	*Type*
0000 0000	Reserved
0000 0001	Unassigned
0000 001	ISO network addresses
0000 010	IPX (Novell) network addresses
0000 011	Unassigned
0000 1	Unassigned
0001	Reserved
001	Reserved
010	**Provider-based unicast addresses**

→**Types of IPv6 Addresses:**-Three categories of IPv6 addresses exist they are;

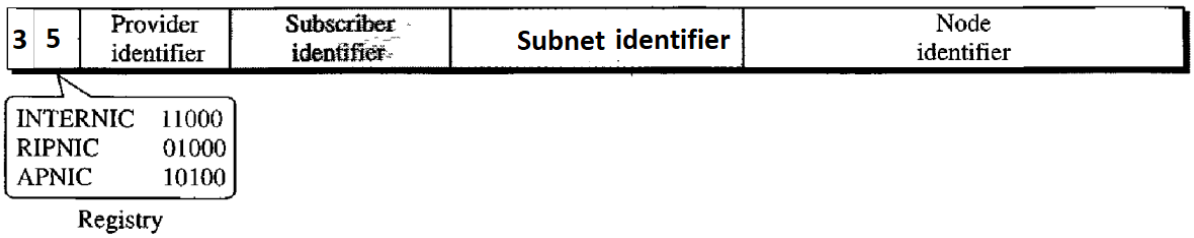
1. **Unicast:**-When a computer that is the source wants to communicate with another computer which is the destination, it uses the unicast address as a destination.



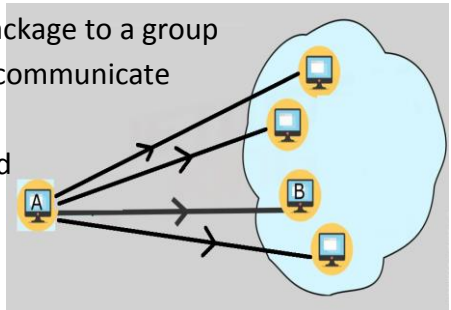
-Therefore, the unicast address refers to a single/individual host.

-The unicast is used to send data to a single destination.

-It communicates one to one.

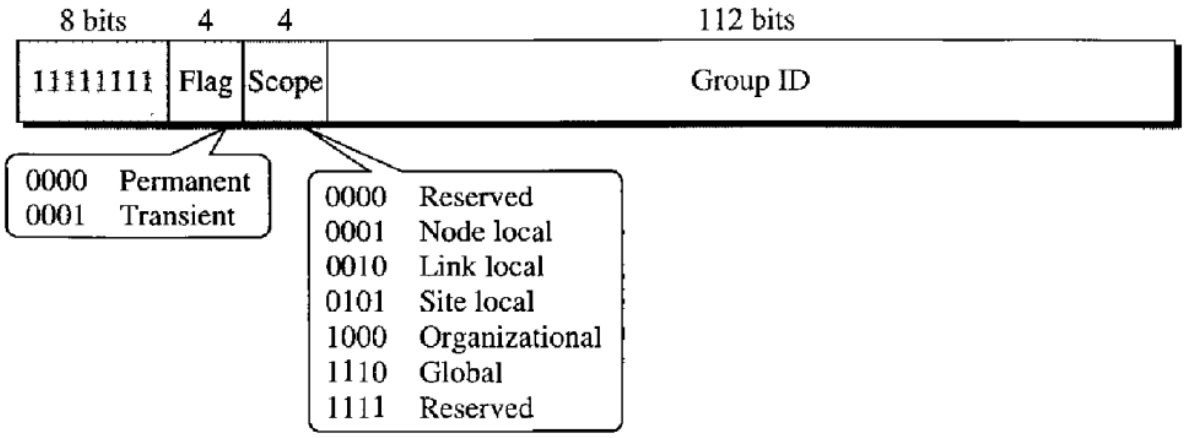


2. **Multicast:**-A multicast address is used to deliver a package to a group of destinations. Therefore, a single source is able to communicate with many other destination hosts.

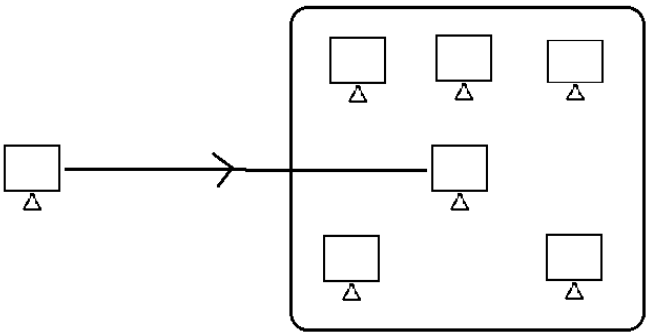


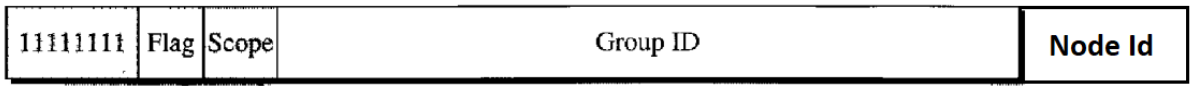
-The packet sent by the multicast address is delivered to every host that is a part of that specific group.

-The multicast has a one-to-many relationship.



3. **Anycast:**-It is used to deliver a packet to a specific host in a group of hosts.





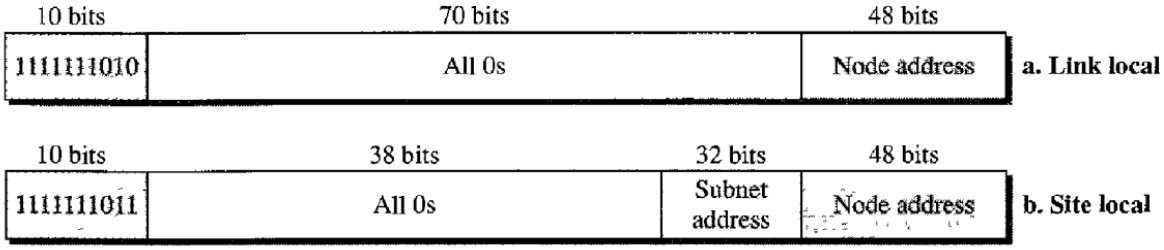
→**Reserved Addresses**:-Another category in the address space is the reserved address. These addresses start with eight 0s (type prefix is 00000000).

- There are few subcategories that are;

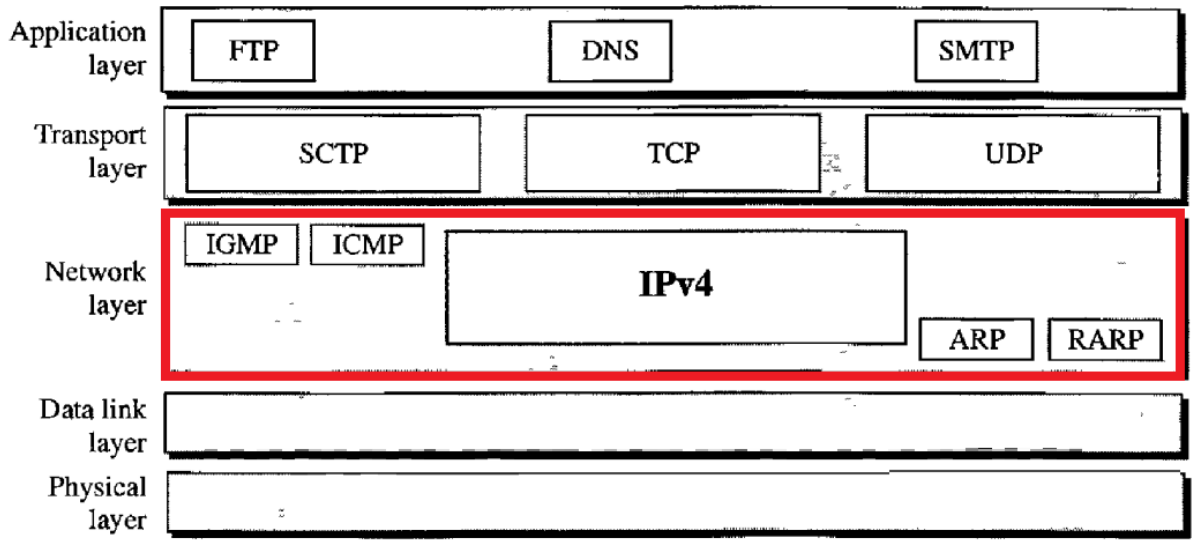
- Unspecified address:-is used when a host does not know its own address **and sends an inquiry to find its address.**
- Loop back address:-is used by a host to test itself without going into the network.
- Compatible address:-is used during the transition from IPv4 to IPv6 .
  -or It is used when a computer using IPv6 wants to send a message to another computer using IPv6, but the message needs to pass through a part of the network that still operates in IPv4.
- Mapped address:-It is used to connection between IPV6 to IPV4.

→**Local Addresses in IPV6**:-These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet.
-In other words, they provide addressing for private networks.
-Nobody outside the organization can send a message to the nodes using these addresses.
-Two types of local addresses are defined for this purpose they are;

1. Link local:-It is used to identify a single subnet.
2. Site local:-It is used to identify groups of subnet.



***IPv4 protocol**:-The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.



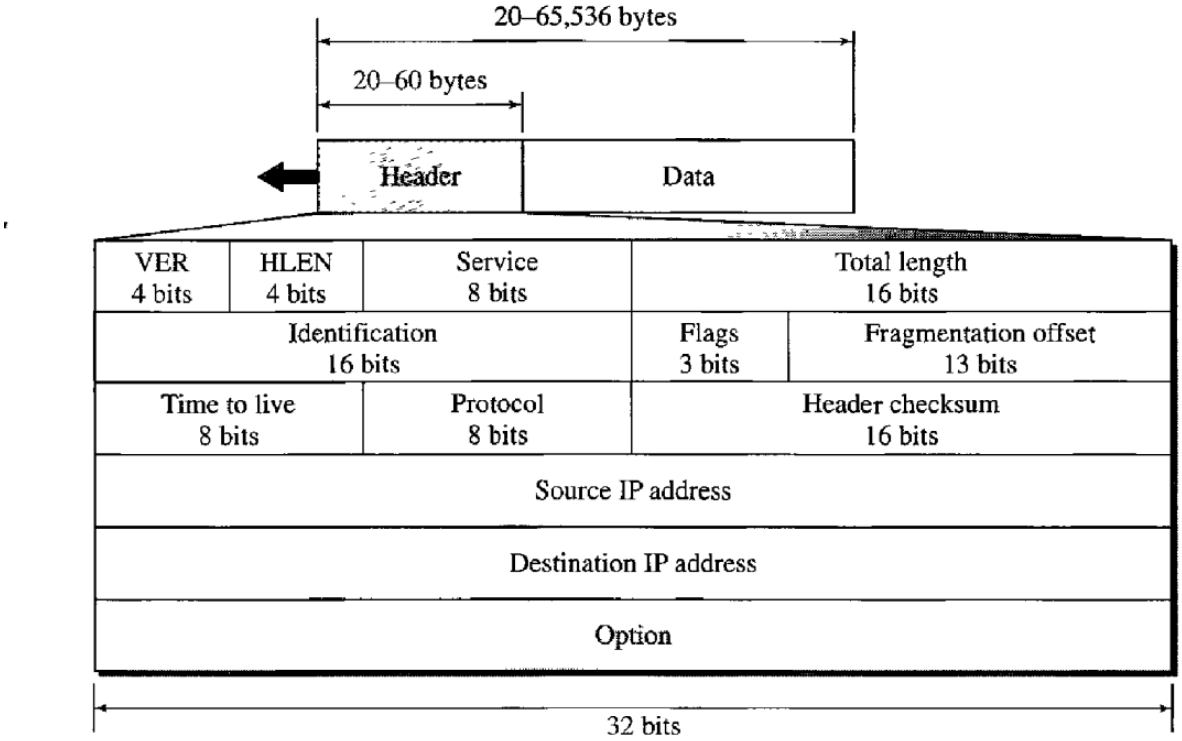
-IPv4 is an **unreliable** and **connectionless** protocol.
-It use package switching,contain datagram approach.
-To support IPV4 protocol ,we use 4 protocols they are;

- **IGMP**
- **ICMP**
- **ARP**
- **RARP**

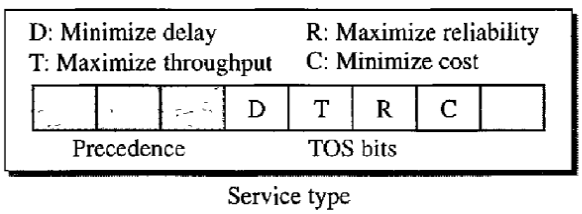
-If reliability is important, IPv4 must be paired with a reliable protocol such as TCP.

-IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach.
-This means that each datagram is handled independently, and each datagram can follow a different route to the destination.
This implies that datagrams sent by the same source to the same destination could arrive out of order.
-Also, some could be lost or corrupted during transmission.

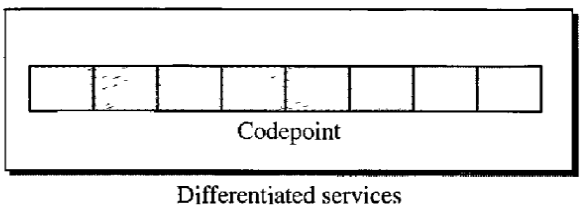
→**Datagram:**-Packets in the IPv4 layer are called datagrams.
the IPv4 datagram format shown below.



-A datagram consisting of two parts: **header and data**.
-The header is 20 to 60 bytes in length and **contains information essential to routing and delivery**.
>Version (VER): This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4.
>Header length (HLEN): This 4-bit field defines the total length of the datagram header in 4-byte words.
-This field is needed because the length of the header is variable (between 20 and 60 bytes).
>Services: IETF has changed the interpretation and name of this 8-bit field.
-This field, previously called **service type**, is now called **differentiated services**.



Service type



Differentiated services

- Precedence bits:-is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary).
- Type of service (TOS):-is a 4-bit subfield with each bit having a special meaning.
  -Although a bit can be either 0 or 1, 0nly one bits can have the value of 1 in each datagram.
  -With only 1 bit, we can have five different types of services.
  -fig—---------->

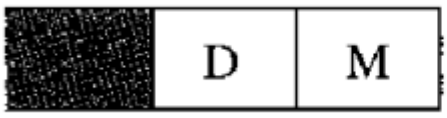
TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

- Differentiated Services:-In this interpretation, the first 6 bits make up the codepoint subfield, and the last 2 bits are not used.

>Total length: This is a In-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes.
-To find the length of the data coming from the upper layer, subtract the header length from the total length.
-The header length can be found by multiplying the value in the HLEN field by 4.

Length of data =total length - header length

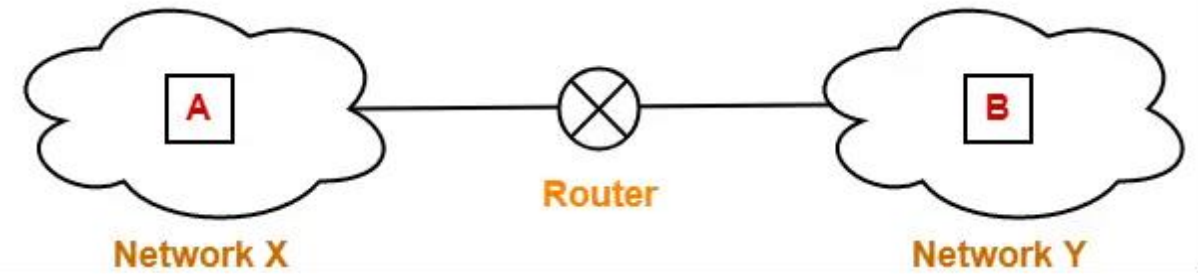
>Flags:This is a 3-bit field. The first bit is reserved. The second bit is called the **do Not fragment bit(D/DF)**.
- If Its value is 1, the machine must not fragment the datagram.
-If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host.
-If its value is 0, the datagram can be fragmented if necessary.
-The third bit is called the **more fragment bit(MF/M)**.
-If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one.
-If its value is 0, it means this is the last or only fragment.



D: Do not fragment
M: More fragments

→**Fragmentation**:-Fragmentation is a process of dividing the datagram into fragments during its transmission.

-It is done by intermediary devices such as routers at the destination host at the network layer.



-Each network has its **maximum transmission unit (MTU)**.
-It dictates the maximum size of the packet that can be transmitted through it.
-Data packets of size greater than MTU can not be transmitted through the network.
-So, datagrams are divided into fragments of size less than or equal to MTU.

-When router receives a datagram to transmit further, it examines the following-

- Size of the datagram
- MTU of the destination network
- DF/D bit value in the IP header

-Then, following cases are possible;

1. **Case-01**:-Size of the datagram is found to be smaller than or equal to MTU.
   -In this case, router transmits the datagram without any fragmentation.
2. **Case-02**:Size of the datagram is found to be greater than MTU and DF bit set to 1.
   -In this case, router discards the datagram.
3. **Case-03**:Size of the datagram is found to be greater than MTU and DF bit set to 0.
   -In this case, router divides the datagram into fragments of size less than or equal to MTU.

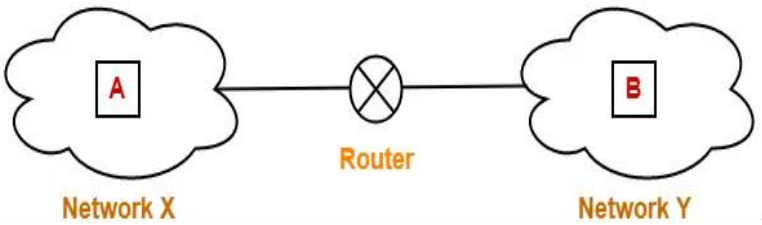
-Router makes the following changes in IPV4 header of each fragment-

- It changes the value of total length field to the size of fragment.
- It sets the MF bit to 1 for all the fragments except the last one.
- For the last fragment, it sets the MF bit to 0.
- It sets the fragment offset field value.
- It recalculates the header checksum.

Examples:Now, lets us discuss some examples of IPV4 fragmentation to understand how the fragmentation is actually carried out.
-There is a host A present in network X having MTU = 520 bytes.
-There is a host B present in network Y having MTU = 200 bytes.
-Host A wants to send a message to host B.

-Consider router receives a datagram from host A having-

Header length = 20 bytes
Payload length = 500 bytes
Total length = 520 bytes
DF bit set to 0



-Now, router works in the following steps-

1. Router examines the datagram and finds-
   - Size of the datagram = 520 bytes
   - Destination is network Y having MTU = 200 bytes
   - DF bit is set to 0

   -Router concludes-
   - Size of the datagram is greater than MTU.
   - So, it will have to divide the datagram into fragments.
   - DF bit is set to 0.
   - So, it is allowed to create fragments of the datagram.
2. Router decides the amount of data that it should transmit in each fragment.

   -Router knows
   - MTU of the destination network = 200 bytes.
   - So, maximum total length of any fragment can be only 200 bytes.
   - Out of 200 bytes, 20 bytes will be taken by the header.
   - So, maximum amount of data that can be sent in any fragment = 180 bytes.

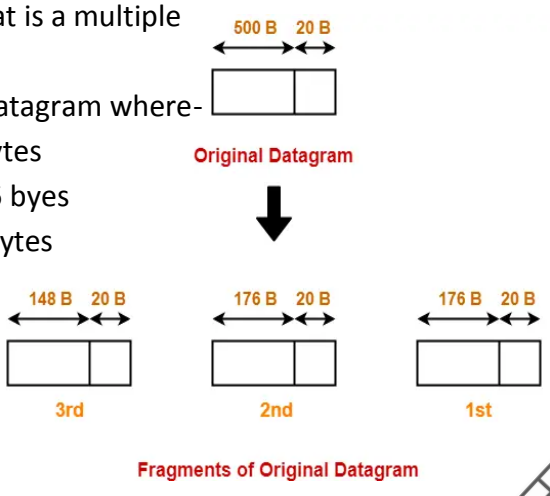
   -Router uses the following rule to choose the amount of data that will be transmitted in one fragment-

Rules: The amount of data sent in one fragment is chosen such that-
- It is as large as possible but less than or equal to MTU.
- It is a multiple of 8 so that pure decimal value can be obtained for the fragment offset field.

-Following the above rule,
- Router decides to send maximum 176 bytes of data in one fragment.
- This is because it is the greatest value that is a multiple of 8 and less than MTU.

3. Router creates three fragments of the original datagram where-
   - First fragment contains the data = 176 bytes
   - Second fragment contains the data = 176 byes
   - Third fragment contains the data = 148 bytes

   -Router transmits all the fragments.



4. At destination side,
    - Receiver receives 3 fragments of the datagram.
    - Reassembly algorithm is applied to combine all the fragments to obtain the original datagram.

***IPV6 protocols**:-Internet Protocol version 6 (IPV 6) is the replacement for version 4 (IPV 4).

-The main difference between IPv4 and IPv6 is the address size of IP addresses.

-The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit address.

- IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

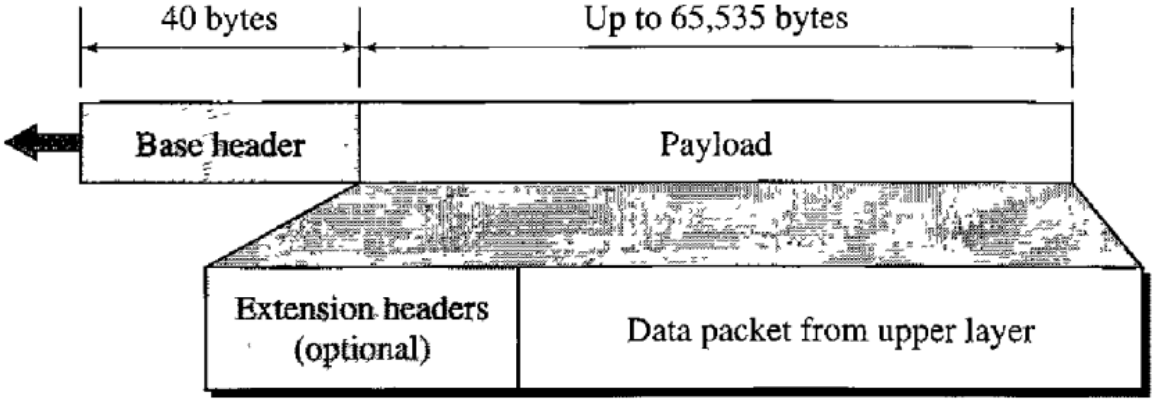
-It provides more security than the IPV4 protocol.

-IPv6 (Internetworking Protocol, version 6), also known as **IPng.**

→**Packet Format**:-Each packet is contain a **base header** and a **payload**.

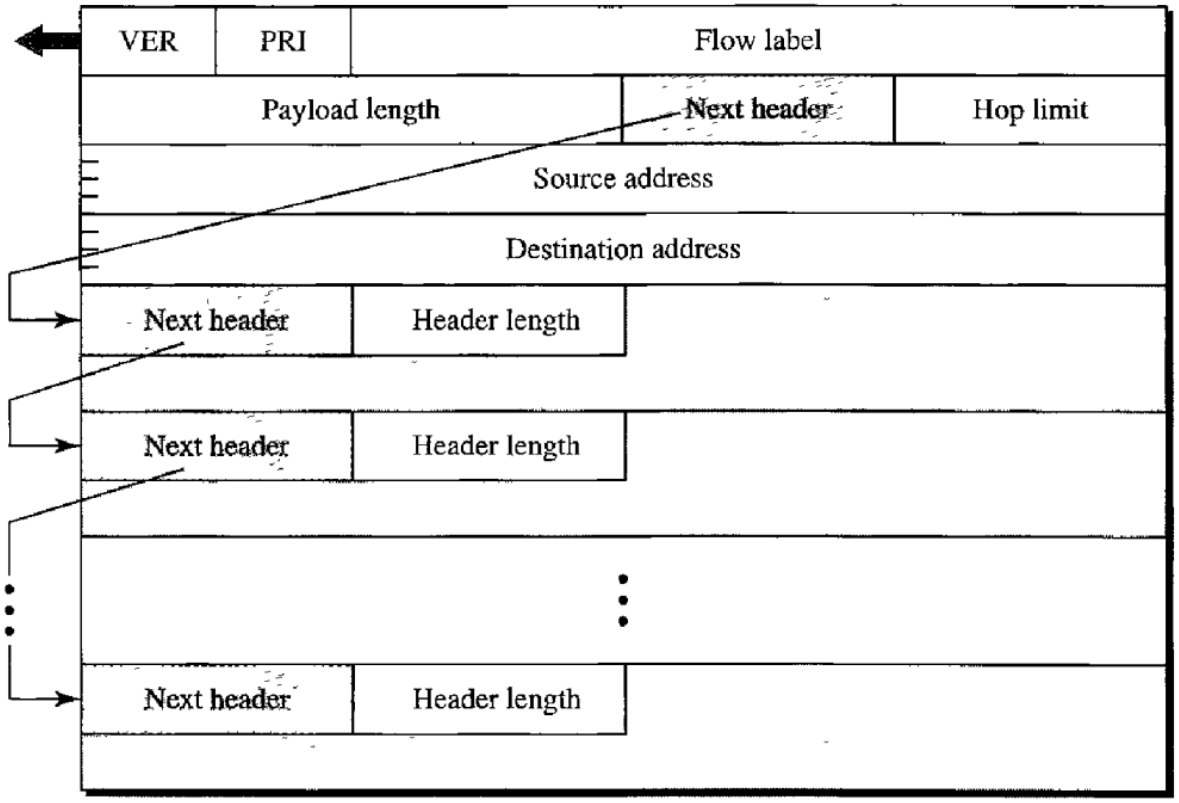
-The payload consists of two parts: **optional extension headers** and **data from an upper layer**.

-The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.



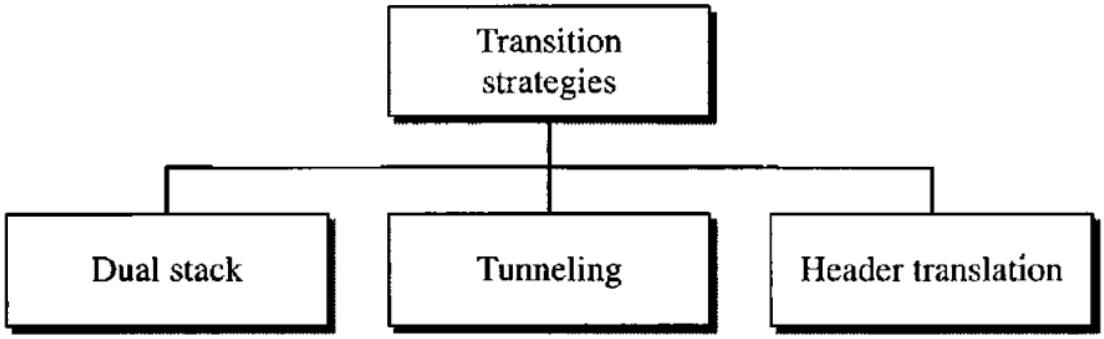
-Base header contains eight fields.,These fields are as follows:

1. Version:This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
2. Priority:The 4-bit priority field defines the priority of the packet with respect to traffic congestion.
3. Flow label:The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.
4. Payload length. The 2-byte payload length field defines the length of the IP datagram excluding the base header.
5. Next header: The next header is an 8-bit field defining the header that follows the base header in the datagram.
   -The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP.

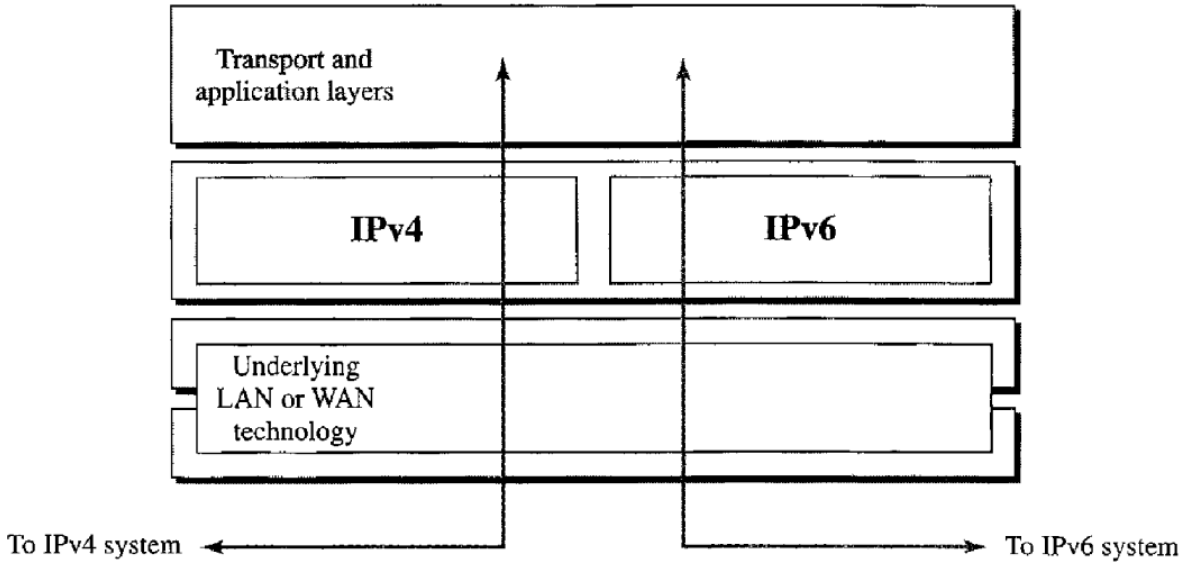


6. Hop limit:It is similar to IPV4 TTL.
7. Source address:Sender logical address
8. Destination address: receiver logical address.

*__Transition from IPv4 to IPv6__:-Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly.
-It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6.
-The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.
-There are three strategies used for transition they are;

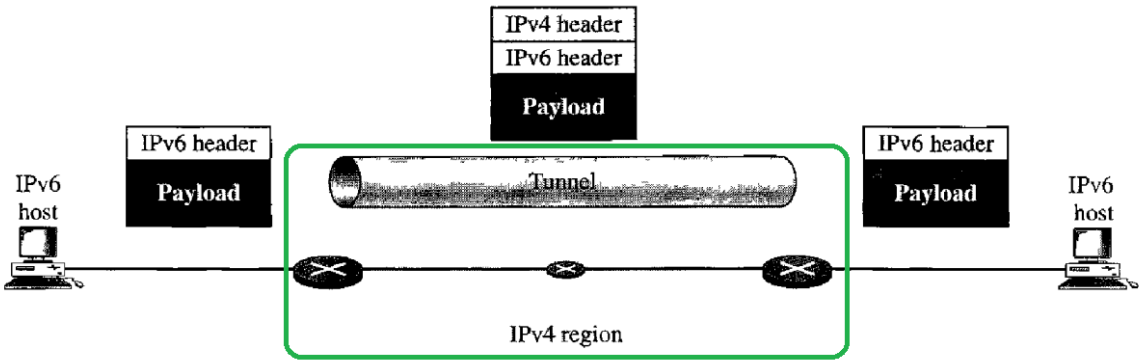


1. **Dual Stack**:-It is recommended that all hosts, before migrating completely to version 6, have a dual stack of protocols.
   -In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6.



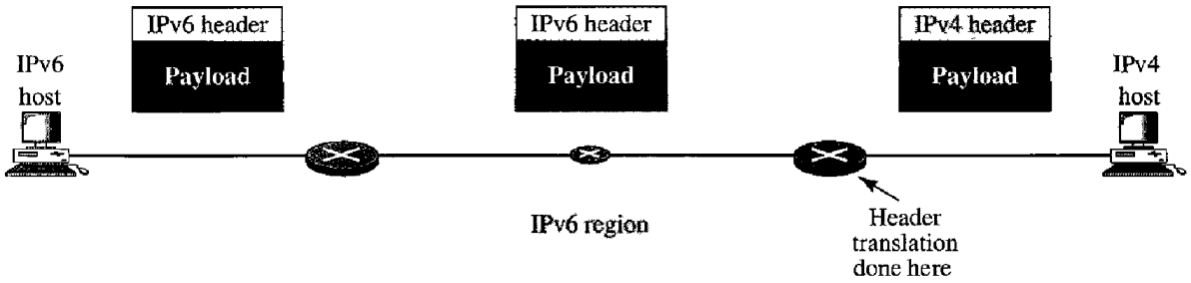
-To determine which version to use when sending a packet to a destination, the source host queries the DNS.
-If the DNS returns an IPv4 address, the source host sends an IPv4 packet.
-If the DNS returns an IPv6 address, the source host sends an IPv6 packet.

2. **Tunneling**:-Thnneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4.



-It is used when sender and receiver is IPV6 ,but in between is IPV4 region.

3. **Header translation**:-Header translation is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4.
-The sender wants to use IPv6, but the receiver does not understand IPv6.
-Tunneling does not work in this situation because the packet must be in the IPv4 format to be understood by the receiver.
-In this case, the header format must be totally changed through header translation.
-The header of the IPv6 packet is converted to an IPv4 header.
-or Here sender is using IPV6 and receiver is using IPV4.



#Address mapping protocols:-Address mapping is a process of determining a logical address knowing the physical address of the device and determining the physical address by knowing the logical address of the device.

-Address mapping is required when a packet is routed from source host to destination host in the same or different network.

-We know that the Internet is a collection of several physical networks that are interconnected using routers.

- Now when in an Internet, a source node sends a packet to the destination node the packet has to travel through different physical networks before it is delivered to the destination node.

-The physical address and the logical address both are different identifiers and we require both of them as the physical address defines the physical connection between source host to destination host whereas the logical address defines routable connection from source host to the destination host and from network to network.

-So as both physical and logical addresses are essential to route a packet from the source host to the destination host, we require an address mapping mechanism to relate a physical address of the device to its logical address and vice versa.

***Types of Addressing mapping**:-There are two type of address mapping they are;

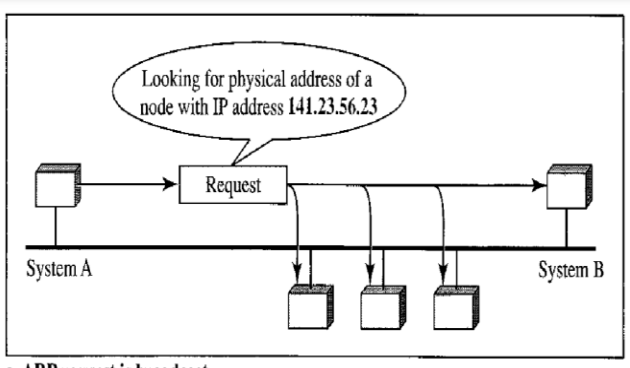
1. **Static mapping**: In static mapping, it creates a table that contains a logical address with a physical address.
   -This table is stored in each machine on the network.
   -Each machine that knows, for example, the IP address of another machine but not its physical address can look it up in the table.
2. **Dynamic mapping**: When a machine knows one of two addresses (logical or physical) through dynamic mapping, it may use this protocol to find the other one address. -There are two protocols designed for dynamic mapping they are;
   - Address Resolution Protocol (ARP)
   - Reverse Address Resolution Protocol (RARP) (ethu explain cheyanda)

***Address Resolution Protocol (ARP)**:-It is used to find physical address of receiver.

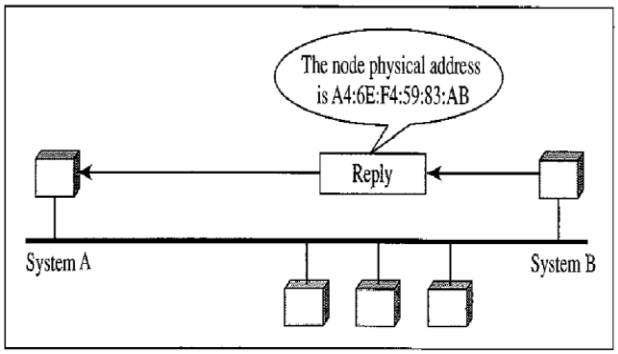
-The sender send ARP request to all machines.

-But only ARP request will accepted by the machine who have logical address is match to the sender logical address.

-then the receiver will send back a ARP response to the sender.



a. ARP request is broadcast



b. ARP reply is unicast

-Working:

- ➢ Firstly, the client broadcasts the ARP request packet to all the hosts in the network.
- ➢ In this ARP request packet, stores the logical address and physical address of the client and the IP address of the receiver.
- ➢ Each host receives this ARP request packet, but only the one who is the authorized host completes the ARP service.
- ➢ Finally, the authorized host sends the ARP response packet to the client in which its physical address is stored.

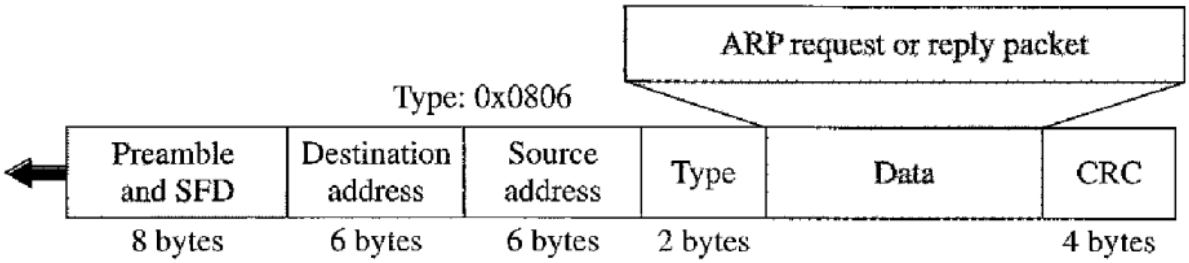
Note: ARP request is broadcast, and ARP response is unicast.

->ARP Packet Format:

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

→**<u>Encapsulation of ARP packet</u>**:-An ARP packet is encapsulated directly into a data link frame.

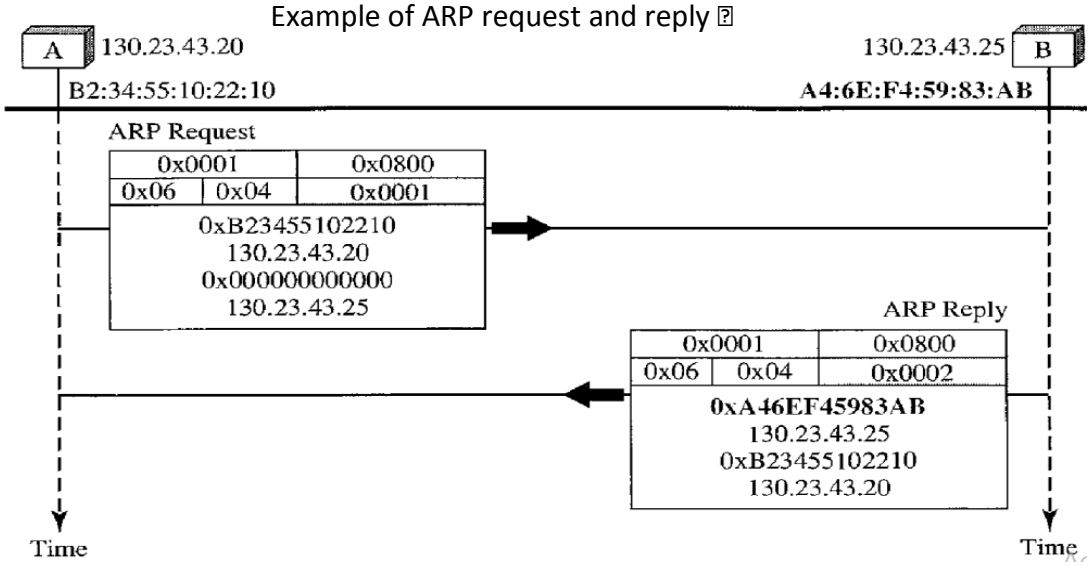
-ARP packet is encapsulated in an Ethernet frame



→**ARP Process/(steps)**:-These are the steps involved in an ARP process:

1. The sender knows the IP address of the target.
2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address.
   -The target physical address field is filled with Os.
3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.
4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP.
   -All machines except the one targeted drop the packet.
   -The target machine recognizes its IP address.
5. The target machine replies with an ARP reply message that contains its physical address.
   -The message is unicast.
6. The sender receives the reply message. It now knows the physical address of the target machine.
7. The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

Example of ARP request and reply



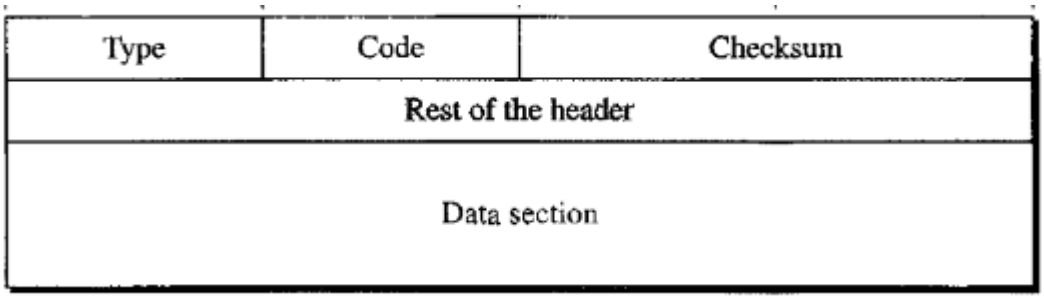
*__DHCP__:-DHCP stands for Dynamic Host Configuration Protocol.
-It is designed to provide the static and dynamic address allocation that can be done manual or automatic.
-It was designed to replace the **BOOTP (Bootstrap Protocol)**.
-bakki google

#Error Reporting protocol

***ICMP/** :-The ICMP stands for Internet Control Message Protocol.
-t is used tor error handling in the network layer, and it is primarily used on network devices such as routers.
-As different types of errors can exist in the network layer, so ICMP can be used to report these errors and to debug those errors.
-For example, some sender wants to send the message to some destination, but the router couldn't send the message to the destination.
-In this case, the router sends the message to the sender that I could not send the message to that destination.
-It has two deficiencies/limitation they are;

- **Lack of Error Control**
- **Lack of assistance mechanisms**

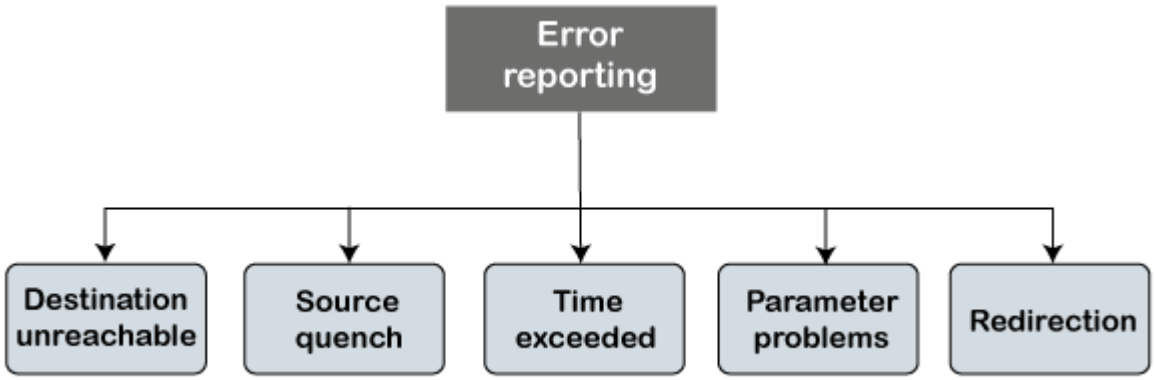
-The IP protocol has no error-reporting or error-correcting mechanism. What happens if something goes wrong?
-if someone sends the message to the destination, the message is somehow stolen between the sender and the destination.
-If no one reports the error, then the sender might think that the message has reached the destination.
-If someone in-between reports the error, then the sender will resend the message very quickly.
-These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.
-Format of ICMP message



>Type: It defines the ICMP message type.
>Code: It defines the subtype of the ICMP message.
>Checksum: It detect whether the error exists in the message or not.

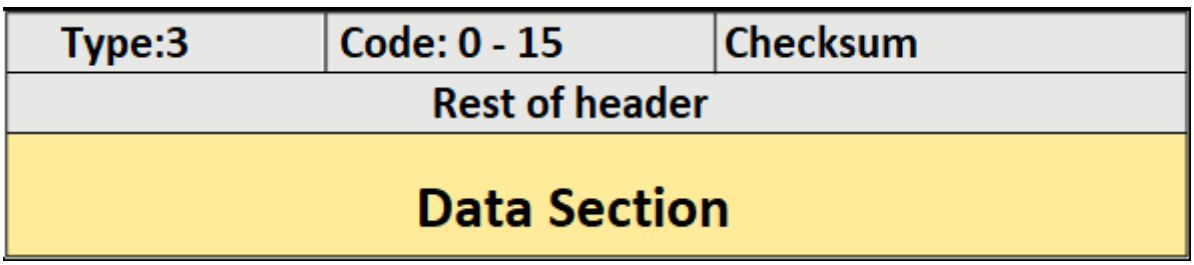
→**Types of Messages**:-ICMP messages are divided into two broad categories:

1. **Error-reporting messages**:-The error-reporting message means that the router face a problem when it processes an IP packet then it reports a message.
   -Or When a packet of data is sended and router can't process it ,so the router sends back a message called Error reporting message.
   -The error reporting messages are broadly classified into the five categories they are:



- Destination unreachable:-The destination unreachable error occurs when the packet does not reach the destination.
  -Suppose the sender sends the message, but the message does not reach the destination, then the intermediate router reports to the sender that the destination is unreachable.
  -**The number 3 specifies that the destination is unreachable**.
  -And its code is from 0 to 15.
  -Each code have appropriate meaning;

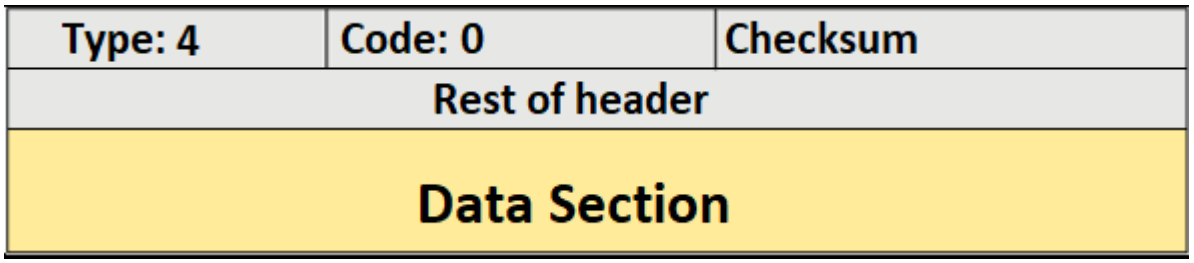
>code 0:hardware failure.
>code 1:Host is unreachable.
>code 3:port is unreachable.
>code 7:Destination host is unreached.



- source quench:-The sender resends the packet at a higher rate, and the router is not able to handle the high data rate.

-To overcome such a situation, the router sends a source quench message to tell the sender to send the packet at a lower rate.

-**It is a type 4 message, and code is zero.**



- Time exceeded:-Sometimes the situation arises when there are many routers that exist between the sender and the receiver.

  -When the sender sends the packet, then it moves in a routing loop.

  -The time exceeded is based on the time-to-live value.

  -When the packet traverses through the router, then each router decreases the value of TTL by one.

  -Whenever a router decreases a datagram with a time-to-live value to zero, then the router discards a datagram and sends the time exceeded message to the original source.

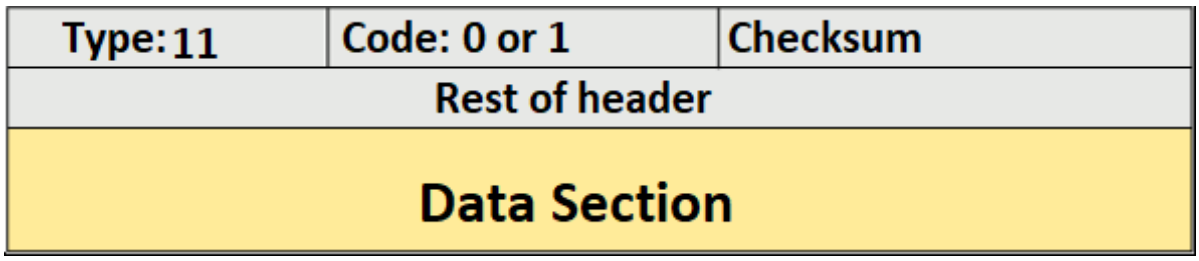
  -Example, some layers can handle upto 1500 data units, and some can handle upto 300 units.

  -When the packet is sent from a layer having 1500 units to the layer having 300 units, then the packet is divided into fragments; this process is known as fragmentation.

  -These 1500 units are divided into 5 fragments, i.e., f1, f2, f3, f4, f5, and these fragments reach the destination in a sequence.

  -If all the fragments are not reached to the destination in a set time, they discard all the received fragments and send a time-exceeded message to the original source.

  -**The type of time-exceeded is 11 and the code can be either 0 or 1.**



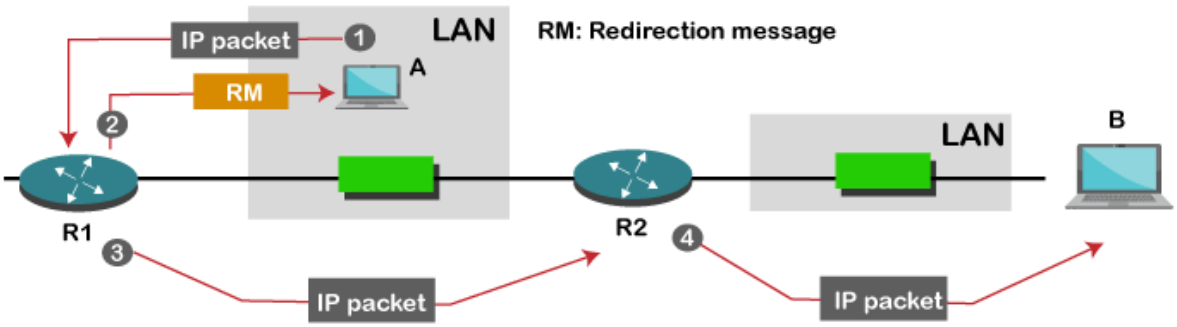
- Parameter problems:-The router and the destination host can send a parameter problem message.

  -This message conveys that some parameters are not properly set.

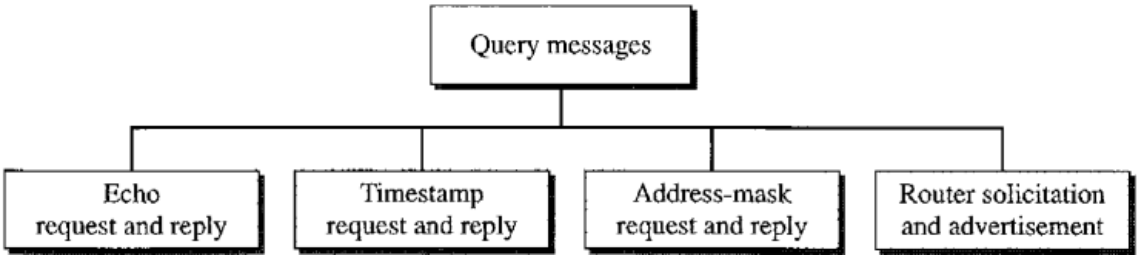
  -**The type of message is 12 and the code can be either 0 or 1.**

Type: 12	Code: 0 or 1	Checksum
Pointer	All zeor's	
Data		

- Redirection:- A wants to send the packet to B, and there are two routers exist between A and B.
  -Router R2 is obviously the most efficient routing choice(shortest path), but host A did not choose router R2.
  -Instead it choose R1.
  - First, A sends the data to the router 1.
  -The router 1 sends the IP packet to router 2 and redirection message to A so that A can update its routing table.



2. **Query messages**:-The ICMP Query message is used for error handling or debugging the internet.



→Echo-request and echo-reply message:-A router or a host can send an echo-request message. It is used to ping a message to another host that "Are you alive".
-If the other host is alive, then it sends the echo-reply message.
-An echo-reply message is sent by the router or the host that receives an echo-request message.
-The message format of echo-request and echo-reply message are;

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Optional data		

-The above diagram shows the message format of the echo-request and echo-reply message.
-The type of echo-request is 13, and the request of echo-reply is 14.
-The code of this message is 0.

"Ethu matram explain cheythal mathi"

***ICMPv6:**-Another protocol that has been modified in version 6 of the TCP/IP protocol suite is ICMP (ICMPv6).
-This new version follows the same strategy and purposes of version 4.
-ICMPv4 has been modified to make it more suitable for IPv6.
-In addition, some protocols that were independent in version 4 are now part of Internetworking Control Message Protocol (ICMPv6).

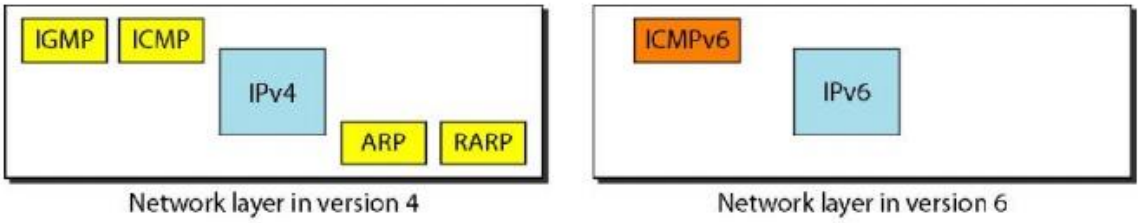


Figure 3.38 Comparison of network layers in version 4 and version 6

-The ARP and IGMP protocols in version 4 are combined in ICMPv6.
-The RARP protocol is dropped from the suite because it was rarely used.
-4 types they are;

1. **Error reporting message:**-The main responsibilities of ICMP is to report errors.
   -Five types of errors are handled:
   - Destination unreachable:- mukalilathathu thanne explain cheythal mathi
   - Packet too big :-The packet-too-big message is added because fragmentation is the responsibility of the sender in IPv6.
     -If the sender does not make the right packet size decision, the router has no choice but to drop the packet and send an error message to the sender.
   - Time exceeded:-mukalilathathu thanne explain cheythal mathi
   - Parameter problems:-mukalilathathu thanne explain cheythal mathi
   - Redirection:-mukalilathathu thanne explain cheythal mathi

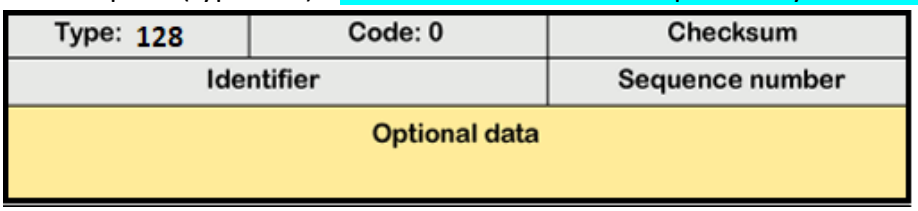
   --The source-quench message is eliminated in version 6 because the priority and the flow label fields allow the router to control congestion and discard the least important messages.

-In this version, there is no need to inform the sender to slow down.

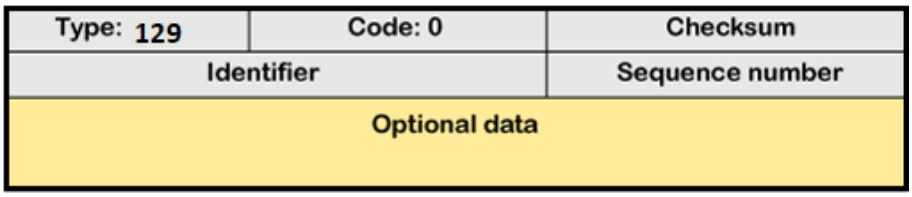
Table 21.3 *Comparison of error-reporting messages in ICMPv4 and ICMPv6*

Type of Message	*Version 4*	*Version 6*
Destination unreachable	Yes	Yes
Source quench	Yes	No
Packet too big	No	Yes
Time exceeded	Yes	Yes
Parameter problem	Yes	Yes
Redirection	Yes	Yes

2. **Information message:**-It is equivalent to query message in ICMPv4.
   -It have two types of messages they are;
   - Echo request (type:128) :-mukalilathathu thanne explain cheythal mathi



   - Echo replay (Type:129):-mukalilathathu thanne explain cheythal mathi



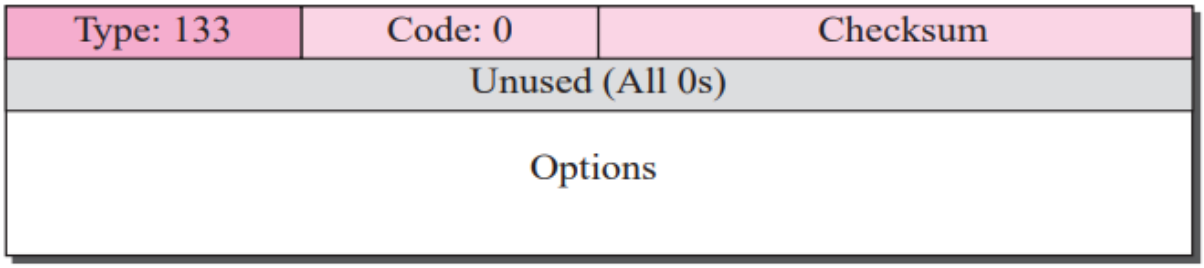
   -the echo request and echo response messages are designed to check if two devices in the Internet can communicate with each other.
   -A host or router can send an echo request message to another host;
   the receiving computer or router can reply using the echo response message.

3. **Neighbor Discovery message:**To support or improve messages there are two additional protocol are used they are;
   - ND (Neighbor-Discovery) protocol :-Its 2 usages are;
     >When it use station ,Then it identifies the router which is close to the station.
     >when it's not use the station ,but it is using the router .Then it find nearest station or router physical address.
   - IND (Inverse-Neighbor-Discovery) protocol:-It is a opposite of ND.
     -It is used to find logical address.

-Neighbor Discovery message can be divided into 7 types they are;

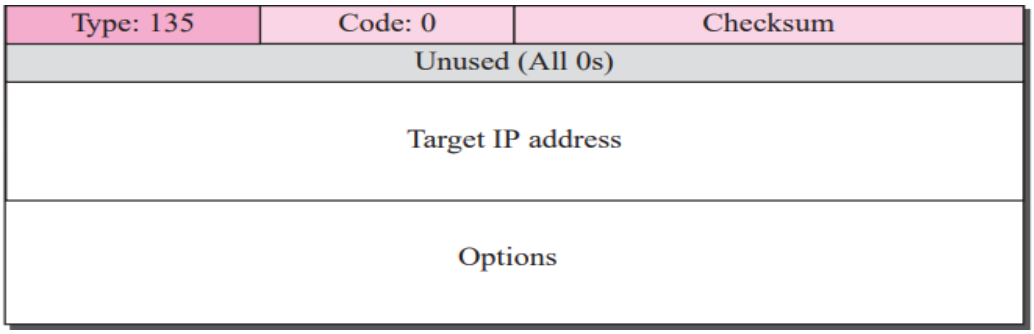
I) Router-Solicitation Message:-It is used to send request to router.

-The format of the message is:



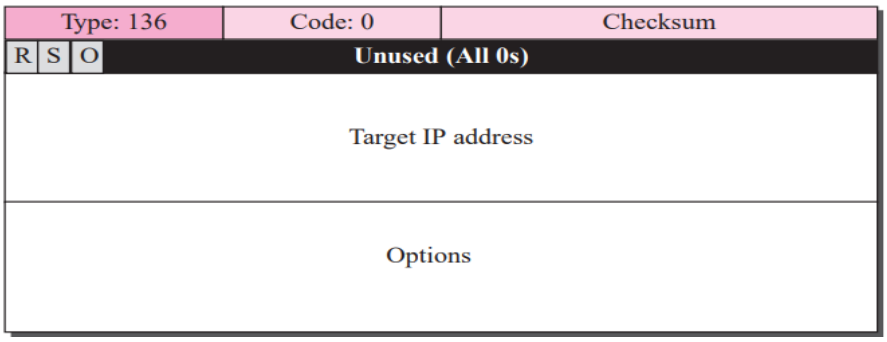
II) Router-Advertisement Message:-After receiving router solicitation message and accept it ans then send message from router to host.

III) Neighbor-Solicitation Message:-It is equivalent to ARP request message.

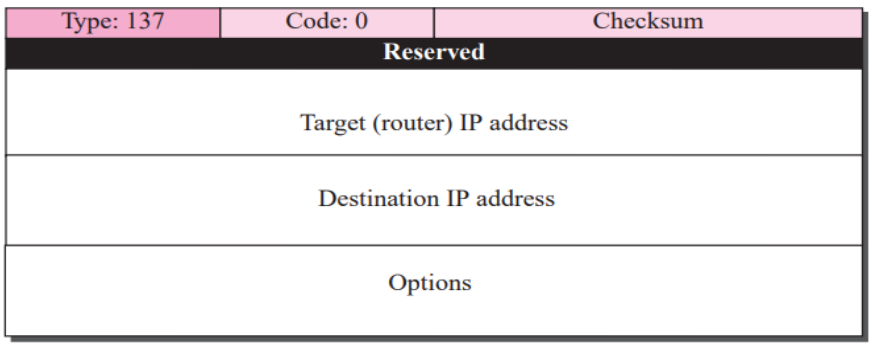


IV) Neighbor-Advertisement Message:-sent in response to the neighbor-solicitation message.

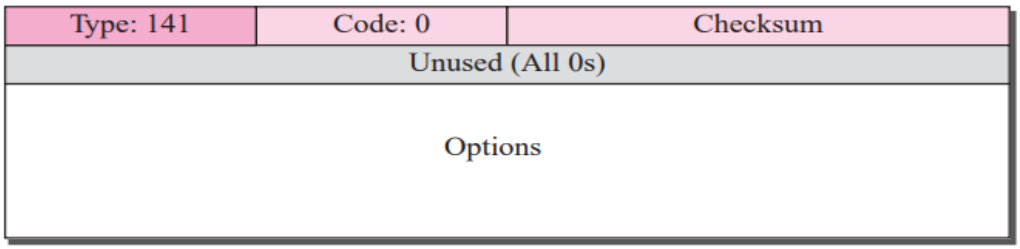
-This is equivalent to the ARP reply message in IPv4.



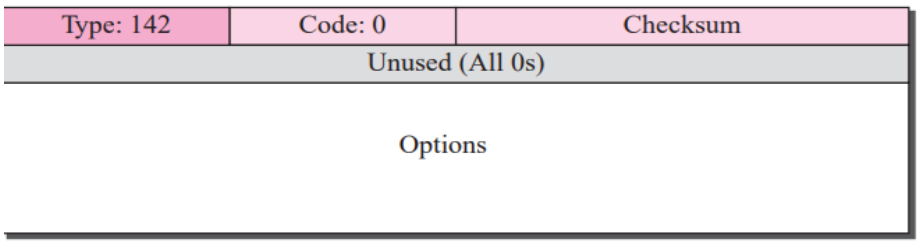
V) Redirection Message:-The purpose of the redirection message is the same as described for version 4.



VI) Inverse-Neighbor-Solicitation Message:-The inverse-neighbor-solicitation message is sent by a node that knows the link layer address of a neighbor, but not the neighbor's IP address.



VII) Inverse-Neighbor-Advertisement Message:-It send back after accepting Inverse-Neighbor-Solicitation Message.
-It used to find the physical address of neighbour node.
-Then the node send the physical address to back.



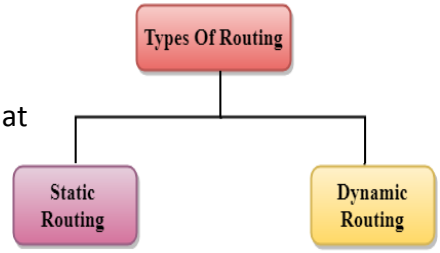
4. **Group membership message:**-It have two types they are;

i) Membership query mess:-A membership-query message is sent by a router to find active group members in the network.
-Mukalilathe same fig annu but type:130

ii) Membership report mess:-The active member send message to router is called membership report message.
-Mukalilathe same fig annu but type:143

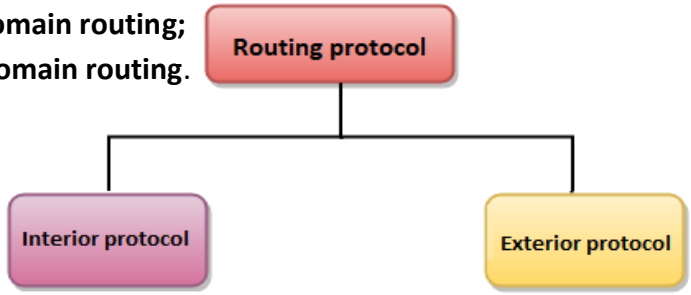
#Routing Protocols:-Routing Protocols are the set of defined rules used by the routers to communicate between source & destination.
-They do not move the information to the source or to a destination, but only update the routing table that contains the information.
-Router protocols helps you to specify way routers communicate with each other.
-A routing table can be either static or dynamic.

- A static table is one with manual entries.
- A dynamic table, on the other hand, is one that is updated automatically when there is a change somewhere in the internet.



-Today, an internet needs dynamic routing tables.
-The tables need to be updated as soon as there is a change in the internet.
-For instance, they need to be updated when a link is down, and they need to be updated

whenever a better route has been found.
-Routing protocols can be either an **interior protocol or an exterior protocol**.
-An interior protocol handles intradomain routing;
-An exterior protocol handles interdomain routing.

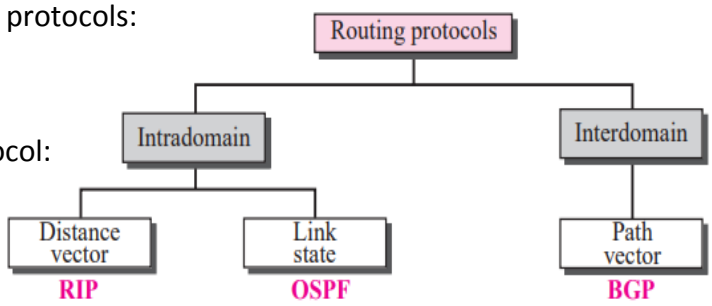


-Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers.
-For this reason, an internet is divided into **autonomous systems**.
-An autonomous system (AS) is a group of networks and routers under the authority of a single administration.
-**Routing inside an autonomous system is referred to as intra-domain routing.**
-Routing between autonomous systems is referred to as inter-domain routing.
-Each autonomous system can **choose one or more intradomain routing protocols** to handle routing inside the autonomous system.
-However, **only one interdomain routing protocol** handles routing between autonomous systems.
-There are Several intra-domain and inter-domain routing protocols are in use they are;
-There are two intra-domain routing protocols:

- distance vector
- link state.

-And one inter-domain routing protocol:

- path vector



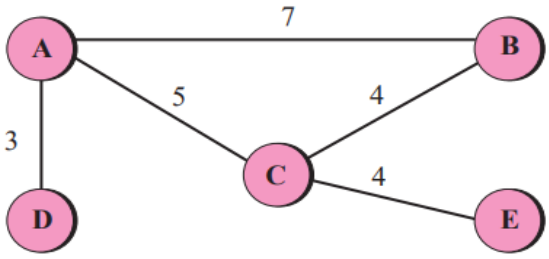
-**Routing Information Protocol (RIP)** is the implementation of the distance vector protocol.
-**Open Shortest Path First (OSPF)** is the implementation of the link state protocol.
-**Border Gateway Protocol (BGP)** is the implementation of the path vector protocol.
-RIP and OSPF are interior routing protocols; BGP is an exterior routing protocol.

***Distance vector routing**:-This method sees an autonomous systems, with all routers and networks, as a graph, a set of nodes and lines (edges) connecting the nodes.
-A router can normally be represented by a node and a network by a link connecting two nodes.
-The graph theory used an algorithm called **Bellman-Ford** (also called Ford-Fulkerson) for a while to find the shortest path between nodes in a graph given the distance between nodes.

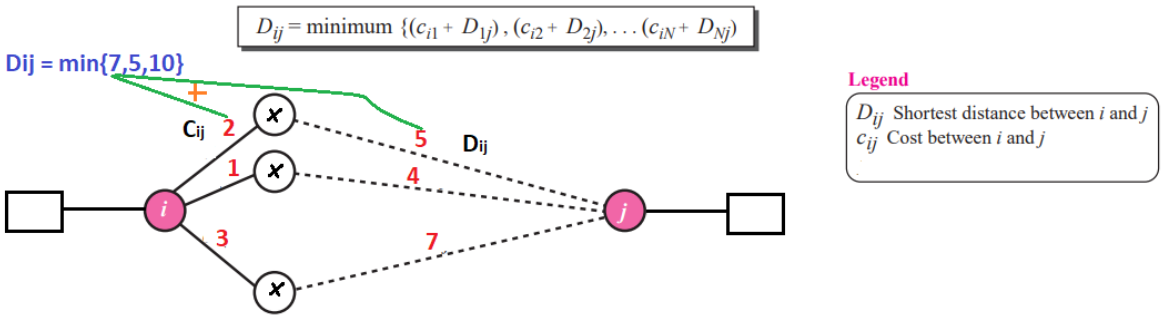
→**Bellman-Ford Algorithm**:-The algorithm is used to find the least cost (shortest path) between any two nodes.

-The cost of each line is given over the line; the algorithm can find the least cost between any two nodes.

-For example, if the nodes represent cities and the lines represent roads connecting them, the graph can find the shortest distance between any two cities.



-The algorithm is based on the fact that if all neighbors of node i know the shortest distance to node j, then the shortest distance between node i and j can be found by adding the distance between node i and each neighbor to the neighbor's shortest distance to node j and then select the minimum.



-**Here minimum cost distance is (1+4) = 5**

→**Distance Vector Routing Algorithm**:-

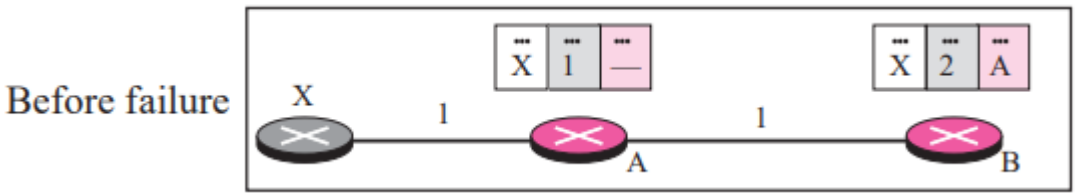
1. In distance vector routing, the cost is normally hop counts (how many networks are passed before reaching the destination).
   -So the cost between any two neighbors is set to 1.
2. Each router needs to update its routing table asynchronously, whenever it has received some information from its neighbors.
3. After a router has updated its routing table, it should send the result to its neighbors so that they can also update their routing table.
4. Each router should keep at least three pieces of information for each route: destination network, the cost, and the next hop..
5. We refer to information about each route received from a neighbor as R (record), which has only two pieces of information: R.dest and R.cost.

-The next hop is not included in the received record because it is the source address of the sender.

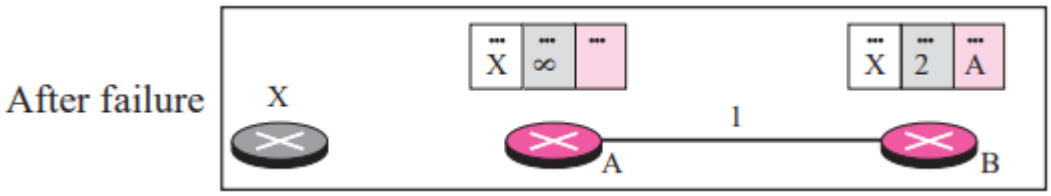
***Count to Infinity:**-A problem with distance vector routing is that any decrease in cost (good news) propagates quickly, but any increase in cost (bad news) propagates slowly.
-For a routing protocol to work properly, if a link is broken (cost becomes infinity), every other router should be aware of it immediately, but in distance vector routing, this takes some time.
-The problem is referred to as **count to infinity**.
-count to infinity is classified into 2 types they are;

Dest	Cost	Next

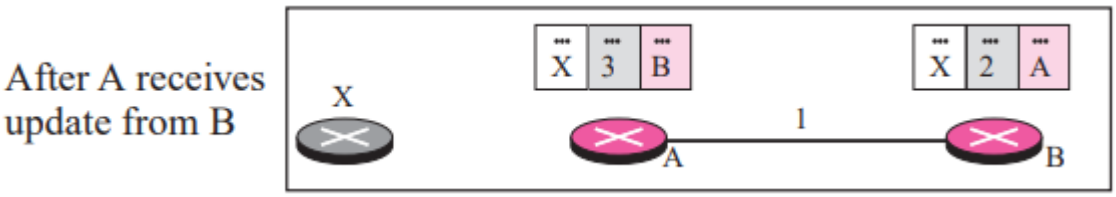
1. **Two-Node Loop**:-Here we have three node .
   -At the beginning, both nodes A and B know how to reach node X.



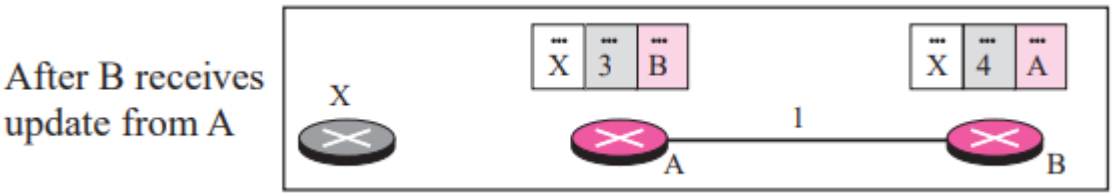
-But suddenly, the link between A and X fails. Node A changes its table.



- If A can send its table to B immediately, everything is fine.
-However, the system becomes unstable if B sends its routing table to A before receiving A's routing table.
-Node A receives the update and, assuming that B has found a way to reach X, immediately updates its routing table of A.



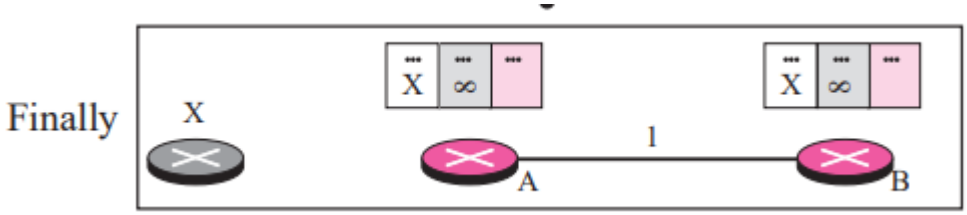
-Now A sends its new update to B.
-Now B thinks that something has been changed around A and updates its routing table.



-The cost of reaching X increases gradually until it reaches infinity.

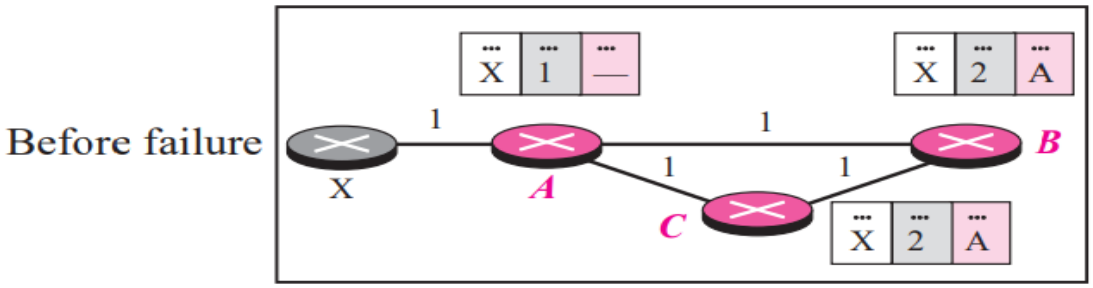
-At this moment, both A and B know that X cannot be reached. However, during this time the system is not stable.
-Node A thinks that the route to X is via B; node B thinks that the route to X is via A.

- If A receives a packet destined for X, it goes to B and then comes back to A. - Similarly, if B receives a packet destined for X, it goes to A and comes back to B. - Packets bounce between A and B, creating a two-node loop problem.

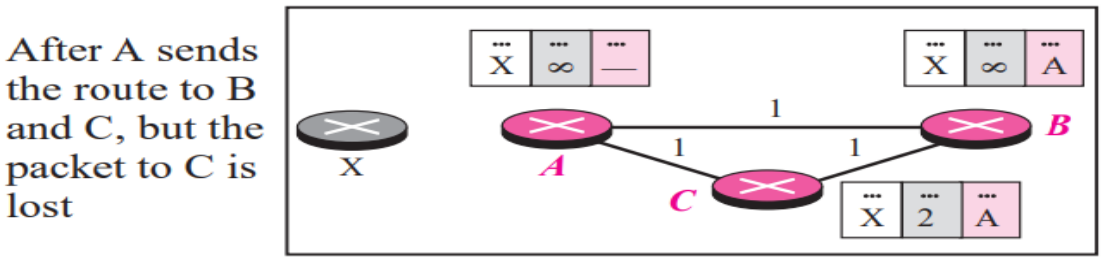


∞ (infinity):-It means that there is a connection failure and update routing table as **∞**

2. **Three-Node Instability**:-Here the instability is between three nodes.

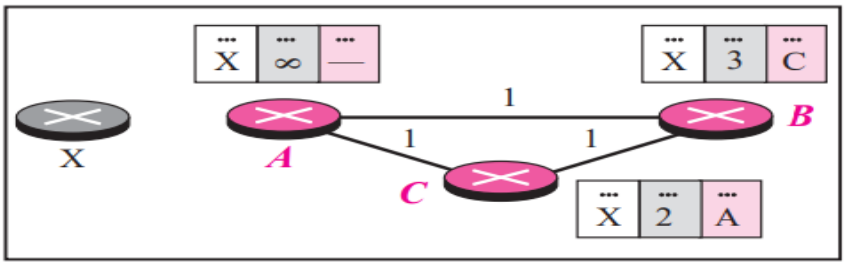


- After finding that X is not reachable, node A sends a packet to B and C to inform them of the situation.
-Node B immediately updates its table, but the packet to C is lost in the network and never reaches C.
-Node C remains in the dark and still thinks that there is a route to X via A with a distance of 2.



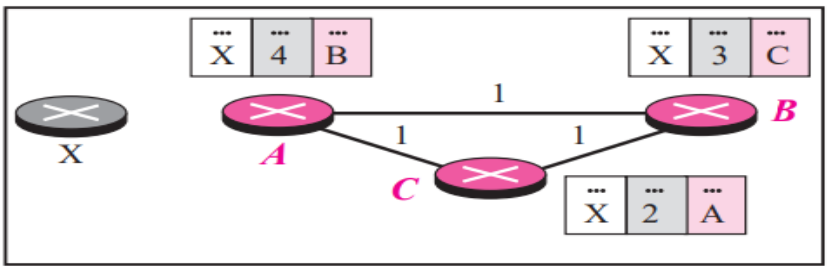
-After a while, node C sends its routing table to B, which includes the route to X. - Node B is totally fooled here.
-It receives information on the route to X from C, and according to the algorithm, it updates its table showing the route to X via C with a cost of 3.
-This information has come from C, not from A,

After C sends the route to B

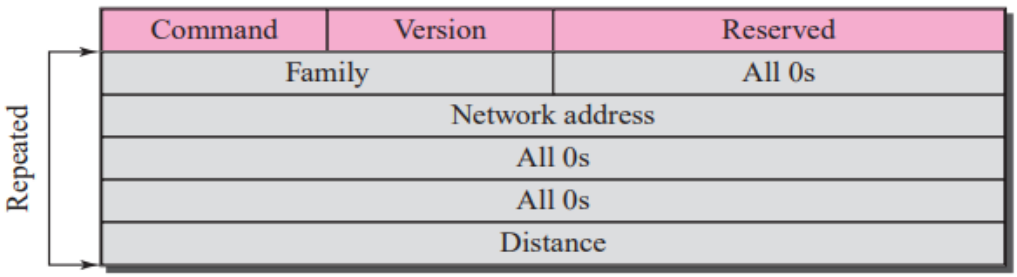


-So after a while node B may advertise this route to A.
-Now A is fooled and updates its table to show that A can reach X via B with a cost of 4. And the loop continues.
-now A advertises the route to X to C, with increased cost, but not to B.
-C then advertises the route to B with an increased cost.
-B does the same to A.
-And so on.
-The loop stops when the cost in each node reaches infinity.

After B sends the route to A



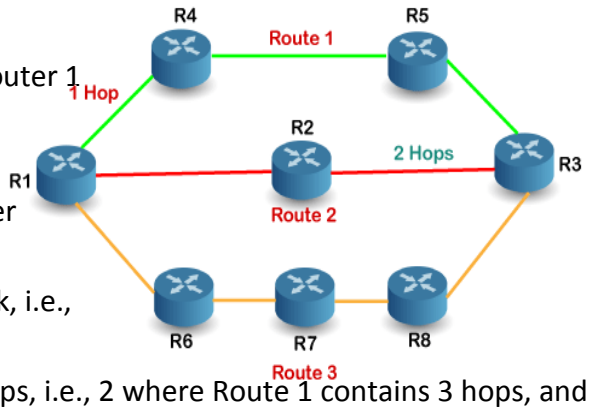
***RIP:**-The Routing Information Protocol (RIP) is an intradomain (interior) routing protocol used inside an autonomous system.
-It is a very simple protocol based on distance vector routing.
-It uses **hop count** find the best path between the source and the destination network.
-Hop count is the number of routers occurring in between the source and destination network. The path with the lowest hop count is considered as the best route.
-RIP prevents routing loops by limiting the number of hops allowed in a path from source and destination.
-The format of the RIP message:



>Command:the type of message: request (1) or response (2).
>Version: show which version is using.

>Family:It defines the family of the protocol used.
>Network address:The address field defines the address of the destination network.
>Distance: It defines the hop count (cost) from the advertising router to the destination network.

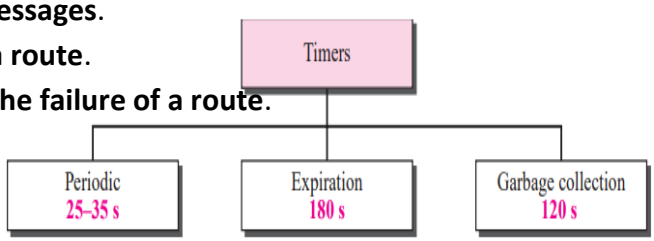
->RIP work:



-If there are 8 routers in a network where Router 1 wants to send the data to Router 3.
-If the network is configured with RIP, it will choose the route which has the least number of hops.
-There are three routes in the above network, i.e., Route 1, Route 2, and Route 3.
The Route 2 contains the least number of hops, i.e., 2 where Route 1 contains 3 hops, and Route 3 contains 4 hops, so RIP will choose Route 2.

Note:If both the routes contain the same number of hops. -So RIP will send the data to both the routes simultaneously. -This way, it manages the load balancing, and data reach the destination a bit faster.

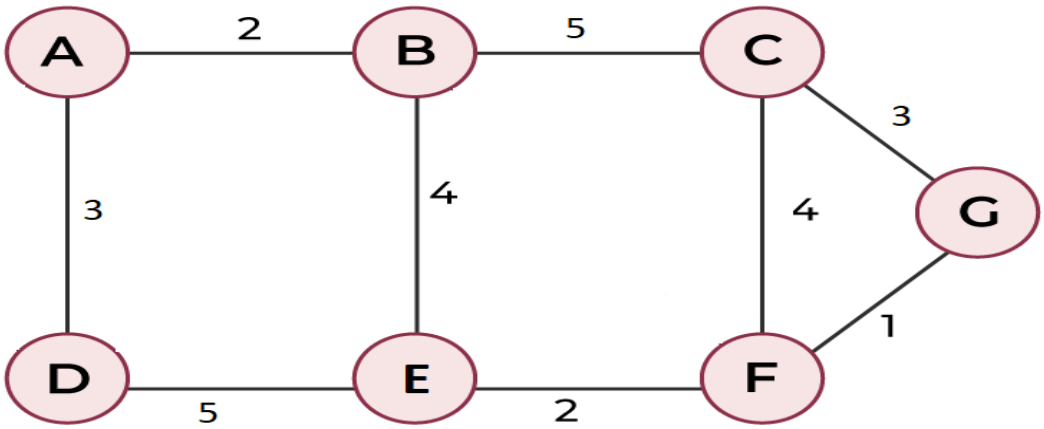
→Timers in RIP:-RIP uses three timers to support its operation.
-The periodic timer **controls the sending of messages**.
-The expiration timer **governs the validity of a route**.
-And the garbage collection timer **advertises the failure of a route**.



-Send routing information to different router in a fixed time. 30 s is default.
-if router information is not received expiration timer enable after 180s and send destination unreachable message.
-After sending destination unreachable message garbage collection enable and after 120s corresponding router is destroyed.

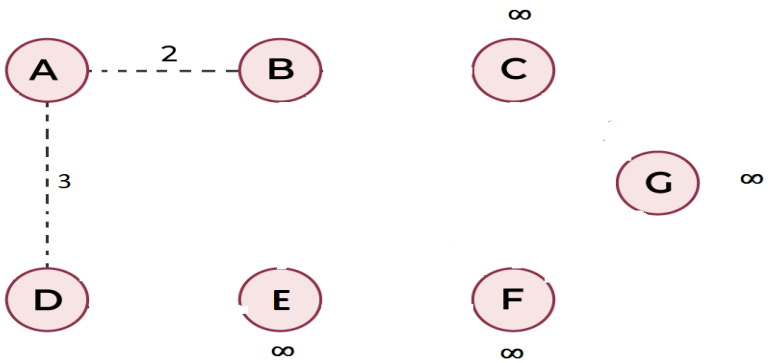
***Link state Routing:**-Link state routing has a different philosophy from that of distance vector routing.
-To find the shortest path, each node needs to run the famous **Dijkstra algorithm**.

Eg:



-first consider A as root.

-And root A have adjacent nodes B and D.

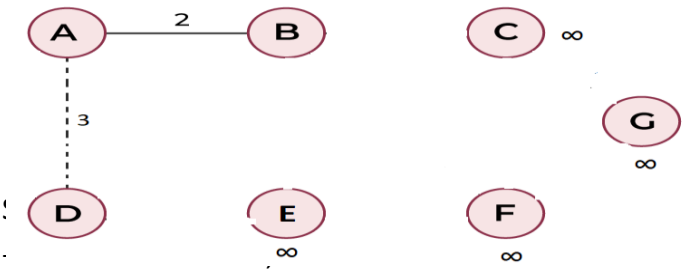


Step 1:Here A to B cost is 2 and A to D cost is 3.

-And here shortest cost is A to B.

-And set a connection between A → B.

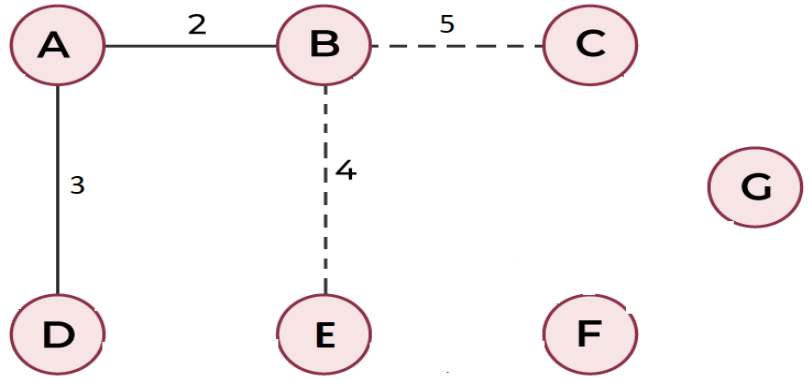
-And mark other node (E,C,F,G) as infinite.



-also consider A to D node here its cost is 3 .

-so we choose shortest path as A → D.

And we mark a solid line between them.



Step 3:Next we have node B and its adjacent node as E and C.

-Also we have node D and its a adjacent node as E.

- here

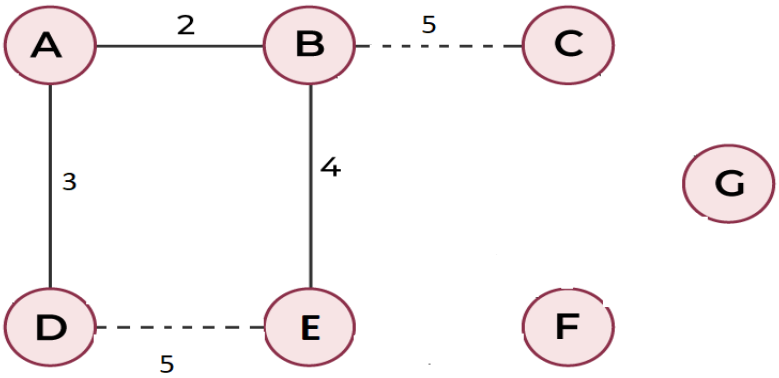
A → B → C =7

A → B → E =6

A → D → E =8

-Here minimum cost is between A → B → E is 6 .

-So mark a connection between them.



-step 4:Next we have node B and its adjacent node as C.

-Also we have node E and its a adjacent node as D and F.

-here

A → B → C =7

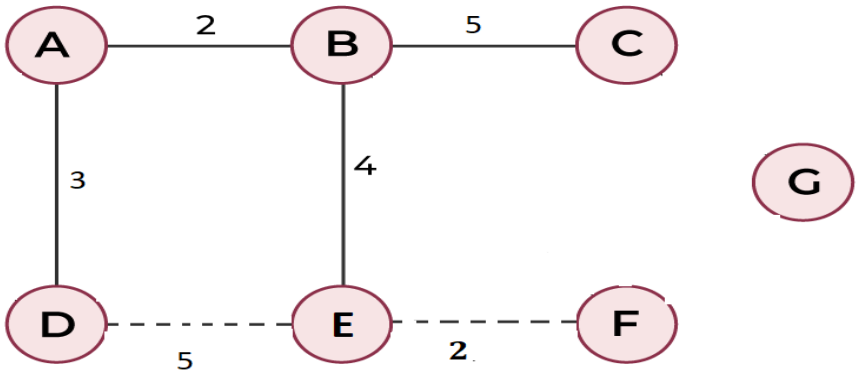
A → B → E → F =8

A → D → E =8

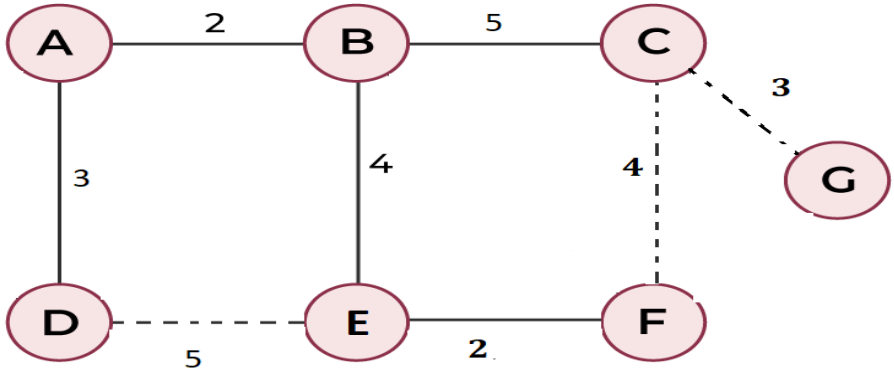
-Here we cannot form a connection between A → D → E because it cause an loop, tree have no loop so we avoid connection between E to D.

-Here minimum cost is between A → B → C =7 .

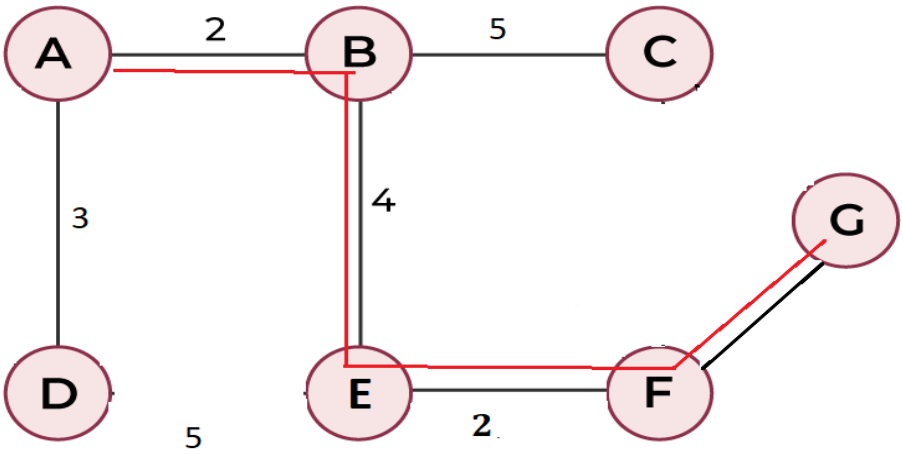
-So mark a connection between them.



-Step 5:Next we have node C and its adjacent nodes G and F.
-Also we have node E and its a adjacent node as F .
-here
A → B → C → G = 10
A → B → C → F = 11
A → B → E → F =8
-Here minimum cost is betweenA → B → E → F is 8 .
-So mark a connection between them.



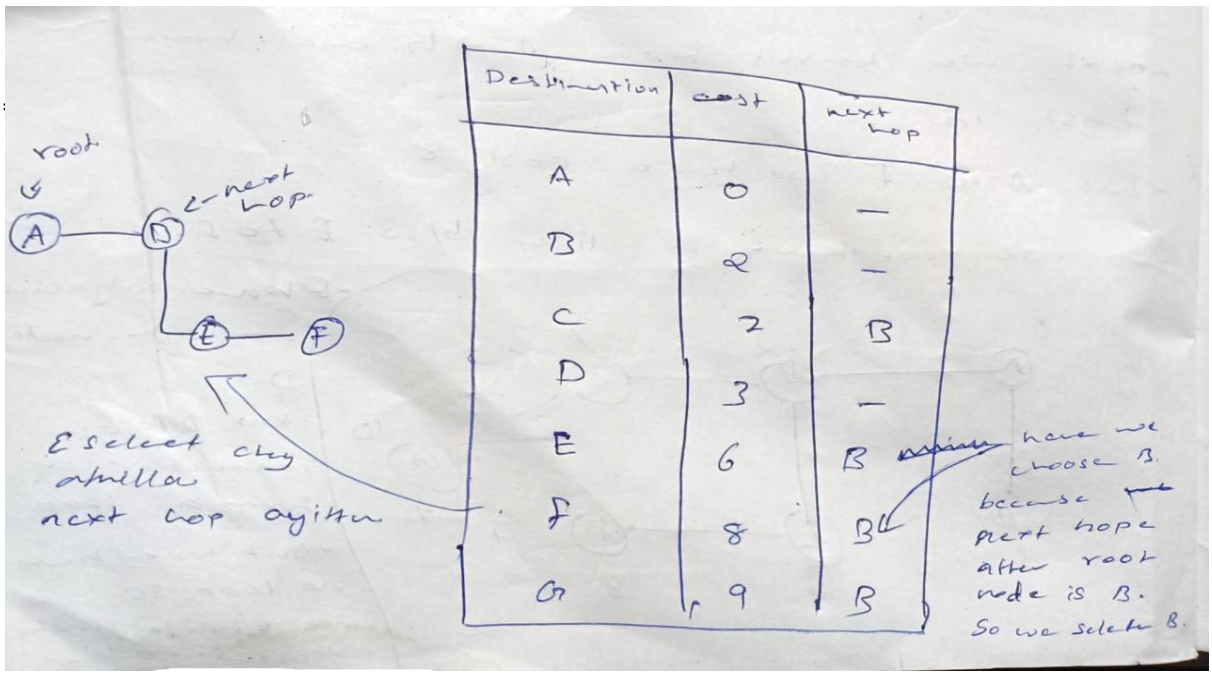
-Step 6:Next we have node C and its adjacent nodes G and F.
-Also we have node F and its a adjacent node as G .
-here
A → B → C → G = 10
A → B → C → F = 11
A → B → E → F→ G =9
--Here we cannot form a connection between A → B → C → F because it cause an loop, tree have no loop so we avoid connection between C to F.
-So we avoid that connection.
-Here minimum cost is between A → B → E → F→ G is 9.
-So mark a connection between them.



-And cancel the path from C to G because we choose path F to G.

-next we have to construct routing table.

Destination	*Cost*	*Next Router*
A	0	—
B	2	—
C	7	B
D	3	—
E	6	B
F	8	B
G	9	B



***OSPF:**-The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on link state routing.

-Its domain is also an autonomous system.

→Areas:-To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas.

-An area is a collection of networks, hosts, and routers all contained within an autonomous system.

-An autonomous system can be divided into many different areas.

-All networks inside an area must be connected.

-Routers inside an area are flooded with routing information.

-At the border of an area, special routers called **area border routers.**

-It summarize the information about the area and send it to other areas.

-the areas inside an autonomous system is a special area called the **backbone**.

-All of the areas inside an autonomous system must be connected to the backbone.

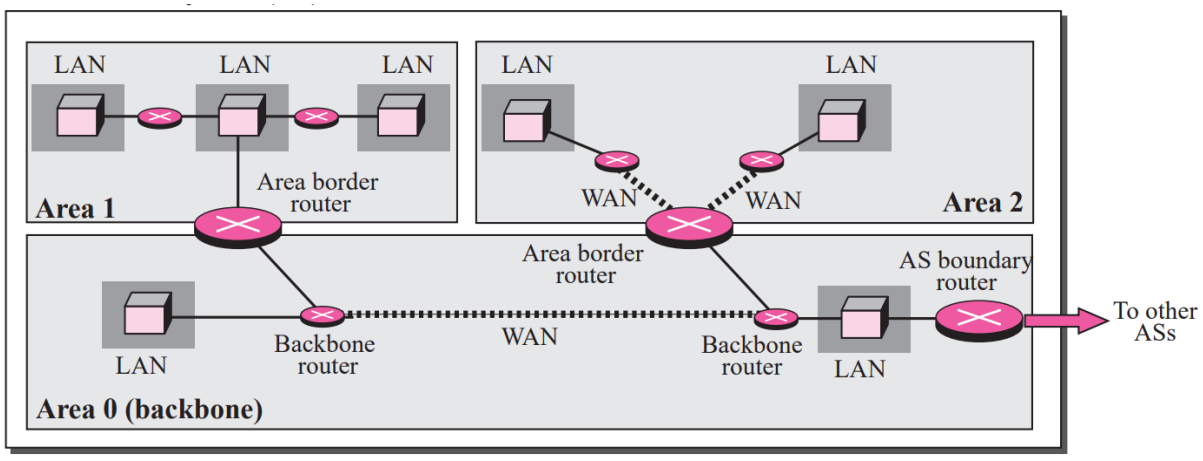
-In other words, the backbone serves as a primary area and the other areas as secondary areas.

-The routers inside the backbone are called the **backbone routers**.

-Because of some problem, the connectivity between a backbone and an area is broken, a virtual link between routers must be created by the administration to allow continuity of the functions of the backbone as the primary area.

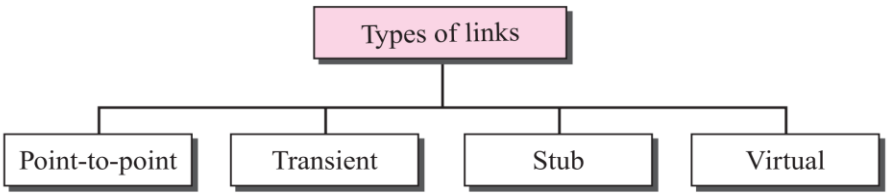
-Each area has an area identification.

-The area identification of the backbone is zero.



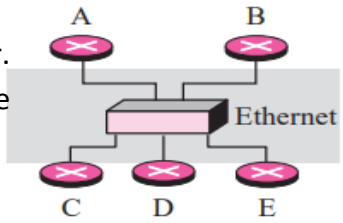
→Types of Links:-In OSPF terminology, a connection is called a **link**.

-Four types of links have been defined: **point-to-point, transient, stub, and virtual**

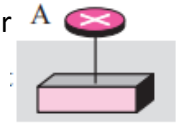


1. **Point-to-Point Link:**-A point-to-point link connects two routers without any other host or router in between.
   -In other words, the purpose of the link (network) is just to connect the two routers.
   -Graphically, the routers are represented by nodes, and the link is represented by a bidirectional edge connecting the nodes.
   -Each router has only one neighbor at the other side of the link.

2. **Transient Link:**-A transient link is a single network with several routers attached to it.
   -Or It is a single network connected with multiple router.
   -The data can enter through any of the routers and leave through any router.
   -In this case, each router has many neighbors.
   -For example, Router A has routers B, C, D, and E as neighbors.
   -Router B has routers A, C, D, and E as neighbors.

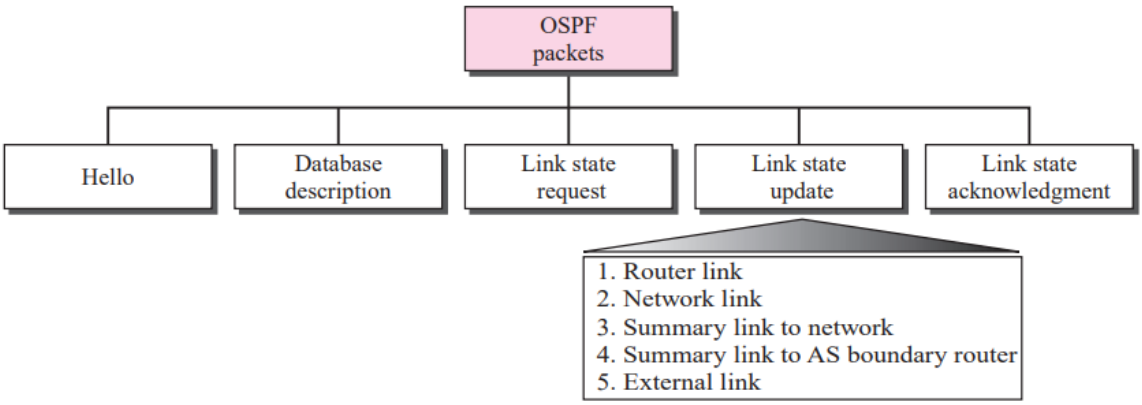


3. **Stub Link:**-A stub link is a single network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router.
   -This is a special case of the transient network.



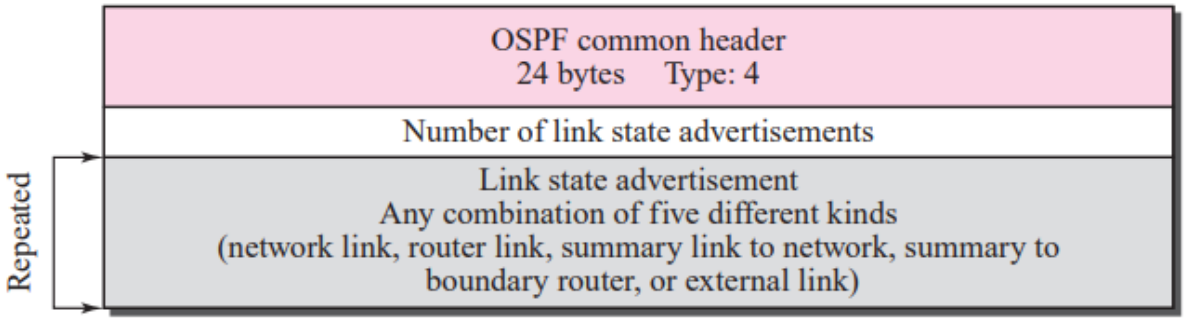
4. **Virtual Link**:-When the link between two routers is broken , weak or damaged.
   -Then administration may create a virtual link between them.

→**OSPF Packets**:-OSPF uses five different types of packets:



1. Hello:- explain cheyandaaa
2. database description :- explain cheyandaaa
3. link state request :- explain cheyandaaa
4. link state update:-The most important one is the link state update that itself has five different kinds.
   -The link state update packet, the heart of the OSPF operation.
   -It is used by a router to advertise the states of its links.

-The general format of the link state update packet .

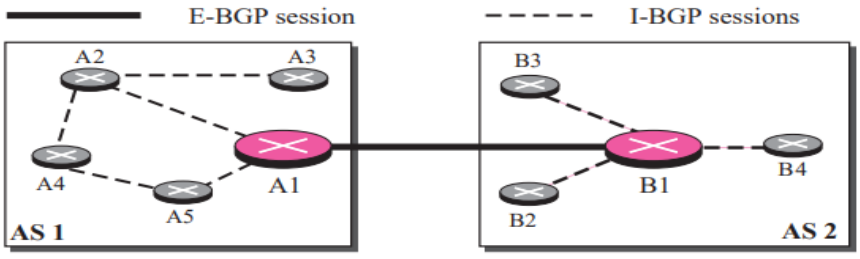


5. link state acknowledgment :- explain cheyandaaa

*** Path vector routing**:-Distance vector and link state routing are both interior routing protocols.
-They can be used inside an autonomous system as intra-domain , but not between autonomous systems.
-Both of these routing protocols become unworkable when the domain of operation becomes large.
-Path vector routing is exterior routing protocol proved to be useful for interdomain.
-The distance vector routing tells us the distance to each network; the path vector routing tells us the path.
-Eg:The difference between the distance vector routing and path vector routing can be compared to the difference between a national map and an international map.
-A national map can tell us the road to each city and the distance to be travelled if we choose a particular route;
-An international map can tell us which cities exist in each country and which countries should be passed before reaching that city.

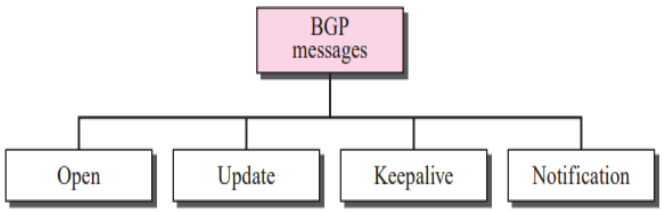
***BGP** :-Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector
routing.
-It first appeared in 1989 and has gone through four versions.
-BGP can have two types of sessions:

1. **External BGP (E-BGP)**:-The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomous systems.
2. **Internal BGP (I-BGP)** :-The IBGP session, on the other hand, is used to exchange routing information between two routers inside an autonomous system.



-The session established between AS1 and AS2 is an E-BGP session.

→Types of Packets/BGP messages:-BGP uses four different types of messages:



- **Open**:-To create a neighborhood relationship, a router running BGP opens connection with a neighbor and sends an **open message**.
  -If the neighbor accepts the neighborhood relationship, it responds with a **keepalive message**,
  -which means that a relationship has been established between the two routers.
- **Update**:-The update message is the heart of the BGP protocol.
  -It is used by a router to withdraw destinations that have been advertised previously, announce a route to a new destination, or both.
  -Note that BGP can withdraw several destinations that were advertised before, but it can only advertise one new destination in a single update message.
- **Keepalive**:-The routers running the BGP protocols exchange keepalive messages regularly to tell each other that they are alive.
- **Notification**:-A notification message is sent by a router whenever an error condition is detected or a router wants to close the connection.

*Types of Autonomous Systems:- The Internet is divided into hierarchical domains called **autonomous systems (ASs**).
-For example, a large corporation that manages its own network and has full control over it is an autonomous system.
-We can divide autonomous systems into three categories:

1. **Stub**:-A stub AS has only one connection to another AS.
   -Or Interconnects with only a single external autonomous system.
2. **Multihomed**:-A multihomed AS has more than one connection to other ASs.
   -Or Interconnects multiple external autonomous systems.
3. **Transit**:-This is an AS that acts as a link between two or more external autonomous systems.

Module -4 -Transport Layer

#Transport Layer

***Transport Layer Services**:-the transport layer is located between the network layer and the application layer.
-The transport layer is responsible for providing services to the application layer.
-It receives services from the network layer.
-The services that can be provided by a transport layer are: (imp)

1. **Process-to-Process Communication**:- The first duty of a transport-layer protocol is to provide process-to-process communication.

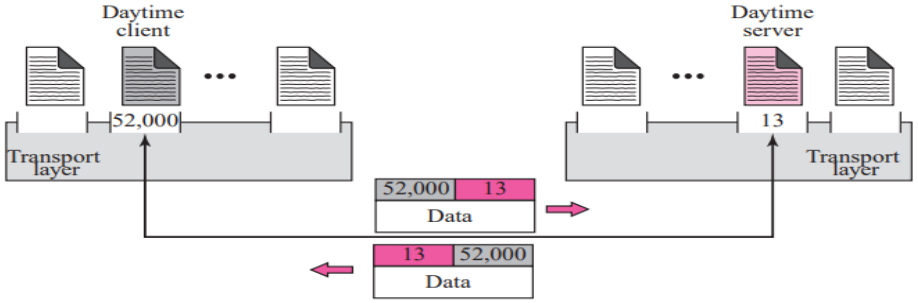
   -The network layer is responsible for communication at the computer level (host-tohost communication).
   -A network layer protocol can deliver the message only to the destination computer.
   -However, this is an incomplete delivery.
   -The message still needs to be handed to the correct process.
   -This is where a transport layer protocol takes over.
   -A transport layer protocol is responsible for delivery of the message to the appropriate process.

2. **Addressing: Port Numbers** :-First we have to define **local host** , **local process** , **Remote host** and **Remote process**.
   -local host and Remote host can be defined using IP address.
   -To access local process and Remote process we use **Port addresses/Port number**.
   Two types of Port address/port number they are;
   - **Ephemeral port number:**-The client program defines itself with a port number, called the ephemeral port number.
     -The word ephemeral means short lived and is used because the life of a client is normally short.
     -An ephemeral port number is recommended to be greater than 1,023 for some client/server programs to work properly.
   - **Well-known port numbers**:-The server process must also define itself with a port number.
     -This port number, however, cannot be chosen randomly.
     -If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client site that wants to access that server and use its services will not know the port number.

-Of course, one solution would be to send a special packet and request the port number of a specific server, but this creates more overhead.
-TCP/IP has decided to use universal port numbers for servers; these are called **well-known port numbers**.

-For example, while the daytime client process can use an ephemeral (temporary) port number 52,000 to identify itself, the daytime server process must use the well-known (permanent) port number 13.

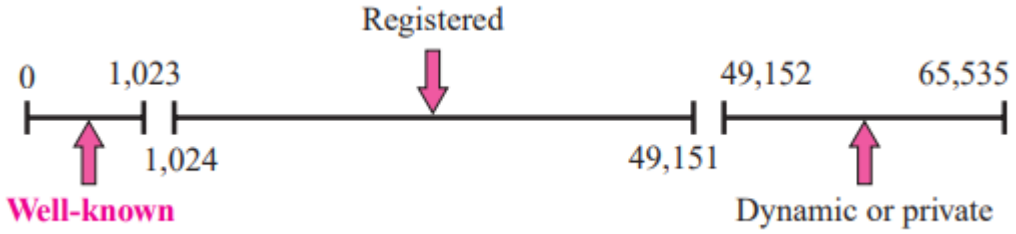
-Figure shows this concept.



→**ICANN Ranges:**-ICANN has divided the port numbers into three ranges:

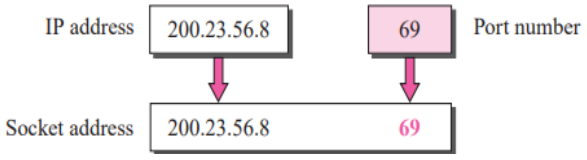
- well-known ports :-The ports ranging from 0 to 1,023 are assigned and controlled by ICANN.
- registered:-The ports ranging from 1,024 to 49,151 are not assigned or controlled by ICANN.
  -They can only be registered with ICANN to prevent duplication.
- dynamic (or private):-The ports ranging from 49,152 to 65,535 are neither controlled nor registered.
  -They can be used as temporary or private port numbers.
  -the ephemeral port numbers for clients be chosen from this range.

- shown in Figure.

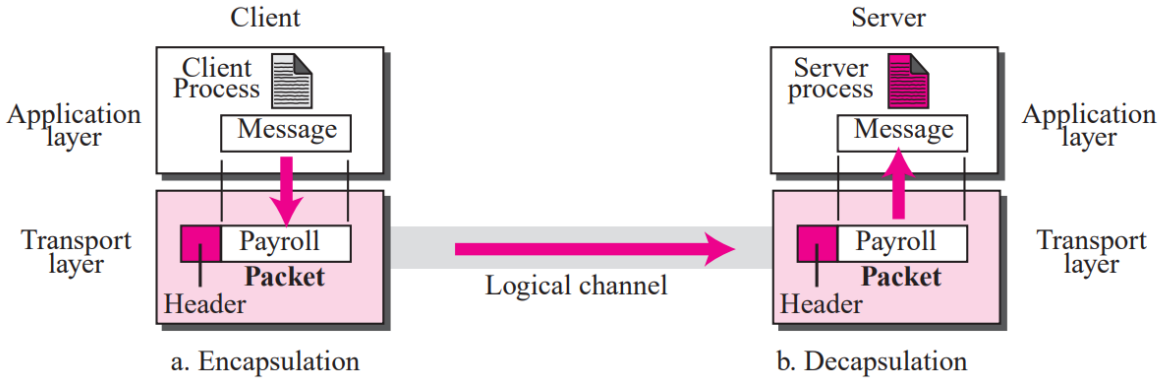


→ **socket address**:-If we need to communicate with two machines we need IP address and port address/port number.

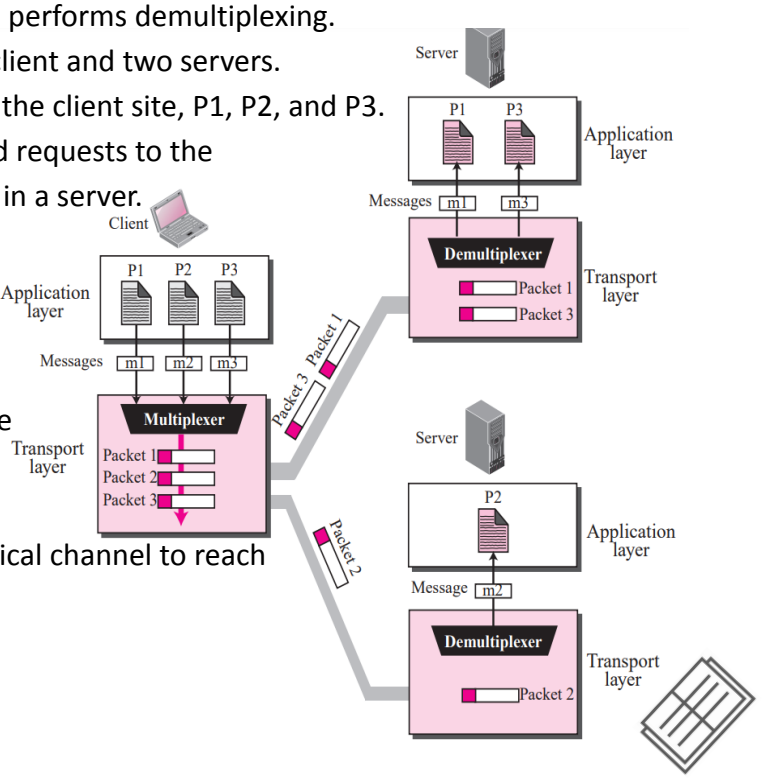
-The combination of an IP address and a port number is called a **socket address**.



3. **Encapsulation and Decapsulation**:- To send a message from one process to another, the transport layer protocol encapsulates and decapsulates messages.
   -Encapsulation happens at the sender site.
   -When a process has a message to send, it passes the message to the transport layer along with a pair of socket addresses and some other pieces of information that depends on the transport layer protocol.
   -The transport layer receives the data and adds the transport-layer header.
   -Decapsulation happens at the receiver site.
   -When the message arrives at the destination transport layer, the header is dropped and the transport layer delivers the message to the process running at the application layer.



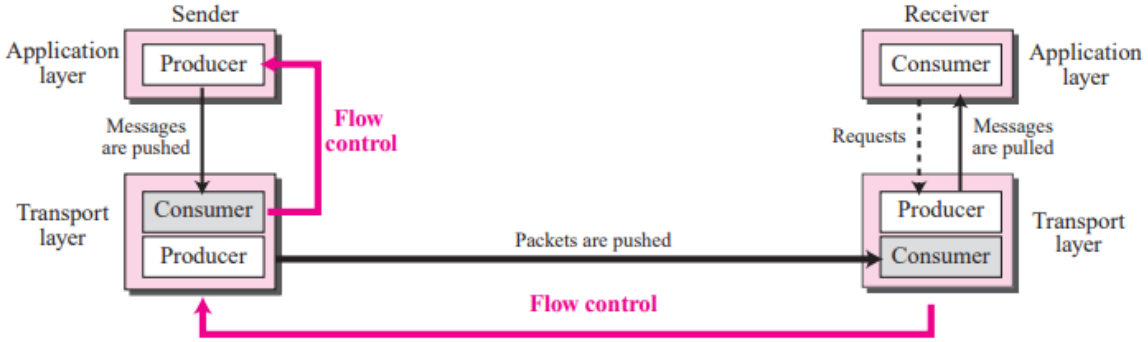
4. **Multiplexing and Demultiplexing** :-Whenever an entity accepts items from more than one source, it is referred to as multiplexing (many to one).
   -whenever an entity delivers items to more than one source, it is referred to as demultiplexing (one to many).
   -The transport layer at the source performs multiplexing.
   -The transport layer at the destination performs demultiplexing.
   Eg:shows communication between a client and two servers.
   -Three client processes are running at the client site, P1, P2, and P3. The processes P1 and P3 need to send requests to the corresponding server process running in a server. The client process P2 needs to send a request to the corresponding server process running at another server.
   -The transport layer at the client site accepts three messages from the three processes and creates three packets.
   -It acts as a multiplexer.
   -The packets 1 and 3 use the same logical channel to reach the transport layer of the first server.



-When they arrive at the server, the transport layer does the job of a multiplexer and distributes the messages to two different processes.
-The transport layer at the second server receives packet 2 and delivers it to the corresponding process.

5. **Flow Control** :-If the items are produced faster than they can be consumed, the consumer can be overwhelmed and needs to discard some items.
-If the items are produced slower than they can be consumed, the consumer should wait; the system becomes less efficient.
-We need to prevent losing the data items at the consumer site.

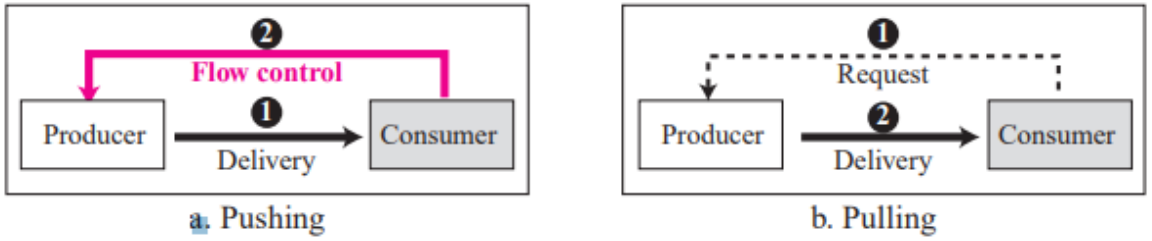
- In communication at the transport layer, we are dealing with four entities: **sender process, sender transport layer, receiver transport layer, and receiver process.**
-The sending process at the application layer is only a producer.
-It produces message chunks and pushes them to the transport layer.
-The sending transport layer has a double role: it is both a consumer and the producer.
-It consumes the messages pushed by the producer.
-It encapsulates the messages in packets and pushes them to the receiving transport layer.
-The receiving transport layer has also a double role: it is the consumer for the packets received from the sender.
-It is also a producer; it needs to decapsulate the messages and deliver them to the application layer.



→**Buffers:**-Although flow control can be implemented in several ways, one of the solutions is normally to **use two buffers**.
-One at the sending transport layer and the other at the receiving transport layer.
-A buffer is a set of memory locations that can hold packets at the sender and receiver.
-The flow control communication can occur by sending signals from the consumer to producer.

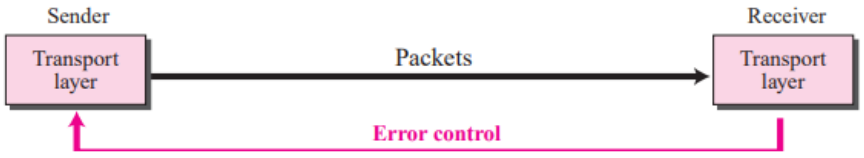
-When the buffer of the sending transport layer is full, it informs the application layer to stop passing chunks of messages; when there are some vacancies, it informs the application layer that it can pass message chunks again.
-When the buffer of the receiving transport layer is full, it informs the sending transport layer to stop sending packets.
-When there are some vacancies, it informs the sending transport layer that it can send message again.

→**Pushing or Pulling**:-Delivery of items from a producer to a consumer can occur in one of the two ways: pushing or pulling.
-If the sender delivers items whenever they are produced without the prior request from the consumer the delivery is referred to as **pushing**.
-If the producer delivers the items after the consumer has requested them, the delivery is referred to as **pulling.**
-When the producer pushes the items, the consumer may be overwhelmed and there is a need for flow control, in the opposite direction, to prevent the discarding of the items.
-In other words, the consumer needs to warn the producer to stop the delivery and to inform it when it is ready again to receive the items.
-When the consumer pulls the items, it requests them when it is ready.
-In this case, there is no need for flow control.



6. **Error Control** :-The responsible to carry the packets from the sending transport layer to the receiving transport layer, is unreliable, we need to make the transport layer reliable if the application requires reliability.
   -Reliability can be achieved to add error control service to the transport layer.
   -Error control at the transport layer is responsible to
   - Detecting and discard corrupted packets.
   - Keep track of lost and discarded packets and resend them.
   - Recognize duplicate packets and discard them.

   -The figure shows the error control between the sending and receiving transport layer.



-The receiving transport layer manages error control, most of the time, by informing the sending transport layer about the problems.
-We can control error using **Sequence number** and **Acknowledgment.**

- **Sequence Numbers**:-Error control requires that the sending transport layer knows which packet is to be resent and the receiving transport layer knows which packet is a duplicate, or which packet has arrived out of order.
  -This can be done if the packets are numbered.
  -Each packets is divided into segments .
  -Each segments have a sequence number.
  -It help to control the error.
- **Acknowledgment:-** The receiver side can send an acknowledgement (ACK) for each or a collection of packets that have arrived safe and sound.
  -The receiver can simply discard the corrupted packets.
  -The sender can detect lost packets if it uses a timer.
  -When a packet is sent, the sender starts a timer; when the timer expires, if an ACK does not arrive before the timer expires, the sender resends the packet.
  -Duplicate packets can be silently discarded by the receiver.
  -Out-of-order packets can be either discarded (to be treated as lost packets by the sender), or stored until the missing ones arrives.

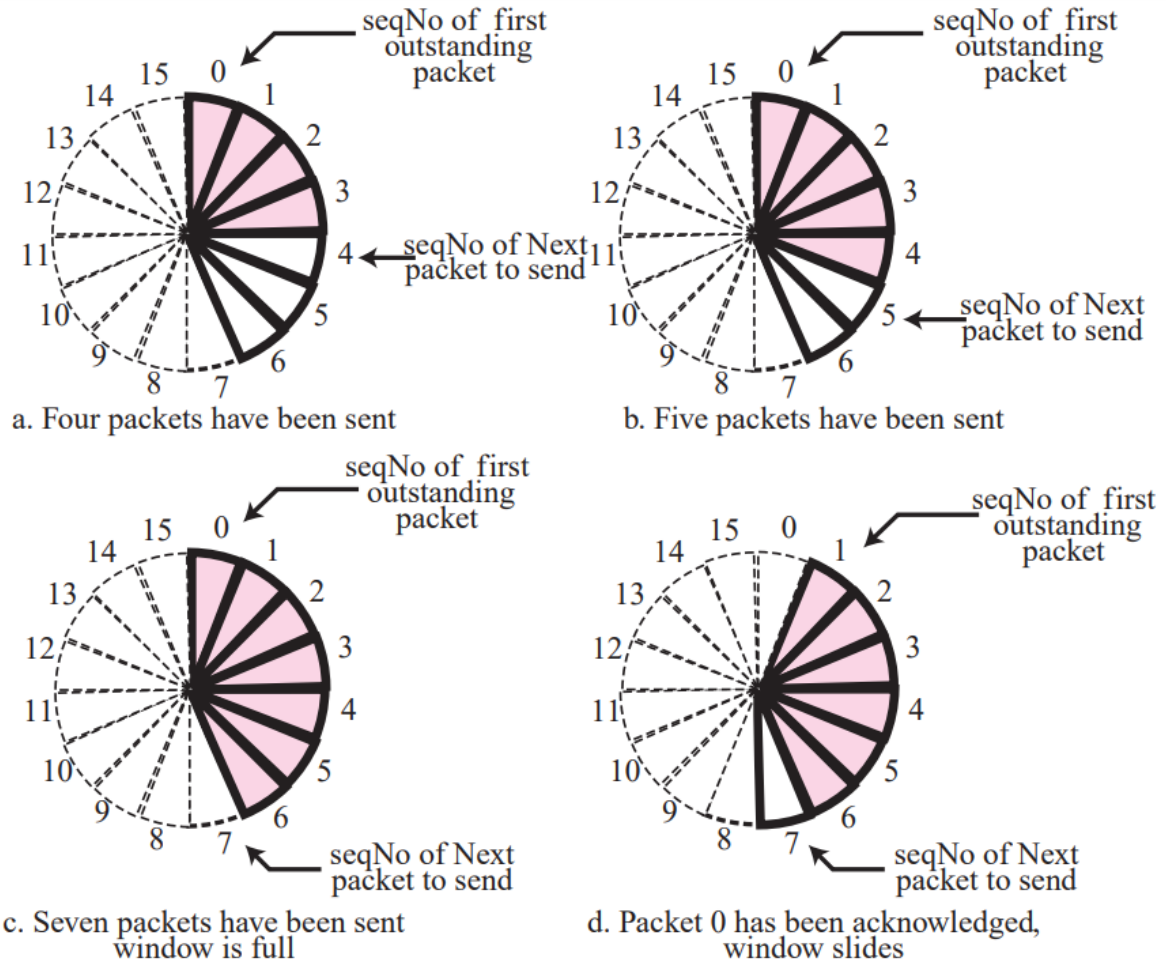
7. **Combination of Flow and Error Control** :-We know that flow control requires the use of two buffers, one at the sender site and the other at the receiver site.
   -We also Know that the error control requires the use of sequence and acknowledgment numbers by both sides.
   -These two requirements can be combined if we use **two numbered buffers, one at the sender, one at the receiver.**
   -At the sender, when a packet is prepared to be sent, we use the number of the next free location, x, in the buffer as the sequence number of the packet.
   -When the packet is sent, a copy is stored at memory location x, awaiting the acknowledgment from the other end.
   -When an acknowledgment related to a sent packet arrives, the packet is purged and the memory location becomes free.
   -At the receiver, when a packet with sequence number y arrives, it is stored at the memory location y until the application layer is ready to receive it.
   -An acknowledgment can be sent to announce the arrival of packet y.

   →**Sliding Window protocol** (9 marks imp):-The buffer is represented as a set of slices, called the **sliding window**.
   -At the sender site, when a packet is sent, the corresponding slice is marked.

-When all the slices are marked, it means that the buffer is full and no further messages can be accepted from the application layer.

-When an acknowledgment arrives, the corresponding slice is unmarked.

- Figure shows the sliding window at the sender.



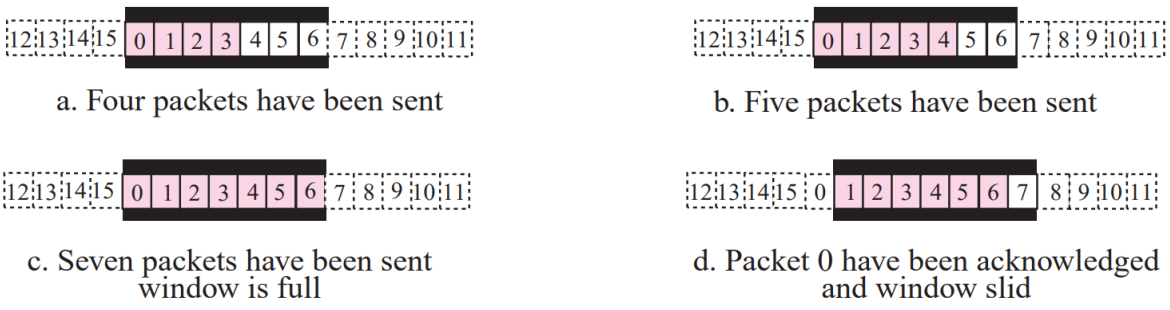
-The sequence number are modulo 16 (m = 4) and the size of the window is 7

note:For error control, the sequence numbers are modulo 2m, -where m is the size of the sequence number field in bits. Eg:For example, if m is 4, the only sequence numbers are 0 to 15.

-Most protocols show the sliding window using linear representation.

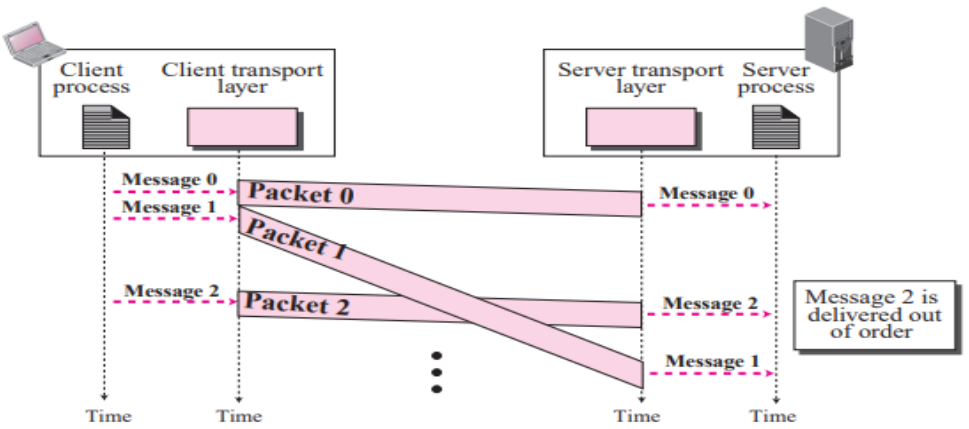
-The idea is the same, but it normally takes less space on paper.

-Both representations tell us the same thing.



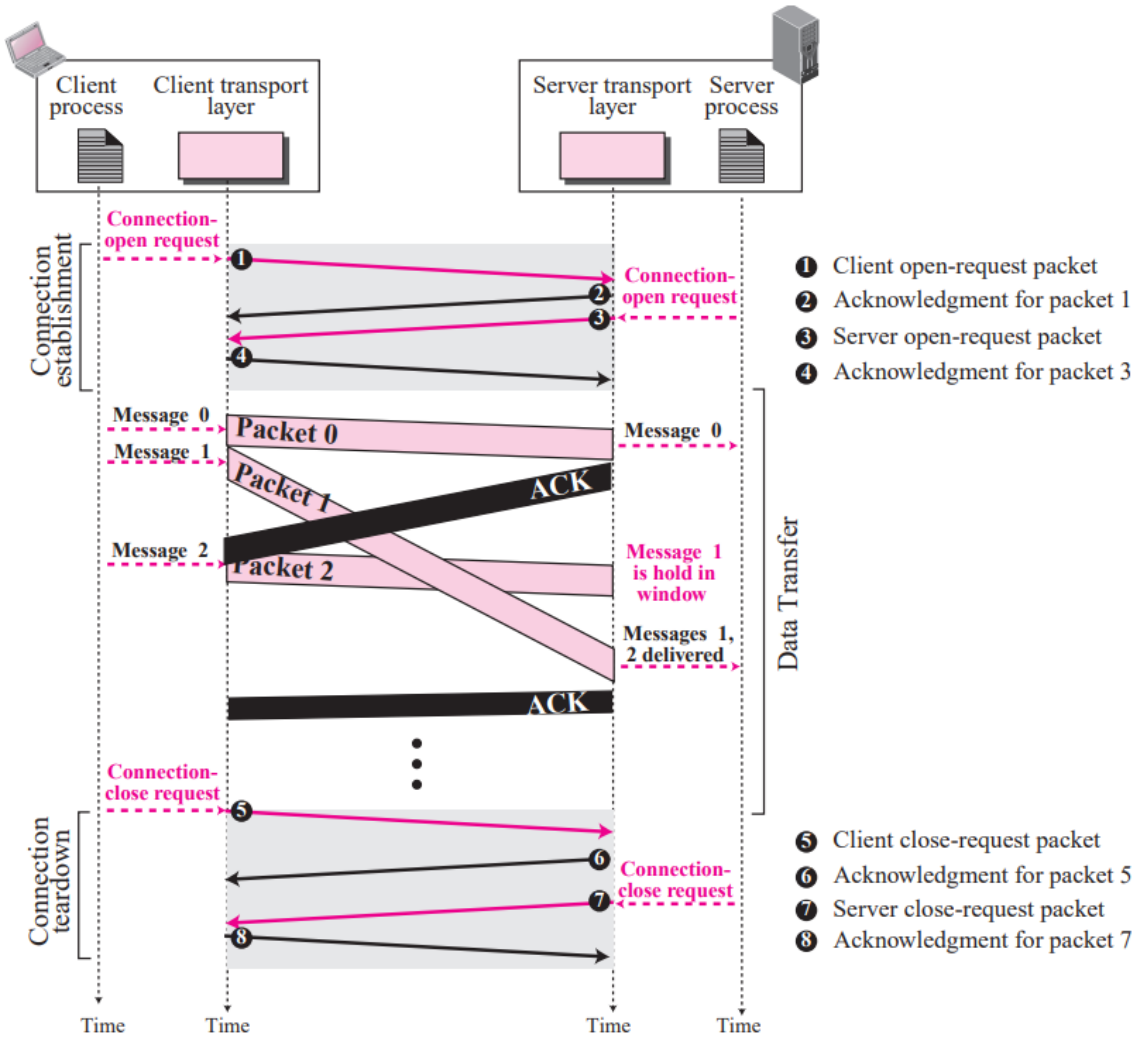
8. **Congestion Control** :-An important issue in the Internet is congestion.
   -It may occur if the load on the network—the number of packets sent to the network—is greater than the capacity of the network—the number of packets a network can handle.
   -Or it occurs due to overloading.
   -Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.
   -It have two mechanism
   1. **Open-Loop Congestion Control**:-It is used to prevent congestion before it happens.
   2. -In these mechanisms, congestion control is handled by either the source or the destination
   3. **Closed-Loop Congestion Control**:-It is used to avoid congestion.

9. **Connectionless and Connection-Oriented Services** :-It can provide two types of services: connectionless and connection-oriented.
   - **Connectionless Service**:-In a connectionless service, the source process (application program) needs to divide its message into chunks of data of the size acceptable by the transport layer and deliver them to the transport layer one by one.
     -The transport layer treats each chunk as a single unit without any relation between the chunks.
     -When a chunk arrives from the application layer, the transport layer encapsulates it in a packet and sends it.
     -Eg:Assume that a client process has three chunks of messages to send a server process.
     -The chunks are handed over to the connectionless transport protocol in order.
     -However, since there is no dependency between the packets at the transport layer, the packets may arrive out of order at the destination and will be delivered out of order to the server process.



-The figure shows that at the client site, the three chunks of messages are delivered to the client transport layer in order (1, 2, and 3).
-Because of the extra delay in transportation of the second packet, the delivery of messages at the server is not in order (1, 3, 2).
-If these three chunks of data belong to the same message, the server process may have received a strange message.
-The situation would be worse if one of the packets were lost.
-Since there is no numbering on the packets, the receiving transport layer has no idea that one of the messages has been lost.
-The receiving transport layer does not know when the first packet will come nor when all of the packets have arrived.
-We can say that no flow control, error control, or congestion control can be effectively implemented in a connectionless service.

- **Connection-Oriented Service**:-In a connection-oriented service, the client and the server first need to establish a connection between themselves.
-The data exchange can only happen after the connection establishment.
-After data exchange, the connection needs to be teared down.



***Transport layer protocol**:-The transport layer is represented by two protocols:

- **UDP**
- **TCP**

1) UDP:-UDP stands for **User Datagram Protocol**.

-UDP is located between the application layer and the IP layer.

-And serves as the intermediary between the application programs and the network operations.

-UDP is a connectionless, unreliable transport protocol.

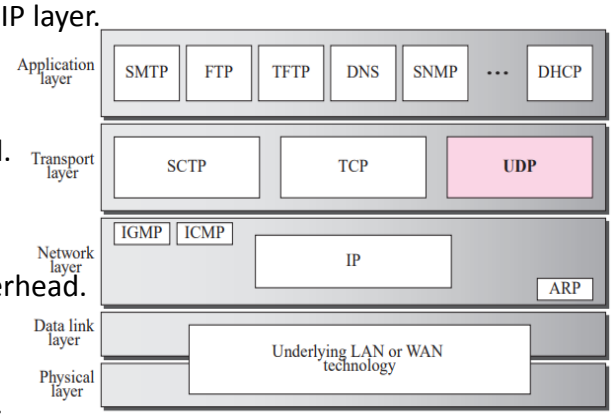
-It is used when reliability and security are less important than speed and size.

-UDP is a very simple protocol using a minimum of overhead.

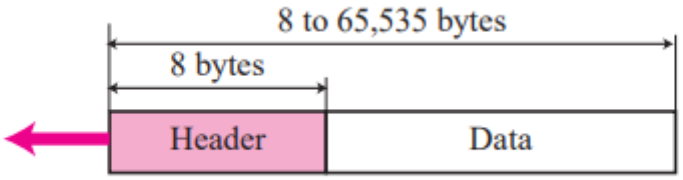
- If a process wants to send a small message and does not care much about reliability, it can use UDP.

-Sending a small message using UDP takes much less time.

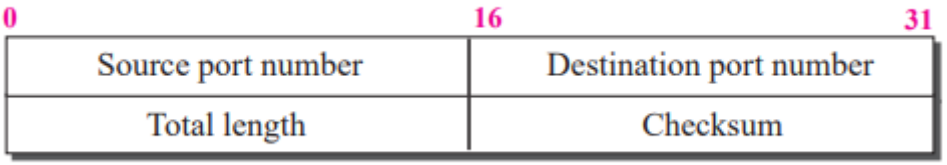
-The packet produced by the UDP protocol is known as a **user datagram**.



→ **User datagram:-UDP packets, called user datagrams**, have a fixed-size header of 8 bytes.



a. UDP user datagram



b. Header format

"UDP uda Header format import annu (9 mark)

>**Source port number**:-It defines the address of the application process that has delivered a message.

-The source port address is of 16 bits address,which means that the port number can range from 0 to 65,535.

-If the source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number.

- If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.

>**Destination port number**:- It defines the address of the application process that will receive the message.

-The destination port address is of a 16-bit address.

- If the destination host is the server (a client sending a request), the port number, in most cases, is a well-known port number.

-If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number.

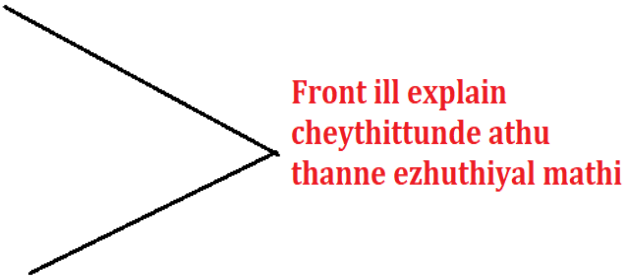
>**Length**:-It defines the total length of the user datagram in bytes.

-The 16 bits can define a total length of 0 to 65,535 bytes.

>**Checksum:**-he checksum is a 16-bit field which is used in error detection.

→**UDP Services:**- the general services are provided by UDP are;

- Process-to-Process Communication
- Connectionless Services:-
- Flow Control:-
- Error Control:-
- Congestion Control :-
- Encapsulation and Decapsulation:-
- Multiplexing and Demultiplexing:-
- Queuing:-google



→**Comparison between UDP and Generic Simple Protocol**

-We can compare UDP with the connectionless simple protocol .

-The only difference is that UDP provides an optional checksum to detect corrupted packets at the receiver site.

-If the checksum is added to the packet, the receiving UDP can check the packet and discard the packet if it is corrupted.

No feedback, however, is sent to the sender.

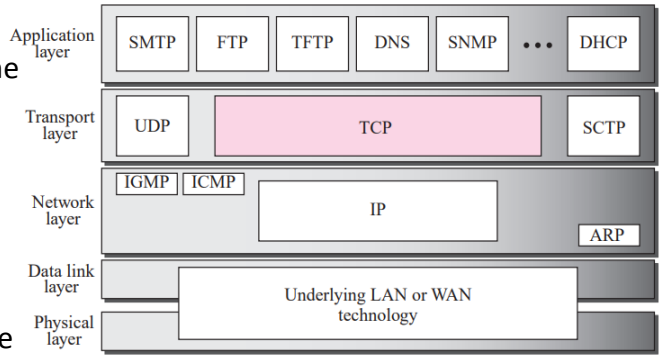
2) TCP:-TCP stands for Transmission Control Protocol.

-It is a connection-oriented protocol means the connection established between both the ends of the transmission.

-First made a connection then transfer data then terminate the connection.

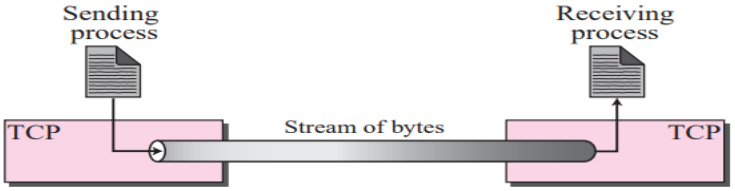
-TCP lies between the application layer and the network layer

-And serves as the intermediary between the application programs and the network operations.



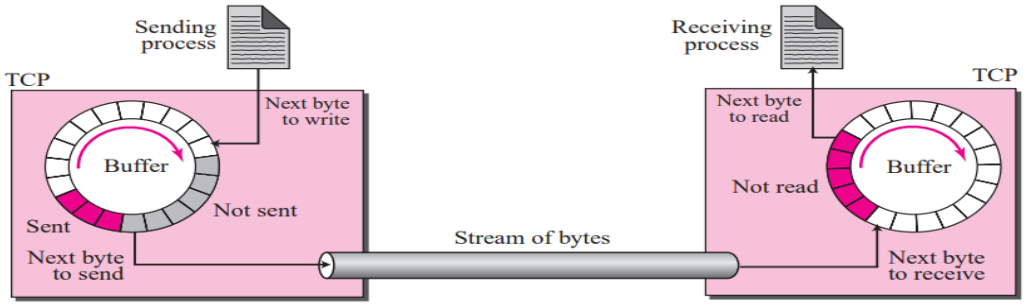
→**TCP Services:**- The services offered by TCP are; (9 mark)

1. **Process-to-Process Communication**:- Front unde.
2. **Stream Delivery Service**:-TCP, unlike UDP, is a stream-oriented protocol.
   -UDP adds its own header to each of these messages and delivers it to IP for transmission.
   -Each message from the process is called a user datagram.
   -TCP, on the other hand, allows the sending process to **deliver data as a stream of bytes** and allows the receiving process to **obtain data as a stream of bytes**.
   -TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their bytes across the Internet.
   -The sending process produces (writes to) the stream of bytes and the receiving process consumes (reads from) them.



   -The sending and the receiving processes may not necessarily write or read data at the same rate, **TCP needs buffers for storage**.
   -**There are two buffers, the sending buffer and the receiving buffer**, one for each direction.
   -These buffers are also necessary for flow- and error-control mechanisms used by TCP.

-One way to implement a buffer is to use a circular array of 1-byte locations as shown in Figure.



-The figure shows the movement of the data in one direction.
-At the sender, the buffer has three types of chambers.
-The white section contains empty chambers that can be filled by the sending process (producer).
-The colored area holds bytes that have been sent but not yet acknowledged.
-The TCP sender keeps these bytes in the buffer until it receives an acknowledgment.
-The shaded area contains bytes to be sent by the sending TCP.
-TCP may be able to send only part of this shaded section.
-This could be due to the slowness of the receiving process, or congestion in the network.
-Also note that after the bytes in the colored chambers are acknowledged, the chambers are recycled and available for use by the sending process.

-The operation of the buffer at the receiver is simpler.
-The circular buffer is divided into two areas (shown as **white and colored**).
-The white area contains empty chambers to be filled by bytes received from the network.
-The colored sections contain received bytes that can be read by the receiving process.
-When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

3. **Full-Duplex Communication**:- TCP offers full-duplex service, where data can flow in both directions at the same time.
   Each TCP endpoint has its own sending and receiving buffer, and segments move in both directions.

4. **Multiplexing and Demultiplexing**:- front ill unde

5. **Connection-Oriented Service** :-TCP, unlike UDP, is a connection-oriented protocol.
   -when a process at site A wants to send to and receive data from another process at site B, the following three phases occur:
   - The two TCPs establish a virtual connection between them.
   - Data are exchanged in both directions.
   - The connection is terminated.

   -Note that this is a virtual connection, not a physical connection.

6. **Reliable Service** :-TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.
   -"bakki **error control** inta bakki ezhuthiyal mathi thaze(👇) explain cheythittunde"

→**TCP features:-**TCP has several features they are; (9mark)

1. **Numbering System** :- the TCP keeps track of the segments being transmitted or received.
   -There are two fields called the **sequence number** and the **acknowledgment number.** These two fields are denoted in byte number.

   Q)Suppose a TCP connection is transferring a file of 5,000 bytes. The first byte is numbered 10,001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1,000 bytes?

   note:
   -sequence number means first byte of the segment.
   eg: 10,001—----------11,000
   here 10,001 is the sequence number.
   -ACK number means it send back the next byte number.
   eg: 10,001—----------11,000
   here ACK number is 11,001.

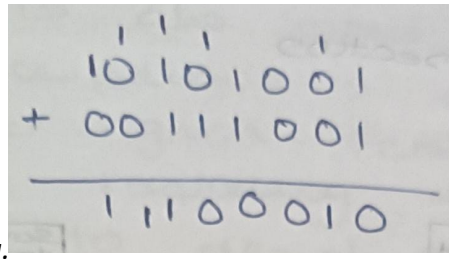
Segment 1	→	Sequence Number:	10,001	Range:	10,001	to	11,000
Segment 2	→	Sequence Number:	11,001	Range:	11,001	to	12,000
Segment 3	→	Sequence Number:	12,001	Range:	12,001	to	13,000
Segment 4	→	Sequence Number:	13,001	Range:	13,001	to	14,000
Segment 5	→	Sequence Number:	14,001	Range:	14,001	to	15,000

2. **Flow Control** :-TCP, unlike UDP, provides flow control. The sending TCP controls how much data can be accepted from the sending process; the receiving TCP controls how much data can to be sent by the sending TCP.
   -Explanation below unde 👇

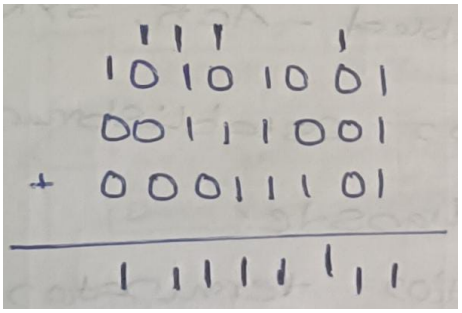
3. **Error Control** :--Explanation below unde 👇
4. **Congestion Control** :--Explanation below unde 👇

→TCP Segment Header:-
- google

Q) suppose the following block of 16 bits to send using a checksum of 8 bit
10101001 00111001 ?



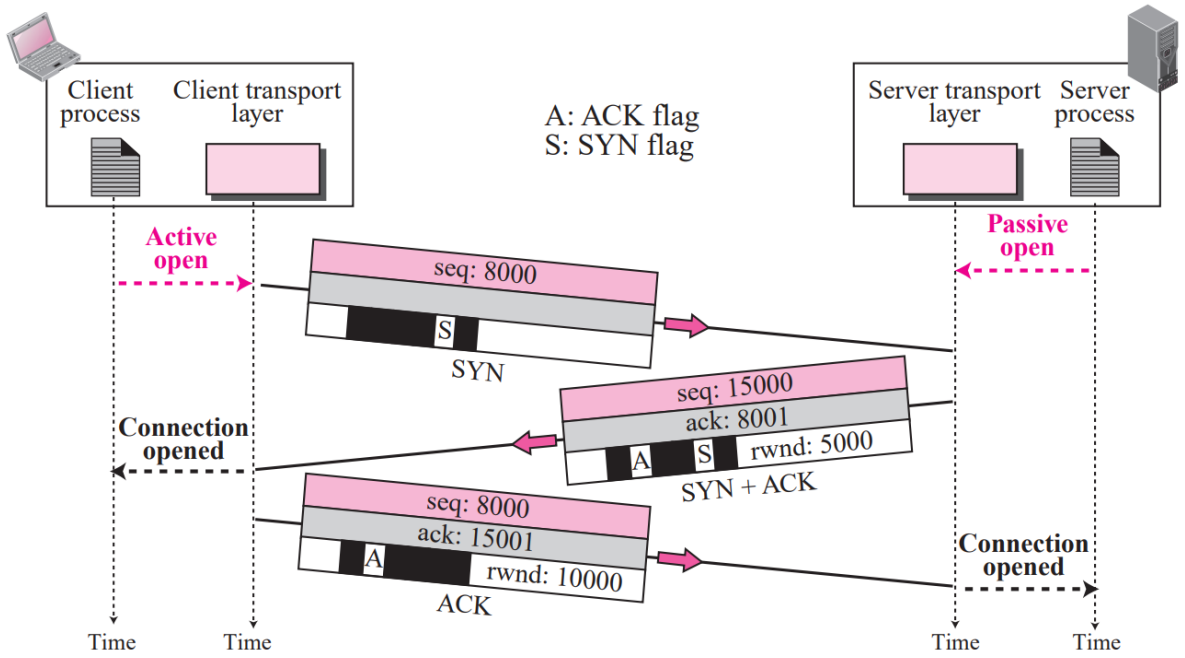
-First add 10101001 + 00111001
=11100010
-And we take its 1's complement =00011101.
-(00011101) it is also called redundant data.
-This complement bit is added to the actual data.
-(10101001 00111001 00011101) = actual data.
-At receiver side receiver received (10101001 00111001 00011101) the data.
-And we add Actual data with redundant data.
-And we take the 1,s complement of the result (11111111).
-Its 1,s complement is 00000000.
-Its all bits are 0 ,it means that there is no error and data is not corrupted.



→**TCP Connection /TCP Connection management**:-TCP is connection-oriented.
-A connection-oriented transport protocol establishes a virtual path between the source and destination.
-All of the segments belonging to a message are then sent over this virtual path.
-Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.
-TCP connection is virtual, not physical.
-If a segment is lost or corrupted, it is retransmitted.
-In TCP, connection-oriented transmission requires three phases:

1. **Connection establishment**:-TCP transmits data in full-duplex mode.
   -The connection establishment in TCP is called **three-way handshaking**.
   →**Three-Way Handshaking**:-The connection establishment in TCP is called three-way handshaking.
   -In our example, an application program, called the **client**, wants to make a connection with another application program, called the **server**, using TCP as the transport layer protocol.

-The process starts with the server.
-The server program tells its TCP that it is ready to accept a connection.
-This request is called a **passive open**.
-Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.



-The client program issues a request for an **active open**.
-A client that wishes to connect to an open server tells its TCP to connect to a particular server.
-Know TCP can now start the three-way handshaking process.
-To show the process we use time lines.
-The three steps in this phase are as follows.

- Step 1:-The client sends the first segment, a **SYN segment**, in which only the SYN flag is set.
  -This segment is for synchronization of sequence numbers and sends this number to the server.
  -This sequence number is called the initial sequence number (ISN).
  -Note that this segment does not contain an acknowledgment number and carries no data.
  -When the data transfer starts, the ISN is incremented by 1.
- Step 2:-The server sends the second segment, a SYN + ACK segment with two flag **SYN and ACK**.
  -The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client (seq-15000).

- The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.

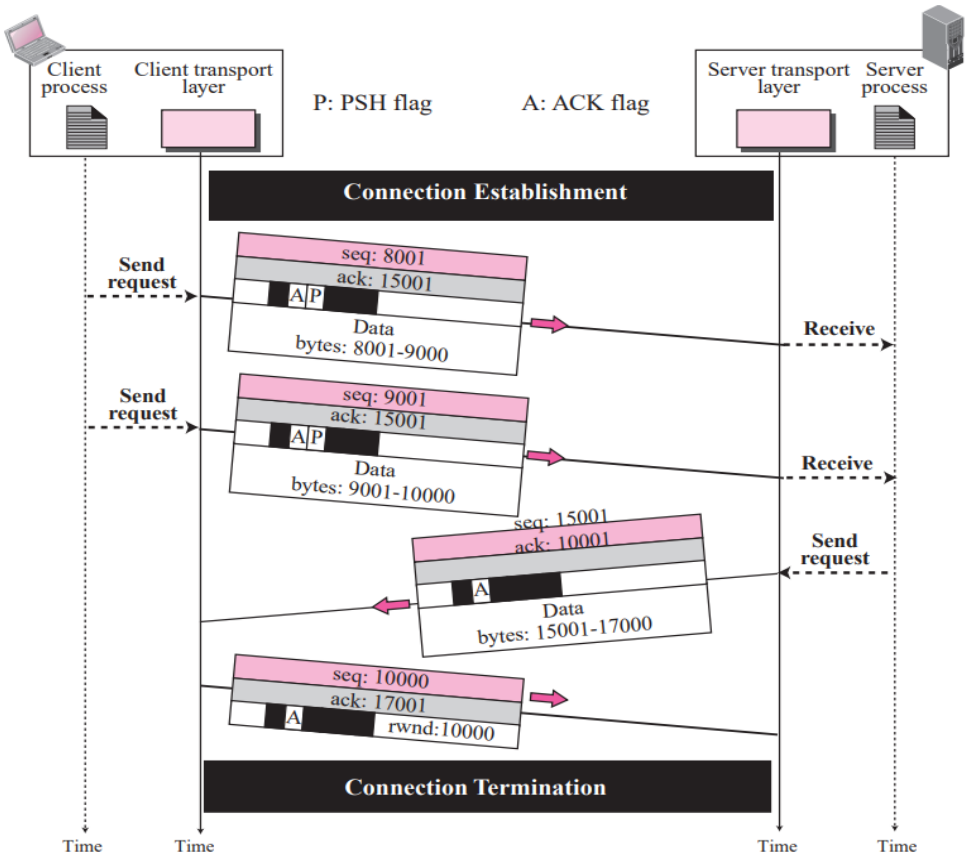
- Step 3:-The client sends the third segment.
  -This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field.
  -Note that the sequence number in this segment is the same as the one in the SYN segment.

→**SYN Flooding Attack**:-A major security issue known as SYN flooding attack can affect the way connections are established in TCP.
-Or It is a problem cause during TCP connection.

-This happens when one or more attackers send a large number of SYN segments to a server pretending that each of them is coming from a different client by faking the source IP addresses .
-The server, assuming that the clients are i**ssuing an active open, allocates the necessary resources,and setting timers.**
-The TCP server then sends the **SYN + ACK segments to the fake clients**,which are lost.
-When the server waits for the third leg of the handshaking process, however, resources are allocated are not used.
- during this short period of time, the server eventually runs out of resources and may be unable to accept connection requests from valid clients.
-It is also called **denial of service attack**.
-The attacker monopolizes a system with so many service requests, then the system overloads and denies service to valid requests.
-TCP have strategies to avoid the effect of a SYN attack they are;

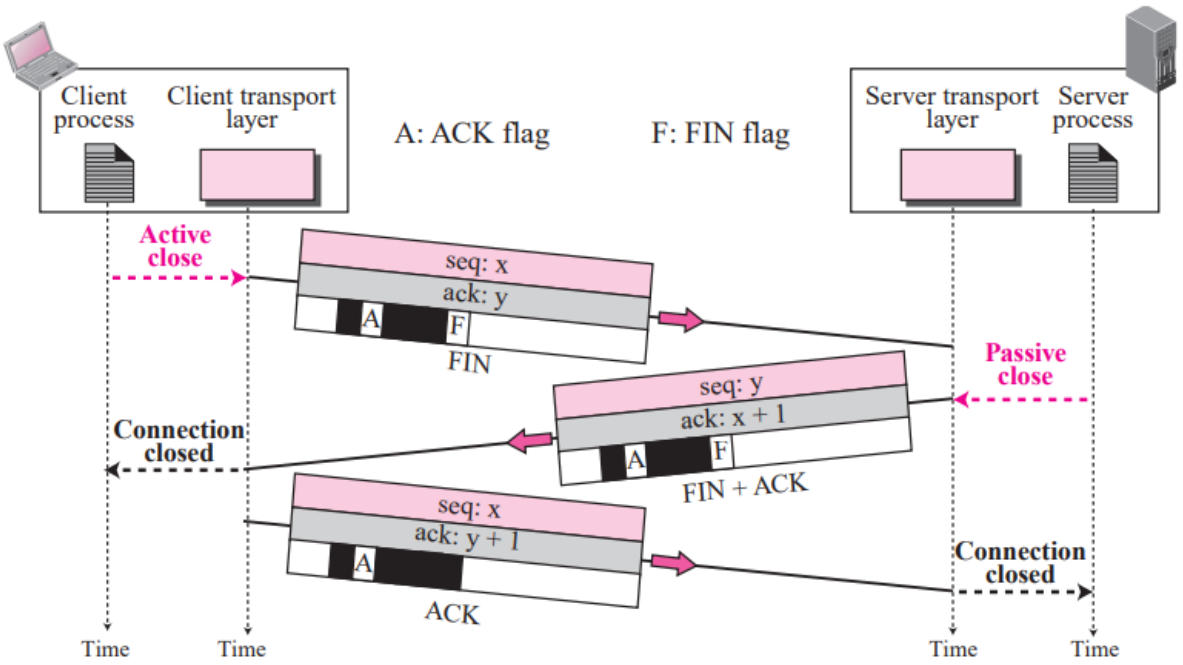
- To limiting connection requests during a specified period of time.
- filter out datagrams coming from unwanted source addresses.
- One recent strategy is to postpone resource allocation until the server can verify that the connection request is coming from a valid IP address, by using what is called a **cookie**.

2. **Data transfer:**-After connection is established, bidirectional data transfer can take place.
   -The client and server can send data and acknowledgments in both directions.



   -In this example, after a connection is established, the client sends 2,000 bytes of data in two segments.
   -The server then sends 2,000 bytes in one segment.
   -The client sends one more segment.
   -The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there is no more data to be sent.
   -The data segments sent by the client have the **PSH (push) flag** set so that the server TCP tries to deliver data to the server process as soon as they are received.
   -URG flag is used to send urgent messages.,By using this flag the urgent data can be placed in the first position in the segment and sent.

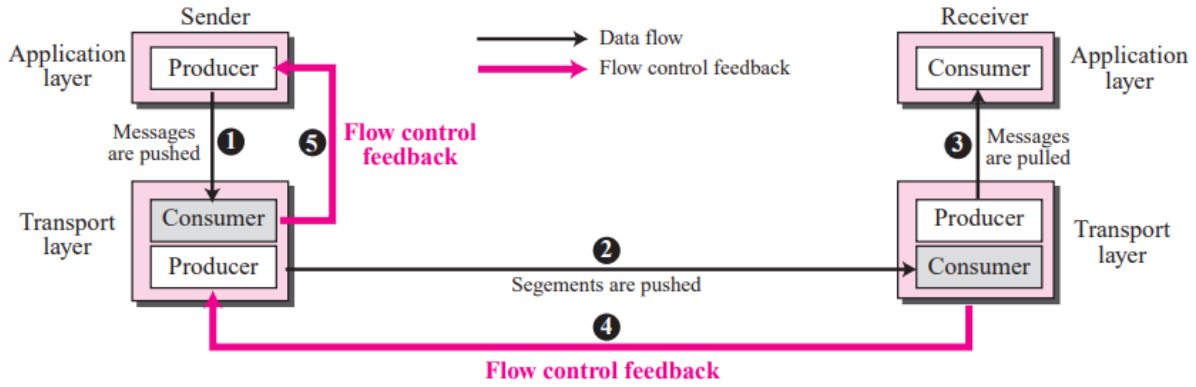
3. **Connection termination**:-Connection Termination is usually initiated by the client.
   -There are two options/methods for connection termination they are;
   - **Three-way handshaking** :-It contain 3 steps they are;
     >Step 1:-After receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.
     -Note that a FIN segment contain only signals, not any data.



>Step 2:-The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN+ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction.
>Step 3:-The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server.
-This segment contains the acknowledgment number, which is one plus the sequence number received in the FIN segment from the server.

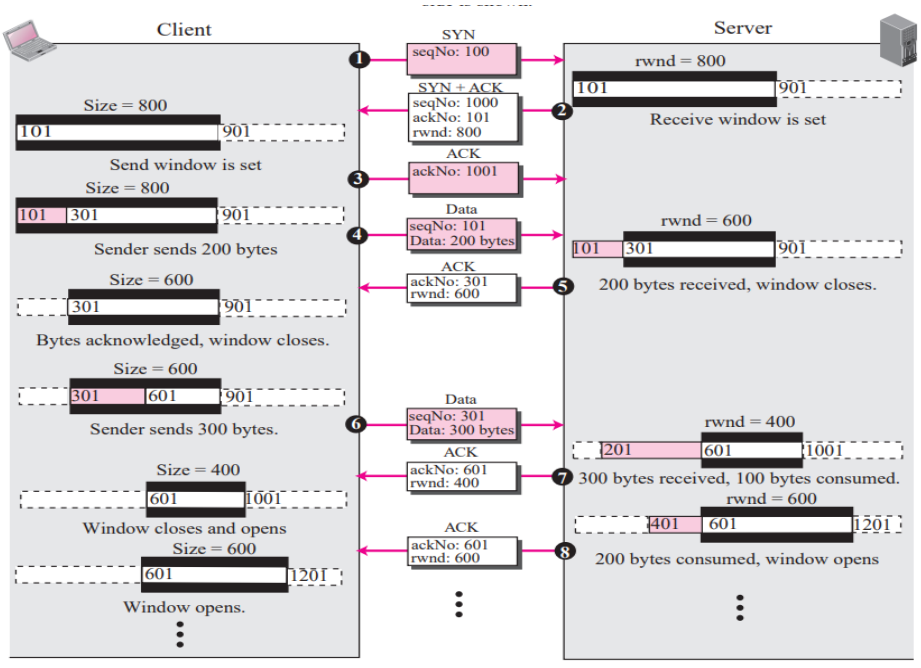
- **Four-way handshaking with a half-close option**:- In TCP, one end can stop sending data while other end can still receiving data.
-This is called a half close.
-Either the server or the client can issue a half-close request.
-It can occur when the server needs all the data before processing can begin.
-A good example is sorting.
-When the client sends data to the server to be sorted, the server needs to receive all the data before sorting can start.
-This means the client, after sending all data, can close the connection in the client-to-server direction.
-However, the server-to-client direction must remain open to return the sorted data.

→**TCP Flow Control**:- flow control balances the rate a producer creates data with the rate a consumer can use the data.



-The figure shows that data travel from the sending process down to the sending TCP, from the sending TCP to the receiving TCP, and from receiving TCP up to the receiving process (paths 1, 2, and 3).
-Flow control feedbacks, however, are traveling from the receiving TCP to the sending TCP and from the sending TCP up to the sending process (paths 4 and 5).
-Flow control feedback from the sending TCP to the sending process (path 5).
→**Opening and Closing Windows:**-To achieve flow control, TCP forces the sender and the receiver to adjust their window sizes, although the size of the buffer for both parties is fixed when the connection is established.
-The receive window closes (moves its left wall to the right) when more bytes arrive from the sender; it opens (moves its right wall to the right) when more bytes are pulled by the process.
-The opening, closing, and shrinking of the send window is controlled by the receiver.
-The send window closes (moves its left wall to the right) when a new acknowledgement allows it to do so.
-The send window opens (its right wall moves to the right) when the receive window size (rwnd) advertised by the receiver allows it to do so.

Eg:Eight segments are exchanged between the client and server:



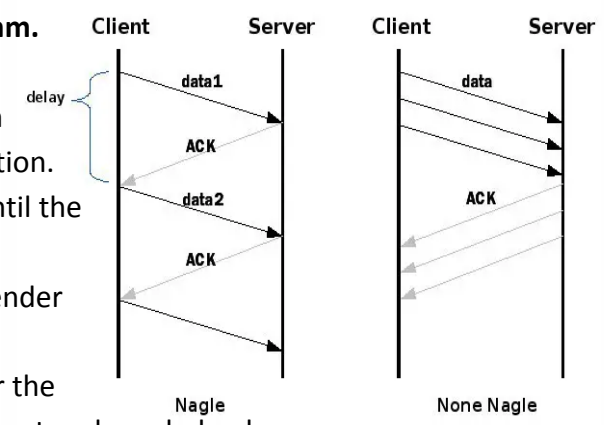
1. The first segment is from the client to the server (a SYN segment) to request connection.
   -The client announces its initial seqNo = 100.
   -When this segment arrives at the server, it allocates a buffer size of 800 (an assumption) and sets its window to cover the whole buffer (rwnd = 800).
   -Note that the number of the next byte to arrive is 101.
2. The second segment is from the server to the client.
   -This is an ACK + SYN segment.
   -The segment uses ackNo = 101 to show that it expects to receive bytes starting from 101.
   -It also announces that the client can set a buffer size of 800 bytes.
3. The third segment is the ACK segment from the client to the server.
4. After the client has set its window with the size (800) dictated by the server, the process pushes 200 bytes of data.
   -The TCP client numbers these bytes 101 to 300.
   -It then creates a segment and sends it to the server.
   -The segment shows the starting byte number as 101 and the segment carries 200 bytes.
   -The window of the client is then adjusted to show 200 bytes of data are sent but waiting for acknowledgment.
   -When this segment is received at the server, the bytes are stored, and the receive window closes to show that the next byte expected is byte 301; the stored bytes occupy 200 bytes of buffer.
5. The fifth segment is the feedback from the server to the client. The server acknowledges bytes up to and including 300 (expecting to receive byte 301). The segment also carries the size of the receive window after decrease (600). The client, after receiving this segment, purges the acknowledged bytes from its window and closes its window to show that the next byte to send is byte 301. The window size, however, decreases to 600 bytes. Although the allocated buffer can store 800 bytes, the window cannot open (moving its right wall to the right) because the receiver does not let it.
6. Segment 6 is sent by the client after its process pushes 300 more bytes. The segment defines seqNo as 301 and contains 300 bytes. When this segment arrives at the server, the server stores them, but it has to reduce its window size. After its process has pulled 100 bytes of data, the window closes from the left for the amount of 300 bytes, but opens from the right for the amount of 100 bytes. The result is that the size is only reduced 200 bytes. The receiver window size is now 400 bytes.
7. In segment 7, the server acknowledges the receipt of data, and announces that its window size is 400. When this segment arrives at the client, the client has no choice but to reduce its window again and set the window size to the value of rwnd = 400 advertised by the server. The send window closes from the left by 300 bytes, and

opens from the right by 100 bytes.

8. Segment 8 is also from the server after its process has pulled another 200 bytes. Its window size increases. The new rwnd value is now 600. The segment informs the client that the server still expects byte 601, but the server window size has expanded to 600. We need to mention that the sending of this segment depends on the policy imposed by the implementation. Some implementations may not allow advertisement of the rwnd at this time; the server then needs to receive some data before doing
so. After this segment arrives at the client, the client opens its window by 200 bytes without closing it. The result is that its window size increases to 600 bytes.

→**Silly Window Syndrome**:-It is a problem that cause flow control of TCP.
-This problem cause because of improper transfer and we can't fully utilize it.
-We solve this problem by using **Nagle's Algorithm.**
-Nagle's Algorithm:-

- Sender should send only the first byte on receiving one byte data from the application.
- Sender should buffer all the rest bytes until the outstanding byte gets acknowledged.
- After receiving the acknowledgement, sender should send the buffered data in one TCP segment. Then, sender should buffer the data again until the previously sent data gets acknowledged.



→**TCP Error Control (9 marks)** :-TCP provides reliability using error control.
-Error control includes mechanisms for detecting and resending corrupted segments, resending lost segments, storing out-of-order segments until missing segments arrive, and detecting and discarding duplicated segments.
-Error control in TCP is achieved through the use of three simple tools:

1. **Checksum**:-Each segment includes a checksum field, which is used to check that segment is corrupted or not.
   -Add extra bits to find error.
   -If a segment is corrupted ,then segment is discarded by the destination TCP and is considered as lost.
2. **Acknowledgment**:-It is used to know that the segment is received properly or not.
3. **Retransmission**:-The heart of the error control mechanism is the retransmission of segments.
   -Retransmit data again if failure occur within a time limit.
   -Bakki mukalill unde

→**TCP Congestion control (9 marks)** :-Congestion control in TCP is based on both open-loop and closed-loop mechanisms.

-Congestion control refers to techniques and mechanisms that can-

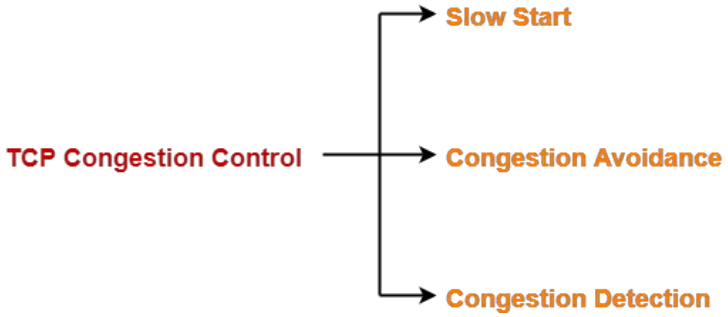
- ➢ Either prevent congestion before it happens
- ➢ Or remove congestion after it has happened

-It use a **congestion window** and a **congestion policy** that avoid congestion and detect and alleviate congestion after it has occurred.

- **Congestion Window**:-Sender should not send data greater than congestion window size.
  -Otherwise, it leads to dropping the TCP segments which causes TCP Retransmission.
  -So, sender should always send data less than or equal to congestion window size.
  -Congestion window is known only to the sender and is not sent over the limit.

Sender window size = Minimum (Receiver window size, Congestion window size) **= Minimum (rwnd, cwnd)**

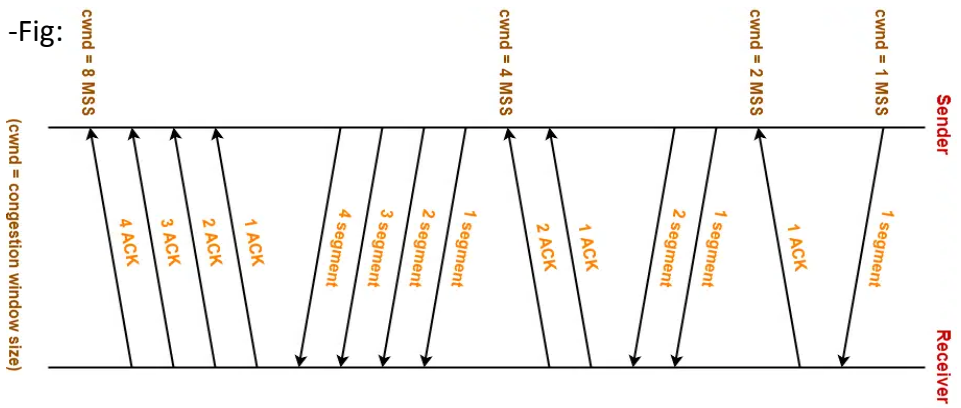
- **Congestion policy**:-TCP's general policy for handling congestion consists of following three phases-



1. **Slow Start**:-Initially, sender sets congestion window size = Maximum Segment Size (1 MSS).
   -After receiving each acknowledgment, sender increases the congestion window size by 1 MSS.
   -In this phase, the size of congestion window increases exponentially.

The followed formula : **Congestion window size = Congestion window size + Maximum segment size**

-Fig:

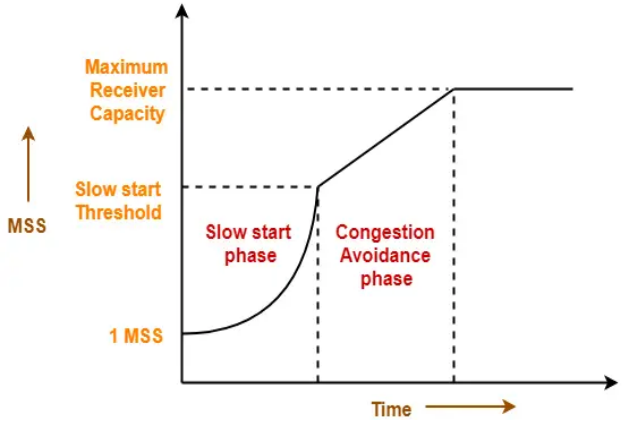


-This phase continues until the congestion window size reaches the slow start threshold.

2. **Congestion Avoidance**:-After reaching the threshold,
   -Sender increases the congestion window size linearly to avoid the congestion.
   -On receiving each acknowledgement, sender increments the congestion window size by 1.
   -The followed formula

Congestion window size = Congestion window size + 1

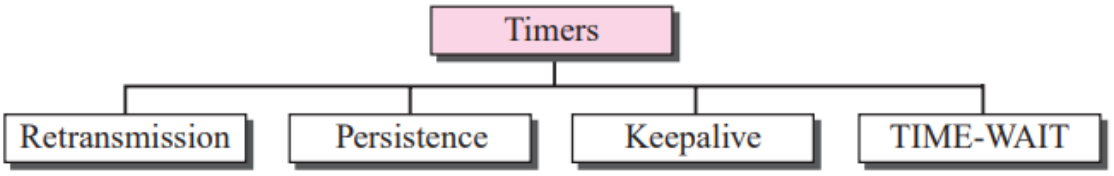
   -This phase continues until the congestion window size becomes equal to the receiver window size.



3. **Congestion Detection**:-When sender detects the loss of segments, it reacts in different ways depending on how the loss is detected
   - Case-01: Detection On Time Out:-Time Out Timer expires before receiving the acknowledgement for a segment.
     -This case suggests the stronger possibility of congestion in the network.
     -There are chances that a segment has been dropped in the network.

   - Case-02: Detection On Receiving 3 Duplicate Acknowledgements:-Sender receives 3 duplicate acknowledgements for a segment.
     -This case suggests the weaker possibility of congestion in the network.
     -There are chances that a segment has been dropped but few segments sent later may have reached.

→**TCP timers (9 marks)** :-TCP uses several timers to ensure that excessive delays are not encountered during communications.
To perform its operation smoothly, most TCP implementations use at least four timers they are:



1. **Retransmission Timer** :-To retransmit lost segments, TCP uses retransmission timeout (RTO).
   -When TCP sends a segment the timer starts and stops when the acknowledgment is received.
   -If sender does not receives any acknowledgement and the timer goes off, then the segment is retransmitted.
   -Sender retransmits the same segment and resets the timer.
2. **Persistence Timer** :-TCP uses a persistent timer to deal with a **zero-widow-size deadlock situation**.
   -When the sending TCP receives an acknowledgment with a window size of zero, it starts a persistence timer.
   -When the persistence timer goes off, the sending TCP sends a special segment called a **probe**.
   -The probe causes the receiving TCP to resend the acknowledgment which was lost.
3. **Keepalive Timer** :-A keepalive timer is used to prevent a long idle connection between two TCPs.
   -If a client opens a TCP connection to a server, transfers some data and becomes silent, because the client crashes. In this case, the connection remains open forever.
   -So a keepalive timer is used.
   -Each time the server hears from a client, it resets this timer.
   -The time-out is usually 2 hours.
   -If the server does not hear from the client after 2 hours, it sends a **probe segment**.
   -If there is no response after 10 probes, it assumes that the client is down and terminates the connection.
4. **TIME-WAIT Timer** :-his timer is used during tcp connection termination.
   -The timer starts after sending the last Ack for 2nd FIN and closing the connection.
   -After a TCP connection is closed, it is possible for datagrams that are still making their way through the network to attempt to access the closed port.
   -The quiet timer is intended to prevent the just-closed port from reopening again quickly and receiving these last datagrams.

Join for more MCA short note : https://t.me/mgu_mca_shortnote

@MGU_MCA_SHORTNOTE