COMPUTER NETWORKS

PLACEMENT PREPARATION [EXCLUSIVE NOTES]

SAVE AND SHARE WITH YOUR CONNECTIONS TO HELP THEM

Curated By- **HIMANSHU KUMAR(LINKEDIN)** https://www.linkedin.com/in/himanshukumarmahuri

Introduction To Computer Networks

Computer Network means an interconnection of autonomous (standalone) computers for information exchange. The connecting media could be a copper wire, optical fibre, microwave or satellite.

Networking Elements - The computer network includes the following networking elements:

1. At least two computers
2. Transmission medium either wired or wireless
3. Protocols or rules that govern the communication
4. Network software such as Network Operating System

Network Criteria:

The criteria that have to be met by a computer network are:

1. Performance - It is measured in terms of transit time and response time.

- Transit time is the time for a message to travel from one device to another

- Response time is the elapsed time between an inquiry and a response.

Performance is dependent on the following factors:

- The number of users
- Type of transmission medium
- Capability of connected network
- Efficiency of software

2. Reliability - It is measured in terms of

- Frequency of failure
- Recovery from failures
- Robustness during catastrophe

3. Security - It means protecting data from unauthorized access.

Goals of Computer Networks:

The following are some important goals of computer networks:

1. **Resource Sharing -** Many organization has a substantial number of computers in operations, which are located apart. Ex. A group of office workers can share a common printer, fax, modem, scanner etc.

2. **High Reliability -** If there are alternate sources of supply, all files could be replicated on two or, machines. If one of them is not available, due to hardware failure, the other

copies could be used.

3. **Inter-process Communication -** Network users, located geographically apart, may converse in an interactive session through the network. In order to permit this, the network must provide almost error-free communications.

4. **Flexible access -** Files can be accessed from any computer in the network. The project can be begun on one computer and finished on another.

   Other goals include Distribution of processing functions, Centralized management, and allocation of network resources, Compatibility of dissimilar equipment and software, Good network performance, Scalability, Saving money, Access to remote information, Person to person communication etc.,

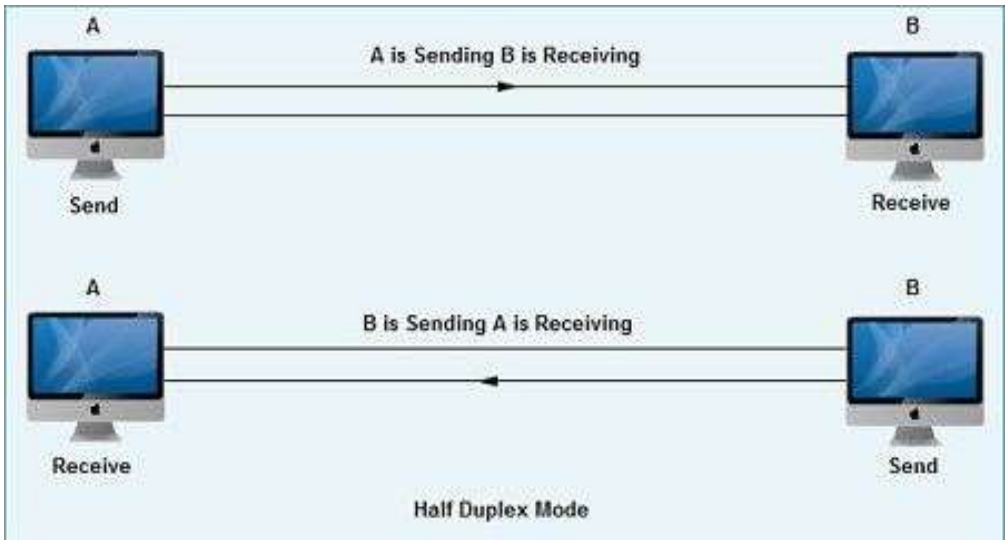
Transmission modes: -

Transmission Modes determine how data is transferred between two devices in a computer network. There are 3 transmission modes in computer networks given below:

Simplex

Communication is uni-directional or one-way in Simplex Mode. i.e. Only one device is allowed to transfer data and the other device simply recieves it. e.g. Radio Station (The station transmits and the radio only recieves).

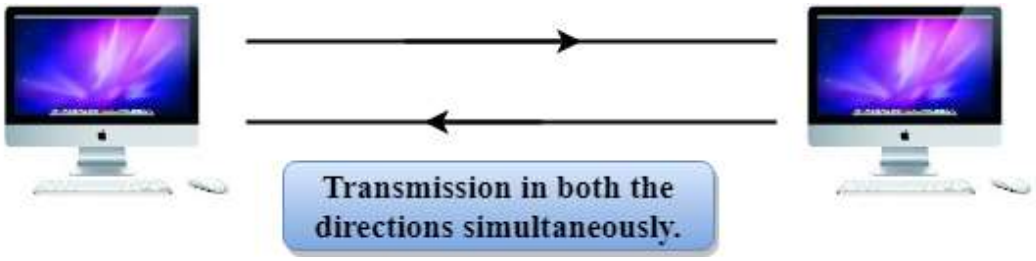
Half-Duplex Communication is possible both-ways but not simultaneously. i.e. Both the devices/stations can transmit and recieve data but not at the same time. At an instant, only communication in a single direction is allowed. e.g. Walkie-Talkie.



Full-Duplex Bidirectional communication is possible simultaneously. This can be possible in the following cases:

- Dedicated seperate channels for transmission and reception
- Capacity is divided between transmission and reception (if single channel is used)

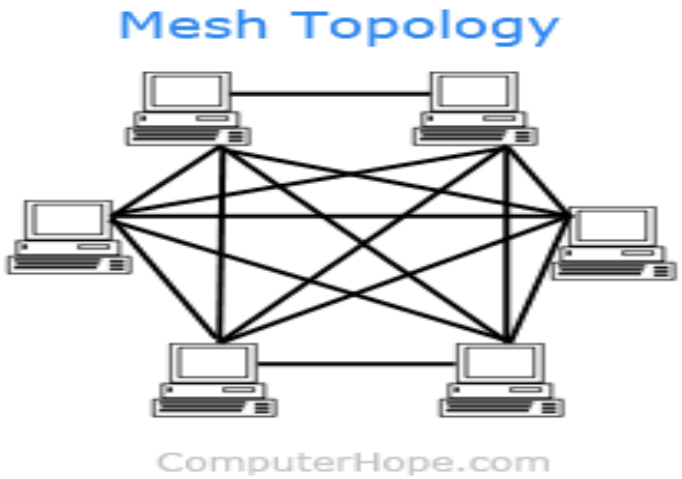
e.g., Cellular/Telephone Network.



Network Topologies: -

The arrangement of nodes in a network generally follows some pattern or organization. Each of these patterns have their set of advantages/disadvantages. Such arrangements are called collectively referred to as network **topologies**. Some of the popular network topologies are as follows:

1- Mesh

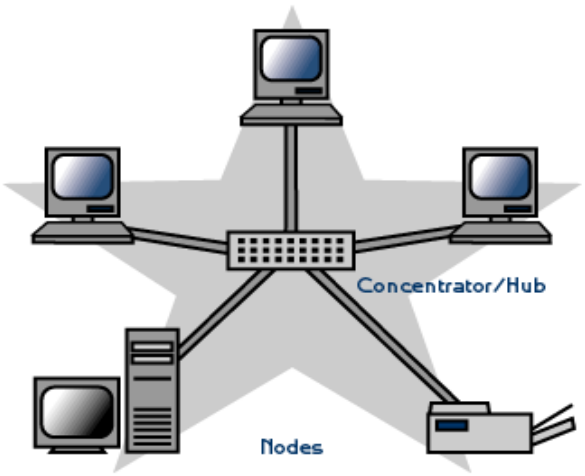


Key points:

1. Robust & Easy fault-detection.

2. Installation is difficult & Expensive (fully-connected ~ lots of cable required = ${}^{n}C_{2}$).

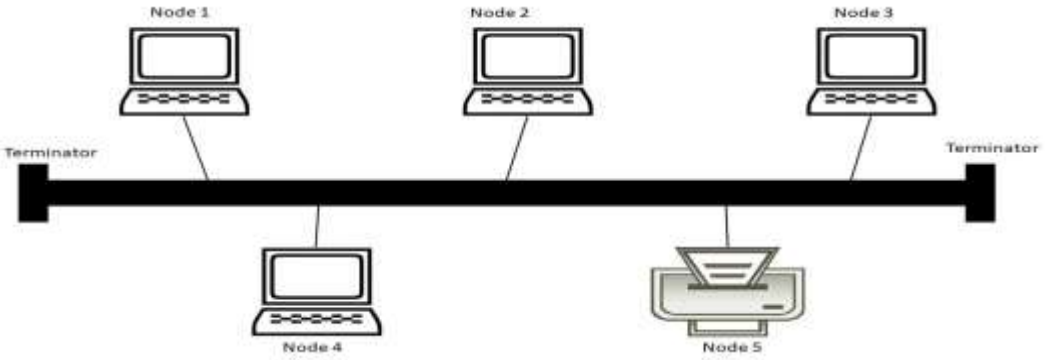
2- Star



Key points:

1. Easy & Cheap Installation (n cables required). Also device needs to have only 1 port.
2. Single point-of-failure (central node).

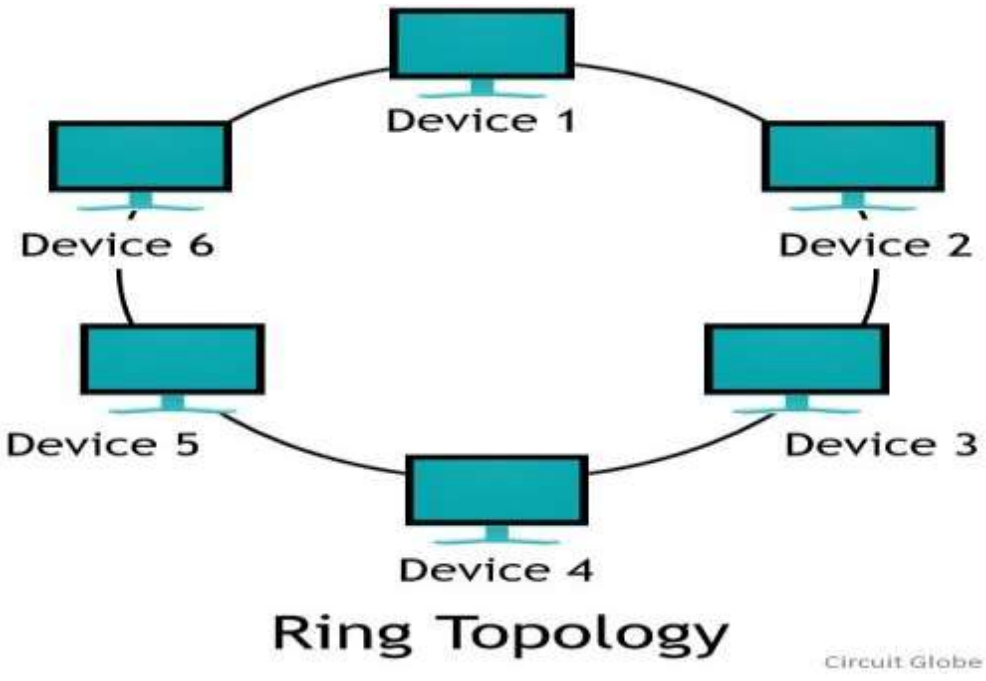
3- Bus



Key points:

1. Easy & Cheap Installation (n + 1(main-line) cables required).
2. Single line-of-failure (main-line).
3. Heavy Traffic causes collisions.

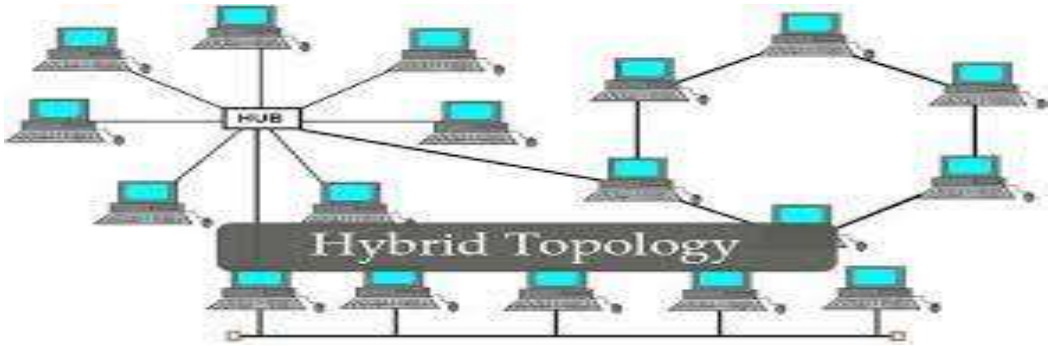
4- Ring



Key points:

1. Easy & Cheap Installation (1 line).
2. Difficulty in Troubleshooting.
3. Addition/Removal of nodes disturbs the topology.

5- Hybrid



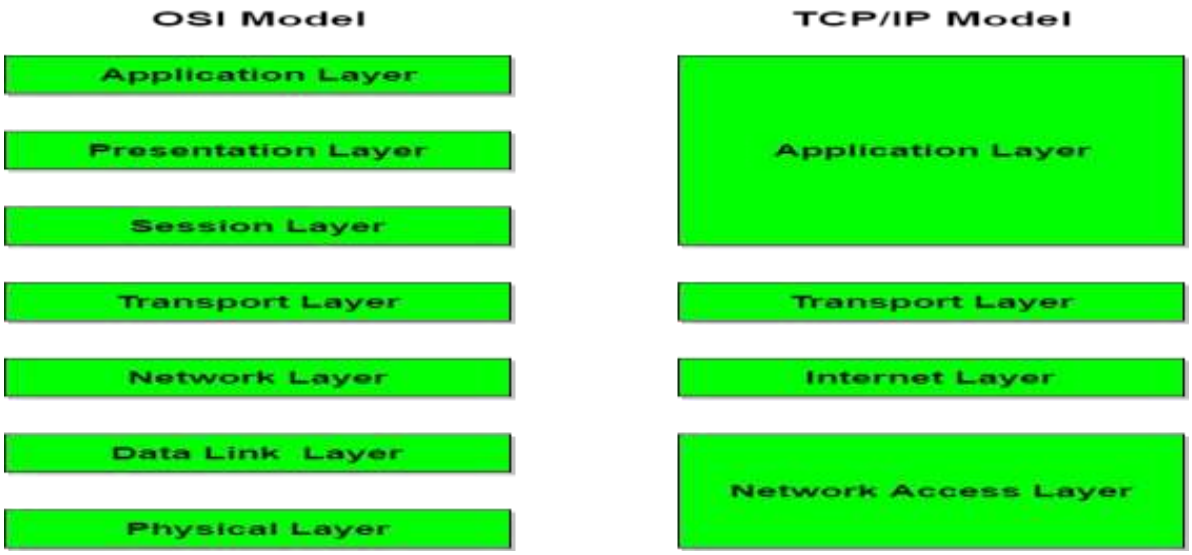
Key points:

1. Combination of all topologies (according to requirement).
2. This kind of topology is scalable and can serve a variety of requirements
3. Due to intermixing of Ring, Bus, Star etc. topologies, it is difficult to develop. (As each of the individual topologies have their own rules and concepts ~ collision detection, protocols for data transfer etc.).

TCP/IP VS OSI

TCP/IP and OSI are reference models which divides the various responsibilities into a logical separation of layers. Practical implementations in devices are based on these 2 popular

models. The difference amongst them lies in the no. of layers and their respective responsibilities:



Brief description of each model:

OSI Model It stands for *Open Systems Interconnection*. It comprises of 7 layers with the following responsibilites (starting from the lowest layer):

- **Physical Layer**: It is responsible for actual physical transmission of data (through channels). It recieves/transmits signals and then converts it to physical bits (0 & 1). It handles *bit-synchronization* (using clock), *bit-rate* control (no.of bits/sec), physical *topology* and *transmission mode* (simplex, half-duplex, full-duplex).
- **Data-link Layer**: It is responsible for *Node-to-Node* delivery of packets, *Framing*, *Error control*, *Flow control*, *Physical Addressing (MAC)*. Upon recieving packets from network layer, it encapsulates it within a

frame with the hardware (MAC) address of the reciever (obtained via *ARP* ~ Address Resolution Protocol).

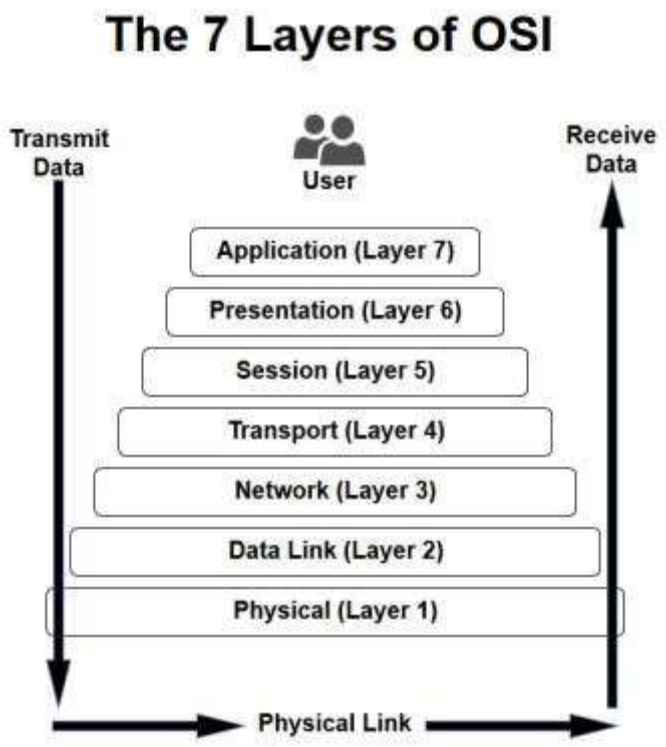
- **Network Layer**: It is responsible for *Logical Addressing* (IPv4/v6) and *Routing*. Various routing algorithms are implemented at this layer, which determines the IP for the next hop in routing.
- **Transport Layer**: It is responsible for *End-to-end* delivery of packets. It also does *Segmentation & Reassembly* of packets (done if packet-size exceeds MTU ~ Max. Transmission Unit). It also does *multiplexing/de-multiplexing* of packets according to the application (using port no.). TCP/UDP (*Connection vs. Connection-less*) protocol is implemented at this layer.
- **Session Layer**: It is responsible for *Session Management* (Establishment, Maintenance, Termination), *Authentication*, *Security*, *Synchronization* & Restoration (check-points are established, such that upon re-connection state is resumed from the last saved point) and *Dialog Control* (synchronization when multiple parties are interacting ~ conference).
- **Presentation Layer**: It is responsible for *Translation* (e.g. ASCII to EBCDIC), *Encryption/Decryption* and *Compression*.
- **Application Layer**: Implements application-specific protocols (HTTP, HTTPS, FTP, SMTP etc.) They produce the data, interacts with the user (input and display of data). e.g. Browsers, Skype, Messaging Apps.

TCP/IP Model

It comprises of 4 layers with the following responsibilities (starting from the lowest layer):

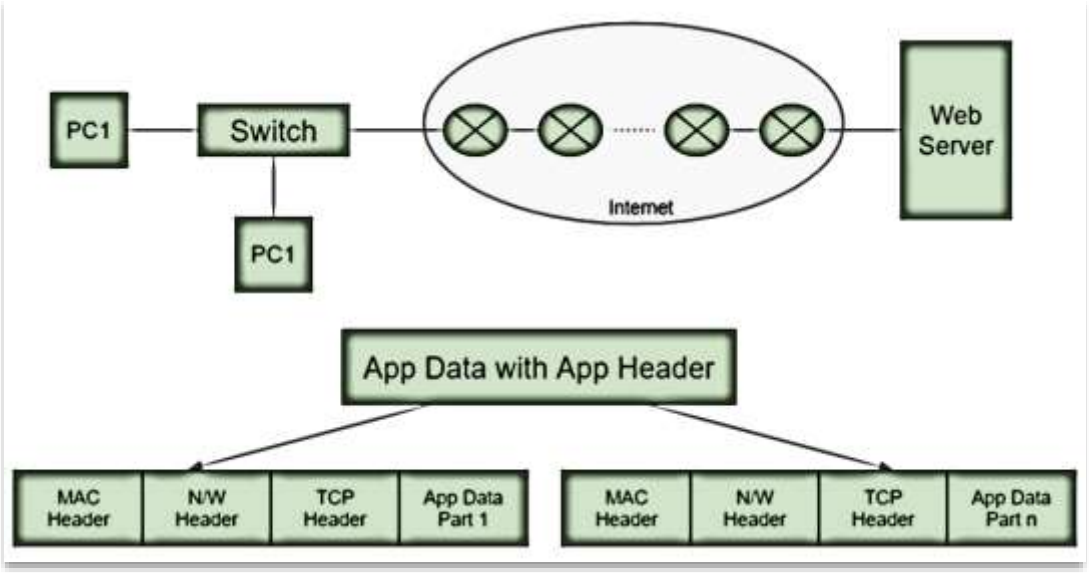
- **Network Access Layer**: It is a combination of the Physical and Data-Link Layer, and is responsible for data transmission and hardware addressing (MAC).
- **Internet Layer**: It is the counterpart of OSI's Network layer, and is responsible for routing and logical addressing. (IP, ICMP, ARP).
- **Transport Layer**: Maintains End-to-end connectivity. It is a counterpart of the OSI's transport layer, and has the same responsibilites (TCP vs. UDP).
- **Application Layer**: Application-specific protocols are implemented here. (HTTP, HTTPS, FTP, SMTP etc.)

TCP/IP	OSI
Implementation of OSI model	Reference model
Model around which Internet is developed	This is a theoretical model
Has only 4 layers	Has 7 layers
Considered more reliable	Considered a reference tool
Protocols are not strictly defined	Stricter boundaries for the protocols
Horizontal approach	Vertical approach
Combines the session and presentation layer in the application layer	Has separate session and presentation layer
Protocols were developed first and then the model was developed	Model was developed before the development of protocols
Supports only connectionless communication in the network layer	Supports connectionless and connection-oriented communication in the network layer
Protocol dependent standard	Protocol independent standard InstrumentationTools.com



APPLICATION LAYER-

The application layer is the topmost layer of the OSI Model. It is the layer in which user applications run. Various protocols run at this layer serving different requirements. Let's understand the working using the below diagram:



When someone browses the internet, then the browser generates Application Data with specific Header files. Following this, the transport layer breaks the Application data into various parts. To this TCP header is added to each part of the Application Data. Next comes the network layer, which adds the destination address in the network header. Then the link is made between the computer and the ISP routers. All the traffic are being sent to the routers. To send the data over the data link layer to any other routers, the computer uses MAC address where the routers MAC address is used to set the link. This MAC address is known using the ARP or Address Resolution Protocol. The switch reads the MAC header of the destination router. The routers laying on the Internet implements three layers namely network layer, DLL, and physical layer. Now finally when the data reached the webserver then using the TCP header, the webserver combines all the data.

We shall go over in brief over each of the protocols:

- **HTTP:** Stands for **H**yper **T**ext **T**ransfer **P**rotocol. It is a request-response protocol that is used to receive web-pages on a client-server architecture. The client requests for a resource (HTML page, javascript file, images or any other file) to the server. The server returns a response accordingly. It uses TCP as the underlying Transport Layer protocol. HTTP supports the following methods (modes) of requests:

0. *GET* - Retrieve information from the server (GET Requests is intended only for data fetching and should not have any side-effect).
1. *HEAD* - Retrieve only header (meta-information) and no response-body.
2. *POST* - Post/Send data back to the server. e.g. User-data, form-data.
3. *DELETE* - Delete the specified resource.
4. *OPTIONS* - Returns the list of HTTP methods supported by the server.

Port no. for HTTP: 80 (8080 occasionally)

- **HTTPS**: It is a secured version of HTTP made possible by encrypting the data transferred using TLS (Transport-Layer-Security). Earlier, its predecessor ~ SSL was used. The use of HTTPs over HTTP increases security by preventing eavesdropping, tampering and man-in-the-middle attacks. Port used in HTTPs is the same as that of HTTP.
- **TELNET**: Stands for **TEL**ecommunications **NET**work. It is used in terminal emulation, which one can use to access a remote system. It is used for file-access and for the initial setup of switches. One may draw it's similarities to SSH (Secure-Shell), but as the name suggests SSH is secure as it uses encryption in addition to normal terminal emulation. Telnet is thus no longer used thanks to SSH.
  Port no. for Telnet: 23
  Port no. for SSH: 22
- **FTP**: Stands for **F**ile **T**ransfer **P**rotocol. It provides reliable and efficient file-transfer between two remote machines.

Port no. for FTP Data: 20
Port no. for FTP Control: 21s

- **SMTP**: Stands for **S**imple **M**ail **T**ransfer **P**rotocol. Uses TCP under the hood. Using a process called “store and forward,” SMTP moves your email on and across networks. It works closely with something called the Mail Transfer Agent (MTA) to send your communication to the right computer and email inbox.
  Port no. for SMTP: 25.
- **DNS**: Stands for **D**omain **N**ame **S**ervice. DNS maps human-addressable english domain names to IP addresses.e.g. www.abc.com might translate to 198.105.232.4. We as humans are comfortable in dealing with named addresses of websites (facebook.com, google.com etc.). However, to uniquely identify the server hosting the application, numeric IP addresses are required by the machine. DNS servers contain this mapping of named addresses to IP addresses. Whenever we us a named address is used, client machine services a request to the DNS server to fetch the IP address.
  Port no. for DNS: 53
- **DHCP**: Stands for **D**ynamic **H**ost **C**onfiguration **P**rotocol. Used for dynamic addressing of devices in a network. DHCP server keeps a pool of available IP addresses. Whenever a new device joins the network, it provides it with an IP from the available pool with an expiration time. DHCP is required in place of static addresses because current requirements involve managing devices which are continuously leaving/joining a network. Thus, a pool of available addresses are required which can be leased to

devices currently residing in the network.
Port no. for DHCP: 67, 68

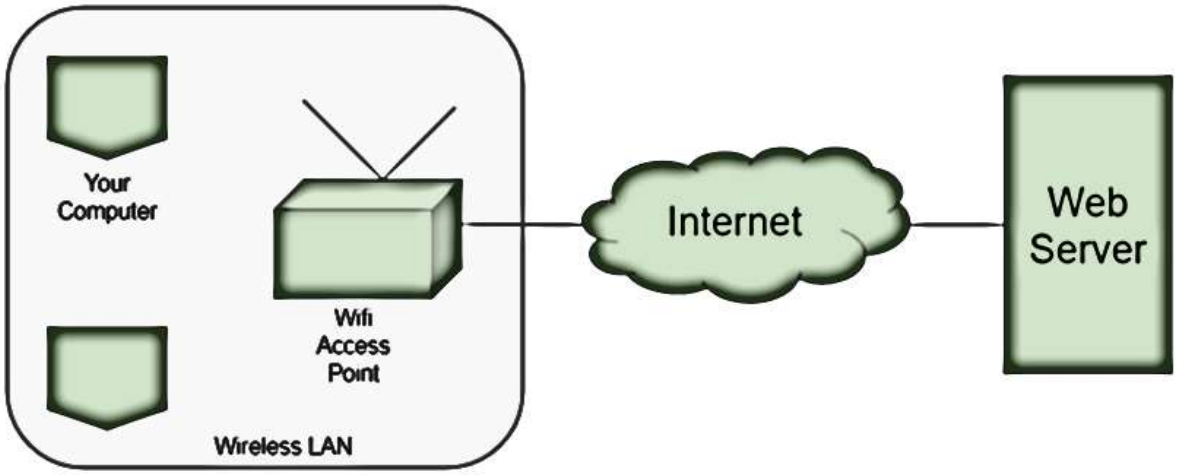
TRANSPORT LAYER: -

Transport Layer is the layer which lies just above the Network layer and is responsible for **end-to-end connectivity**. It is so-called because it provides point-to-point rather than hop-to-hop. The unit of transmission at the transport layer is called segmentation. **TCP** (Transmission Control Protocol), **UDP** (User Datagram Protocol) and **DCCP** (Datagram Congestion Control Protocol) are some of the protocols running in the transport layer. The transport layer also provides the acknowledgement of the successful data transmission and re-transmits the data if an error is found.

At sender's side: Transport layer receives the formatted data from the upper layers, performs **Segmentation** and also implements **Flow & Error control** to ensure proper data transmission. It also adds Source and Destination port number in its header and forwards the segmented data to the Network Layer.
Note: The sender need to know the port number associated with the receiver's application. Generally, this destination port number is configured, either by default or manually. For example, when a web application makes a request to a web server, it typically uses port number 80, because this is the

default port assigned to web applications. Many applications have default port assigned.

At receiver's side: Transport Layer reads the port number from its header and forwards the Data which it has received to the respective application. It also performs sequencing and reassembling of the segmented data.
Let's look at the diagrammatic representation of the working at the transport layer:



Here is a list of few important port numbers and there uses:

PORT number	Use
80	HTTP
443	HTTPS
53	DNS
22	SSH
110	POP3
25	SMTP

Transport Layer has the following responsibilities:

- **Process to process delivery** - While Data Link Layer requires the MAC address (48 bits address contained inside the Network Interface Card of every host machine) of source-destination hosts to correctly deliver a frame and Network layer requires the IP address for appropriate routing of packets , in a similar way Transport Layer requires a Port number to correctly deliver the segments of data to the correct process amongst the multiple processes running on a particular host. A **port number** is a 16 bit address used to identify any client-server program uniquely.

- **End-to-end Connection between hosts -** The transport layer is also responsible for creating the end-to-end Connection between hosts for which it mainly uses TCP and UDP. TCP is a secure, connection- orientated protocol which uses a handshake protocol to establish a robust connection between two end- hosts. TCP ensures reliable delivery of messages and is used in various applications. UDP, on the other hand, is a stateless and unreliable protocol which ensures best-effort delivery. It is suitable for the applications which have little concern with flow or error control and requires to send the bulk of data like video conferencing. It is often used in multicasting protocols.

- **Multiplexing and Demultiplexing -** Multiplexing allows simultaneous use of different applications over a network which is running on a host. The transport layer provides this mechanism which enables us to send packet streams from various applications simultaneously over a network. The transport layer accepts these packets from different processes differentiated by their port numbers and passes them to the network layer after adding proper headers. Similarly, Demultiplexing is required at the receiver side to obtain the data coming from various processes. Transport receives the segments of data from the network layer and delivers it to the appropriate process running on the receiver's machine.

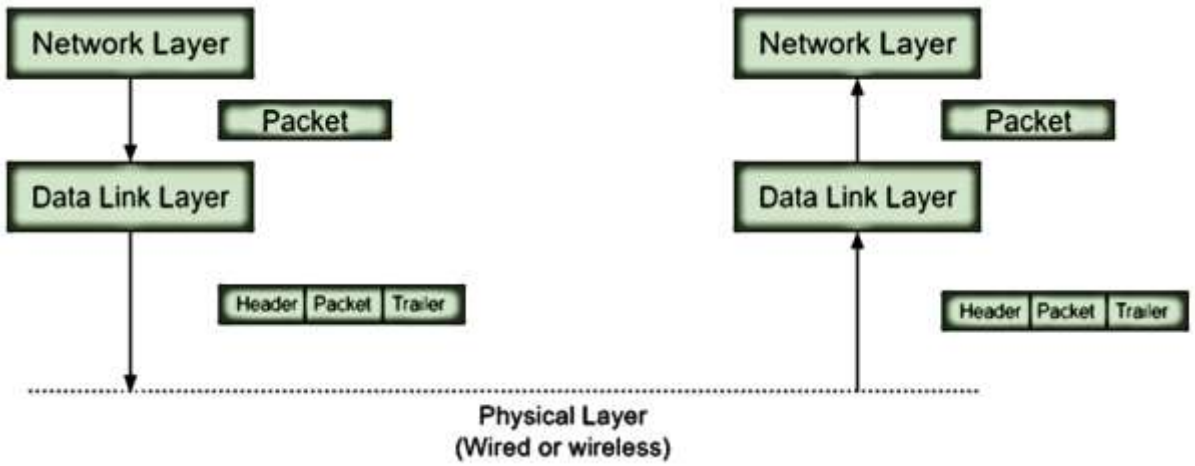
- **Congestion Control -** Congestion is a situation in which too many sources over a network attempt to send data and the router buffers start overflowing due to which loss of packets occur. As a result retransmission of packets from the sources increases the congestion further. In this situation, the Transport layer provides Congestion Control in different ways. It uses **open loop** congestion control to prevent the congestion and **closed loop** congestion control to remove the congestion in a network once it occurred. TCP provides AIMD- additive increase multiplicative decrease, leaky bucket technique for congestion control.

- **Data integrity and Error correction -** Transport layer checks for errors in the messages coming from application layer by using error detection codes, computing checksums, it checks whether the received data is not corrupted and uses the ACK and NACK services to inform the sender if the data has arrived or not and checks for the integrity of data.

- **Flow control -** The transport layer provides a flow control mechanism between the adjacent layers of the TCP/IP model. TCP also prevents data loss due to a fast sender and slow receiver by imposing some flow control techniques. It uses the method of sliding window protocol which is accomplished by the receiver by sending a window back to the sender informing the size of data it can receive.

DATA LINK LAYER: -

The data link layer is responsible for the node to node delivery of the message. The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer. When a packet arrives in a network, it is the responsibility of DLL to transmit it to the Host using its MAC address. The working is as follows:



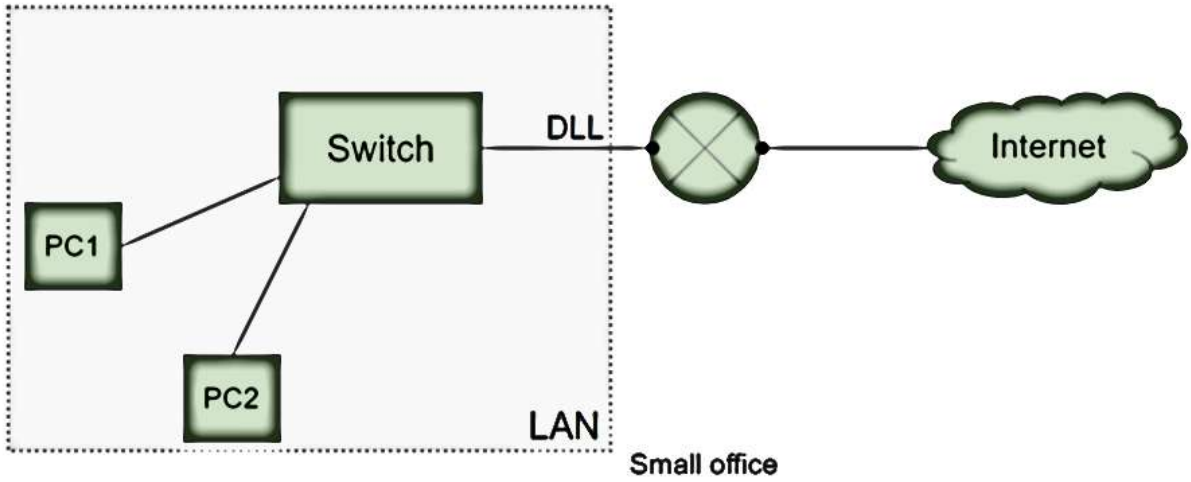
Data Link Layer is divided into two sub layers :

1. Logical Link Control (LLC)
2. Media Access Control (MAC)

The packet received from the Network layer is further divided into frames depending on the frame size of NIC(Network Interface Card). DLL also encapsulates Sender and Receiver's MAC address in the header.

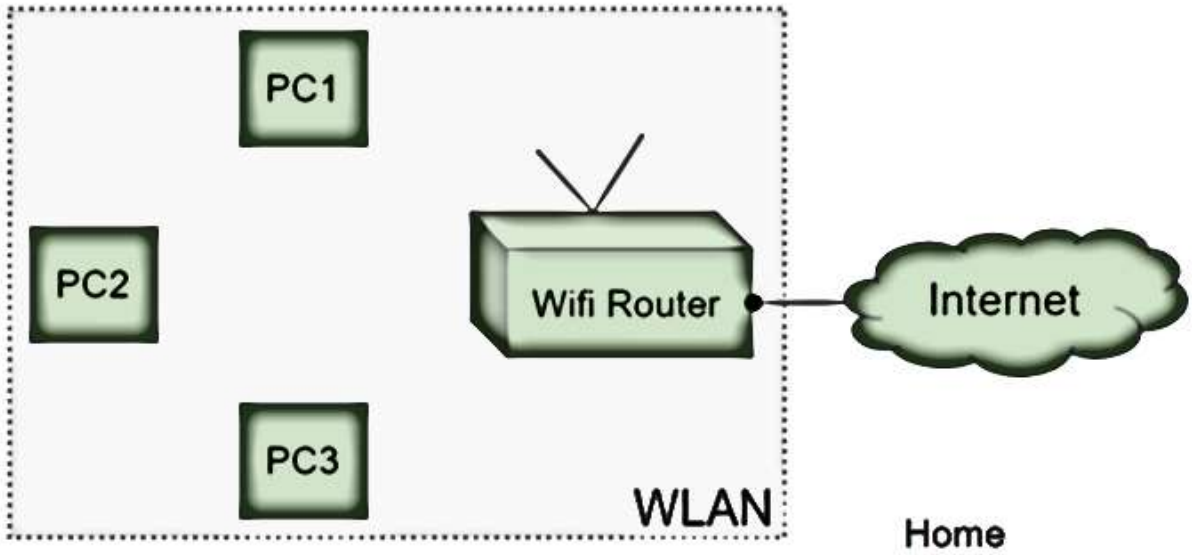
The Receiver's MAC address is obtained by placing an ARP(Address Resolution Protocol) request onto the wire asking "Who has that IP address?" and the destination host will reply with its MAC address.

Now let's see how the DLL works in a small office:



The switch is used to connect multiple computers or laptops which in turn is connected to a router. This is then connected to the internet. All the 1-to-1 connection is done using DLL. The setup is called LAN as they are all connected in Local Area Network.

Now let's see how the DLL works in a small office:



Here the router is used to convey the connection in wireless form. This is then connected to the internet. All the 1-to-1

connection is again done using DLL. The setup is called WLAN as they are all connected in Wireless Local Area Network. This network might have a collision.

The functions of the Data Link layer are :

1. **FRAMING** Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. This can be accomplished by attaching special bit patterns to the beginning and end of the frame.
2. **Physical addressing:** After creating frames, Data link layer adds physical addresses (MAC address) of sender and/or receiver in the header of each frame.
3. ERROR-DETECTION: Data link layer provides the mechanism of error control in which it detects and retransmits damaged or lost frames.
4. **Error and Flow Control:** The data rate must be constant on both sides else the data may get corrupted thus, flow control coordinates that amount of data that can be sent before receiving acknowledgement.
5. **ACCESS CONTROL:** When a single communication channel is shared by multiple devices, MAC sub-layer of data link layer helps to determine which device has control over the channel at a given time.

Packet in Data Link layer is referred as Frame.
Data Link layer is handled by the NIC (Network Interface Card) and device drivers of host machines.Switch & Bridge are Data Link Layer devices.

NETWORK LAYER: -

Network layer works for the transmission of data from one host to the other located in different networks. It also takes care of packet routing i.e. selection of the shortest path to transmit the packet, from the number of routes available. The sender & receiver's IP address are placed in the header by the network layer.
The functions of the Network layer are:

1. **Routing:** The network layer protocols determine which route is suitable from source to destination. This function of network layer is known as routing.
2. **Logical Addressing:** In order to identify each device on internetwork uniquely, network layer defines an addressing scheme. The sender & receiver's IP address are placed in the header by network layer. Such an address distinguishes each device uniquely and universally.

Segment in Network layer is referred as Packet.

Network layer is implemented by networking devices such as routers.

Let's look at some primary needs of the Network layer and why it is so important to implement:

1. **Internetworking:** Made possible using Routers. This can be across various types like 802.11, 3G, Ethernet etc

2. **Addressing:** This involves processing of IP Adresses

3. **Routing and Forwarding:** A routing table is maintained by the routers to decide how a packet must be transmitted globally to its specific IP addresses. This process does the global connection is called routing. Forwarding is more of a local concept instead of global.

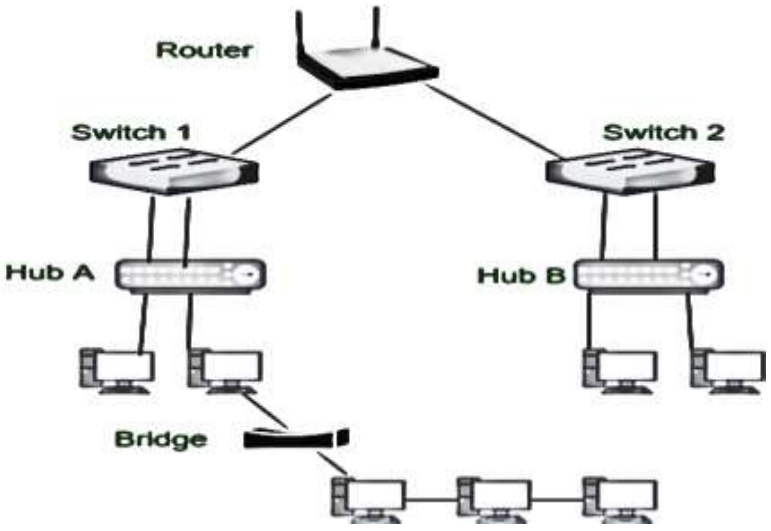
4. **Scalability (Using hierarchy in Networks):** This refers to the hierarchial organisation of packets.

5. **Bandwidth Control:** There must be a good utilisation of Bandwidth.

6. **Fragmentation and Re-assembly:** Division of bigger packets into multiple small packets and rearranging them to get the original packet is called Fragmentation and Re-assembly respectively.

Before understanding the working at the Networking layer, let's get familiar with a few technical devices that has a great role to play in this system:

1. **Switch** - A switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports imply less traffic) and performance. The switch is a data link layer device. The switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only. In other words, switch divides collision domain of hosts, but broadcast domain remains the same.
2. **Routers** - A router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.



3. **Brouter -** It is also known as the bridging router is a device which combines features of both bridge and router. It can work either at the data link layer or a network layer. Working as a router, it is capable of routing packets across networks and working as a bridge, it is capable of filtering local area network traffic.

4. **Repeater -** A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

5. **Hub -** A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, the collision domain of all hosts connected through Hub remains one. Also, they do not have the intelligence to find out the best path for data packets which leads to inefficiencies and wastage.

Types of Hub

- **Active Hub:-** These are the hubs which have their own power supply and can clean, boost and relay the signal along with the network. It serves both as a repeater as well as wiring centre. These are used to extend the maximum distance between nodes.
- **Passive Hub:-** These are the hubs which collect wiring from nodes and power supply from active hub. These hubs relay signals onto the network without cleaning and boosting them and can't be used to extend the distance between nodes.

6. **Bridge** - A bridge operates at data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

Types of Bridges

- **Transparent Bridges:-** These are the bridge in which the stations are completely unaware of the bridge's existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

- **Source Routing Bridges:-** In these bridges, routing operation is performed by source station and the frame specifies which route to follow. The hot can discover frame by sending a special frame called discovery frame, which spreads through the entire network using all possible paths to destination.

7. **Gateway -** A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically works as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.

Functions of Network Layer:

1) It helps in the delivery of data in the form of packets.

2) It helps in the delivery of packets from source host to the destination host.

3) The network layer is basically used when we want to send data over a different network.

4) In this logical addressing is used ie. when data is to be sent in the same network we need an only physical address but if we wish to send data outside network we

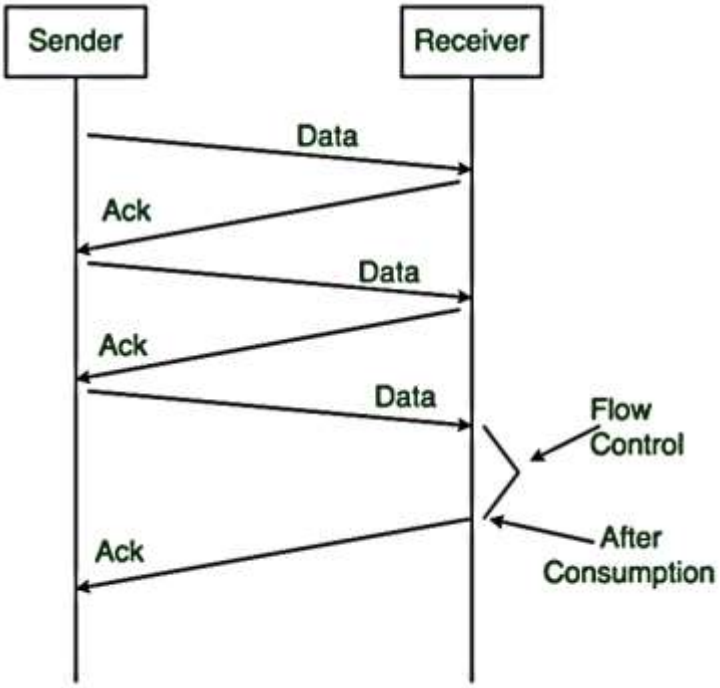
need a logical address.

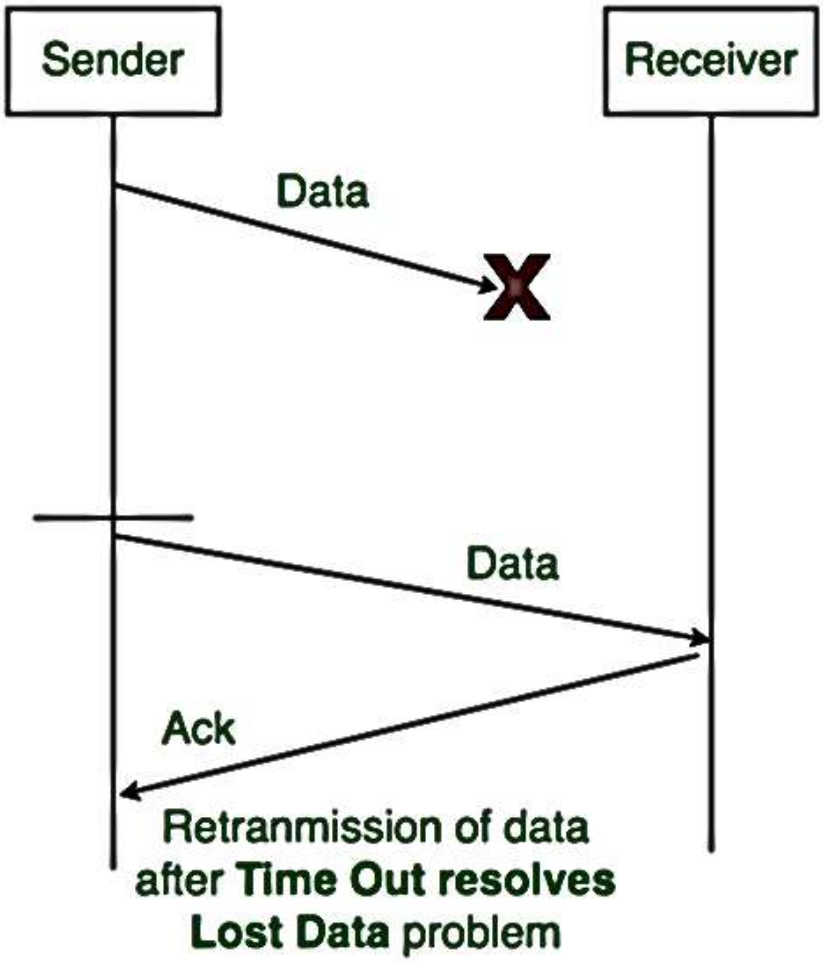
5) It helps in routing ie. routers and switches are connected at this layer to route the packets to its final destination.

FLOW-CONTROL PROTOCOLS: -

Flow Control is required in Computer Networks because, most of the time the sender has no idea about the capacity of buffer at the receiving end, and thus may transmit packets exceeding the current capacity causing them to get dropped at the receiver end. Thus, the flow control mechanism is required for re-transmission in case packets get lost. Some of the popular schemes are as follows:

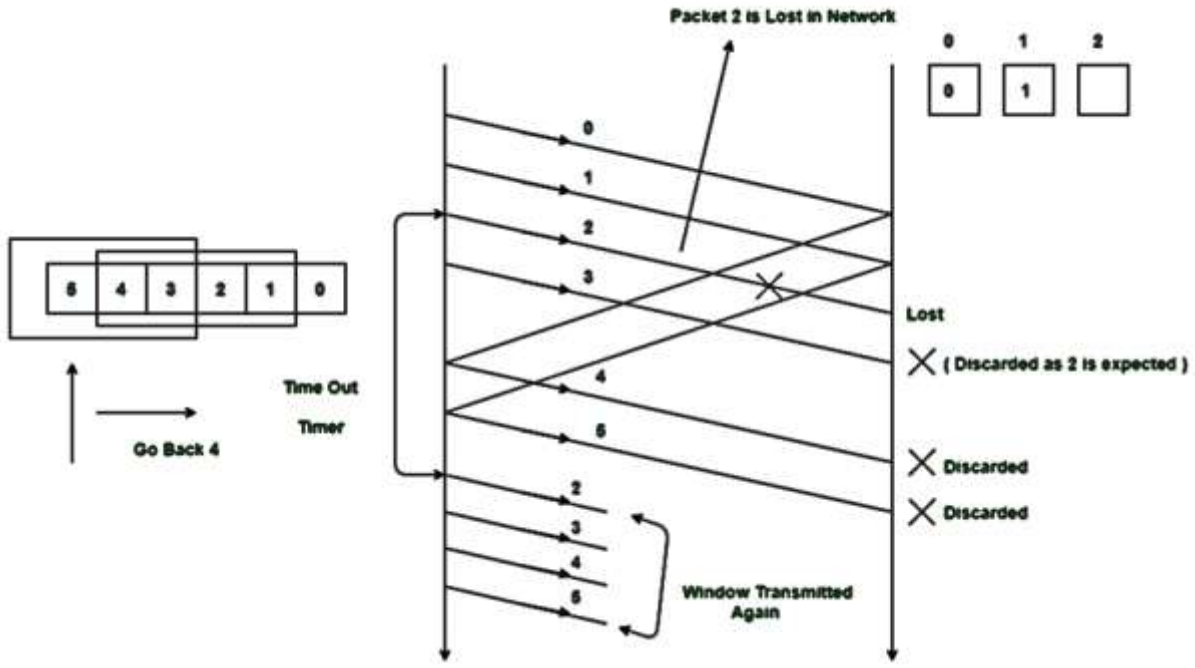
Stop & Wait (ARQ) In this scheme, the sender waits for ACK (acknowledgment) from the receiver before transmitting the next packet. If it doesn't receive ACK for a certain packet within a pre-defined timeout (ARQ variant ~ Automatic Repeat Request), it re-transmits said packet (assuming it got dropped).


Sender
Receiver
Data
Ack
Data
Ack
Data
Flow
Control
After
Consumption
Ack


Sender
Receiver
Data
X
Data
Ack
Retranmission of data
after Time Out resolves
Lost Data problem

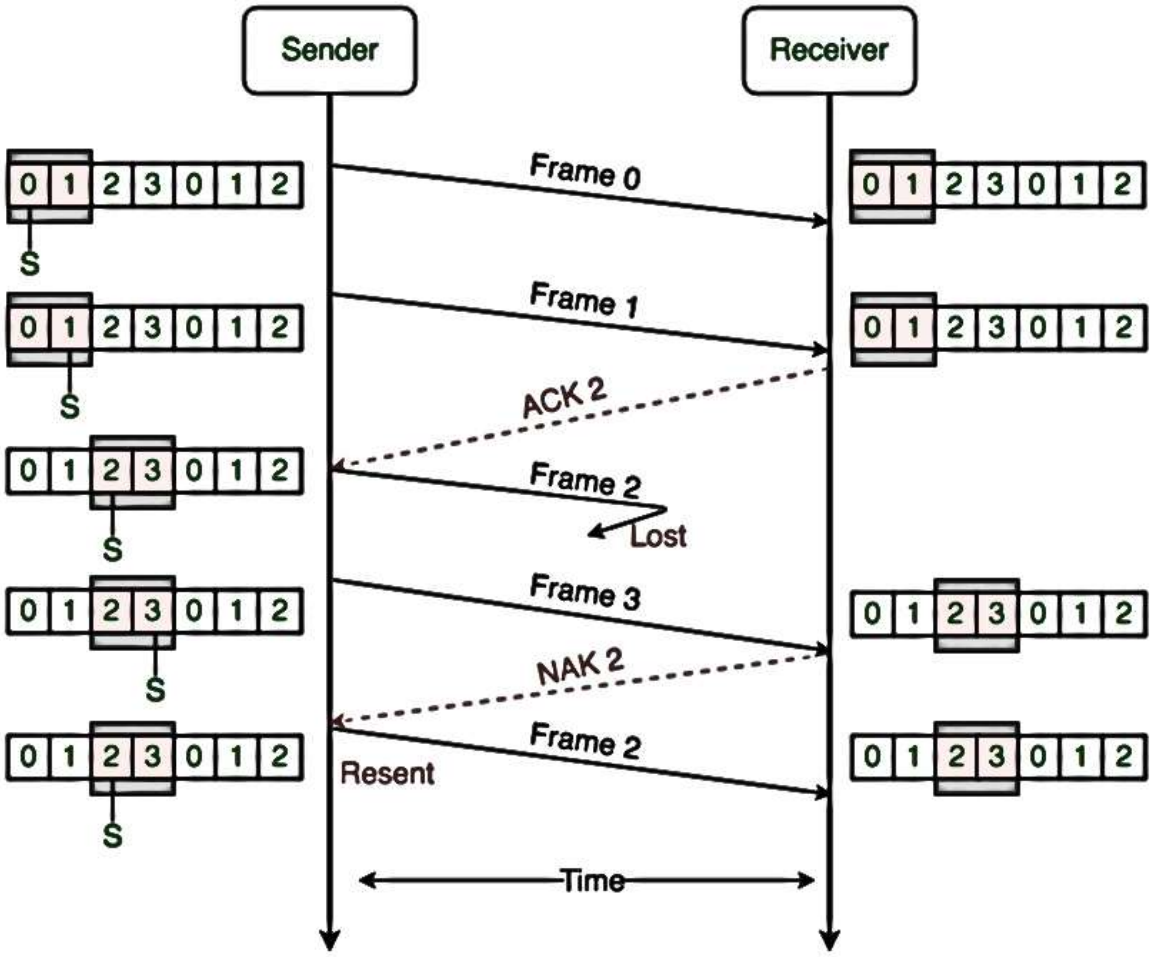
In this scheme, packets are sent one-by-one (inefficient).

Go-back-N In this scheme, the sender sends all the packets equating to the receiver window size (say n) all at once. The receiver then sends ACK_{n+1} (requesting the next packet ~ $(n+1)^{th}$). GBN uses *cumulative acknowledgment*. If any of the transmitted packets get lost, all the subsequent packets are dropped at the receiver end. Instead, a NACK (negative-acknowledgment indicating lost packet no.) is transmitted. Thereafter, all packets starting from the lost packet is re-transmitted.



As can be seen, if packet #1 gets lost, the whole window will be re-transmitted by going back n places, hence the name. It is also not much useful as unnecessarily we are repeating transmission of the whole window. We can do better as in Selective Repeat.

Selective Repeat In this scheme, when a packet gets lost, the receiver sends a NACK, however, unlike GBN, it still receives subsequent packets (GBN drops them as shown in the diagram above). Upon the reception of NACK, only that particular packet is re-transmitted.

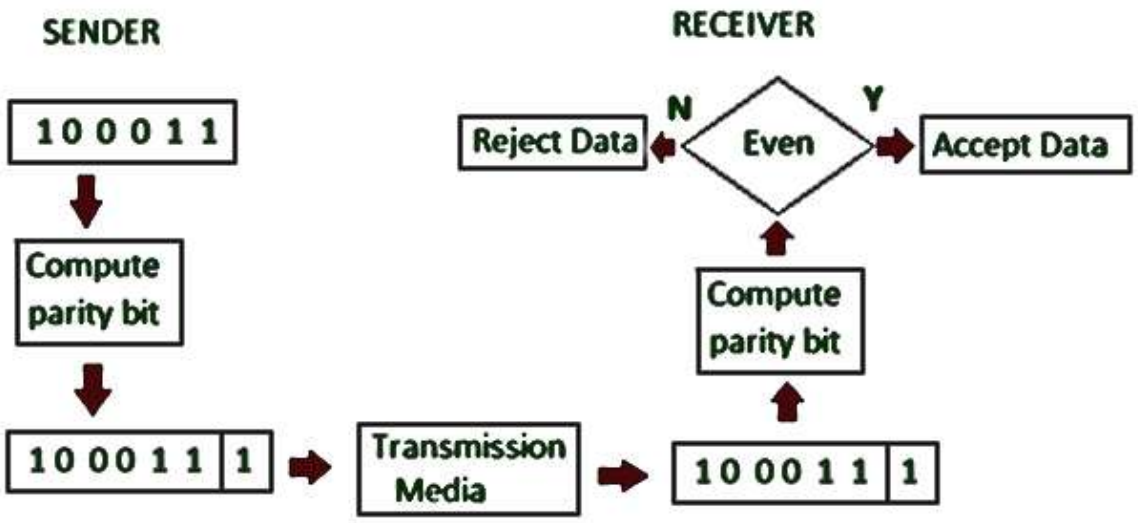


ERROR DETECTION: -

Due to noise in network and signal interference, bit values may get changed during transmission leading to so called errors. They need to be detected at the Data-Link layer, and upon detection re-transmission is requested or correction is done (as in Hamming Code). Some of the common error-detection schemes are given below:

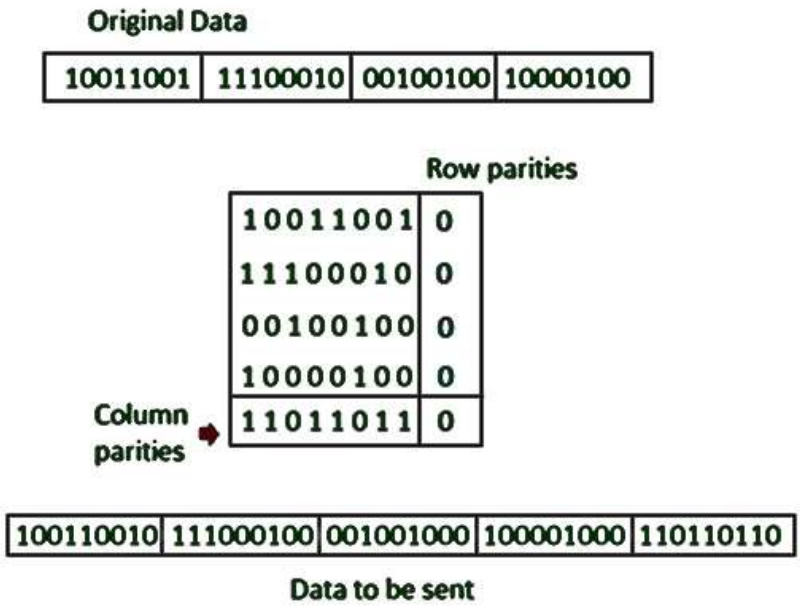
Parity Check

Parity check works by counting the no. of 1s in the bit-representation and then appending 1 in case there exists odd no. of ones, or 0 in case of even no. of 1s. Thus, the total no. of 1s become even. Hence, this scheme is also called even-parity check. Thus, if due to error any bit changes, the total no. of 1s will become odd.



2D Parity Check

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all 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.

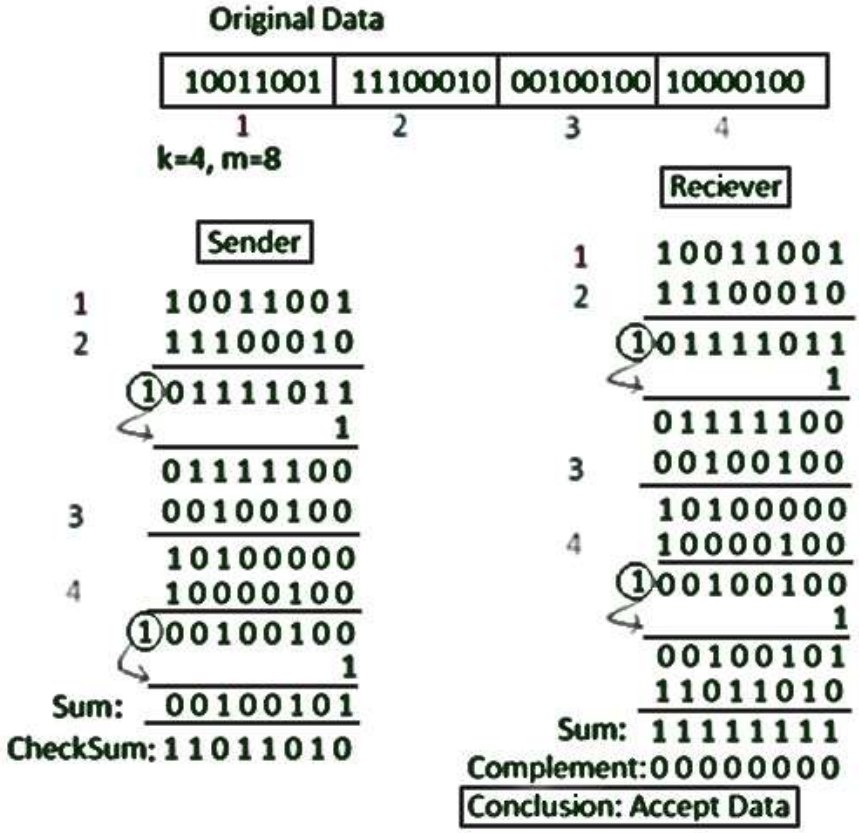


Checksum

The procedure for usage of checksum is as follows:

- Data is divided into k segments each of m-bits.
- At the sender, segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.

- The checksum segment is sent along with the data segments.
- At the receiver, all received segments are added using 1's complement arithmetic. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.



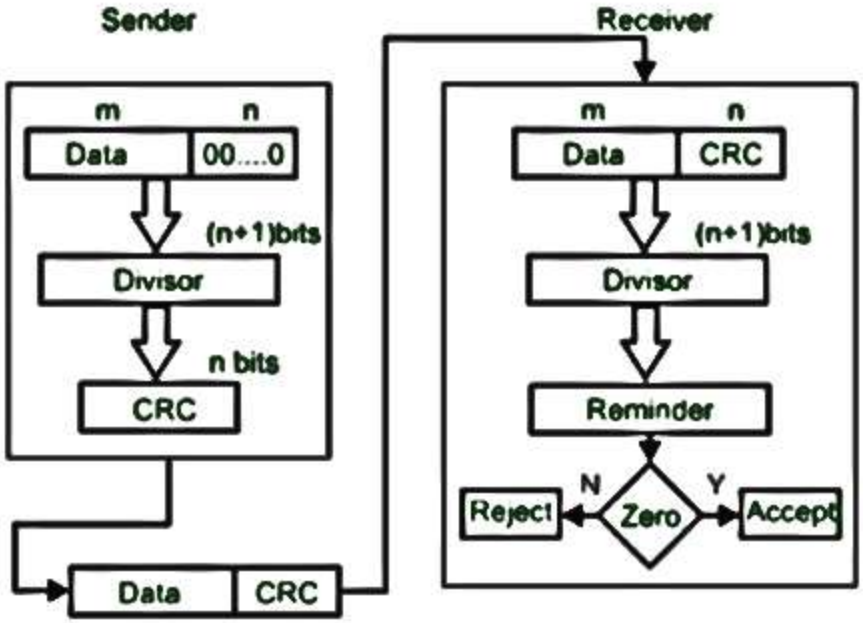
CRC (Cyclic Redundancy Check)

CRC is based on binary division, and it works as:

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

- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

The whole procedure can be better understood with an example:



Pv4 and IPv6: -

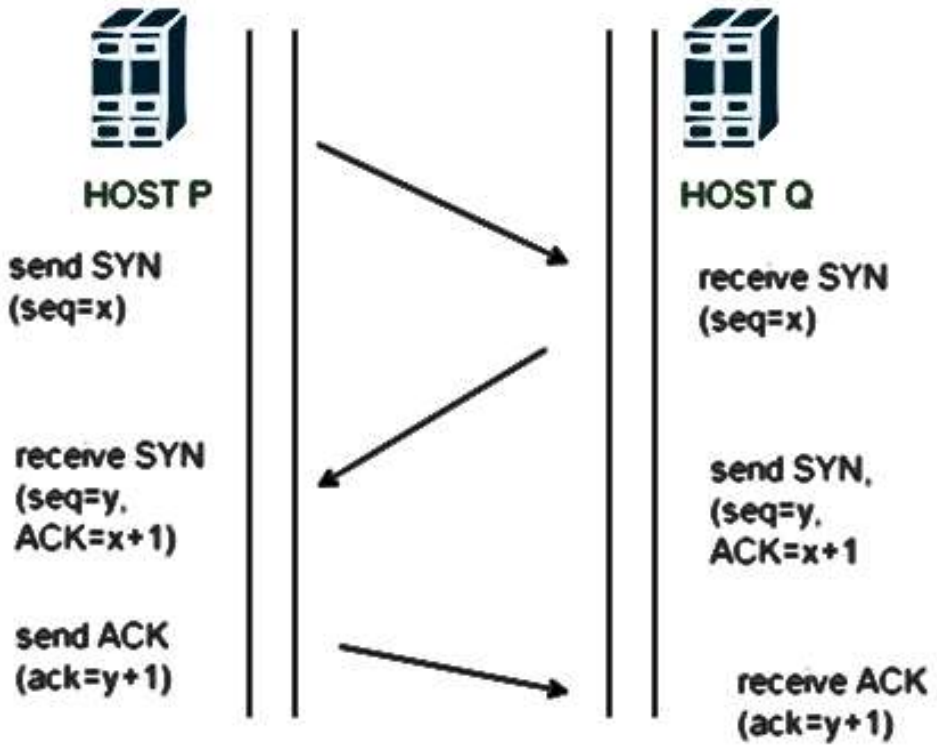
These are internet protocol version 4 and internet protocol version 6, IP version 6 is the new version of Internet Protocol, which is way better than IP version 4 in terms of complexity and efficiency.

address space shortage in IPv4.

Features	IPv4	IPv6
Address	32 bits	128 bits
Checksum in header	Included	No checksum
Header includes options	Required	Moved to IPv6 extension headers
Quality of Services (QoS)	Differentiated Services	Use traffic classes & flow labels
Fragmentation	Done by routers & source node	Only by the source node.
IP configuration	Manually or DHCP	Auto-configuration or DHCP
IPSec support	Optional	Required
Unicast, multicast and broadcast	Use all	Uses unicast, multicast and anycast
Address Resolution Protocol (ARP)	Use to resolve an IPv4 address	replaced by Neighbor Discovery
Internet Group Management Protocol (IGMP)	Use to manage local subnet group.	Replaced with Multicast Listener Discovery (MLD)
Domain Name System (DNS)	Use host address (A) resource records	Use host address (AAAA) resource records
Mobility	Use Mobile IPv4 (MIPv4)	MIPv6 with faster handover, routing and hierarchical mobility

TCP & UDP (DIFFERENCE) : -

TCP is a connection-oriented, stateful protocol which ensures security, reliability in data transfer. It is established as a **3-way handshake process** given below:



1. SYN: In the first step, sender sends a segment with SYN message (containing Synchronize Sequence Number) expressing it's wish to establish a connection. The Sequence No. determines what segment it wants to start the communication with.

2. SYN + ACK: Receiver responds by replying with a segment with *SYN-ACK* bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with.

3. ACK: In the final part, sender acknowledges the response and they both establish a reliable connection with which actual data transfer can occur.

UDP on the other hand is a connection-less protocol, which doesn't care about reliability. Thus, if some packets get lost, they are skipped. It is thus used in applications where it doesn't matter if we lose out some data. e.g. Video Streaming/Call , VoIP (Voice-over-IP), Multiplayer Games. UDP's main focus is transmission speed (thus re-transmissions are not done in this protocol).

More differences between TCP and UDP are given below:

UDP v/s TCP		
Characteristics/ Description	**UDP**	**TCP**
General Description	Simple High speed low functionality "wrapper" that interface applications to the network layer and does little else	Full-featured protocol that allows applications to send data reliably without worrying about network layer issues.
Protocol connection Setup	Connection less data is sent without setup	Connection-oriented; Connection must be Established prior to transmission.
Data interface to application	Message base-based is sent in discrete packages by the application.	Stream-based; data is sent by the application with no particular structure
Reliability and Acknowledgements	Unreliable best-effort delivery without acknowledgements	Reliable delivery of message all data is acknowledged.
Retransmissions	Not performed. Application must detect lost data and retransmit if needed.	Delivery of all data is managed, and lost data is retransmitted automatically.
Features Provided to Manage flow of Data	None	Flow control using sliding windows; window size adjustment heuristics; congestion avoidance algorithms
Overhead	Very Low	Low, but higher than UDP
Transmission speed	Very High	High but not as high as UDP
Data Quantity Suitability	Small to moderate amounts of data.	Small to very large amounts of data.

HIMANSHU KUMAR(LINKEDIN)

https://www.linkedin.com/in/himanshukumarmahuri

CREDITS- INTERNET.

DISCLOSURE- ALL THE DATA AND IMAGES ARE TAKEN FROM GOOGLE AND INTERNET.