UNIT - I

- The old model of a single computer serving all of the organizational's computational needs has been replaced by one in which a large no. of separate but interconnected computers do the job. These systems are called COMPUTER NETWORKS.

- CN means a collection of autonomous computers interconnected by a single technology.
- Two computers are said to be interconnected if they are able to exchange information & share resources
- The connection need not be via a copper wire; fiber optics, microwaves, infrared & communication satellites can also be used.
- Networks are in many sizes, shapes & forms and they are usually connected together to make large networks.

  Ex:- Internet (network of networks)

Uses of COMPUTER N/W's:-

- Business applications
- Home applications
- Mobile users
- Social issues

→ Network Hardware :-

Physical Structures :-

- Before discussing n/w's, we need to define some n/w attributes.

Types of connection :-

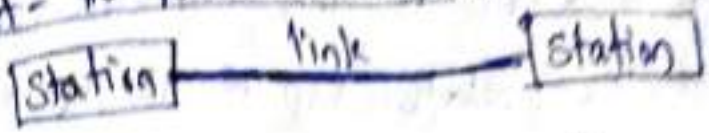
A n/w is two or more (communication) devices connected through link. A link is a communication pathway that transfers data from one device to another.

2 possible Types of connections :-

→ Point-to-point

→ Multipoint

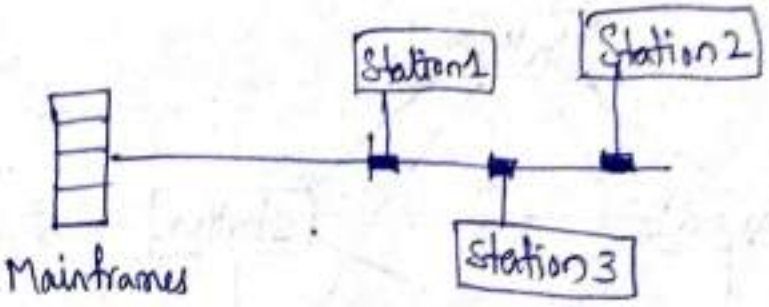
Point-to-point connection:-



point-point connection.

- provids a dedicated link b/n two devices.
- Entire capacity of the link is reserved for transmission b/n those two devices.
- uses wires (or) cables, to connect two ends, as well as microwave (or) satellite links, are also used.

Multipoint connection:-



- In which more than two specific devices share a single link.
- Entire capacity of the link is shared among all devices temporally.

Topologies:-

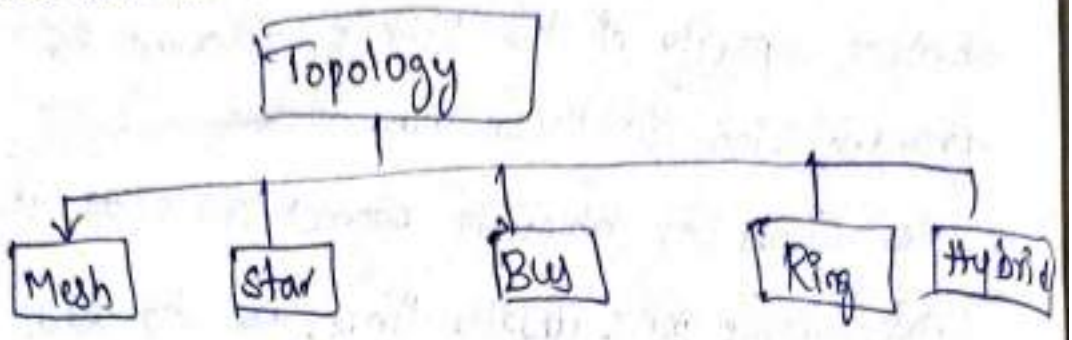
~~The basic top~~

Physical Topology:-

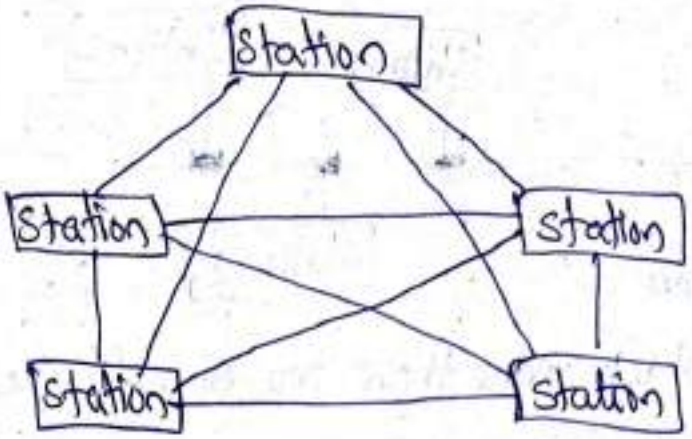
- Term physical topology refers to the way in which a n/w is laid out physically.

Two (or) more devices connect to a link;
two (or) more links form a topology.

- The Topology of a nw is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another.



Mesh Topology:-



- In Mesh topology, each node is connected to every other nodes in the network.

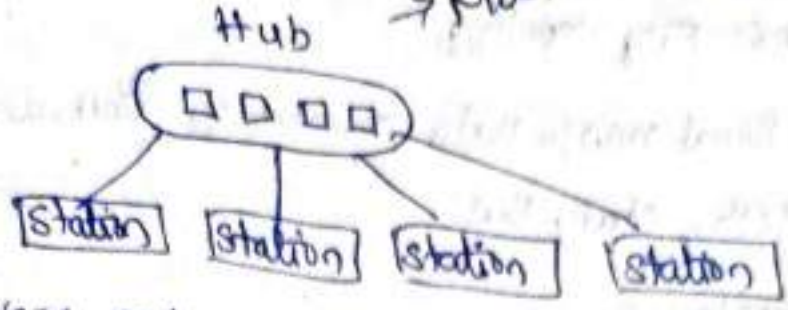
2 types of Mesh topology:-

→ Full mesh Topology:- every computer in the n/w has a connection to each of the other computers in that n/w.

→ Partially connected mesh topology:- Atleast two of computers in the n/w have connection to multiple other connections.

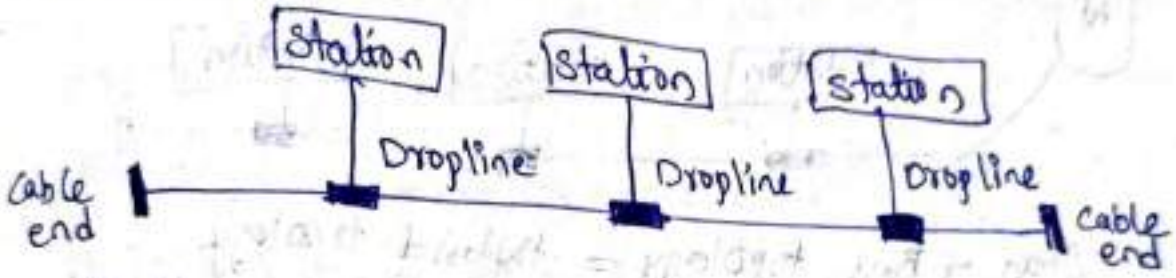
Star Topology:-

→ Most commonly used



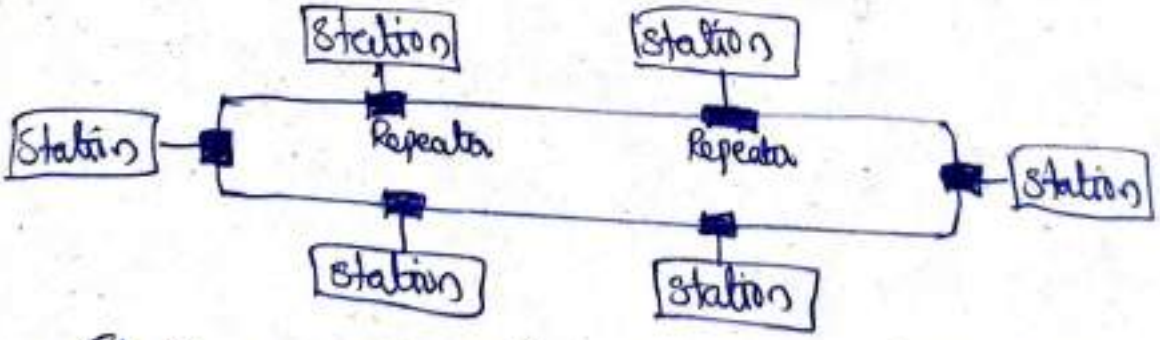
- Every node connects to a central n/w device, like hub, swith (or) computer.
- Central n/w device acts a server and pheripherals devices acts as clients.

Bus Topology:-



- Each computer and n/w device are connected to a single cable

Ring Topology:-



- It is a n/w configuration in which device connections create a circular data path.
- Data travel from one device to the next untill they reach their destination.

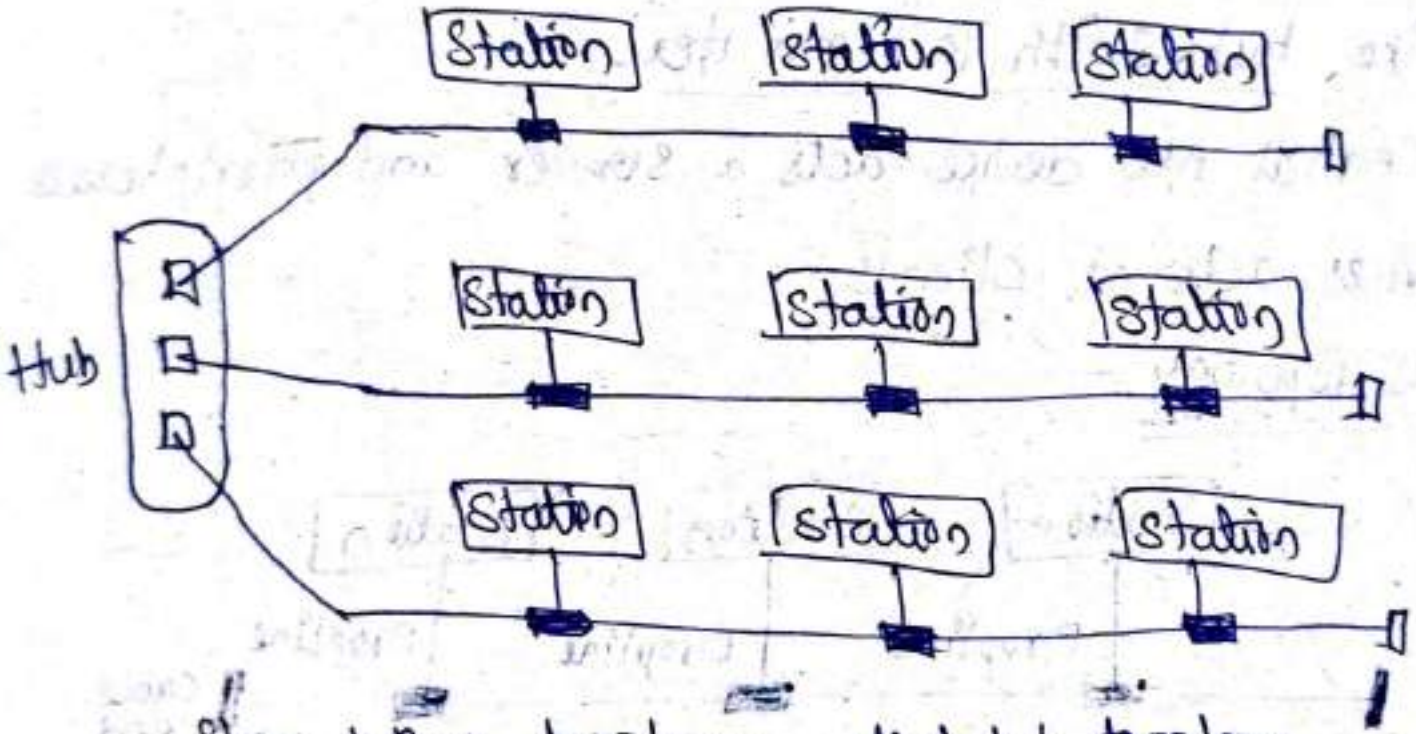
Unidirectional n/w :- data travel in one direction.

Ex:- Ring topology.

Bidirectional n/w :- Data to move in either direction

Ex:- Mesh, star, Bus.

Hybrid Topology :-



Star + Bus topology = Hybrid topology

Network Hardware :-

→ Personal Area Networks (PANs)

→ Local Area Networks (LAN)

→ Metropolitan Area Networks (MAN)

→ Wide Area Networks (WAN)

→ Internetworks

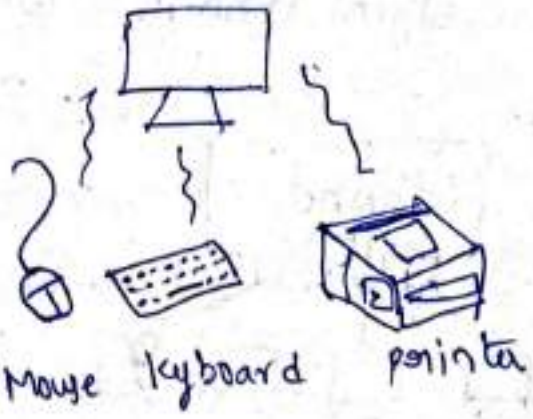
- A criterion for classifying networks is by scale. Means Distance is important as a classification metric because different technologies are used at different scale.

Inta processor distance	Processors located in same	Example
1 M	Square Meter	PAN
10 M	Room	LAN
100 M	Building	LAN
1 KM	Campus	LAN
10 KM	City	MAN
100 KM	Country	WAN
1000 KM	Continent	WAN
10,000 KM	Planet	The Internet.

Classification of interconnected processors by scale.

PAN :-

- n/w organized by the individual user for its personal use.
- The devices are communicated over a range of a person.
- Ex:- Wireless peripherals connected to a computer.
- To overcome the wire plugging & cables connection in wrong way, some companies got together to design a short-range wireless n/w called Bluetooth.
- Bluetooth is a wireless n/w and Master-slave network.,
  - PC is the Master
  - Peripherals like mouse, keyboard are slaves.
- Bluetooth technology is used in headsets to connect to mobile phones; In the car music system bluetooth connects to mobile phone for playing music.



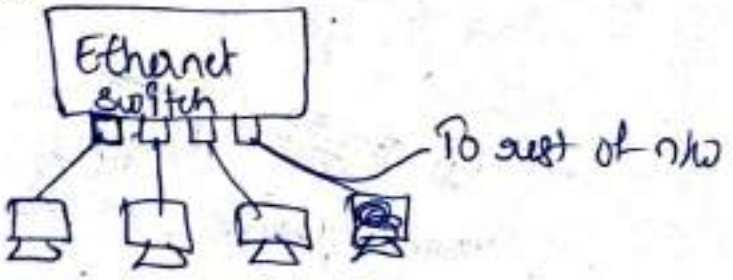
Bluetooth PAN configuration.

LAN:-

- A LAN is a privately owned n/w's
- LAN's are widely used to connect personal computers and, pheripherals to exchange information and sharing resources under a room (or) building (or) factory (or) office.
- LAN's use - Ring, star, bus topologies.
- LAN's are wired & wireless.
- Wired LAN that is most commonly used as IEE 802.3 standard, popularly called as Ethernet. Wired Lan's use commonly copper wire (or) some use optic fiber.

Wired LAN's run at a speed of 100Mbps to 1 Gbps, have low delay & make few errors. Newer LAN's can operate at upto 10 Gbps.

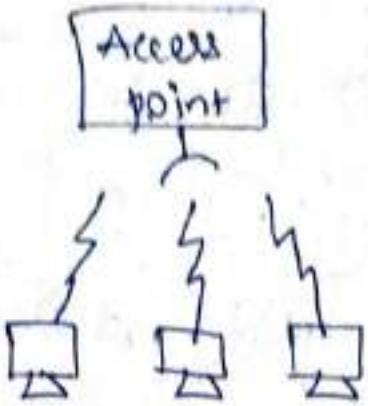
- compared to wireless n/ws, wired LAN's exceed them in all dimentions of performance. It is just easier to send a signal through a wire (or) optics than the air.



switched n/w.

Wireless LAN:- very popular these days, especially at homes, older offices building, other where it is too much trouble to install wired LAN.

- uses Radio Modem & Antenna to communicate with other computers.
- Access point (AP), wireless router (or) base station provides communication



- Standard wireless LAN called IEEE 802.11 popularly known as WIFI
- speed 100 Mbps.

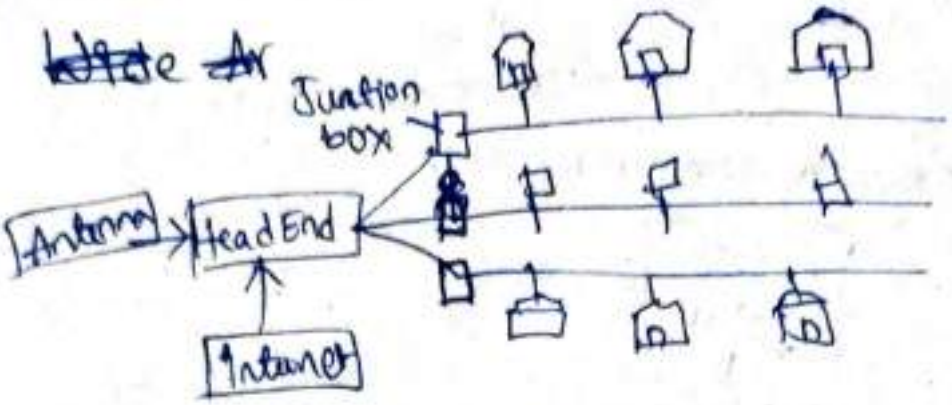
Properties of LAN:-

→ N/w'ed devices are easy to install
→ Expanding the n/w
→ convenience and cost favors wireless technology.

Metropolitan Area Networks:-

→ It covers a city.
→ Ex:- Cable television n/w available in the cities
- Earlier community antenna systems used, & placed on top of the hill & the signals are piped to the users.
- Later entire city wired up for television programming and channels through cables.
- Later Internet signals & television signals are centralized at "cable headend", for subsequent distribution to people's home.

— IEEE 802.16 called as WiMAN



Wide Area Networks:-

— covers large geographical areas, often a country (or) continent.

— Subnet - carry messages from host to host

components :- → Transmission lines :- transfer bits b/n machines; They can be copper wire, optical (or) radio links. (E1 & BSNL cable - other vendors)

→ switching elements; (or) switches :-

— are specialized computers that connect two (or) more transmission lines. It transfers the data from source to destinations. Routers are used as switching elements

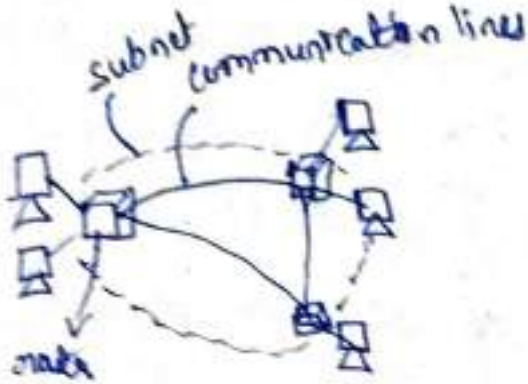
— WAN looks similar to Large LAN's but, Hosts & subnets are operated by different people.

— WAN's forms Internetworks by connecting diff. n/w

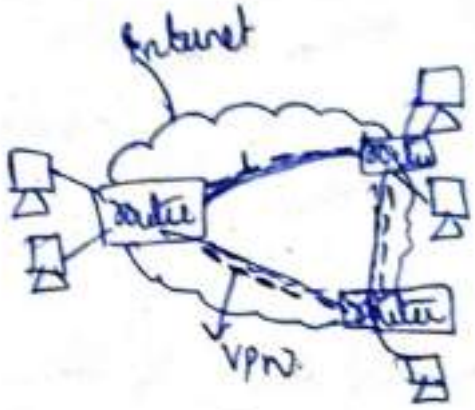
— subnets could connect individual computers for LAN's and it could be entire LAN.

two other varieties of WAN:-

→ Virtual private nw :- Instead of connecting threw wires, an VPN connects to Internet.



WAN with routers & subnet



WAN with VPN

→ ~~Internet Service Provider~~

→ Network Service provider :- Subnets may be given by a diff. company called NSP.

Two (or) subnets are operated & given by a single NSP. called as Internet Service Provider (ISP)

Internetworks/-

- A collection of interconnected networks is called an Internetworking / Internet.
- connecting a LAN and a WAN (or) connecting two LAN's with the help of gateways.

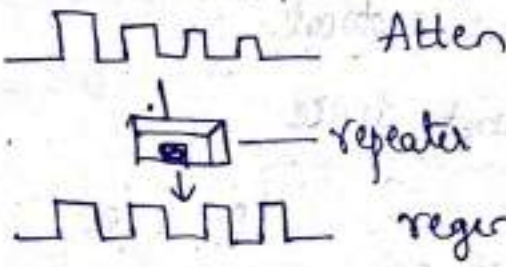
Network Devices :-

- Network devices are physical devices, that are used for communication & interaction b/n hardware of computer networks.
- Devices are :-

① Hub ② Repeater ③ switches ④ router ⑤ bridges ⑥ gateways ⑦ Modems.

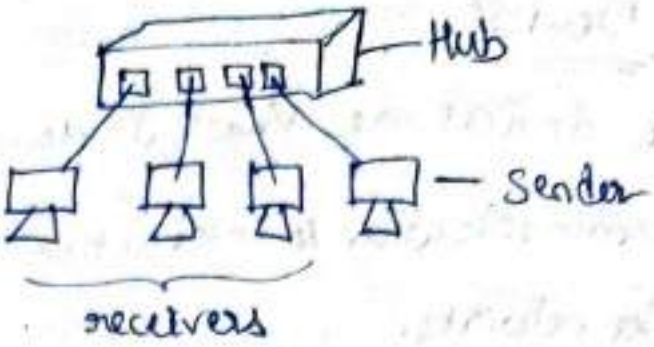
① ~~Repeater~~ - Repeater :-

- Repeaters are operated at physical layer.
- Repeater receive a signal and re-transmits. and it is also used to extend transmissions.
- So that the signal can cover longer distance.
- When the signal becomes weak, they copy the signal bit by bit & regenerate it at original strength.
- 2 port device - ~~Input~~ & output.



Hub :- It is also regenerated the signal. but it is having multiport repeater.

- It is also physical layer device
- Generally used to connect computer in a LAN



Hubs are of 3 types:-

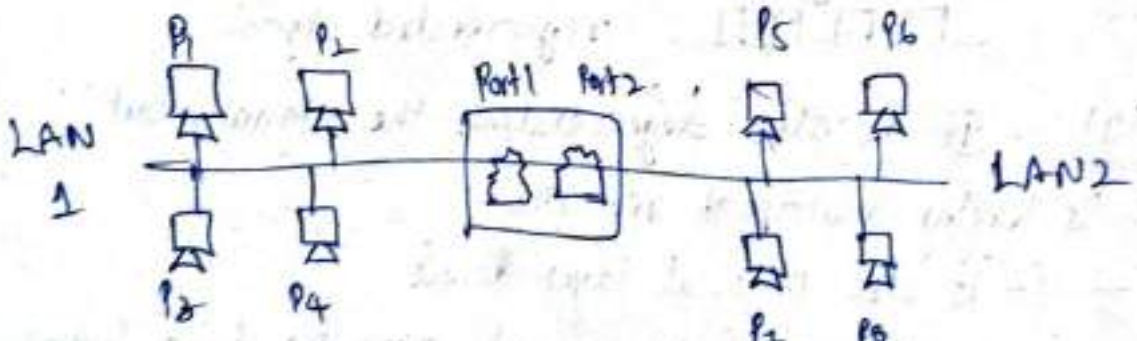
→ Active hubs:- It has own power supply. It works as repeaters

→ Passive hub :- It has power supply from active hub

→ Intelligent hub :- works like active hub & includes remote management capabilities.

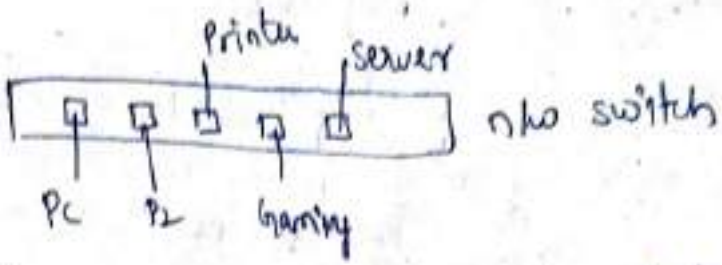
Bridges:- It operates at data link layer

- A bridge is a repeater, with adding functionality of filtering content by reading the mac address of source and dusting.
- It is also used for inter connecting two LAN's working on same protocal.
- Bridge is a 2 port device



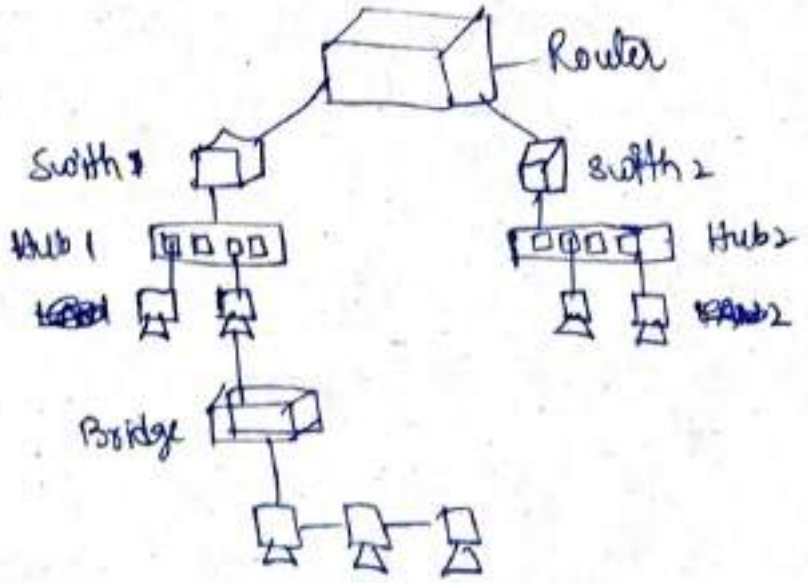
Switch:- It is a data link layer device

- It is a multiport bridge with a buffer and a design that can boost its efficiency & performance
- It can perform error checking before forwarding data and send good packets selectively to correct port only.



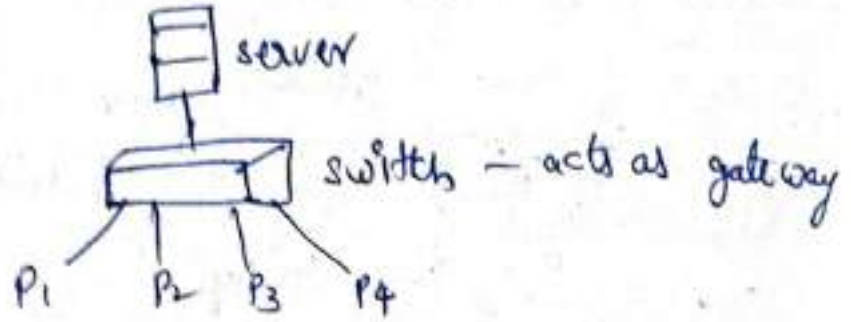
Routers:- It is a device like a switch, that routes the data packets based on the IP address.

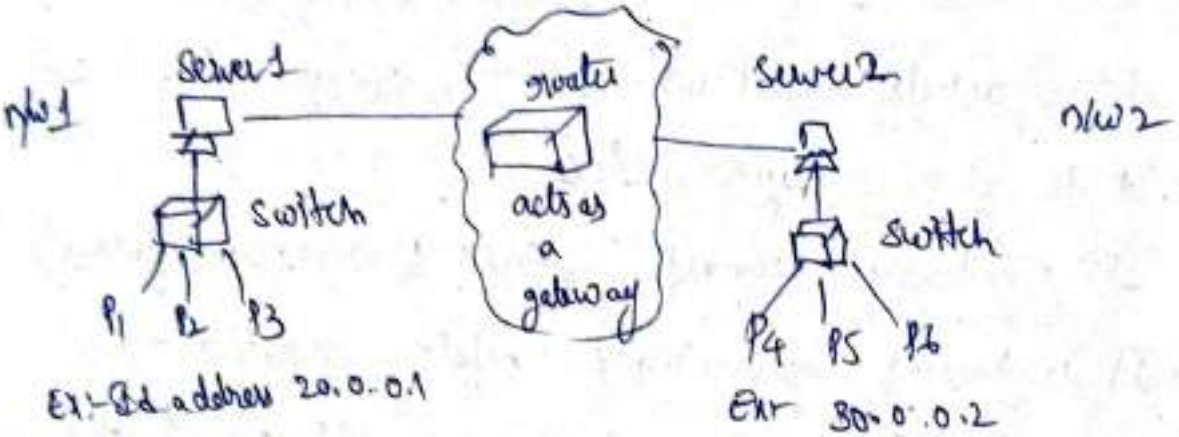
- It is a n/w layer device
- It normally connects LAN's & WAN's together
- It is having dynamically updated routing table.
- Routing table contains all the IP addresses of the n/w devices.



Gateway :- It is a n/w node that connects 2 n/ws with diff. protocols together.

- It is also acts a gate b/n 2 n/w's
- It can be called as router, firewall, server, or other devices
- It enables traffic flow in & out of n/w

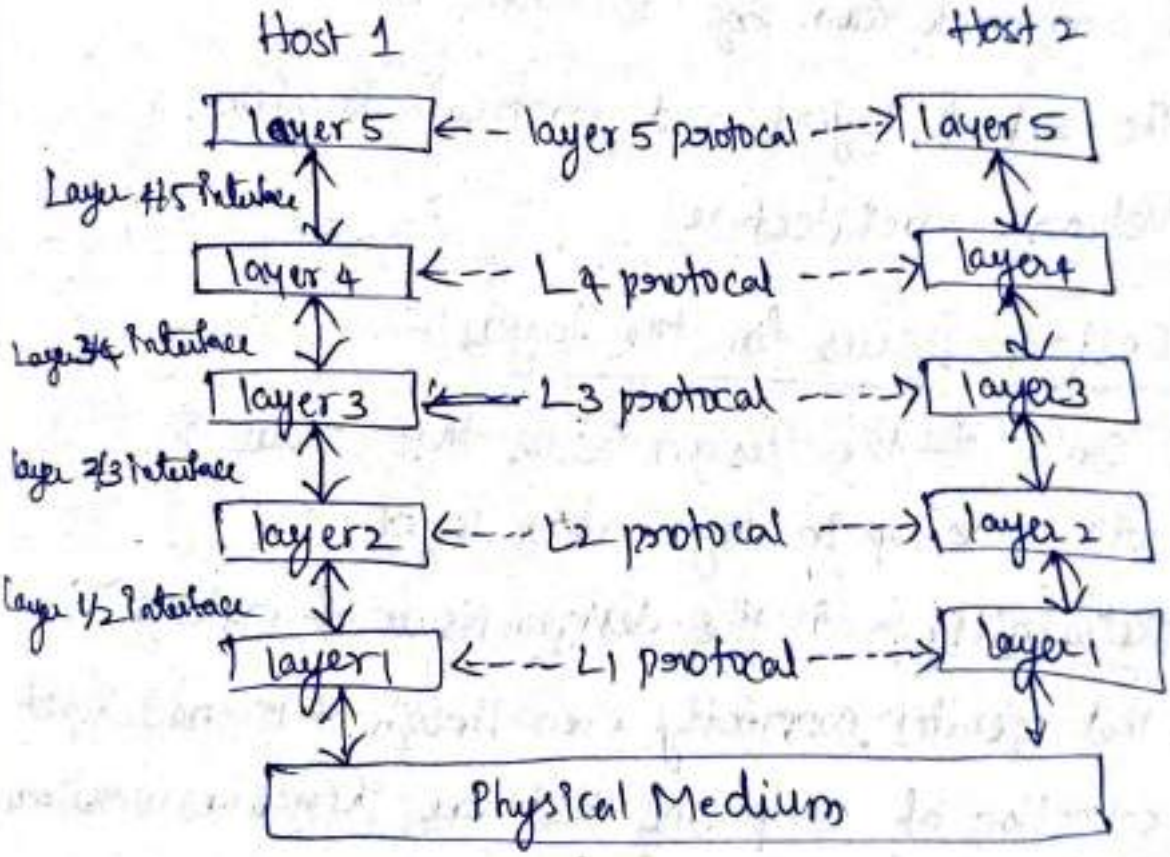




Network software:-

→ → Protocal Hierarchies
→ Design issue for the layers
→ connection oriented v/s connectionless service
→ Service Primitives
→ The relationship of services to protocals.

① Protocal Hierarchies :-



Network Architecture with layers, protocals & Interface.

- To reduce design complexity, n/w's are organized in a layer format.
- Protocal, an aggreement b/n the communicating

parties on how communication is to proceeds.

- Entities on each layer corresponding to layers are called peers. Peers may be s/w process, h/w devices or even human beings.

In other words, it is the peers that communicate by using the protocol to talk to each other.

- Below Layer 1 is physical medium, from which actual communication occurs.
- B/n Each layer pair of layers there is a communication page is called Interface.
- The set of layers and protocols is called a Network Architecture.

① Design Issues for the layers:-

Some of the design issues that occur in CN will come up in layer after layer.

- Reliability :- is the design issue of making a n/w that operates correctly even though it is made up of a collection of components that are themselves unreliable.
- One Mechanism for finding errors in received information is called Error detection.

Information that is incorrectly received can then be retransmitted until it is received correctly. Is called Error correction

-Another reliability issue is finding a working path through a n/w. If any path failure is called routing.

- Second design issue is Evolution of n/w:-

— Hiding implementation mechanism while n/w growth is called protocal layering.

— Each layer need a mechanism for identifying sender and receiver that are involved in message passing called Addressing (or) Naming.

— Designs that continue to work well when the n/w gets large are said to be scalable.

-Third Design issue is resource allocation:-

— N/w is overloading b'coz too many computers want to send too much traffic is called congestion.

-4th Design issue security :=

-securing the n/w by defending it against diff. kinds of threats.

Mechanisms confidentiality, authentication & integrity are used for providing security.

⑧ connection oriented v/s connectionless service:-

→ It is related to telephone system, like a physical connection

⇒ Establish connection.

Terminate connection

⇒

connection oriented is modeled after the telephone system.

To talk to someone
1. Pick up phone
2. dial the no
3. then hangup.

Similarly in CON/N service,
- establish a connection
- use the connection
- Release the connection

connectionless service is modeled after the postal system. Each message carries a full destination address. & each one is routed through the intermediate nodes inside the system. Independent of all the subsequent messages.

There are different names for messages in different contexts!

— A packet :- is a msg at the n/w layer. When the intermediate nodes receive a msg in full before sending it on to the next node is called store and forward switching.

The onward transmission of a message at a node starts before it is completely received by the node is called cut through switching.

	Service	Example
connection oriented	Reliable msg stream	sequence of pages
	Reliable byte stream	Movie Download
connection-less	Unreliable connection	Voice over IP
	Acknowledge datagram	Text messaging
	Request Query	DB Query

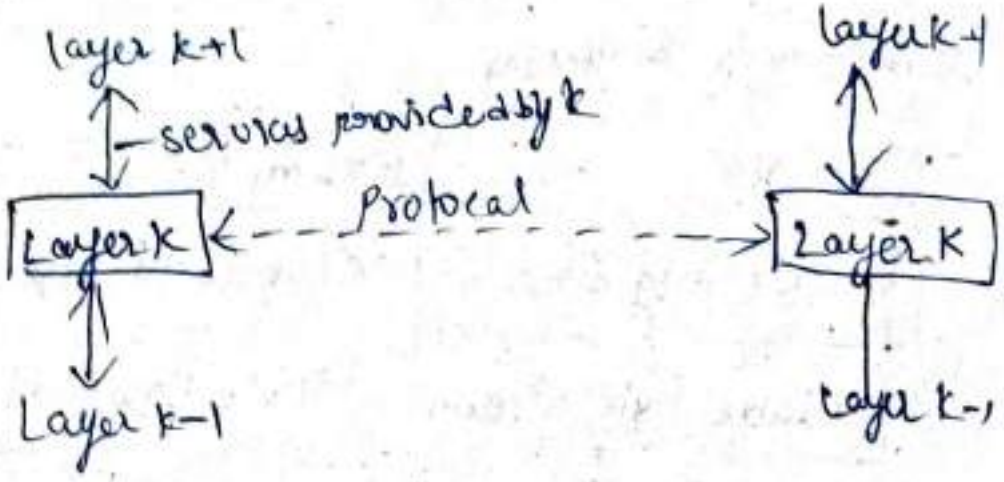
Service Primitives:-

- A service is formally specified by a set of primitives (operations) available to access the service. These primitives tell the service to perform some action (or report on an action taken by a peer entity.

Primitives that provide a simple connection-oriented service.

primitive	Meaning
LISTEN	Block waiting for an incoming
connect	Establish a connection with a waiting peer.
ACCEPT	Accept incoming conn. from a peer
RECEIVE	Waiting for an incoming message
SEND	send a msg to the peer
DISCONNECT	Terminate a connection

The Relationship of Services to protocals :-



OSI Reference Model :-

- Developed by ISO (International Standard org).
- called as ISO OSI (open system Interconnection)
- It hay 7 layers.

Principles of OSI Reference Model :-

1. A layer should be created where a diff. abstraction is needed
2. Each layer perform a well defined function
3. The function of each layer should be chosen with an eye toward defing internationally standardized protocal
4. The layer boundaries should be choses to minimize the information flow across the interfaces
5. The root 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 unwidely.

The Physical layer:-

- transmitting raw bits over a communication channel
- It uses mechanical, electrical & timing interfaces for signals, connection establishment & for time for transmission.
- It uses physical transmission medium which lies below the Physical layer.

The Data link layer:-

- Main task to transform a raw transmission facility into a line that appears free of undetected transmission errors

- It breaks up input data in data frames (typically a few hundred or a few thousand bytes) & transmit the frames sequentially.
- Receiver sends an acknowledgement frame when it gets correct data frame.
- A special sub layer called medium access control sublayer, deals with ~~broadcast~~ access to shared channel control.

→ Network layer :-

- data shared in the form of packets.
- It transfers packets from source to destination
- It also selects the packet routing, i.e. selection of the shortest path to transmit the packet from the no. of o

- Transport Layer :-

- Heart of the OSI model.
- It provides the services to the application layer and get the services from n/w layer.
- The data in Transport Layer called as segments
- It splits the large data & ~~transf~~ send to session layer
- Responsible for end-to-end delivery of data segments.

Session layer:-

- Responsible for establishment of connection, maintanance, authentication, & also ensures security.
- Functions:-
  - → session establishment
  - → synchronization - Identifying errors
  - → Dialog controller - communication with each other

Presentation layer:-

- Functions:-
  - → Translation
  - → Encryption / decryption
  - → compression.

also called as Translation layer.

Data from Application layer extracted & manipulated as per the required formats to transmit over n/w

Application Layer:-

- Functions:
  - → File transfer & access
  - → Email & communication
  - → N/w access

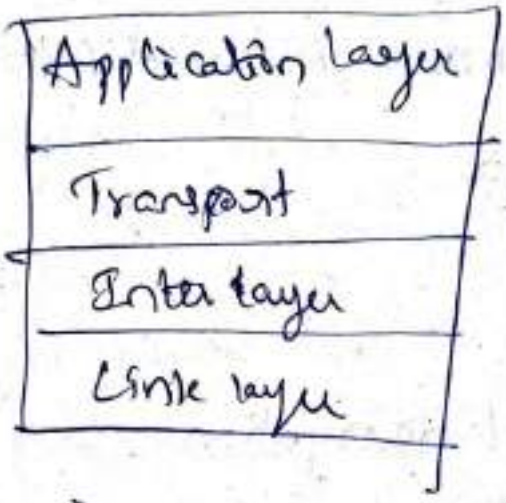
Top of the OSI model

- User can interact. provides services to user
- programs based on client server model.

TCP/IP :-

→ 4 layer architecture
→ popular model
→ Defence project Research Agency (developed by

TCP/IP



Link layer :-

→ lowest layer

Internet layer :-

→ ~~Focus~~: It defines official packet formats protocols called IP & ICMP (Internet control Message protocol)

Transport layer :-

→ TCP – allows byte stream
→ UDP – connection less protocol

Application layer :-

→ FTP, SMTP, HTTP, DNS

ARPANET & Internet :- ARPANET designed by

→ Advanced Research projects Agency (ARPANET).

→ It was the first n/w implement TCP/IP protocol.

→ It was basically beginig of Internet.

→ ARRANET was created for transmission of confidential information of Defence sectors.

→ In the year 1969, advanced research projects agency (ARPA) introduced this n/w, with bunch of PC's of various colleges for sharing information & messages.

→ Later in 1980, It was handover to different milatary departments

→ This ARPANET does not support Scalability of n/w, If n/w size increased, it does n't perform well.

→ This ARPANET is not support updated n/w technologies That why this is not used popularly.

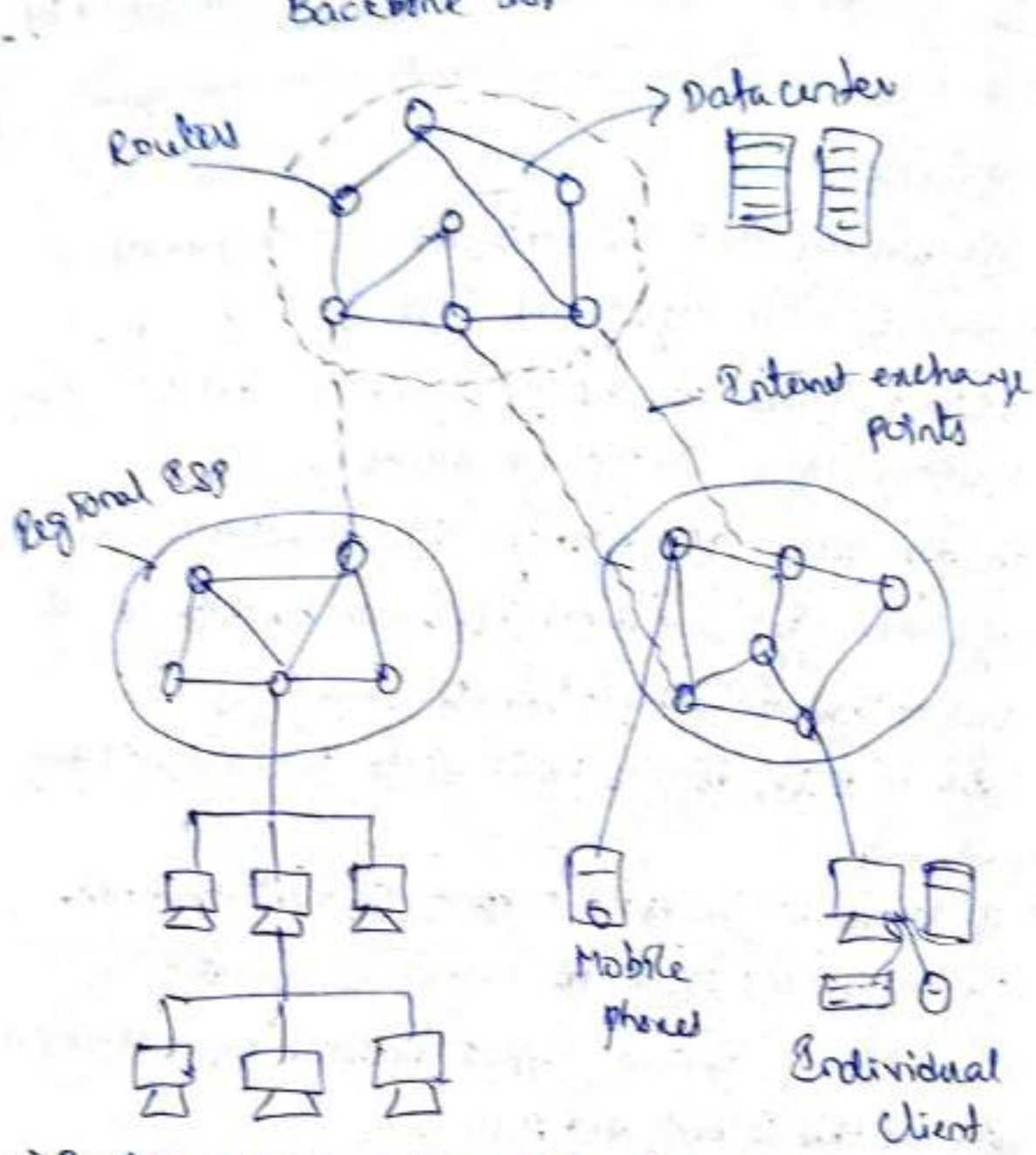
Internet :
- → Backbone ISP
- → Regional ISP
- → clients

} Architecture of Internet

→ connection of different n/w through different regions.

→ Internet is network of multiple networks

→ The architecture of Internet is ever-changing due to continuous changes on the technologies as well as the nature of services.



→ Backbone ISP are large international backbone networks equipped with thousands of routers & data centers (that stores large amount of data).

→ Everyone needs to connect with a Backbone ISP to access the entire Internet.

→ Datacenters are servers & LAN, PC, mobile phones are clients in the Internet

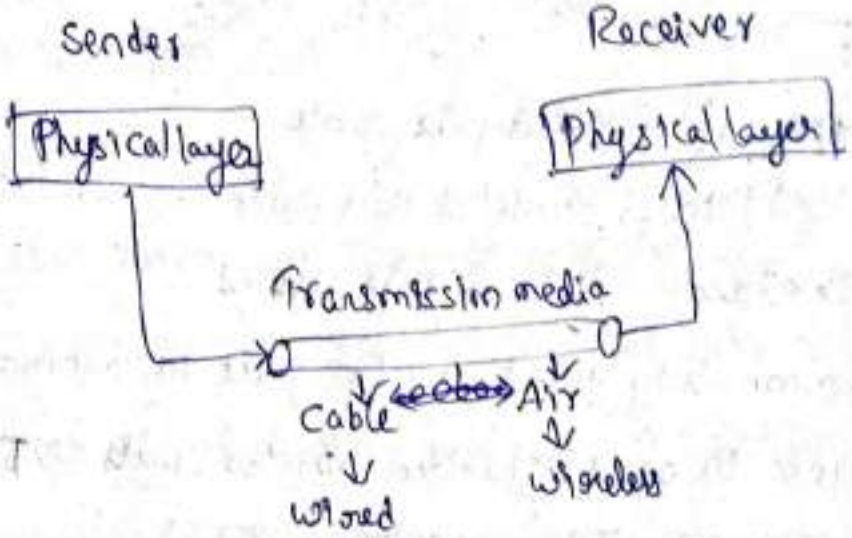
→ Internet is a client-server architecture.

Physical Layer

→ It is the lowest layer of the OSI reference model. It is responsible for sending bits from one computer to another with the help of transmission media.

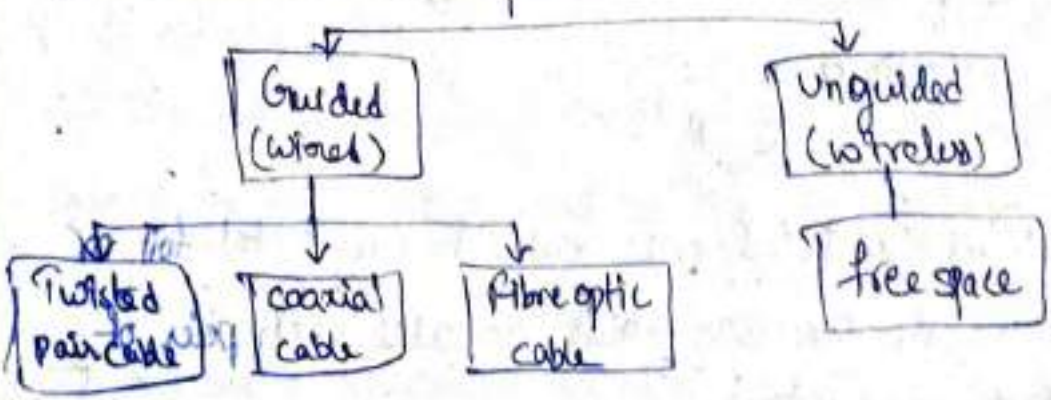
Transmission Media :-

→ To send our data from one place to another place
→ The first layer (Physical layer) of communication n/w in OSI seven layer model is dedicated to transmission media
→ It is a physical path b/n transmitter & receiver
→ Repeaters (or) amplifiers may be used to extend the length of medium.



→ Transmission media may be wired (or) wireless.

Classes of Transmission media :-

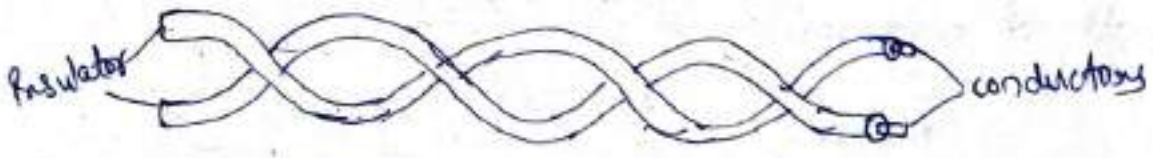


Guided Media :- Guided media, which are those that provide a medium from one device to another, include twisted-pair cable, coaxial cable and fiber-optic cable.

- Twisted pair Cable :-

— A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together.

— one of the signals to the receiver, and the other is used only as a ground reference



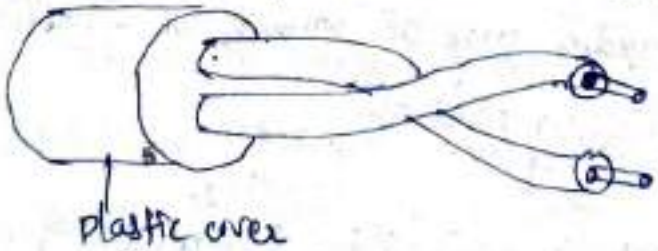
2 types of Twisted pair cable :-

→ Unshielded Twisted pair cable

→ Shielded Twisted pair cables

* Most commonly used twisted pair in communication is referred to as Unshielded twisted pairs (UTP).

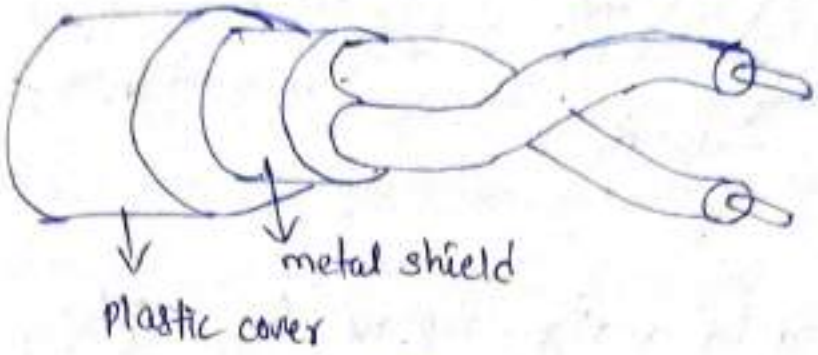
* Most common utp connector is RJ45.



UTP

* Shielded Twisted pair cable has a metal foil or braided mesh covering that encases each pair of insulated conductors.

- metal casing improves the quality of cable by preventing the penetration of noise (or) crosstalk, it is bulkier and more expensive.



STP

Applications of Twisted pair:-

→ These cables are used in telephone lines to provide voice & data channels.

→ used in LAN.

- Coaxial Cable:-

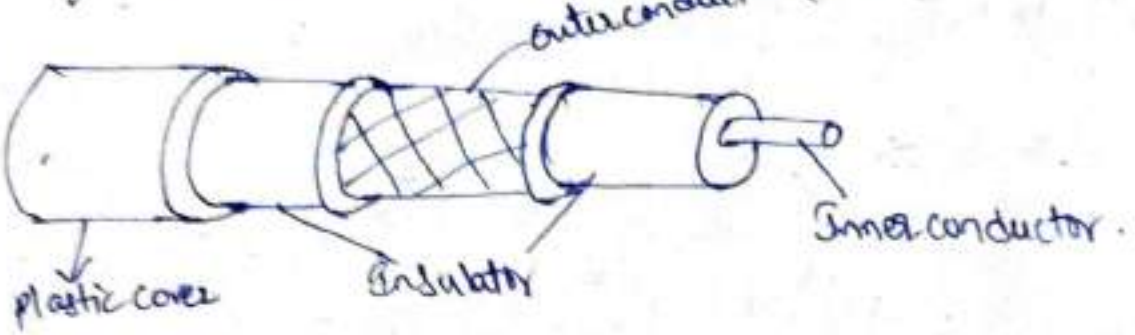
→ Coaxial cable (or) coax carries signals of higher frequency ranges than those in twisted pair cable.

→ coax has a central core conductor of solid (or) standard wire (usualy copper) enclosed in an insulating sheath, which is in turn, encased in an outer conductor of metal foil, braid (or) a ~~communication~~ combination of the two

→ The outer metalic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit.

→ This outer conductor is also enclosed in an

insulating sheath, and the whole cable is protected by a plastic cover.



Applications :-

→ widely used in analog telephone n/w's, digital telephone n/w's

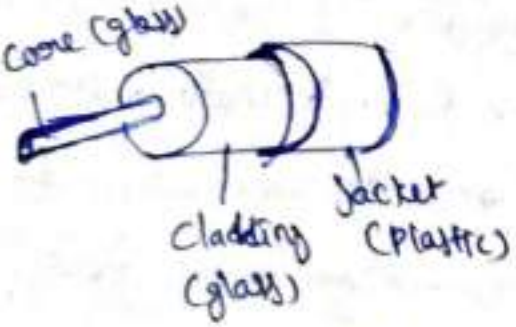
→ cable TV

→ Traditional Ethernet LANs.

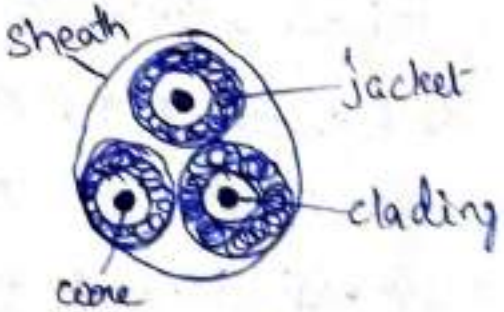
Fiber-Optic Cable :-

→ It is made of glass (or) plastic and transmits signals in the form of light.

→ Light travels in a straight line as long as it is moving through a single uniform substance.



Side view of fiber



end view of a sheath with 3 fibers

Applications :-

- often found in backbone n/w's bcoz its wide bandwidth is cost effective.
- some cable TV companies use a combination of

optical fiber and coaxial cable, thus creating a hybrid n/w.

Advantages of fiber optics:-

→ High bandwidth.

→ Less signal regeneration. A signal can run for 50 km without repeater.

→ Resistance to corrosive materials.

→ Light weight

Disadvantages:

→ Propagation of light is unidirectional. If we need bidirectional communication, 2 fibers are needed.

→ The cable and interfaces of fiber are very costly.

Unguided Media:-

→ Wireless communication:- Unguided media transport electromagnetic waves without using a physical conductor. This type of communication is often referred to as wireless communication.

→ Radio Waves

→ Micro waves

→ Infrared.

→ Wireless signals are spread over in the air and are received and interpreted by appropriate antennas.

→ When an antenna is attached to electrical circuit of a computer or wireless device, it converts the digital data into wireless signals and spread all over

within its frequency range.

The receptor on the other end receives these signals and converts them back to digital data.

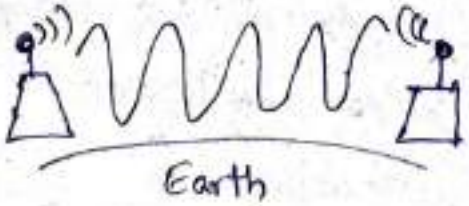
-Radio Transmission (or) Waves!-

-Radio frequency is easier to generate & because of its large wavelength it can penetrate through walls & structures.

-Radio waves can have wavelength from 1mm - 100m and have frequency ranging from ~~3kHz~~ to 300GHz

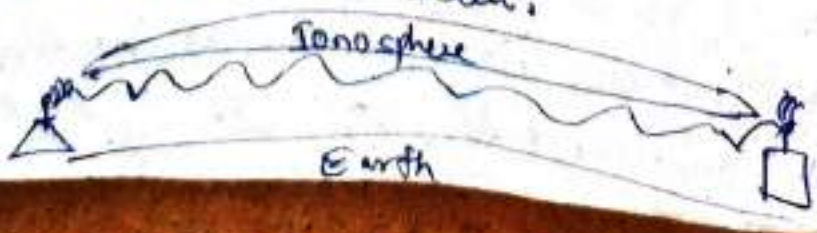
-Radio waves at lower frequencies can travel through walls whereas higher Radio Frequency can travel in straight

- Lower frequencies such as VLF, LF, MF bands can travel on the ground upto 1000km, over the earth surface



- Radio waves of high frequencies are prone to be absorbed by rain and other obstacles. They are reflected by ionosphere of earth atmosphere.

-High frequency radio waves such as HF & VHF bands are spread upwards. When they reach Ionosphere, they refracted back to the earth.

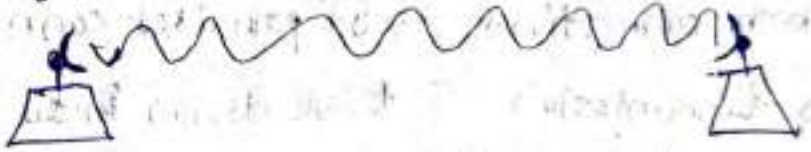


Microwave Transmission :-

→ Electromagnetic waves above 100 MHz to travel in a straight line & signals over them can be sent by beaming those waves towards one particular station.

→ Because microwaves travels in straight lines, both sender and receiver must be aligned to be strictly in line of sight. Repeaters are needed for longer distance.

→ Microwaves can have wavelength ranging from 1 mm - 1 meter and frequency ranging from 300 MHz to 300 GHz.



→ Microwave antennas concentrate the waves making a beam of it., as shown in picture above, multiple antennas can be aligned to reach target

→ Microwaves have higher frequencies & do not penetrate wall like obstacles

Infrared Transmission :-

→ Infrared waves lies in b/n visible light spectrum of microwaves. ↑

→ It has wavelength of 780 nm to 1 mm, & frequency ranges from 300 GHz to 400 THz.

→ Infrared waves used for very short range communication.

→ Ex: Television & its remote.

→ It travels in straight line hence it is unidirectional

→ Difference b/n OSI & TCP/IP Reference Model:-

OSI	TCP/IP
→ OSI stands for open system Interconnection	→ TCP/IP stands for Transmission control protocol / Internet protocol.
→ OSI is protocol independent standard.	→ TCP/IP depends on protocols
→ OSI model developed first, then protocols were created to fit the architecture needs	→ In TCP/IP protocols were created first & then built the TCP/IP model.
→ It provides both connection & connectionless transmission in the n/w layer, however only connection oriented transmission in the transport layer.	→ It provides connectionless transmission in the n/w layer supports connection & connectionless oriented transmission in the transport layer.
→ It uses horizontal approach	→ It uses vertical approach
→ It is a 7 layer standard	→ It is a 4 Layer Structure
→ It mention services, Interfaces & protocols	→ It doesn't provide services, Interfaces & protocols.

UNIT-II Syllabus

Data Link Layer:-

→ Design Issues

→ framing

→ Error Detection & correction.

Elementary datalink protocols

→ Simplex protocol

→ A Simplex stop & wait protocol for an error-free channel.

→ A simplex stop & wait protocol for noisy channel.

Sliding Window Protocols:-

→ A one bit sliding window protocol.

→ A protocol using GO-Back-N

→ A protocol using selective Repeat

Example Data link protocols

A Medium Access control Sublayer

→ The channel allocation problem

→ Multiple Access protocols

→ ALOHA

→ carrier sense multiple Access protocols,

→ collision free protocols.

→ Wireless LAN's

→ Data Link Layer Switching

Data Link Layer :-

→ Data link layer is the second layer after physical layer.

→ The DLL is responsible for maintaining the data link b/n two hosts (or) nodes.

→ The DLL uses the services of the physical layer to send and receive bits over communication channel.

It has a no. of functions (Design issues) including :

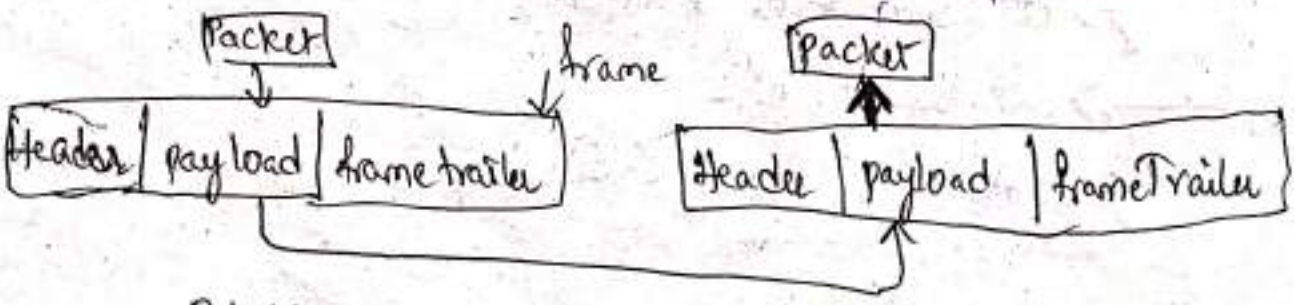
① providing a well-defined service interface to the n/w layer.

② Dealing with transmission errors

③ Regulating the flow of the data so that slow receivers are not swamped by fast senders.

To accomplish these goals, the data link layer takes the packets it gets from the n/w layer and encapsulates them into frames for transmission.

Each frame contains a frame header, a payload field for holding the packet, & a frame trailer.



Relationship b/n packets & frames

UNIT-II Syllabus

DataLink Layer:-

→ Design Issues

→ framing

→ Error Detection & correction.

Elementary datalink protocals

→ Simplex protocal

→ A simplex stop & wait protocal for an error-free channel.

→ A simplex stop & wait protocal for noisy channel.

Sliding Window Protocols:-

→ A one bit sliding window protocol.

→ A protocol using Go-Back-N

→ A protocol using selective Repeat

Example Datalink protocols

A Medium Access control Sublayer

→ The channel allocation problem

→ Multiple Access protocols

→ ALOHA

→ carrier sense multiple Access protocols,

→ collision free protocols.

→ Wireless LAN's

→ DataLink Layer Switching

Data Link Layer :-

→ Data link layer is the second layer above physical layer.

→ The DLL is responsible for maintaining the data link b/n two hosts (or) nodes.

→ The DLL uses the services of the physical layer to send and receive bits over communication channel.

Design issues

It has a no. of functions & including :

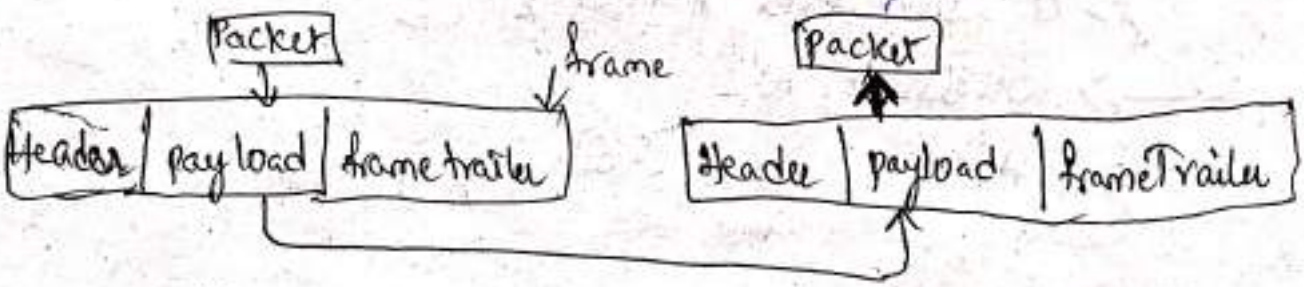
① providing a well-defined service interface to the n/w layer.

② Dealing with transmission errors

③ Regulating the flow of the data ∴ so that slow receivers are not swamped by fast senders.

To accomplish these goals, the data link layer takes the packets it gets from the n/w layer and encapsulates them into frames for transmission.

Each frame contains a frame header, a payload field for holding the packet, & a frame trailer.



Relationship b/n packets & frames

Design Issues :-

① → services provided to n/w layer
② → Framing
③ → Error control
④ → Flow control

① Services provided to n/w layer :-

→ The function of the datalink layer is to provide services to the n/w layer.

→ The principle service is transferring data from the n/w layer on the source machine to the n/w layer on the destination machine.

→ The datalink layer can be designed to offer various services :-

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

① Unacknowledged connectionless service consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them.

Ethernet is the good example of a DLL that provides this class of service.

No If a frame is lost due to noise on the line, no attempt is made to detect the loss (or) recover

from it in the data link layer.

② In acknowledged connectionless service, there are still no logical connections used, but each frame sent is individually acknowledged.

In this way the sender knows whether a frame has arrived correctly (or) been lost. If it has not arrived within a specified time interval, it can be sent again.

This service is useful over wireless systems.

Ex:- 802.11 (WiFi)

③ The most sophisticated service the DLL can provide to the nw layer is a acknowledged connection-oriented service.

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

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

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

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

→ In the first phase, the connection is established

by having both sides initialize variables and ~~having both sides~~ counters needed to keep track of which frames have been received and which ones have not.

→ In the second phase, one or more frames are actually transmitted.

→ In the third and final phase, the connection is released, freeing up the variables, buffers, & other resources used to maintain the connection.

② Framing :-

→ To provide service to the n/w layer, the DLL must use the service provided to it by the physical layer.

→ What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination

If the channel is noisy, as it is for most wireless and some wired links, the physical layer will add some redundancy to its signals to reduce the bit error rate to a ~~value~~ tolerable level.

→ However, the bit stream received by the DLL is not guranteed to be error free.

→ Some bits may have diff. values and the no of bits received may be less than, equal to, or more than the no. of bits transmitted. It is upto the DLL to detect & if necessary, correct errors.

→ The usual approach is for the DLL to break up the bit stream into discrete frames, compute a short token called a checksum for each frame, and include the checksum in the frame when it is transmitted.

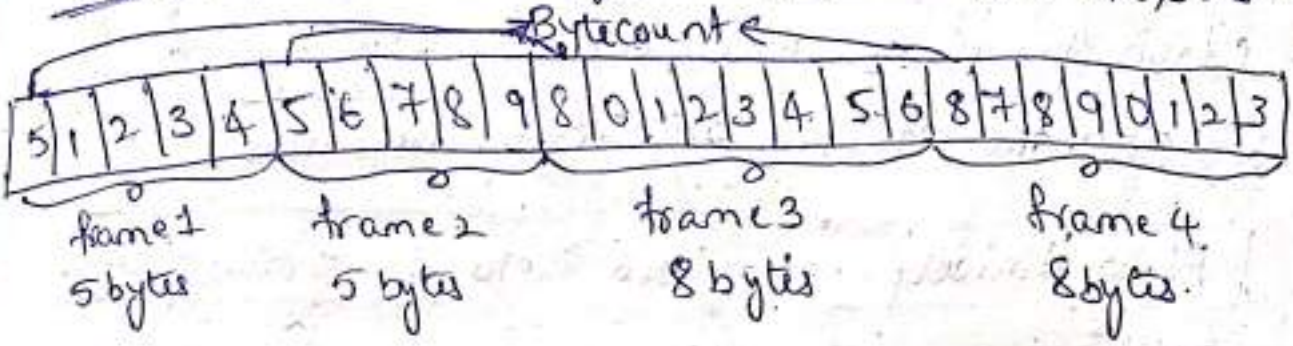
When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the DLL knows that an error has occured and takes steps to deal with it.

Breaking up bit streams into frames, is more difficult than it at first appears. A good design must make it easy for a receiver to find the start of new frames while using little of the channel bandwidth. We will look at four methods:

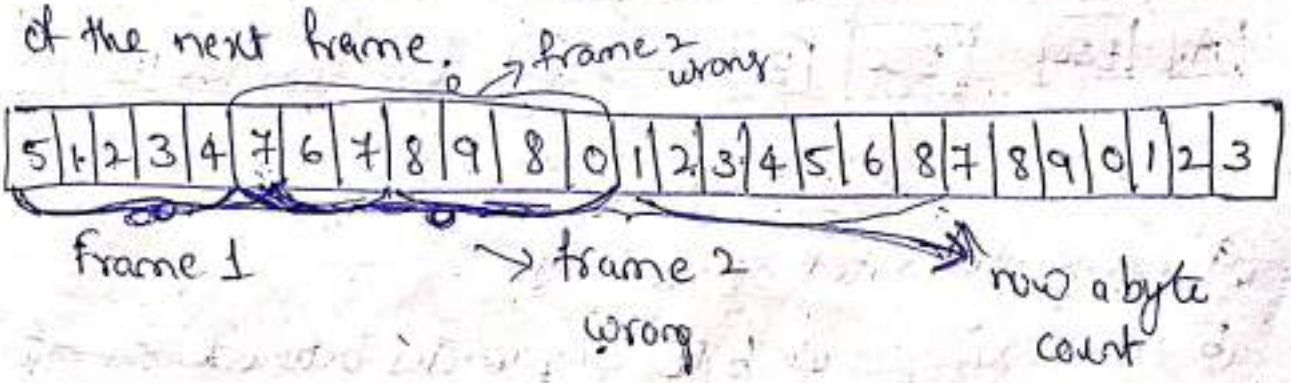
1. Byte count
2. Flag bytes with byte stuffing
3. Flag bits with bit stuffing
4. Physical layer coding violations.

① Byte Count:- The first framing method uses a field in the header to specify the no. of bytes in the frame. When the DLL at the destination sees the byte count, it knows how many bytes follow and hence where the end of the frame is

Ex:(a) 4 small Example frames of sizes 5, 5, 8 & 8.



The trouble with this algorithm is that the count can be garbled by a transmission error. For example: If the byte count of 5, in the second frame of below example becomes a 7 due to a single bit flip, the destination will get out of synchronization. It will then be unable to locate the correct start of the next frame.

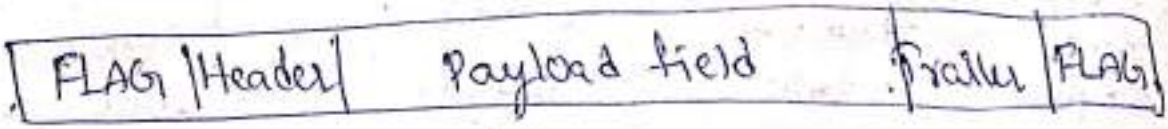


-sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many bytes to skip over to get to the start of the retransmission. For this reason the byte count method is rarely used by itself.

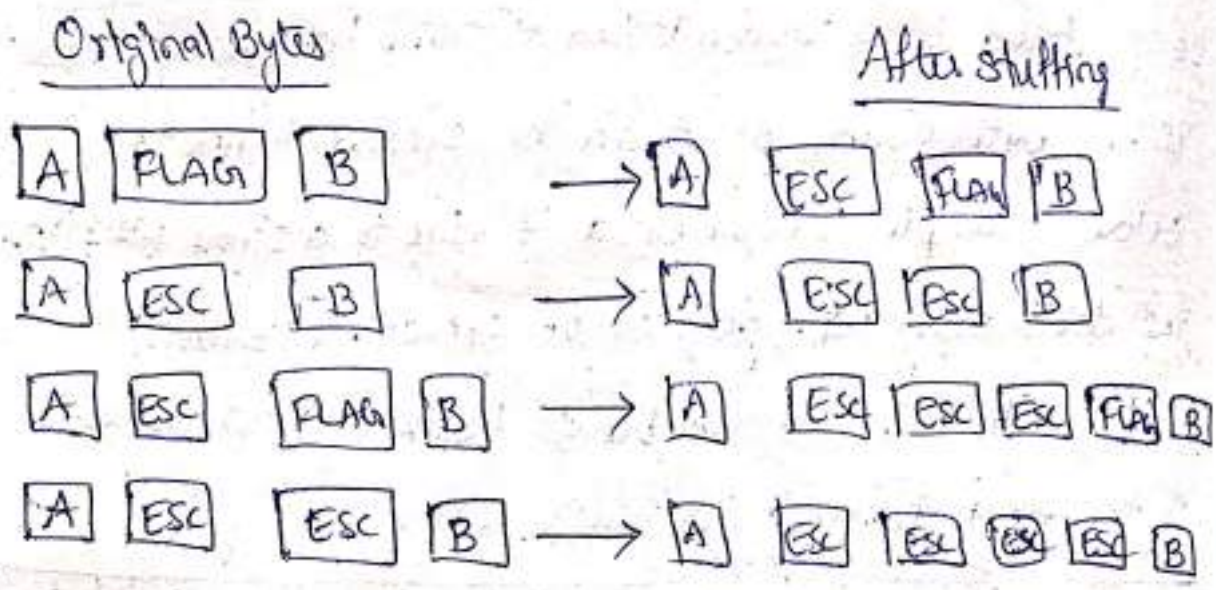
Flag bytes with byte stuffing:-

The second framing method is to ~~gets~~ solved: The problem of resynchronization after an error by having

each frame start and end with special bytes. Often the same byte, called a flag byte, is used as both the starting and ending delimiter.



(a)



(b)

(a) Frame delimited by flag bytes.

(b) Four examples of byte sequencies before & after byte stuffing

→ Two consecutive flag bytes indicate the end of one frame & the start of the next frame. Thus, if the receiver ever loss synchronization, it can just search for two flag bytes, to find the end of the current frame & the start of the next frame.

→ If flag bytes occurs within data, to solve this problem, the sender DLL insert a special escape byte

(ESC) just before each accidental flag byte in the data. Thus a framing flag byte can be distinguished from one in the data by the absence (or) presence of an escape byte before it.

The datalink layer on the receiving end removes the escape bytes before giving the data to the n/w layer. This technique is called Byte stuffing.

→ If an escape byte occurs in the middle of data, it is stuffed with an escape byte. At the receiver, the first escape byte is removed, leaving the databyte that follows. ie. [[Ex: diagram (b)]]

Flag bits with bit stuffing:-

- Each frame begins & ends with special bit pattern called as a flag byte
- When the sender's datalink layer encounters five consecutive 1's in the data, it automatically stuffs a '0' (zero) bit into the outgoing bitstream. This is called bit stuffing.
- When the receiver sees five consecutive incoming 1 bits, followed by a '0' (zero) bit, it automatically destuffs (i.e., deletes) the '0' bit.

Original data:- 01101111111111111110010

data stuffing : 011011110111110111110010010

Stuffed bits

Receiver's end :
data destuffing : 01101111111111111110010

Physical layer coding violations :-

- at physical layer signals includes redundancy data, ~~to~~ ~~original data~~. means some signals will ~~not~~ occurs in regular data.
- We can use some reserved signals to indicate the start & end of frames.
- In effect we are using "coding violations" to delimit frames.
- The beauty of this ~~is~~ scheme is that, because they are reserved signals, it is easy to find the start & end of frames and there is no need to stuff the data.

③ Error Control :-

The frames from DLL ~~need~~ surely delivered to the n/w layer, at the destination & in the proper order.

For unacknowledged connectionless service it might be fine if the sender just kept outputting frames without regard to whether they were arriving

properly.

But for reliable connection-oriented service it would not be fine at all.

The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the other end of the line. If the sender receives a positive acknowledgement about a frame, it knows the frame has arrived safely. If negative acknowledgement means that something has gone wrong and the frame must be transmitted again.

If hadware troubles may cause a frame to vanish completely. In this case, timers are added into the DLL. When the sender transmits a frame, it generally also starts a timer. The timer is set to expire after an interval long enough for the frame to reach the destination, be processed there, & have the acknowledgement propagate back to the sender. The frame will be received correctly and get acknowledgement back before the timer runsout, then the timer will be cancelled. However, if either the frame or the acknowledgement is lost, the timer will go off, alerting the sender to a

potential problem. The obvious solution is to just transmit the frame again.

If the frames transmitted multiple times & the receiver accept the same frame two or more times and pass it to the n/w layer more than once. To overcome this, assign sequence numbers to outgoing frames, so that the receiver can distinguish retransmitted from originals.

④ Flow control :-

Another important design issue that occurs in the data link layer is what to do with a sender that wants to transmit frames faster than the receiver can accept them. This situation can occur when the sender is running on a fast, powerful computer and the receiver is running on a slow, low-end machine.

Two approaches are commonly used :

→ Feedback-based flow control, the receiver sends back information to the sender giving it permission to send more data, or atleast telling the sender how the receiver is doing.

→ Rate based flow control :- the protocol has a built-in mechanism that limits the rate at which

senders may transmit data, without using feedback from the receiver.

Error Detection and Correction;-

In networks errors are occured while the data is on transmission.

Two strategies are designed for dealing errors:

→ One strategy is to include enough redundant information to enable the receiver to deduce what the transmitted data must have been.

→ The other is to include only enough redundancy to allow the receiver to deduce that an error has occurred. and have it request a retransmission.

The former strategy uses error-correcting codes and the latter uses error-detecting codes.

The use of error-correcting codes is often referred to as forward Error Correction (FEC)

Error Correcting Codes :- 4 diff. error correcting codes:

1. Hamming codes
2. Binary convolutional codes
3. Reed-solomon codes
4. Low density parity check codes

All of these codes add redundancy to the information that is sent.

A frame consists of 'm' data (i.e, message) bits and 'r' redundant (i.e, check) bits. And the total length of frame block be n, i.e, $n = m + r$.

Hamming code Error correction:-

→ In this method, extra bits are appended to the message which are used by the receiver to correct the errors.

Example:-

Suppose sender wants to transmit the message whose bit representation is '1011001'.

In this message:

- Total number bits (m) = 7
- Total of redundant bits (r) = 4 (This is because the message has four 1's in it)
- Thus, total bits $n = m + r = 7 + 4 = 11$

In Hamming code, the redundant bits are always placed in the places which are powers of 2.

Format shown below:-

$R_1 R_2 I_3 R_4 I_5 I_6 I_7$ R are redundant bits

R1, R2, R3, R4 are redundant bits (1011001 – data

–	–	–	2^3	–	–	–	2^2	–	2^1	2^0	Power of 2
1	0	1	R4	1	0	0	R3	1	R2	R1	msg
11	10	9	8	7	6	5	4	3	2	1	n = m+r

Therefore we have R_1, R_2, R_3 and R_4 as redundant bits which will be calculated according to the following:

– R_1 includes all the positions whose binary representation has 1 in their least significant bit. Thus, R_1 covers positions 1, 3, 5, 7, 9, 11.

– R_2 includes all the positions whose binary representation has 1 in the second position from the least significant bit. Thus, R_2 covers positions 2, 3, 6, 7, 10, 11.

– R_3 includes all the positions whose binary representation has 1 in the 3rd position from the LSB. Hence R_3 covers positions 4, 5, 6, 7.

– R_4 includes all the positions whose binary representation has 1 in the 4th position from the LSB. Thus R_4 covers position 8, 9, 10, 11

These rules are illustrated below:

Position	R_4	R_3	R_2	R_1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1

Now we calculate the value of R_1, R_2, R_3 & R_4:

→ Since the total no. of 1's in all the bits position for R_1 is 4, is a even number, so $R_1 = 0$

→ The total no. of 1's in all bit positions for R_2 is 3, is a odd number, $R_2 = 1$

→ The total no. of 1's in all bit positions for R_3 is 1, is a odd number so $R_3 = 1$

→ The total no. of 1's in all bit positions for R_4 is 2, is a even number, so $R_4 = 0$

Therefore the msg's to be transmitted becomes:-

R_1, R_2, R_3, R_4 are redundant bits:

—	—	2^3	2^3	—	—	2^2	2^2	—	2^1	2^0	Power of 2
1	0	1	0	1	0	0	1	1	1	0	msg sequence
11	10	9	8	7	6	5	4	3	2	1	Total bits

The message is :

1010100 1 110.

10101000110

⇒ If the msg is transmitted at the receivers end a bit 6 becomes corrupted and changes to 1, Then the msg becomes 10101101110. So at the receiver's end, the no. of 1's in the respective bit positions of R1, R2, R3 & R4 is checked to correct the corrupted bit. This is done in the following steps. For all the bits we will check the.

★ For R1 : bits 1, 3, 5, 7, 9 & 11 are checked we can see that the no. of 1's in these bit positions is 4(even) so R1 = 0

★ For R2 : bits 2, 3, 6, 7, 10, 11 are checked. You can observe that the no. of 1's in these bit positions is 5 (odd) so we get a R2 = 1.

★ For R3 : bits 4, 5, 6, & 7 are checked. we see that the no. of 1's in these bit positions is 3 (odd). So R3 = 1

★ For R4 : bits 8, 9, 10, 11 are observed. Here, the no. of 1's in these bits positions is 2 & that's even so R4 = 0.

If we observe, they give the binary number 0110 as the redundant bits, whose decimal representation is 6. Thus bit 6 contain error

To correct the error the 6th bit is changed from 1 to 0 to correct the error.

Binary Convolutional codes :-

→ In convolutional codes the message comprises of data streams of length and sequence of output bits are generated by the application of Boolean functions to the data stream.

→ In convolutional codes, output stream depends not only the present I/p bits but also only previous I/p bits stored in memory.

→ For generating a convolutional codes, the information is ~~coded~~ passed sequentially through a linear finite-state shift register. The shift register comprises of stages and Boolean function generators

A convolutional code can be represented as (n, k, K) where.

→ K is the no. of bits shifted into the encoder at one time. & Generally K=1

→ n is the no. of encoder output bits corresponding to k information bits.

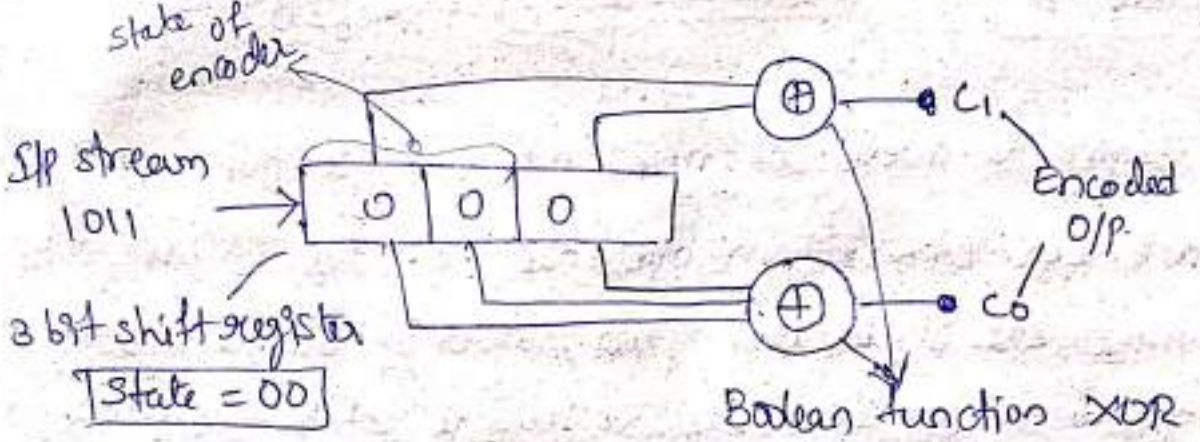
→ The code rate $R_c = k/n$

→ n is a function of the present I/p bits and contents of K

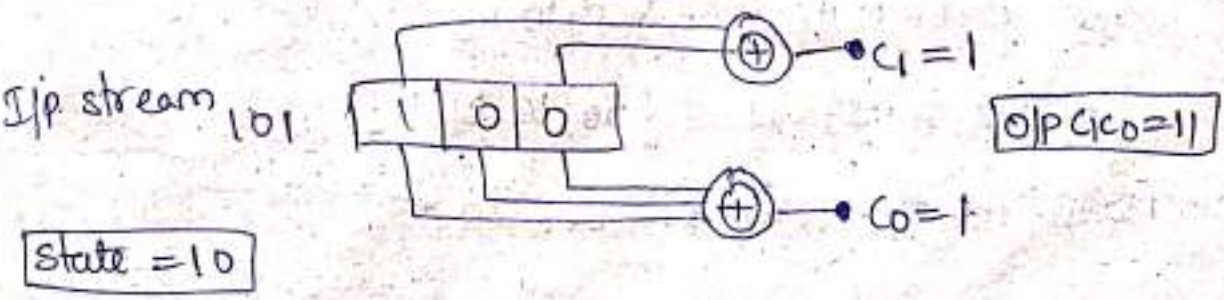
→ The state of encoder is given by the value of (K-1) bits.

Example of Generating a convolutional Code:-

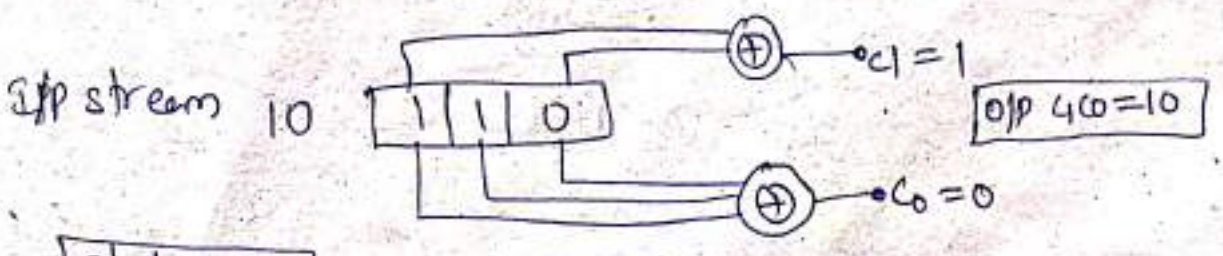
- Let us consider a convolutional encoder with K=1, n=2 and K = 3. & code rate $R_c = K/n = 1/2$.
- The Input stream is 1011.
- Boolean function used is XOR operation.



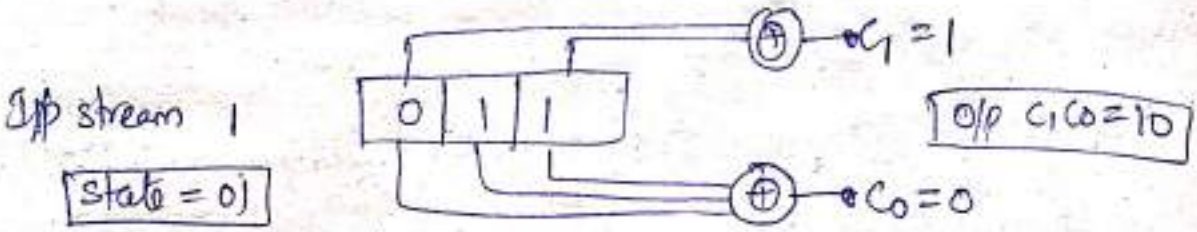
The I/p string is streamed from right to left into the encoder. When the first bit 1, is streamed in the encoder, the contents of encoder will be -



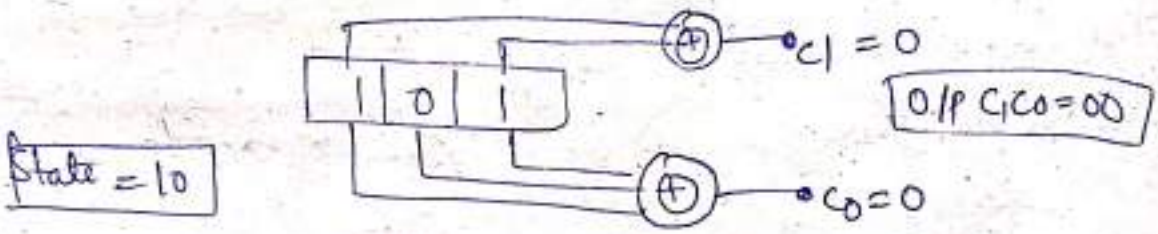
When the next bit 1 is streamed in the encoder,



When the next bit 0 is streamed,



When the last bit 1 is streamed,



From the above example, we can see that any particular binary convolutional encoder is associated with a set of binary inputs, a set of binary o/p's and a set of states. Above are the state transition diagram with state table.

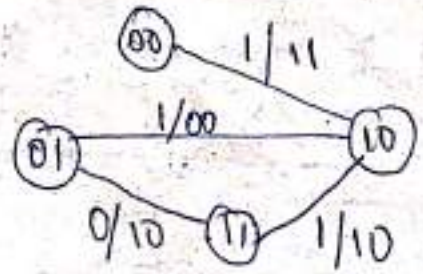
For given example the binary convolutional codes:

The set of inputs = {0,1}

The set of outputs = {00, 10, 11}

The set of states = {00, 10, 01, 11}

Transition diagram will be;



Reed-Solomon Codes:-

Reed-solomon codes operates on 'm' bit symbols. Naturally mathematics are more involved. Reed-solomon encoder accepts a block of data and adds redundant bits before transmitting it over noisy channel. on receiving the data, a decoder corrects the error depending upon the code characteristics.

- A Reed-solomon code is specified as RS(n,k).
- Hence, n is the block length which is recognizable by symbols, holding the relation $n = 2^m - 1$.
- The msg size is k bits
- Parity check size is $(n-k)$ bits.
- Code can correct upto 't' errors in a codeword, where $2t = n-k$.
- Generator function is $g(x) = (x-a)(x-a^2)(x-a^3) \cdots (x-a^{2t})$

n bits

Message of k bits	Parity of 2t bits

Low density parity check (-

→ LDPC code, each output bit is formed from only a fraction of the input bits. This leads to a matrix representation of the code that has a low density of 1's hence the name from the code.

The received code words are decoded with

an approximation algorithm that iteratively improves on a best fit of the received data to a legal code word. This corrects errors.

Error detecting codes :-

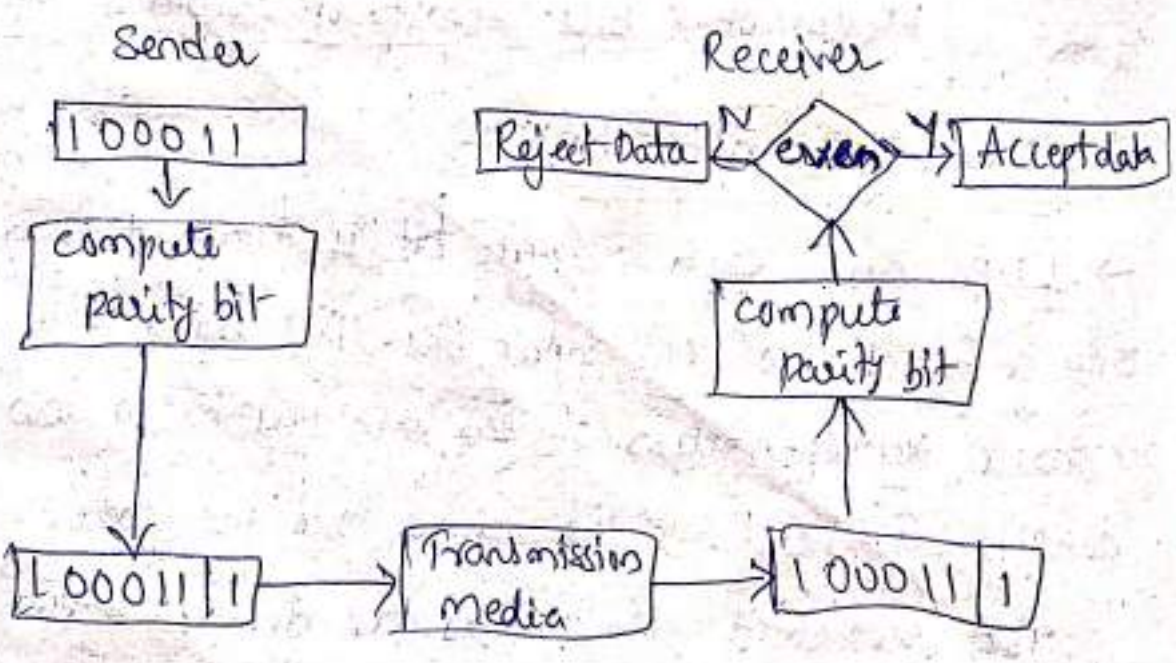
① Parity
② Checksums
③ Cyclic Redundancy Checks (CRC's).

Parity :-

Simple bit parity is a simple error detection method that involves adding an extra bit to a data transmission. It works as:

→ 1 is added to the block if it contains an odd number of 1's and

→ 0 is added to the block if it contains an even number of 1's.



This scheme makes Total no of 1's even, that is why it is called even parity checking

Parity bits are calculated for columns, then both are sent along with the data. At the receiving end, these are compared with the parity bits calculated on the received data.

→ Transmit order

1001110
1100101
1110100
1110111
1101111
1110010
1101011
1011110 → Parity bits

→ channel

→ If errors in data

1001110
1100011 → Burst errors
1101100
1110111
1101111
1110010
1101011
1011110 — Parity errors

Checksum :-

→ In checksum error detection scheme, the data is divided into 'k' segments & each segment contain 'n' bits.

→ Data is represented as 'm' bits.

→ At senders side, All the segments are added using 1's complement arithmetic to get the sum.

→ The sum is complemented to get the checksum

→ Now the checksum is sent along with the data segments to the receiver.

Frames

10011001	11100010	00100100	10000100
1	2	3	4

Sender

① 10011001
② + 11100010
① 01111011
+ 1
01111100
③ + 00100100
10100000
④ + 10000100
① 00100100
+ 1
00100101

1st complement = 11011010

data + 1's complement

Receiver

① 10011001
② + 11100010
10011 1011
+ 1
01111100
③ + 00100100
10100000
④ + 10000100
100100100
+ 1
0 01.00101

⊕ 00100101
1's complement + 11011010
11111111

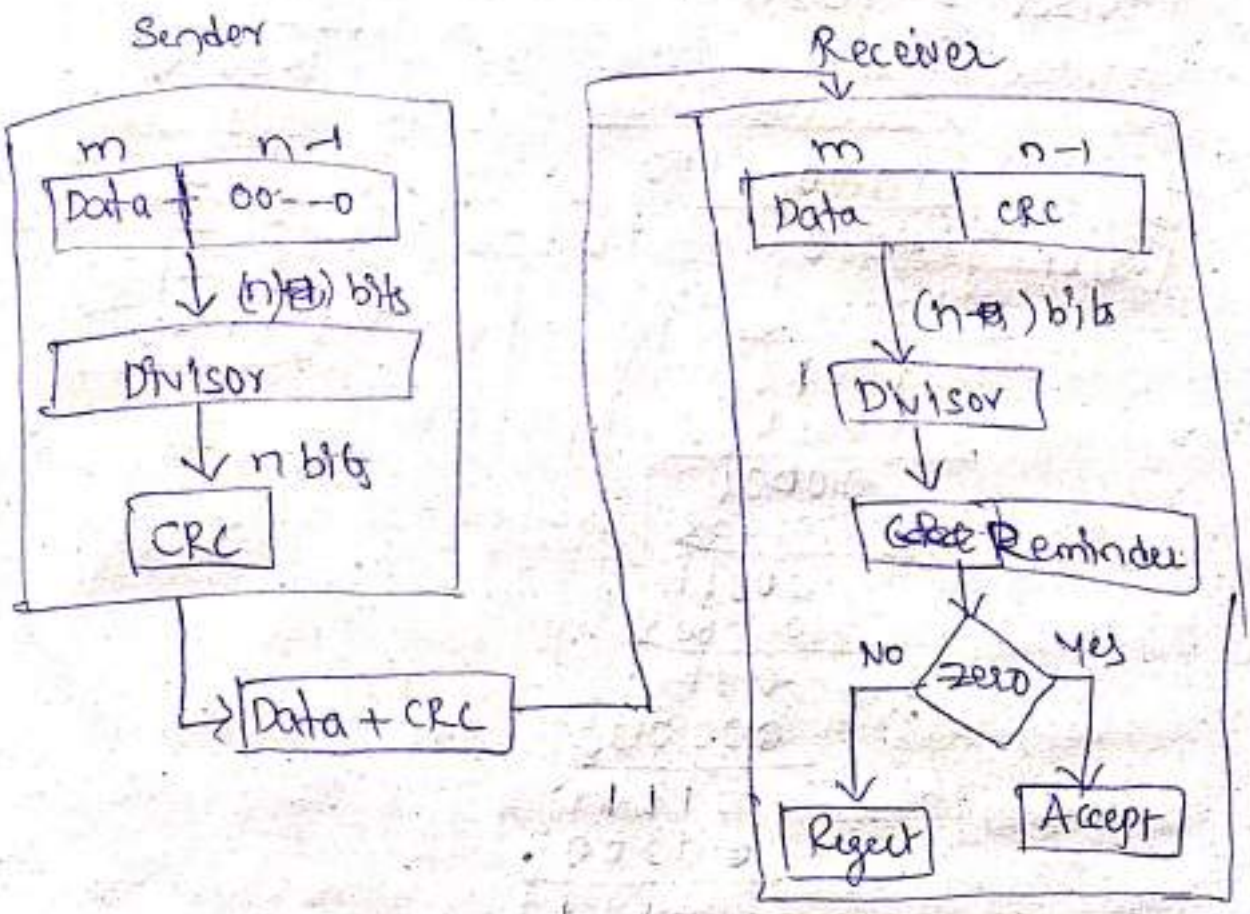
1's complement = 00000000

⇒ Accept Data.

→ At receivers end all segments & checksum data are added with 1's complement arithmetic to get the sum.

→ The sum is again complemented. If the result is '0' (zero) the data is accepted by the receiver

Cyclic Redundancy Check:-



→ Size of the message is m; (n-1) bits of zero's appended

→ Size of the divisor is (n) bits and 'n' bits are the reminder. called as CRC

→ (n-1) bits are appended to the low-order end of data and given to the code of size m+(n-1) bits.

~~Receiver's code is correct~~

→ m+(n-1) bits of data is divided with divisor of size 'n' bits with binary division method. for addition it uses 'XOR' operation.

→ Thus the remainder is the CRC code.

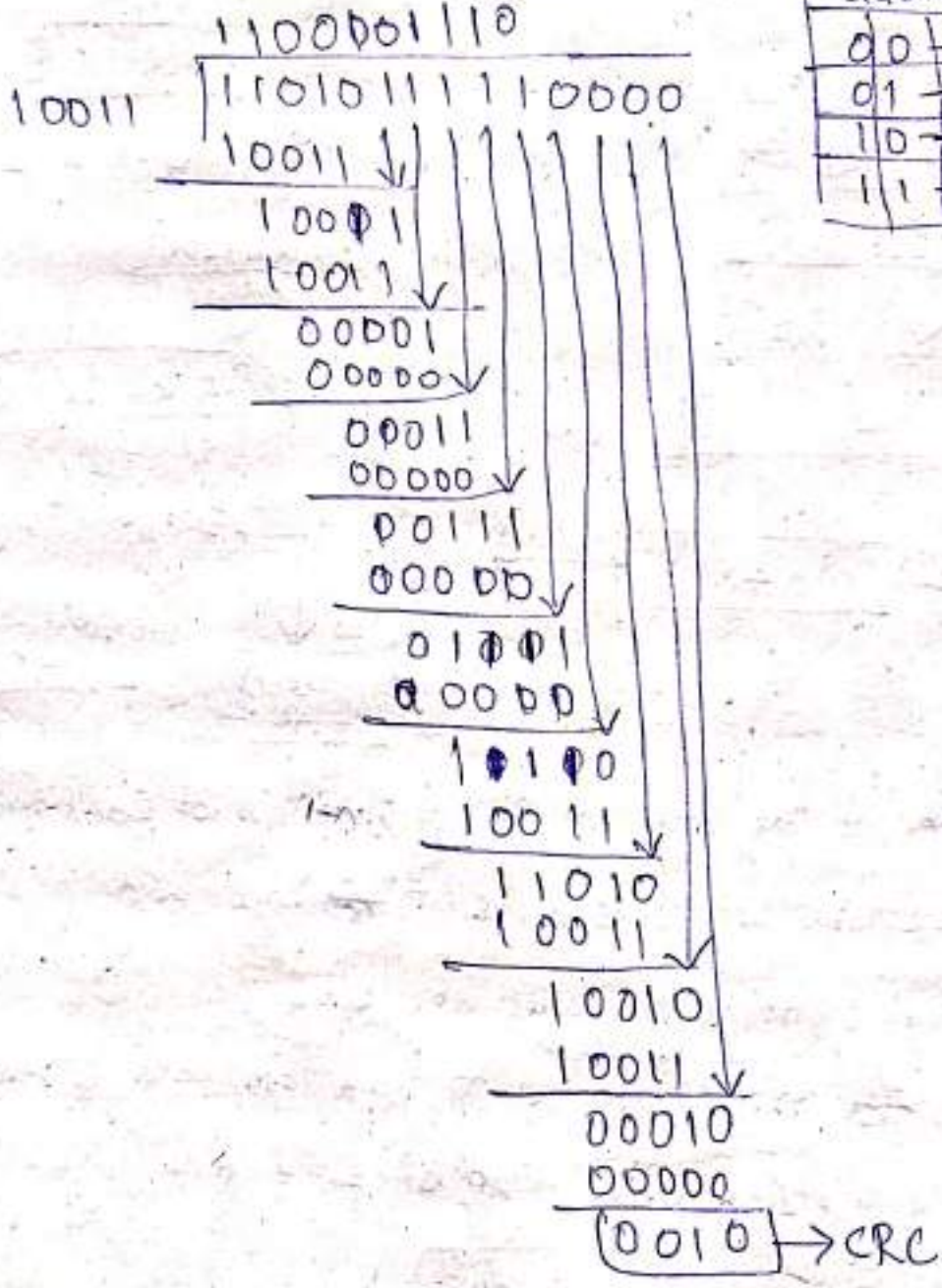
→ This CRC code is added to the m+(n-1) bits data.

Frame = 11010 11111

Divisor = 10011

XOR for addition
0 0 = 0
0 1 = 1
1 0 = 1
1 1 = 0

Sender Side

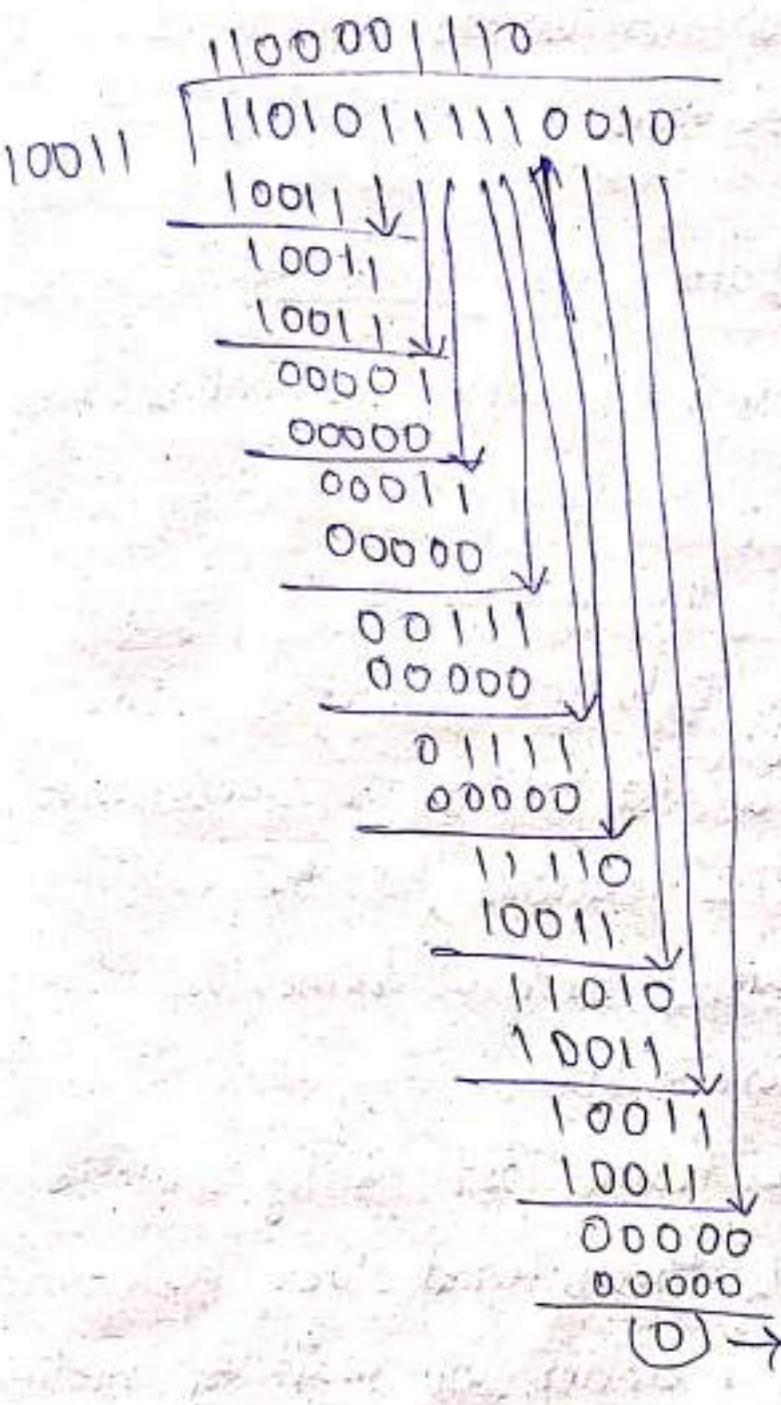


Receiver code = (frame + bits) + CRC

= 11010111110000 + 0010

Receiver code = 11010111110010

Receiver Side



→ The result is [m+CRC], given to the receiver.

→ At receiver end, again [m+CRC] data is divided with divisor with binary division method.

→ If the remainder is all zero's, then the data is accepted by the receiver.

→ If it contain any '1' at the remainder, then the data contain errors and it is rejected by the receiver.

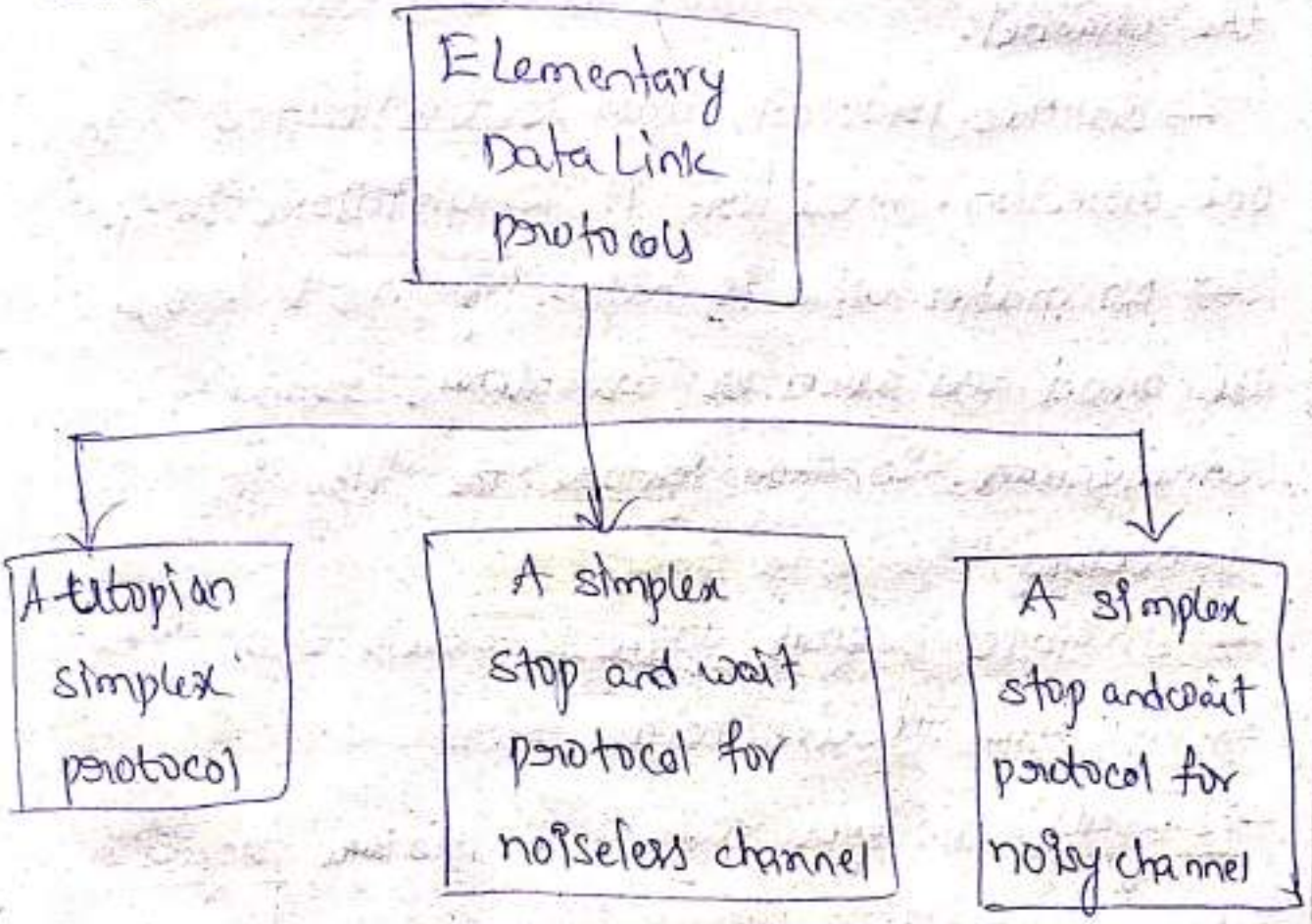
Elementary Data Link Protocols:

DLL communicates with the network layer and physical layer. Generally, the dll receives packets and sends the frames to the physical layer. Elementary Data link protocols ensure reliability over a n/w during the transmission of frames. The EDL protocals help the data link layer solve problems such as frame loss (or) damage and flow control.

If, To recover frame lost during transmission the sender starts the internal clock each time a frame is sent. The sender will wait for sometime and if no reply is received within the predetermined time, the clock times out, and the DLL will receive an interrupt signal.

These are some methods (or) procedures executed by the protocol that turn the timer on and off, respectively. The clock is reset only after an interval is reached.

since the EDL protocol is used at the DLL, the DLL also controls access to the n/w layer. The DLL decides when to enable the n/w layer to sent packets to prevent it from swamping packets with them.



3 types of EDL protocols :-

① A utopian simplex protocol

② A simplex stop-and-wait protocol for noiseless channel

③ A simplex stop and wait protocol for noisy channel.

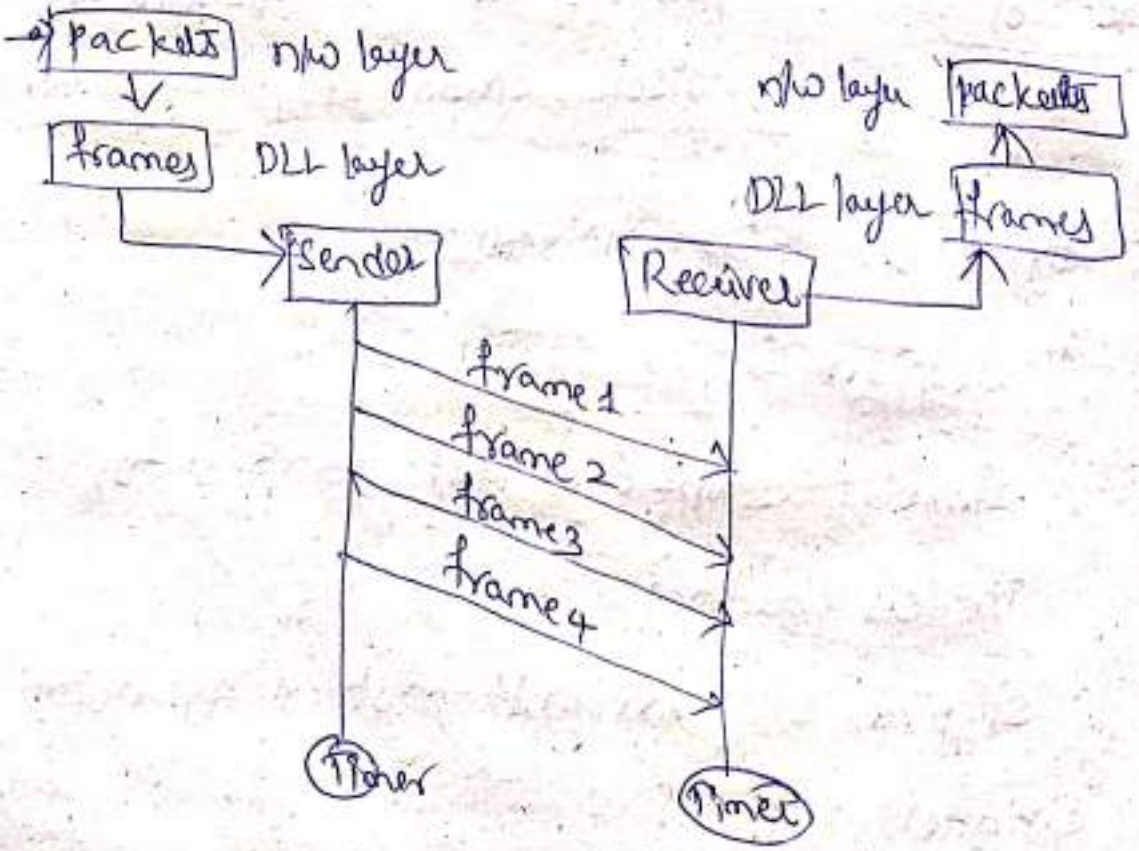
Utopian Simplex Protocol:-

An utopian simplex protocol is a simple protocol because it does not worry about whether something is going right or wrong on the channel.

→ In this protocol, data is transmitted in only one direction. Therefore it is unidirectional.

→ No matter what is happening in the n/w, the sender and receiver are always ready to communicate. So they ignore the delay in processing.

→ Infinite buffer space is available on the sender and receiver sides



Because It has no flow control and error control restrictions, Here, no frame is lost during transmission, and hence no field of the frame is required to control the flow of data and detect the error.

Simplex Stop-and-wait protocol for Noiseless Channel:-

In noiseless channel, there are no errors while transmission of data so frame is never damaged or corrupted. Here the channel is error-free but does not control the flow of data.

Using simplex-stop-and-wait protocol, we can prevent the sender from flooding the receiver with frames faster than the receiver can process them.

Common solution for addressing flooding issues on the receiver side, providing feedback to the sender to reduce the flow rate at the receiver. Receiver sends an acknowledgement feedback to the sender, after arrival of every frame.

The process is as follows:

Step 1: The sender sends a frame to the receiver.

Step 2: The receiver sends an acknowledgement frame back to the sender, telling the sender that the last received frame has been processed and passed to the host.

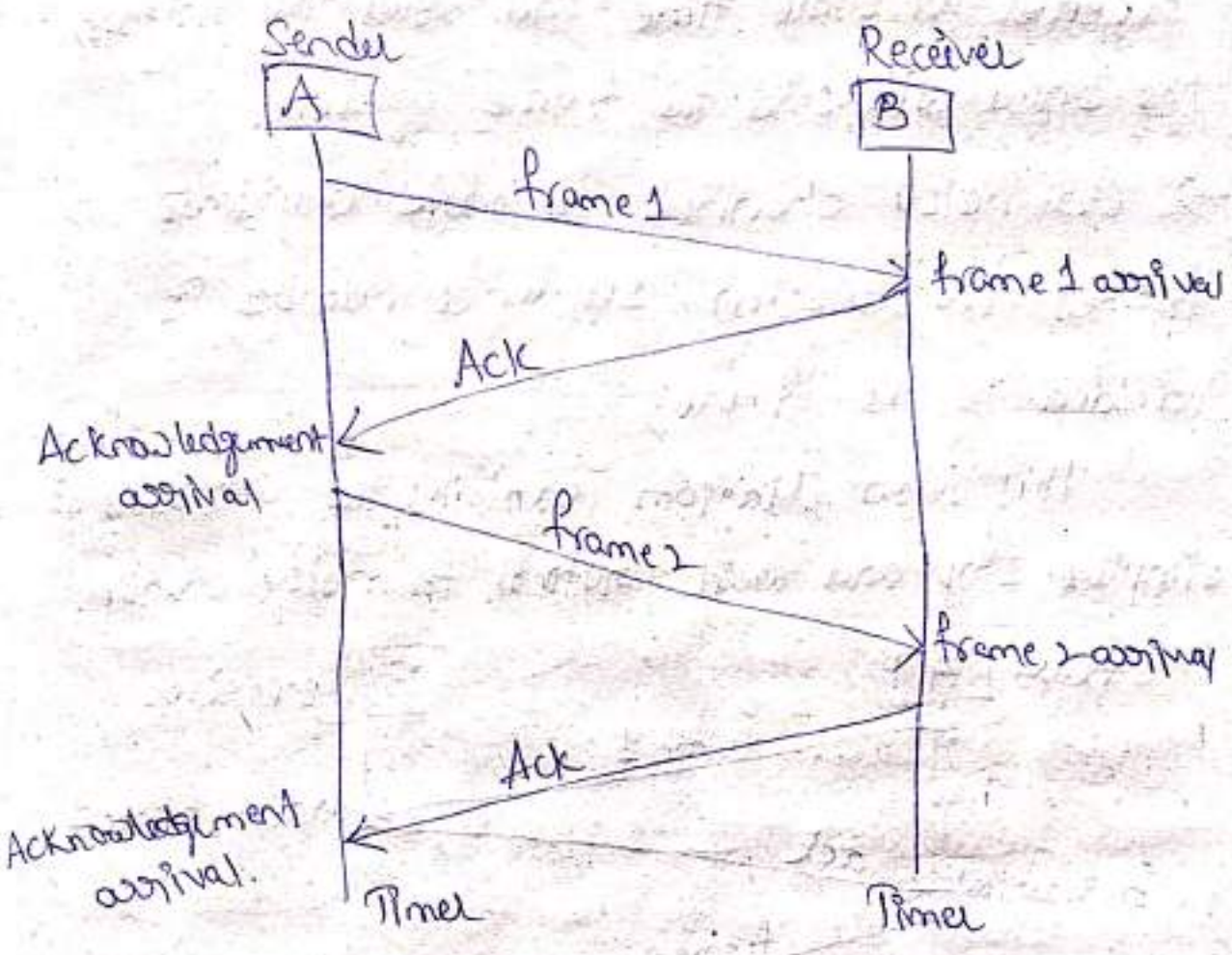
Step 3: permission to send the next frame is granted.

Step 4: After receiving acknowledgement from receiver and permission granted to send next frame, the sender sends next frame to the receiver.

This protocol is called simplex stop-and-wait protocol, because the sender a frame to receiver and waits for feedback from the receiver. when the acknowledgement arrives the sender sends the next frame to the receiver. and the process goes on.

Frames and feedbacks are transmitted or receives from the sender and receiver, so the simplex stop-and-wait protocol is bidirectional.

This diagram explains the working of the simplex stop-and-wait protocol.



Simplex Stop-and-wait protocol for a Noisy channel :-

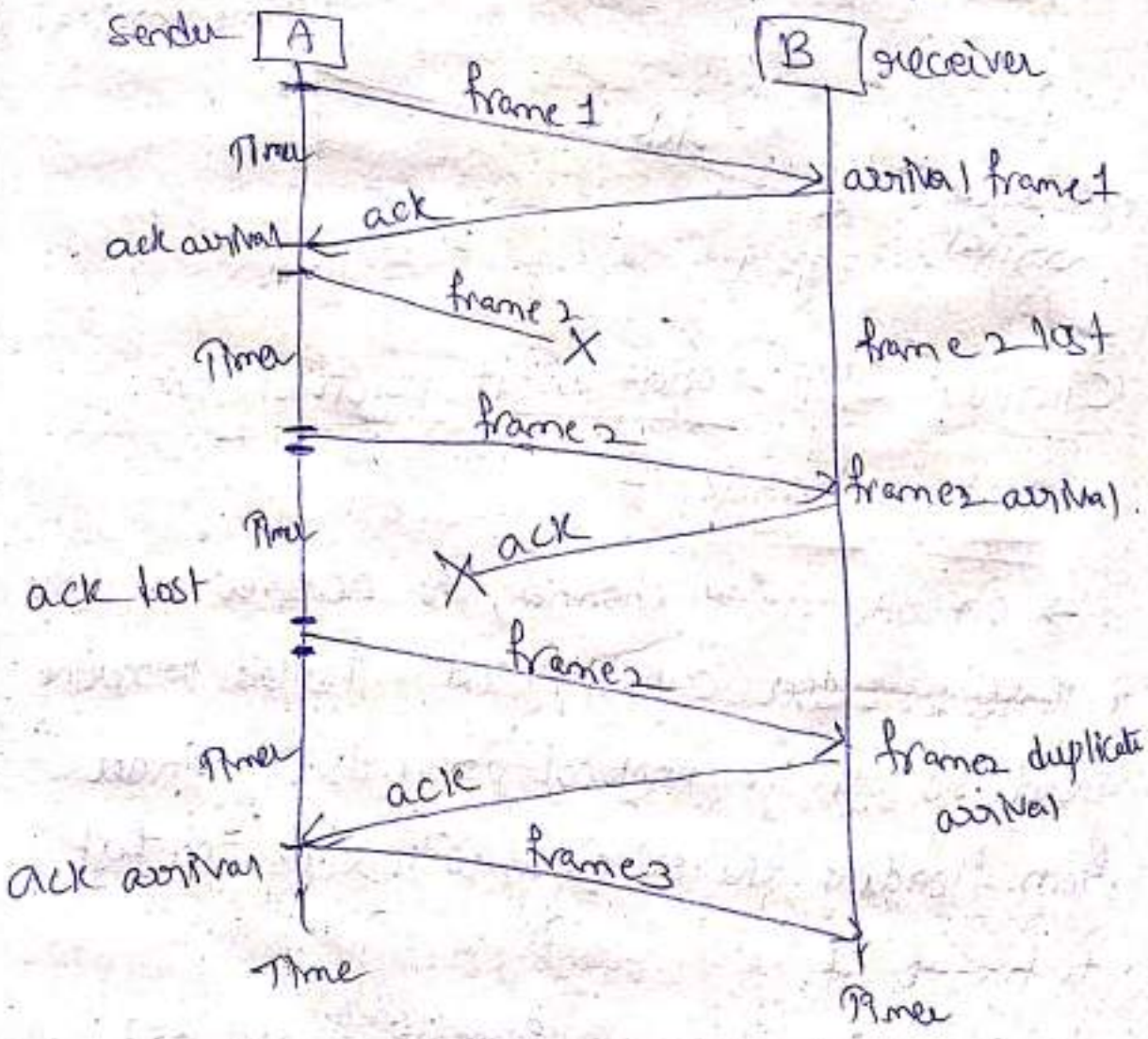
→ On the noisy channel, the receiver has only a limited buffer capacity and a limited processing speed, so that the protocol prevents the sender from flooding the receiver with data too fast to handle it.

→ On noisy channel some of frames and acknowledgement frames are losted in the transmission. To overcome this timers are

added to each frame. The sender will wait for an acknowledgement from the receiver for some time, and after the timeout, the sender will send the frame again.

→ On noisy channel, to avoid duplicate frames transmission, sequence number is added to the frame.

This below diagram explains the working of simplex stop and wait protocol for noisy channel.



→ This protocol also called as ARQ (Automatic Repeat request).

As you can see the diagram:

→ frame 1 successfully reaches to the receiver, the receiver will send an acknowledgement to the sender.

→ Then, the sender send frame 2, but as shown in figure, frame-2 is lost during transmission. Therefore the sender will retransmit the frame 2 after the timeout.

→ Furthermore, the receiver is sending an acknowledgement to the sender after receiving the frame 2. But the acknowledgement is lost during transmission. The sender is waiting for acknowledgement, but the timer is timeout. and acknowledgement not received from the receiver. So the sender again retransmits the frame 2. In second time (duplicate)

→ The receiver identifies this duplicate frame 2 with the help of sequence number. If it not it is accepted and passed to the n/w layer. If not it reject the duplicate frame.

Sliding Window Protocols :-

In previous simplex stop-and-wait protocols (or) Elementary data link protocols, the sender sends only one frame bit at a time and waits for an acknowledgement frame for each frame. For this, we have two circuits each with a forward channel for data and a reverse channel for acknowledgement. So that the bandwidth of reverse channel is almost entirely wasted. A better idea is to use the same circuit for data in both directions. After all, in simplex stop & wait protocol for noisy / noiseless channel it was already being used to transmit frames both ways, and the reverse channel has the same capacity as the forward channel. In this model the data frames from sender to receiver intermixed with the acknowledgement frames from ~~sender~~ receiver to sender. 'Kind' field in the frame header added, which tells ~~that~~ whether the frame is data (or) acknowledgement.

Although interleaving data & ack frames on the same circuit is an improvement over

having two seperate physical circuits. Yet another improvement is possible, when the data frames arrives, instead of immediately sending a seperate control frame, the receiver restrains itself and waits untill the network layer passes it to the next packet, ~~the receives next~~ The acknowledgement is attached to the outgoing data frame (using the ack field in the frame header). So that the acknowledgement get a free ride on the next outgoing data frame. The technique of temporarily delaying outgoing acknowledgements so that they can be hooked onto the next outgoing dataframe is known as Piggy Backing.

The principal advantage of using piggy backing over having distinct acknowledgement frames is a better use of the available channel bandwidth

However the drawback with piggybacking is, how long should the DLL wait for a packet onto which to piggyback acknowledgement? If the datalinglayer waits longer than the sender's timeout period, the frame will be retransmitted, defeating the purpose of having acknowledgements

So that the simplex stop and wait protocol for noisy channel can fail under some condition involving early timeout. To overcome this a protocol, that remained synchronized in the face of any combination of garbled frames, lost frames and premature timeout. The class of this protocols called as Sliding window protocols.

→ In sliding window protocol, each outbound frame contains a sequence number, ranging from 0 to $2^n - 1$; sequence number bits = n.

→ The essence of all sliding window protocols is that at any instant of time, the sender maintains a set of sequence numbers corresponding to frames it is permitted to send. These frames are said to fall within the sending window.

→ Similarly the receiver also maintains a receiver window corresponding to the set of frames it is permitted to accept.

Sliding window protocols classified into 3 types:-

① A one-bit sliding window protocol (1,1)

② A protocol using Go-Back-N (>1, 1)

③ A protocol using selective Repeat (>1, >1).

A one-bit sliding window protocol: (1,1) :-

→ In one-bit sliding window protocol, the size of the window is 1. So the sender transmits a frame, waits for its acknowledgement, then transmits the next frame. Thus it uses the concept of stop and wait for the protocol.

→ The acknowledgement is attached along with the next data frame to be sent by piggybacking.

→ The data frames to be transmitted additionally have an acknowledgement field, 'ack' field that is of a few bits length.

→ The ack field contains the sequence number of the last frame received without error.

→ If the sequence number matches with the sequence number of the frame to be sent, then it is inferred that there is no error and the frame is transmitted.

→ Otherwise it is inferred that there is an error in the frame and the previous frame is retransmitted.

→ It is a bidirectional protocol.

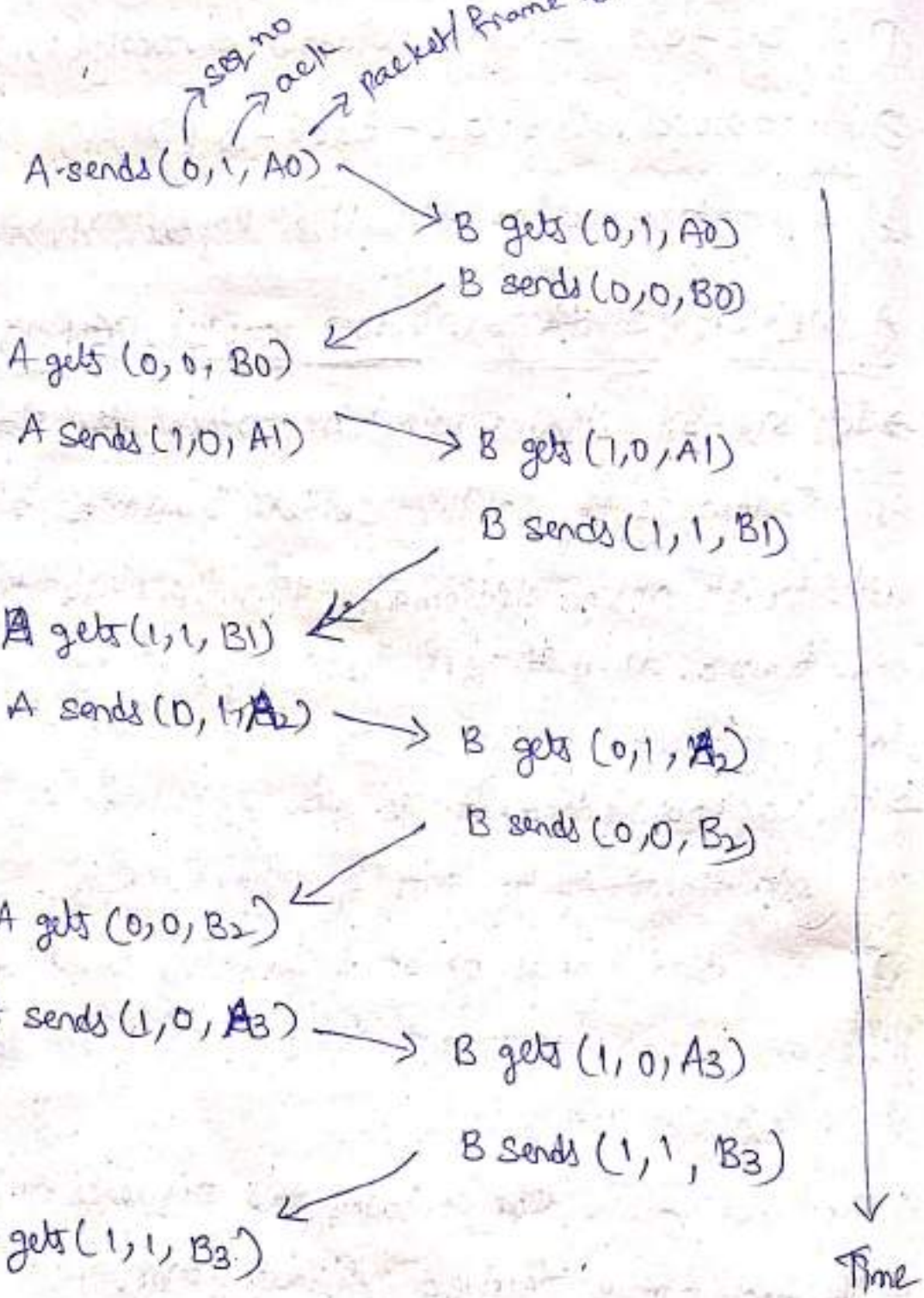


fig (a)

notation is (seq no, ack, packet/frame number)

In fig (a), the normal operation of the protocol shown. It B waits for A's first frame before sending one of its own, and every frame is accepted.

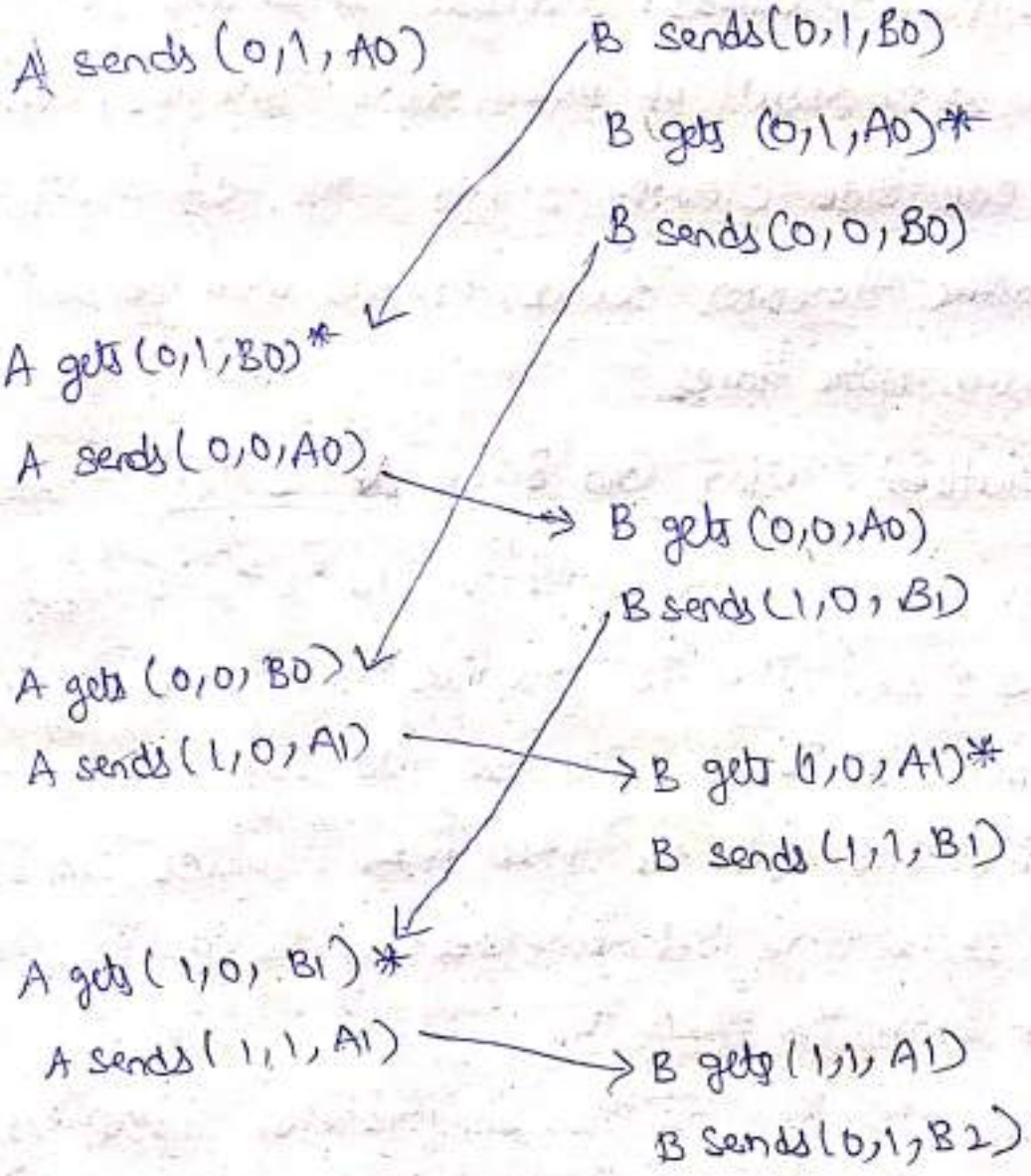


Fig (b)

In the above situation :- Fig (b).

→ A is trying to send its frame 0 to B and that B is trying to send its frame 0 to A.

→ Suppose A sends a frame to B, but A's timeout interval is a little too short. So A may time out repeatedly, sending a series of identical frames containing seq 0, ack 1 ; & seq 1, ack 0.

→ In this situation half of the frames contain duplicates, even though there are no

transmission errors. Similar situations can occur as a result of premature timeouts, even when one side clearly starts first. If multiple premature timeouts occur, frames may be sent three or more times.

A protocol using Go Back N: (N>1) :-

In Go-Back-N protocol, N is the sender's window size. That is greater than 1 and receiver's window size is 1. Suppose we say that Go-Back3 which means that the three ~~from~~ frames can be sent at a time before expecting the acknowledgement from the receiver.

It uses the priciple of protocol pipelining, in which the multiple frames can be sent before receiving the acknowledgement of the first frame.

If we have 5 frames F_1, F_2, F_3, F_4 & F_5

Sliding window size Go-Back-3

Then first 3 frames F_1, F_2 & F_3 are sent before expecting acknowledgement of first frame F_1.

In Go-Back-N, the frames are numbered sequentially as Go-Back-N A sends the multiple frames at a time that requires the numbering approach to distinguish the frame from another frame, and these numbers are known as sequential

numbers.

The no. of frames that can be sent at a time totally depends on the size of the senders window. So we can say that 'N' is the numbr of frames that can be sent at a time before receiving the acknowledgment from the receiver. If acknowledgment of a frame is not received within time period, then all frames available in the current window will be retransmitted.

— N is the window size of sender

— If the size of the senders window is 4, then the sequence number will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 4 and so on.

— The no. of bits in the sequence number is 2, to generate the binary sequence 00, 01, 10, 11.

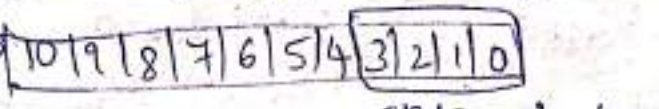
Working of Go-Back-N:-

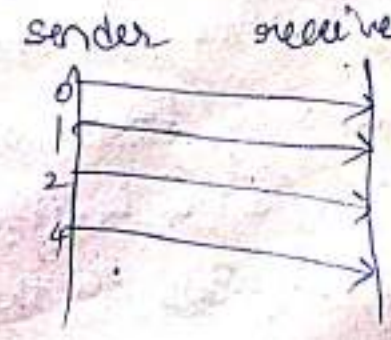
⇒ Ex:- no. of frames = 11, represented as:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

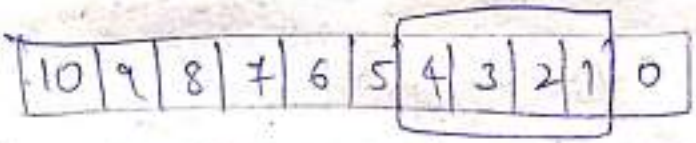
Window size = 4.

→ Firstly, the sender will send first four frames 0, 1, 2, 3 to the receiver as the window size is 4. Now the sender is expected to receive the acknowledgment of the 0th frame (frame zero)



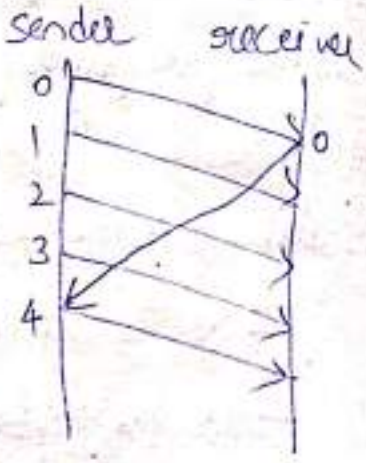


→ receiver send acknowledgement of frame 0, & Sender receives ack, then the sender will send next frame i.e., 4. & the window now contain 4 frames 1,2,3,4.

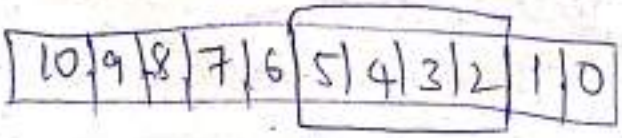


Sliding window

Window size 4

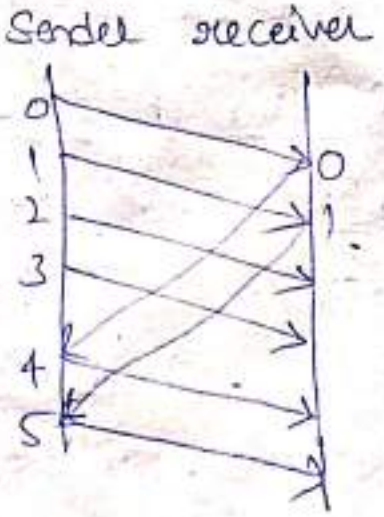


→ receiver send ack of frame 1, & the sender will send the next frame i.e, 5, & the sliding window have 4 frames (2,3,4,5)



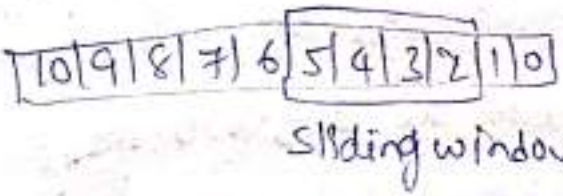
Sliding window

Window size 4

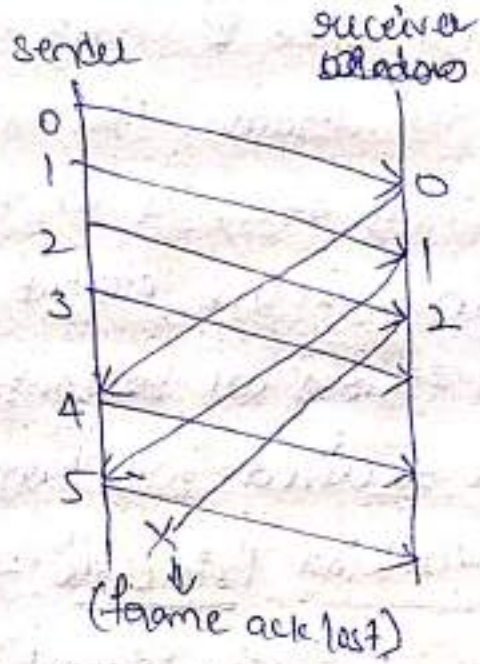


→ Now let us assume that the receiver is not acknowledging the frame no2, either the frame is lost (or) the acknowledgment is lost. Instead of sending the frame 6, the sender Go-Back to frame which is the first of the current window, retransmits all the frames in the current window.

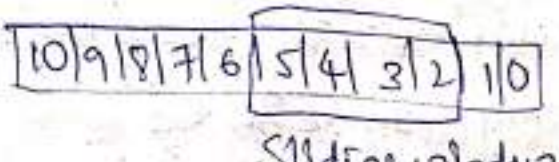
i.e., 2, 3, 4, 5



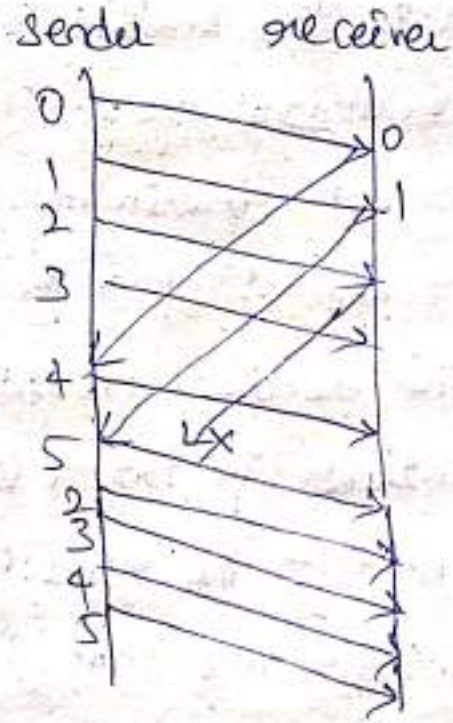
Sliding window



Go-Back - to frame 2 & retransmits all frames in current window.



Sliding window



→ A protocol using Selective Repeat (>1, >1) :-

→ In this protocol, the sliding window size of sender and receiver are same and greater than 1.

→ The sending window that stores the frames to be sent and a receiving window that

stores the frames receive by the receiver.

→ The size is half the maximum sequence number of the frame. For example, if the sequence number is from 0-15, the window size will be 8.

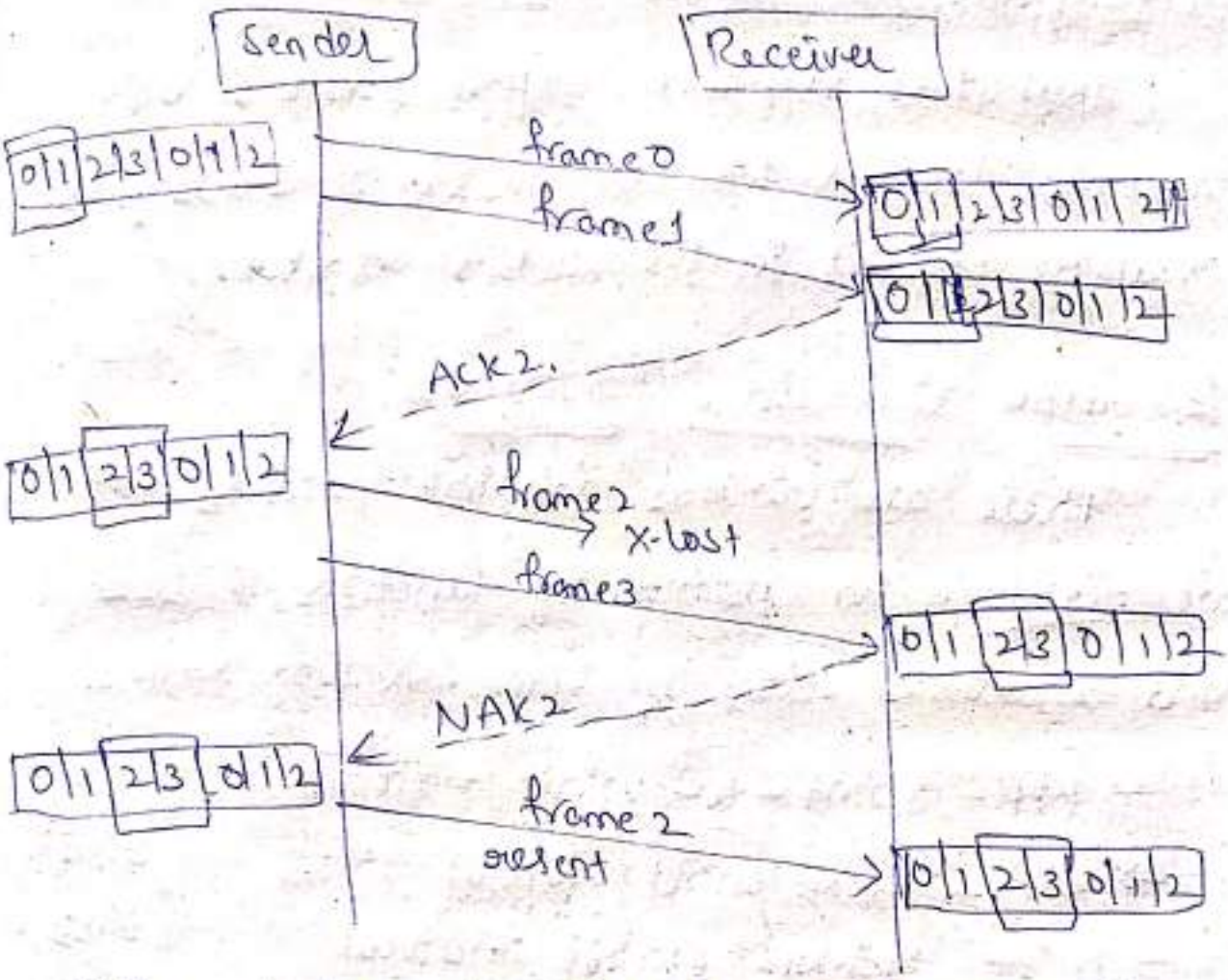
→ In selective repeat protocol only erroneous (or) lost frames are retransmitted, while the good frames are received and buffered.

→ Working Principle:-

Selective repeat protocol provides for sending multiple frames depending upon the availability of frames in the sending window, even if it does not receive aknowledgement for any frame in the interim. The maximum number of frames that can be sent depends upon the size of the sending window

The receiver records the sequence number of the earliest incorrect (or) un-received frame. It then fills the receiving window with the subsequent frames that it has received. It sends the sequence number of the missing frame along with every acknowledgement frame.

The sender continues to send frames that are in its sending window. Once, it has sent all the frames in the window, it retransmits the frame whose sequence number is given by the



acknowledgements. It then continues sending the other frames.

For Example In the above diagram, both sender and receiver window size is 2 Sender first send two frames 0 & 1. and receiver successfully received and acknowledged the first two data frames. When the acknowledgement is received at sender side, it transmits the next 2 frames 2 & 3 to the receiver.

If frame 2 is lost in transmission, the receiver sends a negative acknowledgement

for frame 2 to sender. Now the sender retransmits the frame 2.

Thus this selective repeat protocol only transmits losted frames instead of sending all available frames in the sliding window.

Example Datalink Protocols:-

There are various datalink protocols that are required for performing functions of DLL and as well as setting up LAN and Large WAN's.

- → PPP – point-to-point protocol.
- → LCP – Link control protocol
- → NCP – Network control protocol
- → HDLC – High level data link control

} SONET Links

- → ATM – Asynchronous Transfer mode
- → AAL5 – ATM Adoption Layer 5
- → PPPoA – PPP over ATM

} ADSL links

SONET Links:- Synchronous optical Network (or) SONET is used to transmit a large volume of data over relatively long distances, using optical fiber. It allows multiple digital data streams to be transmitted simultaneously over the same optical fiber using LED's and laser beams.

Point-to-point protocol:-

→ It is a protocol of same functionality of SLIP (serial line Interface protocol). This SLIP is older protocol, is used to add a framing byte to the end of the IP packet. It functions essentially as a DLL control facility that is necessary for the dial up IP packet transfer between an Internet Service Provider (ISP) and a home user. It has no mechanism for error detection and correction

Where ppp is a protocol is used to transport not only IP packets but also other kind of packets. This protocol has an error-detection function.

1 byte	1 byte	1 byte	1 or 2 bytes	variable	2/4 bytes	1 byte
Flag	Address	control	protocol	Data	FCS	Flag

frame format of PPP

The frame format of PPP consists:-

→ flag field – frame always begings and ends with on the frame. It always has a value of a 1 byte i.e, 01111110 binary value

→ Address – It basically broadcast address. In this all 1's simply indicates that all of the stations are ready to accept frame. It has a value of 1 byte i.e, 11111111 binary value

Control field :- This field set to 1 byte i.e, 00000011 binary value. This 1 byte is used for a connection-less datalink.

Protocol field :- It identifies n/w protocol of the datagram. It identifies the kind of packet in the data field i.e, what exactly is being carried in data field. size 1 or 2 bytes

Data field :- N/W layer datagram is particularly encapsulated in this field for regular PPP data frames. Length of this ~~frame~~ field is not constant rather it varies.

FCS field :- It usually contains checksum simply for identification of errors. It can be 16 bits or 32 bits in size.

The PPP additionally deliver 2 protocols namely NCP and LCP protocols.

NCP — Network control protocol (NCP) :-

NCP was additional protocol of PPP and used by ARPANET. In essence it enables users to access computers and a few other devices at distant locations and transfer files between two (or) more computers.

LCP - Link control protocol:-

It was created and developed by IEEE 802.2. The purpose of LCP are: establishing, configuring, testing, maintaining, and ending (or) terminating links for the transmission of dataframes. It is essentially a PPP protocol.

High level datalink control (HDLC):-

HDLC protocol is that many wide area protocols are now thought to fall under. This protocol is based on SDLC (Synchronous data link protocol). HDLC supports services of SDLC like it is used to establish a point-to-point (or) point-to-multipoint connections between all of the remote devices and the mainframe computers at the central machine. Also it ensures data packets are moving correctly & in right order from one n/w point to next. Additionally it offers both reliable and unreliable service with best effort.

ADSL Links:-

ADSL stands for, Asymmetric Digital Subscriber line.

It is a communication technology designed to offer faster speeds and greater bandwidth over traditional dial-up connections. ADSL allows faster transmission and more data to be sent over existing copper telephone lines that are used for landlines.

ADSL supports both data and voice transmission at the same time using a single copper telephone line. ADSL requires a modem to be installed that allows data traffic to the line itself.

ADSL have different types of protocol stack

PPP is point-to-point protocol.

ATM :- Asynchronous Transfer mode (ATM) would solve the world's telecommunications problems by merging voice, data, cable television, telegraph and everything else into an integrated system that could do everything for everyone

ATM is a connection-oriented technology.

Each cell carries a virtual circuit identifier in its header and devices use this identifier to forward cells along the paths of established connections.

AAL5 – ATM Adaptation Layer 5, is used to map send data over sequence of cells. AAL5 is used for Packet data.

PPPoA (PPP over ATM): This is a standard not a protocol, but more specification of how to work with both PPP and AAL5 frames.

Collision Free Protocols:-

Collision free protocols are the resolves the contention for the channel without any collisions at all, not even during the contension period.

We assume that there are exactly N stations, each programmed with a unique address from 0 to N-1. We have 3 collision free protocols:-

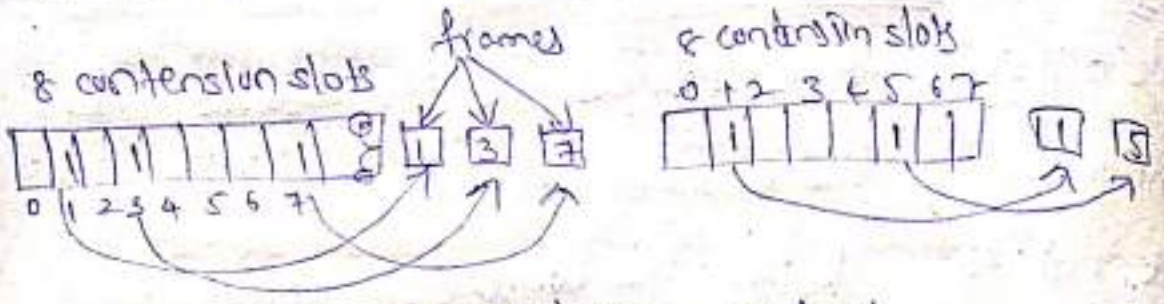
→ A Bit-Map protocol
→ Token passing
→ Binary countdown

A Bit Map protocol:-

→ Each contension period consists of exactly N slots. If station '0' (zero) has a frame to send, it transmits a 1 bit during the slot 0. no other stations is allowed to transmit during this slot. Regardless of what station 0 does, station 1 gets the opportunity to transmit a 1 bit during slot 1, but only if it has a frame queued.

→ In general, station j may announce that it has a frame to send by inserting a 1 bit into slot j. After all N slots have passed by, each station has complete knowledge of which stations wish to transmit. At that point, they begin transmitting

frames in numerical order.



The Basic bit-map protocol

Since everyone agrees on who goes next, there will never be any collisions, After the last ready station has transmitted its frame, an event all stations can easily monitor, another N-bit contention period is begun. If a station becomes ready just after its bit slot has passed by, it is out of luck and must remain silent until every station has had a chance and the bit map has come around again.

Protocols like this in which the desire to transmit is broadcast before the actual transmission are called reservation protocols, because they reserve channel ownership in advance. and prevent collisions.

Binary countdown:-

The problem with basic bit-map protocol is that overhead is 1 bit per station. To overcome this, A station wanting to use the channel now broadcasts its address as a binary bit string, starting with the high-order bit. All addresses are assured to be the same length. The bits in each address

position from different stations are BOOLEAN ORed together. We call this protocol binary count down.

To avoid conflicts, an arbitration rule must be applied, as soon as a station sees that a high-ordered bit position that is 0 (zero) in its address has been overwritten with a 1, it gives up.

For example if stations 0010, 0100, 1001 and 1010 are all trying to get the channel, in the first bit time the stations transmit 0, 0, 1 & 1 respectively. These are ORed together to form a 1. Stations 0010, 0100 see the 1 and know that a higher-numbered station is competing for the channel, so they give up for the current round. Stations 1001 and 1010 continue.

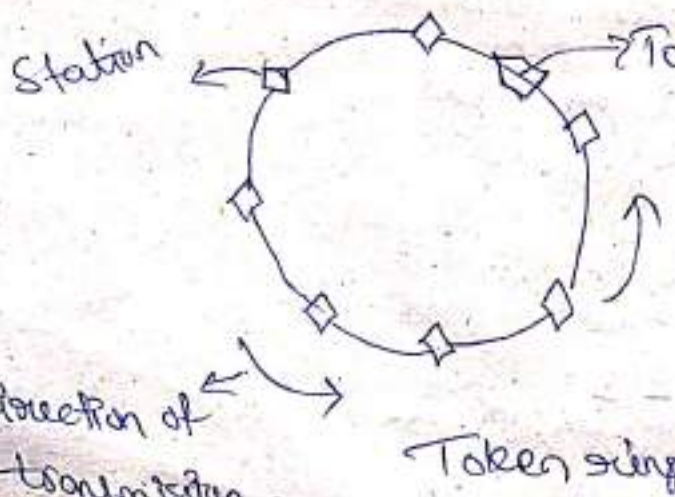
The next bit is 0 (zero), and both stations continue. The next bit is 1, so station 1001 gives up. The winner is station 1010, because it has the highest address.

```
          ↓  ↓  ↓  ↓
0010      0
0100      0
1001      1  0  0
1011      1  0  1  1
OR'd      1  0  1  1
          ↙     ↳ 1001 dropped.
0010, 0100
dropped
```

Token passing :-

Pass a small message called Token from one station to the next in the same predefined order. The token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it simply passes the token.

In a token ring protocol, the topology of the n/w is used to define the order in which stations send. The stations are connected one-to the next in a single ring. Passing the token to the next station then simply consists of receiving the token in from one direction. Frames are also transmitted in the direction of the token. This way they will circulate around the ring and reach whichever station is the destination.



We do not need a physical ring to implement. token passing. The channel ~~ends~~ connecting the tokens might instead be a single long bus. Each station then uses the bus to send the token to the next station in the predefined sequence. This protocol is called token bus.

Medium Access sub layer:-

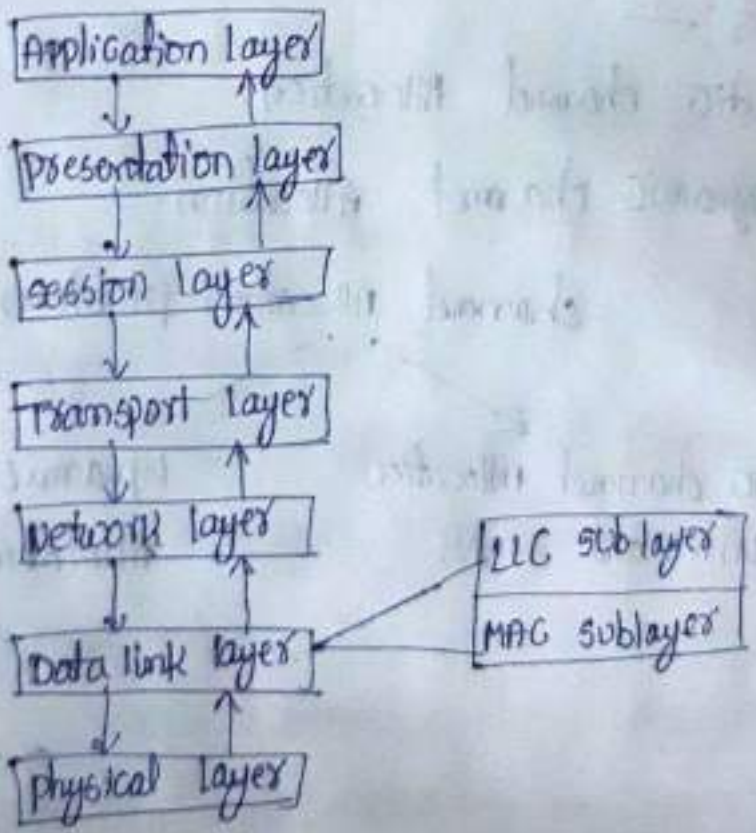
→ the medium access control (MAC) is a sublayer of the data link layer of the open system interconnections (OSI) reference model for data transmission. It is responsible for flow control and multiplexing for transmission medium.

MAC sub layer in the OSI model:-

→ the data link layer is the 2nd lowest layer. It is divided into two sublayers.

→ the logical link control (LLC) sub layer

→ the medium access control (MAC) sublayer.



: position of the MAC layer.

Functions of MAC Layer:—

→ It provides an abstraction of the physical layer to the LLC and upper layers of the OSI N/w.
→ It is responsible for encapsulating frames so that they are suitable for transmission via the physical medium.
→ It resolves the addressing of source station as well as the destination station, or groups of destination stations.
→ It performs multiple access resolutions when more than one data frame is to be transmitted. It determines the channel access methods for transmission.
→ It also performs collision resolution & initiating retransmission in case of collision.

MAC Addresses:—

MAC address is a unique identifier allotted to a n/w interface controller (NIC) of a device. It is used as a n/w address for data transmission within a n/w segment like Ethernet, Wi-Fi, & Bluetooth.

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

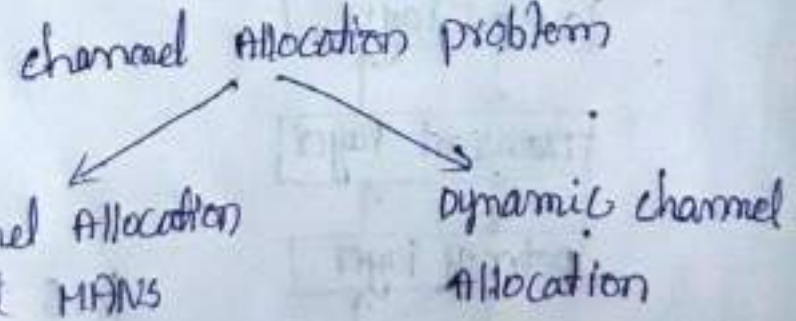
the channel allocation problem :-

Channel allocation is a process in which a single channel is divided and alloted to multiple users in order to carry user specific tasks.

→ If there are N no. of users and channel is divided into N equal-sized sub channels, Each user is assigned one portion. If the no. of users are small and don't vary at times, then frequency Division multiplexing can be used as it is a simple & efficient channel bandwidth allocating technique.

→ channel allocation problem can be solved by two schemes :—

1) static channel Allocation
2) dynamic channel Allocation.



1) Static channel Allocation in LANs and MANs:—

It is the classical or traditional approach of allocating a single channel among multiple competing users Frequency Division Multiplexing (FDM).

→ If there are 'N' users, the bandwidth is divided into N equal sized portions, each user being assigned one portion. since each user has a private frequency band, there is no interface b/n users.

2) Dynamic Channel Allocation:—

possible assumptions include:—

1) Station Model:—

Assumes that each of N stations independently produce frames

2) single channel Assumption:—

In this allocation all stations are equivalent and can send and receive on that channel.

3) collision Assumption:—

If two frames overlap in time-wise, then that's collision, Any collision is an error, and both frames must re transmitted. collisions are only possible errors.

Multiple Access protocols:—

-the Data link layer is responsible for transmission of data b/m two nodes.

Its main functions are:—

→ Data link Control

→ Multiple Access Control.

Data link Control:—

-the data link Control is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control & flow control.

Multiple Access Control:—

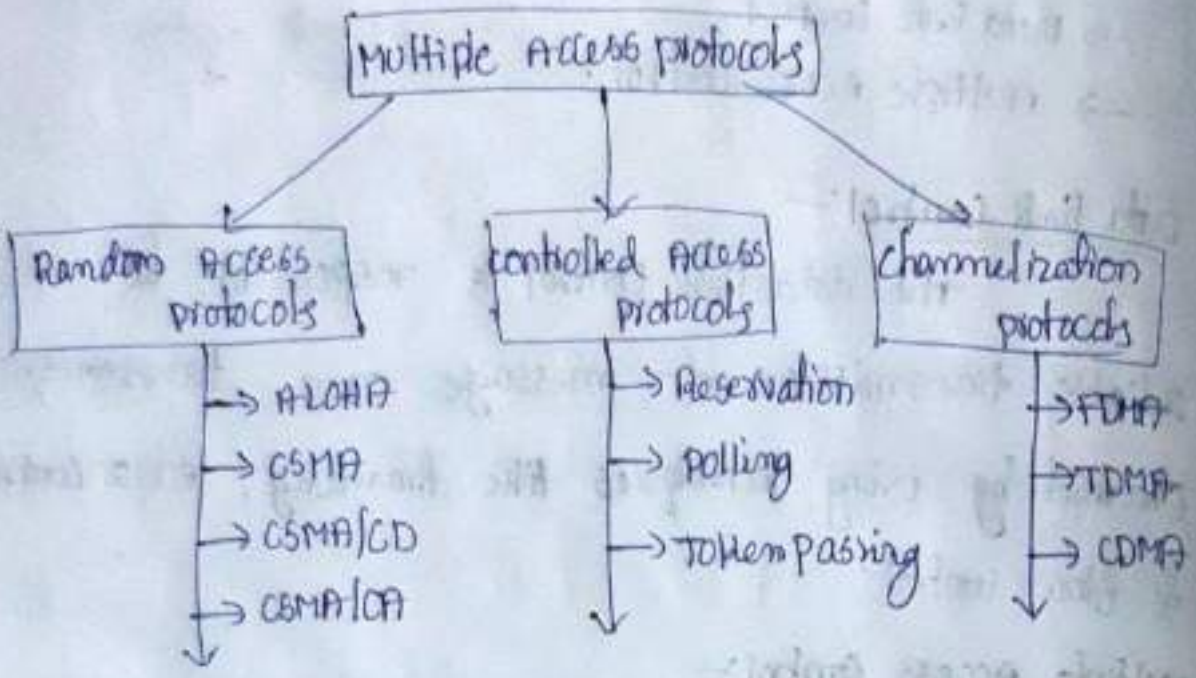
If there is a dedicated link b/m the sender and the receiver then data link control layer is sufficient, however if there is no dedicated link present then multiple stations can access the channel.

→ Hence multiple access protocols are required to decrease collision and avoid crosstalk.

→ for example, in a classroom full of students, when a teacher asks a question and all the students (as stations) start answering simultaneously (send data at same time) then a lot of chaos is is created (data

overlap or data lost) then it is the job of the teacher (multiple access protocols) to manage the students and make them answer one at a time.

→ Multiple access protocols can be divided as :-



1) Random Access protocol :-

In this, all stations have same superiority that is no station has more priority than another station. Any station can send data depending on medium's state. It has two features.

1) there is no fixed time for sending data
2) " " " " sequence of stations sending data.

the different Random access protocols are:—

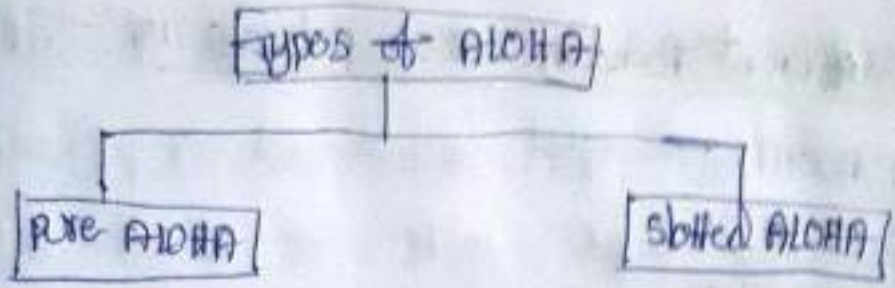
1) ALOHA
2) CSMA
3) CSMA/CD
4) CSMA/CA

1) ALOHA:—

It is designed for wireless LAN. but can also be used in a shared medium to transmit data. using this method, any station can transmit data across a n/w simultaneously when a data frameset is available for transmission.

ALOHA Roles:—

1) Any station can transmit data to a channel at any time.
2) It does not require any carrier seming.
3) collision and data frames may be lost during the transmission of data through multiple stations.
4) Acknowledgment of the frames exists in Aloha.
5) It requires retransmission of data after some random amount of time.



Pure ALOHA :—

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 data frame can be lost. when any station transmits the data frame to a channel, the pure aloha waits for the receiver's ack. 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).

→ An 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.

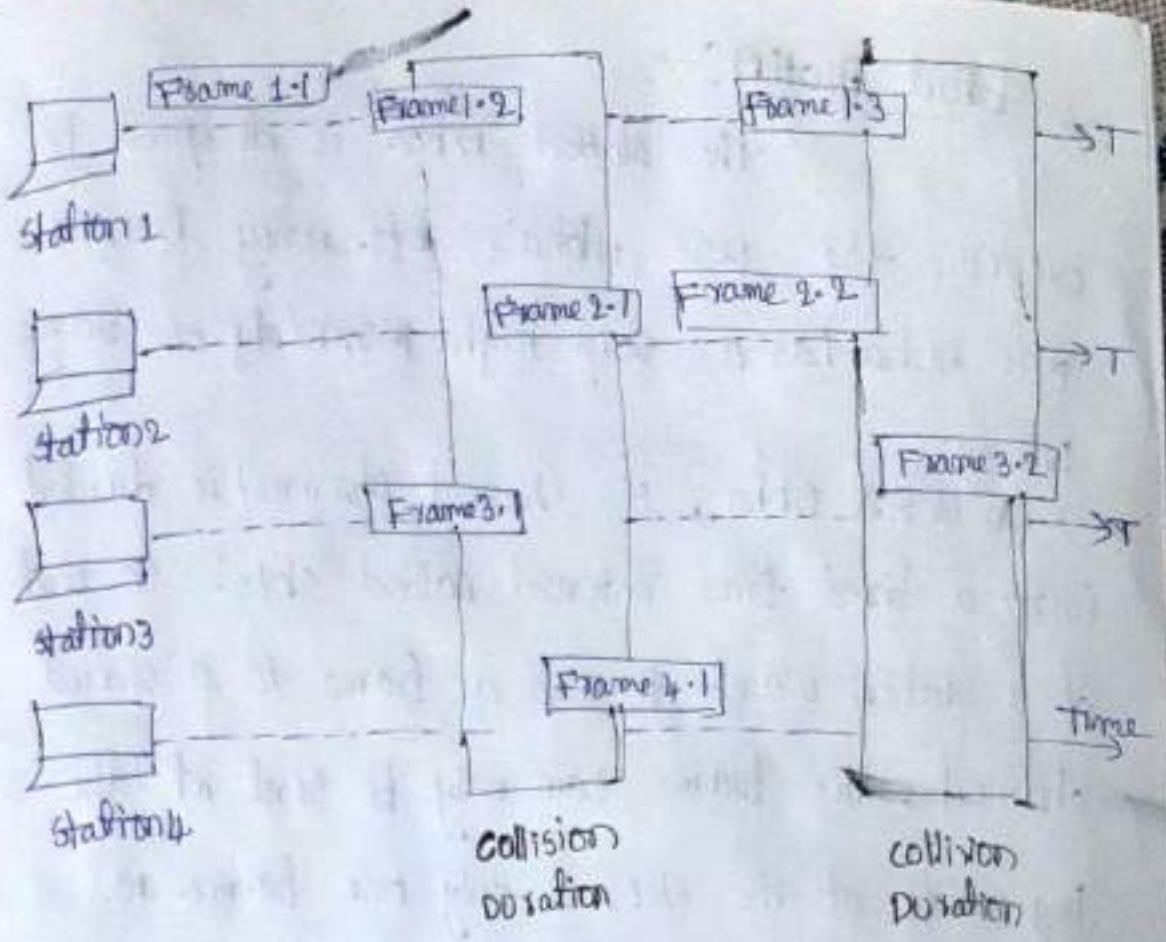


Fig :- Frames in pure ALOHA

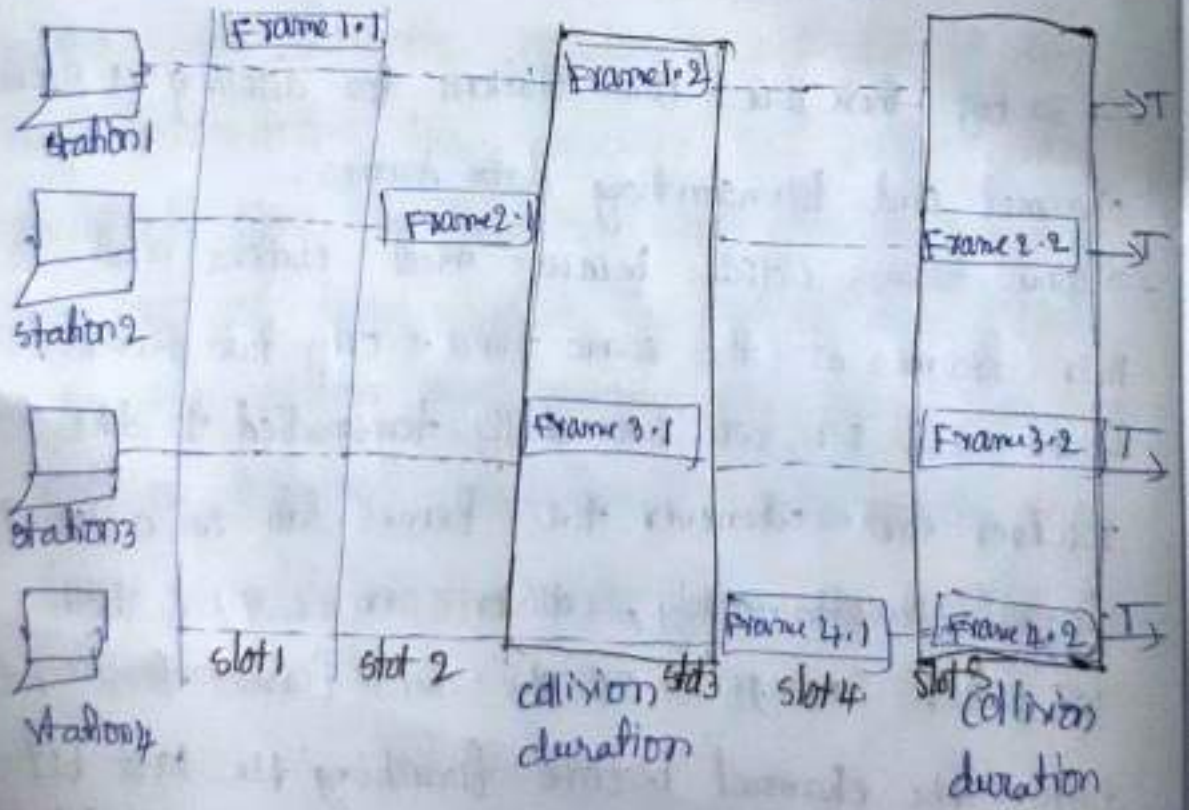
→ In fig. there are four stations for accessing a shared channel and transmitting data frames.

→ some frames collide because most stations send their frames at the same time. only two frames, frame 1-1 & 2-2, are successfully transmitted to the receiver end. whenever two frames fall on a shared channel simultaneously, collision can occur, & both will suffer damage. If the new frames first bit enters the channel before finishing the last bit of the second frame. Both frames are completely finished, & must retransmit the data frame.

Slotted ALOHA :-

the slotted Abha is designed to overcome the pure Abha's efficiency because pure Alohai has a very high possibility of frame hitting.

→ in slotted Abha, 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, & only one frame is allowed to be sent to each slot.



: Frames in slotted Aloha

→ 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.

Carrier Sense Multiple Access Protocols :- (CSMA)

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

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

CSMA access modes:—

1) 1-persistant:— the node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally (with 1 probability) as soon as the channel gets idle.

2) Non-persistent:— the node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.

3) P-persistent:— the node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time & checks the medium again, now if it is found idle then it send with p probability. this repeat continues until the frame is sent. It is used in wifi & packet radio systems.

4) O-persistent:— superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

CSMA/CD Carrier sense multiple Access with collision Detection:-

CSMA/CD is a n/w protocol for carrier transmission that operates in the MAC layer.

→ It senses or listens whether the shared channel for transmission is busy or not, and defers transmission until the channel is free. The collision detection technology detects collisions by sensing transmisions from other stations. On detection of a collision, the station stops transmitting, sends a jam signal, & then waits for a random time interval before retrans-mission.

Algorithms:-

→ when a frame is ready, the transmitting station checks whether the channel is idle or busy.

→ If the channel is busy, the station waits until the channel becomes idle.

→ If the channel is idle, the station starts transmitting & continuously monitors the channel to detect a collision.

→ If a collision is detected, the station starts the collision resolution algorithm.

→ The station resets the retransmission counters & completes frame transmission.

CSMA/CA (Carrier sense Multiple Access with collision Avoidance):—

CSMA/CA is a n/w protocol for carrier transmission that operates in the MAC layer. In contrast to CSMA/CD that deals with collisions after their occurrence, CSMA/CA prevents collisions prior to their occurrence.

Algorithm:—

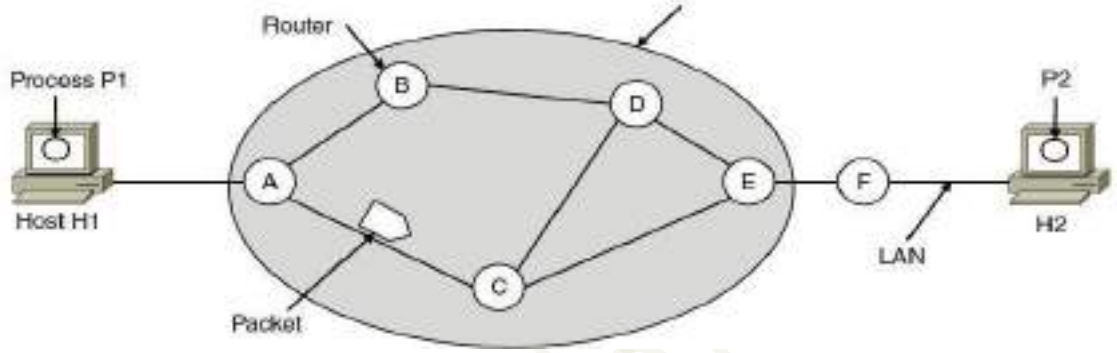
→ when a frame is ready, the transmitting station checks whether the channel is idle or busy.
→ if the channel is busy, the station waits until the channel becomes idle.
→ If the channel is idle, the station waits for an Inter-frame gap amount of time & then sends the frame.
→ After sending the frame, it sets a timer.
→ the station then waits for acknowledgement from the receiver. If it receives the ackt before expiry of timer, it marks a success-ful transmission.
→ otherwise, it waits for a back-off time period & restarts the algorithm.

UNIT-III
NETWORK LAYER

Network Layer Design Issues

Store-and-forward packet switching
Services provided to transport layer
Implementation of connectionless service
Implementation of connection-oriented service
Comparison of virtual-circuit and datagram networks

Store-and-forward packet switching

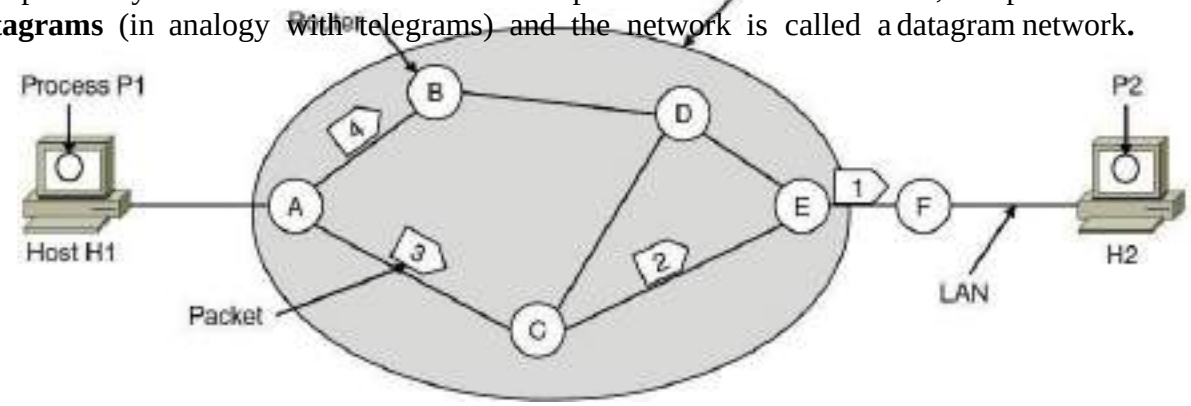


A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the ISP. The packet is stored there until it has fully arrived and the link has finished its processing by verifying the checksum. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

Services provided to transport layer
The network layer provides services to the transport layer at the network layer/transport layer interface. The services need to be carefully designed with the following goals in mind:
Services independent of router technology.
Transport layer shielded from number, type, topology of routers.
Network addresses available to transport layer use uniform numbering plan
– even across LANs and WANs

Implementation of connectionless service

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



A's table (initially) A's table (later) C's Table E's Table

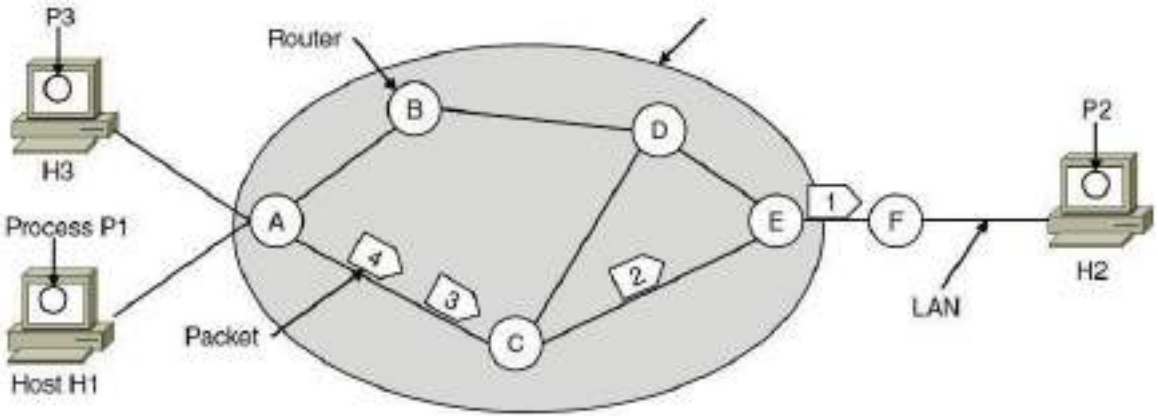
Dest.	Line
A	☒
B	B
C	C
D	B
E	C
F	C

A	☒
B	B
C	C
D	B
E	D
F	D

A	A
B	A
C	☒
D	E
E	E
F	E

A	C
B	D
C	C
D	D
E	☒
F	F

Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router *A.*
Every router has an internal table telling it where to send packets for each of the possible destinations. Each table entry is a pair(destination and the outgoing line). Only directly connected lines can be used.
A's initial routing table is shown in the figure under the label ''initially.''
At *A,* packets 1, 2, and 3 are stored briefly, having arrived on the incoming link. Then each packet is forwarded according to *A*'s table, onto the outgoing link to *C* within a new frame. Packet 1 is then forwarded to *E* and then to *F*.
However, something different happens to packet 4. When it gets to *A* it is sent to router *B*, even though it is also destined for *F*. For some reason (traffic jam along ACE path), *A* decided to send packet 4 via a different route than that of the first three packets. Router A updated its routing table, as shown under the label ''later.''
The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm**.
Implementation of connection-oriented service



A's table C's Table E's Table

In	
H1	1
H3	1

Out	
C	1
C	2

A	1
A	2

E	1
E	2

C	1
C	2

F	1
F	2

If connection-oriented service is used, a path from the source router all the way to the destination router must be established before any data packets can be sent. This connection is called a **VC** (**virtual circuit**), and the network is called a **virtual-circuit network**

When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling

which virtual circuit it belongs to.

As an example, consider the situation shown in Figure. Here, host *H1* has established connection 1 with host *H2*. This connection is remembered as the first entry in each of the routing tables. The first line of *A*'s table says that if a packet bearing connection identifier 1 comes in from *H1*, it is to be sent to router *C* and given connection identifier 1. Similarly, the first entry at *C* routes the packet to *E*, also with connection identifier 1.
Now let us consider what happens if *H3* also wants to establish a connection to *H2*. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the network to establish the virtual circuit.
This leads to the second row in the tables. Note that we have a conflict here because although *A* can easily distinguish connection 1 packets from *H1* from connection 1 packets from *H3*, *C* cannot do this. For this reason, *A* assigns a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets.
In some contexts, this process is called **label switching**. An example of a connection-oriented network service is **MPLS** (**Multi Protocol Label Switching**).

Comparison of virtual-circuit and datagram networks

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

Routing Algorithms

The main function of NL (Network Layer) is routing packets from the source machine to the destination machine.

There are two processes inside router:

One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing table. This process is forwarding.

The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play. This process is routing.

Regardless of whether routes are chosen independently for each packet or only when new connections are established, certain properties are desirable in a routing algorithm **correctness, simplicity, robustness, stability, fairness, optimality**

Routing algorithms can be grouped into two major classes:

nonadaptive (Static Routing)

adaptive. (Dynamic Routing)

Nonadaptive algorithm do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

Adaptive algorithm, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well.

Adaptive algorithms differ in

Where they get their information (e.g., locally, from adjacent routers, or from all routers),

When they change the routes (e.g., every ΔT sec, when the load changes or when the topology changes), and

What metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

This procedure is called dynamic routing

Different Routing Algorithms

Optimality principle

Shortest path algorithm

Flooding

Distance vector routing

Link state routing

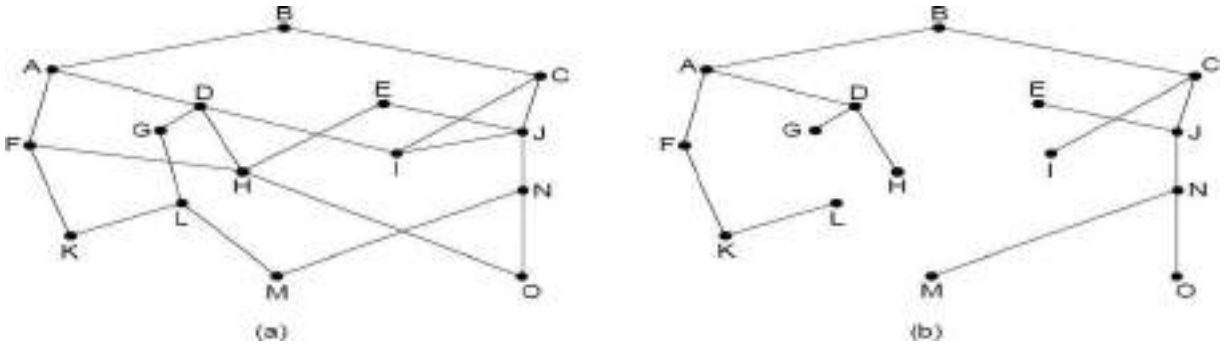
Hierarchical Routing

The Optimality Principle

One can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle.

It states that if router J is on the optimal path ~~from router I to router~~ K, then the optimal path from J to K also falls along the same

As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**. The goal of all routing algorithms is to discover and use the sink trees for all routers



(a) A network. (b) A sink tree for router *B.*

Shortest Path Routing (Dijkstra's)(very very important)

The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.

To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph

Start with the local node (router) as the root of the tree. Assign a cost of 0 to this node and make it the first permanent node.

Examine each neighbor of the node that was the last permanent node.

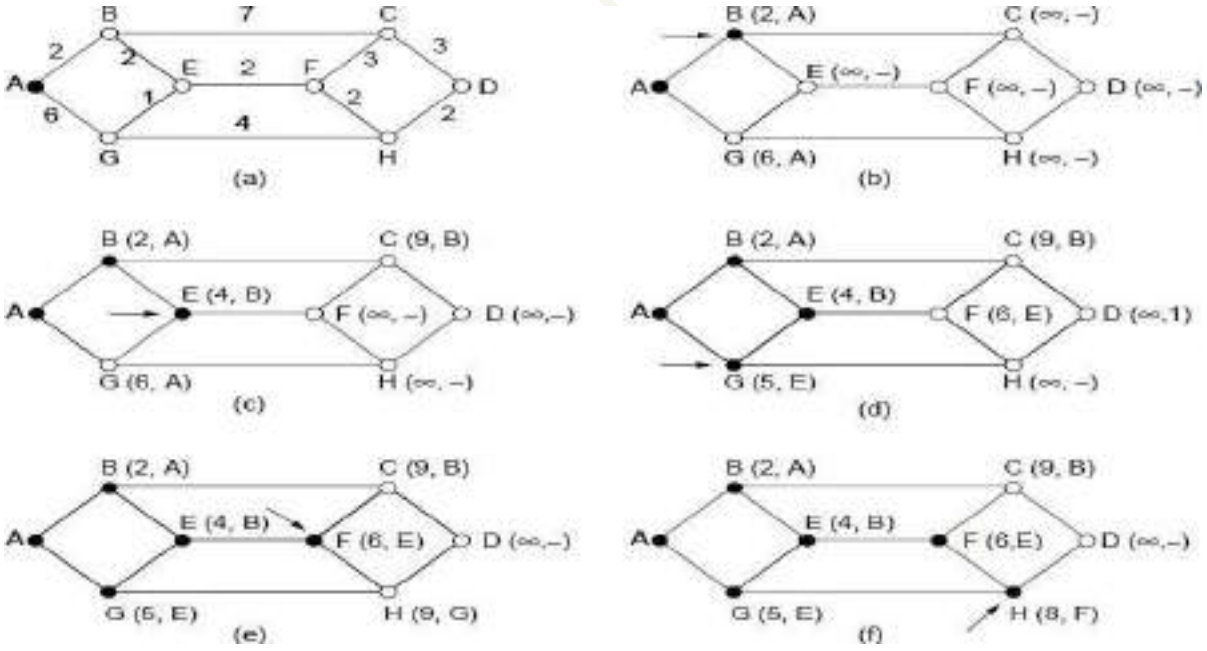
Assign a cumulative cost to each node and make it tentative

Among the list of tentative nodes

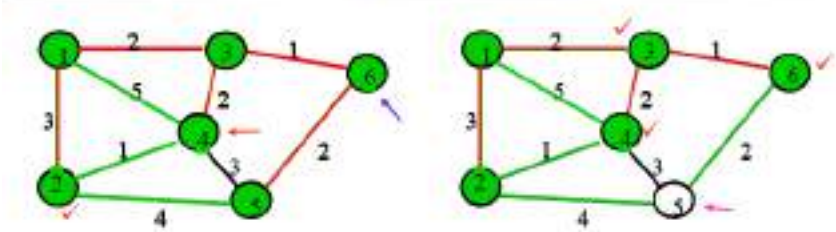
Find the node with the smallest cost and make it Permanent

If a node can be reached from more than one route then select the route with the shortest cumulative cost.

Repeat steps 2 to 4 until every node becomes permanent



Execution of Dijkstra's algorithm



Iteration	Permanent	tentative	D_2	D_3	D_4	D_5	D_6
Initial	{1}	{2,3,4}	3	2 ✓	5	∞	∞
1	{1,3}	{2,4,6}	3 ✓	2	4	∞	3
2	{1,2,3}	{4,6,5}	3	2	4	7	3 ✓
3	{1,2,3,6}	{4,5}	3	2	4 ✓	5	3
4	{1,2,3,4,6}	{5}	3	2	4	5 ✓	3
5	{1,2,3,4,5,6}	{}	3	2	4	5	3

Flooding

Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on.

Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.

One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination.

A variation of flooding that is slightly more practical is **selective flooding**. In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.

Flooding is not practical in most applications.

Distance Vector Routing:(very very important)

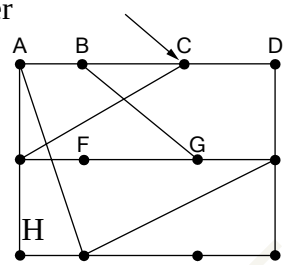
A **distance vector routing** algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there.

These tables are updated by exchanging information

with the neighbors. In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the network. This entry has two parts: the preferred outgoing line to use for that

destination and an estimate of the distance to that destination. The distance might be measured as the number of hops or using another metric.

This updating process is illustrated in Fig. 5-9. Part (a) shows a network. The first four columns of part (b) show the delay vectors received from the neighbors of router *J*. *A* claims to have a 12-msec delay to *B*, a 25-msec delay to *C*, a 40- msec delay to *D*, etc. Suppose that *J* has measured or estimated its delay to its neighbors, *A*, *I*, *H*, and *K*, as 8, 10, 12, and 6 msec, respectively.



(a)

New estimated delay from J

Router To	A	I	H	K	↓ Line	A
A	0	24	20	21		8
B	12	36	31	28		20
C	25	18	19	36		28
D	40	27	8	24		20
E	14	7	30	22		17
F	23	20	19	40		30
G	18	31	6	31		18
H	17	20	0	19		12
I	21	0	14	22		10
J	9	11	7	10		0
K	24	22	22	0		6
L	29	33	9	9		15

I	J	K	L	
JA	JI	JH	JK	
delay	delay	delay	delay	New
is	is	is	is	routing table for J
8	10	12	6	

Vectors received from J's four neighbors

(b)

Figure 5-9. (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Consider how *J* computes its new route to router *G*. It knows that it can get to *A* in 8 msec, and furthermore *A* claims to be able to get to *G* in 18 msec, so *J* knows it can count on a delay of 26 msec to *G* if it forwards packets bound for *G*

to *A*. Similarly, it computes the delay to *G* via *I*, *H*, and *K* as 41 (31 + 10), 18(6 + 12), and 37 (31 + 6) msec, respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to *G* is 18 msec and that the route to use is via *H*. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.

The Count-to-Infinity Problem

The settling of routes to best paths across the network is called **convergence**. Distance vector routing is useful as a simple technique by which routers can col- lectively compute shortest paths, but it has a serious drawback in practice: al- though it converges to the correct answer, it may do so slowly. In particular, it reacts rapidly to good news, but leisurely to bad news. Consider a router whose best route to destination *X* is long. If, on the next exchange, neighbor *A* suddenly reports a short delay to *X*, the router just switches over to using the line to *A* to send traffic to *X*. In one vector exchange, the good news is processed.

To see how fast good news propagates, consider the five-node (linear) net- work of Fig. 5-10, where the delay metric is the number of hops. Suppose *A* is down initially and all the other routers know this. In other words, they have all recorded the delay to *A* as infinity.

A	B	C	D	E		A	B	C	D	E	
•	□•	□•	□•	□•	Initially	•	1 •	2 •	3 •	4 •	Initially
	1	□	□	□	After exchange	1	3	2	3	4	After exchange 1
	1	2	□	□	After exchanges	2	3	4	3	4	After exchanges 2
	1	2	3	□	After exchanges	3	5	4	5	4	After exchanges 3
	1	2	3	4	After exchanges	4	5	6	5	6	After exchanges 4
							7	6	7	6	After exchanges 5
							7	8	7	8	After exchanges 6
							□	. □.	□	□	

(a) (b)

Figure 5-10. The count-to-infinity problem.

When *A* comes up, the other routers learn about it via the vector

exchanges. For simplicity, we will assume that there is a gigantic gong somewhere that is struck periodically to initiate a vector exchange at all routers simultaneously. At the time of the first exchange, *B* learns that its left-hand neighbor has zero delay to *A*. *B* now makes an entry in its routing table indicating that *A* is one hop away to the left. All the other routers still think that *A* is down. At this point, the rout- ing table entries for *A* are as shown in the second row of Fig. 5-10(a). On the next exchange, *C* learns that *B* has a path of length 1 to *A*, so it updates its routing table to indicate a path of length 2, but *D* and *E* do not hear the good news until later. Clearly, the good news is spreading at the rate of one hop per exchange. In a net- work whose longest path is of length *N* hops, within *N* exchanges everyone will know about newly revived links and routers.

Now let us consider the situation of Fig. 5-10(b), in which all the links and routers are initially up. Routers *B*, *C*, *D*, and *E* have distances to *A* of 1, 2, 3, and 4 hops, respectively. Suddenly, either *A* goes down or the link between *A* and *B* is cut (which is effectively the same thing from *B*'s point of view).

At the first packet exchange, *B* does not hear anything from *A*. Fortunately, *C* says ''Do not worry; I have a path to *A* of length 2.'' Little does *B* suspect that *C*'s path runs through *B* itself. For all *B* knows, *C* might have ten links all with sepa- rate paths to *A* of length 2. As a result, *B* thinks it can reach *A* via *C*, with a path length of 3. *D* and *E* do not update their entries for *A* on the first exchange.

On the second exchange, *C* notices that each of its neighbors claims to have a path to *A* of length 3. It picks one of them at random and makes its new distanceto *A* 4, as shown in the third row of Fig. 5-10(b). Subsequent exchanges produce the history shown in the rest of Fig. 5-10(b).

From this figure, it should be clear why bad news travels slowly: no router ever has a value more than one higher than the minimum of all its neighbors. Gradually, all routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity. For this reason, it is wise to set infinity to the longest path plus 1.

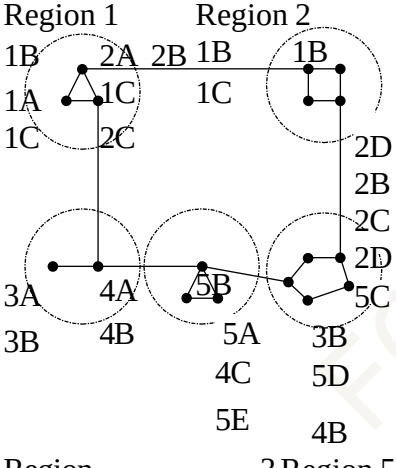
Not entirely surprisingly, this problem is known as the **count-to-infinity** prob-lem.

Hierarchical Routing:

As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them. At a certain point, the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.

When hierarchical routing is used, the routers are divided into what we will call **regions**. Each router knows all the details about how to route packets to dest- inations within its own region but knows nothing about the internal structure of other regions. When different networks are interconnected, it is natural to regard each one as a separate region to free the routers in one network from having to know the topological structure of the other ones.For huge networks, a two-level hierarchy may be insufficient; it may be nec- essary to group the regions into clusters, the clusters into zones, the zones into groups, and so on.

Figure 5-14 gives a quantitative example of routing in a two-level hierarchy with five regions. The full routing table for router *1A* has 17 entries, as shown in Fig. 5-14(b). When routing is done hierarchically, as in Fig. 5-14(c), there are en- tries for all the local routers, as before, but all other regions are condensed into a single router, so all traffic for region 2 goes via the *1B-2A* line, but the rest of the remote traffic goes via the *1C-3B* line. Hierarchical routing has reduced the table from 17 to 7 entries. As the ratio of the number of regions to the number of rout- ers per region grows, the savings in table space increase.



(a)

(b)

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

(a)

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

Figure 5-14. Hierarchical routing.

Broadcast Routing

In some applications, hosts need to send messages to many or all other hosts. Sending a packet to all destinations simultaneously is called broadcasting. Various methods have been proposed for doing it.

One broadcasting method that requires no special features from the network is for the source to simply send a distinct packet to each destination.

multidestination routing, in which each packet contains either a list of destinations or a bit map indicating the desired destinations. Whena packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed.

flooding. When implemented with a sequence number per source, flooding uses links efficiently with a decision rule at routers that is relatively simple.

reverse path forwarding: When a broadcast packet ar- rives at a router, the router checks to see if the packet arrived on the link that is normally used for sending packets *toward* the source of the broadcast. If so, there is an excellent chance that the broadcast packet itself followed the best route from the router.

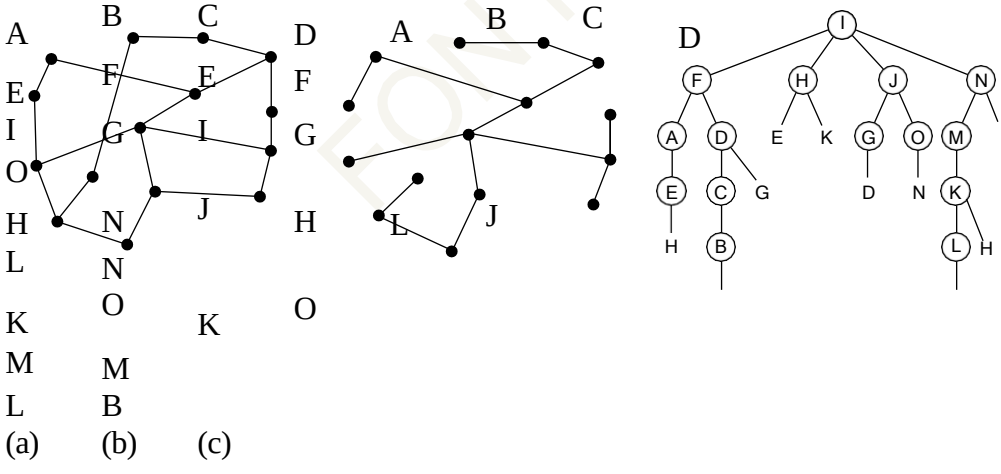


Figure 5-15. Reverse path forwarding. (a) A network. (b) A sink tree. (c) Thetree built by reverse path forwarding.

An example of reverse path forwarding is shown in Fig. 5-15. Part (a) shows a network, part (b) shows a sink tree for router *I* of that network, and part (c) shows how the reverse path algorithm works. On the first hop, *I* sends packets to *F*, *H*, *J*, and *N*, as indicated by the second row of the tree. Each of these packets arrives on the preferred path to *I* (assuming that the preferred path falls along the sink tree) and is so indicated by a circle around the letter. On the second hop,

eight packets are generated, two by each of the routers that received a packet on the first hop. As it turns out, all eight of these arrive at previously unvisited rout- ers, and five of these arrive along the preferred line. Of the six packets generated on the third hop, only three arrive on the preferred path (at *C*, *E*, and *K*); the oth- ers are duplicates. After five hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sink tree been followed exactly.

A **spanning tree** is a subset of the network that includes all the routers but contains no loops. Sink trees are spanning trees. If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on.

Multicast Routing

Some applications, such as a multiplayer game or live video of a sports event streamed to many viewing locations, send packets to multiple receivers.

Sending a message to such a group is called **multicasting**, and the routing al- gorithm used is called **multicast routing**. All multicasting schemes require some way to create and destroy groups and to identify which routers are members of a group.

As an example, consider the two groups, 1 and 2, in the network shown in Fig. 5-16(a). Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure. A spanning tree for the leftmost router is shown in Fig. 5-16(b). This tree can be used for broadcast but is overkill for mu- lticast, as can be seen from the two pruned versions that are shown next. In Fig. 5-16(c), all the links that do not lead to hosts that are members of group 1 have been removed. The result is the multicast spanning tree for the leftmost router to send to group 1. Packets are forwarded only along this spanning tree, which is more efficient than the broadcast tree because there are 7 links instead of

10. Fig. 5-16(d) shows the multicast spanning tree after pruning for group

2. It is efficient too, with only five links this time. It also shows that different multicast groups have different spanning trees.

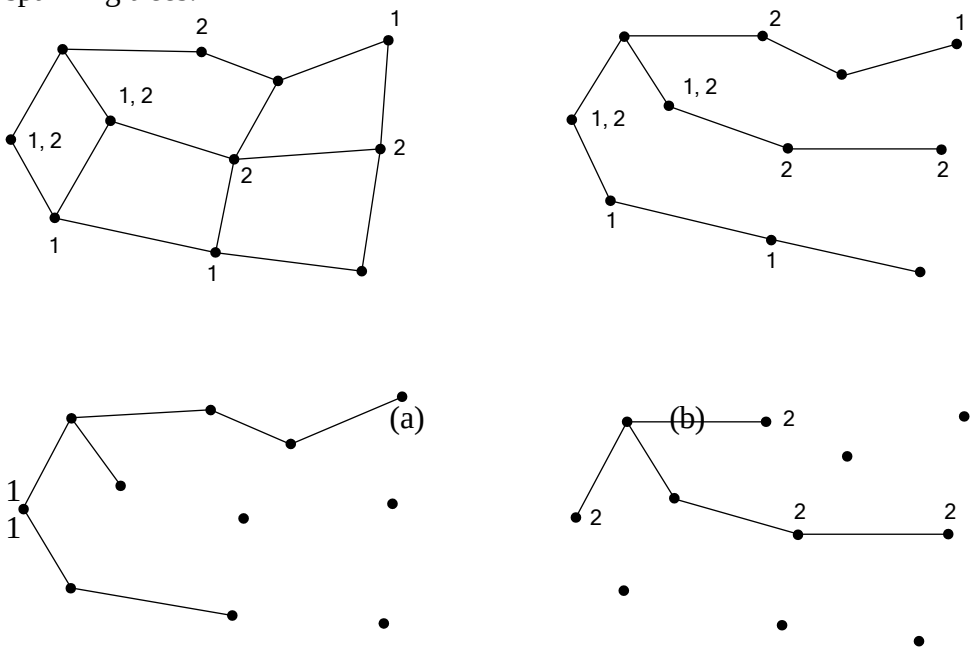


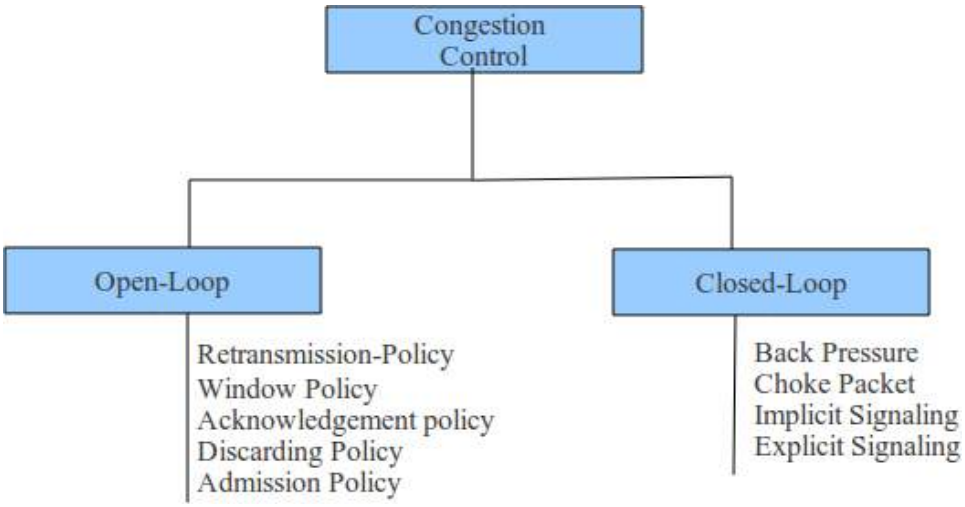
Figure 5-16. (a) A network. (b) A spanning tree for the leftmost router. (c) Amulticast tree for group 1. (d) A multicast tree for group 2.

Congestion control:

Congestion in a network may occur if the load on the network(the number of packets sent to the network) is

greater than the capacity of the network(the number of packets a network can handle). Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity .
When too many packets are pumped into the system, congestion occur leading into degradation ofperformance.
Congestion tends to feed upon itself and back ups.
Congestion shows lack of balance between various networking equipments.
It is a global issue.

In general, we can divide congestion control mechanisms into two broad categories: open- loop congestion control (prevention) and closed-loop congestion control (removal) as shown in Figure



Open Loop Congestion Control:
In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination

Retransmission Policy

Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion. For example, the retransmission policy used by TCP is designed to prevent or alleviate congestion.

Window Policy

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the Go-Back-N window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.
Acknowledgment Policy :
The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion. Several

approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A receiver may decide to acknowledge only N packets at a time. We need to know that theacknowledgments are also part of the load in a network. Sending fewer acknowledgments means imposing less load on the network.

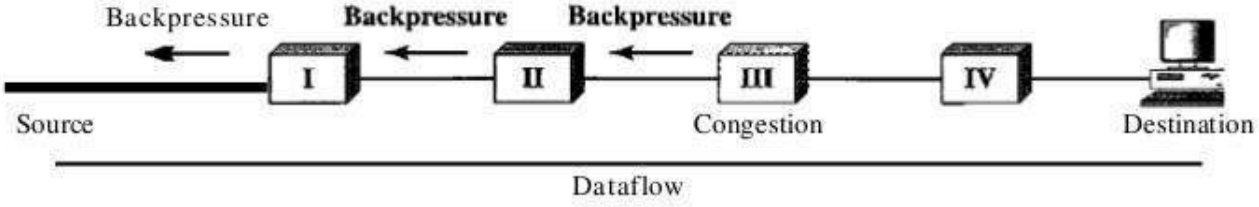
Discarding Policy :

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrityof the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

Admission Policy :
An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks. Switches in a flow first check the resource requirement of a flow before admitting it to the network. A router can deny establishing a virtual- circuit connection if there is congestion in the network or if there is a possibility of future congestion.

Closed-Loop Congestion Control
Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols.

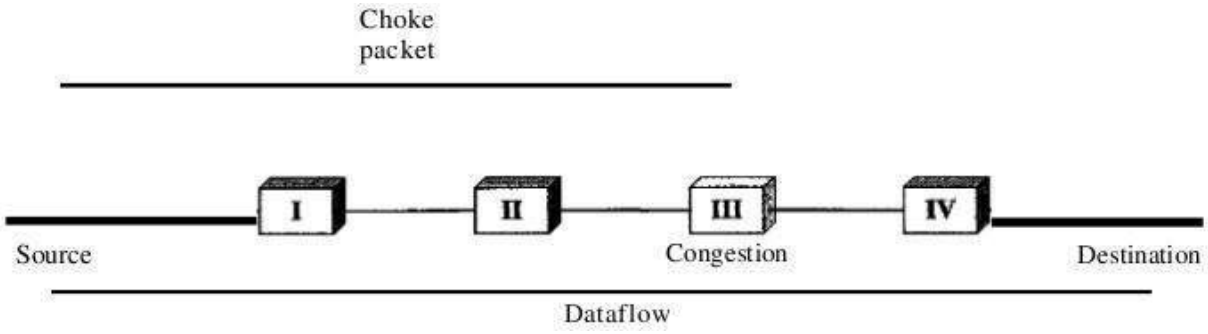
Back-pressure:
The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream nodes or nodes. And so on. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is corning.



Node III in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node II to slow down. Node II, in turn, may be congested because it is slowing down the output flow of data. If node II is congested, it informs node I to slow down, which in turn may create congestion. If so, node I informs the source of data to slow down. This, in time, alleviates the congestion. Note that the pressure on node III is moved backward to the source to remove the congestion. None of the virtual-circuit networks we studied inthis book use backpressure. It was, however, implemented in the first virtual-circuit network, X.25. The technique cannot be implemented in a datagram network because in this type of network, a node (router) does not have the slightest knowledge of the upstream router.

Choke Packet
A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke packet methods. In backpresure, the warning is from one node to its upstream node, although the warning may eventually reach the source station. In the choke packet method, the

warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned. We have seen an example of this type of control in ICMP. When a router in the Internet is overwhelmed datagrams, it may discard some of them; but it informs the source . host, using a source quench ICMP message. The warning message goes directly to the source station; the intermediate routers, and does not take any action. Figure shows the idea of a choke packet.



Implicit Signaling

In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is a congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down. We will see this type of signaling when we discuss TCP congestion control later in the chapter.

Explicit Signaling

The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data. Explicit signaling, as we will see in Frame Relay congestion control, can occur in either the forward or the backward direction.

Backward Signaling A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.

Forward Signaling A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

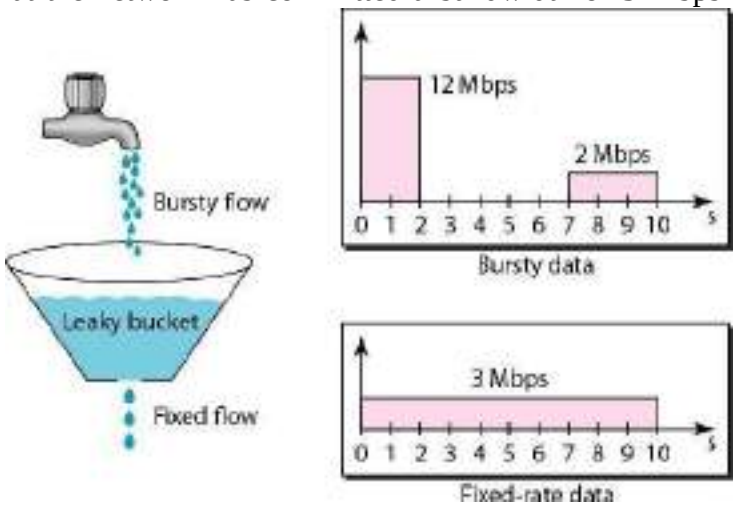
Traffic Shaping

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket and token bucket.

Leaky Bucket

If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rateas long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate. Figure 4.34 shows a leaky bucket and its effects.

In the figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use of the



leaky bucket shapes the input traffic to make it conform to this commitment. In Figure 4.34 the host sends a burst of data at a rate of 12 Mbps for 2 s, for a total of 24 Mbits ofdata. The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 Mbits of data. In all, the host has sent 30 Mbits of data in lOs. The leaky bucket smooth's the traffic by sending out data at a rate of 3 Mbps during the same 10 s.

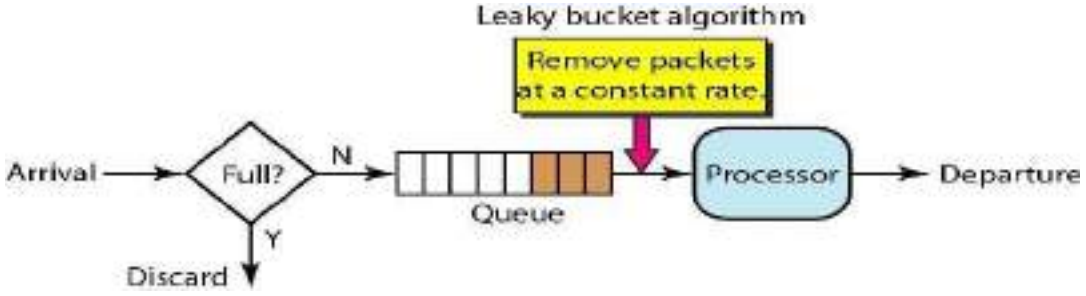


Fig:Leaky bucket implementation

A simple leaky bucket implementation is shown in Figure 4.35. A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consistsof variable-length packets, the fixed output rate must be based on the number of bytes or bits.

The following is an algorithm for variable-length packets:

Initialize a counter to n at the tick of the clock.

If n is greater than the size of the packet, send the packet and decrement the counter bythe packet size. Repeat this step until n is smaller than the packet size.

Reset the counter and go to step 1.

A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

Token Bucket

The leaky bucket is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty. Now if the host has bursty data, the leaky bucket allows only an average rate. The time when the host was idle is not taken into account. On the other hand, the token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends n tokens to the bucket. The system removes one token for every cell (or byte) of data sent. For example, if n is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens.

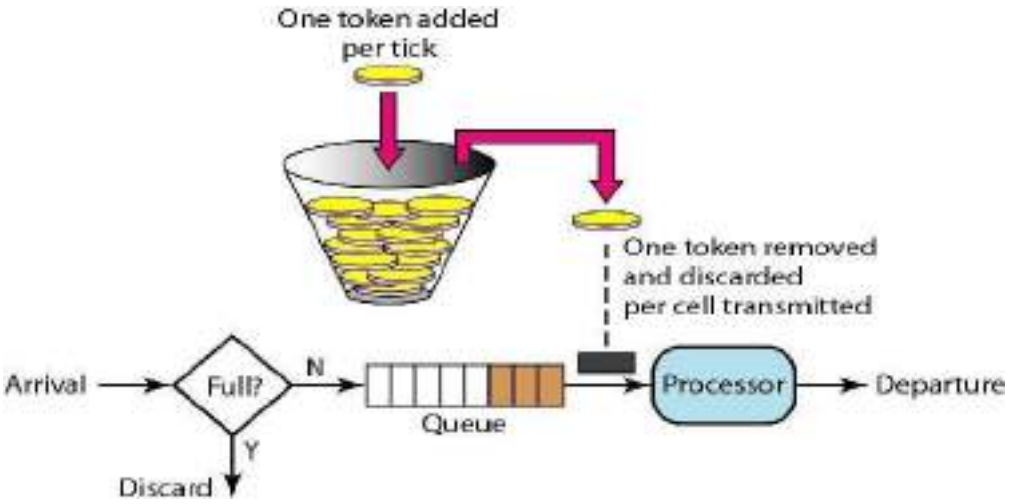


Figure 4.36 Token bucket

The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1. When the counter is zero, the host cannot send data.
The token bucket allows bursty traffic at a regulated maximum rate.

Combining Token Bucket and Leaky Bucket

The two techniques can be combined to credit an idle host and at the same time regulate the traffic. The leaky bucket is applied after the token bucket; the rate of the leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

Resource Reservation

A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand. We discuss in this section one QoS model called Integrated Services, which depends heavily on resource reservation to improve the quality of service.

Admission Control

Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.
7 Quality of services:
QoS is an overall performance measure of the computer network.

Important flow characteristics of the QoS are given below:
Reliability
If a packet gets lost or acknowledgement is not received (at sender), the re-transmission of data will be needed. This decreases the reliability.
The importance of the reliability can differ according to the application.
For example:
E- mail and file transfer need to have a reliable transmission as compared to that of an audio conferencing.
Delay
Delay of a message from source to destination is a very important characteristic. However, delay can be tolerated differently by the different applications.
For example:
The time delay cannot be tolerated in audio conferencing (needs a minimum time delay), while the time delay

in the e-mail or file transfer has less importance.

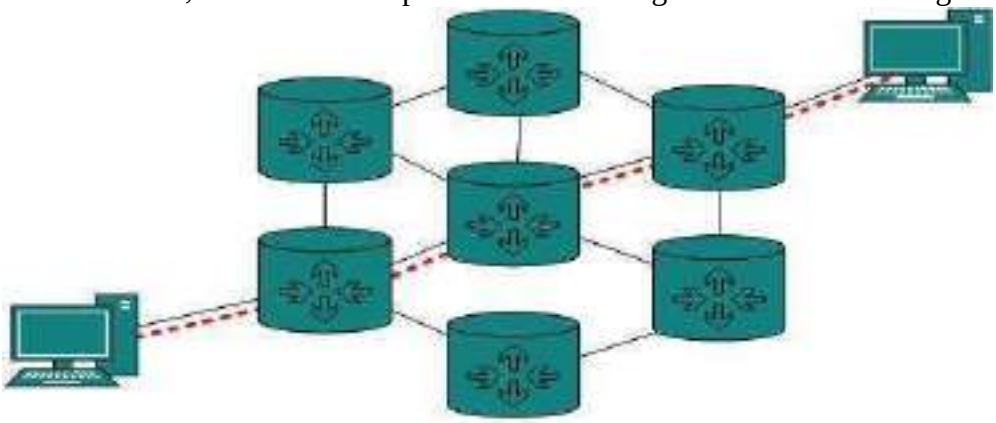
Jitter
The jitter is the variation in the packet delay.
If the difference between delays is large, then it is called as **high jitter.** On the contrary, if the difference between delays is small, it is known as **low jitter.**
Example:
Case1: If 3 packets are sent at times 0, 1, 2 and received at 10, 11, 12. Here, the delay is same for all packets and it is acceptable for the telephonic conversation.
Case2: If 3 packets 0, 1, 2 are sent and received at 31, 34, 39, so the delay is different for all packets. In this case, the time delay is not acceptable for the telephonic conversation.

Bandwidth
Different applications need the different bandwidth.
For example:
Video conferencing needs more bandwidth in comparison to that of sending an e-mail. Integrated Services and Differentiated Service
These two models are designed to provide Quality of Service (QoS) in the network.
Integrated Services(IntServ)
Integrated service is flow-based QoS model and designed for IP.
In integrated services, user needs to create a flow in the network, from source to destination and needs to inform all routers (every router in the system implements IntServ) of the resource requirement.
Following are the steps to understand how integrated services works.
I) Resource Reservation Protocol (RSVP)
An IP is connectionless, datagram, packet-switching protocol. To implement a flow-based model, a signaling protocol is used to run over IP, which provides the signaling mechanism to make reservation (every applications need assurance to make reservation), this protocol is called as RSVP.
Flow Specification
While making reservation, resource needs to define the flow specification. The flow specification has two parts:
Resource specification
It defines the resources that the flow needs to reserve. For example: Buffer, bandwidth, etc.
Traffic specification
It defines the traffic categorization of the flow.

Admit or deny
After receiving the flow specification from an application, the router decides to admit or deny the service and the decision can be taken based on the previous commitments of the router and current availability of the resource.
Classification of services

Internetworking:
In real world scenario, networks under same administration are generally scattered geographically. There may exist requirement of connecting two different networks of same kind as well as of different kinds. Routing between two networks is called internetworking.
Networks can be considered different based on various parameters such as, Protocol, topology, Layer-2 network and addressing scheme.

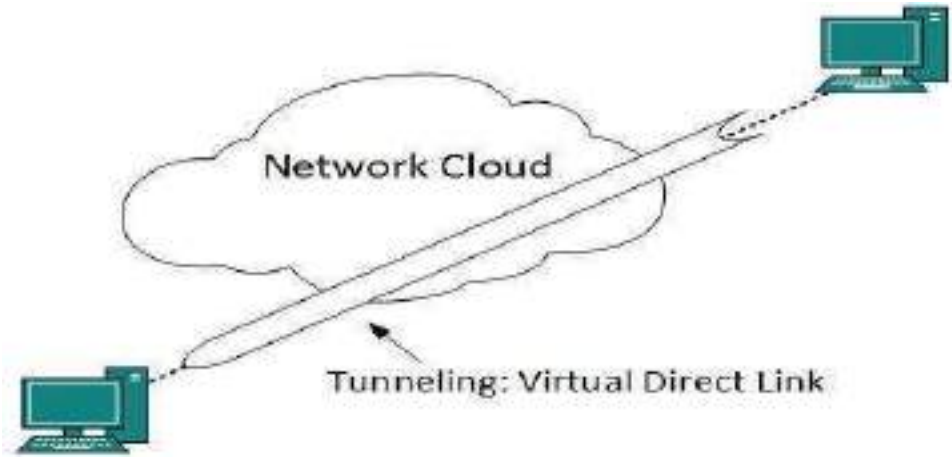
In internetworking, routers have knowledge of each other's address and addresses beyond them. They can be statically configured go on different network or they can learn by using internetworking routing protocol.
Routing protocols which are used within an organization or administration are called Interior Gateway Protocols or IGP. RIP, OSPF are examples of IGP. Routing between different organizations or administrations



may have Exterior Gateway Protocol, and there is only one EGP i.e. Border Gateway Protocol.

Tunneling

If they are two geographically separate networks, which want to communicate with each other, they may deploy a dedicated line between or they have to pass their data through intermediate networks.
Tunneling is a mechanism by which two or more same networks communicate with each other, by passing intermediate networking complexities. Tunneling is configured at both ends.



When the data enters from one end of Tunnel, it is tagged. This tagged data is then routed inside the intermediate or transit network to reach the other end of Tunnel. When data exists the Tunnel its tag is removed and delivered to the other part of the network.
Both ends seem as if they are directly connected and tagging makes data travel throughtransit network without any modifications.

Packet Fragmentation
Fragmentation is an important function of network layer. It is technique in which gateways break up or divide larger packets into smaller ones called fragments. Each fragment is then sent as a separate internal packet. Each fragment has its separate header and trailer.
Sometimes, a fragmented datagram also get fragmented when it encounter a network that handle smaller fragments. Thus, a datagram can be fragmented several times before it reaches final destination. Reverse process of the fragmentation is difficult.
Reassembly of fragments is usually done by the destination host because each fragment has become an

independent datagram.
For the reference of **example** of fragmentation you can refer : Fragmentation Example
There are two different strategies for the recombination or we can say reassembly offragments : Transparent Fragmentation, and Non-Transparent Fragmentation.

Transparent Fragmentation :

This fragmentation is done by one network is made transparent to all other subsequent networks through which packet will pass. Whenever a large packetarrives at a gateway, it breaks packet into smaller fragments as shown in the following figure gateway G1 breaks a packet into smaller fragments.

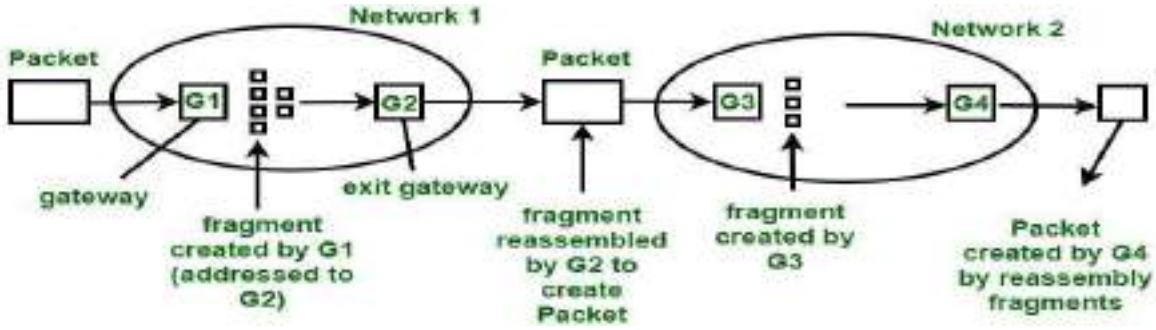


Figure – Transparent Fragmentation
After this, each fragment is going to address to same exit gateway. Exist gateway of a network reassembles or recombines all fragments example is shown in the above figure as exit gateway, G2 of network 1 recombines all fragments created byG1 before passing them to network 2. Thus, subsequent network is not aware that fragmentation has occurred. This type of strategy is used by ATM networks .
These networks use special hardware that provides transparent fragmentation ofpackets.
There are some disadvantages of transparency strategy which are as follows :
Exit fragment that recombines fragments in a network must known when it hasreceived all fragments.
Some fragments chooses different gateways for exit that results in poorperformance.
It adds cosiderable overhead in repeatedly fragmenting and reassembling largepacket.

Non-Transparent Fragmentation :
This fragmentation is done by one network is non-transparent to the subsequent networks through which a packet passes. Packet fragmented by a gateway of a network is not recombined by exit gateway of same network as shown in the belowfigure.

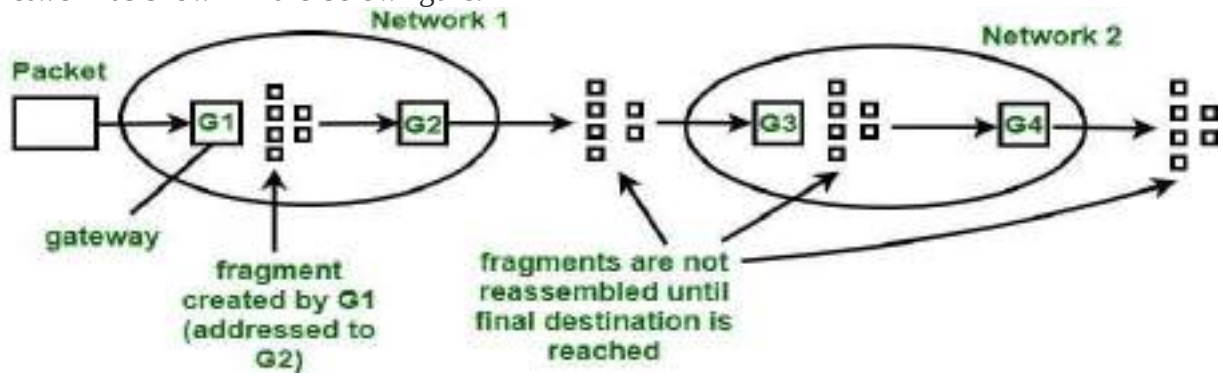
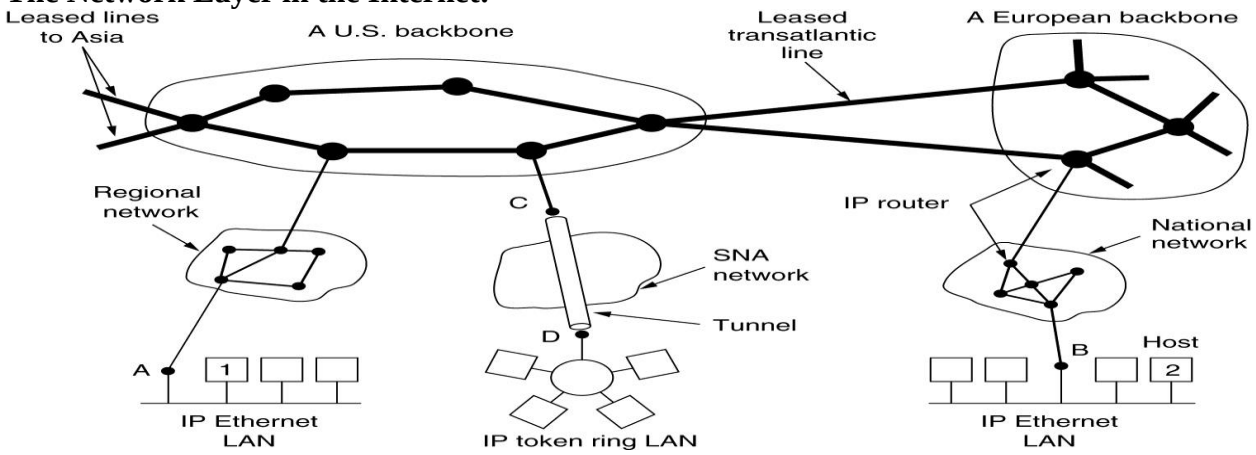


Figure – Non-transparent Fragmentation
Once a packet is fragmented, each fragment is treated as original packet. All fragments of a packet are passed through exit gateway and recombination of thesefragments is done at the destination host.
Disadvantages of Non-Transparent Fragmentation is as follows :
Every host has capability of reassembling fragments.
When a packet is fragmented, fragments should be numbered in such a way that the original data

stream can be reconstructed.

Total overhead increases due to fragmentation as each fragment must have itsown header.

The Network Layer in the Internet:



The Internet is an interconnected collection of many networks.

The network layer uses IP protocol to transmit the data grams .IP is unreliable,connectionless protocol.There are 2 versions of IP.IP v4,IP v6

IPv4

An **IPv4** address is a **32-bit** address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

IPv4 addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time. But by using some strategies, an address may be assigned to a device for a time period and then taken away and assigned to another device.

On the other hand, if a device operating at the network layer has ***m*** connections to the Internet, it needs to have ***m*** addresses. A router is such a device which needs as many IP addresses as the number of ports are there in it.

Address Space

A protocol such as IPv4 that defines addresses has an address space.

- ***An address space is the total number of addresses used by the protocol***. If a protocol uses ***N* bits** to define an address, the **address space is 2^N** because each bit can have two different values (0 or 1) and *N* bits can have 2^N values.
- **IPv4 uses 32-bit addresses**, which means that the **address space is 2^{32}** or

This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet. But the actual number is much less because of the restrictions imposed on the addresses.

IPv4 Address Notations

There are two prevalent notations to show an IPv4 address:

Binary notation and

Dotted decimal notation.

Binary Notation

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So it

is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. The following is an example of an IPv4 address in binary notation:

01110101 10010101 00011101 00000010

Dotted-Decimal Notation

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted decimal notation of the above address:

117.149.29.2

Figure 19.1 shows an IPv4 address in both binary and dotted-decimal notation. Note that because each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

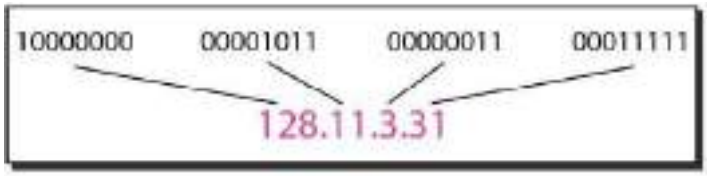


Figure 19.1 Dotted-decimal notation and binary notation for an IPv4 address

Example 19.1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

10000001 00001011 00001011 11101111

11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) andadd dots for separation.

129.11.11.239

193.131.27.255

Example 19.2

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

111.56.45.78

221.34.7.82

Solution

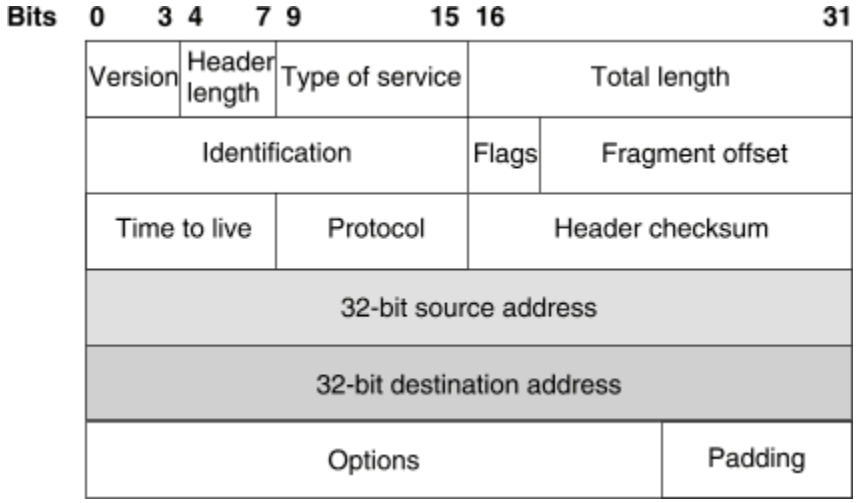
We replace each decimal number with its binary equivalent.

01101111 00111000 00101101 01001110

11011101 00100010 00000111 01010010

IPv4 Header

Fig: IPV4 Header



An IPv4 header contains the following fields:

version The IP version number, 4.

lengthThe length of the datagram header in 32-bit words.

type of service Contains five subfields that specify the precedence, delay, throughput, reliability, and cost desiredfor a packet. (The Internet does not guarantee this request.) This field is not widely used on the Internet.

total length The length of the datagram in bytes including the header, options, and the appended transport protocol segment or packet.

Identification An integer that identifies the datagram.

Flags: Controls datagram fragmentation together with the identification field. The flags indicate whether the datagram may be fragmented, whether the datagram is fragmented, and whether the current fragment is the finalone.

fragment offset The relative position of this fragment measured from the beginning of the original datagram inunits of 8 bytes.

time to live How many routers a datagram can pass through. Each router decrements this value by 1 until itreaches 0 when the datagram is discarded. This keeps misrouted datagrams from remaining on the Internet forever.

Protocol The high-level protocol type.

header checksum A number that is computed to ensure the integrity of the header values.

source address The 32-bit IPv4 address of the sending host.

destination address The 32-bit IPv4 address of the receiving host.

Options A list of optional specifications for security restrictions, route recording, and source routing. Not everydatagram specifies an options field.

Padding Null bytes which are added to make the header length an integral multiple of 32 bytes as required bythe header length field.

Ipv6 header*:*

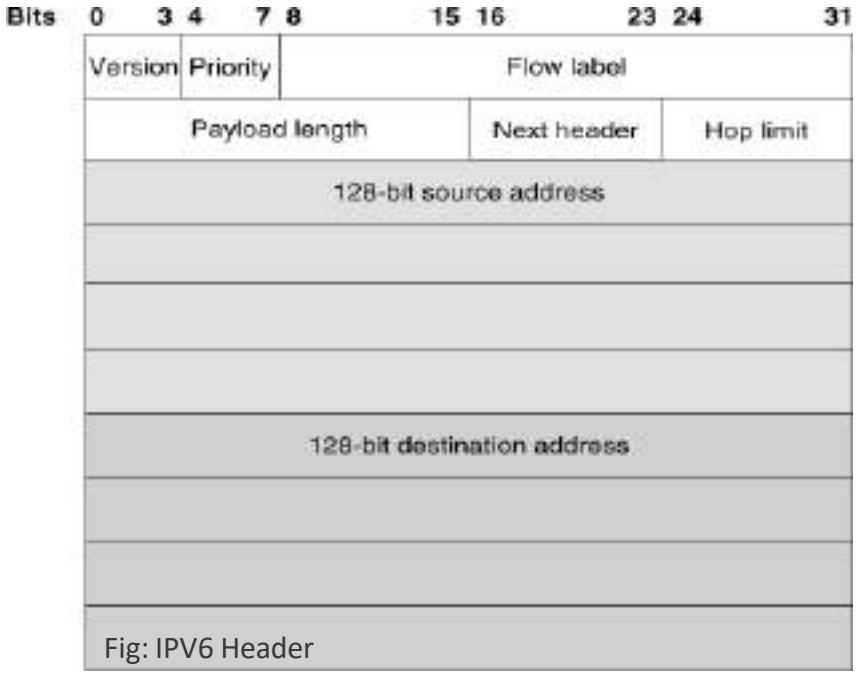


Fig: IPV6 Header

Specifically, IPv6 omits the following fields in its header.
header length (the length is constant)
identification
flags
fragment offset (this is moved into fragmentation extension headers)
header checksum (the upper-layer protocol or security extension header handles data integrity)
IPv6 options improve over IPv4 by being placed in separate extension headers that are located between the IPv6 header and the transport-layer header in a packet. Most extension headers are not examined or processed by any router along a packet's delivery path until it arrives at its final destination. This mechanism improves router performance for packets containing options. In IPv4, the presence of any options requires the router to examine all options.

Another improvement is that IPv6 extension headers, unlike IPv4 options, can be of arbitrary length and the total amount of options that a packet carries is not limited to 40 bytes. This feature, and the manner in which it is processed, permit IPv6 options to be used for functions that were not practical in IPv4, such as the IPv6 Authentication and Security Encapsulation options.
By using extension headers, instead of a protocol specifier and options fields, newly defined extensions can be integrated more easily into IPv6.

IPV6 Addressing:
Address Representation:
Represented by breaking 128 bit into Eight 16-bit segments (Each 4 Hex character each)Each segment is written in Hexadecimal separated by colons.
Hex digit are not case sensitive.
Rule 1:
Drop leading zeros: 2001:0050:0000:0235:0ab4:3456:456b:e560
2001:050:0:235:ab4:3456:456b:e560
Rule2:
Successive fields of zeros can be represented as "::" , But double colon appear only once in the address.
FF01:0:0:0:0:0:0:1
FF01::1

Note : An address parser identifies the number of missing zeros by separating the two parts and entering 0 until the 128 bits are complete. If two "::" notations are placed in the address, there is no way to identify the size of each block of zeros.

Ipv4 vs ipv6

IPV4	IPV6
1. source and destination addresses are 32 bits.)	1. Source and destination addresses are 128 bits.
2. ipv4 support small address space.	2. Supports a very large address space sufficeint foreach and every people on earth.
3. ipv4 header includes checksum.	3. ipv6 header doesn't includes the checksum. (theupper-layer protocol or security extension header handles data integrity)
4. addresses are represented in dotted decimal format.(Eg. 192.168.5.1)	4. Addresses are represented in 16-bit segments Each segment is written in Hexadecimal separated bycolons. (Eg. 2001:0050:020c:0235:0ab4:3456:456b:e560
5. Header includes options.	All optional data is moved to IPV6 extension header..
6. Broadcast address are used to send traffic to all	6. There is no IPV6 broadcast address. Instead a link
nodes on a subnet.	local scope all-nodes multicast address is used.
7. No identification of packet flow for QOS handlingby router is present within the ipv4 header.	7. Packet flow identification for QOS handling by routers is present within the IPV6 header using theflow label field.
8. uses host address (A) resource records in the Domain name system(DNS) to map host names toipv4 addresses.	8. Uses AAAA records in the DNS to map host namesto ipv6 addresses.
9. Both routers and the sending host fragmentpackets.	9. Only the sending host fragments packets; routersdo not.
10. ICMP Router Discovery is used to determine theIPv4 address of the best default gateway, and it is optional.	10. ICMPv6 Router Solicitation and Router Advertisement messages are used to determine the IPaddress of the best default gateway, and they are required.

Classless Inter-Domain Routing (CIDR)

Classless Inter-Domain Routing (CIDR) is a group of IP addresses that are allocated to the customer when they demand a fixed number of IP addresses.

In CIDR there is no wastage of IP addresses as compared to classful addressing because only the numbers of IP addresses that are demanded by the customer are allocated to the customer.

The group of IP addresses is called Block in Classless Inter - Domain (CIDR).

CIDR follows CIDR notation or Slash notation. The representation of CIDR notation is x.y.z.w /n the x.y.z.w is IP address and n is called mask or number of bits that are used in network id.

Properties of CIDR Block

The properties of CIDR block are as follows −

The IP addresses in a block are continuous.

The first address of a block should be exactly divisible by the number of addresses of a block.

The size of the Block should be power of 2.

Use of CIDR

Variable-length subnet masking is the foundation of CIDR (VLSM). It can now specify prefixes of any duration, making it much more powerful than the previous method.

Two collections of numbers make up CIDR IP addresses. The network address is written as a prefix, similar to how an IP address is written (e.g. 192.255.255.255).

The suffix, which means how many bits are in the whole address (e.g. /12), is the second component. A CIDR IP address will look anything like this when put together −

192.255.255.255/12

As part of the IP address, the network prefix is also defined. These changes are based on how many bits are needed. As an illustration, in the above example, the first 12 bits of the address are for the network, while the last 20 bits are for host addresses.

\

CIDR Notation

Using CIDR we can assign an IP address to host without using standard id address classes like Class A, B, and C.

In CIDR we simply tell how many bits are used for network id. The network id bits are provided after the '/' symbol. Like /10 means 10 bits are used for the network id part and remaining 32-10=22 bits are used for the host id part.

The **advantage** of using CIDR notation is that it reduces the number of entries in the routing table and also it manages the ip address space.

Disadvantages

The disadvantages of using CIDR Notation are as follows −

Using CIDR it is complex to determine the route. By using classful addresses, we are directly having separate tables for class A, Class B, Class C.

So we directly go to these tables by seeing the prefix of IP address. But by using CIDR, we don't have these tables separately. All entries are placed in a single table. So, it is difficult to find a route.

Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway. In DHCP, port number 67 is used for the server and 68 is used for the client.

DHCP allows a network administrator to supervise and distribute IP addresses from a central point and automatically sends a new Internet Protocol (IP) address when a computer is plugged into a different place in the network.

DHCP is an application layer protocol that provides −

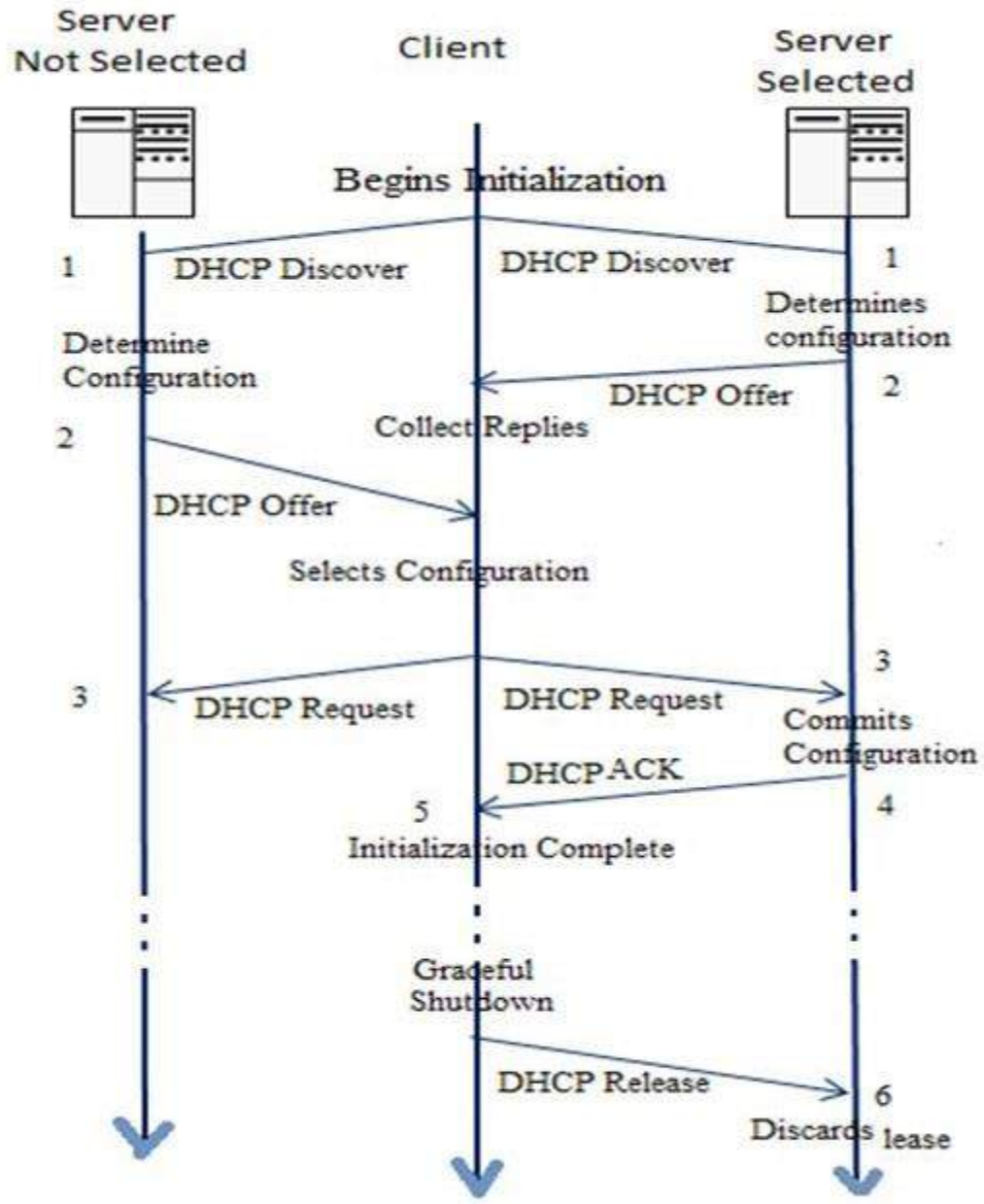
Subnet Mask

Router Address

IP Address

DHCP Client-Server Communication Diagram

In DHCP, the client and the server exchange DHCP messages to establish a connection.



DHCP Discover Message − Client Requests DHCP Information

It is the first message produced by a client in the communication process between the client and server with the target address 255.255.255.255 and the source address 0.0.0.0.

This message is produced by the client host to discover if there are any DHCP servers present in a network or not.

The message might contain other requests like subnet mask, domain name server, and domain name, etc.

The message is broadcast to all the devices in a network to find the DHCP server.

DHCP Offer Message − DHCP Server Offers Information to Client

The DHCP server will reply/respond to the host in this message, specifying the unleashed IP address and other TCP configuration information.

This message is broadcasted by the server.

If there are more than one DHCP servers present in the network, then the client host accepts the first DHCP OFFER message it receives.

Also, a server ID is specified in the packet to identify the server.

DHCP Request Message − Client Accepts DHCP Server Offer

The Client receives the DHCP offer message from the DHCP server that replied/responded to the DHCP discover message.

After receiving the offer message, the client will compare the offer that is requested, and then select the server it wants to use.

The client sends the DHCP Request message to accept the offer, showing which server is selected.

Then this message is broadcast to the entire network to let all the DHCP servers know which server was selected.

DHCP Acknowledgment Message − DHCP server acknowledges the client and leases the IP address.

If a server receives a DHCP Request message, the server marks the address as leased.

Servers that are not selected will return the offered addresses to their available pool.

Now, the selected server sends the client an acknowledgment (DHCP ASK), which contains additional configuration information.

The client may use the IP address and configuration parameters. It will use these settings till its lease expires or till the client sends a DHCP Release message to the server to end the lease.

DHCP Request, DHCP ACK Message − Client attempts to renew the lease

The client starts to renew a lease when half of the lease time has passed.

The client requests the renewal by sending a DHCP Request message to the server.

If the server accepts the request, it will send a DHC ACK message back to the client.

If the server does not respond to the request, the client might continue to use the IP address and configuration information until the lease expires.

As long as the lease is still active, the client and server do not need to go through the DHCP Discover and DHCP Request process.

When the lease has expired, the client must start over with the DHCP Discover process.

The client ends the lease − DHCPRELEASE

The client ends the lease by sending a DHCP Release message to the DHCP server.

The server will then return the client's IP address to the available address pool and cancel any remaining lease time.

Error Reporting

Internet Control Message Protocol:

ICMP protocol reports the error messages to the sender.

The operation of the internet is monitored by router closely when something happened in the network during packet processing , the event is reported to the sender by the ICMP

Each ICMP message type is carried encapsulated in an IP packet.

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

Destination unreachable: The message of "Destination Unreachable" is sent from receiver to the sender when destination cannot be reached, or packet is discarded when the destination is not reachable.

Source Quench: The purpose of the source quench message is congestion control. The message sent from the congested router to the source host to reduce the transmission rate. ICMP will take the IP of the discarded

packet and then add the source quench message to the IP datagram to inform the source host to reduce its transmission rate. The source host will reduce the transmission rate so that the router will be free from congestion.

Time Exceeded: Time Exceeded is also known as "Time-To-Live". It is a parameter that defines how long a packet should live before it would be discarded.

There are two ways when Time Exceeded message can be generated:

Sometimes packet discarded due to some bad routing implementation, and this causes the looping issue and network congestion. Due to the looping issue, the value of TTL keeps on decrementing, and when it reaches zero, the router discards the datagram. However, when the datagram is discarded by the router, the time exceeded message will be sent by the router to the source host.

When destination host does not receive all the fragments in a certain time limit, then the received fragments are also discarded, and the destination host sends time Exceeded message to the source host.

Parameter problems: When a router or host discovers any missing value in the IP datagram, the router discards the datagram, and the "parameter problem" message is sent back to the source host.

Redirection: Redirection message is generated when host consists of a small routing table. When the host consists of a limited number of entries due to which it sends the datagram to a wrong router. The router that receives a datagram will forward a datagram to a correct router and also sends the "Redirection message" to the host to update its routing table.

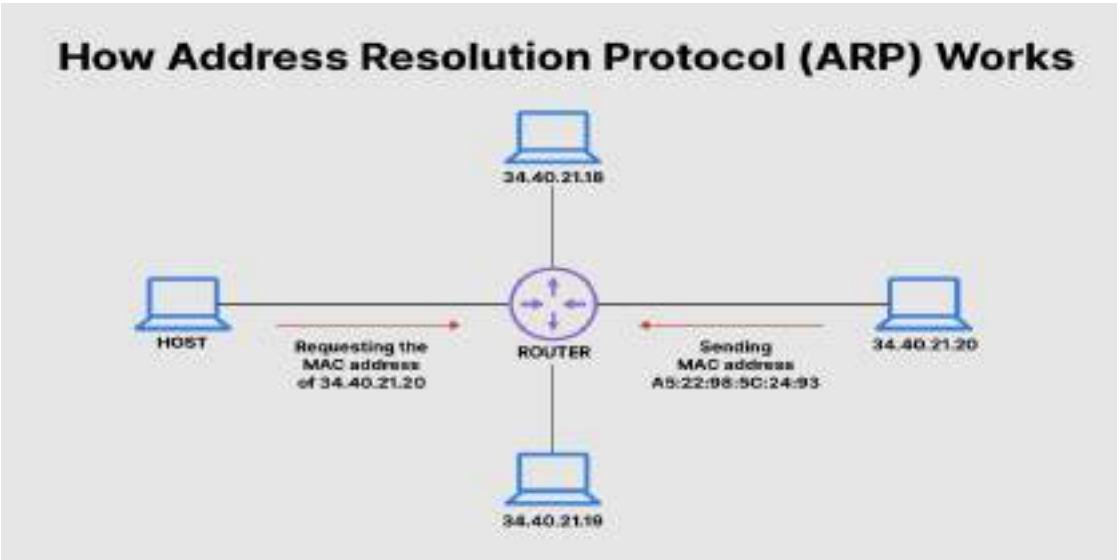
Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) is a protocol or procedure that connects an ever-changing Internet Protocol (IP) address to a fixed physical machine address, also known as a media access control (MAC) address, in a local-area network (LAN).

This mapping procedure is important because the lengths of the IP and MAC addresses differ, and a translation is needed so that the systems can recognize one another. The most used IP today is IP version 4 (IPv4). An IP address is 32 bits long. However, MAC addresses are 48 bits long. ARP translates the 32-bit address to 48 and vice versa.

There is a networking model known as the Open Systems Interconnection (OSI) model. First developed in the late 1970s, the **OSI model** uses layers to give IT teams a visualization of what is going on with a particular networking system. This can be helpful in determining which layer affects which application, device, or software installed on the network, and further, which IT or engineering professional is responsible for managing that layer.

The MAC address is also known as the data link layer, which establishes and terminates a connection between two physically connected devices so that data transfer can take place. The IP address is also referred to as the network layer or the layer responsible for forwarding packets of data through different routers. ARP works between these layers.



What Does ARP Do And How Does It Work?

When a new computer joins a local area network (LAN), it will receive a unique IP address to use for identification and communication.

Packets of data arrive at a gateway, destined for a particular host machine. The gateway, or the piece of hardware on a network that allows data to flow from one network to another, asks the ARP program to find a MAC address that matches the IP address. The ARP cache keeps a list of each IP address and its matching MAC address. The ARP cache is dynamic, but users on a network can also configure a static **ARP table** containing IP addresses and MAC addresses.

ARP caches are kept on all operating systems in an IPv4 **Ethernet network**. Every time a device requests a MAC address to send data to another device connected to the LAN, the device verifies its ARP cache to see if the IP-to-MAC-address connection has already been completed. If it exists, then a new request is unnecessary. However, if the translation has not yet been carried out, then the request for network addresses is sent, and ARP is performed.

An ARP cache size is limited by design, and addresses tend to stay in the **cache** for only a few minutes. It is purged regularly to free up space. This design is also intended for privacy and security to prevent IP addresses from being stolen or spoofed by cyberattackers. While MAC addresses are fixed, IP addresses are constantly updated.

In the purging process, unutilized addresses are deleted; so is any data related to unsuccessful attempts to communicate with computers not connected to the network or that are not even powered on.

What Is Address Resolution Protocol's Relationship With DHCP And DNS? How Do They Differ?

ARP is the process of connecting a dynamic IP address to a physical machine's MAC address. As such, it is important to have a look at a few technologies related to IP.

As mentioned previously, IP addresses, by design, are changed constantly for the simple reason that doing so gives users security and privacy. However changes on IP addresses should not be completely random. There should be rules that allocate an IP address from a defined range of numbers available in a specific network. This helps prevent issues, such as two computers receiving the same IP address. The rules are known as DHCP or **Dynamic Host Configuration Protocol**.

IP addresses as identities for computers are important because they are needed to perform an internet search. When users search for a domain name or Uniform Resource Locator (URL), they use an alphabetical name. Computers, on the other hand, use the numerical IP address to associate the domain name with a server. To connect the two, a **Domain Name System (DNS) server** is used to translate an IP address from a confusing string of numbers into a more readable, easily understandable domain name, and vice versa.

What Are The Types Of ARP?

There are different versions and use cases of ARP. Let us take a look at a few.

Proxy ARP

Proxy ARP is a technique by which a proxy device on a given network answers the ARP request for an IP address that is not on that network. The proxy is aware of the location of the traffic's destination and offers its own MAC address as the destination.

Gratuitous ARP

Gratuitous ARP is almost like an administrative procedure, carried out as a way for a host on a network to simply announce or update its IP-to-MAC address. Gratuitous ARP is not prompted by an ARP request to translate an IP address to a MAC address.

Reverse ARP (RARP)

Host machines that do not know their own IP address can use the Reverse Address Resolution Protocol (RARP) for discovery.

Inverse ARP (IARP)

Whereas ARP uses an IP address to find a MAC address, IARP uses a MAC address to find an IP address

UNIT-4

TransportLayer

Thetransportlayerisa 4th layerofOSIModel.

The main role of the transport layer is to provide the communication servicesdirectlytotheapplicationprocessesrunningondifferenthosts.

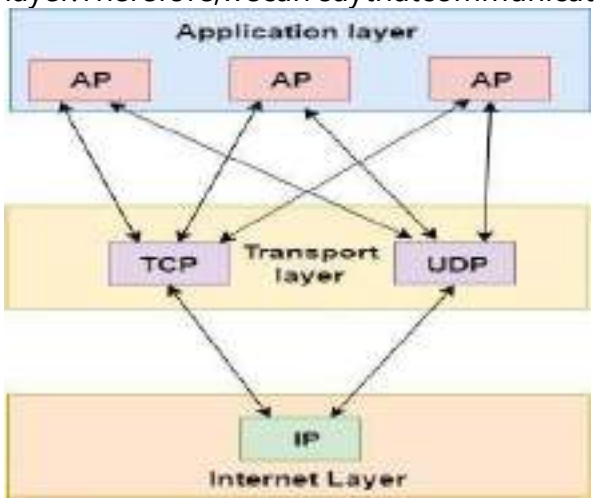
Thetransportlayerprovidesalogicalcommunicationbetweenapplicationprocessesrunning ondifferenthosts.Although the applicationprocessesondifferent hosts are not physically connected, application processes use the logicalcommunication provided by the transport layer to send the messages to eachother.

The transport layer protocols are implemented in the end systems but not in thenetworkrouters.

Acomputernetworkprovidesmorethanoneprotocoltothenetworkapplications. For example, TCP and UDP are two transport layer protocols thatprovideadifferentset ofservices to thenetworklayer.

All transport layer protocols provide multiplexing/demultiplexing service. It alsoprovides other services such as reliable data transfer, bandwidth guarantees, anddelayguarantees.

Each of the applications in the application layer has the ability to send a messageby using TCP or UDP. The application communicates by using either of these twoprotocols. Both TCP and UDP will then communicate with the internet protocol inthe internet layer. The applications can read and write to the transport layer.Therefore,wecan saythatcommunication isatwo-wayprocess.

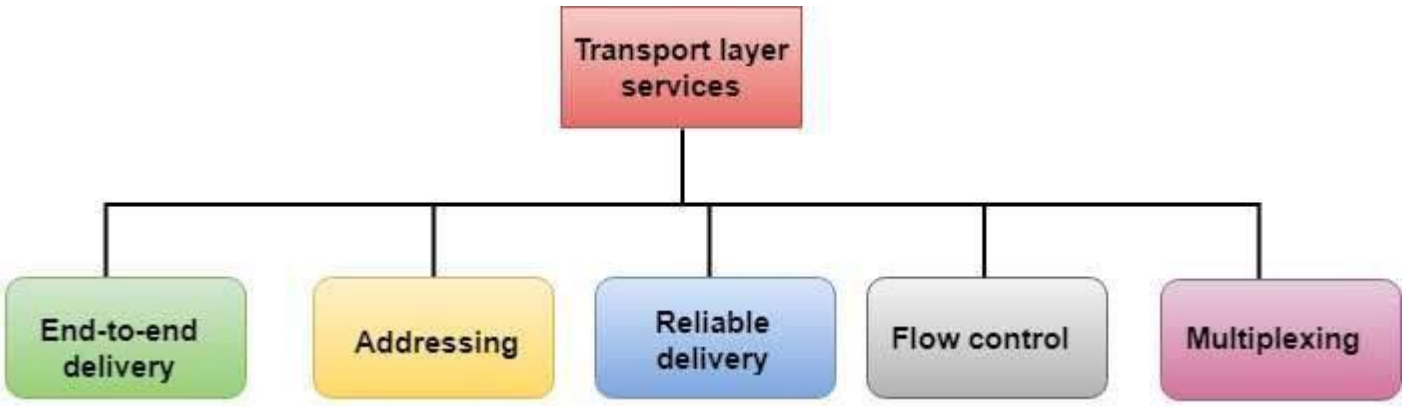


ServicesprovidedbytheTransportLayer

The services provided by the transport layer are similar to those of the data link layer.The data link layer provides the services within a single network while the transport layerprovides the services across an internetwork made up of many networks. The data linklayer controlsthephysicallayerwhilethetransportlayer controlsallthelowerlayers.

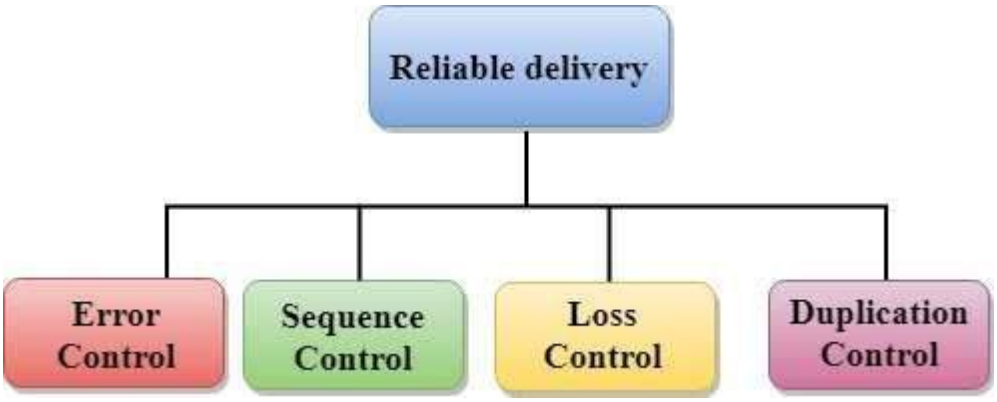
The services provided by the transport layer protocols can be divided into fivecategories:

End-to-enddelivery
Addressing
Reliabledelivery
Flowcontrol
Multiplexing



End-to-enddelivery:
The transport layer transmits the entire message to the destination. Therefore, it ensurestheend-to-enddeliveryofanentiremessagefromasource tothedestination.
Reliabledelivery:
Thetransportlayerprovidesreliabilityservicesbyretransmittingthelostanddamagedpackets. Thereliabledeliveryhasfouraspects:

Errorcontrol
Sequencecontrol
Losscontrol
Duplicationcontrol



ErrorControl
The primary role of reliability is **Error Control**. In reality, no transmission will be100 percent error-free delivery. Therefore, transport layer protocols are designedtoprovideerror-free transmission.
The data link layer also provides the error handling mechanism, but it ensuresonly node-to-node error-freedelivery.However,node-to-node reliabilitydoesnotensure theend-to-endreliability.
The data link layer checks for the error between each network. If an error isintroduced insideoneof the routers,thenthis error willnot becaught bythedata link layer.It only detectsthoseerrors that have beenintroduced betweenthe beginning and end of the link. Therefore, the transport layer performs thecheckingfortheerrorsend-to-endtoensurethatthepackethasarrivedcorrectly.

SequenceControl

The second aspect of the reliability is sequence control which is implemented atthetransportlayer.
On the sending end, the transport layer is responsible for ensuring that thepackets received from the upper layers can be used by the lower layers. On thereceiving end, it ensures that the various segments of a transmission can becorrectlyreassembled.

LossControl

Loss Control is a third aspect of reliability. The transport layer ensures that all thefragments of a transmission arrive at the destination, not some of them. On the sendingend, all the fragments of transmission are given sequence numbers by a transport layer.These sequence numbers allow the receivers? transport layer to identify the missingsegment.

DuplicationControl

Duplication Control is the fourth aspect of reliability. The transport layer guarantees thatno duplicate data arrive at the destination. Sequence numbers are used to identify thelostpackets; similarly,it allowsthereceiver toidentifyanddiscardduplicatesegments.

FlowControl

Flow control isused to prevent the sender from overwhelming the receiver.If thereceiver is overloaded with too much data, then the receiver discards the packets andasking for the retransmission of packets. This increases network congestion and thus,reducing the system performance. The transport layer is responsible for flow control. Ituses the sliding window protocol that makes the data transmission more efficient as wellasitcontrolstheflowofdatasothatthereceiverdoesnotbecomeoverwhelmed.Slidingwindowprotocolisbyteorientedratherthanframeoriented.

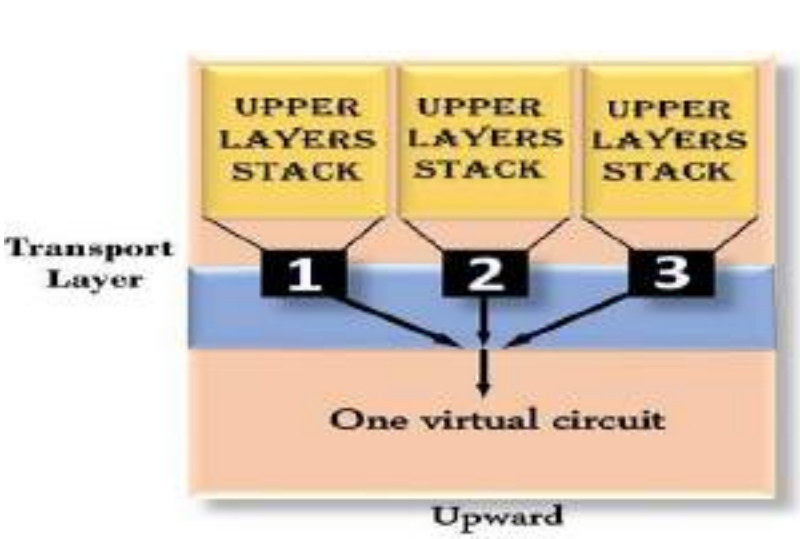
Multiplexing

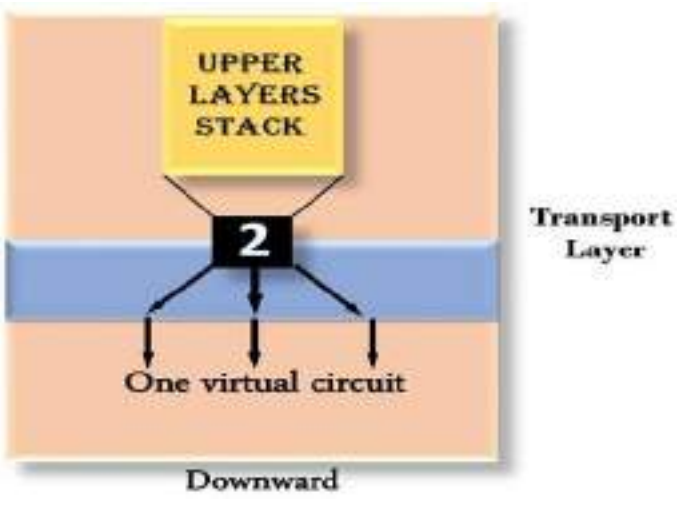
Thetransportlayerusesthemultiplexingtoimprovetransmissionefficiency.

Multiplexingcanoccurintwoways:

Upwardmultiplexing: Upwardmultiplexingmeansmultipletransportlayerconnections use the same network connection. To make more cost-effective, thetransport layer sends several transmissions bound for the same destination alongthesame path;this isachievedthrough upwardmultiplexing.

Downward multiplexing: Downward multiplexing means one transport layerconnectionusesthemultiplenetworkconnections.Downwardmultiplexing allows the transport layer to split a connection among several paths to improvethe throughput. This type of



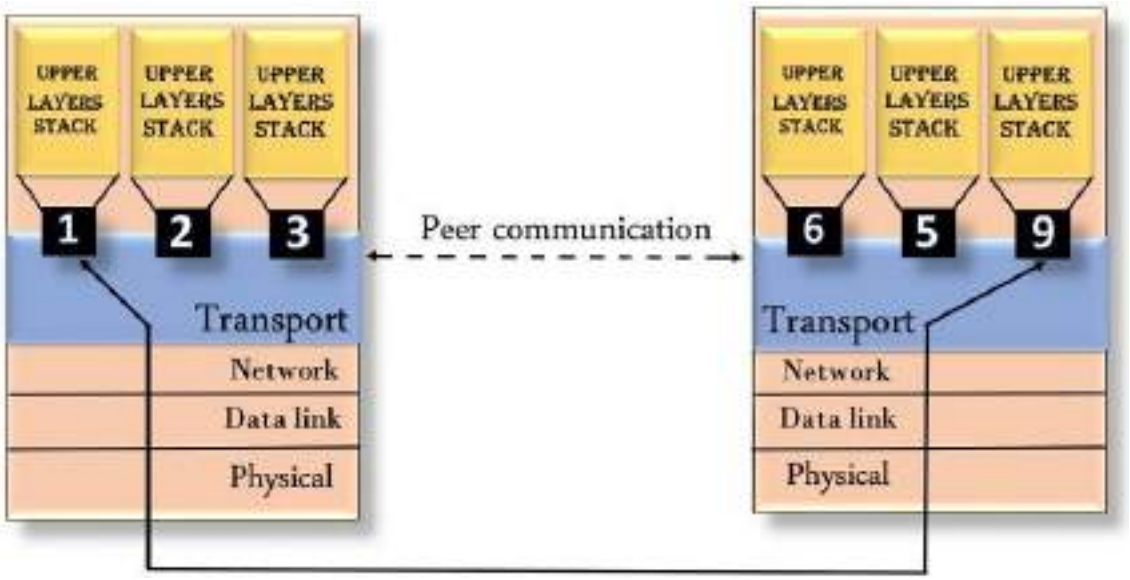


multiplexing is used when networks have a low orslowcapacity.

Addressing

According to the layered model, the transport layer interacts with the functions ofthe session layer. Many protocols combine session, presentation, and applicationlayer protocols into a single layer known as the application layer. In these cases,delivery to the session layer means the delivery to the application layer. Datagenerated by an application on one machine must be transmitted to the correctapplicationonanothermachine.Inthiscase,addressingisprovidedbythetransportlayer.
The transport layer provides the user address which is specified as a station orport. The port variable represents a particular TS user of a specified station knownas a Transport Service access point (TSAP). Each station has only one transportentity.

Thetransportlayerprotocolsneedtoknowwhichupper-layerprotocolsarecommunicating.



ElementsofTransportProtocol:
To establish a reliable service between two machines on a network, transport protocols areimplemented, which somehow resembles the data link protocols implemented at layer 2. Themajor difference lies in the fact that the data link layer uses a physical channel between tworouterswhilethetransportlayeruses asubnet. Followingaretheissuesforimplementingtransportprotocols−

Typesof Service

The transport layer also determines the type of service provided to the users from the sessionlayer. An error-free point-to-point communication to deliver messages in the order in which theyweretransmitted isoneofthekey functionsofthetransport layer.

ErrorControl

Error detection and error recovery are an integral part of reliable service, and therefore they arenecessary to perform error control mechanisms on an end-to-end basis. To control errors fromlost or duplicate segments, the transport layer enables unique segment sequence numbers to thedifferent packets of the message, creating virtual circuits, allowing only one virtual circuit persession.

FlowControl

The underlying rule of flow control is to maintain a synergy between a fast process and a slowprocess.Thetransportlayerenablesafastprocesstokeeppacewithaslowone.Acknowledgements are sent back to manage end-to-end flow control. Go back N algorithms areusedto request retransmission of packets starting with packet number N.Selective Repeat isusedto request specificpackets to beretransmitted.

ConnectionEstablishment/Release

The transport layer creates and releases the connection across the network. This includes anaming mechanism so that a process on one machine can indicate with whom it wishes tocommunicate. The transport layer enables us to establish and delete connections across thenetworkto multiplexseveral messagestreams onto onecommunication channel.

Multiplexing/Demultiplexing

The transport layer establishes a separate network connection for each transport connectionrequired by the session layer. To improve throughput, the transport layer establishes multiplenetwork connections. When the issue of throughput is not important, it multiplexes severaltransport connections onto the same network connection, thus reducing the cost of establishingandmaintainingthenetwork connections.
When several connections are multiplexed, they call for demultiplexing at the receiving end. Inthe case of the transport layer, the communication takes place only between two processes andnot between two machines. Hence, communication at the transport layer is also known as peer-to-peer orprocess-to-processcommunication.

Fragmentationandre-assembly

When the transport layer receives a large message from the session layer, it breaks the messageintosmallerunitsdependinguponthe requirement.Thisprocessiscalledfragmentation.Thereafter, it is passed to the network layer. Conversely, when the transport layer acts as thereceivingprocess,itreordersthe piecesofa messagebefore reassemblingthem intoa message.

Addressing

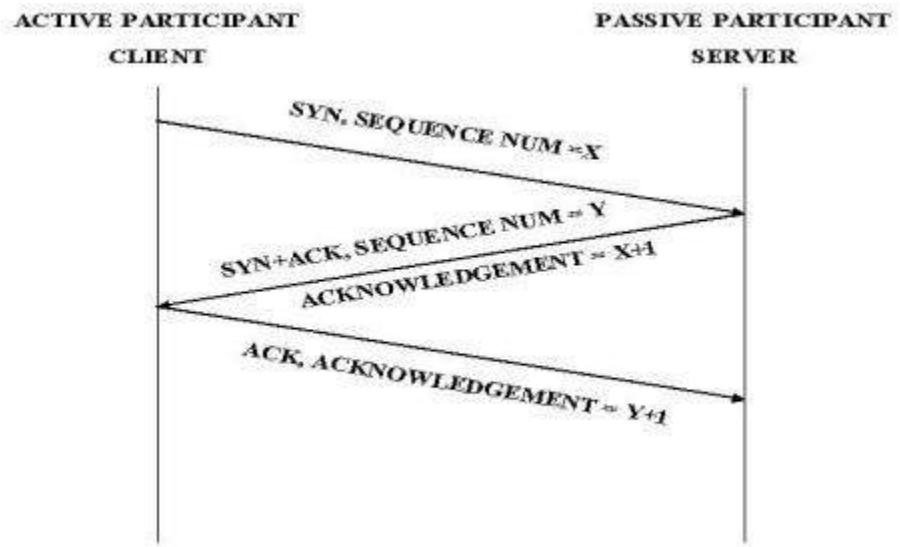
Transport Layer deals with addressing or labelling a frame. It also differentiates between aconnectionandatransaction.Connectionidentifiersareportsorsocketsthatlabeleachframe,sothereceivingdevice knowswhichprocessithasbeensentfrom.Thishelpsinkeepingtrackof multiple-message conversations. Ports or sockets address multiple conservations in the samelocation.

CONNECTIONMANAGEMENT:

A TCP connection begins with a client doing an active open to a server.Assuming that the server had earlier done a passive open, the two sides engage inan exchange of messages to establish the connection. Only after this connectionestablishment phase is over do the two sides begin sending data. Likewise, as soonas a participant is done sending data, it closes one direction of the connection,whichcausesTCPtoinitiatearoundofconnection terminationmessages.
Connection setup is an asymmetric activity (one side does a passive open and theother side does an active open) connection teardown is symmetric (each side has toclose theconnection independently). Therefore it is possible for one side to havedone a close, meaning that it can no longer send data but for the other side to keeptheotherhalfofthebidirectionalconnectionopensandtocontinuesendingdata.

THREEWAYHANDSHAKES:

ThealgorithmusedbyTCPtoestablishandterminateaconnectioniscalleda *three way handshake.*The client (the active participant) sends a segment tothe server(the passive participation) stating the initial sequence number it plans touse(*flag=SYN,SequenceNum=x)*.
The server then responds with a single segment that both acknowledges the client"ssequencenumber(Flags=ACK,Ack=x+1)andstatesitsownbeginningsequencenumber (*Flags=SYN,SequenceNum=y)*



That is, both the SYN and ACK bits are set in the Flags field of this secondmessage. Finally, the client responds with a third segment that acknowledges theserver"ssequencenumberFlags=ACK, Ack=y+1).
The "three-way handshake" is the procedure used to establish a connection. Thisprocedure normally is initiated by one TCP and responded to by another TCP. Theprocedure also works if two TCP simultaneously initiate the procedure. Whensimultaneous attempt occurs,each TCPreceives a"SYN" segment which carriesno acknowledgment after it has sent a "SYN". Of course, the arrival of an oldduplicate "SYN" segment can potentially make it appear, to the recipient, that asimultaneous connection initiation is in progress. Proper use of "reset" segmentscandisambiguatethesecases.
The three-way handshake reduces the possibility of false connections. It is theimplementationofatrade-offbetweenmemoryandmessagestoprovideinformationforthischecking.

The simplestthree-way handshakeisshowninfigurebelow.Thefigures shouldbeinterpretedinthefollowingway.Eachlineisnumberedforreferencepurposes.Rightarrows(-->)indicatedepartureofaTCPsegmentfromTCPAtoTCPB,orarrival of a segment at B from A. Left arrows (<--), indicate the reverse. Ellipsis (...)indicatesasegmentwhichisstillinthenetwork(delayed).TCPstatesrepresentthe state AFTER the departure or arrival of the segment (whose contents are showninthecenterofeachline).Segmentcontentsareshowninabbreviatedform,withsequencenumber,control

flags,andACKfield.Otherfieldssuchaswindow,addresses,lengths,and texthavebeenleftoutin theinterest ofclarity.

TCP A TCP B

1. CLOSED LISTEN
2. SYN-SENT --> <SEQ=100><CTL=SYN> --> SYN-RECEIVED

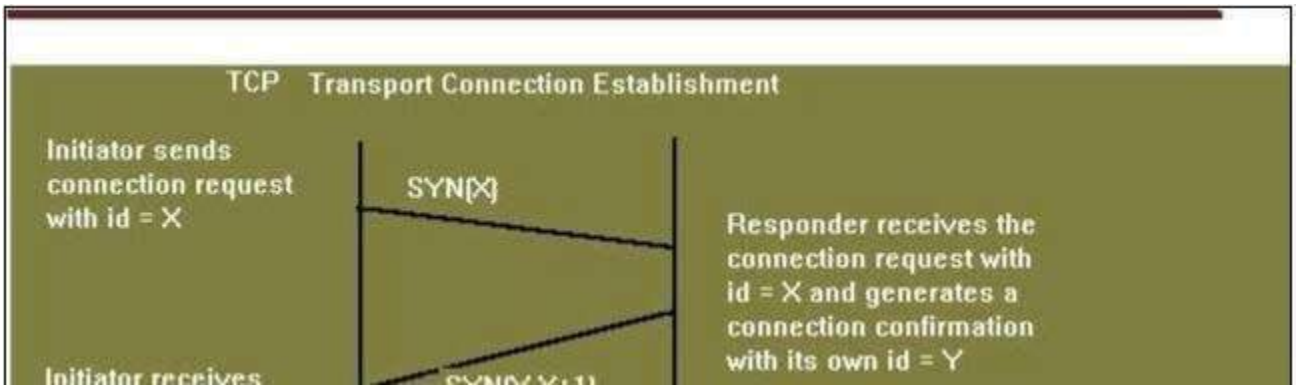
3. ESTABLISHED <-- <SEQ=300><ACK=101><CTL=SYN,ACK> <-- SYN-RECEIVED

4. ESTABLISHED --> <SEQ=101><ACK=301><CTL=ACK> --> ESTABLISHED

5. ESTABLISHED --> <SEQ=101><ACK=301><CTL=ACK><DATA> --> ESTABLISHED

Basic3-WayHandshakeforConnectionSynchronisation
In line 2 of above figure, TCP A begins by sending a SYN segment indicating thatit will use sequence numbers starting with sequence number 100. In line 3, TCP Bsends a SYN and acknowledges the SYN it received from TCP A. Note that theacknowledgment field indicates TCP B is now expecting to hear sequence 101,acknowledgingtheSYNwhichoccupiedsequence100.

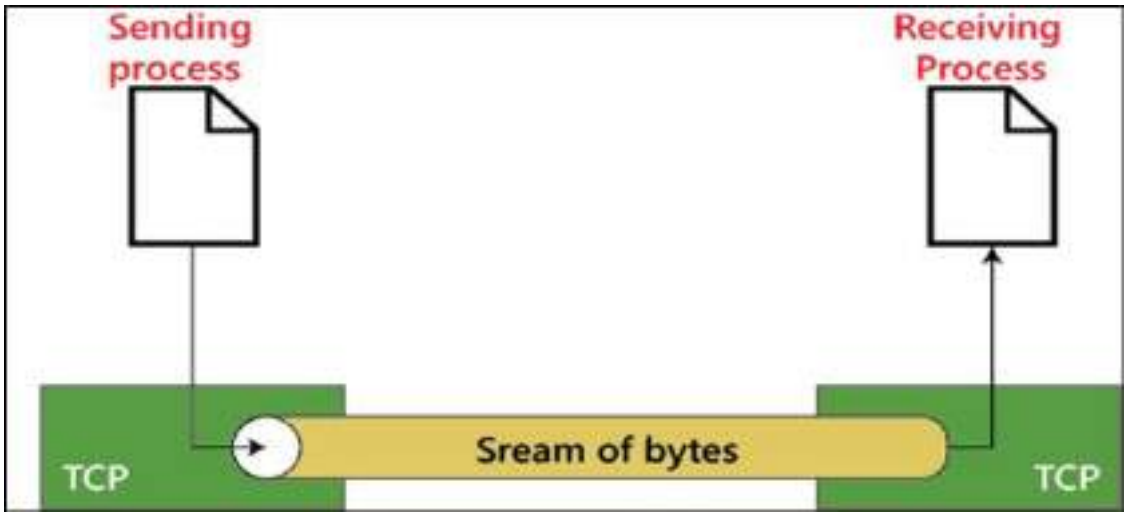
At line 4, TCP A responds with an empty segment containing an ACK for TCP B'sSYN; and in line 5, TCP A sends some data. Note that the sequence number of thesegment in line 5 is the same as in line 4 because the ACK does not occupysequencenumberspace(ifit did, wewouldwind upACKingACK's!).



Transmission ControlProtocol:

TCPstands forTransmissionControlProtocol. It wasintroducedin1974. It is a connection-oriented and reliable protocol. It establishes a connection between the source and destination device before starting the communication. It detects whether the destination device has received the data sent from the source device or not. If the received data is not in the proper format, it sends the data again. TCP is highly reliable because

it uses a handshake and traffic control mechanism. In the TCP protocol, the receiver receives the data in the same sequence in which the sender sends it. We use the TCP protocol services in our daily life, such as HTTP, HTTPS, Telnet, FTP, SMTP, etc.



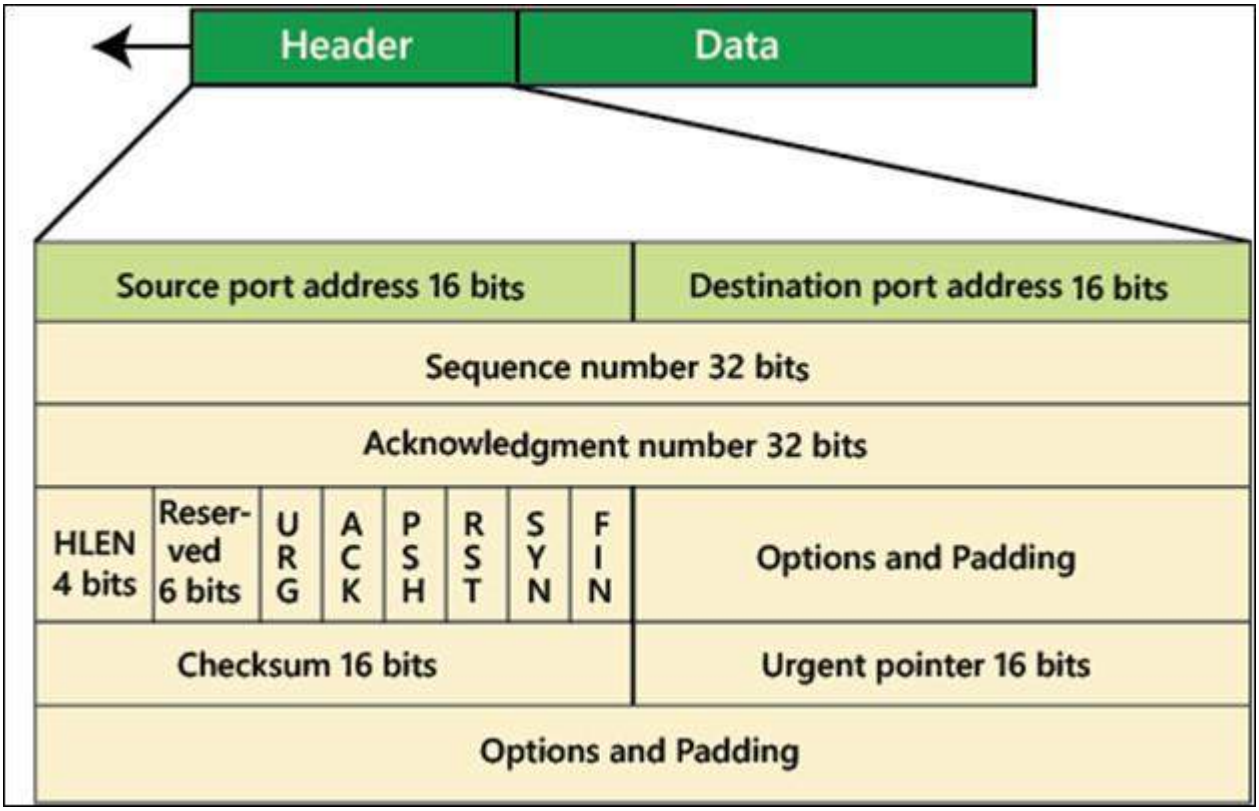
Importanceof TCP

TheTCPprotocolisusedtotransferdatawhere,youneed accuracyandreliabilityrather thanthespeed.

Both TCP and UDP can check for errors, but only TCP can fix the error because it can controlthe traffic and flow of data.

TCPSegmentFormat

In the TCP, the data packet is called a segment.The size of the headerin the segmentis 20 to 60 bytes.The



segment formatof TCP is shown below

Thesegmentconsistsofa20-to60-byteheader,followedbydatafromtheapplication program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

Sourceportaddress.Thisisa16-bit fieldthat definestheport numberoftheapplication program in the host that is sending the segment. This serves the same purpose as the source port address in the UDP header.
Destination port address. This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. This serves the same purpose as the destination port address in the UDP header.
Sequence number. This 32-bit field defines the number assigned to the first byte ofdata contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence comprises the first byte in the segment.
Acknowledgment number. This32-bit field definesthe byte number that the receiver of the segment is expecting to receive from the other party.
Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field can be between 5 (5 x 4 =20) and 15 (15 x 4 =60).
Reserved.This isa6-bitfieldreservedforfutureuse.
Control.This fielddefines6different controlbits orflagsasshowninFigure4.17. One or more of these bits can be set at a time.

Figure4.17Control Field
These bits enable flow control, connection establishment and termination, connectionabortion, and the mode ofdata transfer in TCP. A briefdescription ofeach bit is shown in Table 4.3
Table4.3Descriptionofflagsinthecontrol field

Flag	Description
URG	Thevalueoftheurgentpointerfieldis valid.
ACK	Thevalueoftheacknowledgefieldis valid.
PSH	Pushthedata.
RST	Resettheconnection.
SYN	Synchronizesequencenumbersduringconnection
FIN	Terminatetheconnection.

Window size. This field defines the size of the window, in bytes, thatthe other party must maintain. Note that the length of this field is 16 bits, which means thatthe maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.
Checksum. This 16-bit field contains the checksum. The inclusion of the checksum for TCP is mandatory. For

the TCP pseudoheader, the value for the protocol field is 6.
Urgent pointer. This l6-bit field, which is valid, only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.
Options. Therecanbe upto 40bytesofoptionalinformationintheTCPheader.

A TCPConnection:
TCP is connection-oriented. Aconnection-oriented transport protocolestablishes a virtualpath between the source and destination. All the segments belonging to a message are then sent over this virtual path. Using a single virtual pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

ConnectionEstablishment
TCP transmits data in full-duplex mode. When two TCPs in two machines are connected,they are able to send segments to each other simultaneously. This implies that each party must initialize communication and get approval from the other partybefore any data are transferred.

Three-WayHandshaking*:*
The connection establishment in TCP is called threeway handshaking. In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol.
The process starts with the server. The server programtells its TCP that it is ready to accept a connection. This iscalled a request fora passive open. Althoughthe server TCP isreadytoaccept any connection from anymachine in the world, it cannot make the connection itself.
The client program issues a request for an active open. A client that wishes to connect to an openserver tells its TCP that it needs to be connectedto that particular server. TCP can now start the three-way handshaking process as shown in Figure 4.18.
Toshow the process, we use two time lines: one at each site. Each segment has values forall itsheader fieldsandperhaps forsomeofitsoptionfields,too.However,weshowonlythe fewfields necessarytounderstandeachphase.Weshowthesequencenumber, the acknowledgment number, the control flags (only those that are set), and the window size, if ot empty. The three steps in this phase are as follows.

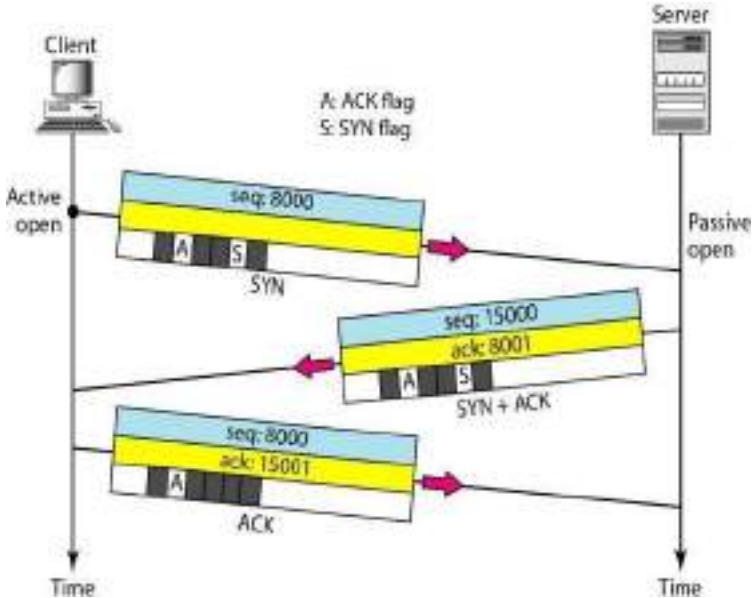


Figure. Connectionestablishmentusingthree-wayhandshaking

The client sends the first segment, a SYN segment, in which onlythe SYN flag is set. This segment is for synchronization of sequence numbers. It consumes one sequence number. Whenthe datatransfer start,the sequence number is incremented by1. We cansaythat the SYN segment carries no realdata, but we canthink of it as containing 1 imaginary byte.
ASYNsegment cannotcarrydata,butitconsumesonesequencenumber.

Theserver sendsthesecondsegment, aSYN+ACKsegment, with2 flag bitsset:SYN and ACK. This segment hasa dualpurpose. It isa SYN segment for communication in the other direction and serves as the acknowledgment for the SYN segment. It consumes one sequence number.
ASYN+ACKsegment cannotcarrydata,butdoesconsumeonesequence number.

The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. Note that the sequence number in this segment is the same as the one in the SYN segment; the ACK segment does not consume any sequence numbers.
AnACKsegment,ifcarryingnodata,consumesnosequence number.

SimultaneousOpen
A rare situation, called a simultaneous open, may occur when both processes issue an active open. In this case, both TCPs transmit a SYN + ACK segment to each other, and one single connection is established between them.

SYNFlooding Attack
The connection establishment procedure in TCP is susceptible to a serious securityproblem calledtheSYN floodingattack.Thishappenswhena maliciousattackersends a large numberofSYN segments to a server, pretending that each of them is corning from a different client by faking the source IP addresses in the datagrams.
DataTransfer
After connection is established, bidirectional data transfer can take place. The client andserver can both send data and acknowledgments. The acknowledgment is piggybacked with the data. Figure 4.19 shows an example.
Inthisexample,afterconnectionisestablished(not showninthe figure),theclient sends 2000 bytes of data in two segments. The server then sends 2000 bytes in one segment.
The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because thereare no more data to be sent. Note the values of the sequence and acknowledgment numbers. The data segments sent by the client have the PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received. The segment fromthe server, on the other hand, does not set the push flag. Most TCP implementations have the option to set or not set this flag.

PushingData
The sending TCP uses a buffer to store the stream of data coming from the sendingapplication program. The

sending TCP can select the segment size. The receiving TCP also buffers the data when they arrive and delivers them to the application program when theapplication program is ready or when it is convenient for the receiving TCP. This type of flexibility increases theefficiency of TCP. However, on occasion the application program has noneed for this flexibility.

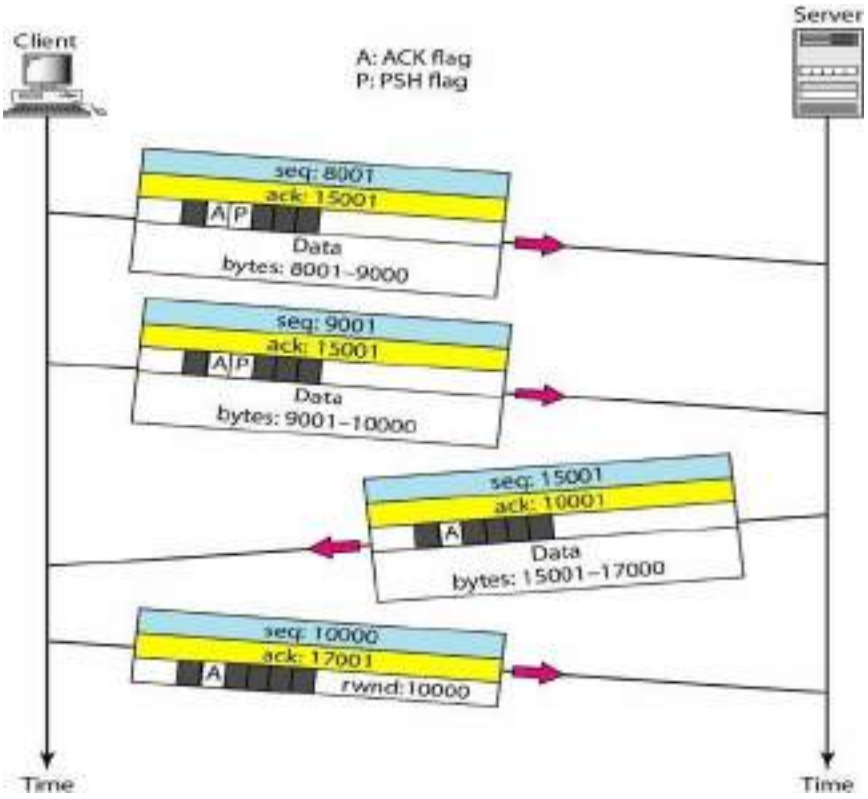


Figure Data transfer

TCP can handle such a situation. The application program at the sending site can requesta push operation. This means that the sending TCP must not wait for the window to be filled. It must createasegment andsend it immediately.ThesendingTCP must also setthepushbit (PSH)to let the receiving TCP know that the segmentincludes data that must be delivered to thereceiving applicationprogramassoonaspossibleandnottowaitformoredatatocome.
Although the push operation can be requested by the application program, most current implementations ignore such requests. TCP can choose whether or not to use this feature.

Urgent Data

TCP is a stream-oriented protocol. This means thatthe dataare presented fromthe application program to TCP as a stream of bytes. Each byte of data has a position in the stream. However, on occasion an application program needs to send urgent bytes. This means that the sending application program wants a piece of data to be read out of order by the receiving application program. As an example, suppose that the sending application program is sending data to be processed by the receiving application program. When the result of processing comes back, the sending application program finds that everything is wrong. It wants to abort the process, but it has already sent a huge amount of data. If it issues an abort command, these two characters will be stored at the end of the receiving TCP buffer. It will be delivered to the receiving application program after all the data have been processed.

ConnectionTermination

Any of the two parties involved in exchanging data (client or server) can close theconnection, although it is usually initiated by the client. Most implementations today allow two Options for connectiontermination:three-wayhandshakingandfour-wayhandshakingwithahalf-close option.

Three-WayHandshaking

Most implementations today allow three-way handshaking for connection termination as shown in Figure 4.20.

Ina normalsituation, the client TCP, afterreceiving a close command fromthe client process, sends the first segment, a FIN segment in which the FIN flag is set. Note that a FIN segmentcan include thelast chunk of data sent bythe client, or it can be just a control segment as shown in Figure 4.20. If it is onlya controlsegment, it consumes onlyone sequence number.

TheFINsegment consumesonesequencenumberifitdoesnotcarry data.

The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN +ACK segment, to confirmthe receipt ofthe FIN segment from the client and at the same time to announce the closing of the connection in the other direction. This segment can also contain the last chunkof data from the server. If it does not carry data, it consumes only one sequence number.

TheFIN+ACKsegmentconsumesonesequencenumberifitdoesnotcarry data.

The client TCP sends the last segment, an ACK segment, to confirmthe receipt of the FIN segment fromthe TCP server.Thissegment containsthe acknowledgment number, which is 1 plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

Half-Close

In TCP, one end can stop sending data while still receiving data. This is called a half- close. Althougheither end can issue a half-close, it is normally initiated bythe client.It canoccurwhenthe server needs all the data before processing can begin. A good example is sorting. When the client sends data to the server to be sorted, the server needs to receive all the data before sorting can start. Thismeans the client, after sending all the data, can close the connectionintheoutbounddirection.

However,theinbounddirectionmustremainopentoreceivethe

sorted data. The server, after receiving the data, still needs time for sorting; itsoutbounddirection must remain open.

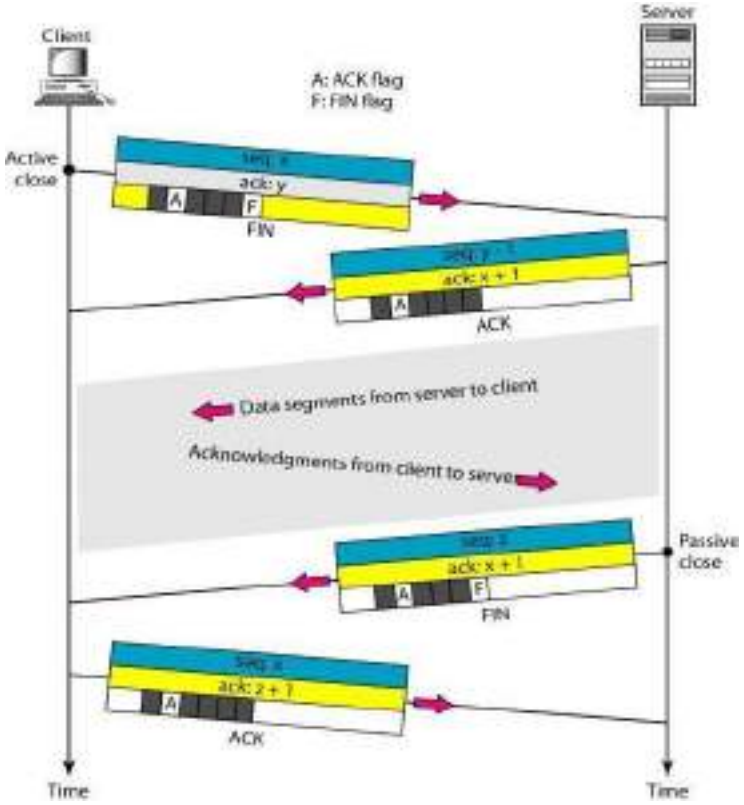


Figure above shows an example of a half-close. The client half-closes the connection bysending a FIN segment. The server accepts the half-close by sending the ACK segment. The data transfer from the client to the server stops. The server, however, can still send data. When the server has sent allthe processed data, it sends a FIN segment, which isacknowledged by an ACK fromthe client.

After half-closing of the connection, data can travel from the server to the client and acknowledgmentscantravelfromtheclient totheserver. Theclientcannotsendanymoredatatothe server.Notethesequencenumberswehaveused. Thesecondsegment(ACK)consumes no sequence number. Although the client has received sequence number y - 1 and is expecting y, the server sequence number is still y - 1. When the connection finally closes, the sequence number of the last ACK segment is still x, because no sequence numbers are consumed during data transfer in that direction.

Flow Control in TCP

TCPusesaslidingwindowto handle flowcontrol. Theslidingwindowprotocolusedby TCP, however, is something between the Go-Back-N and Selective Repeat sliding window. The sliding window protocol in TCP looks like the Go-Back-N protocol because it does not use NAKs; it looks like Selective Repeat because the receiver holds the out-of-order segments until the missing ones arrive.

The window is opened, closed, or shrunk. These three activities, as we will see, are in the control of the receiver (and depend on congestion in the network), not the sender. The sender must obey the commands of the receiver in this matter.

Opening a window means moving the right wall to the right. This allows more new bytesin the buffer that are eligible for sending. Closing the window means moving the left wall to the right.This meansthat

somebytes have beenacknowledgedandthe sender need not worryabout them anymore. Shrinking the window means moving the right wallto the left. This is strongly discouraged and not allowed in some implementations because it means revoking the eligibility ofsome bytes for sending. This is a problem if the sender has already sent these bytes. Note that the left wall cannot move to the left because this would revoke some of the previously sent acknowledgments.

A sliding window is used to make transmissionmore efficient as well as to control the flow of data so that the destination does not become overwhelmed with data. TCP sliding windows are byte-oriented.

The size ofthe window at one end is determined bythe lesser oftwo values: receiver window (rwnd) or congestion window (cwnd). The receiver window is the value advertised by the oppositeend in a segment containing acknowledgment. It is the number ofbytes the other end can accept before its buffer overflows and data are discarded. The congestion window is a value determined by the network to avoid congestion.

Figure SlidingWindow

Some points about TCP sliding windows:

Thesizeofthewindowisthelesser of*rwnd*and*cwnd.*

Thesourcedoesnot have tosendafullwindow'sworthofdata.

Thewindowcanbeopenedorclosedbythereceiver,butshouldnotbeshrunk.

Thedestinationcansendanacknowledgment atanytimeas long as it doesnotresult ina shrinking window.

The receivercan temporarily shutdown the window; the sender,however,can always send a segment of 1 byte after the window is shut down.

ErrorControl in TCP

TCPisareliabletransport layerprotocol.Thismeans that anapplicationprogramthat delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without anypart lost or duplicated.

TCP provides reliability using error control. Error control includes mechanisms for detecting corrupted segments, lost segments, out-of-order segments, and duplicated segments. Error controlalso includes a mechanismfor correcting errors after theyare detected. Error detectionand correction in TCP is achieved throughthe use of three simple tools: checksum, acknowledgment, and time-out.

Checksum

Each segment includes a checksum field which is used to check for a corrupted segment. If the segment is corrupted, it is discarded bythe destinationTCP and is considered as lost. TCP uses a 16- bitchecksum thatis mandatory in every segment. The 16-bitchecksum is considered

inadequate for the new transport layer, SCTP. However, it cannot be changed for TCPbecause this would involve reconfiguration of the entire header format.

Acknowledgment

TCP uses acknowledgments to confirm the receipt of data segments. Control segments that carry no data but consume a sequence number are also acknowledged. ACK segments are never acknowledged.

ACKsegmentsdonotconsumesequencenumbersandarenot acknowledged.

Retransmission

The heart of the error control mechanism is the retransmission of segments. When a segment is corrupted, lost, or delayed, it is retransmitted. In modern implementations, a segment is retransmitted ontwooccasions: whena retransmissiontimer expires or when the sender receives three duplicate ACKs.

In modern implementations, a retransmission occurs if the retransmission timer expiresor three duplicate ACK segments have arrived.

Note that no retransmission occurs for segments that do not consume sequence numbers. In particular, there is no transmission for an ACK segment.

Noretransmission timerisset foranACK segment.

RetransmissionAfterRTO

A recent implementation of TCP maintains one retransmission time-out (RTO) timer for all outstanding (sent, but not acknowledged) segments. When the timer matures,the earliest outstanding segment isretransmitted even though lack ofa received ACK can be due to a delayed segment, a delayed ACK, or a lost acknowledgment. Note that no time-out timer is set for a segment that carries only an acknowledgment, which means that no such segment is resent.

RetransmissionAfterThreeDuplicateACKSegments

The previous rule about retransmission ofa segment is sufficient ifthe value of RTO is not very large. Sometimes, however, one segment is lost and the receiver receives so many out- of-order segments that theycannot be saved (limited buffer size).

Out-of-OrderSegments

When a segment is delayed, lost, or discarded, the segments following that segment arrive out of order. Originally, TCP was designed to discard all out-of-order segments, resulting in the retransmission of the missing segment and the following segments. Most implementations today do not discard the out-of-order segments. They storethemtemporarilyand flag themas out-of-order segments untilthe missingsegment arrives

Some Scenarios

Inthesescenarios,weshowa segmentbya rectangle.Ifthesegmentcarriesdata,
we show the range of byte numbers and the value of the acknowledgment field. If it carries only an acknowledgment, we show only the acknowledgment number in asmaller box.

NormalOperation

The first scenario shows bidirectional data transfer between two systems, as inFigure 4.23. The client TCP sends one segment; the server TCP sends three. The figure shows which rule applies to each acknowledgment. There are data to be sent, so the segment displays the next byte expected. When the client receives the first segment from the server, it does not have any more data to send; it sends only an ACK

segment.

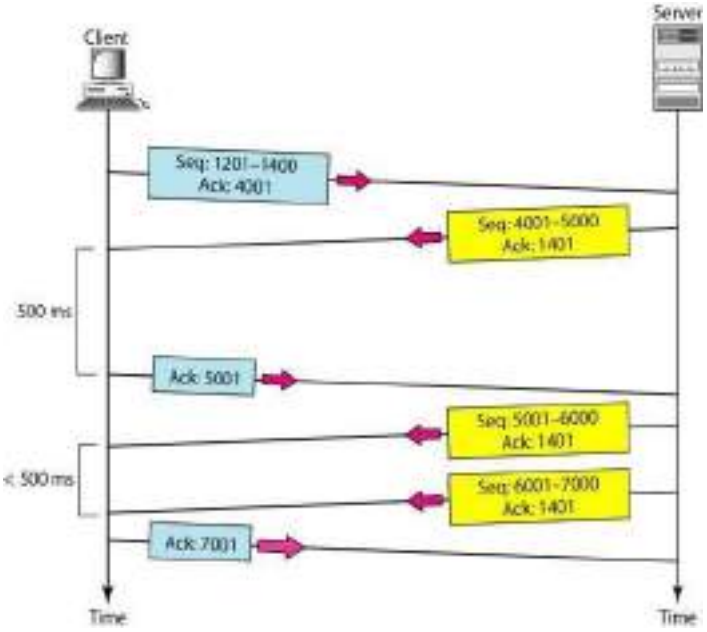


Figure Normaloperation

LostSegment

In this scenario, we show what happens when a segment is lost or corrupted. A lost segment and a corrupted segment are treated the same way by the receiver. A lost segment isdiscardedsomewhere inthenetwork;a corruptedsegment is discarded bythe receiveritself. Both are considered lost. Figure 4.24 shows a situation in which a segment is lost and discarded bysome router in the network, perhaps due to congestion.

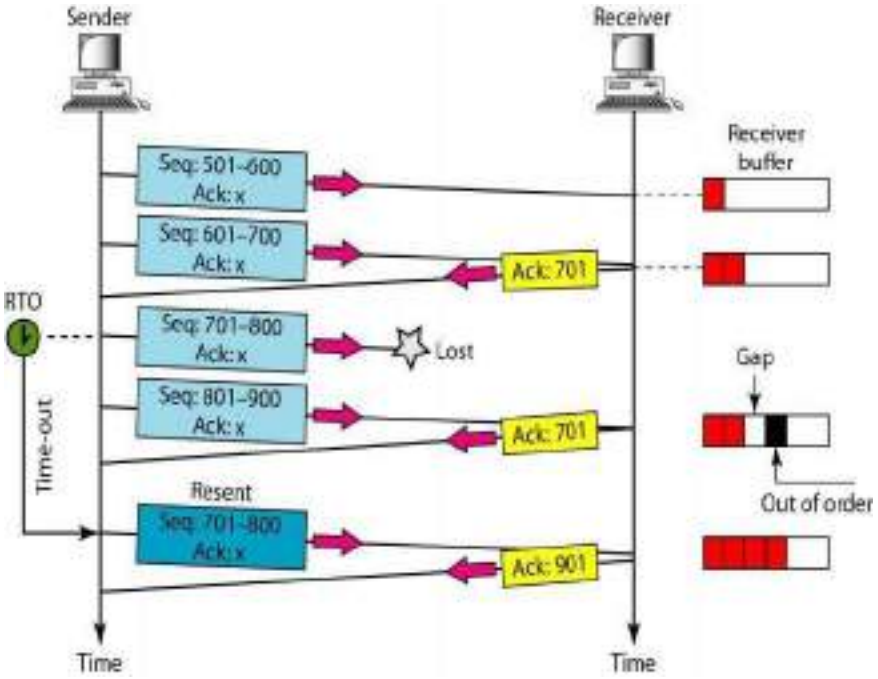


Figure Lostsegment.

We are assuming that data transfer is unidirectional: one site is sending and the other is receiving. In our scenario, the sender sends segments 1 and 2, which are acknowledged immediately by an ACK. Segment 3, however, are lost. The receiver receives segment 4, which is out of order. The receiver stores the data in the segment in its buffer but leavesa gap to indicate that there is no continuity in the data. The receiver immediately sends an acknowledgment tothe sender, displaying the next byte it expects. Notethatthe receiver stores bytes 801 to 900, but never delivers these bytes to the application until the gap is filled. ThereceiverTCPdelivers onlyordereddatatotheprocess.

FastRetransmission

In this scenario, we want to show the idea of fast retransmission. Our scenario is the sameas the second except that the RTO has a higher value (see Figure 4.25).

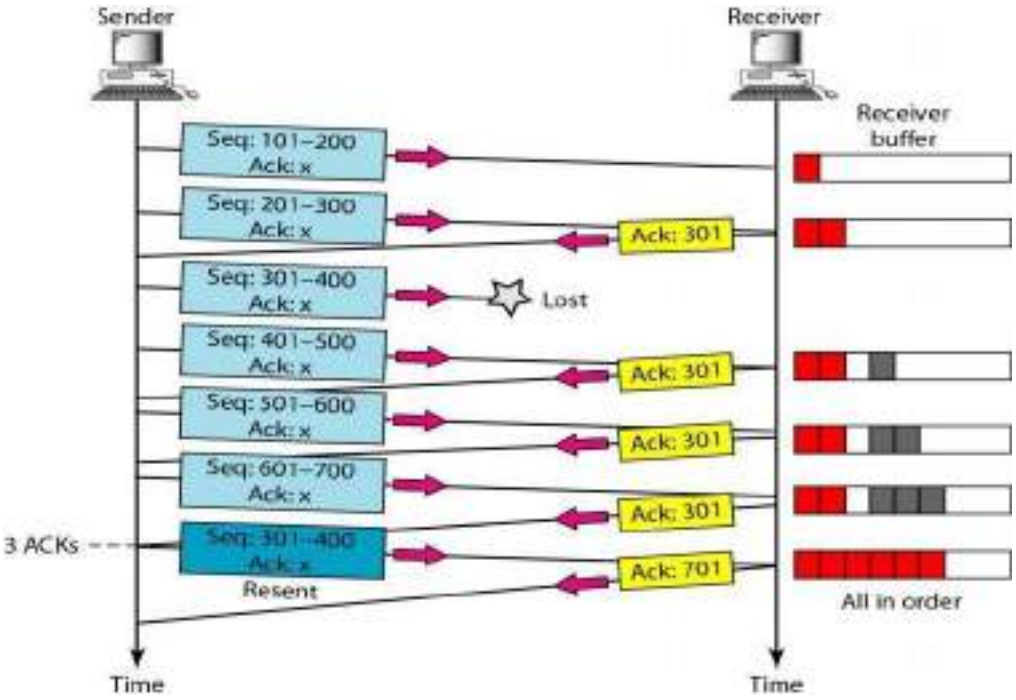


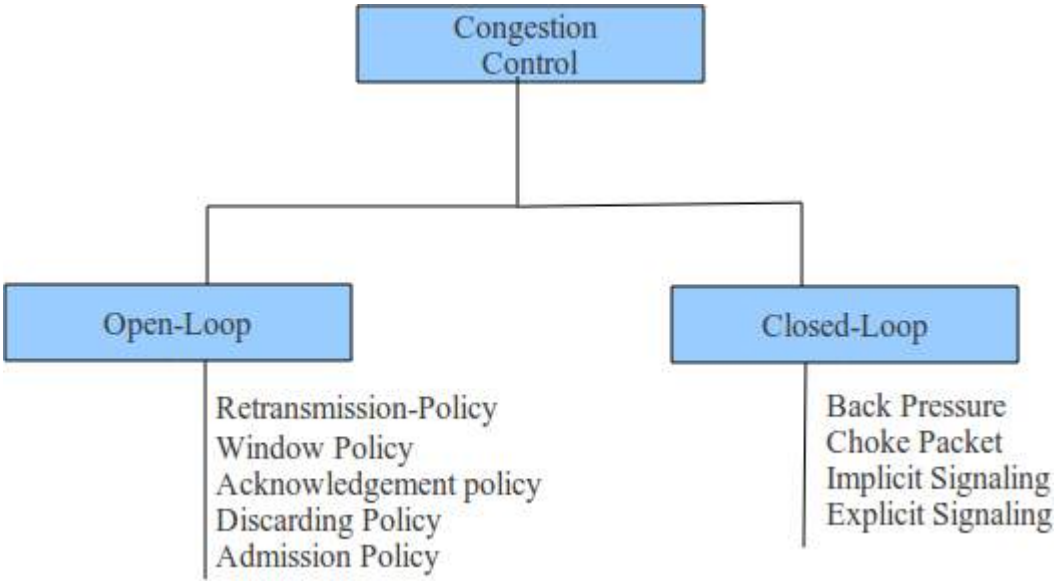
Figure FastRetransmission

When the receiver receives the fourth, fifth, and sixth segments, it triggers an acknowledgment. The sender receives four acknowledgments withthe same value (three duplicates). Although the timer for segment 3 has not matured yet, the fast transmission requires that segment 3, the segment that is expected by all these acknowledgments, be resent immediately.

Congestion control in TCP:

Congestion in a network may occur if the load on the network(the number of packets sent to the network) is greater than the capacityofthe network(the number ofpackets a network can handle). Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity .
Whentoomanypacketsarepumpedintothesystem,congestionoccurleadingintodegradationofperformance.
Congestiontendstofeeduponitselfandbackups.
Congestionshowslackofbalancebetweenvariousnetworkingequipments.
Itisaglobalissue.

In general, we can divide congestion control mechanisms into two broad categories: open- loop congestion control(prevention) and closed-loop congestion control (removal) as shown in Figure



OpenLoopCongestionControl:
Inopen-loopcongestioncontrol, policiesareappliedto prevent congestionbefore it happens. In these mechanisms, congestion control is handled by either the source or the destination

RetransmissionPolicy

Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion. For example, the retransmission policy used byTCPis designed to prevent or alleviate congestion.

WindowPolicy

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the Go-Back-N window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplicationmay make the congestionworse. The Selective Repeat window, on the other hand, tries to send the specificpackets that have been lost or corrupted.

AcknowledgmentPolicy:
The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion. Several approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A receiver may decide to acknowledge onlyN packets at atime. We need to knowthat theacknowledgments are also part ofthe load in a network. Sending fewer acknowledgments means imposing less load on the network.

DiscardingPolicy:

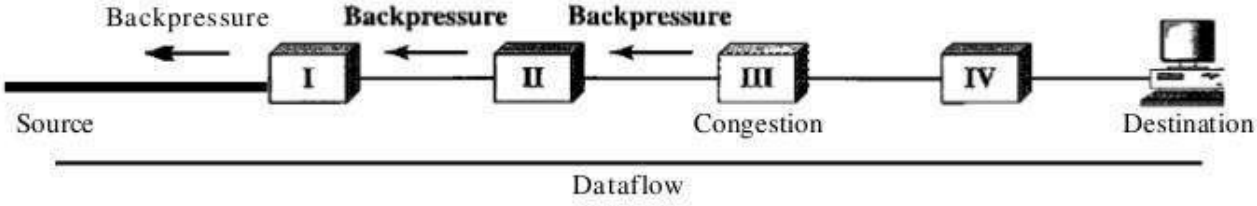
A good discarding policy by the routers may preventcongestion and at the same time may not harmthe integrityofthe transmission. Forexample, inaudio transmission, ifthe policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

AdmissionPolicy:
An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks. Switches in a flow first check the resource requirement ofa flow before admitting it to the network. A router can deny establishing a virtual- circuitconnection if there is congestion in the network or if there is a possibility of future congestion.

Closed-LoopCongestionControl

Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols.

Back-pressure:
The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from theirupstreamnodesornodes.Andso on.Backpressure is a node-to-nodecongestioncontrol that starts with a node and propagates, in the opposite direction of data flow, to the source. The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is corning.

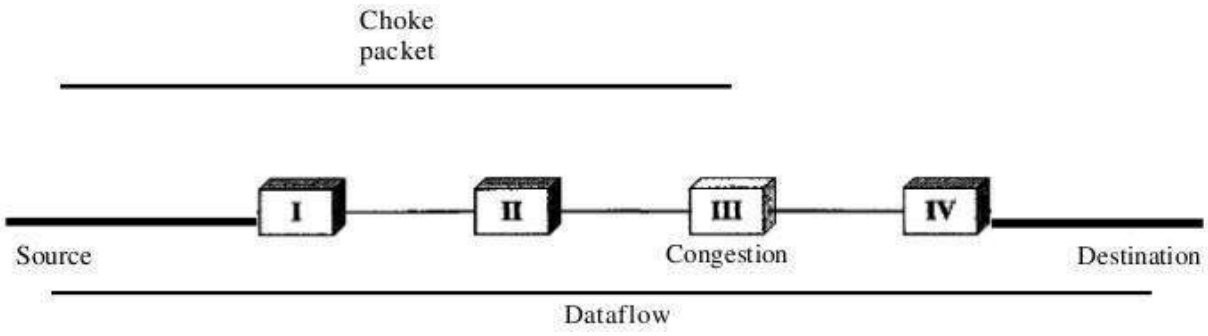


NodeIIlinthefigurehasmoreinputdatathanitcanhandle.Itdropssomepacketsin itsinputbufferandinforms node II to slow down. Node II, in turn, may be congested because it is slowing downthe output flow of data. Ifnode II is congested, it informs node I to slow down, which in turn may create congestion. If so, node Iinforms the source of datato slow down. This, in time, alleviates the congestion. Note that the pressure on node III is moved backward to the source to remove the congestion. None of the virtual-circuitnetworks we studied inthis book use backpressure. It was, however, implemented in the first virtual-circuit network, X.25. The technique cannot be implemented in a datagram network because inthis type ofnetwork, a node(router) does not have the slightest knowledge ofthe upstream router.

ChokePacket

A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke packet methods. In backpresure, the warning is fromone nodeto its upstreamnode, althoughthe warning mayeventuallyreachthe source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet hastraveled arenot warned.

WehaveseenanexampleofthistypeofcontrolinICMP. When arouter in the Internet is overwhelmed datagrams, it may discard some of them; but it informs the source . host, using a source quench ICMP message. The warning message goes directly to the source station; the intermediaterouters, and does not take any action. Figure shows the idea of a choke packet.



ImplicitSignaling

Inimplicitsignaling,thereisnocommunicationbetweenthecongestednodeornodes andthesource.Thesource guesses that there is a congestion somewherein the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested. The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down. We will see this type of signaling when we discuss TCP congestion control later in the chapter.

ExplicitSignaling

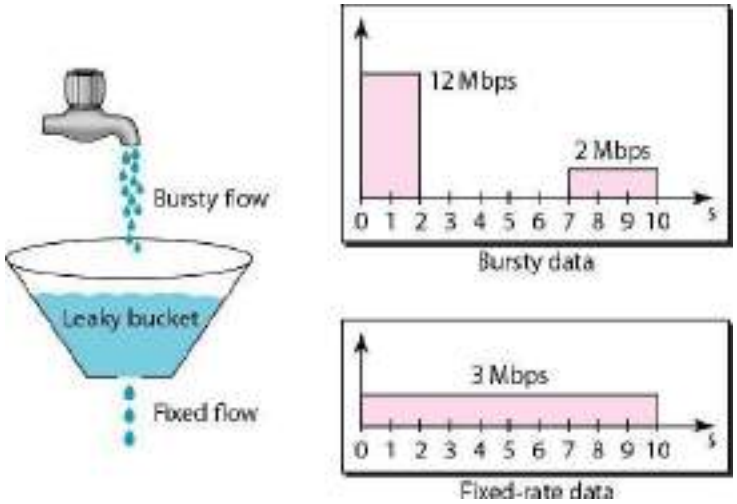
Thenodethatexperiencescongestioncanexplicitlysendasignaltothesourceor destination.The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method,thesignalis included inthepacketsthat carrydata. Explicit signaling, aswe will see in Frame Relay congestion control, can occur in either the forward or the backward direction.

Backward Signaling A bit can be set in a packet moving in the direction opposite to the congestion. This bit canwarn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.
Forward Signaling A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

TrafficShaping

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket and token bucket.

LeakyBucket



If a bucket has a small hole at the bottom, the water leaks from the bucket at a constantrateas long as there is water in the bucket. The rate at which the water leaks does not depend onthe rate at whichthe water is input to the bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant. Similarly, in networking, atechniquecalled leakybucket cansmoothout burstytraffic. Burstychunks arestoredinthe bucket andsentout at anaverage rate. Figure4.34showsa leakybucket and its effectt

In the figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use ofthe leakybucket shapes the input traffic to make it conformtothis commitment. In Figure 4.34 the host sends a burst of data at a rate of 12 Mbps for 2 s,for atotalof24 Mbits ofdata. The host is silent for 5 s and then sends data at arateof2 Mbps for 3 s, for a total of 6 Mbits of data. In all, the host has sent 30 Mbits of data in lOs. The leakybucket smooth's the traffic bysendingout dataat arateof3Mbps during the same 10 s.

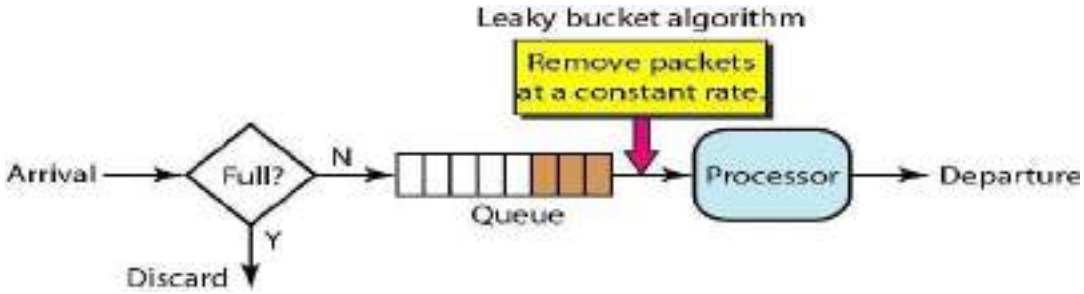


Figure4.35Leakybucketimplementation

A simple leaky bucket implementation is shown in Figure 4.35. A FIFO queue holds the packets. If the traffic consists of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number ofpackets fromthe queue at eachtick of the clock. If the traffic consistsof variable-length packets, the fixed output rate must be based on the number of bytes or bits.
Thefollowingisanalgorithmforvariable-lengthpackets:
Initializeacountertonatthetickoftheclock.
If n is greater than the size of the packet, send the packet and decrement thecounter bythe packet size. Repeat this step until n is smaller than the packetsize.
Reset thecounterandgotostep1.
Aleakybucketalgorithmshapesburstytrafficintofixed-ratetrafficby averaging the data rate. It may drop the packets if the bucket is full.

TokenBucket

The leaky bucket is veryrestrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty. Now if the host has bursty data, the leaky bucket allows only an average rate. The time when the host was idle isnot taken into account. On the other hand, the token bucket algorithm allows idlehosts to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends n tokens to the bucket. The system removes one token for every cell (or byte) of data sent. For example, if n is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens.

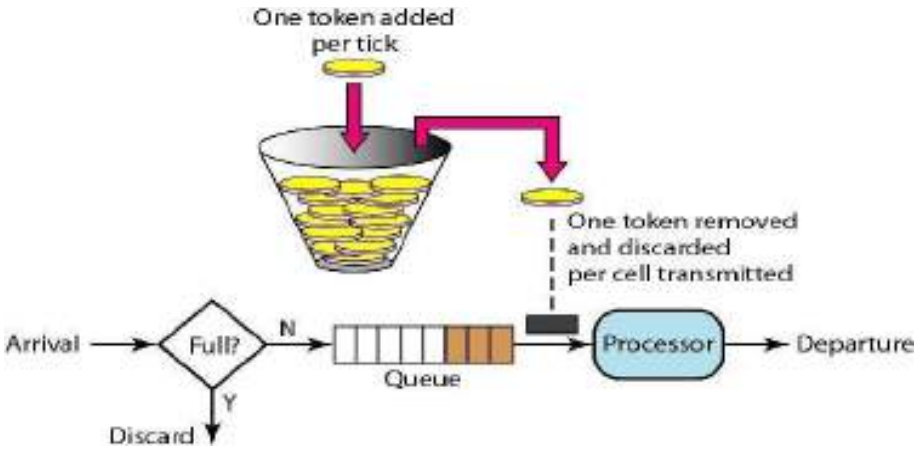


Figure4.36Token bucket

The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by1. When the counter is zero, the host cannot send data.
Thetokenbucketallowsburstytrafficataregulatedmaximumrate.

CombiningTokenBucket andLeakyBucket

The two techniques can be combined to credit an idle host and at the same time regulate the traffic. The leaky bucket is applied after the token bucket; the rate of the leaky bucket needs to be higher than the rate oftokens dropped in the bucket.

ResourceReservation

Aflowofdataneedsresourcessuchasabuffer,bandwidth,CPUtime,andso on.Thequality of service is improved if these resources are reserved beforehand. We discuss inthis section one QoS modelcalled Integrated Services, which depends heavily onresource reservation to improve the quality of service.

AdmissionControl

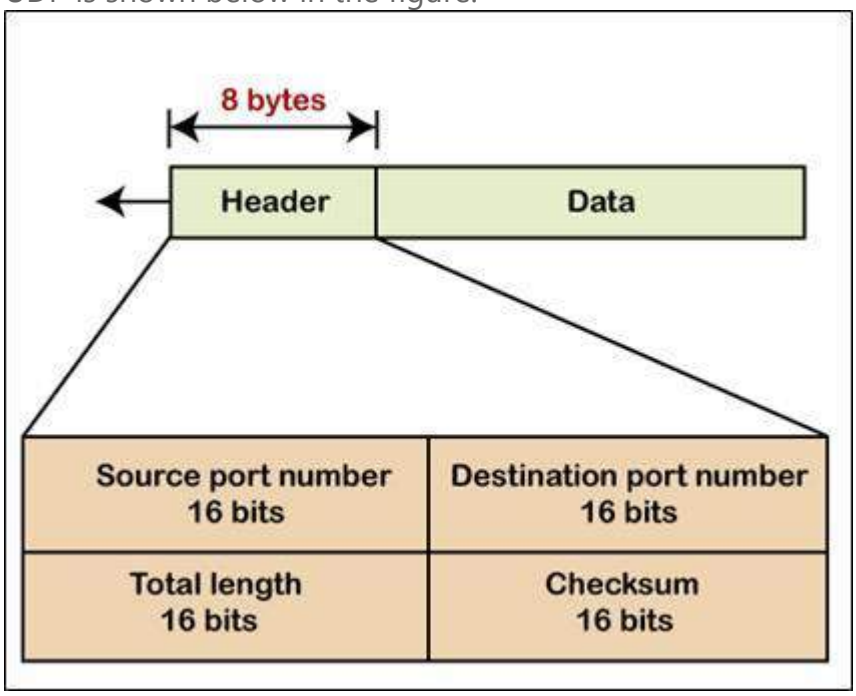
Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a routeraccepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

Future of TCP:

TCP majorly focus on reliable transmission of data than speed
Used in new applications like www, E-Commerce
It increases its security
Adjustable for scalability and flexibility
Adopt new technologies
Built new protocols
Support new computer network architecture

UserDatagramProtocolFormat

The UDP format is very simple. The header size ofthe UDP is 8 bytes (8 bytes mean 64 bits). The format of UDP is shown below in the figure.



It hasfourfieldsshown below:

SourcePortNumber: Thesizeofthesourceportis16bits.Itisusedtoidentifytheprocessofthesender.

Destination Port Number: The size of the destination port is 16 bits. It is used to identifythe process of the receiver.

Totallength: Thesizeofthetotallengthis16bits.ItdefinesthetotallengthoftheUDPandalsostoresthe lengthofthedataandheader.

UDPlength=IPlength–IPheader'slength

Checksum: Thesizeofthechecksumportis16bits.Itisusedtodetecterrorsacrosstheentireuser datagram.

Advantagesof UDP

Youcaneasilybroadcastandmulticasttransmissionthroughthe UDP.

Itis fasterthanTCP.

Itusesthe smallsizeofthe header(8bytes).

Ittakeslessmemorythanotherprotocols.

Wheneverdatapacketsneedtobetransmitted,thenUDPisused.

DisadvantagesofUDP

It isanunreliable protocolfor transmission.

Thereis nosuchfunctioninittoknowthatthedatahas beenreceivedor not.

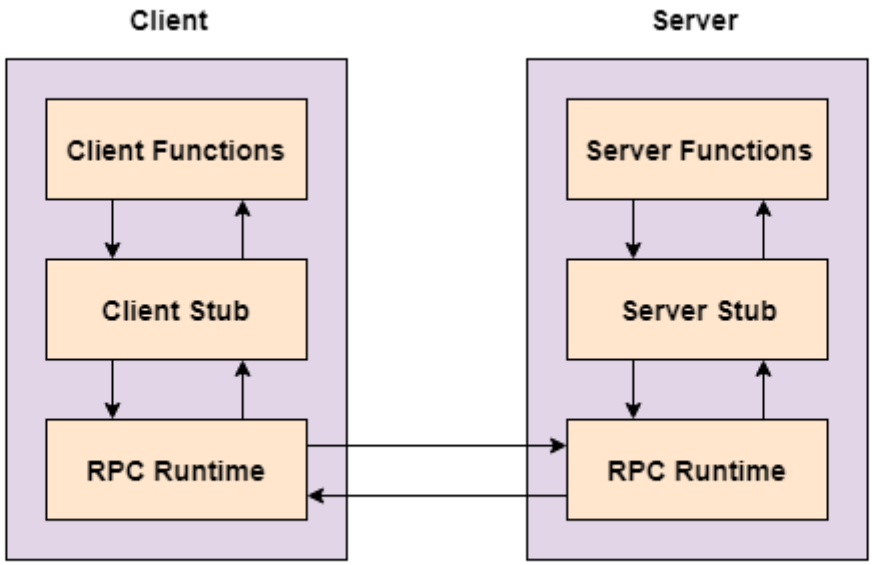
ThehandshakemethodisnotusedinUDP.

Itdoesnotcontrolthecongestion.

The main disadvantage of using routers with UDP is that once transmission failure occurs, the routers do not transmit the datagram again

A remote procedure call is an **interprocess communication** technique that is used for **client-server based applications**. It is also known as a subroutine call or a function call.

A client has a request message that the RPC translates and sends to the server. This request may be a procedure or a function call to a remote server. When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.
The sequence of events in a remote procedure call are given as follows −
The client stub is called by the client.
The client stub makes a system call to send the message to the server and puts the parameters in the message.
The message is sent from the client to the server by the client's operating system.
The message is passed to the server stub by the server operating system.
The parameters are removed from the message by the server stub.
Then, the server procedure is called by the server stub.
A diagram that demonstrates this is as follows −



Advantages of Remote Procedure Call
Some of the advantages of RPC are as follows −
Remote procedure calls support process oriented and thread oriented models.
The internal message passing mechanism of RPC is hidden from the user.
The effort to re-write and re-develop the code is minimum in remote procedure calls.
Remote procedure calls can be used in distributed environment as well as the local environment.
Many of the protocol layers are omitted by RPC to improve performance.
Disadvantages of Remote Procedure Call
Some of the disadvantages of RPC are as follows −
The remote procedure call is a concept that can be implemented in different ways. It is not a standard.
There is no flexibility in RPC for hardware architecture. It is only interaction based.
There is an increase in costs because of remote procedure call.
What is the Real-time Transport Protocol (RTP)?
Real-time Transport Protocol (RTP) is a network protocol for the delivery of audio and video over the

internet. It is designed to provide end-to-end network transport functions suitable for applications transmitting real-time data, such as audio and video.

RTP is used in conjunction with the Real-time Transport Control Protocol (RTCP), which is used to monitor the quality of the data transmission. RTP provides the actual delivery of the media, while RTCP is used to provide feedback on the quality of the transmission and to provide other control information.

RTP is a packet-based protocol, which means that it breaks the media stream into packets for transmission over the network. Each packet is given a sequence number, which allows the receiver to reassemble the packets in the correct order. RTP also includes a timestamp, which allows the receiver to synchronize the audio and video streams.

RTP is widely used in a variety of applications, including voice over IP (VoIP), video conferencing, and streaming media. It is supported by many media players and servers, and it is often used in conjunction with other protocols, such as RTSP and SIP, to deliver audio and video content over the internet.

What applications use the Real-time Transport Protocol?

Real-time Transport Protocol (RTP) is widely used in a variety of applications that require the delivery of real-time audio and video over the internet. Some examples of applications that use RTP include −

Voice over IP (VoIP) − RTP is commonly used in VoIP systems to transmit audio over the internet.It allows for the real-time delivery of voice calls with low latency.

Video conferencing − RTP is often used in video conferencing systems to transmit audio and video in real time. It allows for the synchronous communication of multiple participants.

Streaming media − RTP is used in many streaming media applications to deliver audio and video over the internet. It is often used in conjunction with other protocols, such as RTSP and HTTP, to stream media to clients.

Telephony − RTP is used in many telephony systems to transmit audio and video between devices. It allows for the real-time communication of multiple parties in a call.

Broadcast television − RTP is used in some broadcast television systems to transmit audio and video over the internet. It allows for the delivery of live television streams to viewers.

Overall, RTP is a widely used protocol for the delivery of real-time audio and video over the internet.It is supported by many media players and servers and is an important part of the infrastructure that enables the streaming of multimedia content.

→ Differences btw the TCP & UDP :-

TCP	UDP
1) It stands for Transmission Control protocol.	It stands for user datagram protocol.
2) It is a connection-oriented protocol, which means that the connection needs to be established before the data is transmitted over the N/w.	It is a connectionless protocol, which means that it sends the data without checking whether the system is ready to receive or not.
3) TCP is a reliable protocol as it provides assurance for the delivery of data packets.	UDP is an unreliable protocol as it does not take the guarantee for the delivery of packets.
4) TCP is slower than UDP as it performs error checking, flow control.	UDP is faster than TCP.
5) The size of TCP is 20 bytes.	5) The size of the UDP is 8 bytes.
6) TCP uses the three way-handshake concept. In this concept, If the sender receives the ack, then the sender will send the data. TCP also has the ability to resend the lost data.	UDP does not wait for any acknowledgment. It just sends the data.

7) It follows the flow control mechanism in which too many packets cannot be sent to the receiver at the same time.

this protocol follows no such mechanism.

8) TCP performs error checking by using a checksum. when the data is corrected, then the data is retransmitted to the receiver.

It does not perform any error checking, & also does not resend the lost data packets.

a) this protocol is mainly used where a secure & reliable communication process is required, like military services, web browsing, & e-mail.

this protocol is used where fast communication is required & does not care about the reliability like game streaming, video & music streaming etc.

UNIT-V

UNIT- IV Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Content Delivery.

Application Layer:

The Application Layer is topmost layer in the Open System Interconnection (OSI) model. This layer provides several ways for manipulating the data (information) which actually enables any type of user to access network with ease. This layer also makes a request to its bottom layer, which is presentation layer for receiving various types of information from it. The Application Layer interface directly interacts with application and provides common web application services. This layer is basically highest level of open system, which provides services directly for application process.

Present Layer => Application Layer

Presentation LayerSession

Layer Transport Layer

Network Layer

Data Layer Physical

Layer

Functions of Application Layer :

The Application Layer, as discussed above, being topmost layer in OSI model, performs several kinds of functions which are requirement in any kind of application or communication process.
Following are list of functions which are performed by Application Layer of OSI Model –

```
Data from User <=> Application layer <=> Data from Presentation Layer
```

- Application Layer provides a facility by which users can forward several emails and it also provides a storage facility.
- This layer allows users to access, retrieve and manage files in a remote computer.
- It allows users to log on as a remote host.
- This layer provides access to global information about various services.
- This layer provides services which include: e-mail, transferring files, distributing results to the user, directory services, network resources and so on.
- It provides protocols that allow software to send and receive information and present meaningful data to users.
- It handles issues such as network transparency, resource allocation and so on.

- This layer serves as a window for users and application processes to access network services.
- Application Layer is basically not a function, but it performs application layer functions.
- The application layer is actually an abstraction layer that specifies the shared protocols and interface methods used by hosts in a communication network.
- Application Layer helps us to identify communication partners, and synchronizing communication.
- This layer allows users to interact with other software applications.
- In this layer, data is in visual form, which makes users truly understand data rather than remembering or visualize the data in the binary format (0's or 1's).
- This application layer basically interacts with Operating System (OS) and thus further preserves the data in a suitable manner.
- This layer also receives and preserves data from it's previous layer, which is Presentation Layer (which carries in itself the syntax and semantics of the information transmitted).
- The protocols which are used in this application layer depend upon what information users wish to send or receive.
- This application layer, in general, performs host initialization followed by remote login to hosts.

Working of Application Layer in the OSI model :

In the OSI model, this application layer is narrower in scope.
The application layer in the OSI model generally acts only like the interface which is responsible for communicating with host-based and user-facing applications. This is in contrast with TCP/IP protocol, wherein the layers below the application layer, which is Session Layer and Presentation layer, are clubbed together and form a simple single layer which is responsible for performing the functions, which includes controlling the dialogues between computers, establishing as well as maintaining as well as ending a particular session, providing data compression and data encryption and so on.
At first, client sends a command t server and when server receives that command, it allocates port number to client. Thereafter, the client sends an initiation connection request to server and when server receives request, it gives acknowledgement (ACK) to client through client has successfully established a connection with the server and, therefore, now client has access to server through which it may either ask server to send any types of files or other documents or it may upload some files or documents on server itself.

Features provided by Application Layer Protocols :

To ensure smooth communication, application layer protocols are implemented the same on source host and destination host.

The following are some of the features which are provided by Application layer protocols-

- The Application Layer protocol defines process for both parties which are involved in communication.
- These protocols define the type of message being sent or received from any side (either source host or destination host).
- These protocols also define basic syntax of the message being forwarded or retrieved.
- These protocols define the way to send a message and the expected response.
- These protocols also define interaction with the next level.

Application Layer Protocols:

The application layer provides several protocolswhich allow any software to easily send and receive information and present meaningful data to its users. The following are some of the protocols which are provided by the application layer.

- **TELNET:** Telnet stands for Telecommunications Network. This protocol is used for managing files over the Internet. It allows the Telnet clients to access the resources of Telnet server. Telnet uses port number 23.
- **DNS:** DNS stands for Domain Name System. The DNS service translates the domain name (selected by user) into the corresponding IP address. For example- If you choose the domain name as www.abcd.com, then DNS must translate it as 192.36.20.8 (random IP address written just for understanding purposes). DNS protocol uses the port number 53.
- **DHCP:** DHCP stands for Dynamic Host Configuration Protocol. It provides IP addresses to hosts. Whenever a host tries to register for an IP address with the DHCP server, DHCP server provides lots of information to the corresponding host. DHCP uses port numbers 67 and 68.
- **FTP:** FTP stands for File Transfer Protocol. This protocol helps to transfer different files from one device to another. FTP promotes sharing of files via remote computer devices with reliable, efficient data transfer. FTP uses port number 20 for data access and port number 21 for data control.
- **SMTP:** SMTP stands for Simple Mail Transfer Protocol. It is used to transfer electronic mail from one user to another user. SMTP is used by end users to send emails with ease. SMTP uses port numbers 25 and 587.
- **HTTP:** HTTP stands for Hyper Text Transfer Protocol. It is the foundation of the World Wide Web (WWW). HTTP works on the client server model. This protocol is used for transmitting hypermedia documents like HTML. This protocol was designed particularly for the communications between the web browsers and web servers, but this protocol can also be used for several other purposes. HTTP is a stateless protocol (network protocol in which a client sends requests to server and server responses back as per the given

state), which means the server is not responsible for maintaining the previous client's requests. HTTP uses port number 80.

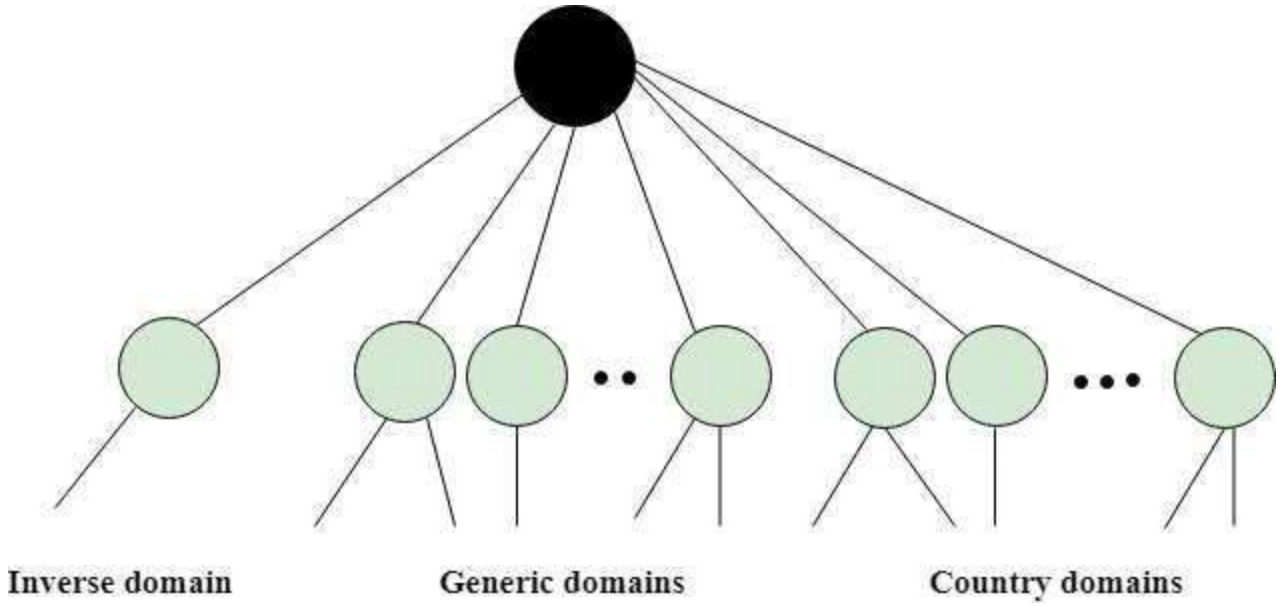
- **NFS:** NFS stands for Network File System. This protocol allows remote hosts to mount files over a network and interact with those file systems as though they are mounted locally. NFS uses the port number 2049.
- **SNMP:** SNMP stands for Simple Network Management Protocol. This protocol gathers data by polling the devices from the network to the management station at fixed or random intervals, requiring them to disclose certain information. SNMP uses port numbers 161 (TCP) and 162 (UDP).

Domain Name System (DNS):

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at facebook.com had an IP address of 132.147.165.50,most people would reach this site by specifying ftp.facebook.com. Therefore, the domain name is more reliable than IP address.

DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.

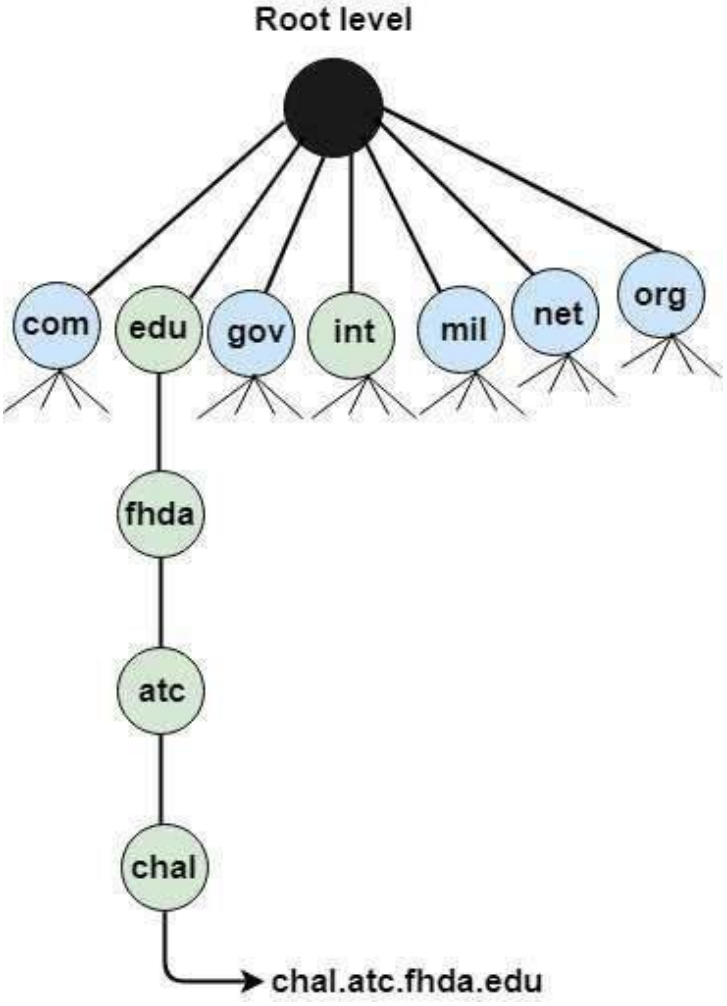


Generic Domains

- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International Organizations

mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations



Country Domain

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

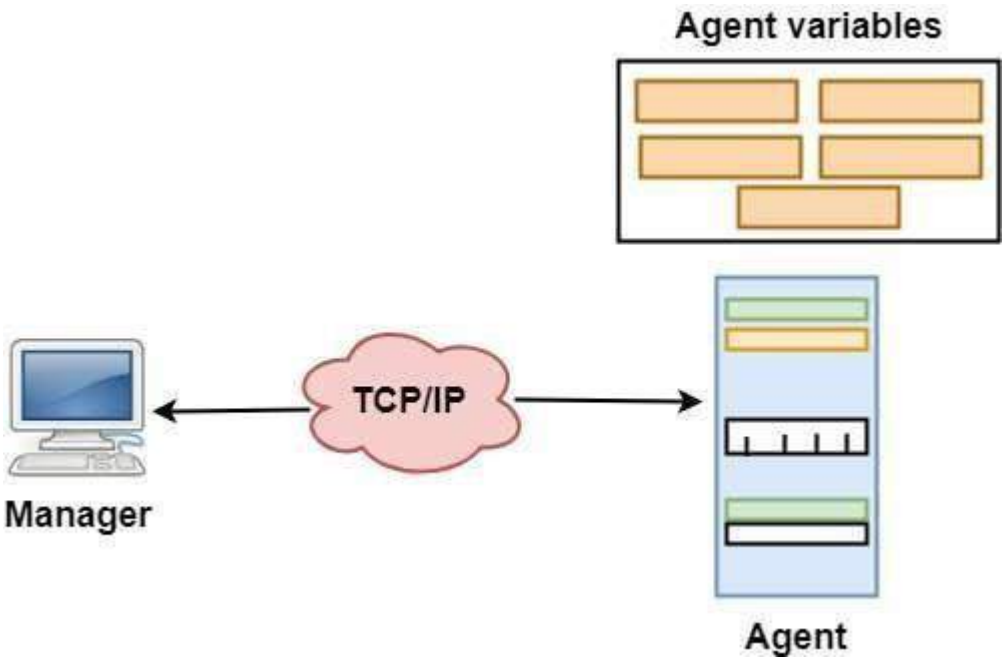
Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as **DNS resolver** sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

SNMP

- SNMP stands for **Simple Network Management Protocol**.
- SNMP is a framework used for managing devices on the internet.
- It provides a set of operations for monitoring and managing the internet.

SNMP Concept



- SNMP has two components Manager and agent.
- The manager is a host that controls and monitors a set of agents such as routers.
- It is an application layer protocol in which a few manager stations can handle a set of agents.
- The protocol designed at the application level can monitor the devices made by different manufacturers and installed on different physical networks.
- It is used in a heterogeneous network made of different LANs and WANs connected by routers or gateways.

Managers & Agents

- A manager is a host that runs the SNMP client program while the agent is a router that runs the SNMP server program.
- Management of the internet is achieved through simple interaction between a manager and agent.
- The agent is used to keep the information in a database while the manager is used to access the values in the database. For example, a router can store the appropriate variables such as a number of packets received and forwarded while the manager can compare these variables to determine whether the router is congested or not.

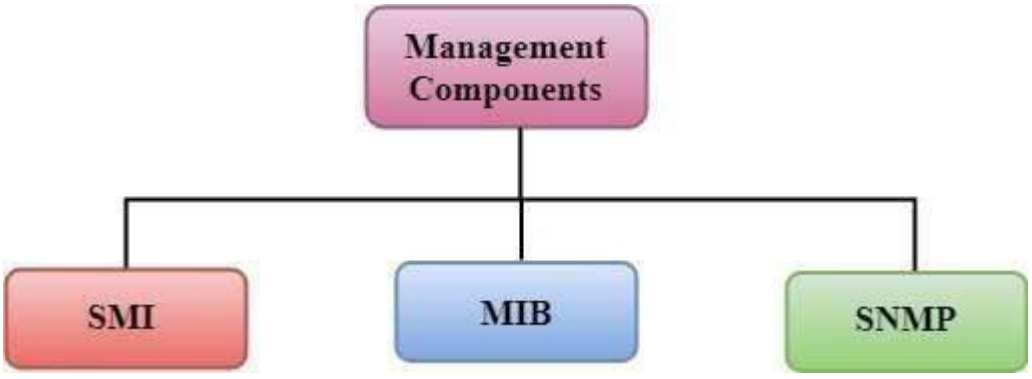
- Agents can also contribute to the management process. A server program on the agent checks the environment, if something goes wrong, the agent sends a warning message to the manager.

Management with SNMP has three basic ideas:

- A manager checks the agent by requesting the information that reflects the behavior of the agent.
- A manager also forces the agent to perform a certain function by resetting values in the agent database.
- An agent also contributes to the management process by warning the manager regarding an unusual condition.

Management Components

- Management is not achieved only through the SNMP protocol but also the use of other protocols that can cooperate with the SNMP protocol. Management is achieved through the use of the other two protocols: SMI (Structure of management information) and MIB(management information base).
- Management is a combination of SMI, MIB, and SNMP. All these three protocols such as abstract syntax notation 1 (ASN.1) and basic encoding rules (BER).

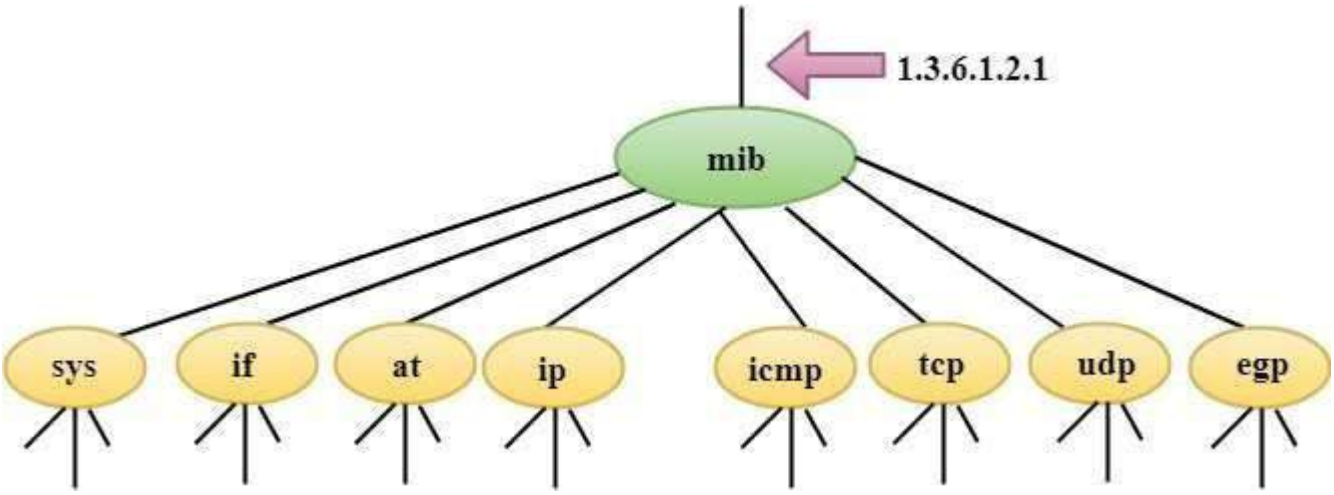


SMI

The SMI (Structure of management information) is a component used in network management. Its main function is to define the type of data that can be stored in an object and to show how to encode the data for the transmission over a network.

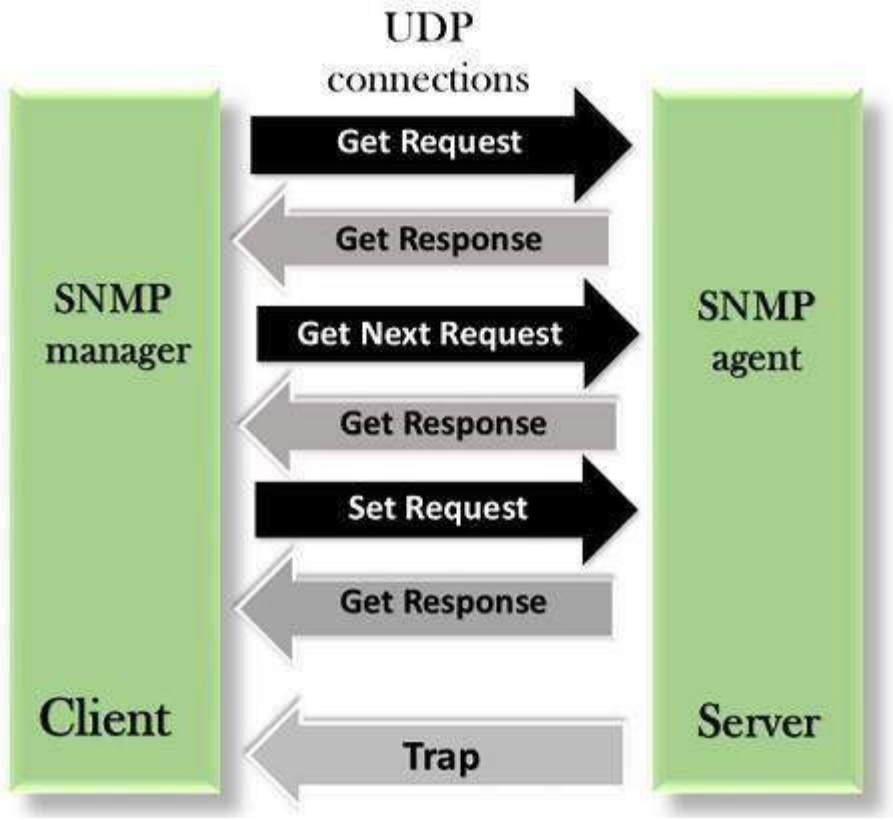
MIB

- The MIB (Management information base) is a second component for the network management.
- Each agent has its own MIB, which is a collection of all the objects that the manager can manage. MIB is categorized into eight groups: system, interface, address translation, ip, icmp, tcp, udp, and egp. These groups are under the mib object.



SNMP

SNMP defines five types of messages: GetRequest, GetNextRequest, SetRequest, GetResponse, and Trap.



GetRequest: The GetRequest message is sent from a manager (client) to the agent (server) to retrieve the value of a variable.

GetNextRequest: The GetNextRequest message is sent from the manager to agent to retrieve the value of a variable. This type of message is used to retrieve the values of the entries in a table. If the manager does not know the indexes of the entries, then it will not be able to retrieve the values. In such situations, GetNextRequest message is used to define an object.

GetResponse: The GetResponse message is sent from an agent to the manager in response to the GetRequest and GetNextRequest message. This message contains the value of a variable requested by the manager.

SetRequest: The SetRequest message is sent from a manager to the agent to set a value in a variable.

Trap: The Trap message is sent from an agent to the manager to report an event. For example, if the agent is rebooted, then it informs the manager as well as sends the time of rebooting.

Electronic Mail:

Electronic mail (e-mail) is a computer-based programme that allows users to send and receive messages. E-mail is the electronic version of a letter, but with time and flexibility advantages. While a letter can take anywhere from a week to a couple of months to reach its intended destination, an e-mail is sent virtually almost instantly.
Messages in the mail contain not just text but also photos, audio, and video data. A person sending an e-mail is a **sender**, and the person receiving it is the **recipient**.

Features Of E-Mail

Spontaneity**:** In a couple of seconds, you may send a message to anybody on the globe.

Asynchronous: You may send the e-mail and let the recipient view it at their leisure.

Attachments of data, pictures, or music, frequently in compressed forms, can be delivered as an e-mail to a person anywhere in the world.

Addresses can be stored in an address book and retrieved instantly.

Through an e-mail, a user can transfer multiple copies of a message to various individuals.

Services offered by E-mail system :

Composition - Creating messages and responses is referred to as composition.

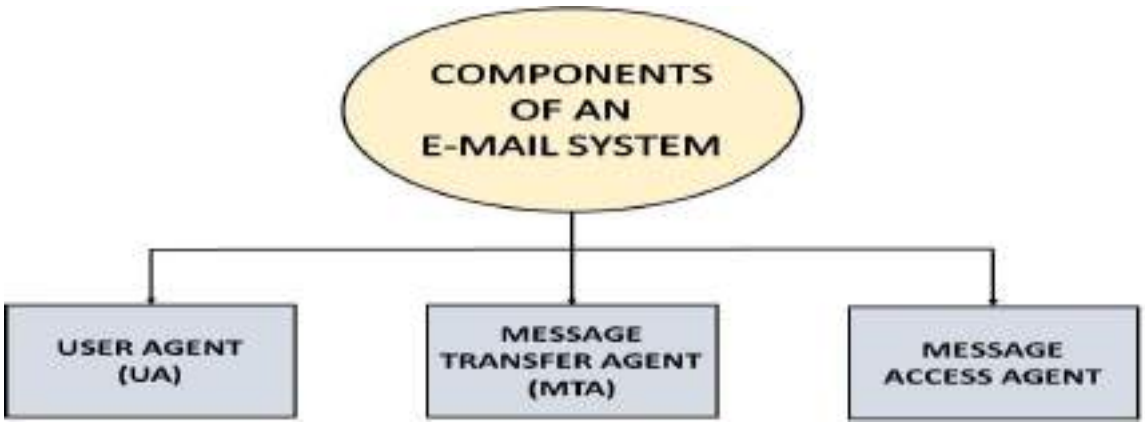
Transfer - Sending mail from the sender to the receiver is referred to as a transfer.

Reporting - Mail delivery confirmation is known as reporting. It allows users to see if
their mail has been delivered, misplaced, or rejected.
Displaying – This refers to presenting messages so that the user can understand.
Disposition — This stage is concerned with the recipient's actions after receiving mail, such as saving it, deleting it before reading it, or after reading it.

Components Of E-mail System

The following are the essential components of an e-mail system:

1. User Agent (UA)
2. Message Transfer Agent (MTA)
3. Message Access Agent



User Agent (UA): The User-Agent is a simple software that sends and receives mail. It is also known as a mail reader. It supports a wide range of instructions for sending, receiving, and replying to messages, as well as manipulating mailboxes.
Some of the services supplied by the User-Agent are listed below:

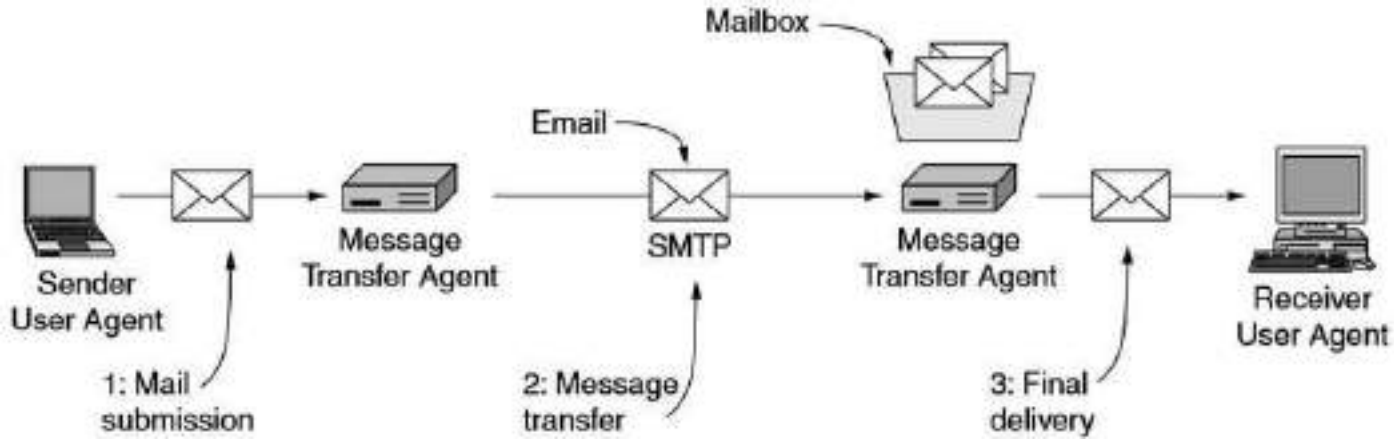
- Reading a Message
- Sending a reply to a Message

- Message Composition
- Forwarding a Message
- Handling the Message

Message Transfer Agent: The Message Transfer Agent manages the actual e-mail transfer operation (MTA). Simple Mail Transfer Protocol sends messages from one MTA to another. A system must have both a client MTA and a system MTA to send an e-mail. If the recipients are connected to the same computer, it sends mail to their mailboxes. If the destination mailbox is on another computer, it sends mail to the receiver's MTA.

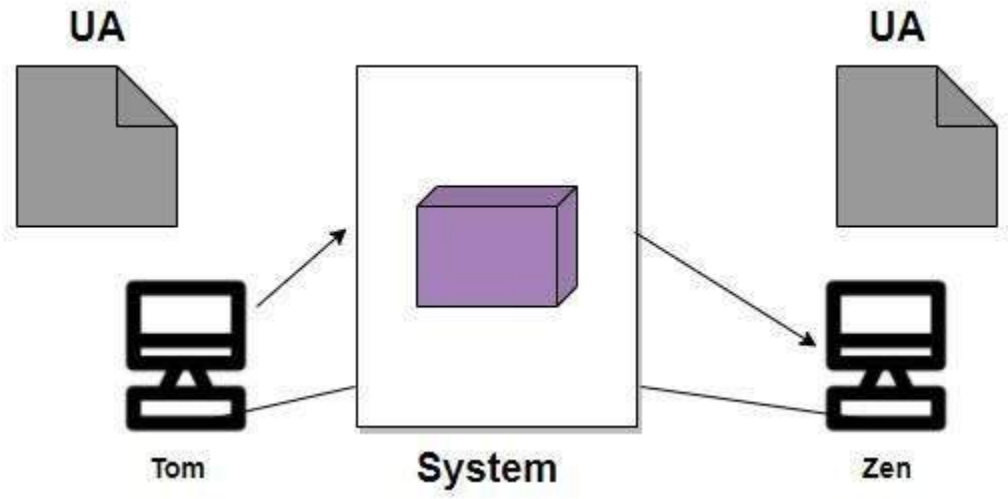
Message Access Agent: Simple Mail Transfer Protocol is used for the first and second stages of e-mail delivery.
The pull protocol is mainly required at the third stage of e-mail delivery, and the message access agent is used at this point.
POP and IMAP4 are the two protocols used to access messages.

Architecture of E-mail

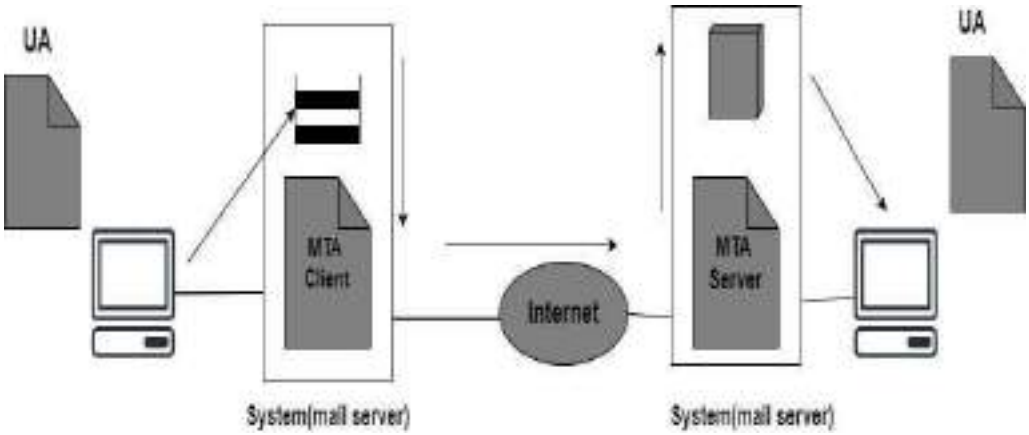


Source: Researchgate.net

First Scenario:
In the first scenario, two user agents are required. The sender and recipient of the e-mail share the same machine directly connected to the server.
For example, let us consider two user agents, Tom and Zen. When Tom sends an e-mail to Zen, the user agent (UA) programme is used to prepare the message. Following that, this e-mail gets saved in Zens inbox.

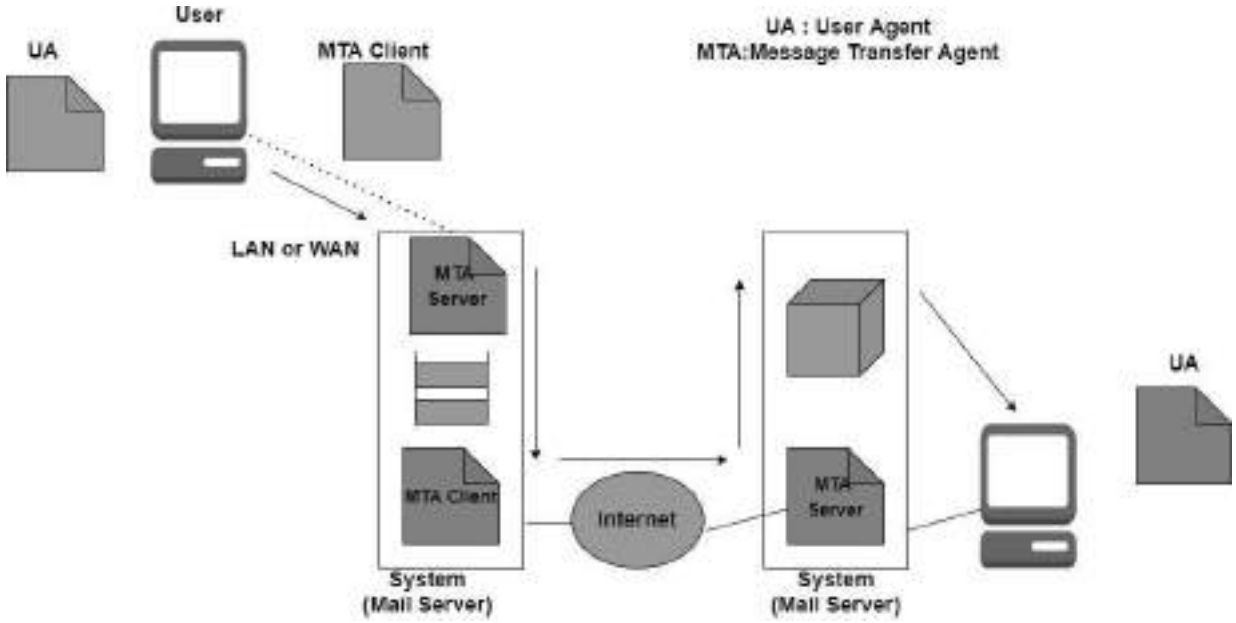


Second Scenario:
The sender and recipient of an e-mail, in this case, are essentially users on two different machines over the internet. User-Agents and Message Transfer Agents(MTA) are required in this scenario.



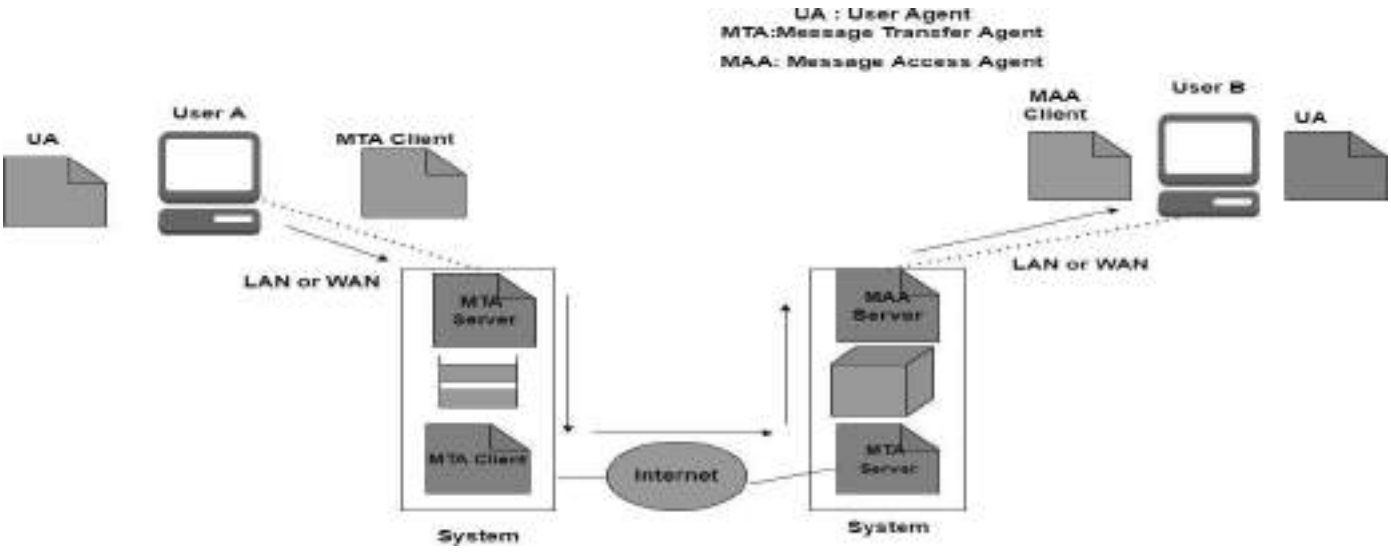
Take, for example, two user agents (John and Micheal), as illustrated in the diagram. When John sends an e-mail to Micheal, the user agent (UA) and message transfer agents (MTAs) programmes prepare the e-mail for transmission over the internet. Following that, this e-mail gets stored in Micheal's inbox.

Third Scenario:
The sender is connected to the system by a point-to-point WAN, which can be a dial-up modem or a cable modem in this case. On the other hand, the receiver is directly attached to the system, as it was in the second scenario.
The sender also needs a User agent (UA) to prepare the message in this situation. After preparing the statement, the sender delivers it over LAN or WAN via a pair of MTAs.



Fourth Scenario:
The recipient is linked to the mail server via WAN or LAN in this scenario. When the message arrives, the recipient must retrieve it, which needs additional client/server agents. This scenario requires two user

agents (UAs), two pairs of message transfer agents (MTAs), and a couple of message access agents (MAAs).



Message Format in E-Mail

Let us understand the RFC 822 message format in an email.

Messages consist of a primitive envelope, some header fields and a blank line, and the message body. Each header field logically includes a single line of ASCII text which contains the field name, a colon and a field. RFC 822 is an old standard. Usually, the user agent builds a message and passes it to be the message transfer agent with the user's header fields to construct an envelope.

The following table shows the principal header fields related to message transport.

RFC 822 header fields related to message transport

Header	**Meaning**
To –	E-mail address of primary recipients.
Header	**Meaning**

Cc −	E-mail address of secondary recipients.
BCC	E-mail address for blind carbon copies.
From −	Person who created the message.
Sender	E-mail address of the actual sender.
Received	Line inserted by each transfer agent along the route.
Return Patch	It can use it to identify the path to the sender.

The To − field

The field gives the DNS address of the primary recipient. It is allowed to have multiple recipients.

The Cc − field

This field gives the addresses of any secondary recipients.

The Bcc

The long form of Bcc is Blind Carbon Copy. This field is such as the Cc field, except that this is removed from all the copies shared with the primary and secondary recipients. This feature allows people to send copies to third parties without primary and secondary recipients knowing this.

From − and Sender fields

These fields tell about who wrote the message and who sent the message, respectively, because the person who creates the message and the person who sends it can be different.

The from the field is required, but the sender field can be omitted if it is the same as the one from the field. These fields are required in case the message is undeliverable and is to be returned to the sender.

Received field

A-line containing the Received field is added by each message transfer agent along the way. This line carries the agent's identity, date and time at which they received the message. It also contains some other information that can be used to find bugs in the routing system.

The Return-Path− field

The final message transfer agent adds this field, and it is predetermined to tell how to receive back to the sender. It can gather this information from all the received headers.

Other header fields

In addition to the field to table below, RFC 822 messages may contain various header fields used by user agents or human recipients. Many of them are shown in the table below

Some fields in RFC 822 message header are as follows :

Header	**Meaning**
Date:	The date and time of the message.
Reply− To	The E-mail address to which the reply is to be sent.
Message-Id:	Message identifying number
In− Reply − To:	Message-Id of the message to which this is a reply.
References:	Other relevant messages identifying numbers.
Keywords:	Keywords chosen by users.
Subject:	Summary of the message for the one-line display.

The RFC 822 allows the users to invent new headers for their private use, but these headers must start with the string X − Event of the week.

Message Body

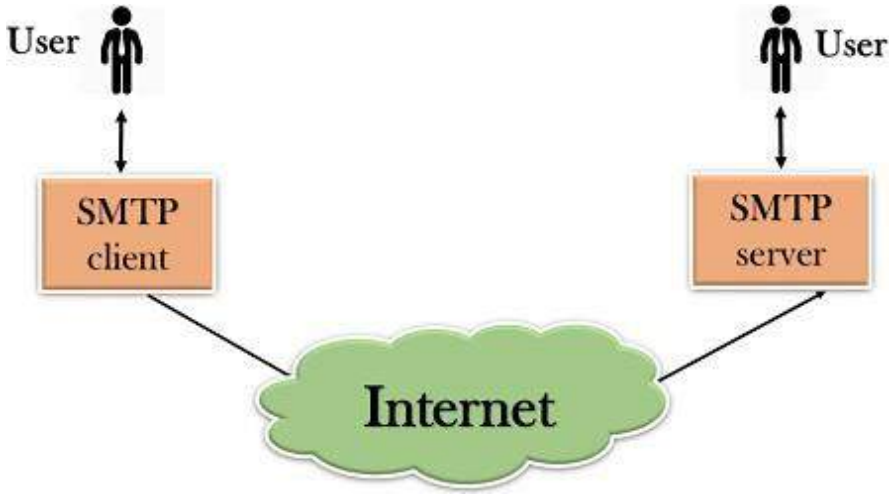
The message body comes after the header. The users can put whatever they want to send in the message body. It is possible to terminate the messages with ASCII cartoons, quotations, and political statements.

Message Transfer By SMTP:

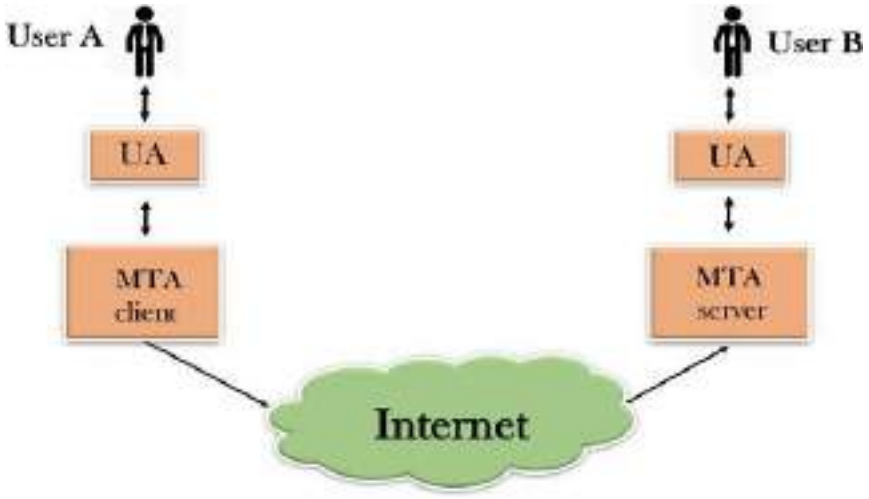
SMTP

- SMTP stands for Simple Mail Transfer Protocol.
- SMTP is a set of communication guidelines that allow software to transmit an electronic mail over the internet is called **Simple Mail Transfer Protocol**.
- It is a program used for sending messages to other computer users based on e-mail addresses.
- It provides a mail exchange between users on the same or different computers, and it also supports:
  - It can send a single message to one or more recipients.
  - Sending message can include text, voice, video or graphics.
  - It can also send the messages on networks outside the internet.
- The main purpose of SMTP is used to set up communication rules between servers. The servers have a way of identifying themselves and announcing what kind of communication they are trying to perform. They also have a way of handling the errors such as incorrect email address. For example, if the recipient address is wrong, then receiving server reply with an error message of some kind.

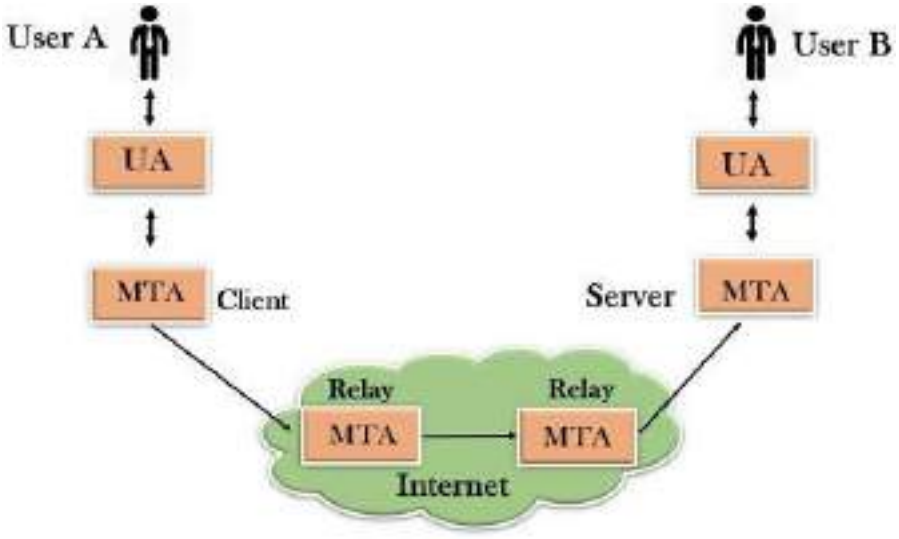
Components of SMTP



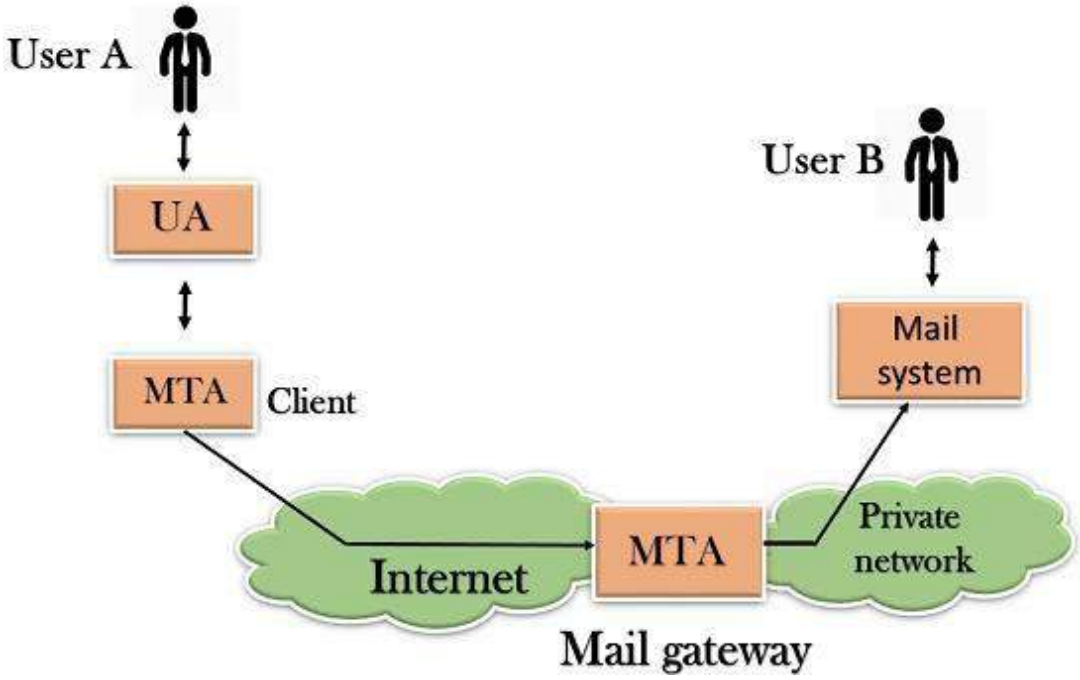
- First, we will break the SMTP client and SMTP server into two components such as user agent (UA) and mail transfer agent (MTA). The user agent (UA) prepares the message, creates the envelope and then puts the message in the envelope. The mail transfer agent (MTA) transfers this mail across the internet.



- SMTP allows a more complex system by adding a relaying system. Instead of just having one MTA at sending side and one at receiving side, more MTAs can be added, acting either as a client or server to relay the email.



- The relaying system without TCP/IP protocol can also be used to send the emails to users, and this is achieved by the use of the mail gateway. The mail gateway is a relay MTA that can be used to receive an email.


User A
UA
MTA
Client
User B
Mail system
Internet
MTA
Private network
Mail gateway

Working of SMTP

1. **Composition of Mail:** A user sends an e-mail by composing an electronic mail message using a Mail User Agent (MUA). Mail User Agent is a program which is used to send and receive mail. The message contains two parts: body and header. The body is the main part of the message while the header includes information such as the sender and recipient address. The header also includes descriptive information such as the subject of the message. In this case, the message body is like a letter and header is like an envelope that contains the recipient's address.
2. **Submission of Mail:** After composing an email, the mail client then submits the completed e-mail to the SMTP server by using SMTP on TCP port 25.
3. **Delivery of Mail:** E-mail addresses contain two parts: username of the recipient and domain name. For example, vivek@gmail.com, where "vivek" is the username of the recipient and "gmail.com" is the domain name. If the domain name of the recipient's email address is different from the sender's domain name, then MSA will send the mail to the Mail Transfer Agent (MTA). To relay the email, the MTA will find the target domain. It checks the MX record from Domain Name System to obtain the target domain. The MX record contains the domain name and IP address of the recipient's domain. Once the record is located, MTA connects to the exchange server to relay the message.
4. **Receipt and Processing of Mail:** Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the e-mail where it waits for the user to retrieve it.
5. **Access and Retrieval of Mail:** The stored email in MDA can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.

World Wide Web:

The **World Wide Web** is abbreviated as WWW and is commonly known as the web. The WWW was initiated by CERN (European library for Nuclear Research) in 1989.

It mainly consists of a worldwide collection of electronic documents (i.e, Web Pages). It is a project created, by Timothy Berner Lee in 1989, for researchers to work together effectively at CERN. is an organization, named the World Wide Web Consortium (W3C), which was developed for further development of the web.

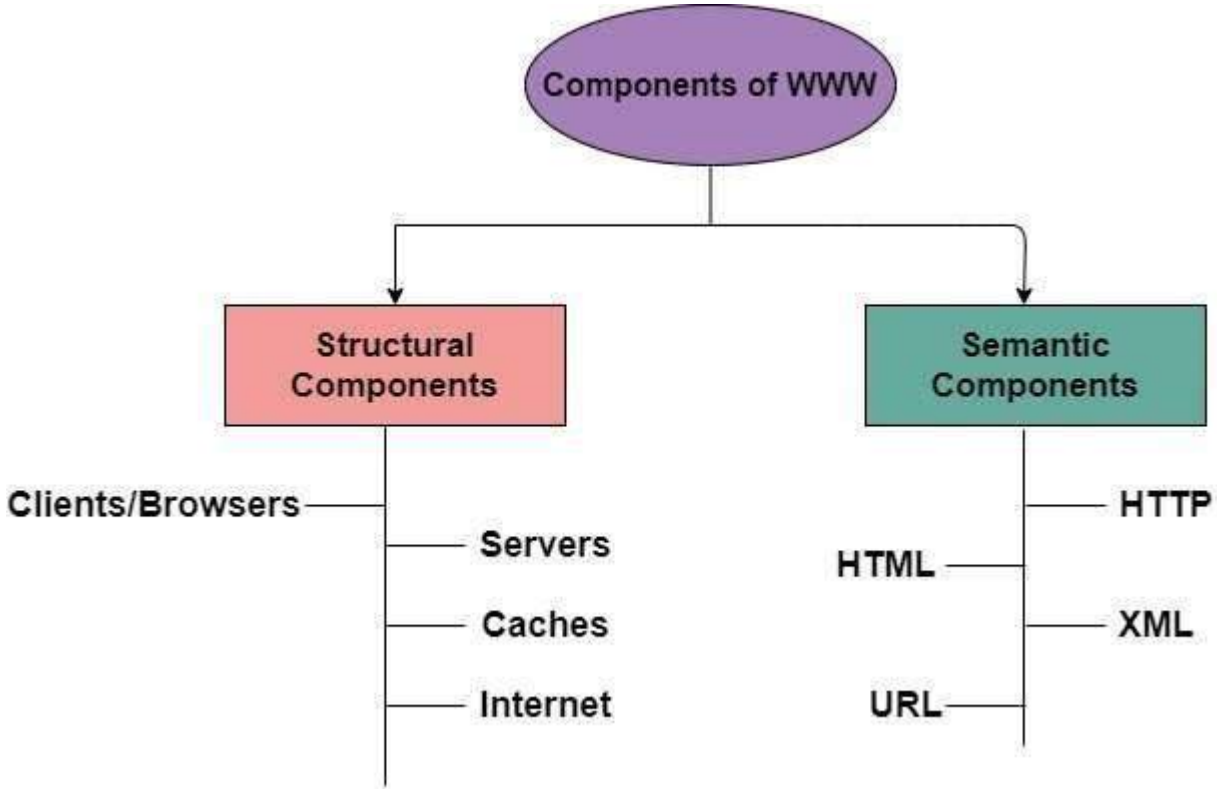
This organization is directed by Tim Berner's Lee, the father of the web.

- World wide web provides flexibility, portability, and user-friendly features.
- It is basically a way of exchanging information between computers on the Internet.
- The WWW is mainly the network of pages consists of images, text, and sounds on the Internet which can be simply viewed on the browser by using the browser software.

Components of WWW

The Components of WWW mainly falls into two categories:

1. Structural Components
2. Semantic Components

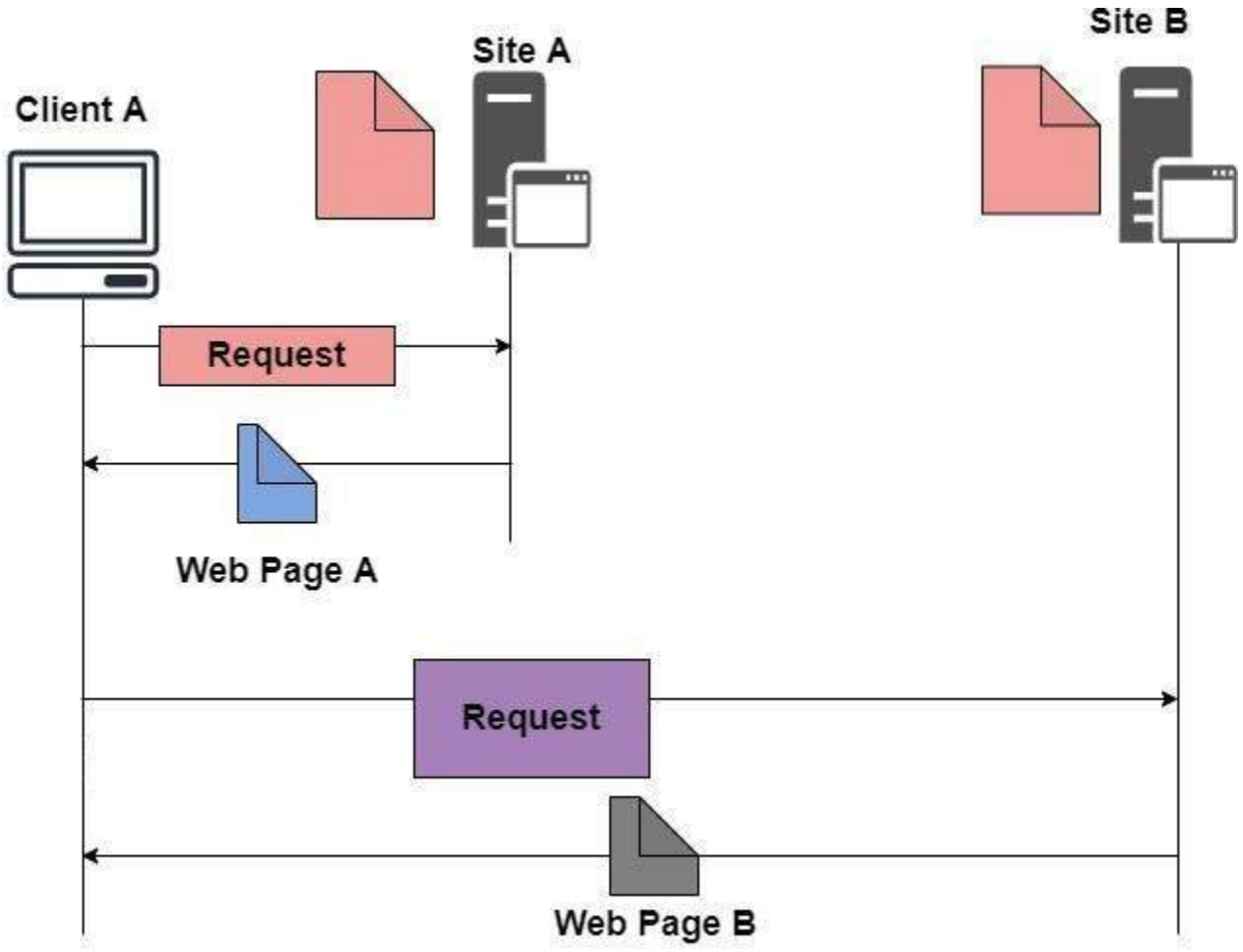


Architecture of WWW

The **WWW** is mainly a distributed **client/server** service where a client using the browser can access the service using a server. The Service that is provided is distributed over many different locations commonly known as **sites/websites.**

- Each website holds one or more documents that are generally referred to as **web pages.**

- Where each web page contains a link to other pages on the same site or at other sites.
- These pages can be retrieved and viewed by using browsers.



In the above case, the client sends some information that belongs to **site A**. It generally sends a request through its browser (It is a program that is used to fetch the documents on the web).

and also the request generally contains other information like the address of the site, web page(URL).

The server at **site A** finds the document then sends it to the client. after that when the user or say the client finds the reference to another document that includes the web page at **site B**.

The reference generally contains the URL of site B. And the client is interested to take a look at this document too. Then after the client sends the request to the new site and then the new page is retrieved.

Now we will cover the components of WWW in detail.

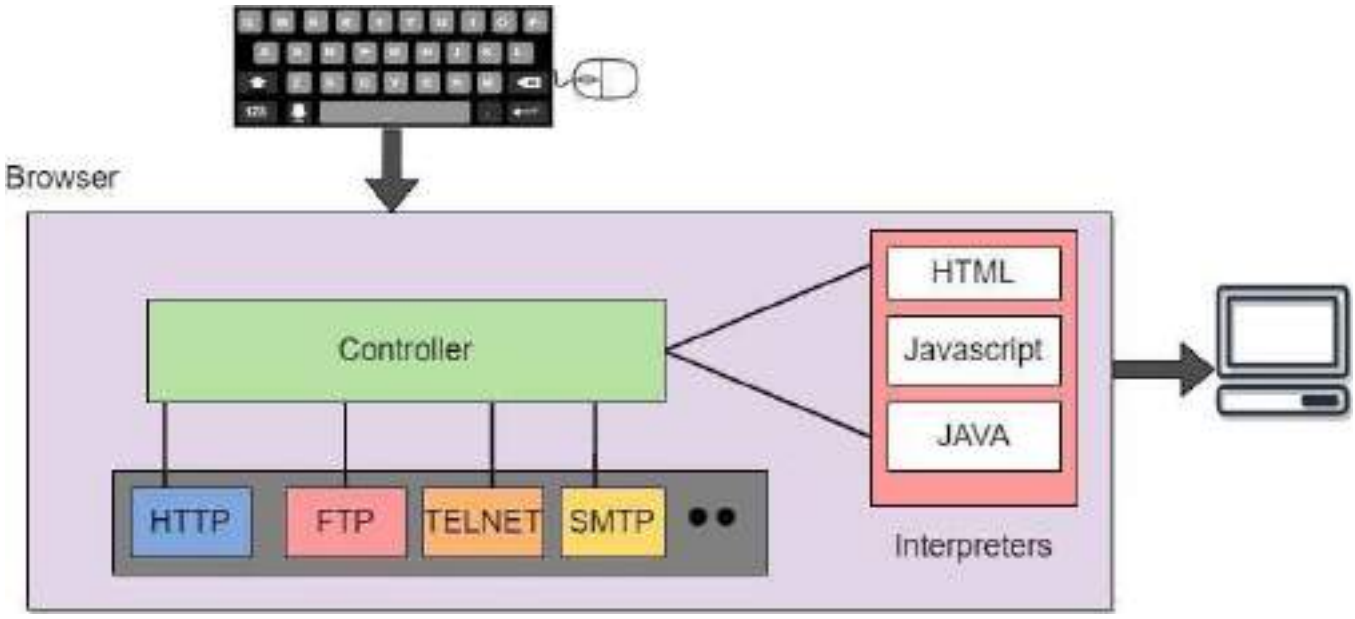
1. Client/Browser

The Client/Web browser is basically a program that is used to communicate with the webserver on the Internet.

Each browser mainly comprises of three components and these are

- Controller
- Interpreter
- Client Protocols

- The Controller mainly receives the input from the input device, after that it uses the client programs in order to access the documents.
- After accessing the document, the controller makes use of an interpreter in order to display the document on the screen.
- An interpreter can be Java, HTML, javascript mainly depending upon the type of the document.
- The Client protocol can be FTP, HTTP, TELNET.



2. Server

The Computer that is mainly available for the network resources and in order to provide services to the other computer upon request is generally known as the **server.**

- The Web pages are mainly stored on the server.
- Whenever the request of the client arrives then the corresponding document is sent to the client.
- The connection between the client and the server is TCP.
- It can become more efficient through multithreading or multiprocessing. Because in this case, the server can answer more than one request at a time.

3. URL

URL is an abbreviation of **the Uniform resource locator**

- It is basically a standard used for specifying any kind of information on the Internet.
- In order to access any page the client generally needs an address.
- To facilitate the access of the documents throughout the world HTTP generally makes use of Locators.

URL mainly defines the four things:

Protocol It is a client/server program that is mainly used to retrieve the document. A commonly used protocol is HTTP.

Host Computer It is the computer on which the information is located. It is not mandatory because it is the name given to any computer that hosts the web page.

Port The URL can optionally contain the port number of the server. If the port number is included then it is generally inserted in between the host and path and is generally separated from the host by the colon.

Path It indicates the pathn ame of the file where the information is located.

- www is addressable by uniform resource locator(URL).
- Ex: http://www.facebook.com

4. HTML

HTML is an abbreviation of Hypertext Markup Language

- It is generally used for creating web pages.
- It is mainly used to define the contents, structure, and organization of the web page.

5. XML

XML is an abbreviation of Extensible Markup Language. It mainly helps in order to define the common syntax in the semantic web.

Features of WWW

Given below are some of the features provided by the World Wide Web:

- Provides a system for Hypertext information
- Open standards and Open source
- Distributed.
- Mainly makes the use of Web Browser in order to provide a single interface for many services.
- Dynamic
- Interactive
- Cross-Platform

Advantages of WWW

Given below are the benefits offered by WWW:

- It mainly provides all the information for Free.
- Provides rapid Interactive way of Communication.
- It is accessible from anywhere.
- It has become the Global source of media.
- It mainly facilitates the exchange of a huge volume of data.

Disadvantages of WWW

There are some drawbacks of the WWW and these are as follows;

- It is difficult to prioritize and filter some information.
- There is no guarantee of finding what one person is looking for.
- There occurs some danger in case of overload of Information.
- There is no quality control over the available data.
- There is no regulation.

Hyper Text Transfer Protocol (HTTP):

- HTTP stands for **Hyper Text Transfer Protocol**.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
- HTTP is used to carry the data in the form of MIME-like format.
- HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

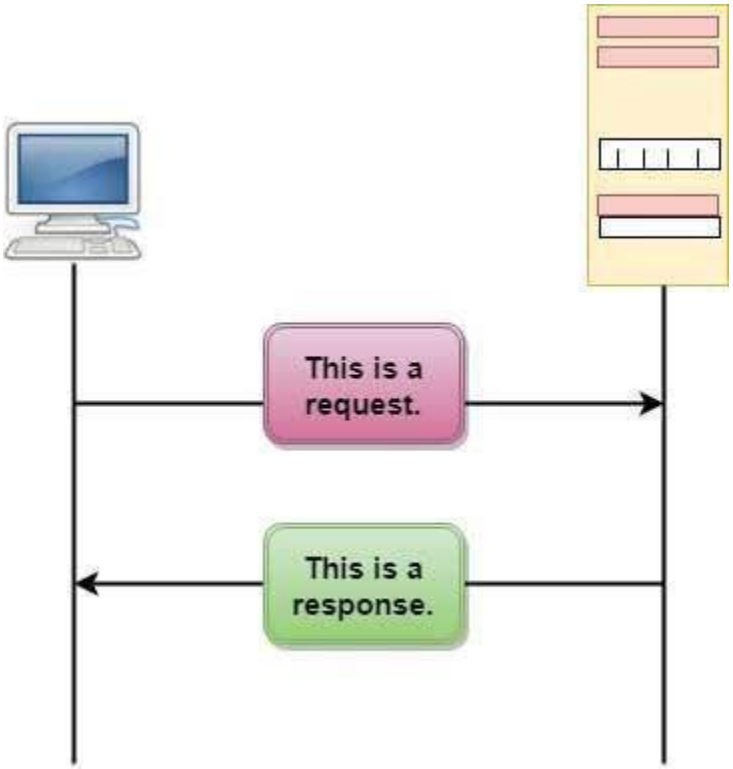
Features of HTTP:

- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection

between client and server exist only during the current request and response time only.

- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.
- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

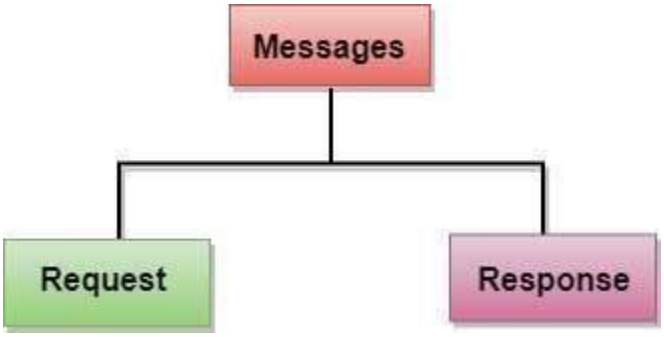
HTTP Transactions



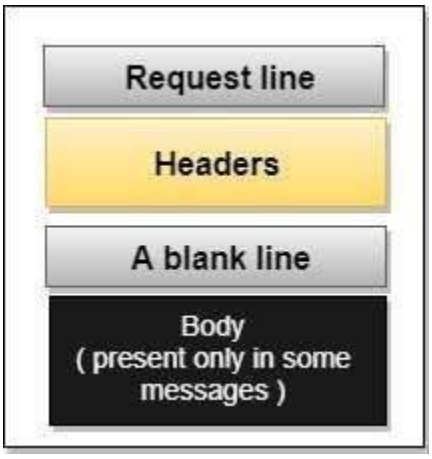
The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

Messages

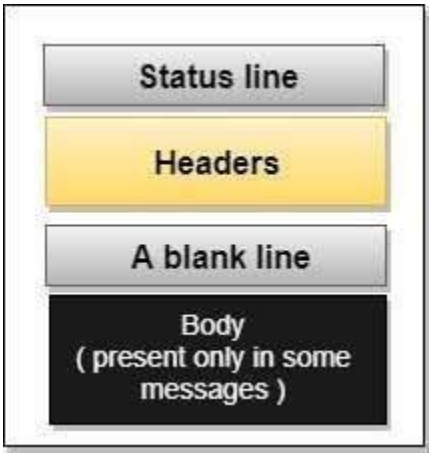
HTTP messages are of two types: request and response. Both the message types follow the same message format.



Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.



Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.



Uniform Resource Locator (URL)

- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL defines four parts: method, host computer, port, and path.

- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.

- **Path:** Path is the pathname of the file where the information is stored. The path itself contain slashes that separate the directories from the subdirectories and files

Content delivery Network or Content Distribution Network:

Over the last few years there has been a huge increase in the number of Internet users. YouTube alone has 2 Billion users worldwide, while Netflix has over 160 million users. Streaming content to such a wide demographic of users is no easy task. One can think that a straightforward approach to this can be building a large data center, storing all the content in the servers, and provide it to the users worldwide. But there are issues that arise when this approach is followed-

1. Firstly if the data center is in the USA and the user is in India there will be slower delivery of content.
2. Secondly, a single data center represents a single point of failure.
3. Thirdly, if some content is being accessed frequently from a remote area then it is likely to follow the same links, and this, in turn, results in wastage of bandwidth.

CDN – Content Distribution Network or Content Delivery Network is a solution that provides faster delivery of content to the users distributed worldwide.

What is a CDN?

A CDN is essentially a group of servers that are strategically placed across the globe with the purpose of accelerating the delivery of web content. A CDN-

1. Manages servers that are geographically distributed over different locations.
2. Stores the web content in its servers.
3. Attempts to direct each user to a server that is part of the CDN so as to deliver content quickly.

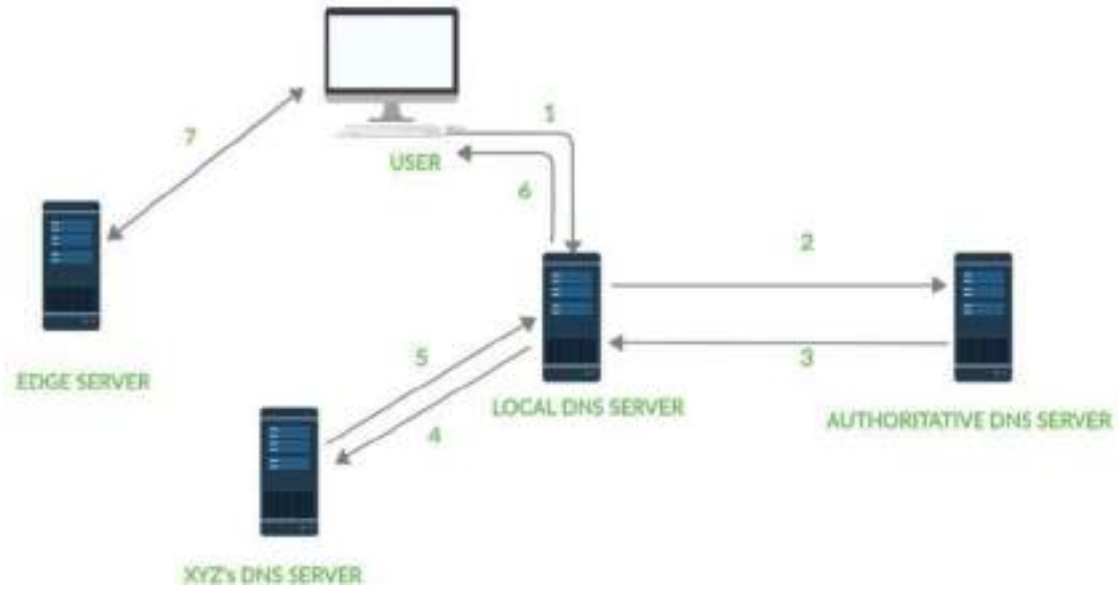
How does CDN work?

To minimize the distance between the visitors and your website's server, a CDN stores a cached version of original content in multiple geographical locations (a.k.a., points of presence/ PoPs). Each PoP contains a number of caching servers known as edge servers that are responsible for content delivery to visitors within its proximity. CDN caches content in many places at once, ensuring quick delivery of content.

Let's consider an example:

Suppose you are hosting a website, wherein your origin server(server containing the primary source of your website's data, where website files are hosted) is

located in Australia and a company XYZ provides you the CDN service.
When a user in India clicks on a video on your website, the request goes to the user's local DNS server(See DNS), which relays the request to the authoritative DNS server of your website.
The authoritative DNS server then identifies that the user is situated far away and therefore relays the request to its XYZ's DNS server. Now the DNS query enters XYZ's network which provides the address of the edge server that is closest to the user to the Local DNS server. The video is delivered by this edge server.
From this point onwards the local DNS server knows the address of the edge server. So whenever users within its network send a request for content from your website, the local DNS server shall relay the request to the edge server.
CDN thus minimizes the number of hops required to deliver the data to a user's browser due to the POPs that are located near the user.
Following image depicts the same:



Following Image depicts the difference between how a request is handled with and without a CDN respectively:

WITH CDN(2 SECONDS)

WITHOUT CDN(5 SECONDS)

Multimedia:

Multimedia is a set of more than one media element used to produce a concrete and more structured communication method. In other words, multimedia is the simultaneoususe of data from different sources. These sources in multimedia are known as media elements.

Multimedia is the holy grail of networking. It brings vast technical challenges in providing (interactive) video on demand to every home and equally immense profits out of it. Multimedia is just two or more media. Generally, the term multimedia means the combination of two or more continuous media. In practice, the two media usually are audio and video.

Elements of Multimedia

There are many types of multimedia components. These are audio, video, pictures, animations, etc.

Text

The inclusion of textual information in multimedia is essential for developing multimedia software. Text can be of any type, maybe a word, a single line, or a paragraph. The textual data for multimedia can be developed using any text editor. The text can have a different type, size, color and style to suit the multimedia software's professional requirement.

Graphics

Another exciting element in multimedia is graphics. Considering human nature, a subject is more explained with some pictorial/graphical representation rather than as a large chunk of text. This also helps to develop a clean multimedia screen, whereas the use of a large amount of text in a screen makes it dull in presentation.

Animation

Moving images have an overpowering effect on the human peripheral vision. The following points are for the popularity of animation.

- Animation is a set of static states related to each other with the transition.
- It can indicate dimensionality in transitions.
- It can illustrate change over time.
- It is used to multiplex the display.
- It can enrich graphical representations.
- It can be visualizing three-dimensional structures.

Audio

The representation, processing, storage and transmission of audio signals are a major part of the study of multimedia systems. The frequency range of the human ear runs from 20 Hz to 20K Hz. The ear is compassionate to sound variations lasting only a few milliseconds. The eye, in contrast, does not notice changes in light level lasting only a few milliseconds.

Video

The human eye has the property when an image is flashed on the retina. It is retained

for a few milliseconds before decaying. If a sequence of images is flashed at 50 or more images/sec, the eye does not notice that it is looking at discrete images. All TV systems exploit this property to produce moving pictures.

Video is used for the following:

- Promoting television shows, films, or other non-computer media that traditionally have used trailers in their advertising.
- Giving users an impression of a speaker's personality.
- Showing things that move. For example, a clip from a motion picture. Product demos of physical products are also well suited for video.

Audio and video:

Recent advances in technology have changed our use of audio and video. In the past, we listened to an audio broadcast through a radio and watched a video program broadcast through a TV. We used the telephone network to interactively communicate with another party. But times have changed. People want to use the Internet not only for text and image communications, but also for audio and video services.

We can divide audio and video services into three broad categories:

1. Streaming stored audio/video
2. Streaming live audio/video
3. Interactive audio/video

Figure 5.17 Internet audio/video

1. **Streaming stored audio/video**, the files are compressed and stored on a server.A clientdownloads the files through the Internet.

2. **Streaming live audio/video**, a user listens to broadcast audio and video throughthe Internet.

3. **Interactive audio/video**, people use the Internet to interactively

communicatewith oneanother.

1. Digitizing Audio and Video

Before audio or video signals can be sent on the Internet, they need to be digitized.

Digitizing Audio

When sound is fed into a microphone, an electronic analog signal is generated which represents the sound amplitude as a function of time. The signal is called an analog audio signal. An analog signal, such as audio, can be digitized to produce a digital signal. According to the Nyquist theorem, if the highest frequency of the signal is f, we need to sample the signal 21 times per second. There are other methods for digitizing an audio signal, but the principle is the same.

Digitizing Video

A video consists of a sequence of frames. If the frames are displayed on the screen fast enough, we get an impression of motion. The reason is that our eyes cannot distinguish the rapidly flashing frames as individual ones. There is no standard number of frames per second; in North America 25 frames per second is common. However, to avoid a condition known as flickering, a frame needs to be refreshed. The TV industry repaints each frame twice. This means 50 frames need to be sent, or if there is memory at the sender site, 25 frames with each frame repainted from the memory.

2. Audio and Video Compression

To send audio or video over the Internet requires compression

Audio Compression

Audio compression can be used for speech or music. For speech, we need to compress a 64-kHz digitized signal; for music, we need to compress a 1.41 I-MHz signal. Two categories of techniques are used for audio compression: predictive encoding and perceptual encoding.

a. Predictive Encoding

In predictive encoding, the differences between the samples are encoded instead of encoding all the sampled values. This type of compression is normally used for speech. Several standards have been defined such as GSM (13 kbps), G.729 (8 kbps), and G.723.3 (6.4 or 5.3 kbps).

b. Perceptual Encoding: MP3

The most common compression technique that is used to create CD-quality audio is based on the perceptual encoding technique. As we mentioned before, this type of audio needs at least 1.411 Mbps; this cannot be sent over the Internet without compression. MP3 (MPEG audio layer 3), a part of the MPEG standard (discussed in the video compression section), uses this technique.

Video Compression

As we mentioned before, video is composed of multiple frames. Each frame is one image. We can compress video by first compressing images. Two standards are prevalent in the market. Joint Photographic Experts Group (JPEG) is used to compress images. Moving Picture Experts Group (MPEG) is used to compress video.

a. Image Compression: JPEG

If the picture is not in color (gray scale) then each pixel can be represented by an 8-bit integer (256 levels). If the picture is in color, each pixel can be represented by 24 bits (3 x 8 bits), with each 8 bits representing red, blue, or green (RBG). To simplify the discussion, we concentrate on a gray scale picture.

b. Video Compression: MPEG

The Moving Picture Experts Group method is used to compress video. In principle, a motion picture is a rapid flow of a set of frames, where each frame is an image. In other words, a frame is a spatial combination of pixels, and a video is a temporal

combination of frames that are sent one after another. Compressing video, then, means spatially compressing each frame and temporally compressing a set of frames.

3. Streaming Live Audio/video

Streaming live audio/video is similar to the broadcasting of audio and video by radio and TV stations. Instead of broadcasting to the air, the stations broadcast through the Internet. There are several similarities between streaming stored audio/video and streaming live audio/video. They are both sensitive to delay; neither can accept retransmission. However, there is

a difference. In the first application, the communication is unicast and on-demand. In the second, the communication is multicast and live. Live streaming is better suited to the multicast services of IP and the use of protocols such as UDP and RTP.

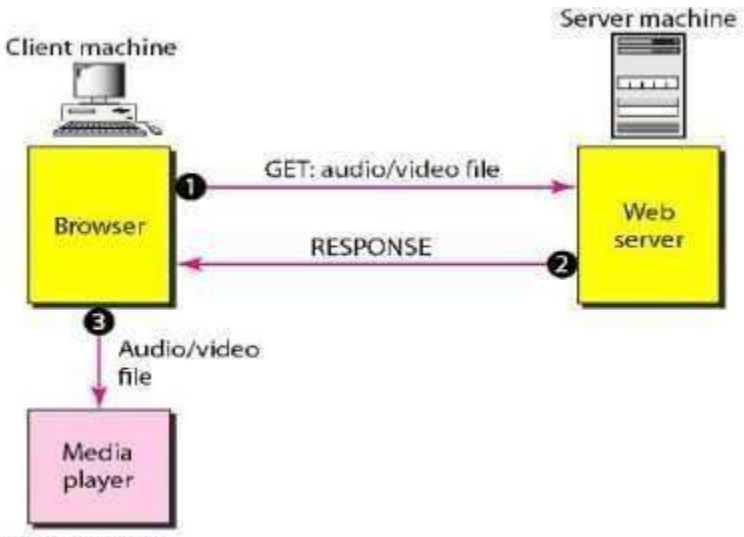


Figure 5.18 Using a Web server

4. Real-Time Interactive Audio/video

In real-time interactive audio/video, people communicate with one another in real time. The Internet phone or voice over IP is an example of this type of application. Video conferencing is another example that allows people to communicate visually and orally.

1. INTRODUCTION

We can divide audio and video services into three broad categories: **streaming stored audio/video, streaming live audio/video,** and **interactive audio/video**. Streaming means a user can listen (or watch) the file after the downloading has started.

In the first category, streaming stored audio/video, the files are compressed and stored on a server. A client downloads the files through the Internet. This is sometimes referred to as on-demand audio/video. In the second category, streaming live audio/video refers to the broadcasting of radio and TV programs through the Internet. In the third category, interactive audio/video refers to the use of the Internet for interactive audio/video applications. A good example of this application is Internet telephony and Internet teleconferencing.

2. STREAMING STORED AUDIO/VIDEO

Downloading these types of files from a server can be different from downloading other types of files.

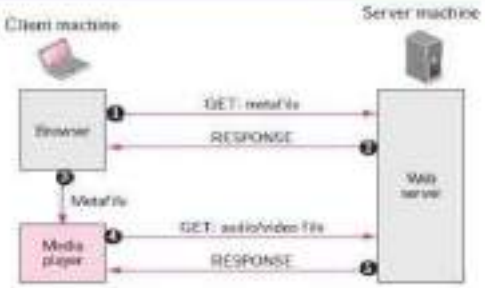
2.1. First Approach: Using a Web Server

A compressed audio/video file can be downloaded as a text file. The **client (browser)** can use the services of **HTTP** and send a GET message to download the file. The **Web server** can send the compressed file to the browser. The browser can then use a help application, normally called a **media player**, to play the file. The file needs to download completely before it can be played.

2.2. Second Approach: Using a Web Server with Metafile

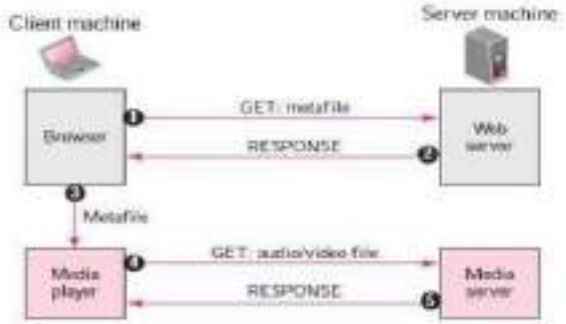
In another approach, the media player is directly connected to the Web server for downloading the audio/video file. The Web server stores two files: **the actual audio/video file** and a **metafile** that holds information about the audio/video file.

1. The HTTP client accesses the Web server using the GET message.
2. The information about the metafile comes in the response.
3. The metafile is passed to the media player.
4. The media player uses the URL in the metafile to access the audio/video file.
5. The Web server responds.



2.3. Third Approach: Using a Media Server

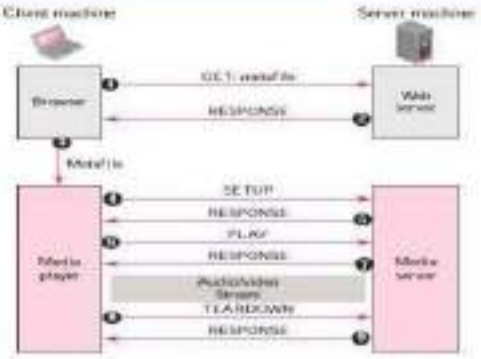
The problem with the second approach is that **the browser and the media player both use the services of HTTP**. HTTP is designed to run over TCP. This is appropriate for retrieving the metafile, but not for retrieving the audio/video file. **The reason is that TCP retransmits a lost or damaged segment, which is counter to the philosophy of streaming**. We need to dismiss TCP and its error control; we need to use UDP. However, HTTP, which accesses the Web server, and the Web server itself are designed for TCP; we need another server, a **media server.**



1. The HTTP client accesses the Web server using a GET message.
2. The information about the metafile comes in the response.
3. The metafile is passed to the media player.
4. The media player uses the URL in the metafile to access the media server to download the file. Downloading can take place by any protocol that uses UDP.
5. The media server responds.

2.4. Fourth Approach: Using a Media Server and RTSP

The **Real-Time Streaming Protocol (RTSP)** is a control protocol designed to add more functionalities to the streaming process. Using RTSP, we can control the playing of audio/video. Figure 5 shows a media server and RTSP.



1. The HTTP client accesses the Web server using a GET message.
2. The information about the metafile comes in the response.
3. The metafile is passed to the media player.
4. The media player sends a SETUP message to create a connection with the media server.
5. The media server responds.
6. The media player sends a PLAY message to start playing (downloading).
7. The audio/video file is downloaded using another protocol that runs over UDP.
8. The connection is broken using the TEARDOWN message.
9. The media server responds.

3. STREAMING LIVE AUDIO/VIDEO

Streaming live audio/video is similar to the broadcasting of audio and video by radio and TV stations. Instead of broadcasting to the air, the stations broadcast through the Internet. There are several similarities between streaming stored audio/video and streaming live audio/video. They are both sensitive to delay; neither can accept retransmission. However, there is a difference. In the first application, the communication is unicast and on-demand. In the second, the communication is multicast and live. Live streaming is better suited to the multicast services of IP and the use of protocols such as UDP and RTP.

Examples: Internet Radio, Internet Television (ITV), Internet protocol television (IPTV)

4. REAL-TIME INTERACTIVE AUDIO/VIDEO

In real-time interactive audio/video, people communicate with one another in real time. The Internet phone or voice over IP is an example of this type of application. Video conferencing is another example that allows people to communicate visually and orally.

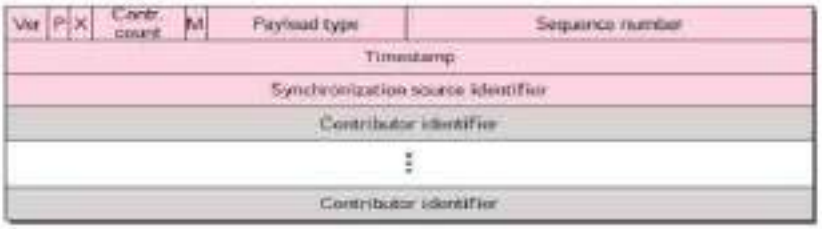
Before discussing the protocols used in this class of applications, we discuss some characteristics of real-time audio/video communication.

5. Real-time Transport Protocol (RTP)

Real-time Transport Protocol (RTP) is the protocol designed to handle real-time traffic on the Internet. RTP does not have a delivery mechanism (multicasting, port numbers, and so on); it must be used with UDP. RTP stands between UDP and the application program. The main contributions of RTP are timestamping, sequencing, and mixing facilities.

5.1. RTP Packet Format

The format is very simple and general enough to cover all real-time applications. An application that needs more information adds it to the beginning of its payload. A description of each field follows.



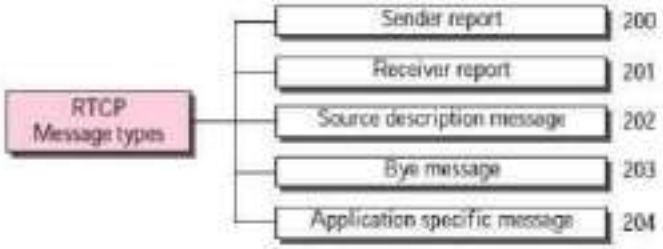
- **Ver.** This 2-bit field defines the version number.
- **P.** This 1-bit field, if set to 1, indicates the presence of padding at the end of the packet. There is no padding if the value of the P field is 0.
- **X.** This 1-bit field, if set to 1, indicates an extra extension header between the basic header and the data. There is no extra extension header if the value of this field is 0.
- **Contributor count.** This 4-bit field indicates the number of contributors. Note that we can have a maximum of 15 contributors because a 4-bit field only allows a number between 0 and 15.
- **M.** This 1-bit field is a marker used by the application to indicate, for example, the end of its data.
- **Payload type.** This 7-bit field indicates the type of the payload. Several payload types have been defined so far.
- **Sequence number.** This field is 16 bits in length. It is used to number the RTP packets. The sequence number of the first packet is chosen randomly; it is incremented by 1 for each subsequent packet. The sequence number is used by the receiver to detect lost or out of order packets.
- **Timestamp.** This is a 32-bit field that indicates the time relationship between packets.
- **Synchronization source identifier.** If there is only one source, this 32-bit field defines the source. However, if there are several sources, the mixer is the synchronization source and the other sources are contributors. The value of the source identifier is a random number chosen by the source. The protocol provides a strategy in case of conflict (two sources start with the same sequence number).
- **Contributor identifier.** Each of these 32-bit identifiers (a maximum of 15) defines a source. When there is more than one source in a session, the mixer is the synchronization source and the remaining sources are the contributors.

5.2. UDP Port

Although RTP is itself a transport layer protocol, the RTP packet is not encapsulated directly in an IP datagram. Instead, RTP is treated like an application program and is encapsulated in a UDP user datagram. However, unlike other application programs, **no well-known port is assigned to RTP.** The port can be selected on demand with only one restriction: The port number must be an even number. The next number (an odd number) is used by the companion of RTP, Real-Time Transport Control Protocol (RTCP).

6. Real-Time Transport Control Protocol (RTCP)

RTP allows only one type of message, one that carries data from the source to the destination. In many cases, there is a need for other messages in a session. **These messages control the flow and quality of data and allow the recipient to send feedback to the source or sources. Real-Time Transport Control Protocol (RTCP)** is a protocol designed for this purpose. RTCP has five types of messages, as shown in the following Figure. The number next to each box defines the type of the message.



6.1. Types of messages

- Sender Report:
  The sender report is sent periodically by the active senders in a conference to report transmission and reception statistics for all RTP packets sent during the interval.
- Receiver Report:
  The receiver report is for passive participants, those that do not send RTP packets. The report informs the sender and other receivers about the quality of service.
- Source Description Message:
  The source periodically sends a source description message to give additional information about itself.
- Bye Message:
  A source sends a bye message to shut down a stream. It allows the source to announce that it is leaving the conference.
- Application-Specific Message
  The application-specific message is a packet for an application that wants to use new applications (not defined in the standard). It allows the definition of a new message type.

6.2. UDP Port

RTCP, like RTP, does not use a well-known UDP port. It uses a temporary port. The UDP port chosen must be the number immediately following the UDP port selected for RTP, which makes it an odd-numbered port.

7. VOICE OVER IP (Real-time interactive audio/video application)

The idea is to use the Internet as a telephone network with some additional capabilities. Instead of communicating over a circuit-switched network, this application allows communication between two parties over the packet-switched Internet. Two protocols have been designed to handle this type of communication: SIP(Session Initiation Protocol) and H.323.