DATA COMMUNICATIONS AND NETWORKS

B.TECH

(III YEAR – I SEM)

(2023-24)

Prepared by

Mr.Shaik.Sohel Pasha,

Assistant Professor

Department of Electronics and Communication Engineering

MALLA REDDY COLLEGE OF ENGINEERING

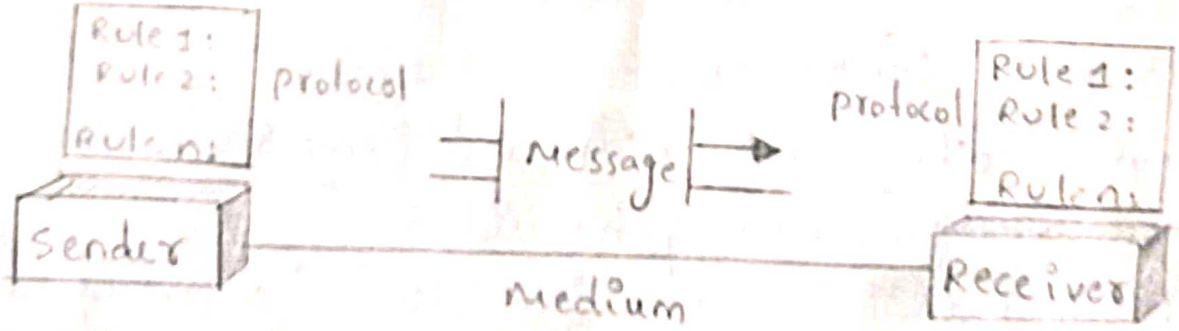
Introduction to Data communications

The word data refers to information presented in what-ever form is agreed upon by the parties creating and using the data

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

Data communications (DC) is the process of using computing and communicating technologies to transfer data from one place to another, and vice versa, It enables the movement of electronic or digital data between two or more nodes, regardless of geographical location, technological medium or data contents

components:- A data communications system has five components



1) message: The message is the information (data) to be communicated. Popular forms of information include

text, numbers, pictures, audio, and video.

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

3) Receiver :- The receiver is the device that receives the message. It can be a computer, workstation, telephone hand set, television, and so on

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

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

Data Representation :-

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

Text :- In data communications, text is represented as a bit pattern, a sequence of bits. Different sets of bit patterns have been designed to represent text symbols.

Each set is called a code, and the process of representing symbols is called coding. Today, the prevalent coding system is called unicode, which uses 32 bits to represent a symbol or character used in any language in the world. The American standard code for information interchange (ASCII), developed some decades ago in the united states, now constitute the first 127 characters in unicode and is also referred to as Basic Latin.

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

Images: Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels, where each pixel is a small dot. The size of the pixel depends on the resolution. For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image, but more memory is needed to store the image.

Audio: Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal.

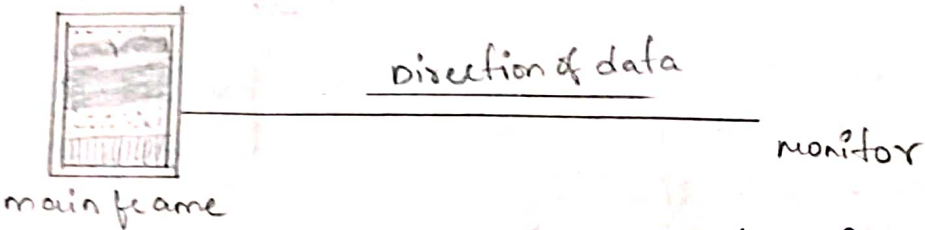
video: video refers to the recording or broadcasting of a picture or movie. video can either be produced as a

continuous entity or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion. Again we can change video to a digital or an analog signal.

Data flow :-

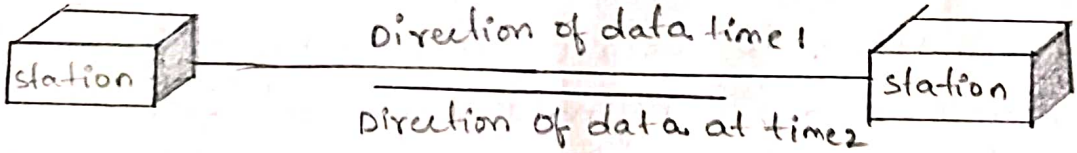
communication between two devices can be simplex, half duplex or full duplex as shown in figure.

a. simplex :-



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

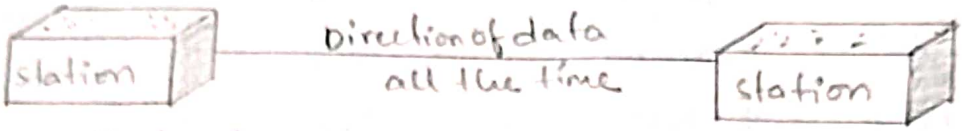
b. half duplex :-



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

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

c) Full duplex:-



In full duplex both stations can transmit and receive simultaneously. The full duplex mode is like a two-way street with traffic flowing in both directions at the same time. In full duplex mode, signals going to one direction share the capacity of the link: with signals going in the other directions.

Networks:- A network is a set of devices connected by communication links. A node can be a computer, printer, or any other devices or sending and/or receiving data generated by other nodes on the network

Distributed processing: most networks use distributed processing, in which a task is divided among multiple computers, instead of one single large machine being responsible for all aspects of a process, separate computers handle a subset.

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

Performance:- performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response. the performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capability of the connected hardware, and the efficiency of the software

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

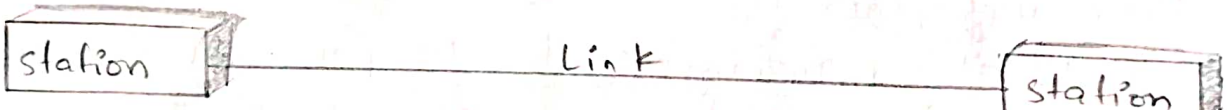
Security:- Network security issue include protecting data from unauthorized access, protecting data from damage & development, and implementing policies & procedures for recovery from breaches and data losses.

physical structures:-

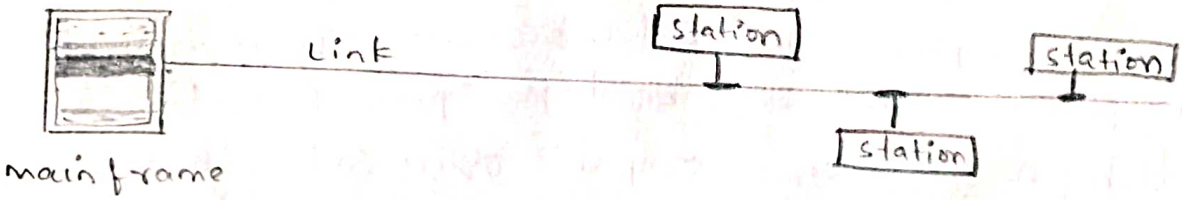
A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. for visualization purpose, it is simplest to imagine any link as a line drawn between two points. for communication to occur, two devices must be connected in some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

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

multi point :- A multipoint connection is one in which more than two specific devices share a single link. in a multipoint environment, the capacity of the channel is shared, either spatially or temporally. if several devices can use the link simultaneously. it is a spatially shared connection. if users must take turns, it is a time shared connection.



a) point-to-point



b) multipoint.

Network models:-

computer networks are created by different entities. standards are needed so that these heterogeneous networks can communicate with one another. The two best-known standards are the OSI model and the Internet model on chapter 2. we discuss these two models. The OSI (open system inter connection) model defines a seven-layer network; the Internet model defines a five-layer network.

Inter connection of networks:-

It is very rare to see a LAN, a MAN, or a LAN in isolation; they are connected to one another. when two or more networks are connected, they become an internetwork, or internet

As an example, assume that an organization has two offices, one on the east coast and the other on the west coast. The established office on the west coast has a bus topology LAN: the newly opened office on the east coast has a star topology LAN. The president of the company lives some where in the middle and needs to have control over the company from her home. to create a backbone WAN for connecting these three entities, a switched WAN has been leased. To connect the LANs to this switched WAN, however, three point-to-point WANs are required. These point-to-point WANs can be high speed DSL line offered by a telephone company or a cable modem line offered by a cable TV provider. a

Categories of networks:-

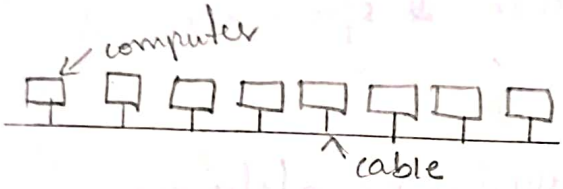
We are generally referring to two primary categories: local area networks & wide area networks. The category into which a network falls is determined by its size. A LAN normally covers an area less than 2 mi; a WAN can be world wide. Networks of a size in between are normally referred to as metropolitan area networks & span tens of miles

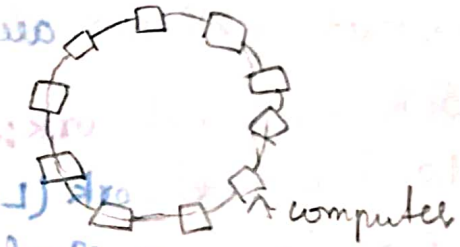
Local Area network:-

A Local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus. Depending on the needs of an organization & the type of technology used, a LAN can be simple as two PCs & a printer in some one's home office, or it can extend throughout a company & include audio and video peripherals, currently LAN size is limited to a few kilometers.

LANs are designed to allow resources to be shared between personal computers or work stations. the resources to be shared can include hardware, software, & data. A common example of LAN, found in many business environments links a work group of task-related computers. for example, engineering work stations or accounting PCs. one of the computers may be given a large capacity disk drive and may become a server to clips. soft ware can be stored on this central server

and used as needed by the whole group. In this example, the size of the LAN may be determined by licensing restrictions on the number of users per copy of software, or by restrictions on the number of users licensed to access the operating system.

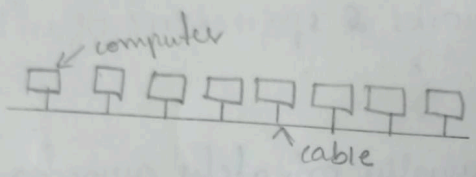


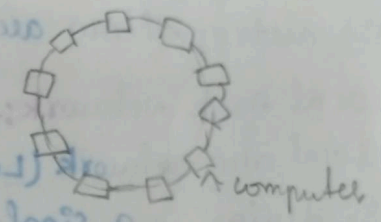


Metropolitan Area Network:-

A metropolitan area network (MAN) is a network with a size between a LAN & a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high speed DSL line to the customers. Another example is the cable TV network that originally was designed for cable TV, but today can also be used for high-speed data connection to the Internet. We discuss DSL lines and cable TV networks.

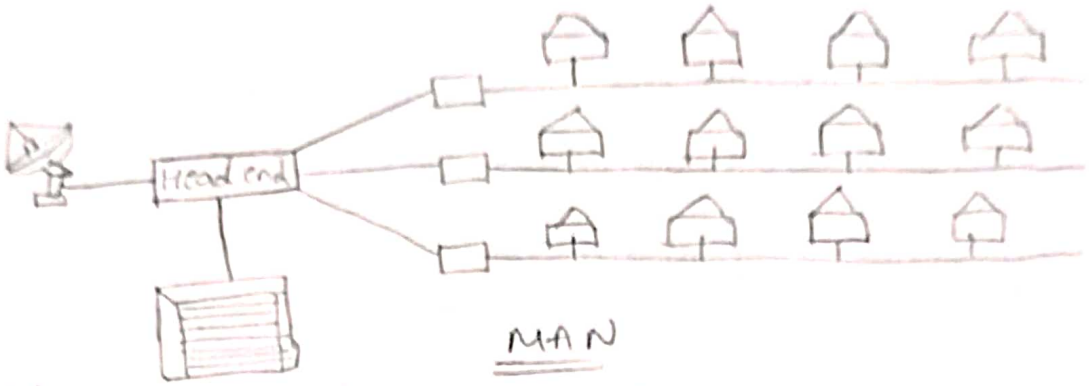
and used as needed by the whole group. In this example the size of the LAN may be determined by licensing restrictions on the number of users per copy of software, or by restrictions on the number of users licensed to access the operating system.





Metropolitan Area Network:-

A metropolitan area network (MAN) is a network with a size between a LAN & a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high speed DSL line to the customer. Another example is the cable TV network that originally was designed for cable TV, but today can also be used for high-speed data connection to the Internet. We discuss DSL lines and cable TV networks.

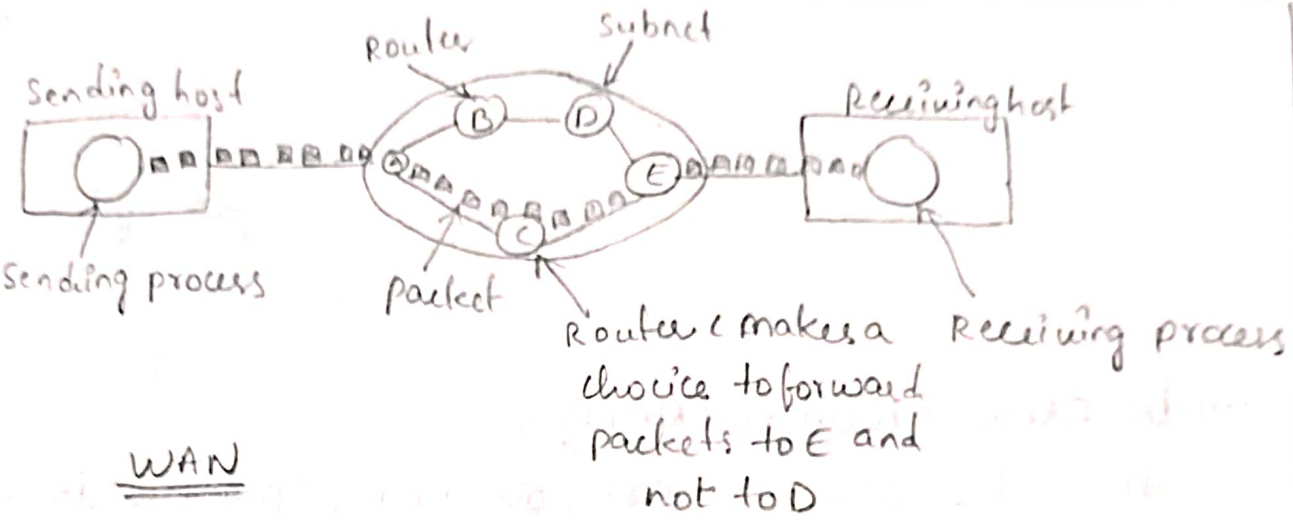


MAN

wide Area Network (WAN) :-

A wide area network, or WAN, spans a large geographical area, often a country or continent. It contains a collection of machines intended for running user programs. these machines are called as hosts. the hosts are connected by a communication subnet, or just subnet for short. the hosts are owned by the customers whereas the communication subnet is typically owned and operated by a telephone company or internet service provider. the job of the student subnet is to carry messages from host to host, just as the telephone system carries words from speaker to listener

In this figure, all the packets follow the route ACE, rather than ABDE or ACDE. In some networks all packets from a given message must follow the same route; in others each packed is routed separately. Of course, if ACE is the best route, all packets may be sent along it, even if each packets is individually routed.



WAN

The principle of a packet-switched WAN is so important. Generally, when a process on some host has a message to be sent to a process on some other host, the sending host first cuts the message into packets, each one bearing its number in the sequence. These packets are then injected into the network one at a time in quick succession. The packets are transported individually over the network and deposited at the receiving host, where they are reassembled into the original message and delivered to the receiving process. A stream of packets resulting from some initial message is illustrated

THE INTERNET

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

A Brief History

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letters) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letters), a collabration of more than hundreds of thousands of interconnected networks Private individuals as well as various organizations such

as government agencies, schools, research facilities, corporations, and libraries in more than 100 countries use the Internet Millions of people are users. yet this extraordinary communication system only came into being in 1969.

In the mid-1960's mainframe computers in research organizations were standalone devices. Computers from different manufactures were unable to communicate with one another The Advanced Research Projects Agency (ARPA) in the Department of Defense (DoD) was intrested in finding away to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

In 1967, at an Association for computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an inteiface message processor (IMP). The IMPs in tum, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host. By 1969, ARPANET was a reality. Four nodes, at the university of california at Los Angeles (UCLA)

the University of California at Santa Barbara (UCSB), Stanford Research institute (SRI) and the University of Utah, were connected via the IMPs to form a network. Software called the Network control protocol (NCP) provided communication between the hosts.

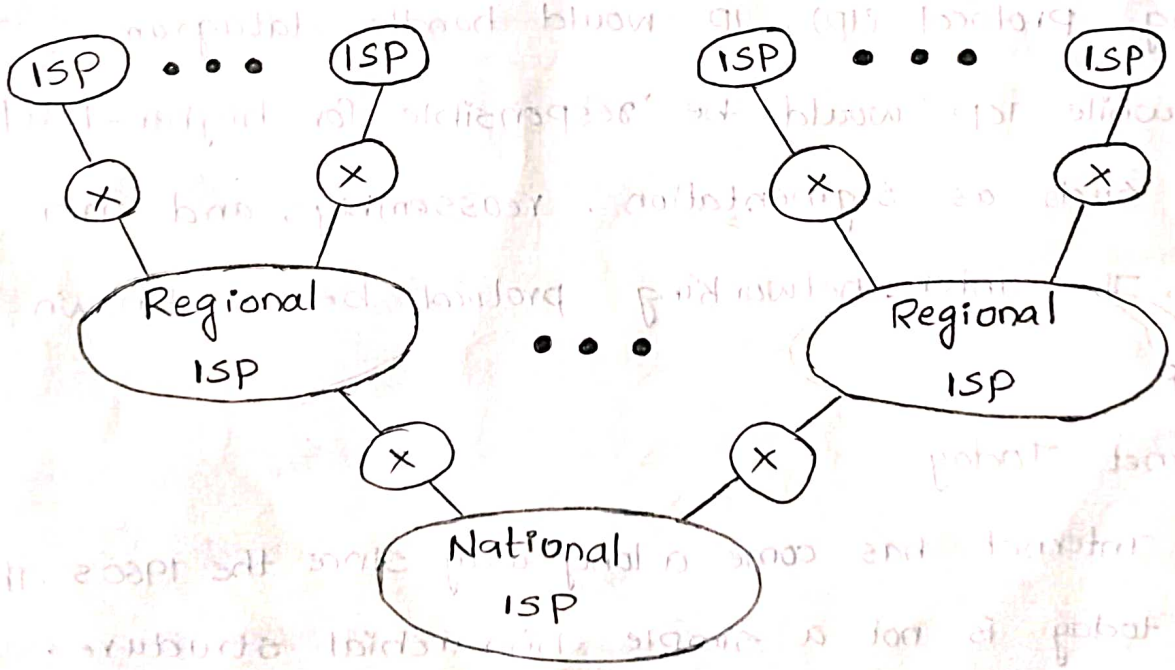
In 1972 Vint cerf and Bob kahn, both of whom were part of the core ARPANET group, callaborated on what they called the Internetting projec.1 cerf and kahn's landmark 1973 paper outlined the protocols to achiere end-to-end delivery of packets. This paper on Transmission control Protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. shortly there after, authorities made a decision to split TCP into two Protocols. Transmission control protocol (TCP) and Inter networking Protocol (IP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The inter networking protocol become known as TCP/IP

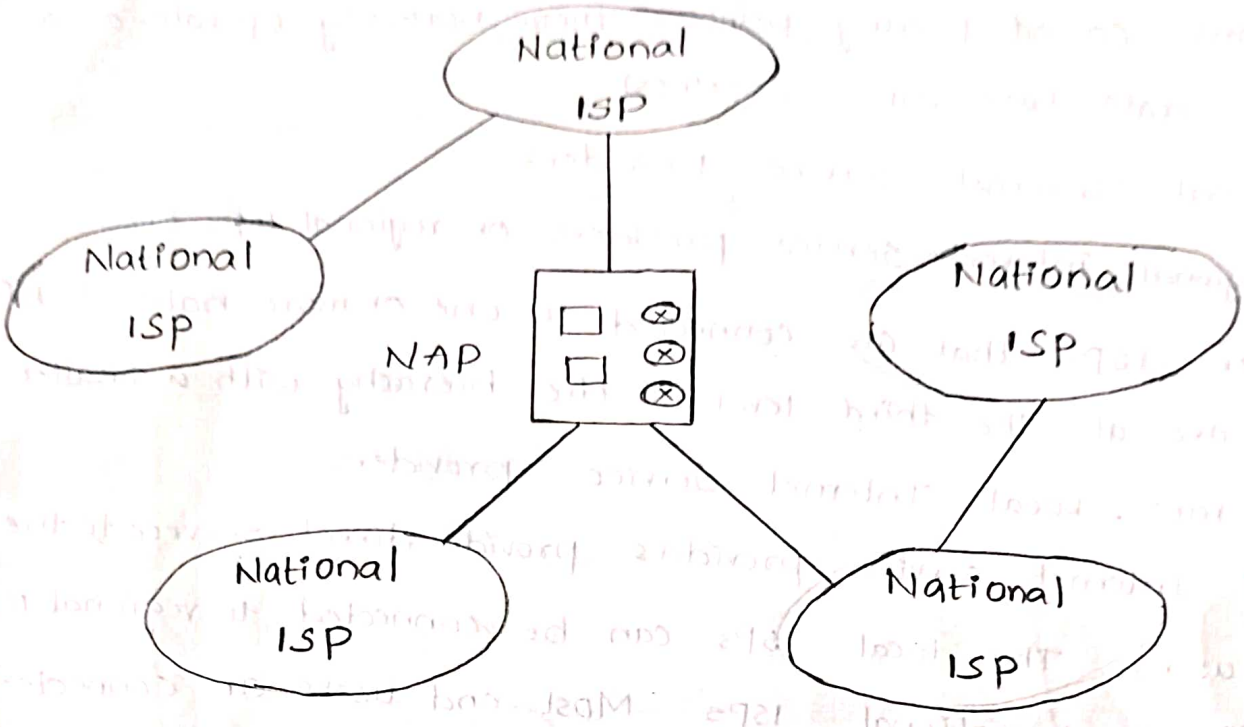
The Internet Today

The Internet has come a long way since the 1960s. The Internet today is not a simple hierarchial structure.

It is made up of many wide-and local-area networks joined by connecting devices and switching stations. It is difficult to give an accurate representation of the Internet because it is continually changing - new networks are being added, existing networks and adding addresses, and networks of defunct companies are being removed. Today most end users who want Internet connection use the services of Internet service provider (ISPs). There are international service providers, national service providers, regional service providers, and local service providers. The Internet today is run by private companies, not the government. Figure shows a conceptual (not geographic) view of the Internet.

a. Structure of a national ISP





b. Interconnection of national ISPs

International Internet service providers:

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

National Internet service providers;

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

stations called peering points. These normally operate at a high data rate (up to 600Mbps)

Regional Internet service providers:

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

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

PROTOCOLS AND STANDARDS

protocols:

In computer networks, communication occurs between entities in different systems. An entity is anything capable of sending or receving information. However. two entities cannot simply send bit stream to each other and expect

to be understood. For communication to occur, the entities must agree on a protocol. A protocol is a set of rules that govern data communications. A protocol defines what is communicated, how it is communicated and when it is communicated.

The key elements of a protocol are syntax, semantics and timing

→ Syntax. The term syntax refers to the structure or format of the data, meaning the order in which they are presented. For example, a simple protocol might expect the first 8 bits of data to be the address of the receiver, and the rest of the stream to be the message itself

→ Semantics: The word semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation? For example, does an address identify the route to be taken or the final destination of the message?

→ Timing. The term timing refers to two characterstics: when data should be sent and how fast they can be sent. For example, if a sender produces data at 100Mbps

but the receiver can process data at only 1Mbps, the transmission will overload the receiver and some data will be lost

Standards:

Standards are essential in creating and maintaining an open and competitive market for equipment manufacturers and in guaranteeing national and International interoperability of data and telecommunic-ations technology and processes. Standards provide guideliness to manufacturers, vendors, government agencies and other service providers to ensure the kind of interconnectivity necessary in today's marketplace and in international communications.

Data communication standards fall into two categories: de facto (meaning "by fact" or "by convention") and de jure (meaning "by law" or "by regulation")

→ De facto: Standards that have not been approved by an organized body but have been adopted as standards through widespread use are de facto standards through are often established orginally by manufacturers who seek to define the functionality of a new product or technology

→ Dejure: Those standards that have been legislated by an officially recognized body are de jure standards

LAYERED TASKS:

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

Sender, Receiver and carrier

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

At the Sender site.

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

→ Higher layer. The Sender writes the letter, inserts the letter in an envelope, writes the Sender and receiver addresses, and drops the letter in a mailbox.

→ Middle layer. The letter is picked up by a letter carrier and delivered to the post office.

→ Lower layer. The letter is sorted at the post office; a carrier transports the letter

On the way: The letter is then on its way to the recipient

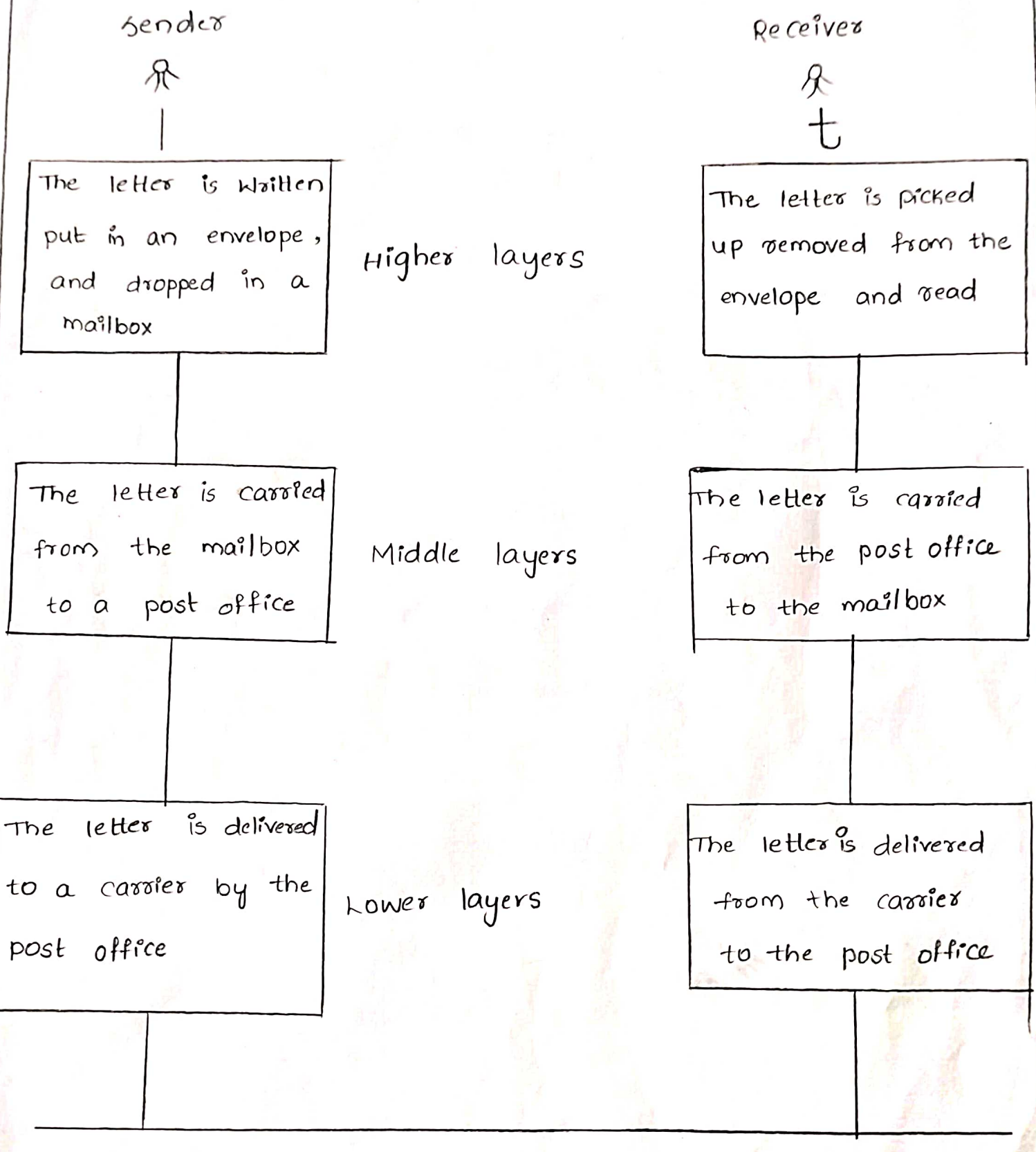
on the way to the recipient's local post office, the letter may actually go through a central office. In addition it may be transported by truck, train, airplane, boat; or a combination of these.

At the Receiver Site.

→ Lower layer. The carrier transports the letter to the post office

→ Middle layer: The letter is sorted and delivered to the recipient's mailbox

→ Higher layer: The receiver pics up the letters, opens the envelope and reads it.

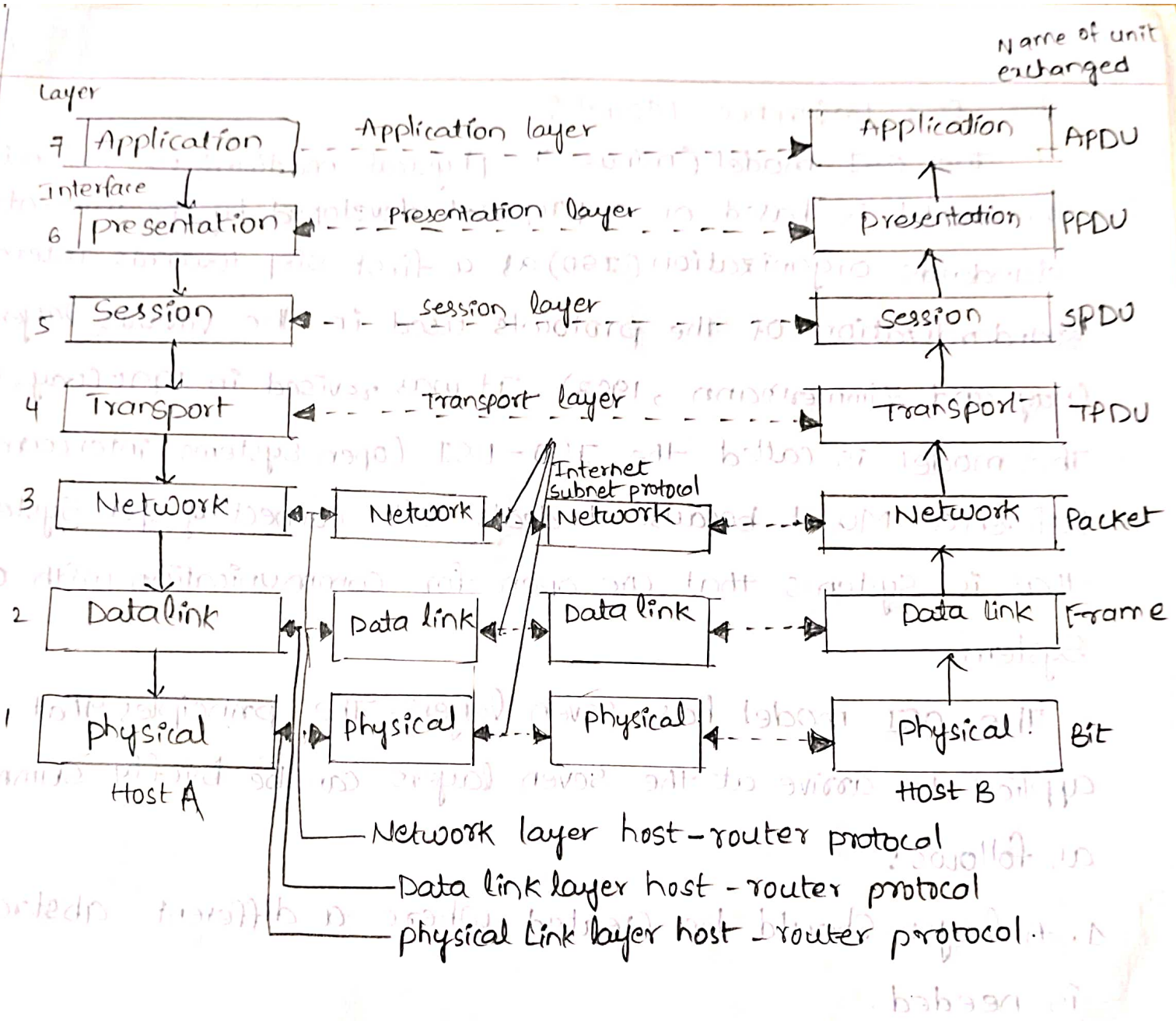

Sender
Receiver
The letter is written put in an envelope, and dropped in a mailbox
Higher layers
The letter is picked up removed from the envelope and read
The letter is carried from the mailbox to a post office
Middle layers
The letter is carried from the post office to the mailbox
The letter is delivered to a carrier by the post office
Lower layers
The letter is delivered from the carrier to the post office
The parcel is carried from the source to the destination

The OSI Reference Model :-

The OSI model (minus the physical medium) is shown in fig. This model is based on a proposal developed by the international standards organization (ISO) as a first step towards international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995). The model is called the ISO-OSI (open systems interconnection) Reference Model because it deals with connecting open systems that is, systems that are open for communication with other system.

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

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


Name of unit exchanged
Layer
7 Application
Application layer
Application
APDU
Interface
6 presentation
presentation layer
Presentation
PPDU
5 Session
session layer
Session
SPDU
4 Transport
Transport layer
Transport
TPDU
3 Network
Network
Internet subnet protocol
Network
Network
Packet
2 Datalink
Data link
Data link
Data link
Frame
1 physical
physical
physical
physical
Bit
Host A
Host B
Network layer host-router protocol
Data link layer host-router protocol
physical Link layer host-router protocol.

The physical layer:

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

The Data Link Layer:

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

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

The Network layer:

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

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

when a packet has to travel from one network to another to get to its destination, many problems can arise The addressing used by the second network may be different from the first one. The second one may not accept the packet at all because it is too large.

The transport layer :-

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

The transport layer is a true end-to-end layer, all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers the protocols are between each machine and its immediate neighbours, and not between the ultimate source and destination machines, which may be separated by many routers.

The session layer:- the session layer allows users on different machines to establish sessions between them. Sessions offer various services, including dialog control (keeping track of whose

turn it is to transmit), token management (preventing two parties for attempting the same critical operating at the same time), and synchronization (check pointing long transmissions to allow them to continue from where they were after a crash).

The presentation layer:-

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

The Application layer:-

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

The TCP/IP Reference model :-

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

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

unlike OSI reference model, TCP/IP reference model has only 4 layers. they are.

1. Host - to - Network layer.
2. Internet layer
3. Transport layer
4. Application layer.

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

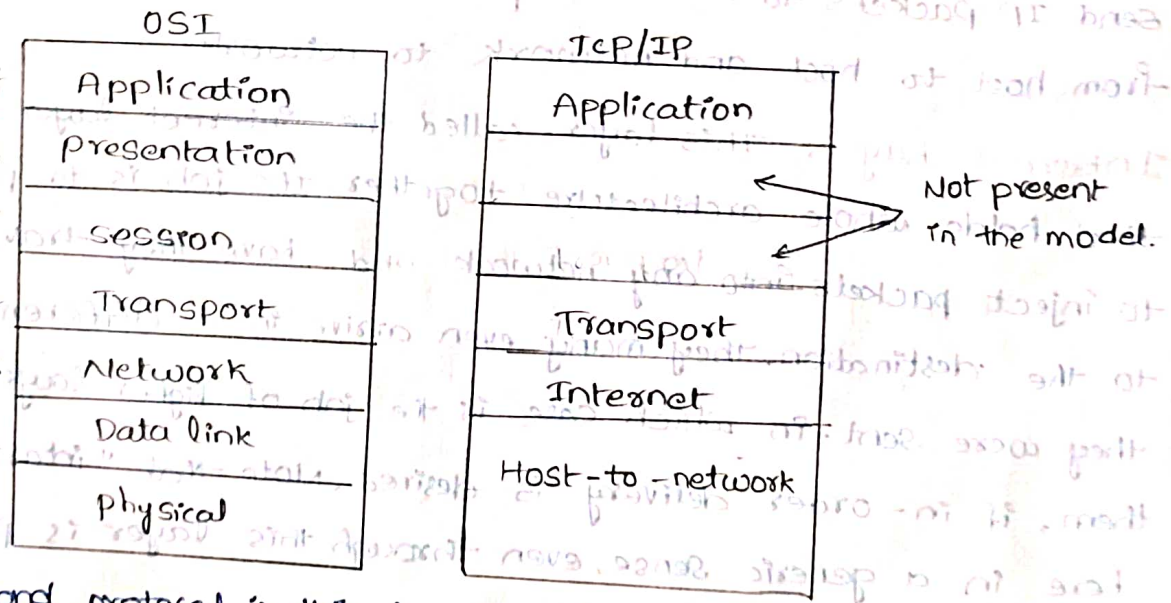
Internet Layer :- This layer, called the internet layer, is the linchpin that holds whole architecture together. Its job is to permit hosts to inject packets into any network and have they travel independently to the destination. they many even arrive in a different order than they were sent, in which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

The internet layer defines an official packet format and protocal called IP (Internet protocal). The job of the internet layer is

to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is similar in functionality to the OSI network layer.

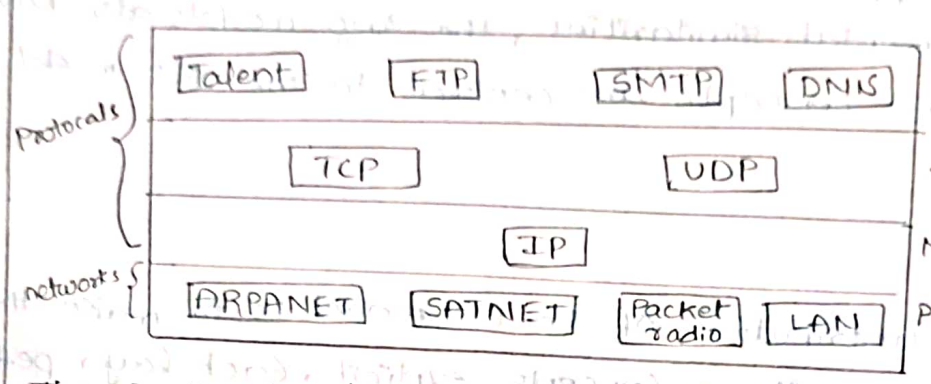
The Transport Layer :-

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



The second protocol in this layer, UDP (user datagram protocol), is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their

own. It is also widely used for one-shot, client-server-type request-reply queries and applications in which prompt delivery is more important than accurate delivery. such as transmitting speech or video. The relation of IP, TCP, and UDP is shown fig. Since the model was developed, IP has been implemented on many other networks



The Application layer:-

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

Comparison of the OSI and TCP/IP Reference models:-

The OSI and TCP/IP reference models have much in common Both are based on the concept of a stack independt protocols. Also the functionality of the layers is roughly similar. For example,

In both models the layers up through and including the transport layer are there to provide end-to-end, network-independent transport service to process wishing to communicate. These layers form the transport provider. Again in both models, the layers above transport are application oriented users of the transport service. Despite these fundamental similarities, the two models also have many differences three concepts are central to the OSI model.

1. Service
2. Interfaces
3. Protocols.

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

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

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

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

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

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

Introduction to Addressing.

Addressing hosts is the process of assigning an address. Hosts are uniquely identifies through the unique address. Addressing of 2 types:

1. Hardware Addressing
2. Logical Addressing.

Hardware addressing is used to identify the host in the local and it identifies and logical addressing host on the network level. These two types of addressing are being explained in detail below

Hardware Addressing:- Hosts are uniquely identified in the local area network with the help of a hardware address. If you

look at the OSI model, then the hardware addressing is done at layer 2 (Data link layer) + hardware address is also called MAC (media access control) address. MAC address is hard-coded on the network interface cards (NIC).

Example :- 06:5f:39:ac:dd:2c.

The first 6 digits of the MAC address are identified by the manufacturer of the NIC (network interface card). These early 6 bits are also called the OUI (organizational unique identifier). The remaining 6 digits host is used to uniquely identify the network. These last 6 digits are called the host id.

There is a decrease in the MAC address, so that you can not identify the network. If you want to identify which host is in the network then you will see its IP address. IP addresses are explained in logical addressing.

Logical Addressing :- According to the OSI model, logical addressing is used in the network layer. Through logical address, you uniquely identify a host in the entire network. Logical address also separate a network from another network.

The IP address size is 32 bit. An IP address is divided into 2 parts the first part is the network ID from which it is identified that what is the host's network. the second part is the host ID from which the host is uniquely identified.

Internet protocal (IP) is responsible for logical addressing. Internet protocal performs 2 tasks. First logical addressing and second routing.

Internet protocal provides 2 types of addressing. An IPV4 addressing and the second IPV6 addressing are done.

Wireless links :- Now days there are people who need to be online all the time. For these mobile users, twisted pair and fiber optics are of no use. They need to get data for their laptop or wristwatch computers without being tethered to the terrestrial communication infrastructure. For these users, wireless communication is the answer. Computer networks can take advantage of the wireless infrastructure where physical wires can not be laid out. Some people believe that the future holds only two kinds of communication : fiber and wireless. All non mobile computers, telephones, printers, and so on will use fiber, and all mobile ones will use wireless.

One of the key challengers in wireless networking the efficient utilization of the available transmission spectrum. Because the frequency spectrum available for wireless communication is limited, frequencies must be reused within the same geographic area. The open-air interface makes it difficult to prevent snooping

The link level design techniques involve making trade-offs among the various parameters relavant to the link layer. The optimum design would involve the use of minimum bandwidth and transmit power while maintaining a high data rate, low latency, and low bit error rates (BER). These design challenges must be achieved in the presence of channel imperfections, such as interface.

Wireless links use devices as an antenna for transmitting signals through vacuum, space, air, or substances. Electromagnetic waves can be propagated through the first three, as well as through water and wood.

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

The properties of radio waves are frequency dependent. At low frequencies, radio waves pass through obstacles well, but the power falls off sharply with distance from the source, roughly as $1/r^2$ in air. At all frequencies, radio waves are subject to interference from motors electrical equipment.

Due to radio's ability to travel long distances, interference between users is a problem.

Network characteristics :-

A list of computer network features is given below.

- Communication speed
- File sharing
- Back up and Roll back is easy.
- Software and Hardware sharing
- Security
- Scalability
- Reliability.

IEEE 802.11 (wireless LAN)

- IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications for implementing wireless local area network (WLAN) computer communication in the

900MHz and 2.4, 3.6, 5 and 60 GHz frequency bands.

- IEEE developed an International standard for WLANs. The 802.11 Standard focuses on the bottom two layers of the OSI model, the physical layer (PHY) and data link layer (DLL)
- The objective of the IEEE 802.11 Standard was to define a medium access control (MAC) sublayer, MAC management protocols and services, and three PHYs for wireless connectivity of fixed, portable, and moving devices within a local area.

802.11 Features:-

The architecture of the IEEE 802.11 WLAN is designed to support a network where most decision making is distributed to mobile stations.

- CSMA/CA based MAC protocol
- DCF (Distributed Coordination Function)
- Support for both time-critical.
- PCF (point co-ordination Function and non-critical traffic (DCF)
- Support multiple priority levels.
- Spread Spectrum technology power managment allows a node to doze off.

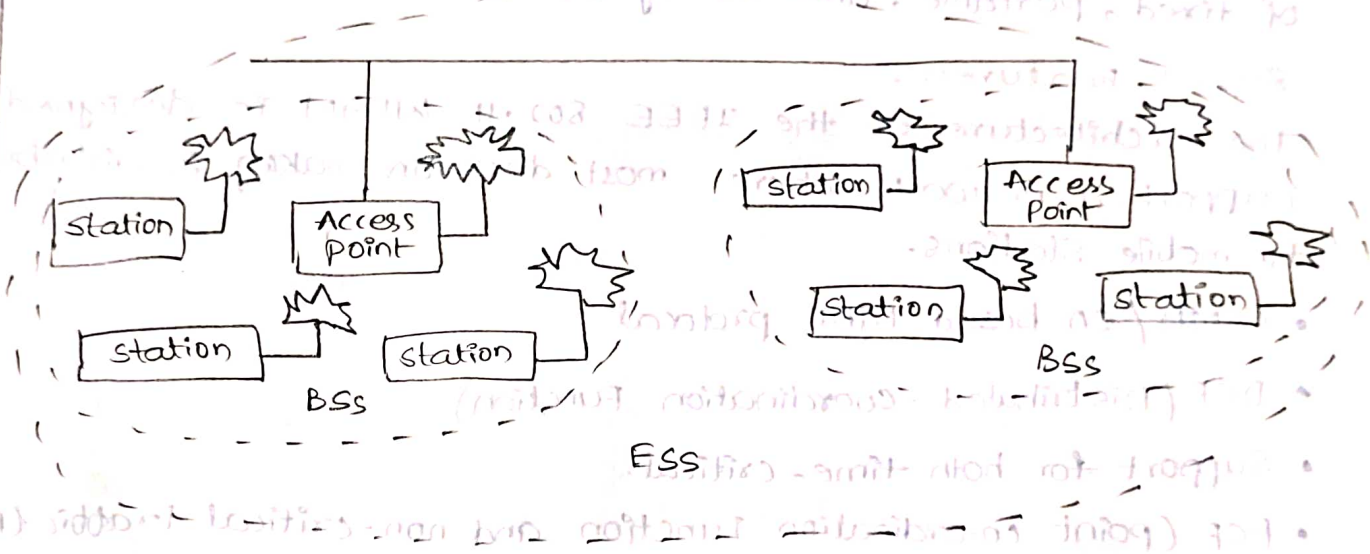
IEEE 802.11 Architecture:

The architecture of the IEEE 802.11 WLAN is designed to support a network where most decision making is distributed to mobile stations. Two network architectures are defined in the IEEE 802.11 standard:

Infrastructure network:- An infrastructure network is the network architecture for providing communication between wireless clients and wired network resources. The transition of data from the

wireless to wired medium occurs via an AP. An AP and its associated wireless clients define the coverage area.

→ Point-to-Point (ad hoc) network: An ad hoc network is the architecture that is used to support mutual communication between wireless clients. Typically, an ad hoc network is created spontaneously and does not support access to wired networks.



STA : station
AP : Access point
BSS : Basic service point
ESS : Extended service set.

802.11 physical layer (PHY):

- At the physical layer, IEEE 802.11 defines three physical characteristics for WLAN's
- Diffused infrared (base band), DSSS, and FHSS. All three support a 1 to 2 Mbps data rate. Both DSSS and FHSS use the 2.4 GHz ISM band (2.4-2.4835 GHz).
- These include: (1) frame exchange between the MAC and PHY

under the control of the physical layer convergence procedure (PLCP) sublayer ; (2) a

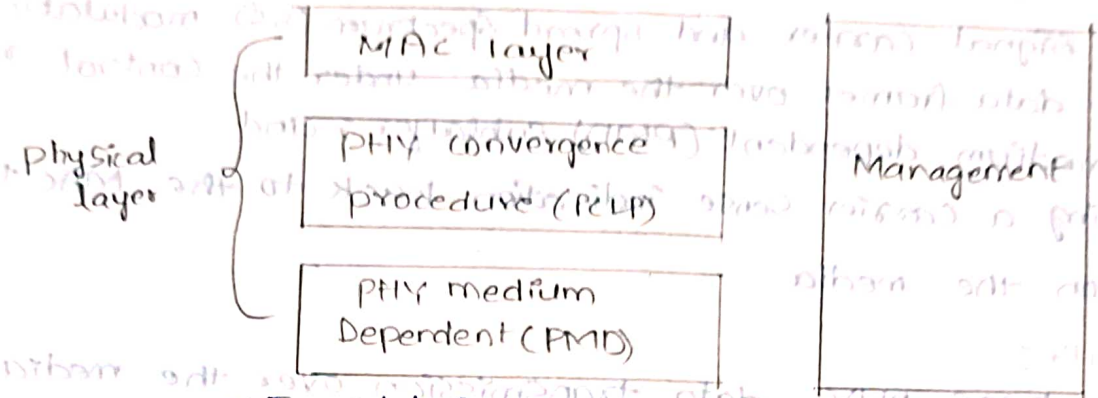
(2) use of signal carrier and spread spectrum (SS) modulation to transmit data frames over the media under the control of the physical medium dependent (PMD) sublayer ; and

(3) providing a carrier sense indication back to the MAC to verify activity on the media.

DSSS PHY :-

- In the DSSS PHY, data transmission over the media is controlled by the PMD sublayer as directed by the PLCP sublay -er. The PMD sublayer takes the binary information bits from the PLCP protocol data unit (PPDU) and converts them into RF signals by using modulation and DSSS techniques.
- In the PPDU frame, which consists of a PLCP preamble, PLCP header, and MAC protocol data unit (MPDU). The PLCP preamble and PLCP header are always transmitted at 1 Mbps, and the MPDU can be sent at 1 or 2 Mbps.
- The start of frame delimiter (SFD) contains information that marks the start of the PPDU frame. The SFD specified is common for all IEEE 802.11 DSSS radios.
- The signal field indicates which modulation scheme should be used to received the incoming MPDU. The binary value in this field is equal to the data rate multiplied by 100kbps
- The service field is reserved for future use. The length field indicates the number of microseconds necessary to transmit the MPDU. The MAC layer uses this field to determine the end of a PPDU frame

- The CRC field contains the results of calculated frame check sequence from the sending station.



OSI model for IEEE 802.11 WLAN.

FHSS PHY :-

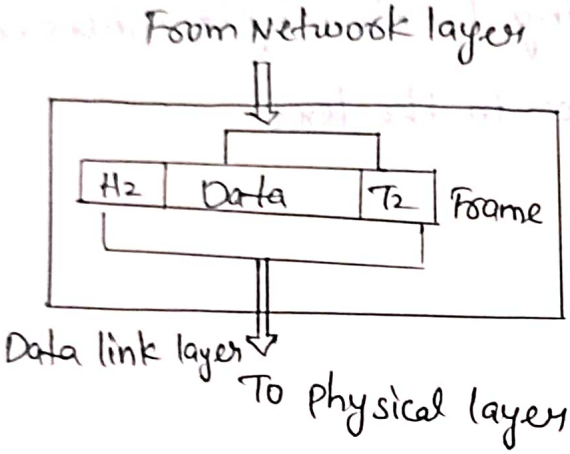
- In FHSS PHY, data transmission over media is controlled by the FHSS PMD sublayer as directed by the FHSS PLCP sublayer. as The FHSS PMD takes the binary information bits from the whitened PSDU and converts them into RF signals by using carrier modulation and FHSS techniques.
- the format of ~~got~~ the PPDU is shown in fig. It is consists of the PLCP preamble, PLCP header, and PLCP service data unit (PSDU). the PLCP preamble is used to acquire the incoming signal and synchronize the receiver's demodulator.
- the PLCP header contains information about PSDU from the sending physical layer. the PLCP preamble and header are transmitte -d at 1Mbps.

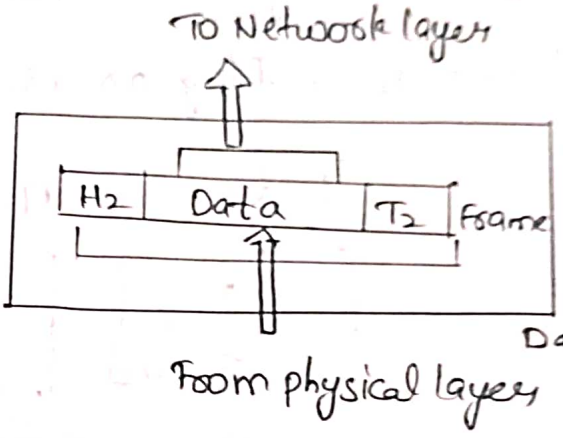
Unit-II

DATA LINK LAYER

Introduction to data link layer :-

Data link layer is second layer of OSI layered model. The main task of the data link layer is to transform a raw bits transmission facility into a line that appears free of undetected transmission errors to the network layer.





Data link layer has two sub-layers :-

→ logical link control :- It deals with protocols, flow control a error control.

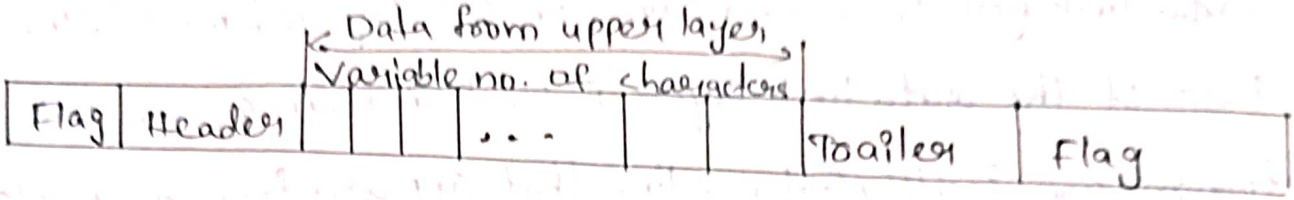
→ Media Access control :- It deal with actual control of medi

Functions of Data-link layer :-

Framing :- Data-link layer takes packets from network layer a encapsulates them into frames. Then, it sends each frame bit bit on the hardware. At receiver end, data link layer picks signals from hardware and assembles them into frames. They

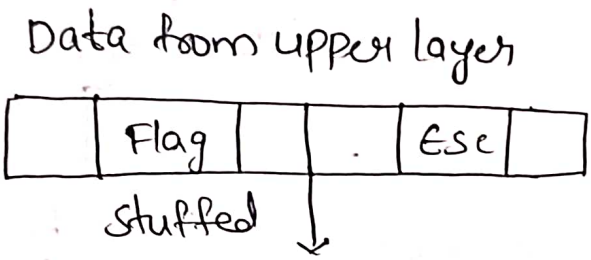
two types.

1. Fixed length frame
2. Variable length

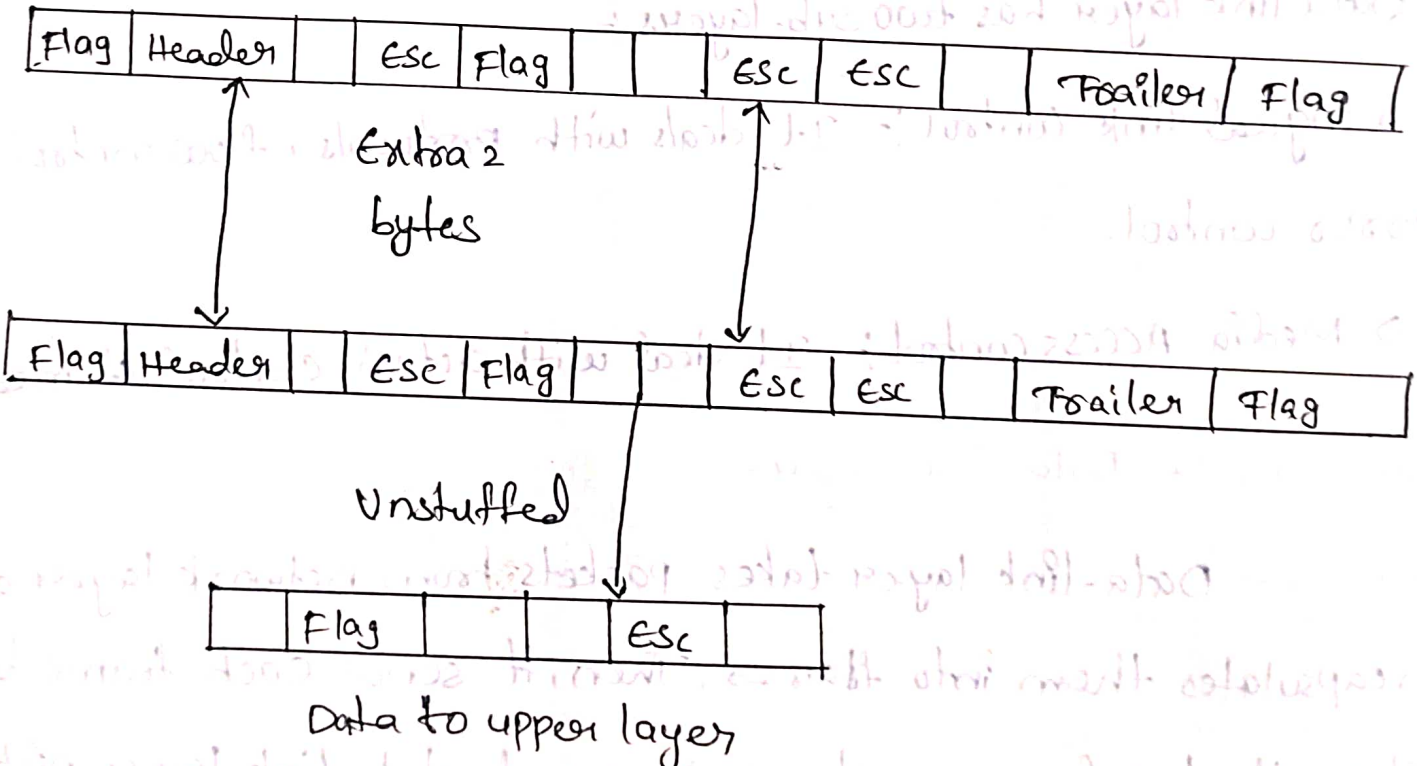


Byte stuffing and unstuffing:-

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



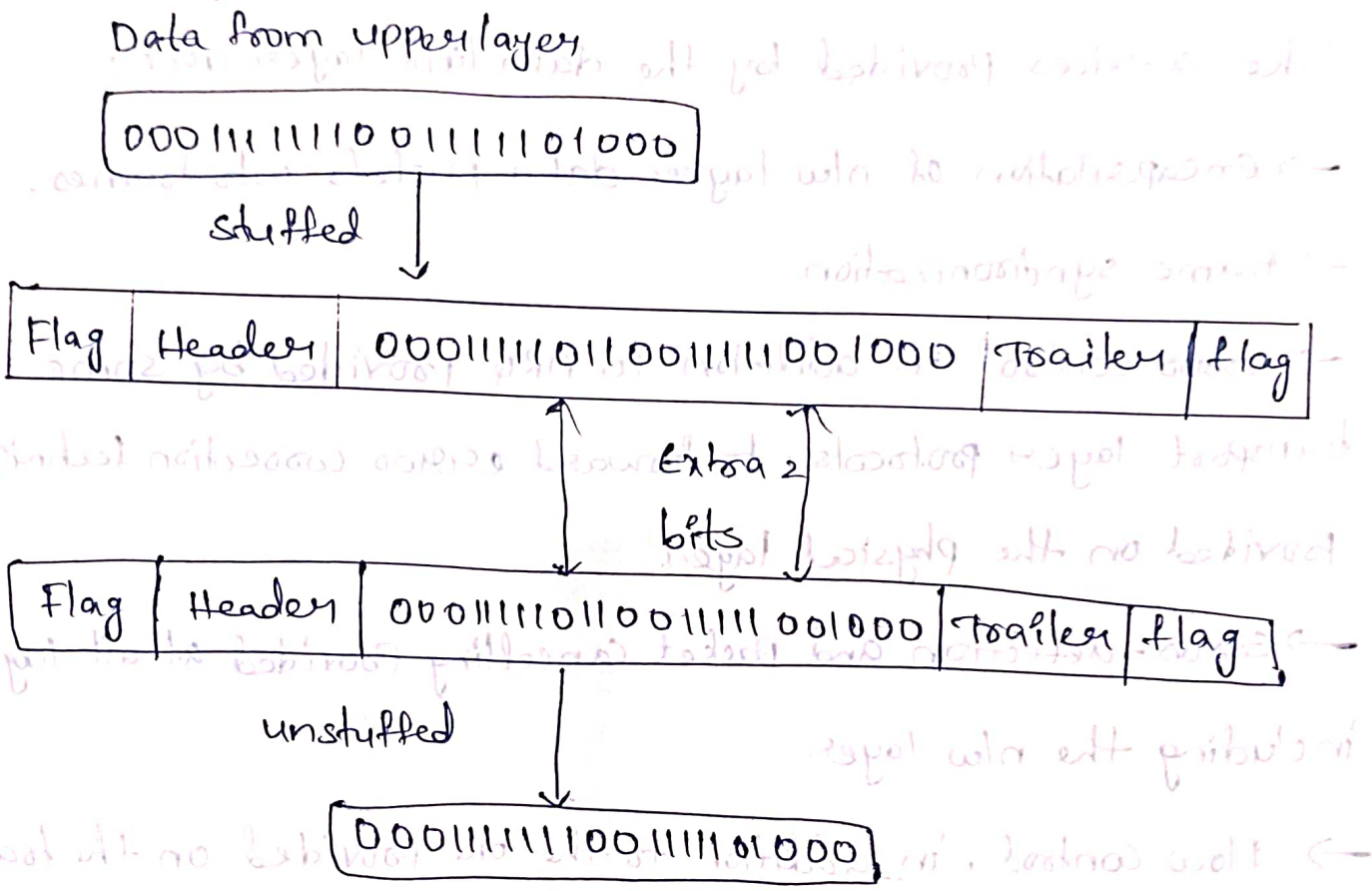
Frame sent



Bit stuffing and unstuffing :

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

The pattern 0111110 for a flag



Error control:- Sometimes signals may have encountered Problem in transition and the bits are flipped. These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

Flow control:- stations on same link may have different speed or capacity.

Data-link layer ensures flow control that enables both machines to exchanges data on same speed.

→ Protocols of Data Link Layer are Ethernet, PPP (point to point protocol)

The services provided by the link layer:-

The services provided by the data link layer are:

→ Encapsulation of n/w layer data packets into frames.

→ Frame synchronization.

→ Error control in addition to ARQ provided by some transport-layer protocols, to forward error correction techniques provided on the physical layer.

→ Error-detection and packet cancelling provided at all layers including the n/w layer.

→ Flow control, in addition to the one provided on the transport layer. Data link layer flow control is not used in LAN protocols such as Ethernet: but in modems and wireless networks.

Links of data link layer:-

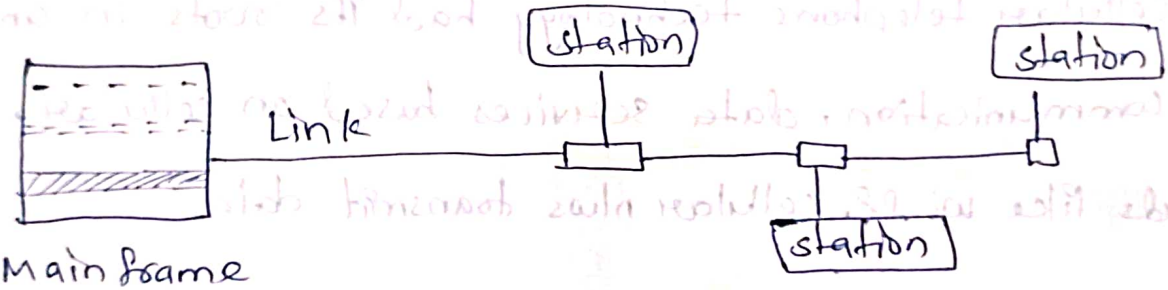
There are mainly 2 types

1. Broad cast link
2. Point-to-point link

1. Broad cast link :- single communication channel shared data by all machines. A single communication channel is used for broadcasting (or) multicasting. Each packet consist of a special code in address field is called broad casting. Eg:- News channels.

2. Point-to point link :- A Point-to-point nlw is one of the simplest nlw because it involves only 2 nodes.

a. Point-to-Point



b) Multipoint.

Access Networks :-

→ In addition to the ethernet and wi-fi connections we typically use to connect to the Internet at home, at work, at school and in many public places, most of us connect to the Internet over an access or broadband service that we buy from an ISP.

Access nlws describes two such technologies:

1. Passive optical nlw
2. cellular nlw that connect our mobile devices.

Passive optical nlw :-

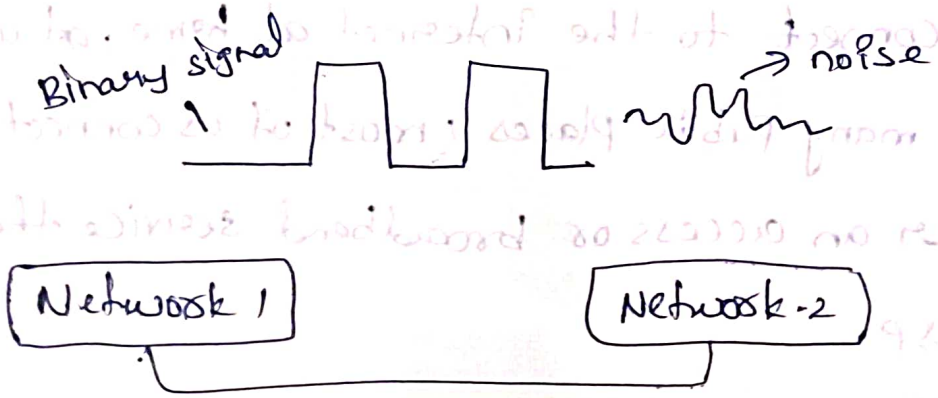
PON is the technology most commonly used to deliver fiber-based broadband to homes and business. PON adopter a Point-to-multipoint design, which means the nlw is structured as a tree.

2. Cellular Nlw :-

While cellular telephone technology had its roots in analog voice communication, data services based on cellular standards like wi-fi. cellular nlws transmit data.

Error and types of errors :-

Errors :- The data can be corrupted during transmission. It may be affected by external noise or some other physical imperfections. In this case.



Access Networks

An **access network** is a type of network whic
the immediate router (also known as the "edg
system to any other distant end system. Exan
networks, enterprise networks, ADSL, mobile network, FTTH etc.

Types of access networks:

- **Ethernet** – It is the most commonly installed wired LAN technology and it provides services on the Physical and Data Link Layer of OSI reference model. Ethernet LAN typically uses coaxial cable or twisted pair wires.
- **DSL** – DSL stands for Digital Subscriber Line and DSL brings a connection into your home through telephone lines and a DSL line can carry both data and voice signals and the data part of the line is continuously connected. In DSL you are able to use the Internet and make phone calls simultaneously. **DSL** modem uses the telephone lines to exchange data with digital subscriber line access multiplexer (DSLAMs). In DSL we get **24 Mbps** downstream and **2.5 Mbps** upstream.
- **FTTH** – Fiber to the home (FTTH) uses optical fiber from a central Office (CO) directly to individual buildings and it provides high-speed Internet access among all access networks.It ensures high initial investment but lesser future investment and it is the most expensive and most future-proof option amongst all these access networks.
- **Wireless LANs** – It links two or more devices using wireless communication within a range. It uses high-frequency radio waves and often include an access point for connecting to the Internet.
- **3G and LTE** – It uses cellular telephony to send or receive packets through a nearby base station operated by the cellular network provider. The term "3G internet" refers to the third generation of mobile phone standards as set by the International Telecommunications Union (ITU). Long Term Evolution (LTE) offers high-speed wireless communication for mobile devices and increased network capacity.
- **Hybrid Fiber Coaxial (HFC)** – HFC is a combination of fiber optic and coaxial cable that is widely used by cable television operators to provide high-speed internet access. The fiber optic cable is used to connect the headend to the neighborhood

communication to deliver internet access to remote and rural areas. It has a higher latency and lower bandwidth compared to other access networks, but it can provide internet access in areas where other options are not available.

- **Power Line Communication (PLC)** – PLC uses the existing electrical wiring in a building to transmit data signals. It is a low-cost alternative to traditional wired networks and can be used to provide internet access in buildings where it is difficult to install new cables.
- **WiMAX** – WiMAX (Worldwide Interoperability for Microwave Access) is a wireless access network technology that provides high-speed internet access over a wide area. It is commonly used in rural and suburban areas where it is difficult or expensive to deploy wired networks.
- **5G** – 5G is the latest wireless communication technology that offers high-speed internet access and increased network capacity. It is designed to support a wide range of applications, including virtual and augmented reality, autonomous vehicles, and smart cities. 5G networks are being rolled out globally, and they are expected to transform the way we connect to the internet.
- **Wi-Fi:** Wi-Fi is a wireless access network technology that allows devices to connect to a local area network (LAN) or the Internet using radio waves. It is commonly used in homes, offices, public places, and other areas where people need wireless internet access.
- **Bluetooth:** Bluetooth is a short-range wireless communication technology that is used to connect devices within a limited range, typically up to 30 feet. It is commonly used to connect mobile phones, laptops, and other devices to speakers, headphones, and other accessories.
- **Wi-Fi Direct:** Wi-Fi Direct is a technology that allows devices to connect to each other without the need for a wireless access point or network. It is commonly used to transfer files and other data between devices in close proximity.
- **Near Field Communication (NFC):** NFC is a wireless communication technology that allows devices to exchange data when they are held close together. It is commonly used for mobile payments, access control, and other applications where security is important.
- **ZigBee:** ZigBee is a wireless communication technology that is used for low-power, low-speed applications such as home automation, industrial control, and sensor networks. It is designed to be simple, reliable, and easy to use.
- **LoRaWAN:** LoRaWAN is a wireless communication technology that is used for long-range, low-power applications such as smart cities, agriculture, and environmental monitoring. It is designed to be low-cost, low-power, and easy to

Types of errors:- There are mainly 3 different types of errors they are

1. single bit error
2. Multiple bit error
3. Burst error

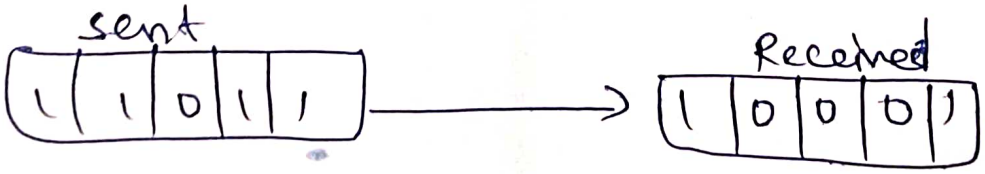
1. Single bit error:- One bit of data changes from 0 to 1 and 1 to 0 is called as single bit error.

Single bit error:- In a frame, there is only one bit, anywhere through which is corrupt.

2. Multiple bit error:-

Two or more non consective bits in a data unit have changed from 1 to 0 and 0 to 1.

Frame is received with more than one bits in corrupted sta



3. Burst error :- Two or more consecutive bits in the data unit have changed from 1 to 0 and 0 to 1

Frame contains more than 1 consecutive bits corrupted

Error detection and correction techniques:-

Types of Error detections:-

1. parity checking
2. Longitudinal Redundancy check (LRC)
3. cyclic Redundancy check (CRC)
4. check sum

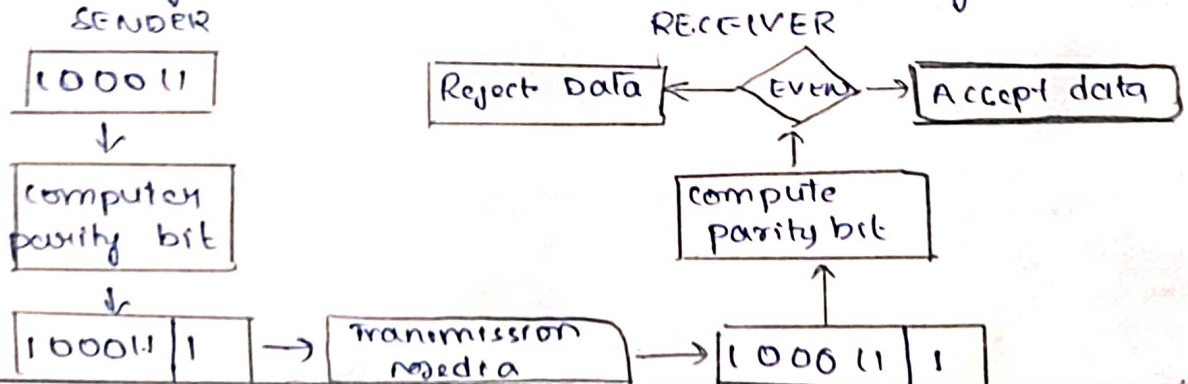
1) parity checking:

→ parity bit means nothing but an additional bit added to the data at the transmitter before transmitting the data. Before adding the parity bit, number of 1s (or) zeros is calculated in the data. Based on this calculation of data an Extra bit is added to the actual information/data. The addition of parity bit to the data will result in the change of data spring size

→ There are 2 types of parity bits in Error detection they are

1) Even parity: If the data has even number of 1's the parity bit is 0

2) odd parity: odd number of 1's the parity bit is 1



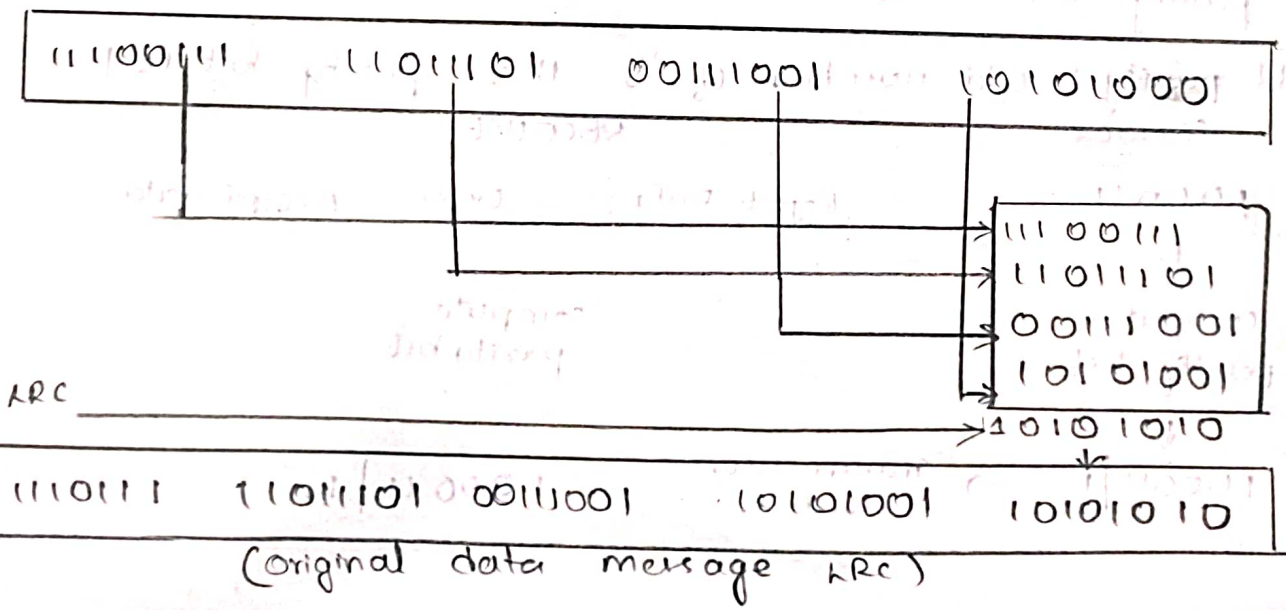
Disadvantages:- only single bit Error is detected it fails for multi bit error values

3 bit data			message with Even parity		message with odd parity	
A	B	C	message	parity	message	parity
0	0	0	000	0	000	1
0	0	1	001	1	001	0
0	1	0	010	1	010	0
0	1	1	011	0	011	1
1	0	0	100	1	100	0
1	0	1	101	0	101	1
1	1	0	110	0	110	1
1	1	1	111	1	111	0

*Longitudinal Redundancy check (LRC) / Two dimensional parity check :-

Disadvantage: if 2 bits are corrupted in 1 data Unit and another data unit is exactly at same position is corrupted will not be

In logitudinal redundancy method a Block of bits are arranged in a table format (in rows and columns) and we will calculate the parity bit for each column seperatly. The set of three parity bits are also sent along with our original data



Example :-

original message

1010000

@ means x-OR

Generator polynomial

x^3+1

$1x^3+0\cdot x^2+0x^1+1\cdot x^0$

crc generator

1001 4-bit

If crc generator is of n bit then append (n-1) zeros in the end of original message.

Sender

```
1001 | 1010000000
       1001
       ----
       0011000000
     @ 1001
       ----
       0$10100000
     @ 1001
       ----
       0$11000
     @ 1001
       ----
       01$100
     @ 1001
       ----
       00$11
     @ 1001
       ----
       0111
```

Message to be transmitted

```
10·1000000000
      0$11
-------------
1 0 1 0 0 0 0 $ 1 1
-------------
```

Receiver

```
1001 | 1 0 1 0 0 0 0 $ 1 1
       1001
       ----
       0 0 1 1 0 0 0 $ 1 1
     @ 1 0 0 1 0 0 1 1
       ----------
       0 1 0 1 0 $ 1 1
     @ 1 0 0 1
       ----------
       0 0 1 1 $ 1 1
     @ 1 0 0 1
       ----------
       0 1 0 0 1
     @ 1 0 0 1
       ----------
       0 0 0 0
       ----------
```

↗ zero means data is accepted

(4) checksum :-

→ The checksum method includes parity bits, check digits and longitudinal redundancy check (LRC). For example if we have to transfer and detect error for a long data sequence (also called as data storing) then we divide that into shorter words - and we can store the

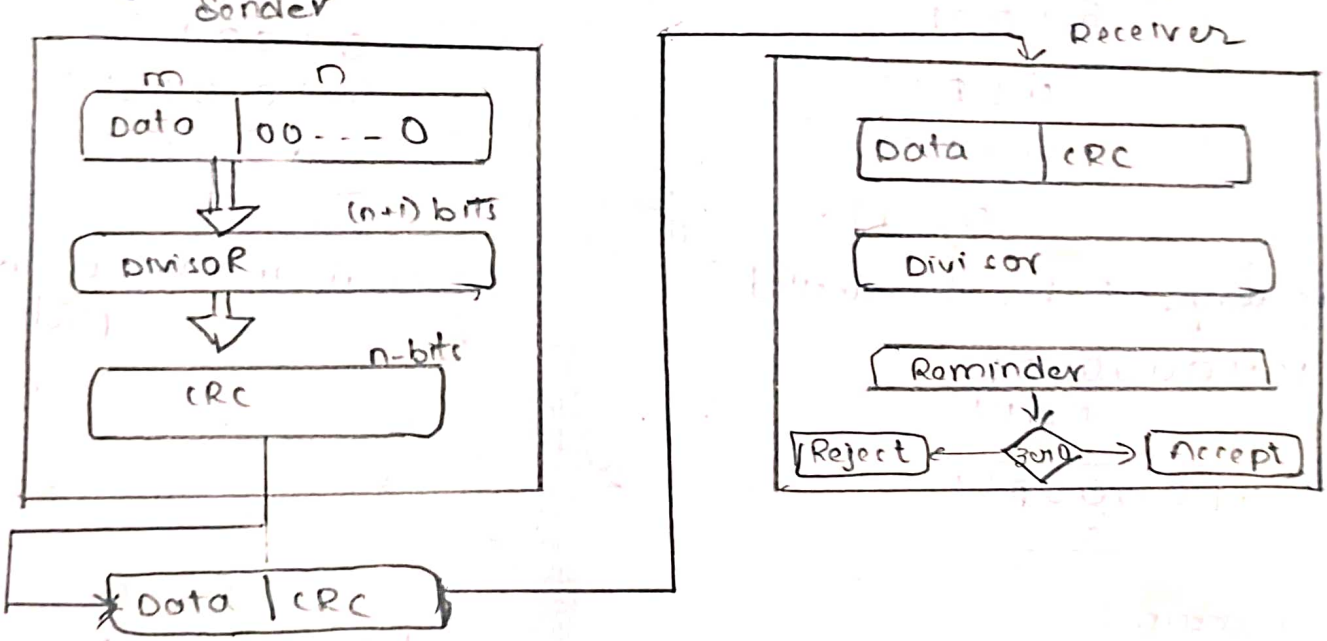
* cyclic Redundancy check (CRC) :-

→ It is based on addition, CRC is based on binary division

→ In CRC, a sequence of redundant bits called cyclic redundancy check bits are appended to the end of data unit. so that the resulting data unit becomes exactly divisible by a second predetermining binary number

→ At the destination the incoming data unit is divided by the same number. If at this step there is no remainder the data unit is assumed to be correct and is therefore accepted

→ A remainder indicate that the data unit has been damaged in transit and therefore must be rejected



Example :-

original message)

1010000

conclusion: Accept data,

Error correcting codes (Hamming code):-

→ Hamming code is a block code that is capable of detecting up to 2 simultaneous bit errors and correcting single-bit errors

→ In the late 1940's Richard Hamming recognized that the further evolution of computer required greater reliability. In particular the ability to not only detect errors, but correct them

→ Search for Error-correcting codes led to the Hamming codes, perfect 1-Error correcting codes, and the extended, Hamming codes 1-Error correcting and 2-Error detecting codes

→ once the receiver get an incoming message, it performs recalculations, to detect errors and correct them the steps for recalculation

Step 1: calculate of the number of parity bits

$2^p \rightarrow D+P+1$

Here D = data bits

P - parity value

'p' value must satisfies the formulae

Step 2: How many total bits in hamming code (D+P) & positioning the parity bits

use 2^n $(n = 0, 1, 2, 3 \ldots) = 1, 2, 4, 8$

		P_1	P_2		P_4	P_8
D_7	D_6	D_5	P_4	P_3	P_2	P_1

the data with a word of some width. For each another incoming bit we will add them to the already stored data. At every instance, the newly added word is called 'checksum'

→ In checksum error detection scheme, the data is divide into k segments each of m bits

→ In the sender's end the segment are added using 1's complement arithmetic to get the sum. The sum is complemented to get checksum

→ The checksum segment is sent along with the data segments

→ At the receiver's end all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented

If the result is zero the Received data is accepted otherwise discarded

Example orginal data.

10011001 11100010 00100100 10000100

k=4, m=8

SENDER

```
  10011001
  11100010
 ----------
  01111011
         1
 ----------
  01111100
  00100100
 ----------
  10100000
  10001001
 ----------
  10101001
         1
 ----------
Sum: 00100101
```

RECEIVER

```
  10011001
  11100010
 ----------
  01111011
         1
 ----------
  01111100
  00100100
 ----------
  10100000
  10001001
 ----------
  10101001
         1
 ----------
  00100101
  11011010
 ----------
Sum: 11111111
Complement: 00000000
```

step3: parity checking:

parity bits are calculated based upon the data bits and the parity bits using the same rule as during generation of P_1, P_2, P_4, P_8 etc

P_1 = Parity (1) – Depends on position of bits in stream 3, 5, 7, 9, 11

P_2 = Parity (2) – Depends on position of bits in stream 3, 6, 7, 10, 11

P_4 = parity (4) – Depends on position of bits in stream 5, 6, 7, 12

P_8 = parity (8) – Depends on position of bits in stream 9, 10, 11, 12

step4: Error detection and correction

The decimal equivalent of the parity bits binary value is calculated. If it is 0, there is no Error, otherwise the decimal value gives the bit position which has Error

for Example for 4-bit transmission. If $P_8P_2P_1 = 100$. It implies that the data bit at position 4, decimal Equivalent of 100 has Error - the bit is inverted to get the correct message

(1) Encode the message / Information code / input / data code 1011 in to hamming code by Even parity:

Sol (1) Find the how many parity bits required

$2^p = D+p+1$. Here D = no of data bits, P = parity value

Assume that starts $P=2$

$2^p = 4+2+1$

$4 = 4$ (not satisfied)

$p = 3$

$2^3 = 4+3+1$

$8 = 8$ (satisfied)

so, 3 parity bits required for data transmission of 1011

step 2: how many total bits in hamming code $(D+p) = 4+3 = 7$ bits

positioning the parity bits use 2^n $(n = 0, 1, 2, 3) = 1, 2, 4$ $P_1 P_2 P_4$

$P_7\ D_6\ D_5\ P_4\ D_3\ P_2\ P_1 \longrightarrow 1\ 0\ 1\ P_4\ 1\ P_2\ P_1$

$D_7\ D_6\ D_5\ P_4\ P_3\ P_2\ P_1 \longrightarrow 1\ 0\ 1\ P_4\ 1\ P_2\ P_1$

step 3:- parity checking (Given it is Even parity)

parity bits are calculated the values of P_1, P_2, P_4

for $P_1 = 3, 5, 7$ of data bits = 011

there total no of 1's is odd so it is Error

how to correct it as odd to Even $P_1 = 1$

for $P_8 = 3, 6, 7$ of data bits = 101

there total no of 1's is Even so it is correct so $P_2 = 0$

for $P_4 = 5, 6, 7$ of data bits = 101

there total no of 1's is Even so it is correct so, $P_4 = 0$

P_7	P_6	P_5	P_4	D_3	P_2	P_1	→	1	0	1	0	1	0	1

Redundancy

The main concept in the detection of error is redundancy, but to be able to detect or correct the errors some extra bits are needed and these bits are called the redundant bit that are added by the sender and removed by the receiver.

Error detection versus correction

In case of error detection one has to check whether an error has occurred or not the answer lies in a yes or no.

Whereas in case of error correction one needs to know the number of errors, location of errors, location from which the message has been received and also the error has to be fixed.

Forward error correction, retransmission

There are basically two main methods of error correction-:

- Forward error correction method
- Correction by retransmission

The forward error correction method is one in which the receiver tries to guess the message using the redundant bit.

Where as the correction by retransmission the receiver detects the occurrence of the error and asks the sender to resend the information.

Network redundancy is a process of providing alternate paths for traffic in a network. This ensures all the data flows seamlessly in the event of a failure. A reliable network is critical for businesses and organizations that require continuous connectivity to ensure maximum efficiency and minimize downtime. The idea behind network redundancy is to provide multiple paths for traffic, which ensures that if one device fails, another can take over automatically, minimizing downtime and ensuring continuity of service.

Types of Network Redundancy

There are two primary types of network redundancy: fault tolerance and high availability.

Fault Tolerance

Fault tolerance uses complete hardware redundancy, meaning that there is a complete duplicate of the system hardware running side-by-side with the primary system. This type of redundancy delivers near-zero downtime but is expensive to implement.

High Availability

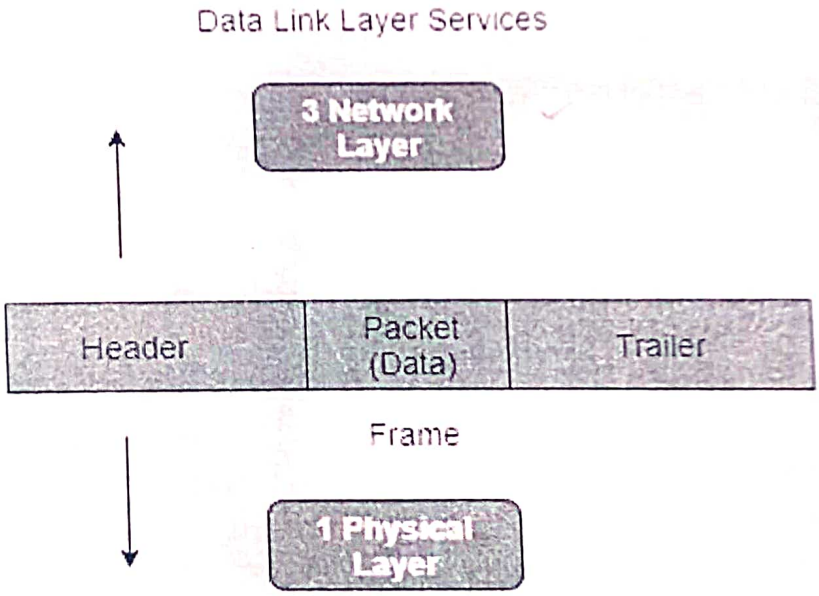
High availability, on the other hand, does not duplicate all of the physical hardware. Instead, a cluster of servers is run together & the servers monitor each other with failover capabilities. It means that if there is a problem on one server, a backup can take over. Although installing high-availability infrastructure is less expensive, there is a chance that service disruptions could have a slight negative impact.

Framing in Data Link Layer

Read Discuss Courses Video

Frames are the units of digital transmission, particularly in computer networks and telecommunications. Frames are comparable to the packets of energy called photons in the case of light energy. Frame is continuously used in Time Division Multiplexing process.

Framing is a point-to-point connection between two computers or devices consisting of a wire in which data is transmitted as a stream of bits. However, these bits must be framed into discernible blocks of information. Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures. Frames have headers that contain information such as error-checking codes.



At the data link layer, it extracts the message from the sender and provides it to the receiver by providing the sender's and receiver's addresses. The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption.

The process of dividing the data into frames and reassembling it is transparent to the user and is handled by the data link layer.

Framing is an important aspect of data link layer protocol design because it allows the

able to detect it. Station detects frames by looking out for a special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimiter).

- **How does the station detect a frame:** Every station listens to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.
- **Detecting end of frame:** When to stop reading the frame.
- **Handling errors:** Framing errors may occur due to noise or other transmission errors, which can cause a station to misinterpret the frame. Therefore, error detection and correction mechanisms, such as cyclic redundancy check (CRC), are used to ensure the integrity of the frame.
- **Framing overhead:** Every frame has a header and a trailer that contains control information such as source and destination address, error detection code, and other protocol-related information. This overhead reduces the available bandwidth for data transmission, especially for small-sized frames.
- **Framing incompatibility:** Different networking devices and protocols may use different framing methods, which can lead to framing incompatibility issues. For example, if a device using one framing method sends data to a device using a different framing method, the receiving device may not be able to correctly interpret the frame.
- **Framing synchronization:** Stations must be synchronized with each other to avoid collisions and ensure reliable communication. Synchronization requires that all stations agree on the frame boundaries and timing, which can be challenging in complex networks with many devices and varying traffic loads.
- **Framing efficiency:** Framing should be designed to minimize the amount of data overhead while maximizing the available bandwidth for data transmission. Inefficient framing methods can lead to lower network performance and higher latency.

Types of framing

There are two types of framing:

1. **Fixed-size:** 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.

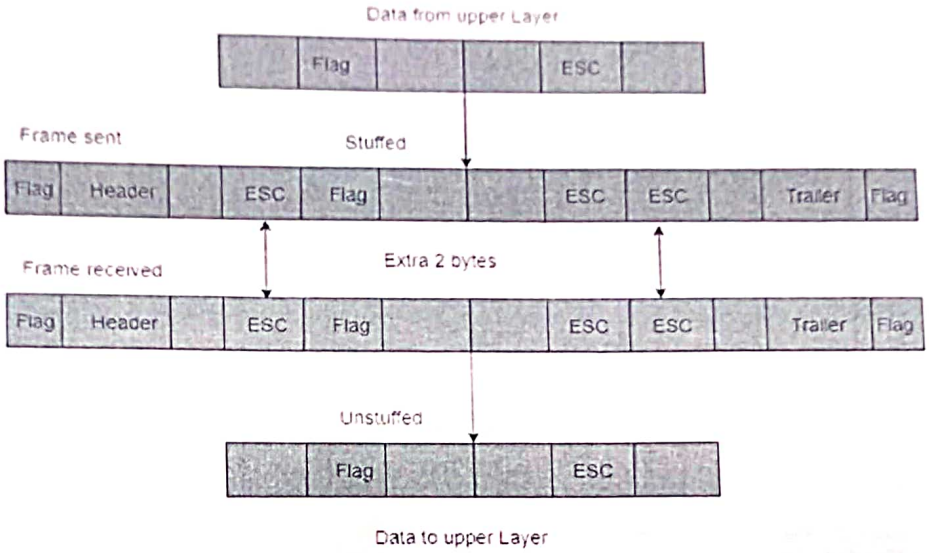
- **Drawback:** It suffers from internal fragmentation if the data size is less than the frame size
- **Solution:** Padding

2. **Variable size:** In this, there is a need to define the end of the frame as well as the beginning of the next frame to distinguish. This can be done

frame. Used in **Token Ring**. The problem with this is that ED can occur in the data. This can be solved by:

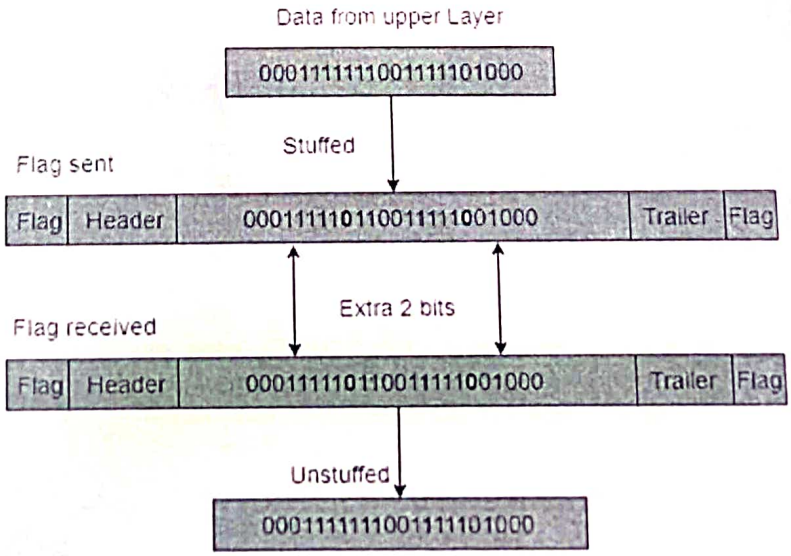
1. Character/Byte Stuffing: Used when frames consist of characters. If data contains ED then, a byte is stuffed into data to differentiate it from ED.

Let ED = "$" –> if data contains '$' anywhere, it can be escaped using '\O' character.
–> if data contains '\O$' then, use '\O\O\O$'($ is escaped using \O and \O is escaped using \O).



Disadvantage – It is very costly and obsolete method.

2. Bit Stuffing: Let ED = 01111 and if data = 01111
–> Sender stuffs a bit to break the pattern i.e. here appends a 0 in data = 011101.
–> Receiver receives the frame.
–> If data contains 011101, receiver removes the 0 and reads the data.



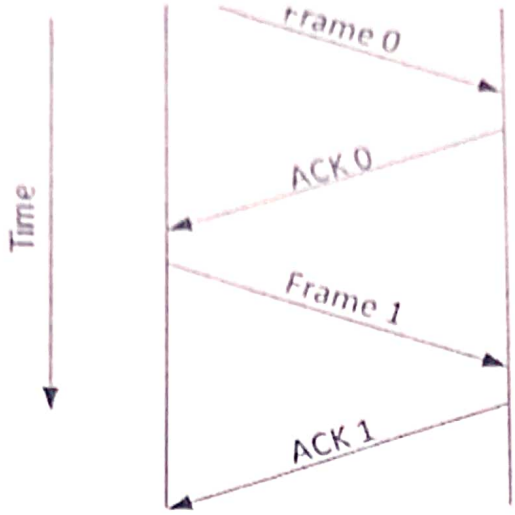
Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Two types of mechanisms can be deployed to control the flow:

- **Stop and Wait**
  This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



- **Sliding Window**
  In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

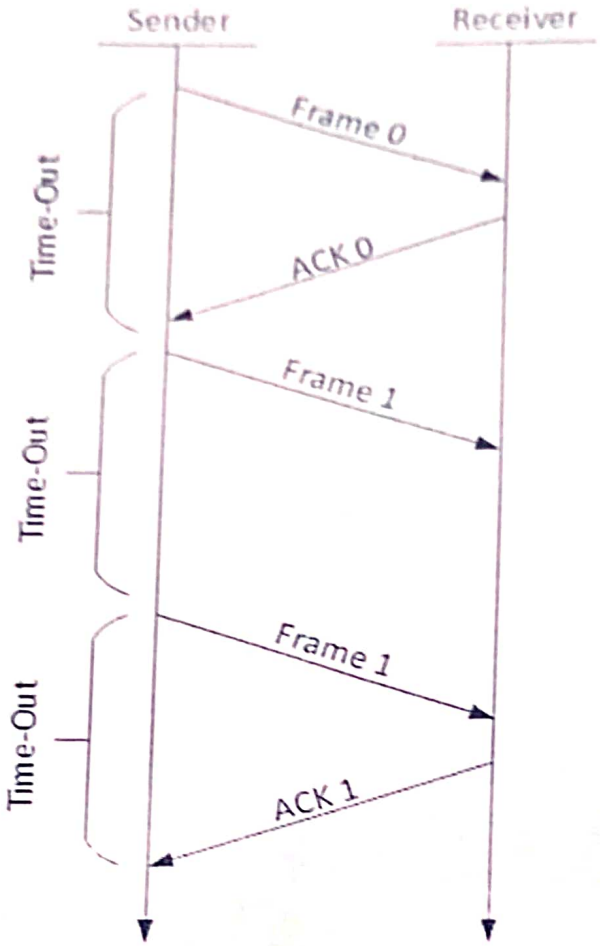
Error Control

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

- **Error detection** - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
- **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.
- **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
- **Retransmission:** The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

- Stop-and-wait ARQ

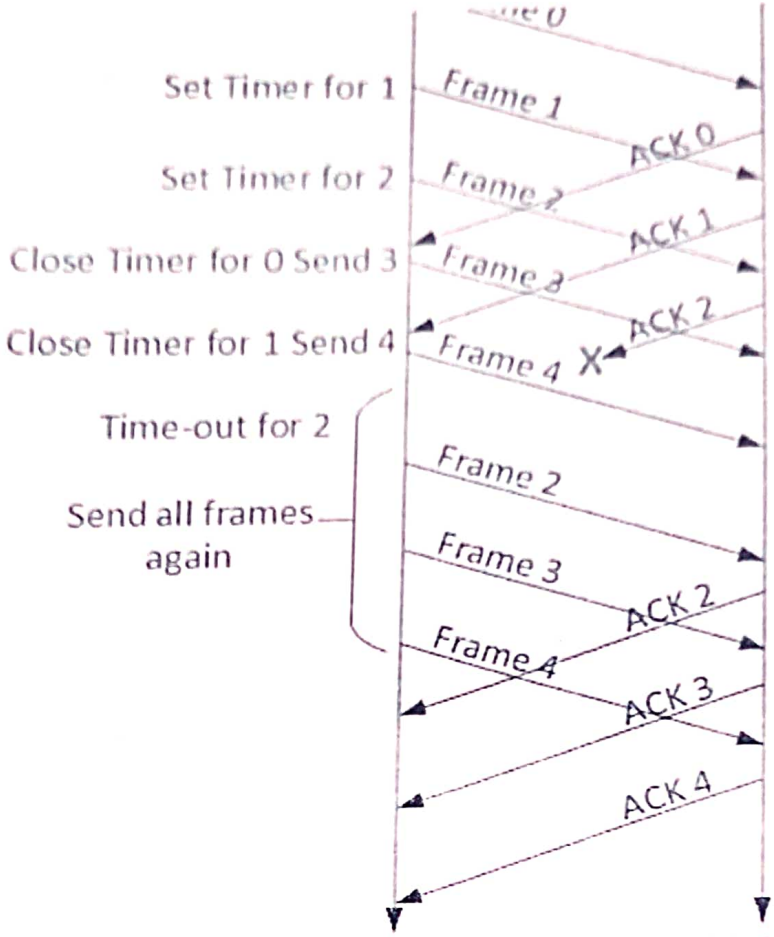


The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.

- Go-Back-N ARQ (Automatic Repeat request)

Stop and wait ARQ mechanism does not utilize the resources at their best.When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window

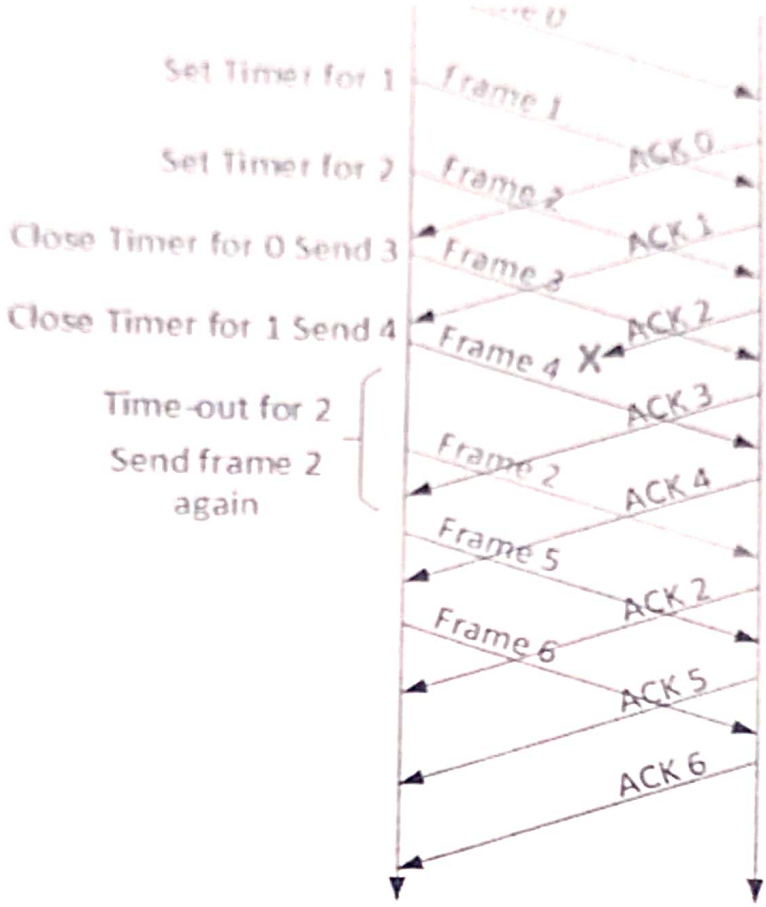


The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

- ## Selective Repeat ARQ

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.

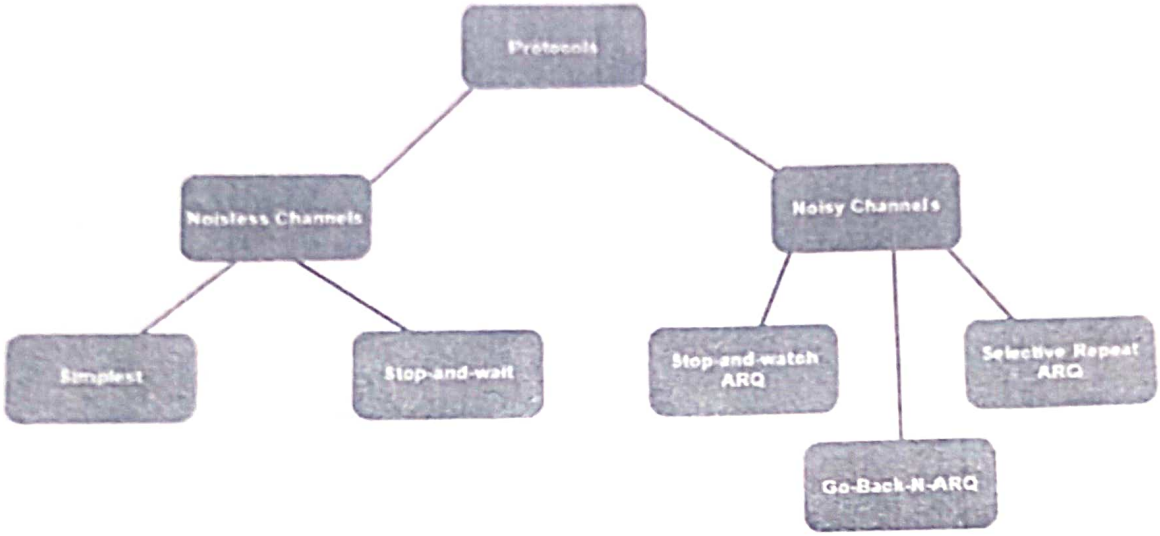


In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.

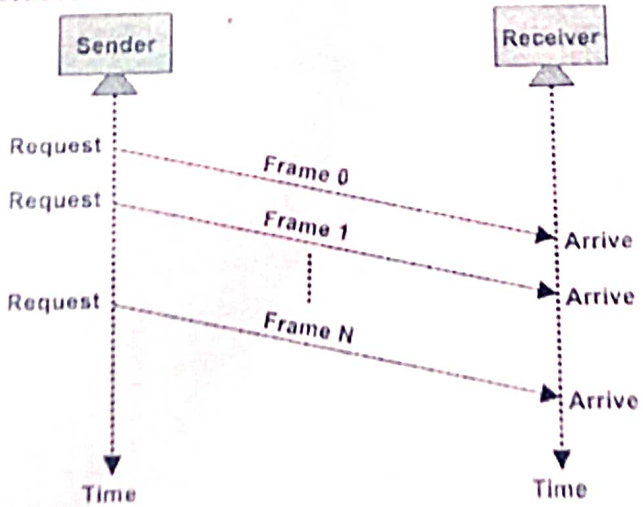
Protocols in Noiseless and Noisy Channel

The study of protocols is divided into two categories: those that can be applied to channels with no noise or errors and those that can be applied to channels with noise or errors. Although the first group of protocols cannot be applied in real-world situations, they provide a foundation for protocols for noisy channels.



The noiseless channel has the following two protocols:

1. **SIMPLEST Protocol** - A simple protocol for a noiseless channel would be one that involves the direct transfer of data from the source to the destination without any intermediate processing. In this scenario, the channel is assumed to be noise-free, which means that the data transmitted remains intact and does not get corrupted. In a noiseless channel, a simple protocol could consist of a straightforward method for transmitting data, such as sending one bit at a time, with no error correction or flow control mechanisms in place.



Simplest Protocol

A data flow diagram (DFD) is a graphical representation of the flow of data in a system. In the context of the simplest protocol, a DFD can illustrate the movement of data between the sender and the receiver. The DFD would show how the sender sends the data frames to the receiver, how the receiver processes the data, and what happens if any errors occur during the transfer. It could also show the absence of flow control and error control mechanisms, which are typically included in more complex protocols. The DFD can help to clarify the basic functioning of the simplest protocol, making it easier to understand and implement.

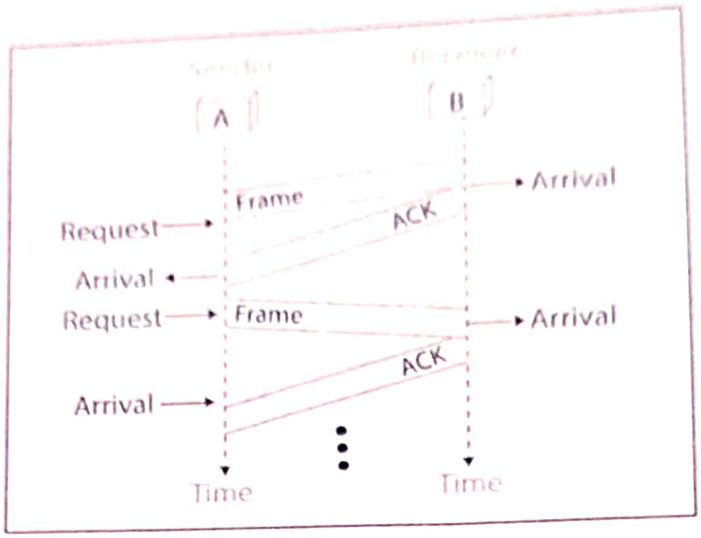
2. **STOP-AND-WAIT Protocol -** Stop and wait is a protocol that is used for reliable data transmission in a noiseless channel. In this protocol, the sender sends a single packet at a time and waits for an acknowledgment (ACK) from the receiver before sending the next packet. This way, the sender can ensure that each packet is received by the receiver and has been successfully processed. If the sender does not receive an ACK within a certain time frame, the packet is considered lost and must be retransmitted, The stop and wait protocol is simple and efficient, but it has one major drawback. Because only one packet can be transmitted at a time, the overall data transmission rate is relatively slow. To overcome this limitation, the sliding window protocol was developed. In the sliding window protocol, multiple packets can be transmitted at the same time, allowing for faster data transmission. Despite this limitation, the stop and wait protocol is still widely used in many applications due to its simplicity and reliability.

Flow Diagram

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

1. The sender transmits a data frame to the receiver.
2. The sender waits for an acknowledgment (ACK) from the receiver.
3. The receiver processes the received data frame.
4. The receiver sends an ACK to the sender to confirm receipt of the data frame.

5. The sender continues to transmit the next data frame, repeating the process from step i.



Noisy Channel protocols

A Noisy Channel Protocol is a type of communication protocol that is used in communication systems where the transmission channel may introduce errors into the transmitted data. This type of protocol is designed to deal with errors in the communication channel and ensure that the data being transmitted is received accurately at the receiver end. The main objective of Noisy Channel Protocols is to minimize the error rate in the transmitted data by using techniques such as error detection and correction, flow control, and retransmission of lost or corrupted data frames. Some examples of Noisy Channel Protocols include the Stop-and-Wait Protocol, the Sliding Window Protocol, and the Automatic Repeat Request (ARQ) Protocol.

STOP-AND-WAIT ARQ Protocol - The Stop and Wait protocol is a protocol used for reliable data transmission over a noisy channel. In this protocol, the sender only sends one frame at a time and waits for an acknowledgment (ACK) from the receiver before sending the next frame. This helps to ensure that the receiver receives the data correctly and eliminates the need for retransmission in the case of errors caused by the

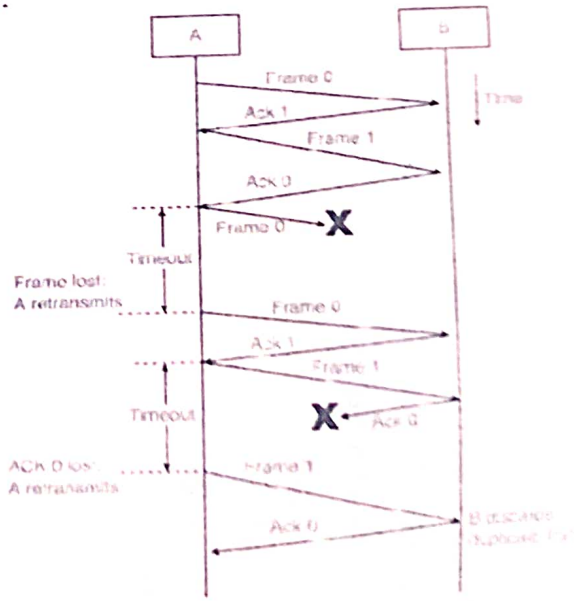
noisy channel. The sender continuously monitors the channel for errors, and if an error is detected, it waits for the next ACK before resending the frame. This protocol adds error control to the basic unidirectional communication of data frames and ACK frames in the opposite direction.

Flow Diagram

A data flow diagram in the Stop-and-Wait protocol in a noisy channel can be used to describe the flow of data between the sender and the receiver. This diagram generally includes the following components:

***Sender*:** The sender sends data frames one at a time, and waits for a response (ACK or NACK) from the receiver before sending the next data frame.

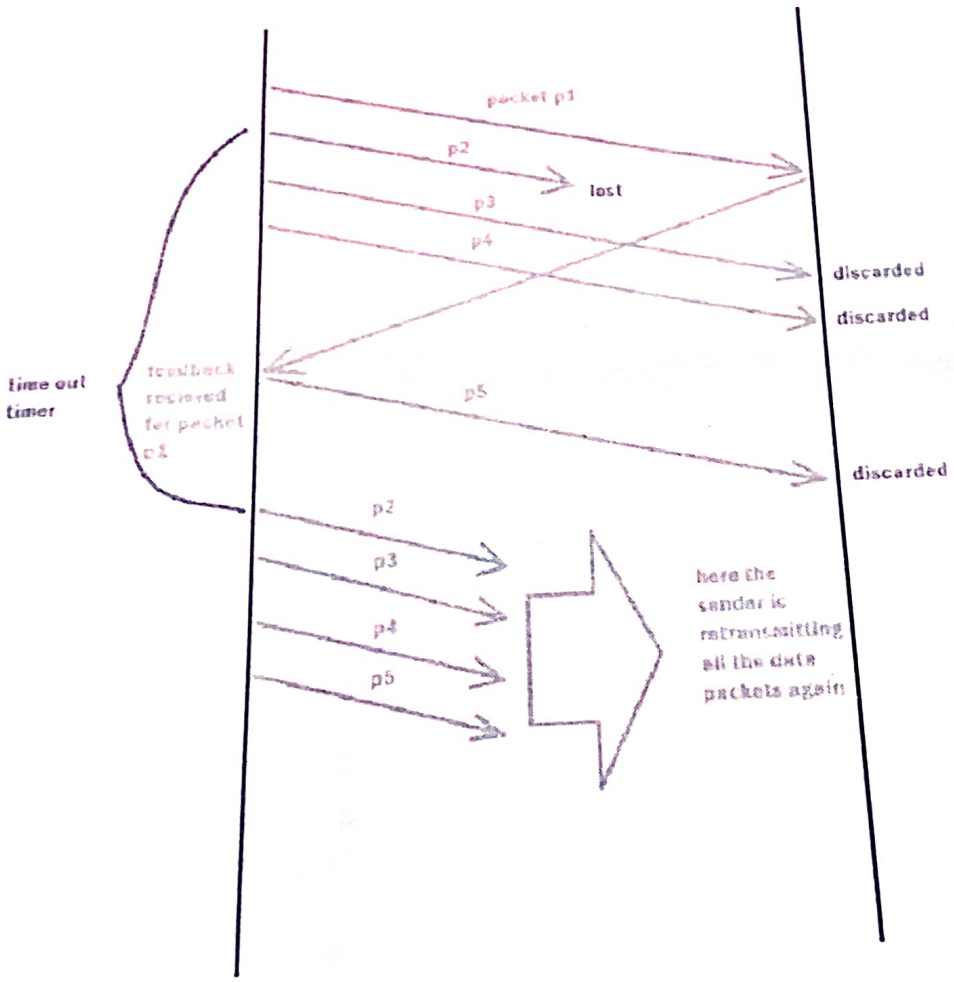
1. ***Receiver*:** The receiver receives the data frames and processes them. If the frame is received correctly, the receiver sends an ACK signal to the sender. If the frame is not received correctly, the receiver sends a NACK signal to the sender.
2. ***Noisy Channel*:** The noisy channel is the medium through which the data frames are transmitted from the sender to the receiver. The channel can add noise to the data frames, resulting in errors and corruption of the data.
3. ***Error Detection*:** The receiver uses error detection techniques such as checksums to detect errors in the received data frames.
4. ***Error Correction*:** If an error is detected, the receiver sends a NACK signal to the sender, requesting a retransmission of the frame.



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

Flow Diagram

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



Sender Side:

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

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

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

Receiver Side:

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

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

c. If a packet is received with errors, the receiver discards the packet and sends a negative acknowledgment (NAK) to the sender with the sequence number of the next expected packet.

Sender Side (in case of no ACK received):

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

Sender Side (in case of NAK received):

a. If the sender receives a NAK from the receiver, the sender retransmits only the packets that were not correctly received by the receiver.

b. The sender resets the timer for each packet that was retransmitted.

c. The sender waits for an ACK from the receiver.

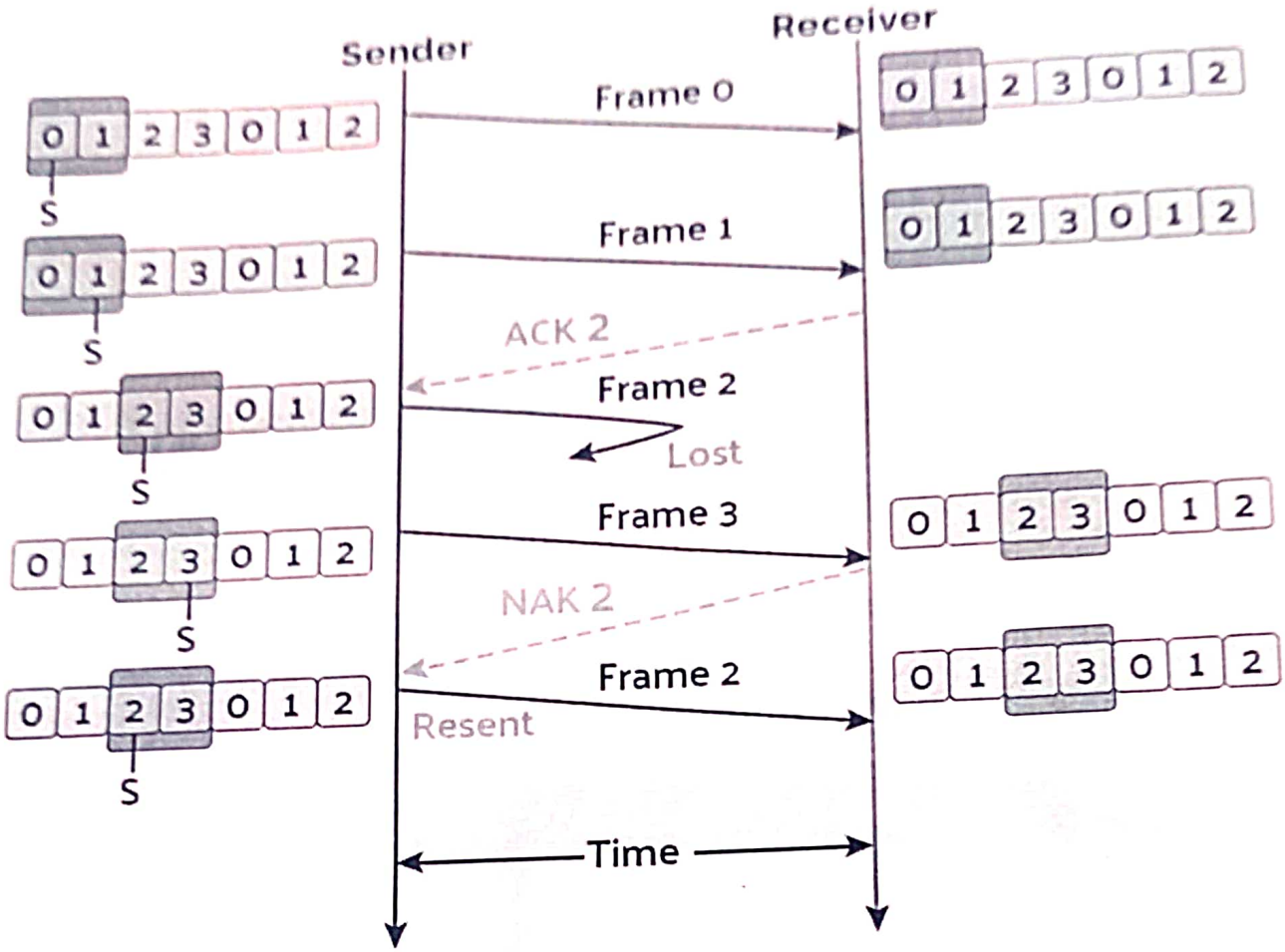
The above steps are repeated until all packets have been successfully received by the receiver. The Go-Back-N ARQ protocol provides a reliable mechanism for transmitting data over a noisy channel while minimizing the number of retransmissions required.

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

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

Flow Diagram

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



Sender Side:

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

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

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

Receiver Side:

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

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

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

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

Sender Side (in case of no ACK received):

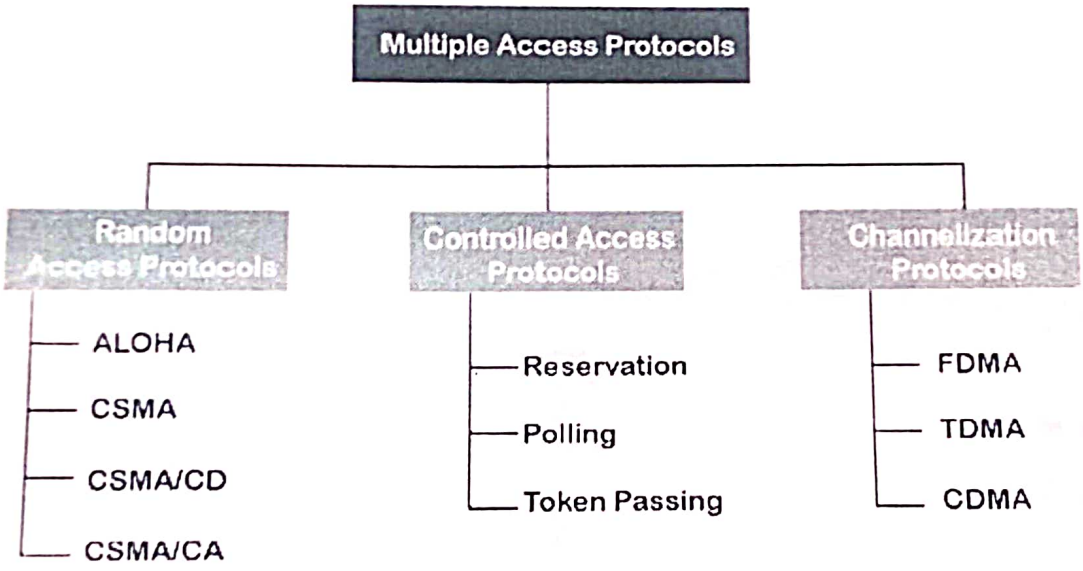
1. If the sender does not receive an ACK before the timer for a packet expires, the sender retransmits only that packet.
2. The sender resets the timer for the retransmitted packet.
3. The sender waits for an ACK from the receiver.

Sender Side (in case of NAK received):

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

When a sender and receiver have a dedicated link to transmit data packets, the data link control is enough to handle the channel. Suppose there is no dedicated path to communicate or transfer the data between two devices. In that case, multiple stations access the channel and simultaneously transmits the data over the channel. It may create collision and cross talk. Hence, the multiple access protocol is required to reduce the collision and avoid crosstalk between the channels.

Following are the types of multiple access protocol that is subdivided into the different process as:



A. Random Access Protocol

In this protocol, all the station has the equal priority to send the data over a channel. In random access protocol, one or more stations cannot depend on another station nor any station control another station. Depending on the channel's state (idle or busy), each station transmits the data frame. However, if more than one station sends the data over a channel, there may be a collision or data conflict. Due to the collision, the data frame packets may be lost or changed. And hence, it does not receive by the receiver end.

Following are the different methods of random-access protocols for broadcasting frames on the channel.

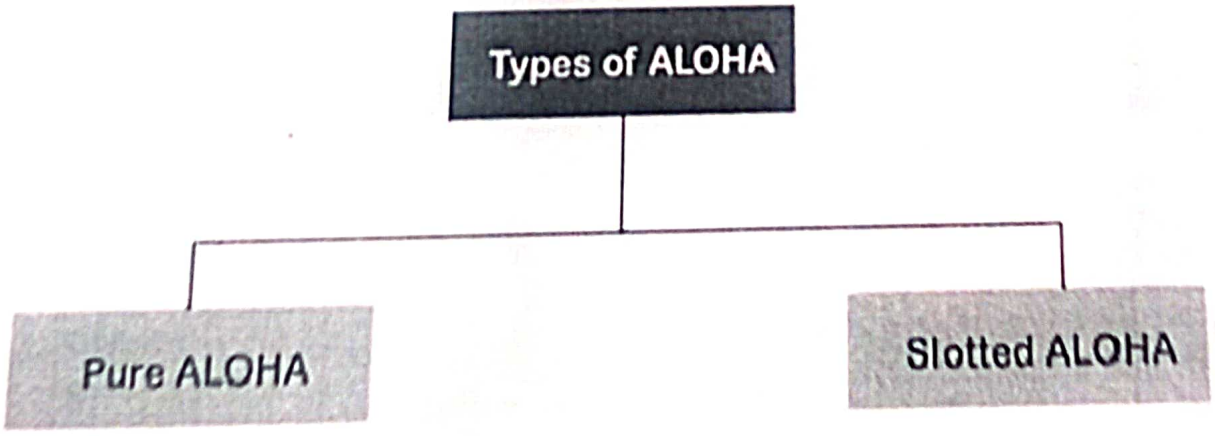
- Aloha
- CSMA
- CSMA/CD
- CSMA/CA

ALOHA Random Access Protocol

It is designed for wireless LAN (Local Area Network) but can also be used in a shared medium to transmit data. Using this method, any station can transmit data across a network simultaneously when a data frameset is available for transmission.

Aloha Rules

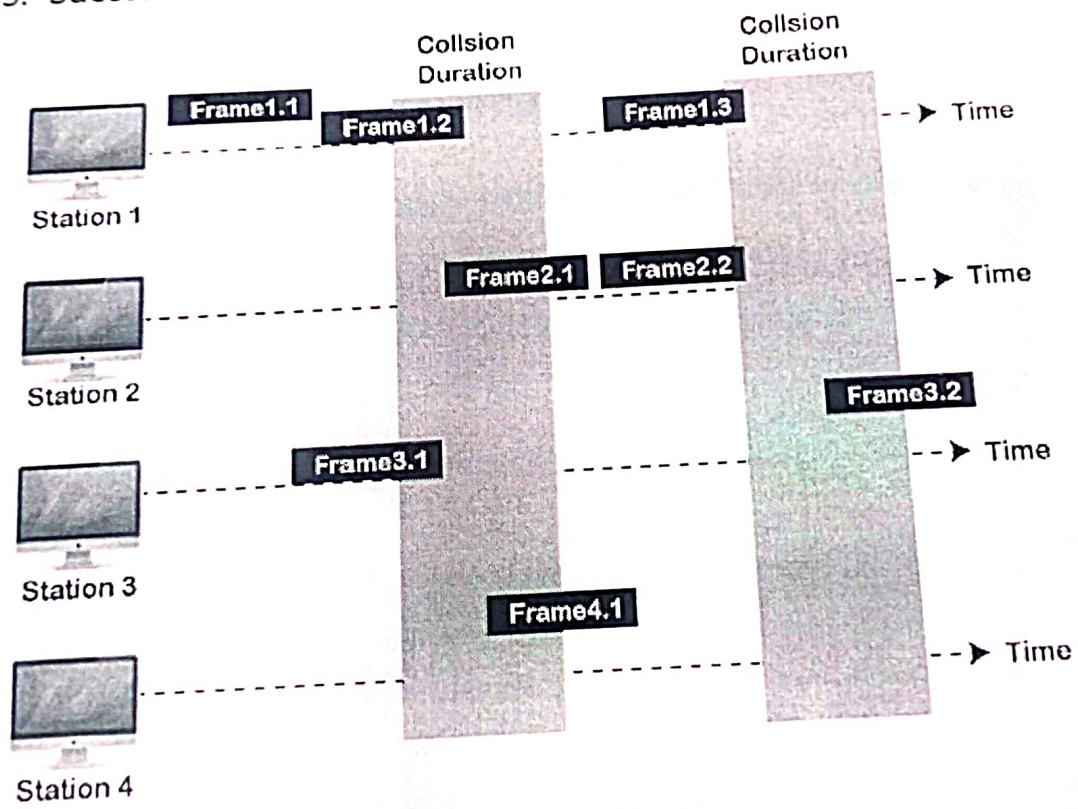
1. Any station can transmit data to a channel at any time.
2. It does not require any carrier sensing.
3. Collision and data frames may be lost during the transmission of data through multiple stations.
4. Acknowledgment of the frames exists in Aloha. Hence, there is no collision detection.
5. It requires retransmission of data after some random amount of time.



Pure Aloha

Whenever data is available for sending over a channel at stations, we use Pure Aloha. In pure Aloha, when each station transmits data to a channel without checking whether the channel is idle or not, the chances of collision may occur, and the data frame can be lost. When any station transmits the data frame to a channel, the pure Aloha waits for the receiver's acknowledgment. If it does not acknowledge the receiver end within the specified time, the station waits for a random amount of time, called the backoff time (Tb). And the station may assume the frame has been lost or destroyed. Therefore, it retransmits the frame until all the data are successfully transmitted to the receiver.

1. The total vulnerable time of pure Aloha is 2 * Tfr.
2. Maximum throughput occurs when G = 1/ 2 that is 18.4%.
3. Successful transmission of data frame is S = G * e ^ - 2 G.



Frames in Pure ALOHA

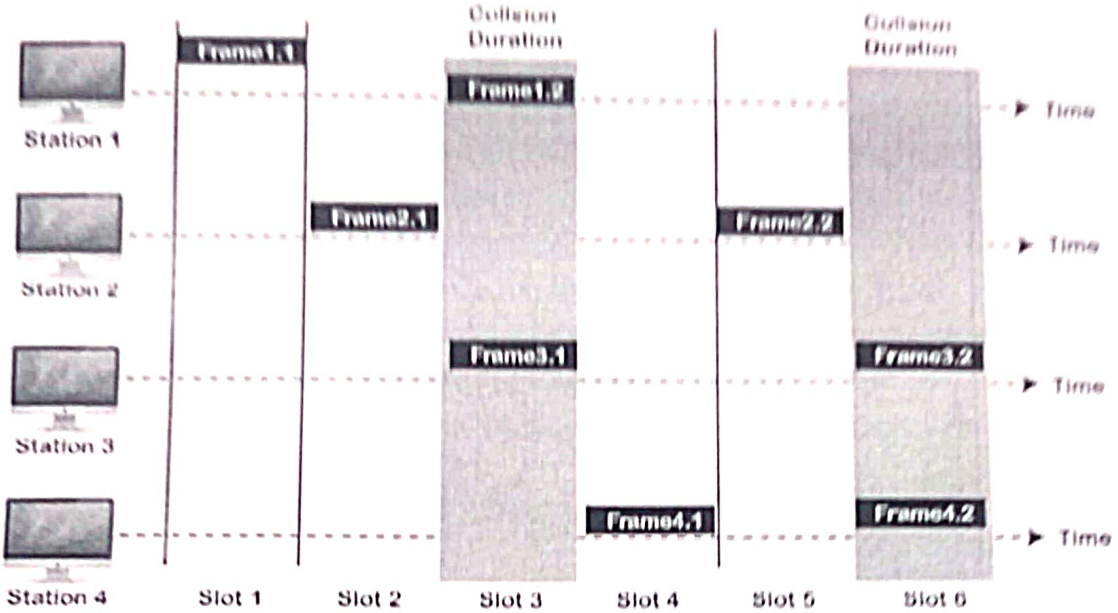
As we can see in the figure above, there are four stations for accessing a shared channel and transmitting data frames. Some frames collide because most stations send their frames at the same time. Only two frames, frame 1.1

and frame 2.2, are successfully transmitted to the receiver end. At the same time, other frames are lost or destroyed. Whenever two frames fall on a shared channel simultaneously, collisions can occur, and both will suffer damage. If the new frame's first bit enters the channel before finishing the last bit of the second frame. Both frames are completely finished, and both stations must retransmit the data frame.

Slotted Aloha

The slotted Aloha is designed to overcome the pure Aloha's efficiency because pure Aloha has a very high possibility of frame hitting. In slotted Aloha, the shared channel is divided into a fixed time interval called **slots**. So that, if a station wants to send a frame to a shared channel, the frame can only be sent at the beginning of the slot, and only one frame is allowed to be sent to each slot. And if the stations are unable to send data to the beginning of the slot, the station will have to wait until the beginning of the slot for the next time. However, the possibility of a collision remains when trying to send a frame at the beginning of two or more station time slot.

1. Maximum throughput occurs in the slotted Aloha when G = 1 that is 37%.
2. The probability of successfully transmitting the data frame in the slotted Aloha is S = G * e ^ - 2 G.
3. The total vulnerable time required in slotted Aloha is Tfr.



Frames in Slotted ALOHA

CSMA (Carrier Sense Multiple Access)

It is a **carrier sense multiple access** based on media access protocol to sense the traffic on a channel (idle or busy) before transmitting the data. It means that if the channel is idle, the station can send data to the channel. Otherwise, it must wait until the channel becomes idle. Hence, it reduces the chances of a collision on a transmission medium.

CSMA/ CD

It is a **carrier sense multiple access/ collision detection** network protocol to transmit data frames. The CSMA/CD protocol works with a medium access control layer. Therefore, it first senses the shared channel before broadcasting the frames, and if the channel is idle, it transmits a frame to check whether the transmission was successful. If the frame is successfully received, the station sends another frame. If any collision is detected in the CSMA/CD, the station sends a jam/ stop signal to the shared channel to terminate data transmission. After that, it waits for a random time before sending a frame to a channel.

CSMA/ CA

It is a **carrier sense multiple access/collision avoidance** network protocol for carrier transmission of data frames. It is a protocol that works with a medium access control layer. When a data frame is sent to a channel, it receives an acknowledgment to check whether the channel is clear. If the station receives only a single (own) acknowledgments, that means the data frame has been successfully transmitted to the receiver. But if it gets two signals (its own and one more in which the collision of frames), a collision of the frame occurs in the shared channel. Detects the collision of the frame when a sender receives an acknowledgment signal.

B. Controlled Access Protocol

It is a method of reducing data frame collision on a shared channel. In the controlled access method, each station interacts and decides to send a data frame by a particular station approved by all other stations. It means that a single station cannot send the data frames unless all other stations are not approved. It has three types of controlled access: **Reservation, Polling**, and **Token Passing**.

Reservation

- In the reservation method, a station needs to make a reservation before sending data.
- The timeline has two kinds of periods:
    1. Reservation interval of fixed time length
    2. Data transmission period of variable frames.

Polling

- Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.

- In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.
- The message sent by the controller contains the address of the node being selected for granting access.

Token Passing

- In token passing scheme, the stations are connected logically to each other in form of ring and access to stations is governed by tokens.
- A token is a special bit pattern or a small message, which circulate from one station to the next in some predefined order.
- In Token ring, token is passed from one station to another adjacent station in the ring whereas incase of Token bus, each station uses the bus to send the token to the next station in some predefined order.

C. Channelization Protocols

It is a channelization protocol that allows the total usable bandwidth in a shared channel to be shared across multiple stations based on their time, distance and codes. It can access all the stations at the same time to send the data frames to the channel.

Following are the various methods to access the channel based on their time, distance and codes:

1. FDMA (Frequency Division Multiple Access)
2. TDMA (Time Division Multiple Access)
3. CDMA (Code Division Multiple Access)

FDMA

It is a frequency division multiple access (**FDMA**) method used to divide the available bandwidth into equal bands so that multiple users can send data through a different frequency to the subchannel. Each station is reserved with

a particular band to prevent the crosstalk between the channels and interferences of stations.

TDMA

Time Division Multiple Access (**TDMA**) is a channel access method. It allows the same frequency bandwidth to be shared across multiple stations. And to avoid collisions in the shared channel, it divides the channel into different frequency slots that allocate stations to transmit the data frames. The same **frequency** bandwidth into the shared channel by dividing the signal into various time slots to transmit it. However, TDMA has an overhead of synchronization that specifies each station's time slot by adding synchronization bits to each slot.

CDMA

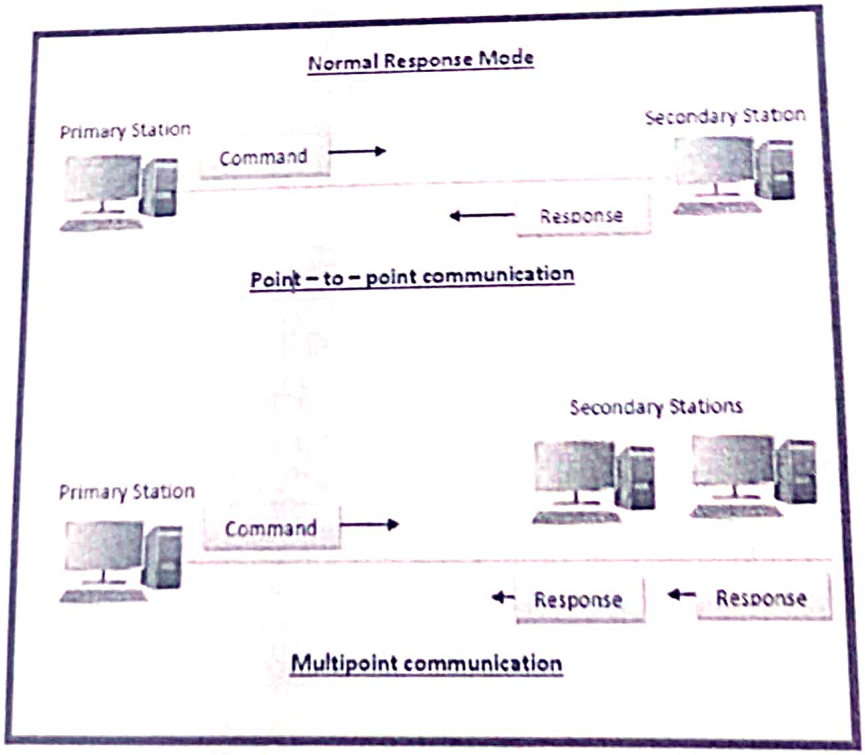
The code division multiple access (CDMA) is a channel access method. In CDMA, all stations can simultaneously send the data over the same channel. It means that it allows each station to transmit the data frames with full frequency on the shared channel at all times. It does not require the division of bandwidth on a shared channel based on time slots. If multiple stations send data to a channel simultaneously, their data frames are separated by a unique code sequence. Each station has a different unique code for transmitting the data over a shared channel. For example, there are multiple users in a room that are continuously speaking. Data is received by the users if only two-person interact with each other using the same language. Similarly, in the network, if different stations communicate with each other simultaneously with different code language.

High-level Data Link Control (HDLC) is a group of communication protocols of the data link layer for transmitting data between network points or nodes. Since it is a data link protocol, data is organized into frames. A frame is transmitted via the network to the destination that verifies its successful arrival. It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.

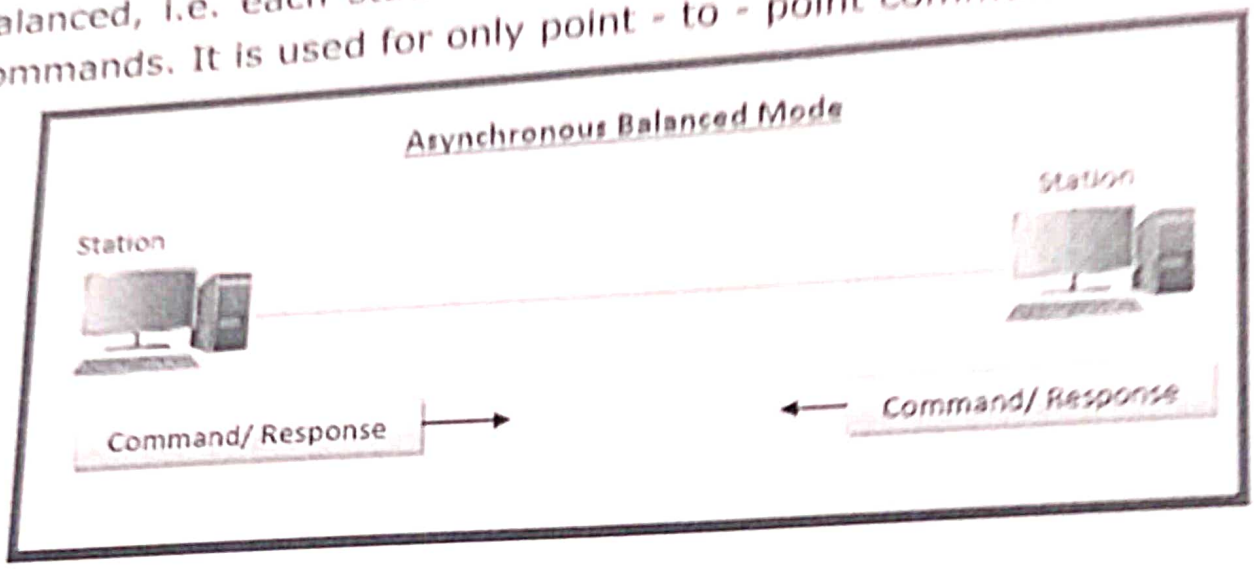
Transfer Modes

HDLC supports two types of transfer modes, normal response mode and asynchronous balanced mode.

- **Normal Response Mode (NRM)** – Here, two types of stations are there, a primary station that send commands and secondary station that can respond to received commands. It is used for both point - to - point and multipoint communications.



Asynchronous Balanced Mode (ABM) – Here, the configuration is balanced, i.e. each station can both send commands and respond to commands. It is used for only point - to - point communications.



HDLC Frame

HDLC is a bit - oriented protocol where each frame contains up to six fields. The structure varies according to the type of frame. The fields of a HDLC frame are –

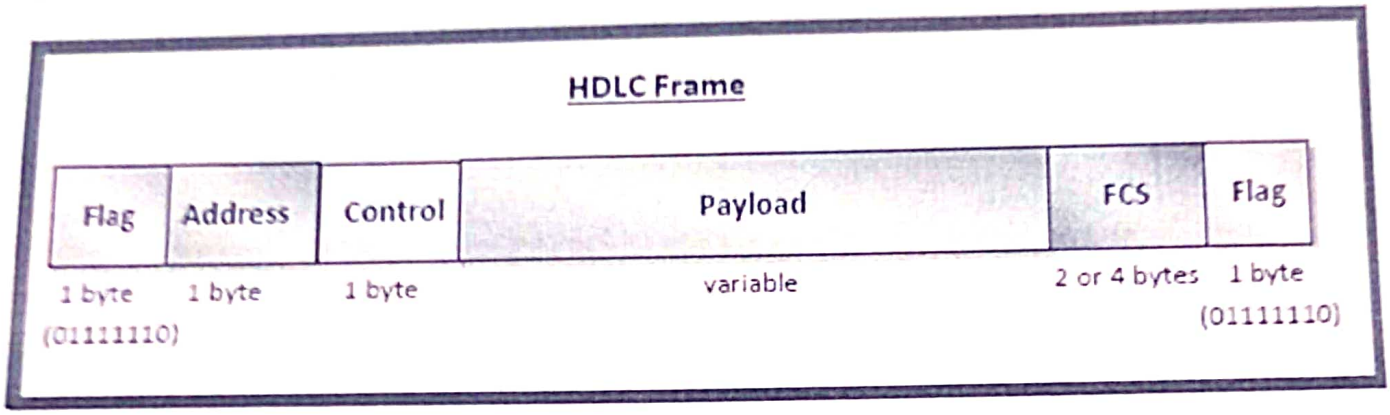
Flag – It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.

Address – It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.

Control – It is 1 or 2 bytes containing flow and error control information.

Payload – This carries the data from the network layer. Its length may vary from one network to another.

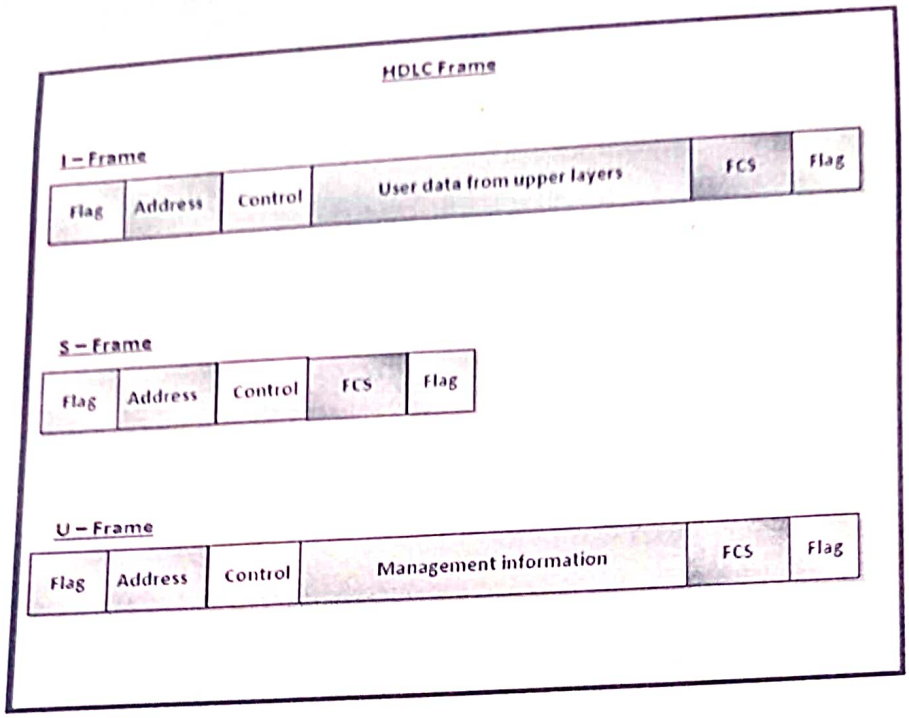
FCS – It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)



Types of HDLC Frames

There are three types of HDLC frames. The type of frame is determined by the control field of the frame –

- **I-frame** – I-frames or Information frames carry user data from the network layer. They also include flow and error control information that is piggybacked on user data. The first bit of control field of I-frame is 0.
- **S-frame** – S-frames or Supervisory frames do not contain information field. They are used for flow and error control when piggybacking is not required. The first two bits of control field of S-frame is 10.
- **U-frame** – U-frames or Un-numbered frames are used for myriad miscellaneous functions, like link management. It may contain an information field, if required. The first two bits of control field of U-frame is 11.

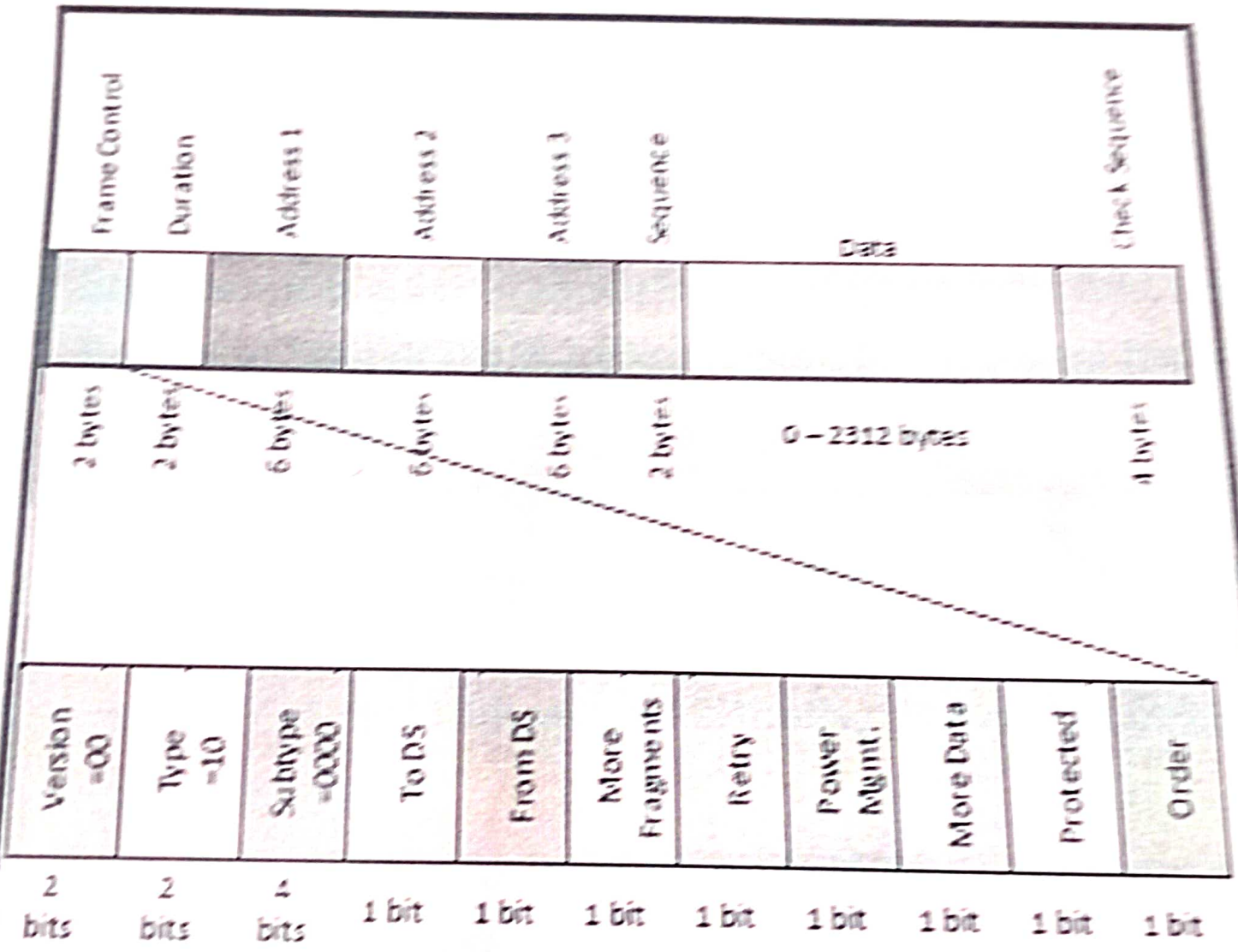


The IEEE 802.11 standard, lays down the architecture and specifications of wireless local area networks (WLANs). WLAN or WiFi uses high frequency radio waves instead of cables for connecting the devices in LAN. Users connected by WLANs can move around within the area of network coverage.

The 802.11 MAC sublayer provides an abstraction of the physical layer to the logical link control sublayer and upper layers of the OSI network. It is responsible for encapsulating frames and describing frame formats.

MAC Sublayer Frame Structure of IEEE 802.11

The main fields of a frame in WLANs as laid down by IEEE 802.11 are as depicted in the following diagram –



MAC Sublayer Frame Structure of IEEE 802.11

- **Frame Control** –It is a 2 bytes starting field composed of 11 subfields. It contains control information of the frame. The 11 subfields are –
- **Protocol version** – The first sub-field is a two – bit field set to 00. It has been included to allow future versions of IEE 802.11 to operate simultaneously.
- **Type** – It is a two-bit subfield that specifies whether the frame is a data frame, control frame or a management frame.
- **Subtype** – it is a four – bit subfield states whether the field is a Request to Send (RTS) or a Clear to Send (CTS) control frame. For a regular data frame, the value is set to 0000.
- **To DS** – A single bit subfield indicating whether the frame is going to the access point (AC), which coordinates the communications in centralised wireless systems.
- **From DS** – A single bit subfield indicating whether the frame is coming from the AC.
- **More Fragments** – A single bit subfield which when set to 1 indicates that more fragments would follow.
- **Retry** – A single bit subfield which when set to 1 specifies a retransmission of a previous frame.
- **Power Management** – A single bit subfield indicating that the sender is adopting power-save mode.
- **More Data** – A single bit subfield showing that sender has further data frames for the receiver.

- **Protected Frame** – A single bit subfield indicating that this is an encrypted frame.
- **Order** – The last subfield, of one – bit, informs the receiver that to the higher layers the frames should be in an ordered sequence.
- **Duration** – It is a 2-byte field that specifies the time period for which the frame and its acknowledgement occupy the channel.
- Address fields: There are three 6-byte address fields containing addresses of source, immediate destination and final endpoint respectively.
- **Sequence** – It a 2 bytes field that stores the frame numbers. It detects duplicate frames and determines the order of frames for higher layers. Among the 16 bits, the first 4 bits provides identification to the fragment and the rest 12 bits contain the sequence number that increments with each transmission.
- **Data** – This is a variable sized field that carries the payload from the upper layers. The maximum size of data field is 2312 bytes.
- **Frame Check Sequence (FCS)** – It is a 4-byte field containing error detection information.

Network Layer Services- Packetizing, Routing and Forwarding

The network Layer is the third layer in the OSI model of computer networks. Its main function is to transfer network packets from the source to the destination. It is involved both the source host and the destination host. At the source, it accepts a packet from the transport layer, encapsulates it in a datagram, and then delivers the packet to the data link layer so that it can further be sent to the receiver. At the destination, the datagram is decapsulated, and the packet is extracted and delivered to the corresponding transport layer.

Features of Network Layer

1. The main responsibility of the Network layer is to carry the data packets from the source to the destination without changing or using them.
2. If the packets are too large for delivery, they are fragmented i.e., broken down into smaller packets.
3. It decides the route to be taken by the packets to travel from the source to the destination among the multiple routes available in a network (also called routing).
4. The source and destination addresses are added to the data packets inside the network layer.

Services Offered by Network Layer

The services which are offered by the network layer protocol are as follows:

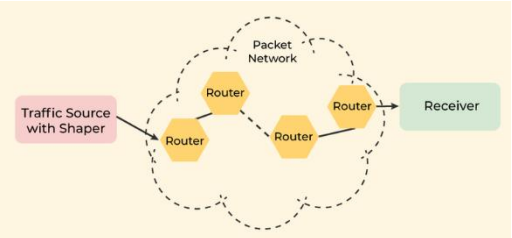
Packetizing
Routing
Forwarding

Packetizing

The process of encapsulating the data received from the upper layers of the network (also called payload) in a network layer packet at the source and decapsulating the payload from the network layer packet at the destination is known as packetizing.

The source host adds a header that contains the source and destination address and some other relevant information required by the network layer protocol to the payload received from the upper layer protocol and delivers the packet to the data link layer.

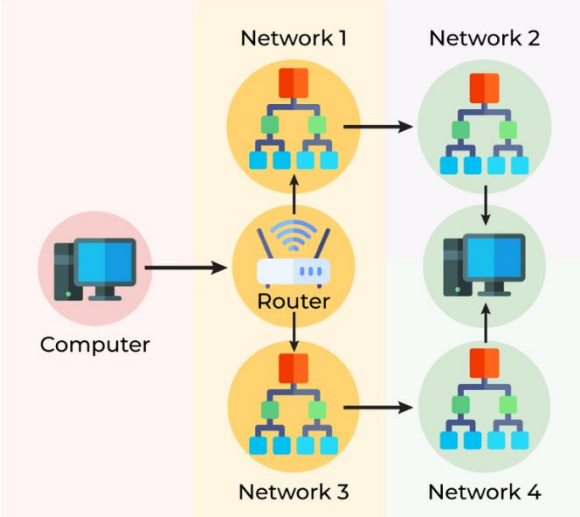
The destination host receives the network layer packet from its data link layer, decapsulates the packet, and delivers the payload to the corresponding upper layer protocol. The routers in the path are not allowed to change either the source or the destination address. The routers in the path are not allowed to decapsulate the packets they receive unless they need to be fragmented.



Routing

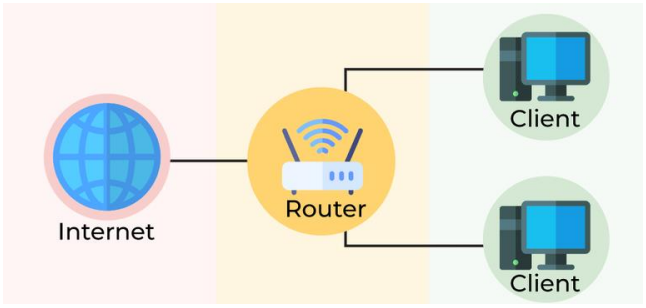
Routing is the process of moving data from one device to another device. These are two other services offered by the network layer. In a network, there are a number of routes available from the source to the destination. The network layer specifies some strategies which find out the best possible route. This process is referred to as routing. There are a number of routing

protocols that are used in this process and they should be run to help the routers coordinate with each other and help in establishing communication throughout the network.



Forwarding

Forwarding is simply defined as the action applied by each router when a packet arrives at one of its interfaces. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (unicast routing) or to some attached networks (in the case of multicast routing). Routers are used on the network for forwarding a packet from the local network to the remote network. So, the process of routing involves packet forwarding from an entry interface out to an exit interface.



Network Service Model

The network-service model defines the characteristics of end-to-end transport of data between one "edge" of the network and the other, that is, between sending and receiving end systems.

Lets now consider some possible services that the network layer can provide. In the sending host when the transport layer passes a packet to the network layer specific services that could be provided by network layer include:

a.This service guarantees that the packet will eventually arrive at tits destination.

b.Offers uniform services various types of sub network . the sub network may be defined as a set of one or more intermediate nodes which are used for establishing network connection between end systems and provide routing and relaying of frames thought it.

c.Offers uniform addressing and services to LAN and WAN the network layer defines network addresses which are being used by transport enmities to access network services. Each transport entity is a unique network connection end point identifier defined by the network layer which may be independent of addressing required by underlying layer. The network entities define network connection and may also include intermediate nodes which usually provide relaying. The network connection over intermediate subnets is known as a subnet connection and is usually handled by underlying protocols.

d.Establishes control and maintains a logical connection between the transport layer entities for exchanging the data across the network. The network connection offers point to point connection and more than one network connection may be defined between the same pair of network addresses. The network service data units are transferred transparently between transport entities over the network connection and generally do not have upper limit on their size.

e.The transport layer can adapt to different types of sub network (it is independent of the characteristics of sub network e, g routing strategies considered topologies and type of the sub network).

f.Offers an acceptable quality of services during the network connection which in turn is based on parameters such as residual error rate service availability throughput reliability etc. It also reports any undetectable error to the transport layer.

g.Provides sequencing for NDSU over the network connection when requested by transport entities.

h.Provides both connection oriented and connectionless services to the users.

In connection oriented services a logical connection is set up between two transport entities including agreement for the type of service desired parameters cost priorities transfer of data in both direction with appropriate flow control and finally the termination of the connection.

On the other hand in connectionless service network layer primitives SEND and RECEIVE are being used to send and receive the packet across the networks. The users are responsible for flow control and error control on their hosts.

Virtual Circuits & Datagram Networks

Virtual Circuits are computer networks that offer connection-oriented services, whereas Datagram networks offer connection-less services. The Internet that we use is actually built on a Datagram network (connection-less) at the network level since not all packets that travel from a source to a destination use the same route.

Virtual Circuits

Connection-oriented switching is another name for virtual circuits. Before messages are sent, a virtual circuit switching sets a predetermined routing. This route is referred to as a virtual circuit since it gives the user the impression that a passionate physical circuit exists. The call request and call accept packets are used to establish the connection between the sender and the recipient.

The term "virtual circuit" refers to a logical link between two network nodes, typically in a communications network. The path consists of many network parts that are connected by switches

There are three identifiable phases in a

- VC setup. During the setup phase, the sender contacts the network layer, specifies the receiver address, and waits for the network to set up the VC. The network layer determines the path between sender and receiver, that is, the series of links and packet switches through which all packets of the VC will travel. this typically involves updating tables in each of the packet switches in the path. During VC setup, the network layer may also reserve resources (for example, bandwidth) along the path of the VC.
- Data transfer. Once the VC has been established, data can begin to flow along the VC.
- Virtual-circuit teardown. This is initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC. The network layer will then typically inform the end system on the other

side of the network of the call termination and update the tables in each of the packet switches on the path to indicate that the VC no longer exists.

- The messages that the end systems send to the network to indicate the initiation or termination of a VC, and the messages passed between the switches to set up the VC (that is, to modify switch tables) are known as **signaling messages** and the protocols used to exchange these messages are often referred to as **signaling protocols.**

Datagram Networks

It is a method of switching packets in which every packet, or "datagram," is seen as a distinct entity. The switch uses the destination information contained in each packet to direct it to the intended location. Since no specific channel is classified for a connection session, there is no need to reserve resources. As a result, packets have a header with all of the information about the destination. A packet's header is examined by the intermediate nodes, which then select an appropriate link to another node that is closer to the destination.

Datagram networks assign resources according to the First-Come-First-Serve (FCFS) principle. Regardless of its source or destination, if another packet is being processed when a packet arrives at a router, it must wait.

Constant bit rate (CBR) network service was the first ATM service model to be standardized, probably reflecting the fact that telephone companies were the early prime movers behind ATM, and CBR network service is ideally suited for carrying real-time, constant-bit-rate audio (for example, a digitized telephone call) and video traffic. The goal of CBR service is conceptually simple--to make the network connection look like a dedicated copper or fiber connection between the sender and receiver.

A second conceptually simple ATM service class is **Unspecified bit rate (UBR)** network service. Unlike CBR service, which guarantees rate, delay, delay jitter, and loss, UBR makes no guarantees at all other than in-order delivery of cells (that is, cells that are fortunate enough to make it to the receiver). With the exception of in-order delivery, UBR service is thus equivalent to the Internet best-effort service model. As with the Internet best-effort service model, UBR also provides no feedback to the sender about whether or not a cell is dropped within the network.

If UBR can be thought of as a "best-effort" service, then **available bit rate (ABR)** network service might best be characterized as a "better" best-effort service model. The two most important additional features of ABR service over UBR service are:

A minimum cell transmission rate (MCR) is guaranteed to a connection using ABR service. If, however, the network has enough free resources at a given time, a sender may actually be able to successfully send traffic at a higher rate than the MCR.

Congestion feedback from the network. We saw in Section 3.6.3 that an ATM network can provide feedback to the sender (in terms of a congestion notification bit, or a lower rate at which to send) that controls how the sender should adjust its rate between the MCR and the peak cell rate (PCR). ABR senders control their transmission rates based on such feedback.

The final ATM service model is **variable bit rate (VBR)** network service. VBR service comes in two flavors (perhaps indicating a service class with an identity crisis!). In real-time VBR service, the acceptable cell-loss rate, delay, and delay jitter are specified as in CBR service. However, the actual source rate is allowed to vary according to parameters specified by the user to the network. The declared variability in rate may be used by the network (internally) to more efficiently allocate resources to its connections, but in terms of the loss, delay, and jitter seen by the sender, the service is essentially the same as CBR service. While early efforts in defining a VBR service model were clearly targeted toward real-time services

S. no.	Comparison Criteria	Virtual Circuits	Datagram Networks
1.	**Definition**	It is connection-oriented, which means that resources like buffers, CPU, bandwidth, etc., are	The service isconnection-less. Since there isn't a path specifically

		reserved for the period of time that a data transfer session will use the newly established VC.	designated for a connection session, no resource reservations are required.
2.	**Path**	Each server along the path has resources reserved when the first packet is sent. For the duration of the connection, subsequent packets will take the same route as the first one transmitted.	Every packet is permitted to follow any accessible path. Because routing tables on routers change continuously, intermediary routers must constantly calculate routes.
3.	**Header**	A global header is needed because every packet will take the same route. The remaining packets often do not need global headers; just the first packet of the connection does.	All packets must be linked to a header that has the correct information about the source and the upper layer data since every packet has the freedom to select any path.
4.	**Data Flow**	All packets follow a precise path, so when they arrive at their destination, they are all received in sequence.	The connection-less attribute allows data packets to arrive at their destination in any sequence, which raises the possibility that they will not be received in the correct order at the receiver's end.
5.	**Resource Requirements & Utilization**	By using virtual circuit switching, all packets are guaranteed to reach their destination. No packet will	That a packet can only be transmitted if resources like the buffer, CPU, and

		be discarded because there are no resources available.	bandwidth are available is a key disadvantage of datagram packet switching. If not, the packet will be thrown away.
6.	**Phases**	Setup, data transfer, and teardown are the threephases of the transmission process.	Any type of communication phase does not exist in Datagram Networks.
7.	**Addressing**	The routing and addressing are chosen during setup. The VC number is the only thing each packet contains as a result.	Each datagram packet contains a complete list of the source and destination addresses.
8.	**Reliability**	The aforementioned information leads to the conclusion that Virtual Circuits are a very dependable means of data transport.	Datagram networks are less reliable than virtual circuits.
9.	**Implementation and Cost**	Virtual circuits have the drawback that every time a new connection is established, resources and additional data must be allocated at every router along the line. This can be a problem if numerous customers are attempting to reserve a router's resources at the same time.	However, implementing datagram networks is usually simple and inexpensive because there isn't the added hassle of building a dedicated channelevery time an application needs to interact.
10.	**Applications**	Specifically for phone conversations, it is utilized by the ATM (Asynchronous	It is commonly employed by the IP network, which is

		Transfer Mode) Network.	utilized for data services such as the Internet.

Origins of Virtual Circuit and Datagram Networks

The evolution of datagram and virtual circuit networks reflects their origins. The notion of a virtual circuit as a central organizing principle has its roots in the telephony world, which uses real circuits. With call setup and per-call state being maintained at the routers within the network, a VC network is arguably more complex than a datagram network.

This, too, is in keeping with its telephony heritage. Telephone networks, by necessity, had their complexity within the network, since they were connecting dumb end-system devices such as rotary telephones. (For those too young to know, a rotary phone is an analog telephone with no buttons—only a dial.)

The Internet as a datagram network, on the other hand, grew out of the need to connect computers together. Given more sophisticated end-system devices, the Internet architects chose to make the network-layer service model as simple as possible. Additional functionality (for example, in-order delivery, reliable data transfer, congestion control, and DNS name resolution) is then implemented at a higher layer, in the end systems.

This inverts the model of the telephone network, with some interesting consequences:

• Since the resulting Internet network-layer service model makes minimal (no!) service guarantees, it imposes minimal requirements on the network layer.

This makes it easier to interconnect networks that use very different link-layer technologies (for example, satellite, Ethernet, fiber, or radio) that have very different transmission rates and loss characteristics.

• Applications such as e-mail, the Web, and even some network infrastructure services such as the DNS are implemented in hosts (servers) at the network edge.

The ability to add a new service simply by attaching a host to the network and defining a new application-layer protocol (such as HTTP) has allowed new Internet applications such as the Web to be deployed in a remarkably short period of time.

Network devices, also known as networking hardware, are physical devices that allow hardware on a computer network to communicate and interact with one another. For example Repeater, Hub, Bridge, Switch, Routers, Gateway, and NIC, etc.

What is Router?

A Router is a networking device that forwards data packets between computer networks. One or more packet-switched networks or sub networks can be connected using a router. By sending data packets to their intended IP addresses, it manages traffic between different networks and permits several devices to share an Internet connection.

Router

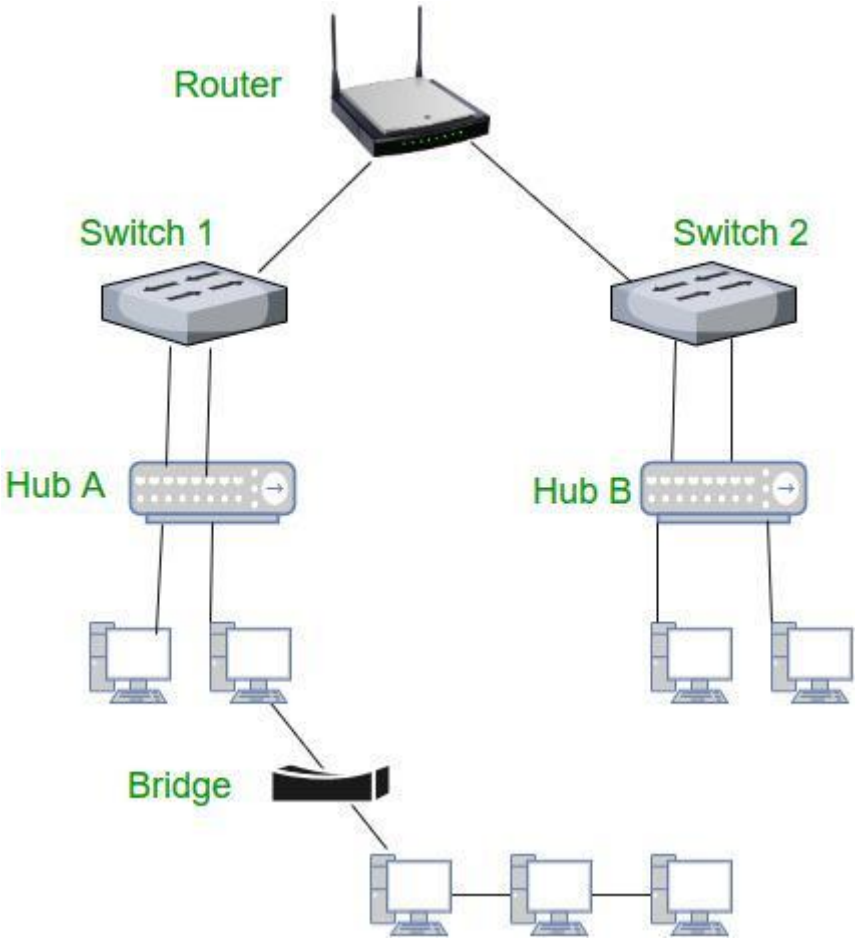
How Does Router Work?

A router determines a packet's future path by examining the destination IP address of the header and comparing it to the routing database. The list of routing tables outlines how to send the data to a specific network location. They use a set of rules to determine the most effective way to transmit the data to the specified IP address.

To enable communication between other devices and the internet, routers utilize a modem, such as a cable, fiber, or DSL modem. Most routers include many ports that can connect a variety of devices to the internet simultaneously. In order to decide where to deliver data and where traffic is coming from, it needs routing tables.

A routing table primarily specifies the router's default path. As a result, it might not determine the optimum path to forward the data for a particular packet. For instance, the office router directs all networks to its internet service provider through a single default channel.

Static and dynamic tables come in two varieties in the router. The dynamic routing tables are automatically updated by dynamic routers based on network activity, whereas the static routing tables are configured manually.



Router Network

Types of Router

There are several types of routers available in the market. Some of them are mentioned below:

1. **Broadband Routers:** These are one of the important kinds of routers. It is used to do different types of things. it is used to connect computers or it is also used to connect to the internet.
2. **Wireless routers:** These routers are used to create a wireless signal in your office or home. Wireless routers receive data packets over wired broadband, convert the packets written in binary code into radio signals that are picked up by electronic devices, and then convert them back into previous packets.
3. **Edge Routers**: As the name indicates, these are located at the edges usually connected to an Internet Service Provider, and distribute packets across multiple packets.
4. **Core Routers:** Core routers distribute packets within the same network. The main task is to carry heavy data transfers.

Functions of Router

The router performs two major functions:

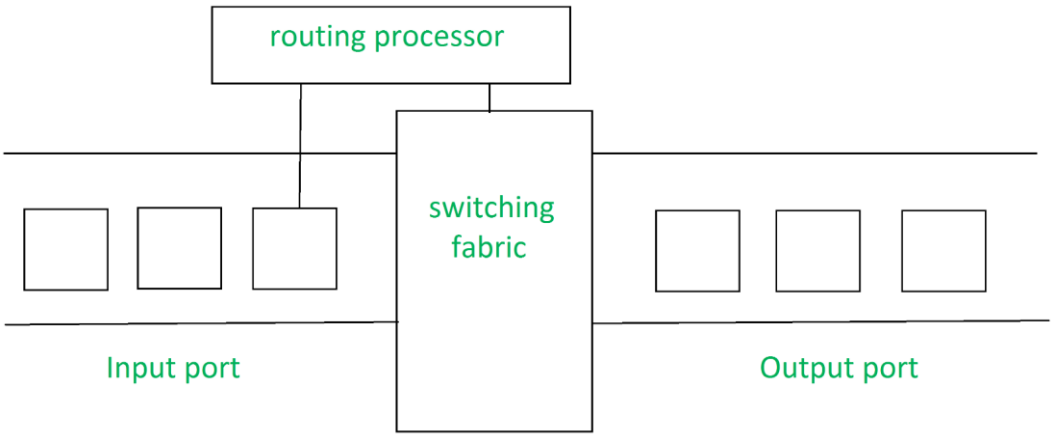
1. **Forwarding:** The router receives the packets from its input ports, checks its header, performs some basic functions like checking checksum, and then looks up to the

routing table to find the appropriate output port to dump the packets onto, and forwards the packets onto that output port.

2. **Routing:** Routing is the process by which the router ascertains what is the best path for the packet to reach the destination, It maintains a routing table that is made using different algorithms by the router only.
3. **Network Address Translation (NAT):** Routers use NAT to translate between different IP address ranges. This allows devices on a private network to access the internet using a single public IP address.
4. **Security:** Routers can be configured with firewalls and other security features to protect the network from unauthorized access, malware, and other threats.
5. **Quality of Service (QoS):** Routers can prioritize network traffic based on the type of data being transmitted. This ensures that critical applications and services receive adequate bandwidth and are not affected by lower-priority traffic.
6. **Virtual Private Network (VPN) connectivity:** Routers can be configured to allow remote users to connect securely to the network using a VPN.
7. **Bandwidth management:** Routers can be used to manage network bandwidth by controlling the amount of data that is allowed to flow through the network. This can prevent network congestion and ensure that critical applications and services receive adequate bandwidth.
8. **Monitoring and diagnostics**: Routers can be configured to monitor network traffic and provide diagnostics information in the event of network failures or other issues. This allows network administrators to quickly identify and resolve problems.

Architecture of Router

A generic router consists of the following components:



The architecture of a Router

1. **Input Port:** This is the interface by which packets are admitted into the router, it performs several key functions as terminating the physical link at the router, this is done by the leftmost part in the below diagram, and the middle part does the work of interoperating with the link-layer like decapsulation, in the last part of the input port the forwarding table is looked up and is used to determine the appropriate output port based on the destination address.
2. **Switching Fabric:** This is the heart of the Router, It connects the input ports with the output ports. It is kind of a network inside a networking device. The switching fabric can be implemented in several ways some of the prominent ones are:

- **Switching via memory:** In this, we have a processor which copies the packet from input ports and sends it to the appropriate output port. It works as a traditional CPU with input and output ports acting as input and output devices.
- **Switching via bus:** In this implementation, we have a bus that connects all the input ports to all the output ports. On receiving a packet and determining which output port it must be delivered to, the input port puts a particular token on the packet and transfers it to the bus. All output ports can see the packets but they will be delivered to the output port whose token has been put in, the token is then scraped off by that output port and the packet is forwarded
- **Switching via interconnection network:** This is a more sophisticated network, here instead of a single bus we use a 2N bus to connect n input ports to n output ports.

3. **Output Port:** This is the segment from which packets are transmitted out of the router. The output port looks at its queuing buffers (when more than one packets have to be transmitted through the same output port queuing buffers are formed) and takes packets, does link layer functions, and finally transmits the packets to an outgoing link.
4. **Routing Processor:** It executes the routing protocols, and it works like a traditional CPU. It employs various routing algorithms like the link-state algorithm, distance-vector algorithm, etc. to prepare the forwarding table, which is looked up to determine the route and the output port.

Advantages of Router

1. **Easier Connection:** Sharing a single network connection among numerous machines is the router's main job. This enables numerous people to connect to the internet, boosting total productivity. In addition, routers have connections between various media and network designs.
2. **Security:** Undoubtedly, installing a router is the first step in securing a network connection. Because using a modem to connect directly to the internet exposes your computer to several security risks. So that the environment is somewhat secure, routers can be utilized as an intermediary between two networks. While not a firewall or antivirus replacement.
3. **NAT Usage:** Routers use Network Address Translation (NAT) to map multiple private IP addresses into one public IP address. This allows for a better Internet connection and information flow between all devices connected to the network.
4. **Supports Dynamic Routing:** The router employs dynamic routing strategies to aid in network communication. The internet work's optimum path is chosen through dynamic routing. Additionally, it creates collision and broadcast domains. Overall, this can lessen network traffic.
5. **Filtering of Packets:** Switching between packets and filtering packets are two more router services. A collection of filtering rules are used by routers to filter the network. The packets are either allowed or passed through.

Disadvantages of Router

1. **Slower:** Routers analyze multiple layers of information, from the physical layer to the network layer, which slows down connections. The same issue can also be encountered when multiple devices are connected to these network devices, causing "connection waiting".
2. **High Cost:** They are more expensive than some other tools for systems administration. This includes security, extension, and the focal point. As a result, routers are typically not the greatest option for issues.

3. **Need for configuration:** The router must be properly configured to work properly. In general, the more complex the intended use, the more configuration is required. This requires professional installation, which can add to the cost of buying a router.
4. **Quality Issues:** The time transitions are not always accurate. Even yet, some modern devices use the 2.4GHz band, which is frequently deactivated. These kinds of separations are frequently possible for those who live in apartments and condominiums.
5. **Bandwidth shortages:** Dynamic routing techniques used by routers to support connections tend to cause network overhead, consuming a lot of bandwidth. This leads to a bandwidth shortage that significantly slows down the internet connection between connected devices.

Types of networking planes

Both traditional and cloud-based networks are based on the following three dimensions of network planes:

1. The **control plane** decides all the functions and processes that determine which path data must take by using routing protocols.
2. The **data plane** controls all the functions and processes that determine how to forward packets from one interface to another.
3. The **management plane** controls all the functions related to the control and monitoring of devices.

1. Control Plane :

In Routing control plane refers to the all functions and processes that determine which path to use to send the packet or frame. Control plane is responsible for populating the routing table, drawing network topology, forwarding table and hence enabling the data plane functions. Means here the router makes its decision. In a single line it can be said that it is responsible for How packets should be forwarded.

2. Data Plane :

In Routing data plane refers to all the functions and processes that forward packets/frames from one interface to another based on control plane logic. Routing table, forwarding table and the routing logic constitute the data plane function. Data plane packet goes through the router and incoming and outgoing of frames are done based on control plane logic. Means in single line it can be said that it is responsible for moving packets from source to destination. It is also called as Forwarding plane.

3. Management Plane:

The management plane is responsible for managing and monitoring the network's operations. In this plane, we can configure devices, monitor the device's performance, and ensure that the network operates efficiently. Moreover, the management plane is responsible for other tasks such as software updates, security, and monitoring.

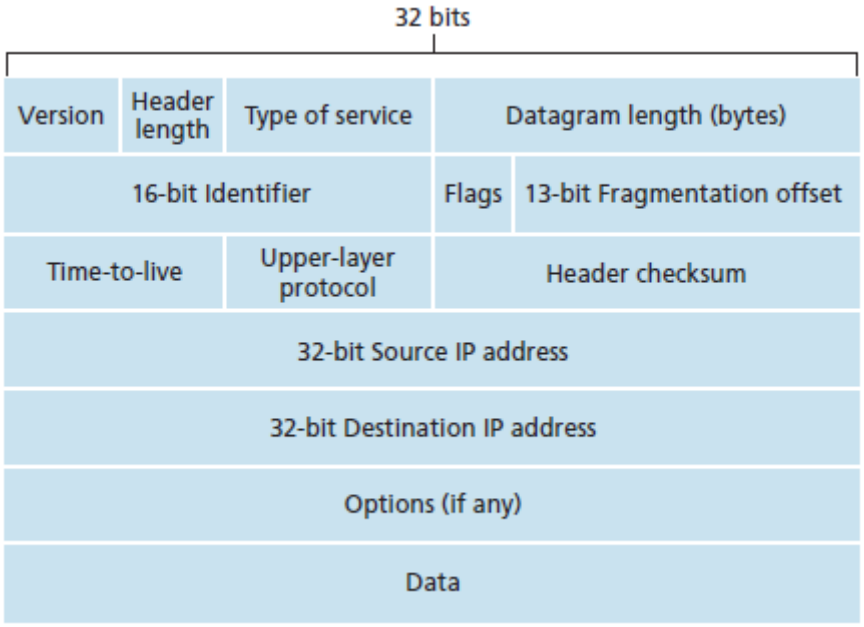
S.No.	CONTROL PLANE	DATA PLANE
01.	Control plane refers to the all functions and processes that determine which path to use to send the packet or frame.	Data plane refers to all the functions and processes that forward packets/frames from one interface to another based on control plane logic.
02.	It is responsible for building and maintaining the IP routing table.	It is responsible for forwarding actual IP packet.
03.	Control plane responsible about how packets should be forwarded.	Data plane responsible for moving packets from source to destination.
04.	Control plane performs its task independently.	Data plane performs its task depending on Control plane.
05.	In general we can say in control plane it is learned what and how it can be done.	In general we can say in data plane the actual task is performed based on what is learned.
06.	Control plane packets are processed by router to update the routing table.	The forwarding plane/data plane forwards the packets based on the built logic of control plane.
07.	It includes Spanning Tree Protocol (STP), Address Resolution Protocol (ARP), Routing Information Protocol (RIP), Dynamic Host Configuration Protocol (DHCP) etc.	It includes decrementing Time To Live (TTL), recomputing IP header checksum etc.
08.	Control plane packets are locally originated by the router itself.	Data plane packets go through the router.
09.	Control plane acts as a decision maker in data forwarding.	Data plane acts as a decision implementer in data forwarding.
10.	Routing is performed in the control plane.	Switching is performed in the data plane.

IPv4:

- The Internet Protocol version 4 (IPv4) is a delivery mechanism used by the TCP/IP protocols. IPv4 is an unreliable and connectionless datagram protocol- which operates on a best effort delivery model. The term best-effort means that IPv4 provides no error control or flow control.
- IPv4 uses 32-bit (4 bytes) addressing, which gives 2^{32} addresses.
- IPv4 addresses are written in the dot-decimal notation, which comprises of four octets of the address expressed individually in decimal and separated by periods, for instance, 192.168.1.5.

IPv4 Datagram Header

IPv4 is 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.



- **Version (VER):** This 4-bit field defines the version of the IPv4 protocol. Currently, the version is 4. However, version 6 (or IPng) may totally replace version 4 in the future.

- **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.

- **Type of service:** Low Delay, High Throughput, Reliability (8 bits)

- **Total Length:** Length of header + Data (16 bits), which has a minimum value 20 bytes and the maximum is 65,535 bytes

- **Identification:** Unique Packet Id for identifying the group of fragments of a single IP datagram (16 bits)

- **Flags:** 3 flags of 1 bit each: reserved bit (must be zero), do not fragment flag, more fragments flag (same order)

- **Fragment Offset:** Specified in terms of number of 8 bytes, which has the maximum value of 65,528 bytes

- **Time to live:** Datagram's lifetime (8 bits), It prevents the datagram to loop in the network

- **Protocol:** Name of the protocol to which the data is to be passed (8 bits)

- **Header Checksum:** 16 bits header checksum for checking errors in the datagram header

- **Source IP address:** 32 bits IP address of the sender

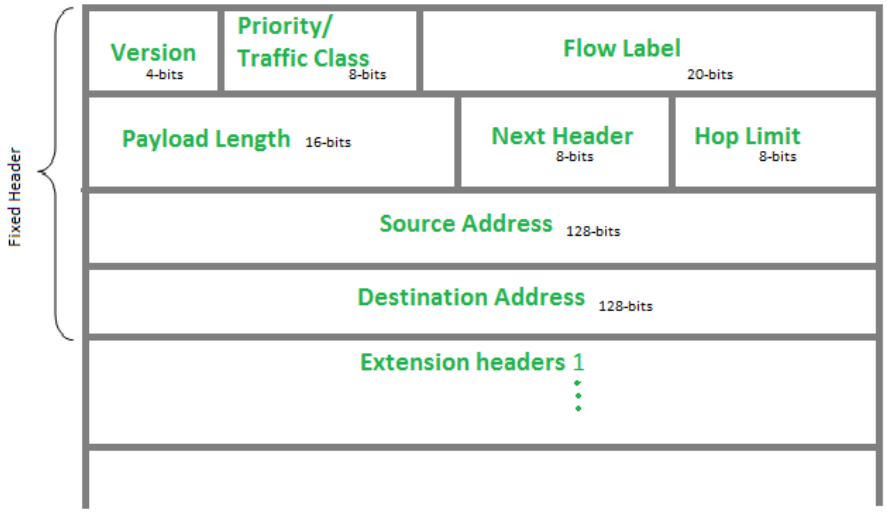
- **Destination IP address:** 32 bits IP address of the receiver

- **Option:** Optional information such as source route. Due to the presence of options, the size of the datagram header can be of variable length (20 bytes to 60 bytes).

IPv6 Header

The wonder of IPv6 lies in its header. An IPv6 address is 4 times larger than IPv4, but surprisingly, the header of an IPv6 address is only 2 times larger than that of IPv4. IPv6 headers have one Fixed Header and zero or more Optional (Extension) Headers. All the necessary information that is essential for a router is kept in the Fixed Header. The Extension Header contains optional information that helps routers to understand how to handle a packet/flow.

Fixed Header



IPv6 fixed header is 40 bytes long and contains the following information.

S.N. Field & Description

1 **Version (4-bits):** It represents the version of Internet Protocol.

2 **Traffic Class (8-bits):** These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router Known what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN). Priority assignment of Congestion controlled traffic :

Priority	Meaning
0	No Specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

3 **Flow Label (20-bits):** This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. This field helps avoid re-ordering of data packets. It is designed for streaming/real-time media.

4 **Payload Length (16-bits):** This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. With 16 bits, up to 65535 bytes can be indicated; but if the Extension Headers contain Hop-by-Hop Extension Header, then the payload may exceed 65535 bytes and this field is set to 0.

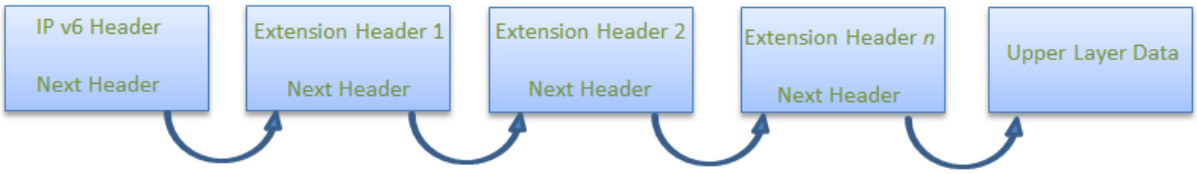
5 **Next Header (8-bits):** This field is used to indicate either the type of Extension Header, or if the Extension Header is not present then it indicates the Upper Layer PDU. The values for the type of Upper Layer PDU are same as IPv4's.

6 **Hop Limit (8-bits):** This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.

7 **Source Address (128-bits):** This field indicates the address of originator of the packet.

8 **Destination Address (128-bits):** This field provides the address of intended recipient of the packet.

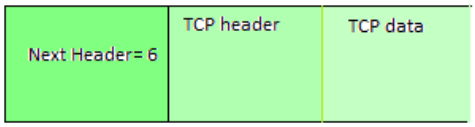
Extension Headers: In order to rectify the limitations of the *IPv4 Option Field*, Extension Headers are introduced in IP version 6. The extension header mechanism is a very important part of the IPv6 architecture. The next Header field of IPv6 fixed header points to the first Extension Header and this first extension header points to the second extension header and so on.



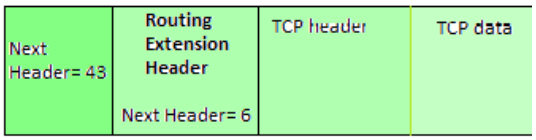
IPv6 packet may contain zero, one or more extension headers but these should be present in their recommended order:

Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

Example: *TCP is used in IPv6 packet*



Example2:



Rule: Hop-by-Hop options header(if present) should always be placed after the IPv6 base header.

Ext. Header	Description
Hop-by-Hop Options	Examined by all devices on the path
Destination Options (with routing options)	Examined by destination of the packet
Routing Header	Methods to take routing decision
Fragment Header	Contains parameters of fragmented datagram done by source
Authentication Header	verify authenticity
Encapsulating Security Payload	Carries Encrypted data

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP) is a network layer protocol used to diagnose communication errors by performing an error control mechanism. Since IP does not have an inbuilt mechanism for sending error and control messages. It depends on Internet Control Message Protocol(ICMP) to provide error control.

ICMP is used for reporting errors and management queries. It is a supporting protocol and is used by network devices like routers for sending error messages and operations information. For example, the requested service is not available or a host or router could not be reached.

Messages

The ICMP messages are usually divided into two categories:

ICMP messages

Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

- **Error-reporting messages**

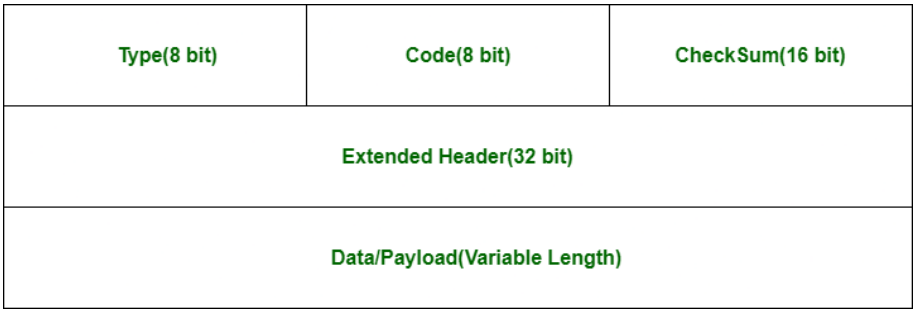
The error-reporting message means that the router encounters a problem when it processes an IP packet then it reports a message.

- **Query messages**

The query messages are those messages that help the host to get the specific information of another host. For example, suppose there are a client and a server, and the client wants to know whether the server is live or not, then it sends the ICMP message to the server.

ICMP Packet Format

ICMP header comes after IPv4 and IPv6 packet header.

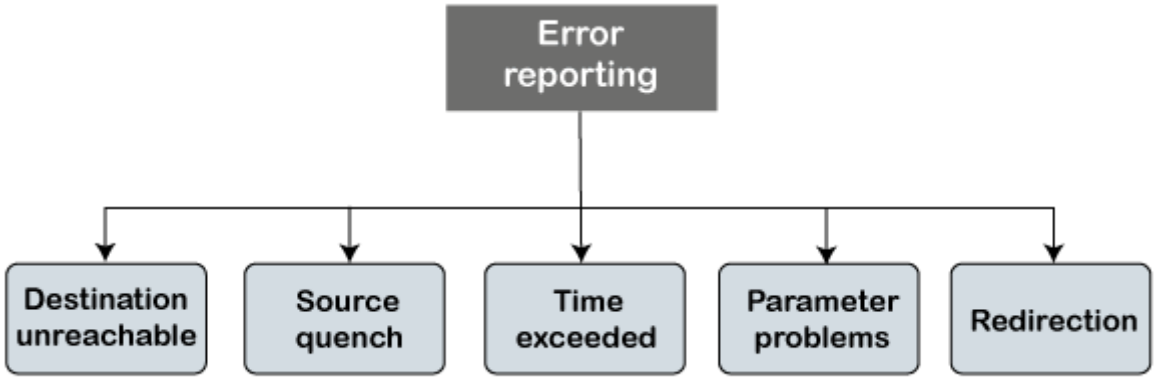


ICMPv4 Packet Format

In the ICMP packet format, the first 32 bits of the packet contain three fields:

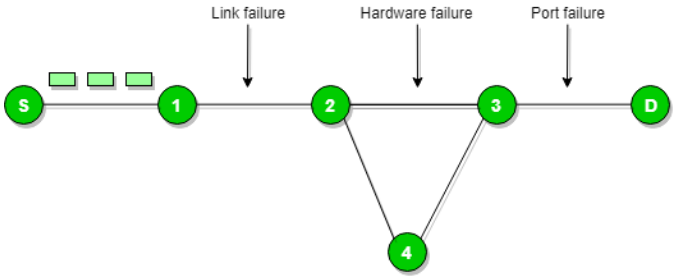
- Type: It is an 8-bit field. It defines the ICMP message type. The values range from 0 to 127 are defined for ICMPv6, and the values from 128 to 255 are the informational messages.
- Code: It is an 8-bit field that defines the subtype of the ICMP message
- Checksum: It is a 16-bit field to detect whether the error exists in the message or not.

Types of Error Reporting messages
The error reporting messages are broadly classified into the following categories:



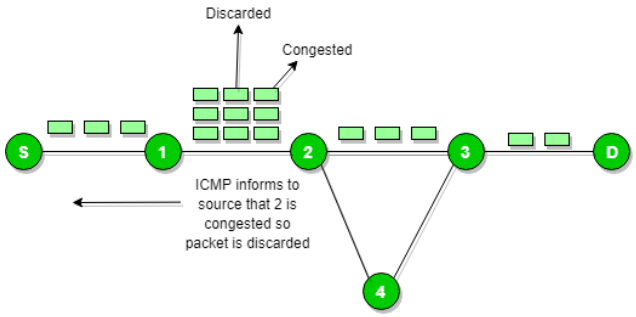
Destination Un-reachable
The destination is unreachable and is generated by the host or its inbound gateway to inform the client that the destination is unreachable for some reason.

There is no necessary condition that only the router gives the ICMP error message time the destination host sends an ICMP error message when any type of failure (link failure, hardware failure, port failure, etc) happens in the network.



Source Quench Message

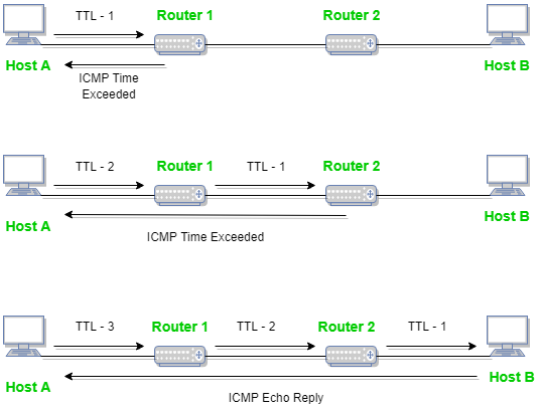
A source quench message is a request to decrease the traffic rate for messages sent to the host destination) or we can say when receiving host detects that the rate of sending packets (traffic rate) to it is too fast it sends the source quench message to the source to slow the pace down so that no packet can be lost.



ICMP will take the source IP from the discarded packet and inform the source by sending a source quench message. The source will reduce the speed of transmission so that router will be free from congestion.

Time Exceeded Message

When some fragments are lost in a network then the holding fragment by the router will be dropped then ICMP will take the source IP from the discarded packet and informs the source, of discarded datagram due to the time to live field reaching zero, by sending the time exceeded message.

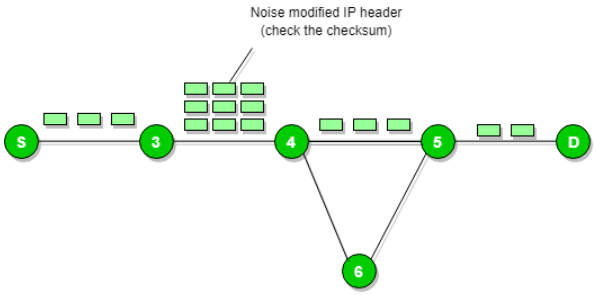


Parameter Problem

Whenever packets come to the router then the calculated header checksum should be equal to the received header checksum then only the packet is accepted by the router.

If there is a mismatch packet will be dropped by the router.

ICMP will take the source IP from the discarded packet and inform the source by sending a parameter problem message.



Redirection Message

Redirect requests data packets are sent on an alternate route. The message informs a host to update its routing information (to send packets on an alternate route).

Whenever a packet is forwarded in the wrong direction later it is re-directed in a current direction then ICMP will send a re-directed message.

ICMP Query Messages

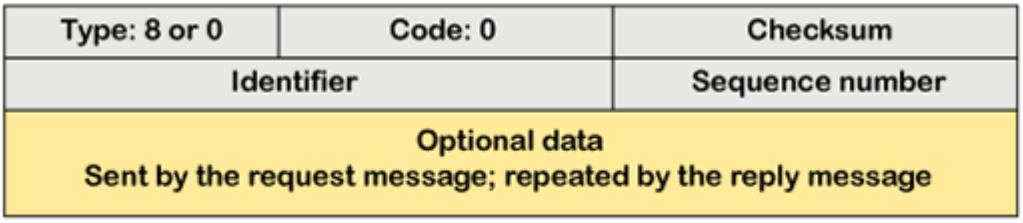
The ICMP Query message is used for error handling or debugging the internet. This message is commonly used to ping a message.

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

Type 8: Echo request
Type 0: Echo reply

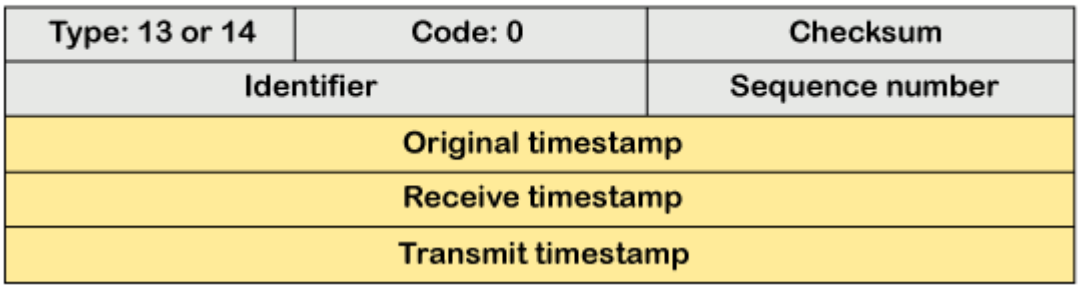


Timestamp-request and timestamp-reply message

The timestamp-request and timestamp-reply messages are also a type of query messages. Suppose the computer A wants to know the time on computer B, so it sends the timestamp-request message to computer B. The computer B responds with a timestamp-reply message.

Message format of timestamp-request and timestamp-reply

Type 13: request
Type 14: reply



Difference between IPv4 and IPv6

IPv4	IPv6
IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
In IPv4 end to end, connection integrity is Unachievable	In IPv6 end-to-end, connection integrity is Achievable
It can generate 4.29×109 address space	The address space of IPv6 is quite large it can produce 3.4×1038 address space
The Security feature is dependent on the application	IPSEC is an inbuilt security feature in the IPv6 protocol
Address representation of IPv4 is in decimal	Address Representation of IPv6 is in hexadecimal
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation is performed only by the sender
In IPv4 Packet flow identification is not available	In IPv6 packet flow identification are Available and uses the flow label field in the header
In IPv4 checksum field is available	In IPv6 checksum field is not available
It has a broadcast Message Transmission Scheme	In IPv6 multicast and anycast message transmission scheme is available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided

IPv4	IPv6
IPv4 has a header of 20-60 bytes.	IPv6 has a header of 40 bytes fixed
IPv4 can be converted to IPv6	Not all IPv6 can be converted to IPv4
IPv4 consists of 4 fields which are separated by addresses dot (.)	IPv6 consists of 8 fields, which are separated by a colon (:)
IPv4's IP addresses are divided into five different classes. Class A , Class B, Class C, Class D , Class E.	IPv6 does not have any classes of the IP address.
IPv4 supports VLSM (Variable Length subnet mask).	IPv6 does not support VLSM.
Example of IPv4: 66.94.29.13	Example of IPv6: 2001:0000:3238:DFE1:0063:0000:0000:FEFB

Introduction to Transport Layer

The transport Layer is the second layer in the TCP/IP model and the fourth layer in the OSI model. It is an end-to-end layer used to deliver messages to a host. It is termed an end-to-end layer because it provides a point-to-point connection rather than hop-to-hop, between the source host and destination host to deliver the services reliably.

The transport layer takes services from the Application layer and provides services to the Network layer

Services Provided by the Transport Layer

The Transport Layer offers a range of services to ensure seamless communication and data exchange between applications.

- **End-to-end delivery:** Ensures data reaches its intended destination accurately and efficiently.
- **Reliable Delivery:** TCP's reliable delivery ensures that data is received at the destination without errors and in the correct order.
- **Error Control:** TCP performs error-checking during data transmission and uses checksums to detect any errors. If mistakes are detected, TCP requests the retransmission of the affected data segments.
- **Sequence Control:** TCP numbers each segment during transmission, enabling the receiving end to reassemble the data in the correct order. This ensures that data is presented to the application in the same sequence it was sent.
- **Flow Control:** TCP employs flow control mechanisms to manage data flow between the sender and receiver. This prevents overwhelming the recipient with too much data, ensuring the receiver can handle the data at its own pace.
- **Congestion Control:** TCP also manages network congestion to prevent data loss and ensure optimal network performance. It dynamically adjusts the transmission rate based on network conditions.
- **Multiplexing:** The Transport Layer can support multiple applications on a single device, ensuring that data from different applications is correctly directed to the appropriate destination.
- **Addressing:** Provides logical addressing through port numbers to establish communication channels between applications.

- **Loss control:** Manages data loss during transmission and requests a retransmission if necessary.
- **Duplication control:** Avoids duplication of data packets to prevent unnecessary overhead.

Relation between Transport and Network Layer

The transport and network layers work hand in hand to ensure smooth data delivery. The transport layer utilizes the services of the network layer to route packets to their destination based on logical addressing.

Aspect	Transport Layer	Network Layer
Primary Function	End-to-end delivery of data packets	Logical addressing and routing of packets
Layer Location in OSI Model	Fourth layer	Third layer
Protocol Examples	TCP, UDP	IP, ICMP
Services Provided	Reliable delivery, error control, flow control	Logical addressing, routing, packet forwarding

Responsibility	Ensures data delivery between applications	Manages data transmission between networks

Transport Layer in the Internet

The transport layer is the fourth layer in the OSI model. Its primary responsibility is to directly provide logical communication between application processes running on various hosts. This allows the application processes to transmit messages to one another even when they are not physically connected.

The network routers do not implement the transport layer protocols, but the end systems do. A computer network gives network applications access to several protocols. The transport layer protocols, TCP and UDP, offer a unique set of services to the network layer.

Transport Layer in OSI Model

This layer bridges the upper layers (application layer, presentation layer, and session layer) and the lower layers (network layer and data link layer). Its primary function is to provide a reliable and orderly data delivery mechanism, irrespective of the underlying network structure.

Transport Layer Protocols

The Transport Layer is responsible for segmenting and reassembling data received from the upper layers into manageable chunks called "segments" before transmitting them to the network layer for further routing and delivery. Two prominent transport layer protocols are:

- **Transmission Control Protocol (TCP):** TCP is a connection-oriented protocol that ensures reliable and error-free data transmission. It establishes a virtual circuit between the sender and receiver before data exchange, providing sequencing, flow control, and acknowledgement mechanisms. TCP guarantees that data packets reach their destination in the correct order and requests the retransmission of any lost or corrupted packets.
- **User Datagram Protocol (UDP):** UDP is a connectionless protocol that offers minimal overhead. It does not establish a dedicated connection before data transmission and lacks the reliability features of TCP. While UDP is faster, it does not guarantee delivery

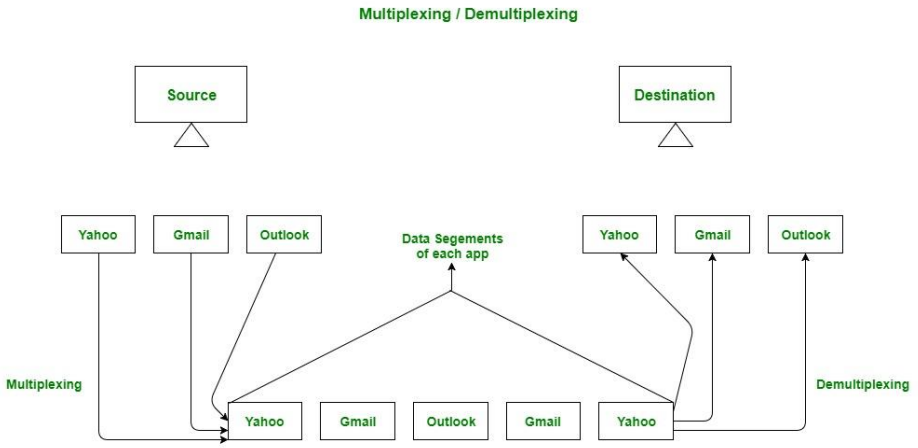
or packet sequencing. It is often used for applications where real-time and low-latency communication is crucial, such as video streaming and online gaming.

Multiplexing and Demultiplexing in Transport Layer

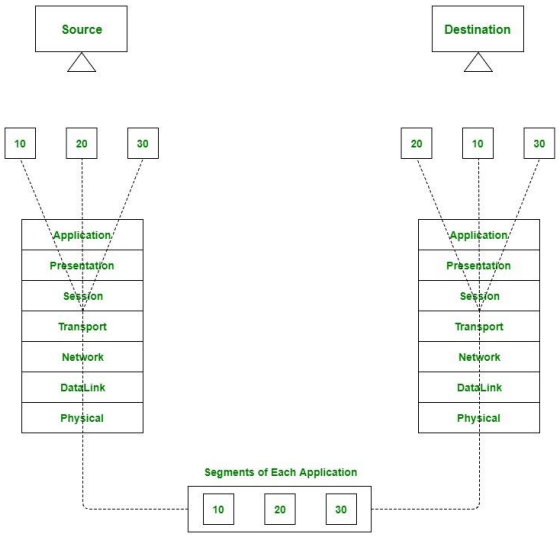
Multiplexing and Demultiplexing services are provided in almost every protocol architecture ever designed. UDP and TCP perform the demultiplexing and multiplexing jobs by including two special fields in the segment headers: the source port number field and the destination port number field.

Multiplexing –
Gathering data from multiple application processes of the sender, enveloping that data with a header, and sending them as a whole to the intended receiver is called multiplexing.

Demultiplexing –
Delivering received segments at the receiver side to the correct app layer processes is called demultiplexing.



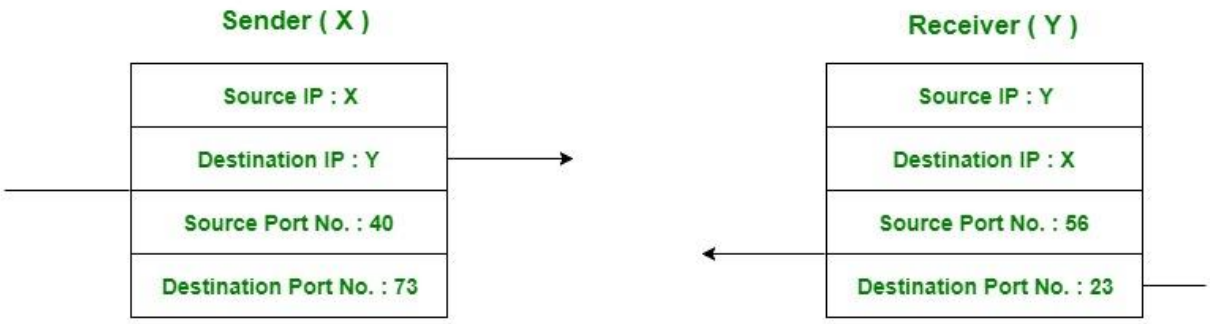
Multiplexing and demultiplexing are the services facilitated by the transport layer of the OSI model.



There are two types of multiplexing and Demultiplexing:

1. Connectionless Multiplexing and Demultiplexing
2. Connection-Oriented Multiplexing and Demultiplexing

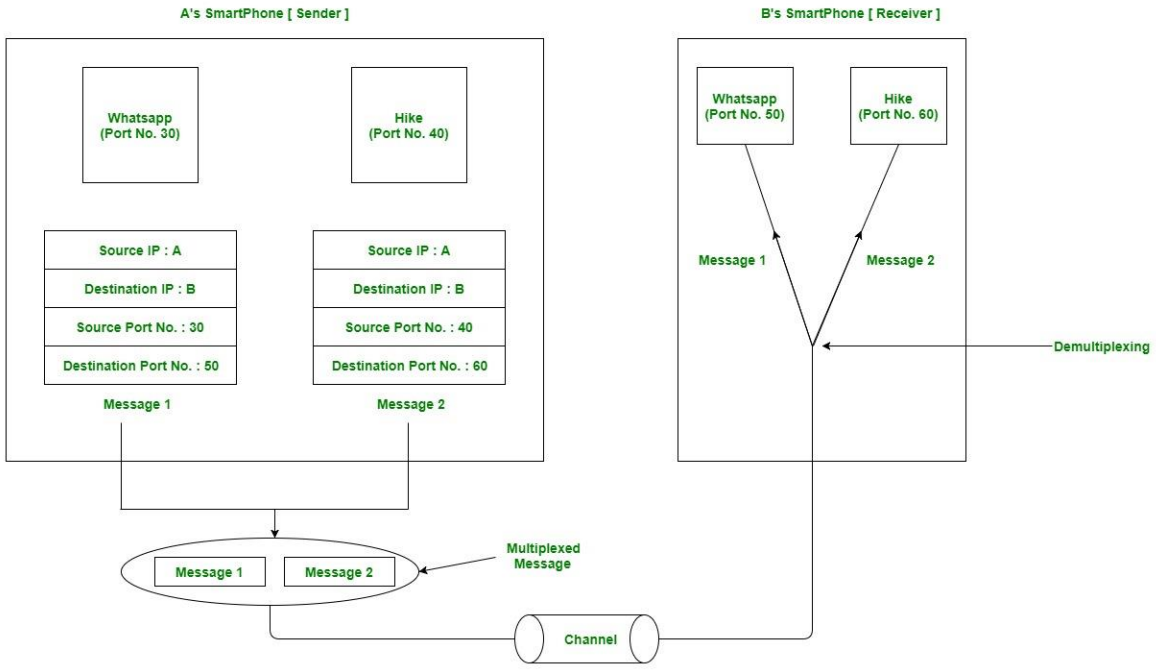
How Multiplexing and Demultiplexing is done For sending data from an application on the sender side to an application at the destination side, the sender must know the IP address of the destination and port number of the application (at the destination side) to which he wants to transfer the data. Block diagram is shown below :



Let us consider two messaging apps that are widely used nowadays viz. Hike and WhatsApp. Suppose A is the sender and B is the receiver. Both sender and receiver have these applications installed in their system (say smartphone). Suppose A wants to send messages to B in

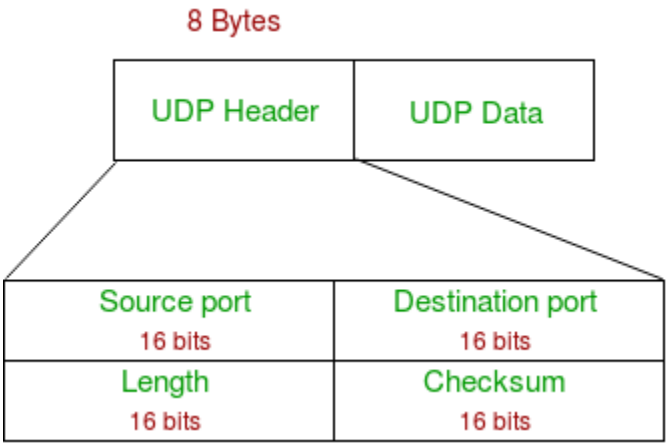
WhatsApp and hike both. In order to do so, A must mention the IP address of B and destination port number of the WhatsApp while sending the message through the WhatsApp application. Similarly, for the latter case, A must mention the IP address of B and the destination port number of the hike while sending the message.

Now the messages from both the apps will be wrapped up along with appropriate headers(viz. source IP address, destination IP address, source port no, destination port number) and sent as a single message to the receiver. This process is called multiplexing. At the destination, the received message is unwrapped and constituent messages (viz messages from a hike and WhatsApp application) are sent to the appropriate application by looking to the destination the port number. This process is called demultiplexing. Similarly, B can also transfer the messages to A.



User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection prior to data transfer. The UDP helps to establish low-latency and loss-tolerating connections establish over the network.The UDP enables process to process communication.
Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of the Internet services; provides assured delivery, reliability, and much more but all these services cost us additional overhead and latency. Here, UDP comes into the picture. For real-time services like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also saves bandwidth. User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

UDP Header –
UDP header is an 8-bytes fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. The first 8 Bytes contains all necessary header information and the remaining part consist of data. UDP port number fields are each 16 bits long, therefore the range for port numbers is defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.



1. Source Port: Source Port is a 2 Byte long field used to identify the port number of the source.
2. Destination Port: It is a 2 Byte long field, used to identify the port of the destined packet.
3. Length: Length is the length of UDP including the header and the data. It is a 16-bits field.
4. Checksum: Checksum is 2 Bytes long field. It is the 16-bit one's complement of the one's complement sum of the UDP header, the pseudo-header of information from the IP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Notes – Unlike TCP, the Checksum calculation is not mandatory in UDP. No Error control or flow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting. Also UDP provides port numbers so that is can differentiate between users requests.

Applications of UDP:

- Used for simple request-response communication when the size of data is less and hence there is lesser concern about flow and error control.
- It is a suitable protocol for multicasting as UDP supports packet switching.
- UDP is used for some routing update protocols like RIP(Routing Information Protocol).
- Normally used for real-time applications which can not tolerate uneven delays between sections of a received message.
- UDP is widely used in online gaming, where low latency and high-speed communication is essential for a good gaming experience. Game servers often send small, frequent packets of data to clients, and UDP is well suited for this type of communication as it is fast and lightweight.

Advantages of UDP:

1. Speed: UDP is faster than TCP because it does not have the overhead of establishing a connection and ensuring reliable data delivery.

2. Lower latency: Since there is no connection establishment, there is lower latency and faster response time.

3. Simplicity: UDP has a simpler protocol design than TCP, making it easier to implement and manage.

Disadvantages of UDP:

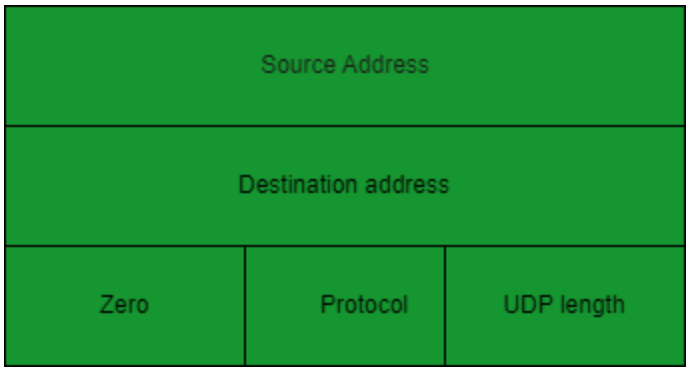
1. No reliability: UDP does not guarantee delivery of packets or order of delivery, which can lead to missing or duplicate data.

2. No congestion control: UDP does not have congestion control, which means that it can send packets at a rate that can cause network congestion.

3. No flow control: UDP does not have flow control, which means that it can overwhelm the receiver with packets that it cannot handle.

UDP PSEUDO HEADER:

- the purpose of using a pseudo-header is to verify that the UDP packet has reached its correct destination
- the correct destination consist of a specific machine and a specific protocol port number within that machine

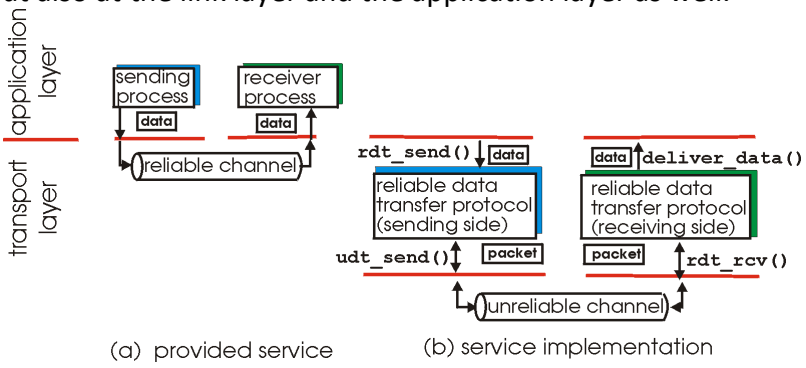


UDP pseudo header details:

- The UDP header itself specifies only protocol port number. Thus , to verify the destination UDP on the sending machine computes a checksum that covers the destination IP address as well as the UDP packet.
- At the ultimate destination, UDP software verifies the checksum using the destination IP address obtained from the header of the IP packet that carried the UDP message.
- If the checksum agrees, then it must be true that the packet has reached the intended destination host as well as the correct protocol port within that host.

Principles of Reliable Data Transfer

In this section, we consider the problem of reliable data transfer in a general context. This is appropriate since the problem of implementing reliable data transfer occurs not only at the transport layer, but also at the link layer and the application layer as well.



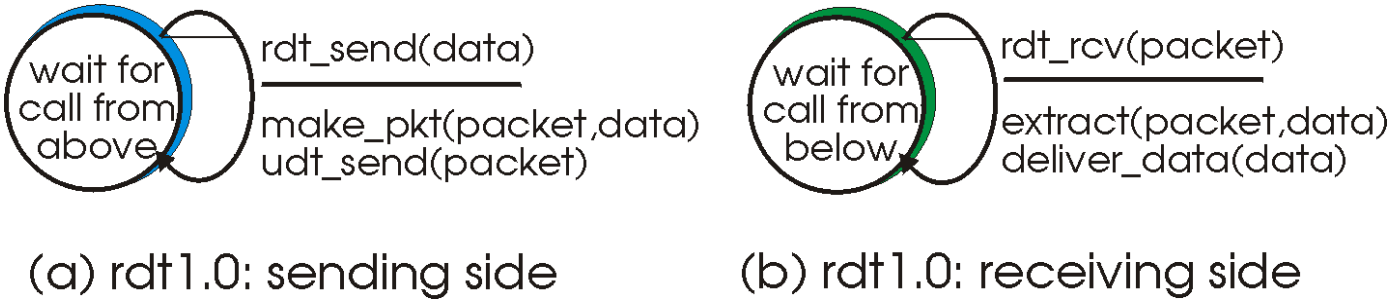
It is the responsibility of a **reliable data transfer protocol** to implement this service abstraction. This task is made difficult by the fact that layer *below* the reliable data transfer protocol may be unreliable. For example, TCP is a reliable data transfer protocol that is implemented on top of an unreliable (IP) end-end network layer. More generally, the layer beneath the two reliably-communicating endpoints might consist of a single physical link (e.g., as in the case of a link-level data transfer protocol) or a global internetwork (e.g., as in the case

of a transport-level protocol). For our purposes, however, we can view this lower layer simply as an unreliable point-to-point channel.
It will be passed the data to be delivered to the upper-layer at the receiving side. (Here rdt stands for ``**r**eliable **d**ata **t**ransfer'' protocol and _send indicates that the sending side of rdt is being called. The first step in developing any protocol is to choose a good name!) On the receiving side, rdt_rcv() will be called when a packet arrives from the receiving side of the channel. When the rdt protocol wants to deliver data to the upper-layer, it will do so by calling deliver_data(). In the following we use the terminology "packet" rather than "segment" for the protocol data unit.. Because the theory developed in this section applies to computer networks in general, and not just to the Internet transport layer, the generic term "packet" is perhaps more appropriate here.

Building a Reliable Data Transfer Protocol

Reliable Data Transfer over a Perfectly Reliable Channel: rdt1.0

We first consider the simplest case in which the underlying channel is completely reliable. The protocol itself, which we will call rdt1.0, is trivial. The **finite state machine** (FSM) definitions for the rdt1.0 sender and receiver are shown in Figure 3.4-2. The sender and receiver FSMs in Figure 3.4-2 each have just one state. The arrows in the FSM description indicate the transition of the protocol from one state to another. (Since each FSM in Figure 3.4-2 has just one state, a transition is necessarily from the one state back to itself; we'll see more complicated state diagrams shortly.). The event causing the transition is shown above the horizontal line labeling the transition, and the action(s) taken when the event occurs are shown below the horizontal line.
The sending side of rdt simply accepts data from the upper-layer via the rdt_send(data)event, puts the data into a packet (via the action make_pkt(packet,data)) and sends the packet into the channel. In practice, the rdt_send(data)event would result from a procedure call (e.g., to rdt_send()) by the upper layer application.
On the receiving side, rdt receives a packet from the underlying channel via the rdt_rcv(packet) event, removes the data from the packet (via the action extract(packet,data)) and passes the data up to the upper-layer. In practice, the rdt_rcv(packet)event would result from a procedure call (e.g., to rdt_rcv()) from the lower layer protocol.
In this simple protocol, there is no difference between a unit of data and a packet. Also, all packet flow is from the sender to receiver - with a perfectly reliable channel there is no need for the receiver side to provide any feedback to the sender since nothing can go wrong!



rdt1.0 - a protocol for a completely reliable channel

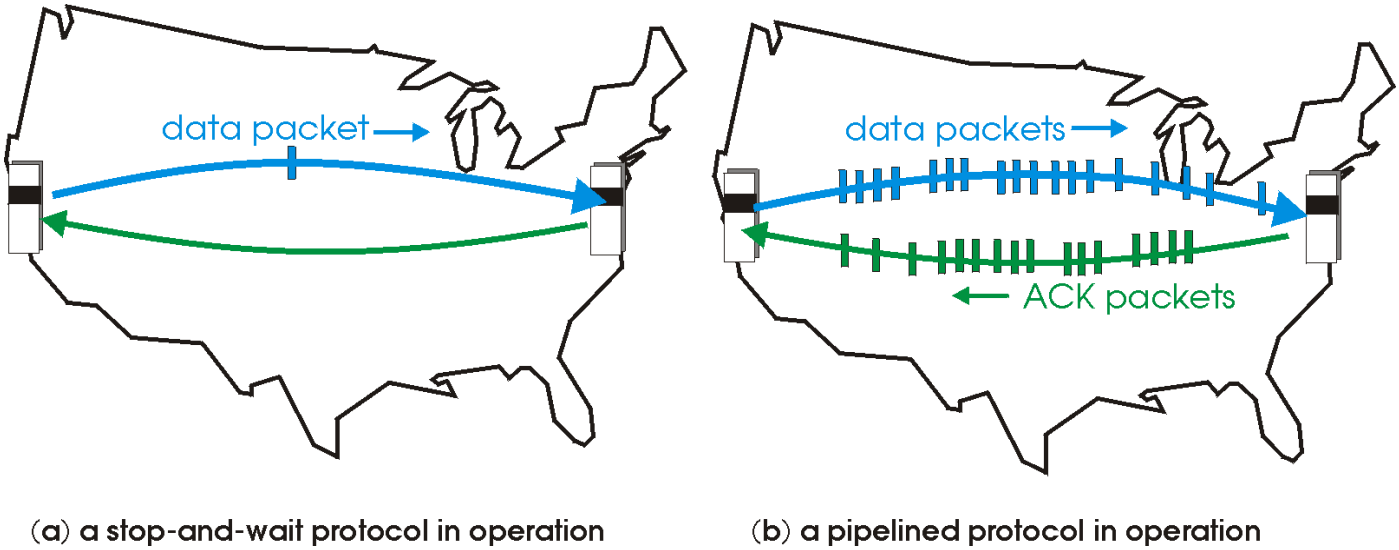
Pipelined Reliable Data Transfer Protocols

Protocol rdt3.0 is a functionally correct protocol, but it is unlikely that anyone would be happy with its performance, particularly in today's high speed networks. At the heart of rdt3.0's performance problem is the fact that it is a stop-and-wait protocol.

To appreciate the performance impact of this stop-and-wait behavior, consider an idealized case of two end hosts, one located on the west coast of the United States and the other located on the east cost. The speed-of-light propagation delay, T_{prop}, between these two end systems is approximately 15 milliseconds. Suppose that they are connected by a channel with a capacity, *C,* of 1 Gigabit (10**9 bits) per second. With a packet size, *SP*, of 1K bytes per packet including both header fields and data, the time needed to actually transmit the packet into the 1Gbps link is

T_{trans} *= SP/C = (8 Kbits/packet)/ (10**9 bits/sec) = 8 microseconds*

With our stop and wait protocol, if the sender begins sending the packet at *t* = 0, then at *t* = 8 microsecs the last bit enters the channel at the sender side. The packet then makes its 15 msec cross country journey, as depicted in Figure 3.4-10a, with the last bit of the packet emerging at the receiver at *t* = 15.008 msec.



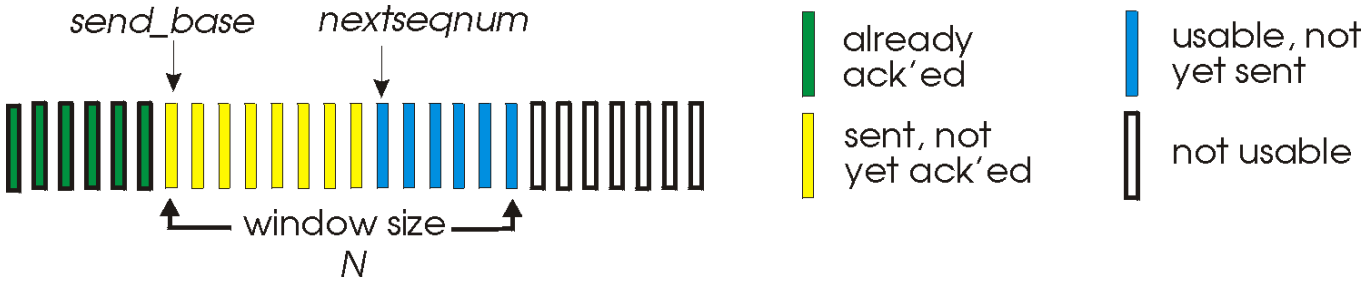
Stop-and-wait versus pipelined protocols

The solution to this particular performance problem is a simple one: rather than operate in a stop-and-wait manner, the sender is allowed to send multiple packets without waiting for acknowledgements, as shown in Figure 3.4-10(b). Since the many in-transit sender-to-receiver packets can be visualized as filling a pipeline, this technique is known as **pipelining**. Pipelining has several consequences for reliable data transfer protocols:

- The range of sequence numbers must be increased, since each in-transit packet (not counting retransmissions) must have a unique sequence number and there may be multiple, in-transit, unacknowledged packets.
- The sender and receiver-sides of the protocols may have to buffer more than one packet. Minimally, the sender will have to buffer packets that have been transmitted, but not yet acknowledged. Buffering of correctly-received packets may also be needed at the receiver, as discussed below.

The range of sequence numbers needed and the buffering requirements will depend on the manner in which a data transfer protocol responds to lost, corrupted, and overly delayed packets. Two basic approaches towards pipelined error recovery can be identified: **Go-Back-N** and **selective repeat.**

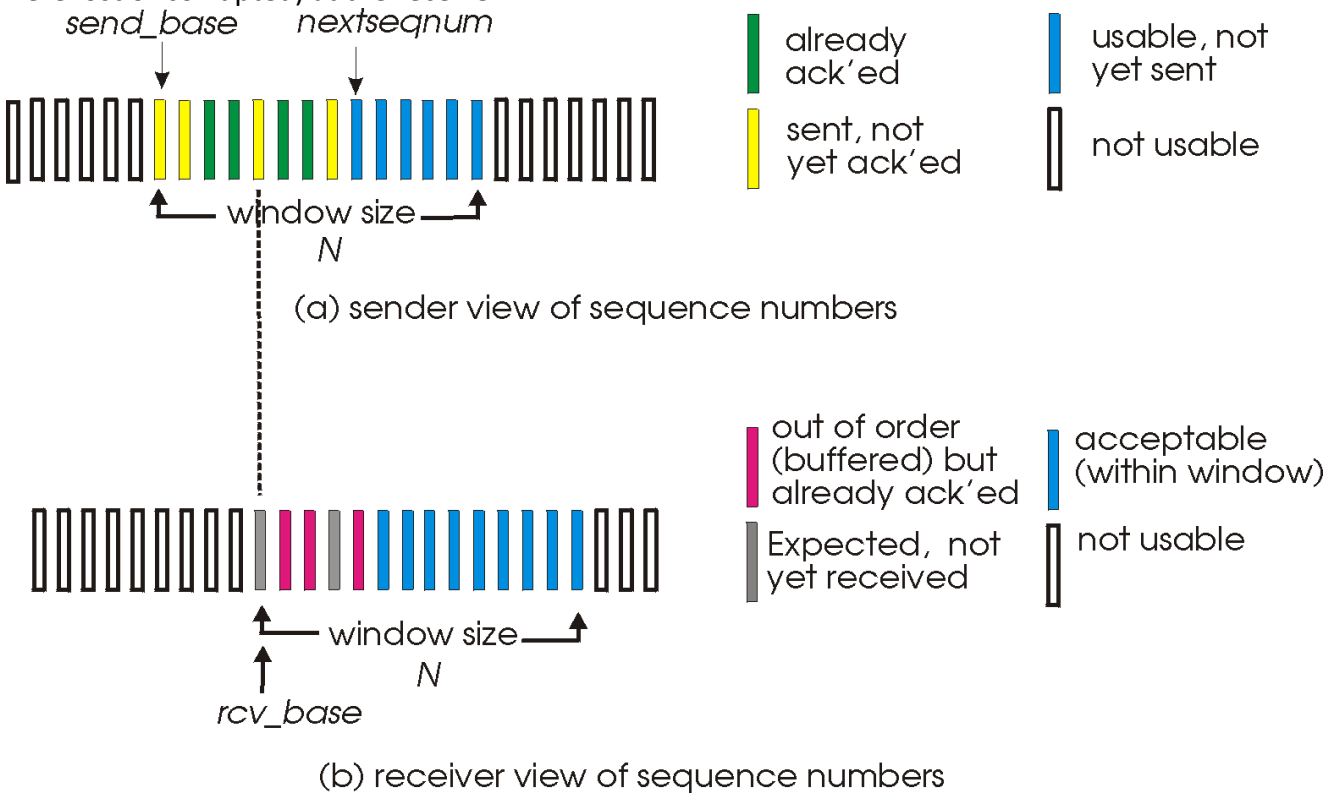
Go-Back-N (GBN)



In a Go-Back-N (GBN) protocol, the sender is allowed to transmit multiple packets (when available) without waiting for an acknowledgment, but is constrained to have no more than some maximum allowable number, *N*, of unacknowledged packets in the pipeline. Figure shows the sender's view of the range of sequence numbers in a GBN protocol. If we define*base* to be the sequence number of the oldest unacknowledged packet and *nextseqnum* to be the smallest unused sequence number (i.e., the sequence number of the next packet to be sent), then four intervals in the range of sequence numbers can be identified. Sequence numbers in the interval *[0,base-1]* correspond to packets that have already been transmitted and acknowledged. The interval *[base,nextseqnum-1]* corresponds to packets that have been sent but not yet acknowledged. Sequence numbers in the interval *[nextseqnum,base+N-1]* can be used for packets that can be sent immediately, should data arrive from the upper layer. Finally, sequence numbers greater than or equal to *base+N* can not be used until an unacknowledged packet currently in the pipeline has been acknowledged.

3.4.4 Selective Repeat (SR)

The GBN protocol allows the sender to potentially ``fill the pipeline'' with packets, thus avoiding the channel utilization problems we noted with stop-and-wait protocols. There are, however, scenarios in which GBN itself will suffer from performance problems. In particular, when the window size and bandwidth-delay product are both large, many packets can be in the pipeline. A single packet error can thus cause GBN to retransmit a large number of packets, many of which may be unnecessary. As the probability of channel errors increases, the pipeline can become filled with these unnecessary retransmissions. Imagine in our message dictation scenario, if every time a word was garbled, the surrounding 1000 words (e.g., a window size of 1000 words) had to be repeated. The dictation would be slowed by all of the reiterated words.
As the name suggests, Selective Repeat (SR) protocols avoid unnecessary retransmissions by having the sender retransmit only those packets that it suspects were received in error (i.e., were lost or corrupted) at the receiver.



---

1. **Data received from above.** When data is received from above, the SR sender checks the next available sequence number for the packet. If the sequence number is within the sender's window, the data is packetized and sent; otherwise it is either buffered or returned to the upper layer for later transmission, as in GBN.
2. **Timeout.** Timers are again used to protect against lost packets. However, each packet must now have its own logical timer, since only a single packet will be transmitted on timeout. A single hardware timer can be used to mimic the operation of multiple logical timers.
3. **ACK received.** If an ACK is received, the SR sender marks that packet as having been received, provided it is in the window. If the packet's sequence number is equal

to sendbase, the window base is moved forward to the unacknowledged packet with the smallest sequence number. If the window moves and there are untransmitted packets with sequence numbers that now fall within the window, these packets are transmitted.

Connection-Oriented Transport: TCP

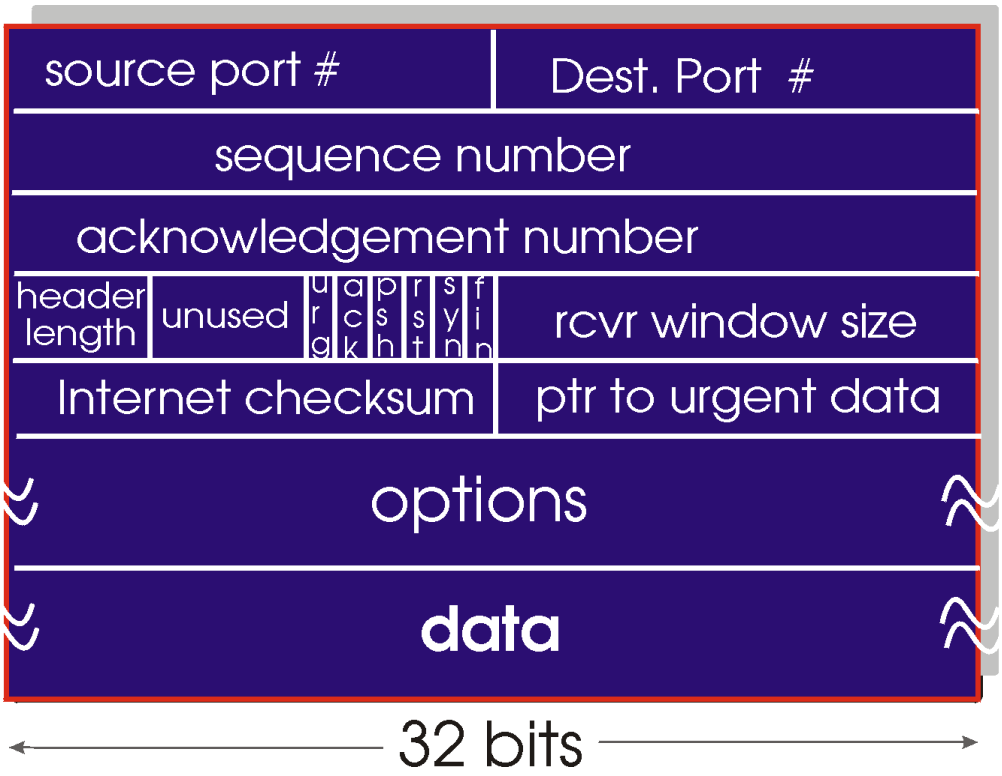
TCP provides multiplexing, demultiplexing, and error detection (but not recovery) in exactly the same manner as UDP. Nevertheless, TCP and UDP differ in many ways. The most fundamental difference is that UDP is **connectionless,** while TCP is **connection-oriented**. UDP is connectionless because it sends data without ever establishing a connection. TCP is connection-oriented because before one application process can begin to send data to another, the two processes must first "handshake" with each other -- that is, they must send some preliminary segments to each other to establish the parameters of the ensuing data transfer. As part of the TCP connection establishment, both sides of the connection will initialize many TCP "state variables"

TCP Segment Structure

The TCP segment consists of header fields and a data field. The data field contains a chunk of application data.

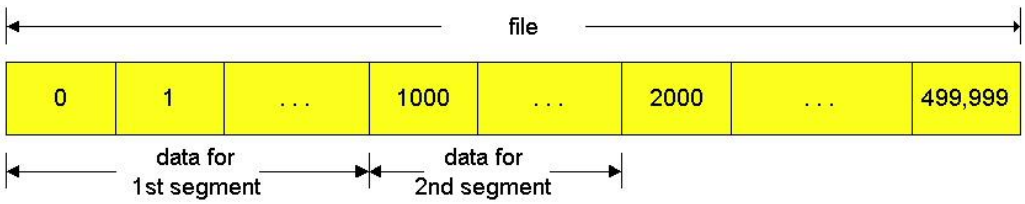
- The32-bit **sequence number field**, and the 32-bit **acknowledgment number field** are used by the TCP sender and receiver in implementing a reliable data transfer service, as discussed below.
- The 16-bit **window size** field is used for the purposes of flow control. We will see shortly that it is used to indicate the number of bytes that a receiver is willing to accept.
- The 4-bit **length field** specifies the length of the TCP header in 32-bit words. The TCP header can be of variable length due to the TCP options field, discussed below. (Typically, the options field is empty, so that the length of the typical TCP header is 20 bytes.)
- The optional and variable length **options field** is used when a sender and receiver negotiate the maximum segment size (MSS) or as a window scaling factor for use in high-speed networks. A timestamping option is also defined.

- The **flag field** contains 6 bits. The **ACK bit** is used to indicate that the value carried in the acknowledgment field is valid. The **RST**, **SYN** and **FIN** bits are used for connection setup and teardown, as we will discuss at the end of this section. When the **PSH** bit is set, this is an indication that the receiver should pass the data to the upper layer immediately. Finally, the **URG** bit is used to indicate there is data in this segment that the sending-side upper layer entity has marked as ``urgent.'' The location of the last byte of this urgent data is indicated by the 16-bit urgent data pointer. TCP must inform the receiving-side upper layer entity when urgent data exists and pass it a pointer to the end of the urgent data. (In practice, the PSH, URG and pointer to urgent data are not used. However, we mention these fields for completeness.)



Sequence Numbers and Acknowledgment Numbers

Two of the most important fields in the TCP segment header are the sequence number field and the acknowledgment number field. These fields are a critical part of TCP's reliable data transfer service



Dividing file data into TCP segments.

Reliable Data Transfer

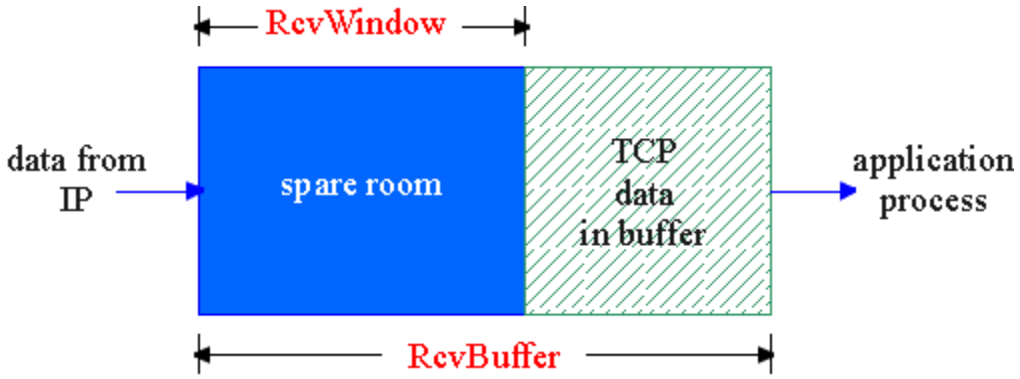
TCP creates a **reliable data transfer** service on top of IP's unreliable best-effort service. Many popular application protocols -- including FTP, SMTP, NNTP, HTTP and Telnet -- use TCP rather than UDP primarily because TCP provides reliable data transfer service. TCP's reliable data transfer service ensures that the data stream that a process reads out of its TCP receive buffer is uncorrupted, without gaps, without duplication, and in sequence, i.e., the byte stream is exactly the same byte stream that was sent by the end system on the other side of the connection. In this subsection we provide an informal overview of how TCP provides reliable data transfer

Flow Control

TCP thus provides a **flow control service** to its applications by eliminating the possibility of the sender overflowing the receiver's buffer. Flow control is thus a speed matching service - matching the rate at which the sender is seding to the rate at which the receiving application is reading. As noted earlier, a TCP sender can also be throttled due to congestion within the IP network; this form of sender control is referred to as **congestion control**

TCP provides flow control by having the sender maintain a variable called the **receive window.** Informally, the receive window is used to give the sender an idea about how much free buffer space is available at the receiver. In a full-duplex connection, the sender at each side of the connection maintains a distinct receive window. The receive window is dynamic, i.e., it changes throughout a connection's lifetime. Let's investigate the receive window in the context of a file transfer.

Suppose that host A is sending a large file to host B over a TCP connection. Host B allocates a receive buffer to this connection; denote its size by `RcvBuffer`. From time to time, the application process in host B reads from the buffer.



The receive window (`RcvWindow`) and the receive buffer (`RcvBuffer`)

Round Trip Time and Timeout

Recall that when a host sends a segment into a TCP connection, it starts a timer. If the timer expires before the host receives an acknowledgment for the data in the segment, the host retransmits the segment. The time from when the timer is started until when it expires is called the **timeout** of the timer.

The sample RTT, denoted `SampleRTT`, for a segment is the time from when the segment is sent (i.e., passed to IP) until an acknowledgment for the segment is received. Each segment sent will have its own associated `SampleRTT`. Obviously, the `SampleRTT` values will fluctuate from segment to segment due to congestion in the routers and to the varying load on the end systems. Because of this fluctuation, any given `SampleRTT` value may be atypical. In order to estimate a typical RTT, it is therefore natural to take some sort of average of the `SampleRTT` values. TCP maintains an average, called `EstimatedRTT`, of the `SampleRTT` values. Upon receiving an acknowledgment and obtaining a new `SampleRTT`, TCP updates `EstimatedRTT` according to the following formula:

```
EstimatedRTT = (1-x) EstimatedRTT + x SampleRTT.
```

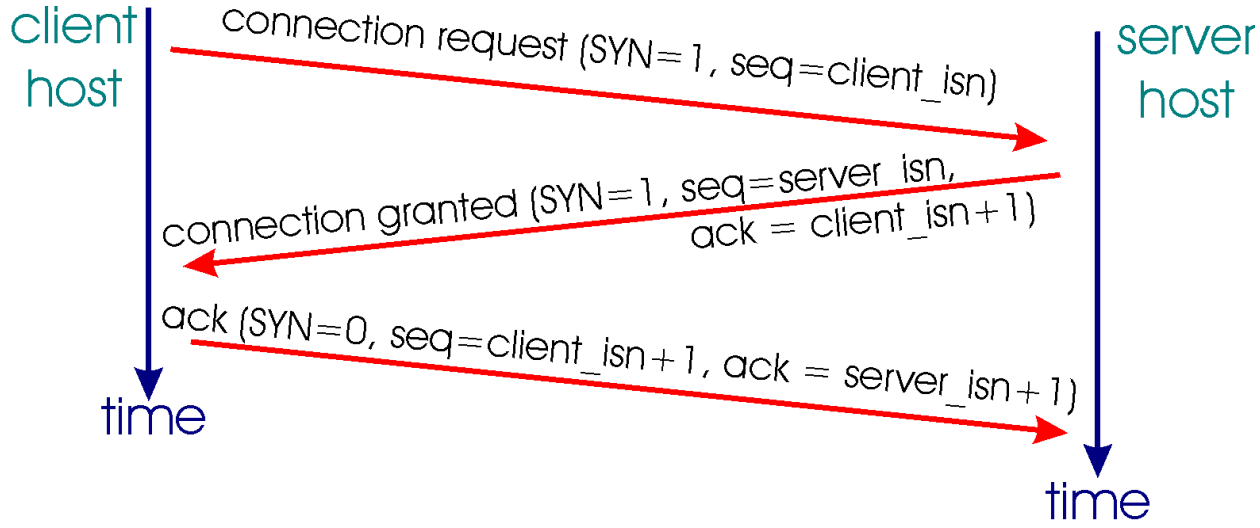
TCP Connection Management

The TCP in the client then proceeds to establish a TCP connection with the TCP in the server in the following manner:

- **Step 1.** The client-side TCP first sends a special TCP segment to the server-side TCP. This special segment contains no application-layer data. It does, however, have one of the flag bits in the segment's header (see Figure 3.3-2), the so-called SYN bit, set to 1. For this reason, this special segment is referred to as a **SYN segment**. In addition, the client chooses an initial sequence number (*client_isn)* and puts this number in the sequence number field of the

initial TCP SYN segment.This segment is encapsulated within an IP datagram and sent into the Internet.

- **Step 2.** Once the IP datagram containing the TCP SYN segment arrives at the server host (assuming it does arrive!), the server extracts the TCP SYN segment from the datagram, allocates the TCP buffers and variables to the connection, and sends a connection-granted segment to client TCP. This connection-granted segment also contains no application-layer data. However, it does contain three important pieces of information in the segment header. First, the SYN bit is set to 1. Second, the acknowledgment field of the TCP segment header is set to *isn+1*. Finally, the server chooses its own initial sequence number (server_isn) and puts this value in the sequence number field of the TCP segment header. This connection granted segment is saying, in effect, "I received your SYN packet to start a connection with your initial sequence number, *client_isn*. I agree to establish this connection. My own initial sequence number is *server_isn*." The conenction-granted segment is sometimes referred to as a **SYNACK** segment.
- **Step 3.** Upon receiving the connection-granted segment, the client also allocates buffers and variables to the connection. The client host then sends the server yet another segment; this last segment acknowledges the server's connection-granted segment (the client does so by putting the value *server_isn+1* in the acknowledgment field of the TCP segment header). The SYN bit is set to 0, since the connection is established.
-

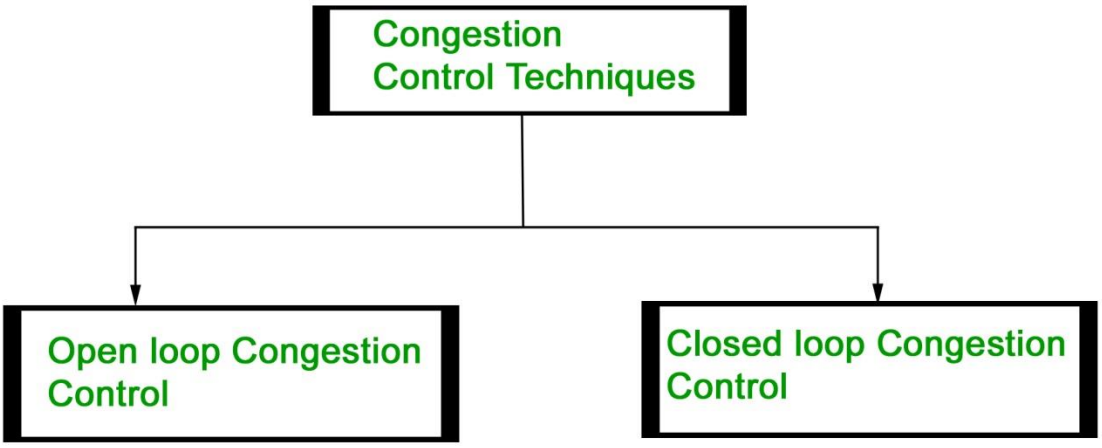


-

TCP three-way handshake: segment exchange

Congestion control

Congestion control refers to the techniques used to control or prevent congestion. Congestion control techniques can be broadly classified into two categories:



Open Loop Congestion Control

Open loop congestion control policies are applied to prevent congestion before it happens. The congestion control is handled either by the source or the destination.

Policies adopted by open loop congestion control –

1. **Retransmission Policy:**
   It is the policy in which retransmission of the packets are taken care of. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. This transmission may increase the congestion in the network.
   To prevent congestion, retransmission timers must be designed to prevent congestion and also able to optimize efficiency.

2. **Window Policy:**
   The type of window at the sender's side may also affect the congestion. Several packets in the Go-back-n window are re-sent, although some packets may be received successfully at the receiver side. This duplication may increase the congestion in the network and make it worse.
   Therefore, Selective repeat window should be adopted as it sends the specific packet that may have been lost.

3. **Discarding Policy:**
   A good discarding policy adopted by the routers is that the routers may prevent congestion and at the same time partially discard the corrupted or less sensitive packages and also be able to maintain the quality of a message.
   In case of audio file transmission, routers can discard less sensitive packets to prevent congestion and also maintain the quality of the audio file.

4. **Acknowledgment Policy:**
   Since acknowledgements are also the part of the load in the network, the acknowledgment policy imposed by the receiver may also affect congestion. Several approaches can be used to prevent congestion related to acknowledgment.
   The receiver should send acknowledgement for N packets rather than sending acknowledgement for a single packet. The receiver should send an acknowledgment only if it has to send a packet or a

timer expires.

5. **Admission Policy :**
   In admission policy a mechanism should be used to prevent congestion. Switches in a flow should first check the resource requirement of a network flow before transmitting it further. If there is a chance of a congestion or there is a congestion in the network, router should deny establishing a virtual network connection to prevent further congestion.

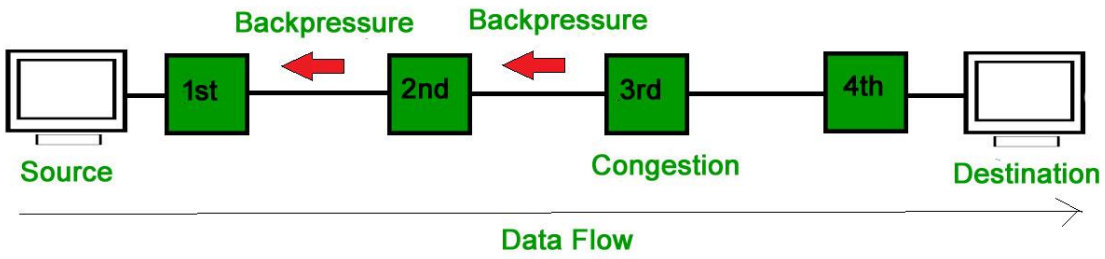
All the above policies are adopted to prevent congestion before it happens in the network.

Closed Loop Congestion Control

Closed loop congestion control techniques are used to treat or alleviate congestion after it happens. Several techniques are used by different protocols; some of them are:

1. Backpressure :

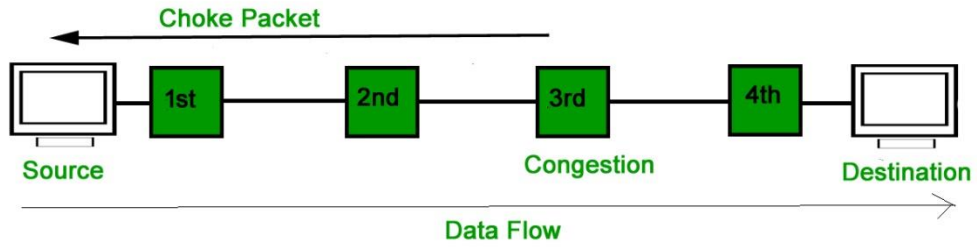
Backpressure is a technique in which a congested node stops receiving packets from upstream node. This may cause the upstream node or nodes to become congested and reject receiving data from above nodes. Backpressure is a node-to-node congestion control technique that propagate in the opposite direction of data flow. The backpressure technique can be applied only to virtual circuit where each node has information of its above upstream node.



In above diagram the 3rd node is congested and stops receiving packets as a result 2nd node may be get congested due to slowing down of the output data flow. Similarly 1st node may get congested and inform the source to slow down.

2. Choke Packet Technique :

Choke packet technique is applicable to both virtual networks as well as datagram subnets. A choke packet is a packet sent by a node to the source to inform it of congestion. Each router monitors its resources and the utilization at each of its output lines. Whenever the resource utilization exceeds the threshold value which is set by the administrator, the router directly sends a choke packet to the source giving it a feedback to reduce the traffic. The intermediate nodes through which the packets has traveled are not warned about congestion.



3. Implicit Signaling :
In implicit signaling, there is no communication between the congested nodes and the source. The source guesses that there is congestion in a network. For example when sender sends several packets and there is no acknowledgment for a while, one assumption is that there is a congestion.

4. Explicit Signaling :
In explicit signaling, if a node experiences congestion it can explicitly sends a packet to the source or destination to inform about congestion. The difference between choke packet and explicit signaling is that the signal is included in the packets that carry data rather than creating a different packet as in case of choke packet technique.
Explicit signaling can occur in either forward or backward direction.

- **Forward Signaling :** In forward signaling, a signal is sent in the direction of the congestion. The destination is warned about congestion. The receiver in this case adopt policies to prevent further congestion.
- **Backward Signaling :** In backward signaling, a signal is sent in the opposite direction of the congestion. The source is warned about congestion and it needs to slow down.

Principles of Network Applications

The Principles of Network Applications are fundamental concepts that govern the design and development of applications that run on a computer network. These principles encompass several key aspects of network applications, including:

- Network Application Architectures
- Processes Communicating
- The Interface Between the Process and the Computer Network
- Transport Services Available to Applications
- Transport Services Provided by the Internet
- Application-Layer Protocols

1. Network Application Architectures refer to the overall design and structure of a network application. It encompasses how the application is divided into different components, and how these components interact with each other. There are several commonly used network application architectures, including:

- **Client-Server Architecture:** In this architecture, one component acts as a client and makes requests to a server component, which provides the requested services. This architecture is commonly used in web applications, where the client is a web browser and the server is a web server.
- **Peer-to-Peer Architecture:** In this architecture, every component is both a client and a server, and each component can communicate directly with any other component. This architecture is commonly used in file-sharing applications, where each user's device acts as both a client and a server.
- **Three-Tier Architecture**: In this architecture, the application is divided into three components: a client component, a middleware component, and a database component. The client component makes requests to the middleware component, which communicates with the database component to retrieve or update data. This architecture provides a layer of abstraction between the client and the database, which can improve performance and scalability.
- **Microservices Architecture**: In this architecture, the application is divided into a collection of small, independently deployable components, known as microservices. Each microservice is responsible for a specific aspect of the application's functionality, and they communicate with one another through APIs. This architecture allows for greater flexibility and scalability compared to other architectures.

The choice of network application architecture depends on several factors, including the requirements of the application, the scale of the application, and the available resources. Regardless of the architecture chosen, the design should consider factors such as scalability, performance, security, and maintainability.

2. **Processes Communicating** refers to the communication between multiple processes in a computer network. Processes can be thought of as individual programs or tasks running on a device, and they may be located on the same device or on different devices connected to the network.

- Communication between processes is facilitated by the use of protocols, which define the rules and formats for exchanging data. The communication between processes can be either synchronous or asynchronous, meaning that either both processes must be

available to communicate at the same time or the communication can occur at different times.

- In a network application, communication between processes is essential for the application to function correctly. For example, in a client-server architecture, the client process makes requests to the server process, and the server process returns the requested information. In a peer-to-peer architecture, each process can communicate directly with any other process.
- It's important to consider the communication between processes when designing a network application. Factors such as the reliability of communication, the security of communication, and the performance of the communication must be taken into account. The choice of protocols used for communication will depend on the requirements of the application and the network infrastructure.

3. The Interface between the Process and the Computer Network refers to the connection between a process running on a device and the underlying computer network. This interface determines how the process communicates with other processes and with the network itself.

The interface between a process and the computer network is usually provided by a network stack, which is a collection of protocols and services that handle the communication between the process and the network. The network stack translates the process's requests and data into the appropriate network protocols, and vice versa, allowing the process to communicate over the network.

The network stack typically includes several layers, each with its own specific responsibilities. The layers may include:

1. **Application Laye**r: This layer provides the interface between the process and the network stack. It defines the protocols and services used by the process to communicate with the network.
2. **Transport Laye**r: This layer provides the underlying transport services that enable the process to communicate with other processes over the network. These services include protocols such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).
3. **Network Layer:** This layer provides the basic mechanisms for routing data between devices on the network. The Internet Protocol (IP) is the most commonly used network layer protocol.
4. **Link Laye**r: This layer provides low-level communication services between devices on the same physical network. The link layer is responsible for error detection and correction, and for determining the best path for data to travel over the network.

The interface between the process and the computer network is a critical component of a network application, and its design must take into account factors such as performance, reliability, security, and compatibility with the network infrastructure. The choice of network stack and protocols used will depend on the requirements of the application and the underlying network.

4. Transport Services Available to Applications are the services provided by the network stack that enable applications to communicate with each other over a computer network. These services are responsible for ensuring that data is reliably delivered

between applications, and they provide the underlying communication infrastructure for the application.
There are several transport services available to applications, including:

1. **TCP (Transmission Control Protocol)**: TCP is a reliable, connection-oriented transport service that provides error-checking and flows control to ensure that data is delivered accurately. Applications that require reliable data delivery, such as email or file transfer, typically use TCP.
2. **UDP (User Datagram Protocol)**: UDP is an unreliable, connectionless transport service that does not provide error checking or flow control. Applications that require low latency or high speed, such as video streaming or online gaming, typically use UDP.
3. **SCTP (Stream Control Transmission Protocol)**: SCTP is a reliable, multi-homed transport service that provides error checking and flow control. SCTP can handle multiple streams of data between applications, allowing for efficient communication between applications.
4. **DCCP (Datagram Congestion Control Protocol)**: DCCP is a transport service that provides congestion control for applications that do not require reliable data delivery.

Transport Services Provided by the InternetThe choice of transport service will depend on the requirements of the application, including reliability, performance, and security requirements. For example, an application that requires reliable data delivery, such as email, would use TCP, while an application that requires low latency and high speed, such as online gaming, would use UDP.

5. Transport Services Provided by the Internet: The Internet provides two primary transport services for applications: TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

1. **TCP**: TCP is a reliable, connection-oriented transport service that provides error-checking and flows control to ensure that data is delivered accurately. Applications that require reliable data delivery, such as email or file transfer, typically use TCP. TCP establishes a reliable connection between two devices and ensures that data is transmitted in the correct order and without errors.
2. **UDP**: UDP is an unreliable, connectionless transport service that does not provide error checking or flow control. Applications that require low latency or high speed, such as video streaming or online gaming, typically use UDP. Because UDP does not provide error checking or flow control, it is faster and more efficient than TCP, but it may not be suitable for applications that require reliable data delivery.

In addition to these two primary transport services, the Internet may also provide other transport services, such as SCTP (Stream Control Transmission Protocol) or DCCP (Datagram Congestion Control Protocol), depending on the specific implementation. The choice of transport service will depend on the requirements of the application and the underlying network infrastructure.

6. Application-layer protocols are data exchange protocols used to allow communication between applications on different devices. They operate at the highest layer of the OSI (Open Systems Interconnection) model, which is the application layer. Application-layer protocols define the rules for exchanging data between applications, such as formatting, error detection and correction, and security. Examples of application-layer protocols

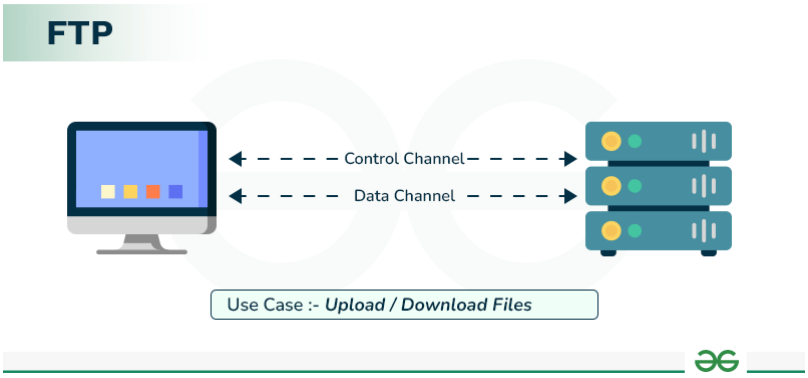
include HTTP, HTTPS, FTP, SMTP, POP3, IMAP, and many others. They are used in various applications such as web browsing, email, file transfer, and more.

File Transfer Protocol (FTP) in Application Layer

File Transfer Protocol (FTP) is an application layer protocol that moves files between local and remote file systems. It runs on top of TCP, like HTTP. To transfer a file, 2 TCP connections are used by FTP in parallel: control connection and data connection.

What is File Transfer Protocol?

FTP is a standard communication protocol. There are various other protocols like HTTP which are used to transfer files between computers, but they lack clarity and focus as compared to FTP. FTP shields the user from these differences and transfers data efficiently and reliably. FTP can transfer ASCII, EBCDIC, or image files. The ASCII is the default file share format,



Types of FTP

There are different ways through which a server and a client do a file transfer using FTP. Some of them are mentioned below:

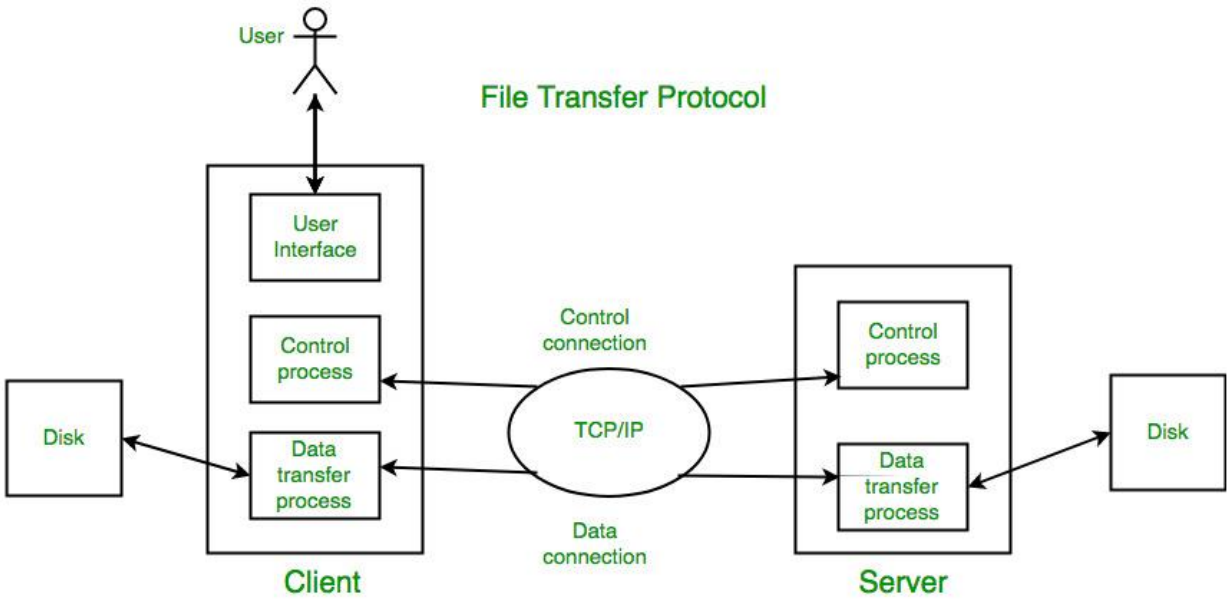
- **Anonymous FTP:** Anonymous FTP is enabled on some sites whose files are available for public access. A user can access these files without having any username or password. Instead, the username is set to anonymous, and the password is to the guest by default. Here, user access is very limited. For example, the user can be allowed to copy the files but not to navigate through directories.
- **Password Protected FTP:** This type of FTP is similar to the previous one, but the change in it is the use of username and password.
- **FTP Secure (FTPS):** It is also called as FTP Secure Sockets Layer (FTP SSL). It is a more secure version of FTP data transfer. Whenever FTP connection is established, Transport Layer Security (TLS) is enabled.
- **FTP over Explicit SSL/TLS (FTPES):** FTPES helps by upgrading FTP Connection from port 21 to an encrypted connection.
- **Secure FTP (SFTP):** SFTP is not a FTP Protocol, but it is a subset of Secure Shell Protocol, as it works on port 22.

How Does FTP Work?

FTP is a client server protocol that has two communication channel, command channel for conversation control and data channel for file content.

Here are steps mentioned in which FTP works:

- A user has to log in to FTP Server first, there may be some servers where you can access to content without login, known as anonymous FTP.
- Client can start a conversation with server, upon requesting to download a file.
- The user can start different functions like upload, delete, rename, copy files, etc. on server.



Types of Connection in FTP

- Control Connection
- Data Connection

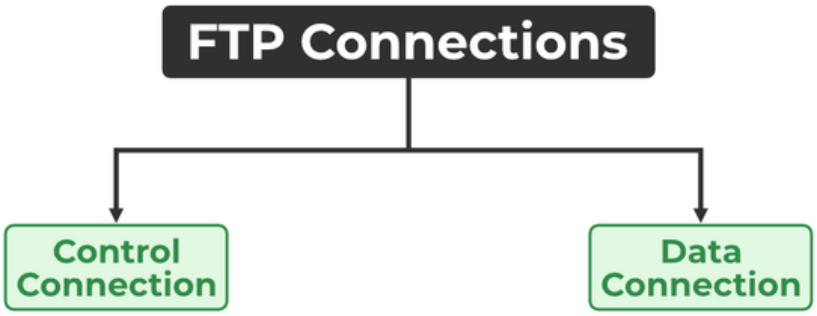
Control Connection

For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc., FTP makes use of a control connection. The control connection is initiated on port number 21.

Data connection

For sending the actual file, FTP makes use of a data connection. A data connection is initiated on port number 20.

FTP sends the control information out-of-band as it uses a separate control connection. Some protocols send their request and response header lines and the data in the same TCP connection. For this reason, they are said to send their control information in-band. HTTP and SMTP are such examples.



FTP Data Types

The data type of a file, which determines how the file is represented overall, is the first piece of information that can be provided about it. The FTP standard specifies the following four categories of data:

- **ASCII:** Describes an ASCII text file in which each line is indicated by the previously mentioned type of end-of-line marker.
- **EBCDIC:** For files that use IBM's EBCDIC character set, this type is conceptually identical to ASCII.
- **Image:** This is the "black box" mode I described earlier; the file has no formal internal structure and is transferred one byte at a time without any processing.
- **Local:** Files containing data in logical bytes with a bit count other than eight can be handled by this data type.

FTP Replies

Some of the FTP replies are:

- 200 – Command okay.
- 530 – Not logged in.
- 331 – User name okay, need a password.
- 221 – Service closing control connection.
- 551 – Requested action aborted: page type unknown.
- 502 – Command not implemented.
- 503 – Bad sequence of commands.
- 504 – Command not implemented for that parameter.

Advantages of FTP

- File sharing also comes in the category of advantages of FTP in this between two machines files can be shared on the network.
- Speed is one of the main benefits of FTP.
- Since we don't have to finish every operation to obtain the entire file, it is more efficient.
- Using the username and password, we must log in to the FTP server. As a result, FTP might be considered more secure.
- We can move the files back and forth via FTP. Let's say you are the firm manager and you provide information to every employee, and they all reply on the same server.

Disadvantages of FTP

- File size limit is the drawback of FTP only 2 GB size files can be transferred.
- More then one receivers are not supported by FTP.
- FTP does not encrypt the data this is one of the biggest drawbacks of FTP.
- FTP is unsecured we use login IDs and passwords making it secure but they can be attacked by hackers.

Simple Mail Transfer Protocol (SMTP)

Email is emerging as one of the most valuable services on the internet today. Most internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) is used to retrieve those emails at the receiver's side.

SMTP Fundamentals

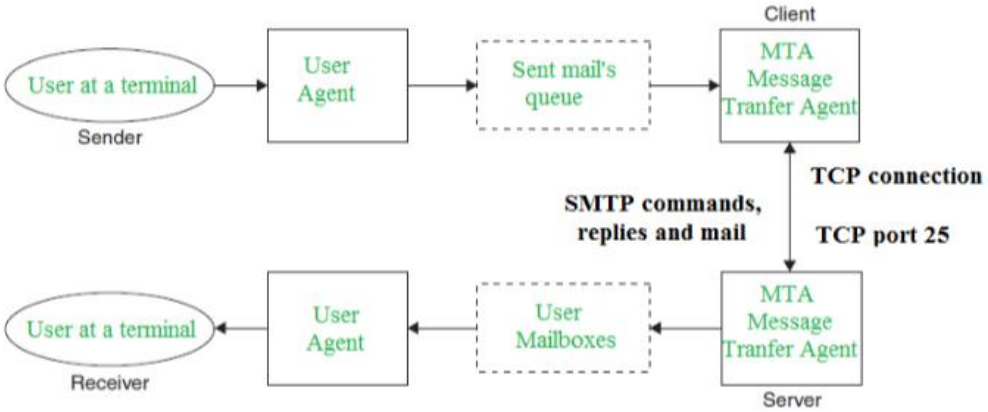
SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is an always-on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection through port 25. After successfully establishing a TCP connection the client process sends the mail instantly

SMTP Protocol

The SMTP model is of two types:

1. End-to-end method
2. Store-and-forward method

The end-to-end model is used to communicate between different organizations whereas the store and forward method is used within an organization.



Components of SMTP

1. Mail User Agent (MUA)
2. Mail Submission Agent (MSA)
3. Mail Transfer Agent (MTA)
4. Mail Delivery Agent (MDA)

1. Mail User Agent (MUA): It is a computer application that helps you in sending and retrieving mail. It is responsible for creating email messages for transfer to the mail transfer agent(MTA).

2. Mail Submission Agent (MSA): It is a computer program that basically receives mail from a Mail User Agent(MUA) and interacts with the Mail Transfer Agent(MTA) for the transfer of the mail.

3. Mail Transfer Agent(MTA): It is basically software that has the work to transfer mail from one system to another with the help of SMTP.

4. Mail Delivery Agent(MDA): A mail Delivery agent or Local Delivery Agent is basically a system that helps in the delivery of mail to the local system.

Working of SMTP

1. Communication between the sender and the receiver:

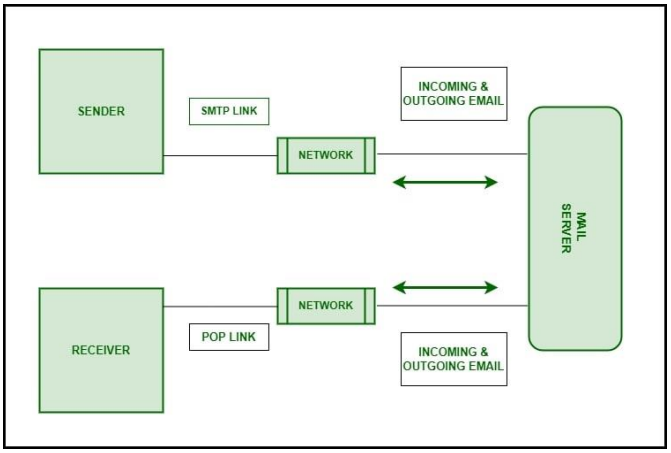
The sender's user agent prepares the message and sends it to the MTA. The MTA's responsibility is to transfer the mail across the network to the receiver's MTA. To send mail, a system must have a client MTA, and to receive mail, a system must have a server MTA.

2. Sending Emails:

Mail is sent by a series of request and response messages between the client and the server. The message which is sent across consists of a header and a body. A null line is used to terminate the mail header and everything after the null line is considered the body of the message, which is a sequence of ASCII characters. The message body contains the actual information read by the receipt.

3.Receiving Emails:

The user agent on the server-side checks the mailboxes at a particular time of intervals. If any information is received, it informs the user about the mail. When the user tries to read the mail it displays a list of emails with a short description of each mail in the mailbox. By selecting any of the mail users can view its contents on the terminal.



Some SMTP Commands

- HELO – Identifies the client to the server, fully qualified domain name, only sent once per session
- MAIL – Initiate a message transfer, the fully qualified domain of the originator
- RCPT – Follows MAIL, identifies an addressee, typically the fully qualified name of the addressee, and for multiple addressees use one RCPT for each addressee
- DATA – send data line by line

Advantages of SMTP

- If necessary, the users can have a dedicated server.
- It allows for bulk mailing.
- Low cost and wide coverage area.
- Offer choices for email tracking.
- Reliable and prompt email delivery.

Disadvantages of SMTP

- SMTP's common port can be blocked by several firewalls.
- SMTP security is a bigger problem.
- Its simplicity restricts how useful it can be.
- Just 7-bit ASCII characters can be used.
- If a message is longer than a certain length, SMTP servers may reject the entire message.
- Delivering your message will typically involve additional back-and-forth processing between servers, which will delay sending and raise the likelihood that it won't be sent.

Difference between SMTP and HTTP

SMTP	HTTP
SMTP is used for mail services.	HTTP is mainly used for data and file transfer.
It uses port 25.	It uses port 80.
It is primarily a push protocol.	It is primarily a pull protocol.
It imposes a 7-bit ASCII restriction on the content to be transferred.	It does not impose a 7-bit ASCII restriction. Can transfer multimedia, hyperlinks, etc.

SMTP	HTTP
SMTP transfers emails via Mail Servers.	HTTP transfers files between the Web server and the Web client.
SMTP is a persistent type of TCP connection.	It can use both Persistent and Non-persistent.
Uses base64 encoding for authentication.	Uses different methods of authentication such as basic, digest, and OAuth.
Does not support session management or cookies.	Supports session management and cookies to maintain state.
Has a smaller message size limit compared to HTTP.	Has a larger message size limit compared to SMTP.
Requires authentication for sending emails.	Does not require authentication for browsing web pages.
Supports both plain text and encrypted communication (SMTPS or STARTTLS).	Supports both plain text and encrypted communication (HTTPS).

Services Provided by DNS

The DNS is

(1) a distributed database implemented in a hierarchy of DNS servers, and

(2) an application-layer protocol that allows hosts to query the distributed database.

• DNS is commonly employed by other application-layer protocols—including HTTP, SMTP, and FTP—to translate user-supplied hostnames to IP addresses.

• **Host aliasing:** A host with a complicated hostname can have one or more alias names. For example, a hostname such as relay1.west-coast.enterprise.com could have, say, two aliases such as enterprise.com and www.enterprise.com. In this case, the hostname relay1.westcoast. enterprise.com is said to be a canonical hostname. Alias hostnames, when present, are typically more mnemonic than canonical Downloaded by MRIET ECE (hodece@mriet.ac.in) lOMoARcPSD|37349925 hostnames. DNS can be invoked by an application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.

• **Mail server aliasing:** For obvious reasons, it is highly desirable that e-mail addresses be mnemonic. For example, if Bob has an account with Hotmail, Bob's e-mail address might be as simple as bob@hotmail.com. However, the hostname of the Hotmail mail server is more complicated and much less mnemonic than simply hotmail.com (for example, the canonical hostname might be something like relay1.westcoast.hotmail.com). DNS can be invoked by a mail application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.

• **Load distribution:** DNS is also used to perform load distribution among replicated servers, such as replicated Web servers. Busy sites, such as cnn.com, are replicated over multiple servers, with each server running on a different end system and each having a different IP address. For replicated Web servers, a set of IP addresses is thus associated with one canonical hostname. The DNS database contains this set of IP addresses. When clients make a DNS query for a name mapped to a set of addresses, the server responds with the entire set of IP addresses, but rotates the ordering of the addresses within each reply. Because a client typically sends its HTTP request message to the IP address that is listed first in the set, DNS rotation distributes the traffic among the replicated servers.

Overview of How DNS Works

• Suppose that some application running in a user's host needs to translate a hostname to an IP address. The application will invoke the client side of DNS, specifying the hostname that needs to be translated.

• DNS in the user's host then takes over, sending a query message into the network.

• All DNS query and reply messages are sent within UDP datagrams to port 53. After a delay, ranging from milliseconds to seconds, DNS in the user's host receives a DNS reply message that provides the desired mapping. This mapping is then passed to the invoking application.

The problems with a centralized design include:

• A single point of failure. If the DNS server crashes, so does the entire Internet!

• Traffic volume. A single DNS server would have to handle all DNS queries.

• Distant centralized database. A single DNS server cannot be “close to” all the querying clients. If we put the single DNS server in New York City, then all queries from Australia must travel to the other side of the globe, perhaps over slow and congested links. This can lead to significant delays.

• Maintenance. The single DNS server would have to keep records for all Internet hosts. Not only would this centralized database be huge, but it would have to be updated frequently to account for every new host.

DNS Records

The DNS servers that together implement the DNS distributed database store resource records

Each resource record contains 4 fields: Name, Value, Type, TTL

TTL stands for the time to live of the resource record; it determines when a resource should be removed from a cache

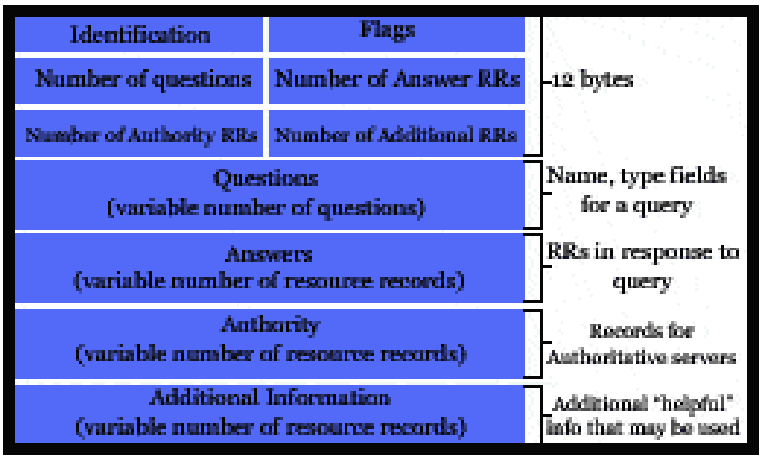
The meaning of Name and Value depend on Type

- If Type = A, then Name is a hostname and Value is the IP address for the hostname; this record type provides the standard hostname to IP address mapping
- If Type = NS, then Nane is a domain and Value is the hostname of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain; this record type is used to route DNS queries further along in the query chain
- If Type = CNAME, then Value is the canonical hostname for the alias hostname Name
- If Type = MX, then Value is the canonical name of a mail server that has an alias hostname Name; MX records allow the hostnames of mail servers to have simple aliases

NS Messages

There are only two types: **query** and **reply**

Both have the same format



The first 12 bytes is the **header section**

Parts of the header section

1. Identification Field
    1. 16-bit number that identifies the query
    2. Identifier is copied into the reply message to a query, allowing the client to
       match received replies with sent queries
2. Flag field
    1. 1-bit query/reply flag indicates whether the message is a query (0) or a reply (1)
    2. 1-bit authoritative flag is set in a reply message when a DNS server is an authoritative server for a queried name
    3. 1-bit recursion-desired flag is set when a client desires that the DNS server
       perform recursion when it doesn't have the record
    4. 1-bit recursion-available flag is set in a reply if the DNS server supports recursion
3. 4 number-of fields
    1. Indicate the number of occurrences of the four types of data sections that
       follow the header

The **question section** contains information about the query that is being made

Parts of the question section

1. A name field that contains the name that is being queried
2. A type field that indicates the type of question being asked about the name

The **answer section** contains the resource records for the name that was originally queried
A reply can return multiple RRs in the answer, since a hostname can have multiple IP addresses

The **authority section** contains records of other authoritative servers

The **additional section** contains other helpful records

Ns lookup program allows you to to send a DNS query message directly from the local host to any

Inserting Records into the DNS Database

- Register your domain name with a registrar
    - Registrar - a commercial entity that verifies the uniqueness of the domain name, enters the domain name into the DNS database, and collects a small fee from you for its services
    - There are many registrars competing for customers, and the Internet Corporation for Assigned Names and Numbers (ICANN) accredits the various registrars
- Provide the registrar with the names and IP addresses or your primary and secondary authoritative DNS servers
- Ther registrar ensures that a type NS and a type A record are entered into the TLD com servers
- You will have to make sure that the type A resource record for your Web server and the type MX resourse record for your mail server are entered into your authoritative DNS servers