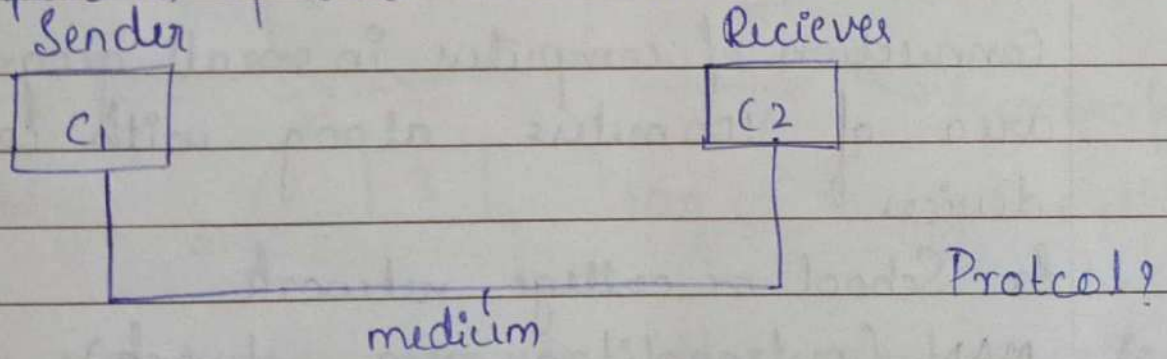
Computer networks

* It is connection b/w two or more computers (nodes) for data exchange.

* In 1969 the first computer network was started called arpanet (American Research & project Agency)
In India 1995 Aug 15 → BSNL/VSNL (videsh sanchar nigam limited)

Components of data communication:



Data/msg/packet

Medium can be two types:

(i) Wired
- Copper cable (1)
- Coxial cable (2)
- Category 5 / category 6 cables (1)
- OFC (optical fibre cable)

(ii) Wireless
It works on radio fq's
2.4 GHz
5:1 GHz

Sender is machine that sends the information
Recieves is machine that recieves the information
Protocol is the set of rules that governs the data transfer.
Medium is through the data is transfered
Data is the information/message/packets that is to be transfered for communication
Data can be processed or unprocessed

Types of networks:-

1) LAN (local area network)
Connection of computers in small geographical area of 100 meters along with connecting devices
Ex: School or college network

2) MAN (metropolitan area network)
Connection of computers in entire city or in geographical area of 30-40km range.
Ex: Cable providers (when DTH was not there)

3) WAN (wide area network) -OR- WWW (world wide web)
Ex: Online gaming
All machines on earth are connected through internet

Topology

Physical structure of a network is called topology

Multiple arrangements are possible for data transfer

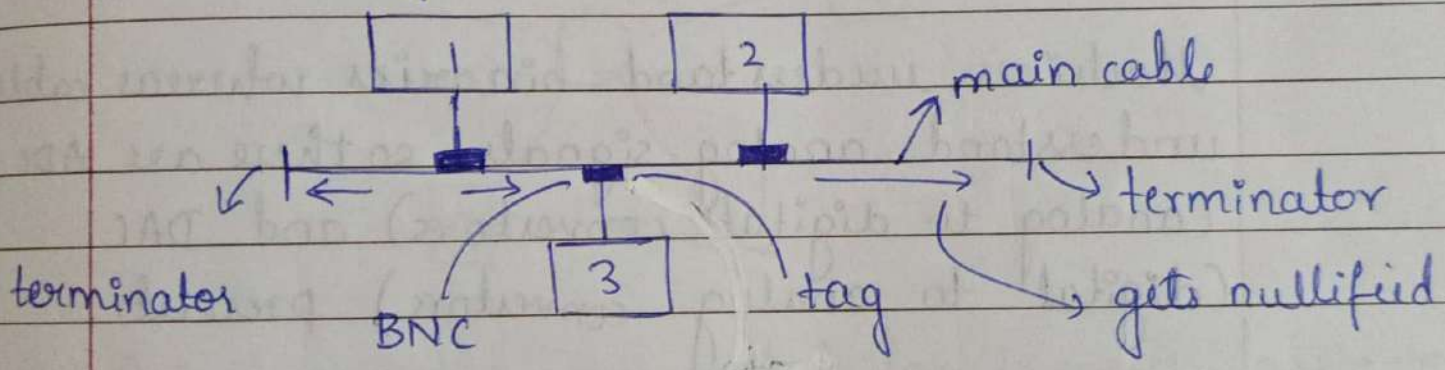
Multiple arrangements may be due to:

Cost effective

Less loss

Convience / fast data transfer

(1) Bus topology



Here 1, 2, 3 are computers that are connected to main cable or backbone or back cable

At the end of the main cable terminators are attached called cable terminator. All machines are connected to the main cable via tag or BNC (bicodd normal form connectors)

Machine1, Machine2 and machine3 all are connected to main cable via BNC. All machines share same cable.

Merits: Simple structure, cost effective

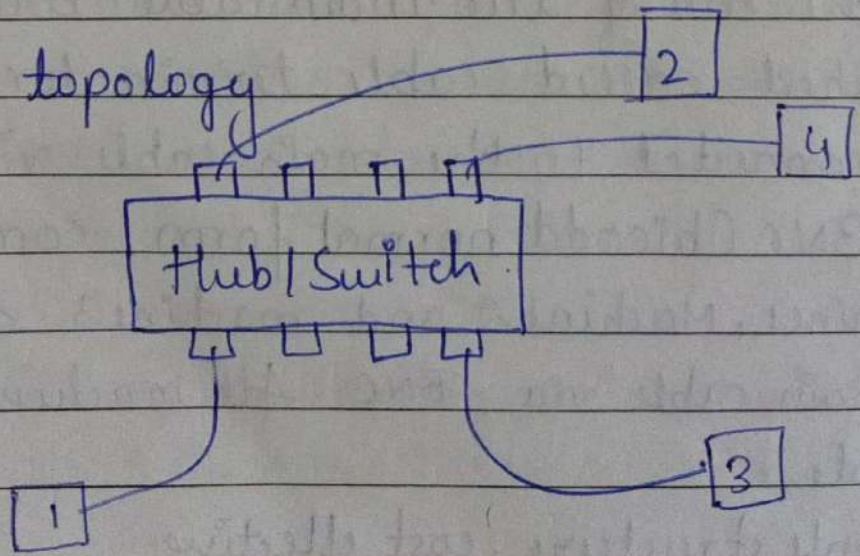
How?:- Information will travel in the form of broadcast

Suppose machine 1 wants to transfer data to machine 2. the information travels from a cable and it's broadcast in nature i.e info travels in all direction. Information will reach machine 2 and machine 3. machine 3 will ignore but machine 2 will accept it. Information travelled till end gets nullified at terminators.

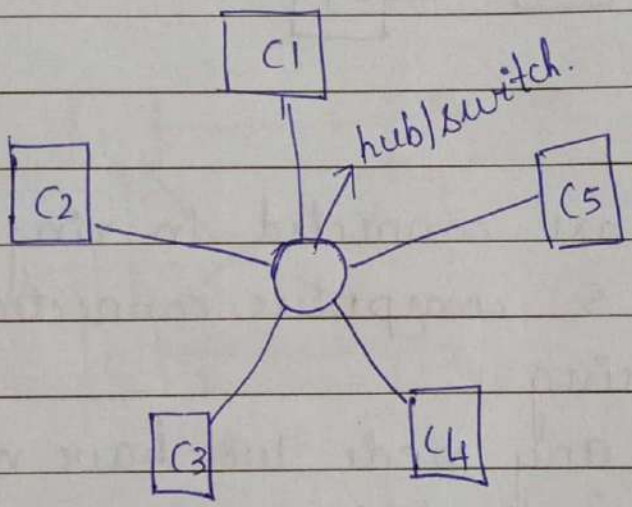
Machines understand binaries whereas cables understand analog signals - so there are ADC (analog to digital convertors) and DAC (digital to analog convertors) present.

Demerits: i) Cable disconnection cause network breakdown.

2) STAR topology



In star topology there is a central hub/switch/bridge (central connecting device). This hub has latches popularly called as RJ-45 jacks where we connect different computers with cables. In star topology center will be controlles or he acts as connector for different machines.

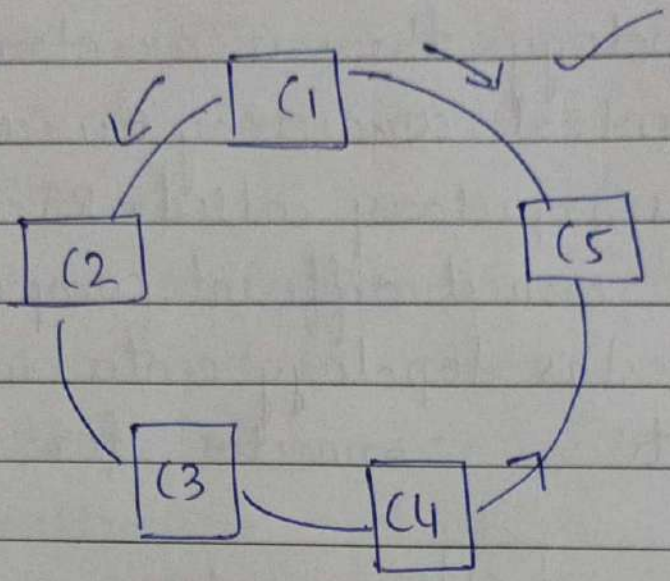


Merit2: Only two paths are needed for transfer. Source to hub and hub to destination.
Any one node fails network will not get affected other nodes can communicate

Merit1: Center controlled

Demerit: Any damage to hub/switch entire network fails even if all nodes are working no data transfer happens.

3) RING



Computers are connected in ring like passion There are 5 computers connected in the shape of ring

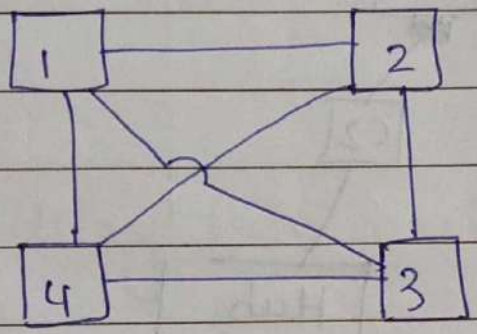
Merit: To reach any node we have multiple paths (no single point of failure)
To reach from 1 to 4 $1 \rightarrow 5 \rightarrow 4$ or $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$. In nature it always take shortest path $1 \rightarrow 5 \rightarrow 4$.
If one path gets disconnected there is another path

Demerit: No. of cables required are more compared to Bus or star.
If node 1 is failed 4 doesn't even come to know that 1 has gone down it tries sending data in first path if its not

successful then it tries second path also. Because it assumes it has two paths to reach one.

"Any situation where one machine fails data transfer happens twice"

4) MESH

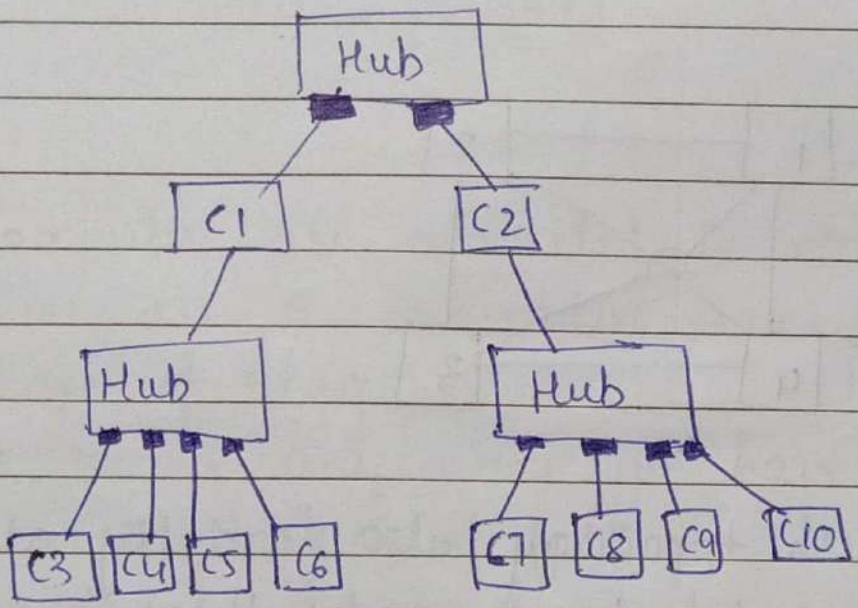


It is mesh topology also generally called fully connected. From node1 there is connection node 2, 3 and 4. From any node to any and every node there is connection. So there are three paths for every node to reach other. One/two cable disconnection doesn't affect the network but no. of cables are more so costly. Connectivity is ensured with multiple paths.

Speed, reliability, data loss and other factors are considered for using different topologies

5) Hybrid
This is mixture of all types

6) Tree.
Structure looks like tree



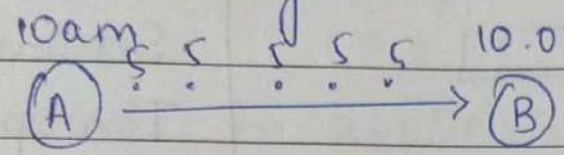
Network Performance (QoS (Quality of Service))

1) Bandwidth : Capacity of a network to transfer the data. Unit measured is one second

2) Throughput: Effective data transfer. or actual data transferred.
Bandwidth ≥ Throughput.

Attenution, data loss, thermal or induced heat may be different factors for difference.

3) Delay or latency:

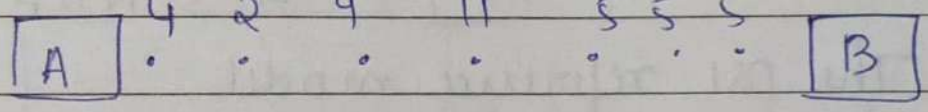


for data packet.

Total time taken to reach destination

4) Jitter.

Uneven delay: From packet to packet the delay differs.

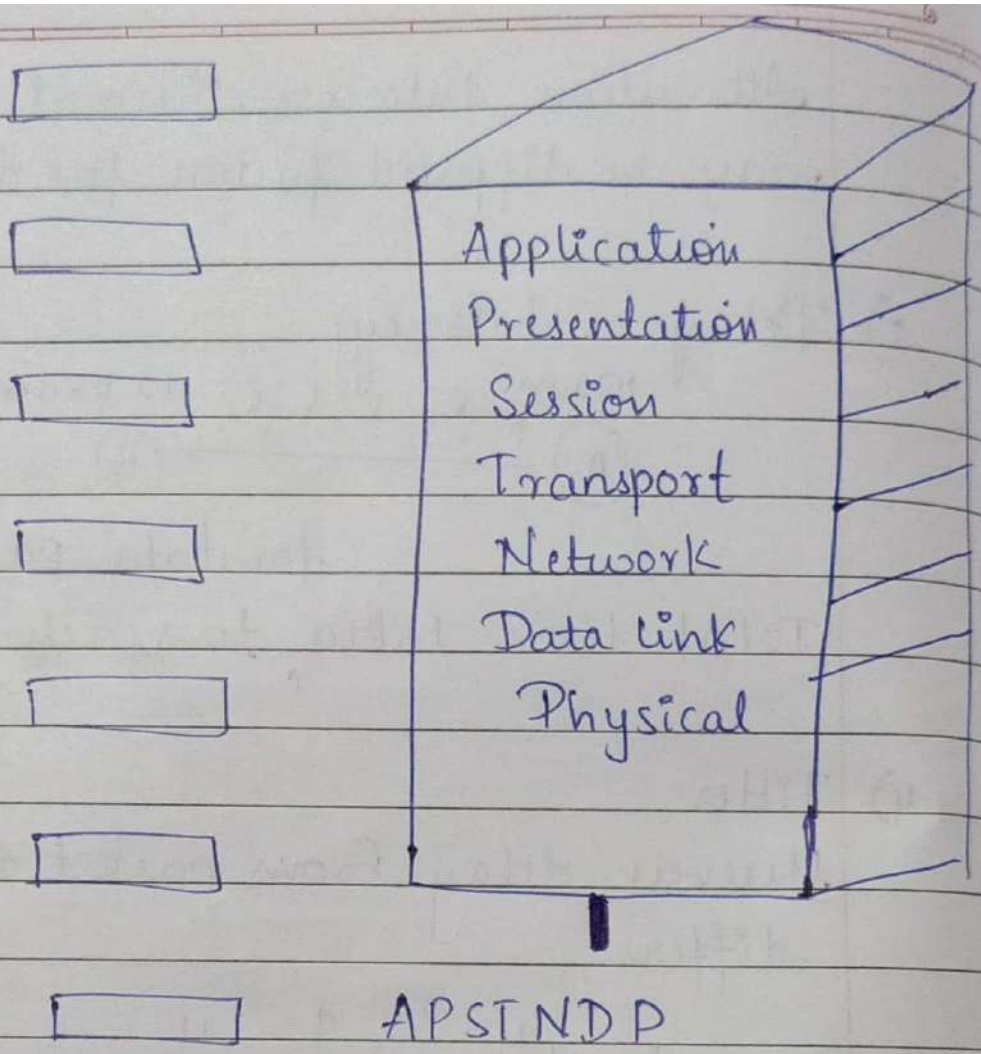


Software concepts of networks

1) OSI Model (Open System for Interconnection) is approved by ISO = international organization of Standardization.

OSI-7 layer model.

There are 7 layers in model.



The OSI reference model.

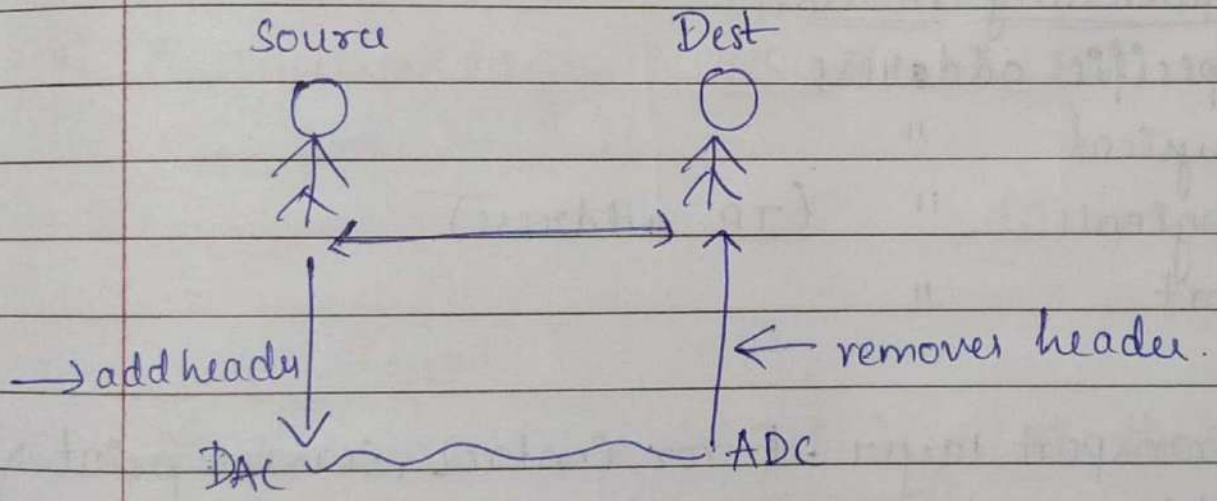
Layered architecture → Every layer looks independent of each other. Moduldrization is applid The job of a layer is independent of other.

Physical layer → the first layer connected to hardware

Application layer → directly connected to user for its application

They are popularly known as drivers or network drivers.
In both client and server machines these software should be there.

User reacts to application layer which in turn reacts with presentation --- so-on - physical / cable to server cable then upto application layer of receiver



Every layer is interconnected Every layer adds its header.

Description of layers.

1) Physical layer -> physically connected layer Representation of bits, data rate, topologies, line configuration.

It checks line configuration.
L1 layer is technical name.

2) Data link layer - physical addressing, flow control, Access control.

3) Network layer or L3 layer - routing, logical addressing
↓
shortest path

Addressing modes:
Specific address
physical "
logical " (IP address)
Port "

4) Transport layer - Error Control, Service point addressing, segmentation and reassembling
↓
making datapacket
Stop intermixing of concurrent accus
reassembling data packets

5) Session layer - Dialog Control.
login, logout, session time, session expire.

6) Presentation / Syntax layer : Translation, Encryption

Convert information to standarized format.
It helps in encryption decryption algos

7) Application layer - DNS, Email, FTP, remote login
↓
File transfer.

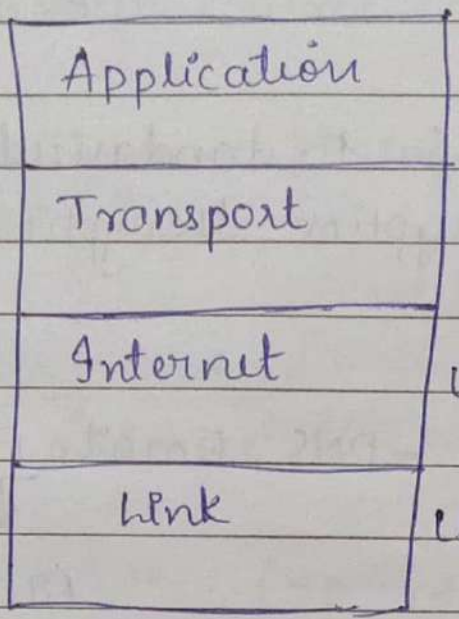
OSI layer is a conceptual layer On the top of that present day models are built
The current model is TCP/IP model

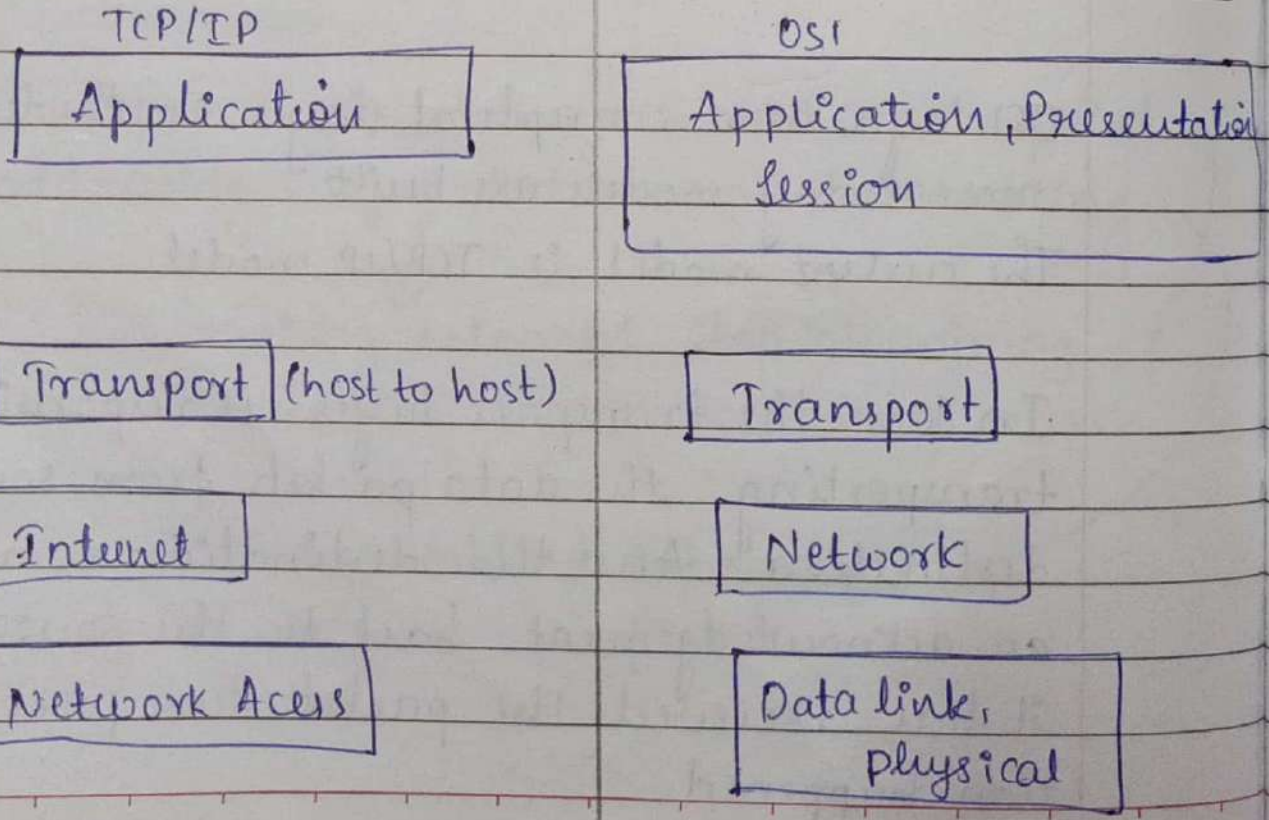
In OSI the transport layer is responsible for transporting the data packets from source to destination. And the destination would send an acknowledgement back to the source saying it has recieved the packet. Two packet transaction happened.

DATE : / /

TCP/IP (Transmission Control Protocol / Internet Protocol)

It is four layered active architecture.





1) Network Layer

Protocols:

PPP - point to point protocol

Ethernet

HDLC - High level data link control.

Interface drivers

2) Internet layer (find shortest route)

RIP - Routing information protocol

IP -

ICMP -

OSP - open shortest path first protocol

3) Transport layer - port to port delivery

UDP - Not crucial no acknowledgement (connectionless)

TCP - Crucial and acknowledgement (Connection oriented)

One protol gives acknowledgement.

One protocol does not give acknowledgement.

4) Application layer.

HTTPS is more secured

SMTP - Simple mail transfer protocol.

DHCP - dynamic host configuration protocol

DNS - Domain name servers

Four layers with different protocols are kept to all jobs of 7 layers with choices.

RAR - Reverse address. Registration protocol.

Current mode on which all machines are working.
The device / network drives running on our system - TCP/IP suite.

Small protocols are next level of modularization

Network Attacks

Network can get hacked by external bodies or software popularly called as viruses

1) Virus / Malware / worm - Is a program if it enters your system it can corrupt your data, system down or cause network can go down.
Trojan, I love you virus

2) Dos - or - Denial of Service attacks
DDos - Distributed denial of service.

3) Packet sniffer
Change / sniff info from packet

4) Masquerade.

Tutorial-01

Devices that work on OSI layers

1 NIC card

Network Interface card – which is considered as physical layer as well as data link layer due to its dual functionlity. It acts as l1 and L2 However its main purpose is associated with the use of MAC address. So its commonly refered to as data link layer / device.

Physical address of a device its stored at NIC card. The NIC of the system is a place where its MAC address is stored. It is primary device that lets the computer to connect to network. The NIC is generally built into the motherboard of computer this ensures that every device on a network has unique MAC address. ∴ of having a different NIC

2) Cables - connects two device there is no interconnection / interaction with data its merely a physical connection Device used in first layer. It is medium for data transfer

Ex: CAT5 - Twisted pair cable
Coxial cable
OFC

Optic fibres are used for speed data transfer and more data communication. Works at layer1

3) Hub. Used in physical layer works in layer1
Basic job is to connect to different devices and generate signals Hub forwards msgs to all devices it's connected except the sender device

4) Access point - is a wireless network device that acts as a portal for devices to connect to LAN

It is used to extend the wireless coverage of an existing network & for increasing the no. of users that can connect to it.
A cable from router is dragged to access point.

Access point converts wired signals to wireless one
It works in layer2

5) Multilayer switch - multi layer device. Work in L2 and L3.

The data link - successfully could perform the task of switch by forwarding all the frames to the required devices using the MAC addresses.

Network layer -

It can perform task of router. Recieve & forward the data packets to their destination IP addresses. It avoids need of the separate switch & router.

6) Router - works at layer 3 of OSI model.

Router is known as network layer device.

It basically sends the packet formed during network layer to different devices based on their IP address. Each packet has an IP address associated with it.

Chapter-02 : Application layer

Principles of network applications

(i) Architecture

Client - Server Architecture

Peer to Peer architecture.

Client-Server architecture

→ Here we have server device always in ON state. client initiates connection to server. Server is going to have permanent IP address

Whenever client connects to server the client is given a dynamic IP address to communicate with server

Peer-to-Peer

Here two peers talk to each other. Scalability is not the issue.

Management of communication b/w two peers is easy.

Server is not in ON state.

Process: programming running within the host

→ IPC (defined by OS). (within same device)

→ processes in different hosts communicate by exchanging messages

client process: process that initiates communication

server process: process that waits to be contacted

Sockets

→ Process sends/recieves messages to/from its sockets

→ Sockets analogous to door

Addressing processes

→ To recieve messages process must have identifier

→ So identifier includes both IP addresses and port numbers associated with process on host

Port number for HTTP: 80

DHCP (Dynamic Host Configuration Protocol)

→ Host is any machine or device connected to internet

→ Configurations are set of instructions the network administrator uses to configure the host machine to get connected to internet

→ To automate the configuration of host device we use DHCP

→ So protcol is set of rules to do so

⇒ Now host can join the network and leave the network dynamically

⇒ When host joins/leaves the network the IP address of host must be configured

i.e. host device must obtain IP address dynamically from server.

Allocation of IP address to host machine happens in four steps :-

(i) Discover (ii) offer (iii) Request (iv) ACK (DORA)

1. DHCP gives discover option for host machine
2. Server offers IP address to host machines
3. Host machine accepts the IP address that server offers
4. Acknowledges the request given by server

Pool of IP address those can be reused by DHCP server

Quiz DHCP uses UDP port 67 for sending data to server

HTTP Request cycle.

505- HTTP version not supported

400 - Bad Request

404 - Not found

200 - OK

301 - moved permanently

Commands:

ping google.com

ifconfig

Web Servers Installation.

1. Open terminal
2. sudo apt-get update // update packages
3. sudo apt-get install apache2 -y
4. sudo systemctl status apache2
5. sudo system services

Open browser - localhost

Web hosting

1. cd /var/www/html
2. ls
3. sudo cp index.html index1.html
4. ls
5. sudo rm index.html
6. ls
7. Create html file // sudo vim index.html.

sudo vim /hostname

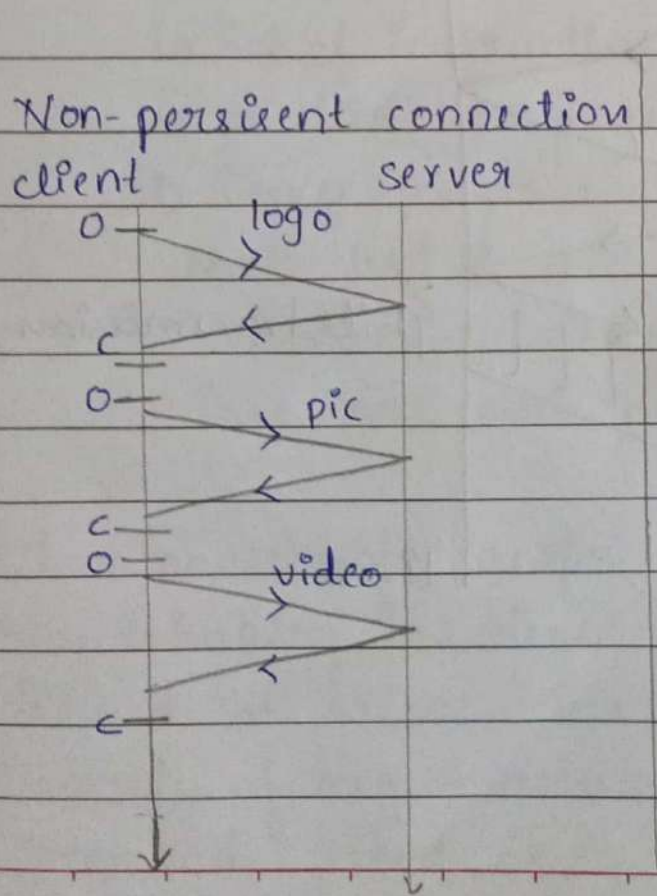
sudo vim host

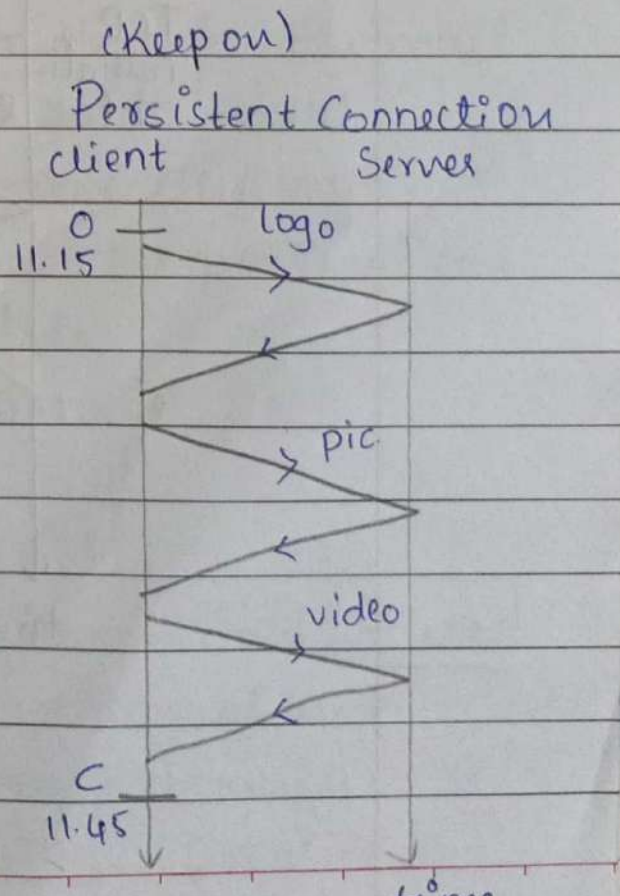
Application layer [contd]

There are two types of HTTP connections:

1. Non persistent connection
   ↳ one object is sent at a time
2. Persistent connection (HTTP 1.1)
   ↳ multiple objects are sent

1 TCP connection open
2. Object transfer
3 TCP connection close



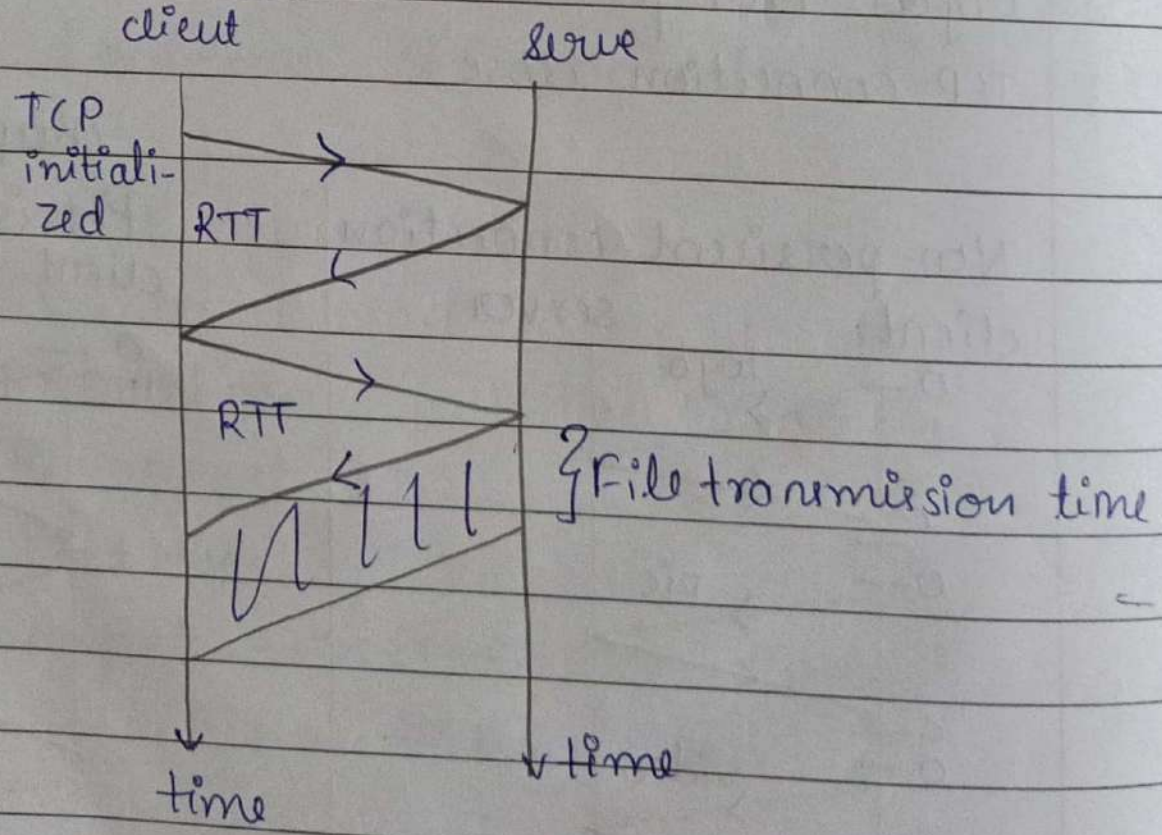


In persistent connection multiple objects are retrieved from the server where as in non-persistent only one object is retrieved at a time

Non-persistent HTTP : Response time.

RTT: (Round Trim Time)
time for a small packet to travel from client to server and back)

Non-persistent HTTP response time: 2RTT + file transmission time



DATE : / /

- Persistent (HTTP 1.1)
  - Server leaves the connection open after sending response

- SMTP (Simple Mail Transfer Protocol)
  SMTP in Electronic Mail application

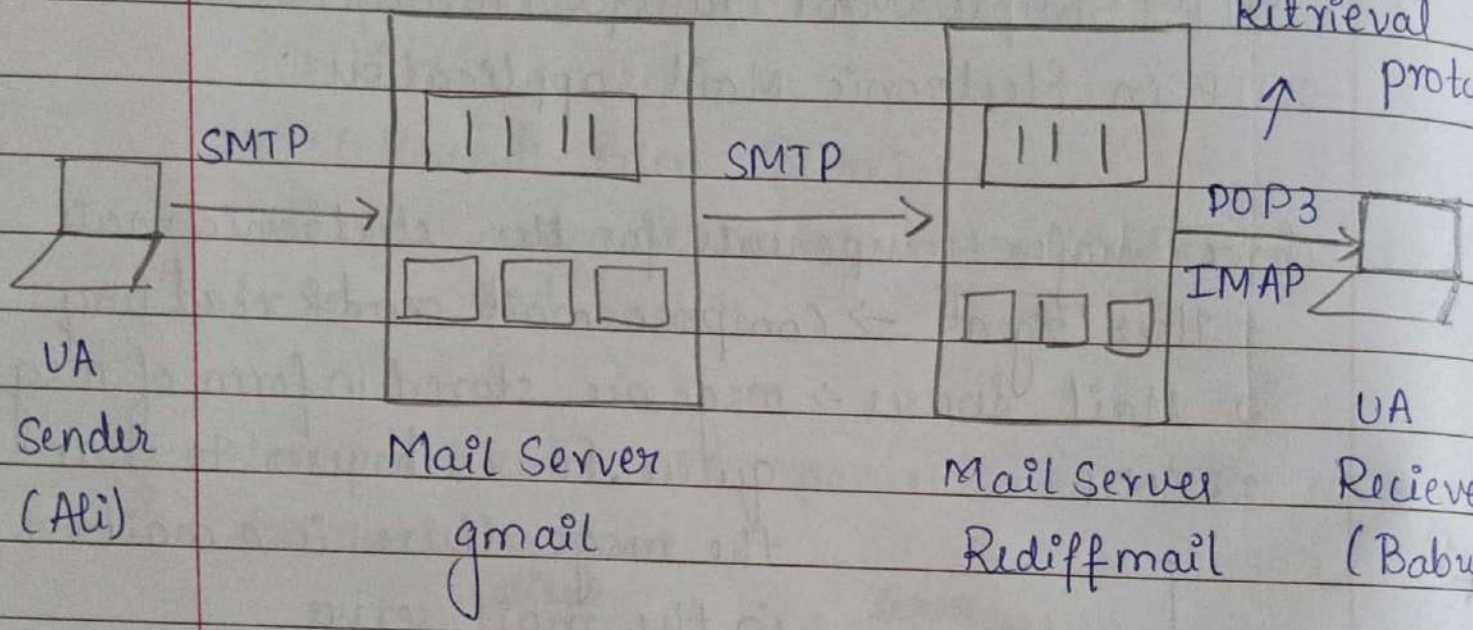
Three major components for the electronic mail
1. User Agent → Compose mail, send & read msg
2. Mail Server → msgs are stored in form of msg queue. For each user to store the msgs there is a mail box in the mail server
3. SMTP
   - Set of instructions to transfer the mail
   - A client program to send a mail
   - Server program to recieve the mail
   - Mail that is sent uses TCP protocol for reliable transfer of data
   - Port no used by SMTP is 25

RTT

Three phases of transfer
1. Handshaking → Here the command [Helo] is used at client side (greeting phase)
2. Transfer of msg → actual data transfer
   Commaud used is [DATA] Data is

transferred b/w client device and server device

3. Closure → When client quits QUIT



POP3 → Post Office protocol
IMAP → Internet mail access protocol
IMAP → has more features than POP3

SMTP uses TCP

URL: Domain Name Service (DNS)
www.kletech.ac.in
↓ ↳ countrycode
academics

DNS uses well-known port 53

PAGE NO.:
DATE: / /

The server gets identified by URL. The server has a unique IP address. All host machines are identified by unique IP. So the domain name internally maps to the IP.

www.google.com
↳ commercial

www.wikipedia.org
↳ organisation

DNS maps the domain name to IP address and vice versa

Types of Domains:-

1. Top level Domain (TLD)
   → .com → .org

2. Country Code TLD (ccTLD)
   .in .us .jp

.com → corporation

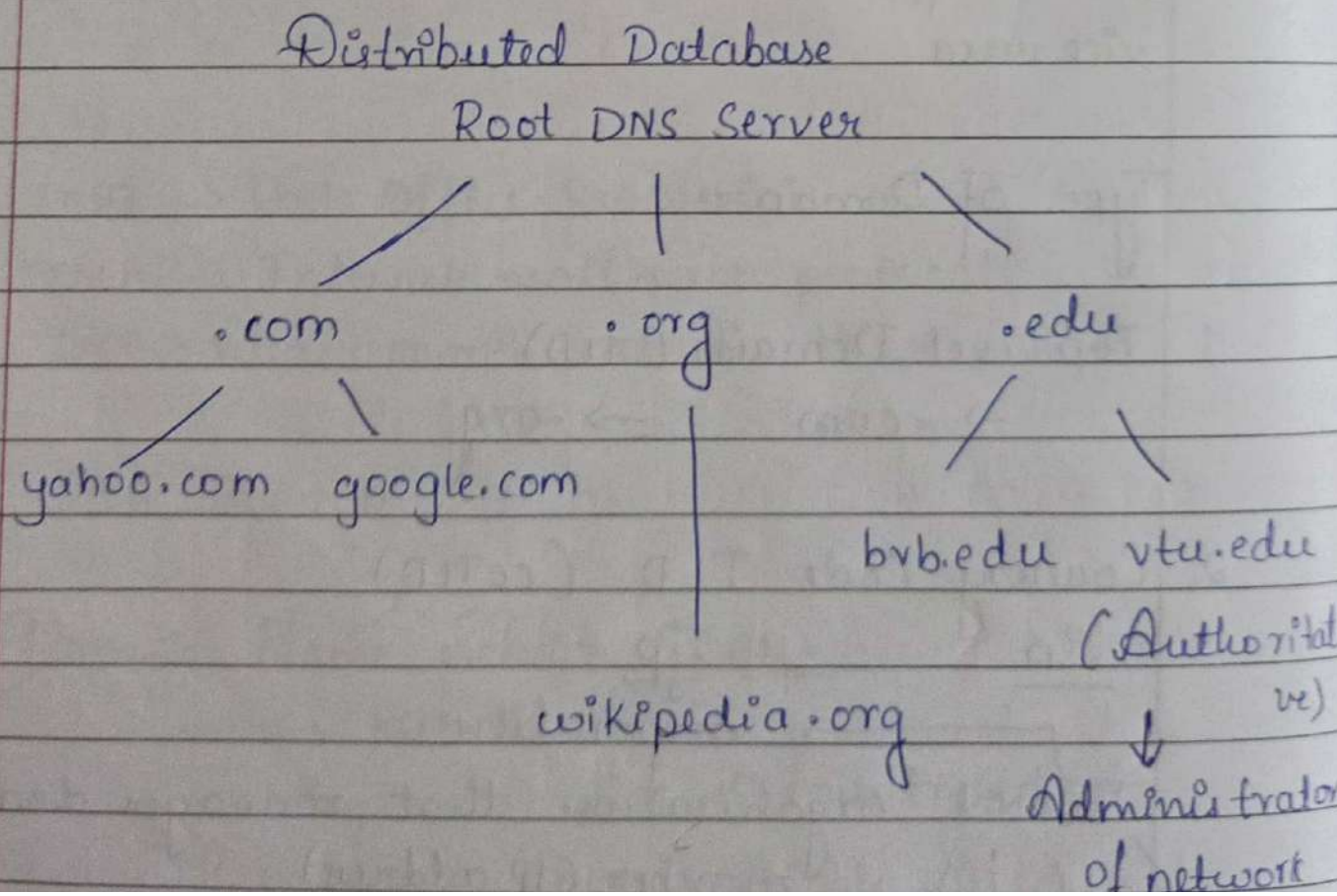
ICANN - Organization that manages domain name
↳ Internet
↳ number (IP address)
↳ names for domain
↳ assigned

www.klelech.ac.in

klelech.ac.in → domain name
ac.in → domain extension
.in → Top level domain
TLD type - ccTLD

DNS: Domain Name System (Port 53)

Distributed database implemented in hierarchy of many name servers. It is directory service.



Domain Name Resolution

Resolve : : ; domain name to get IP address
Two ways:
1. Iterated Query
2. Recursive Query

Peer-to-Peer application (P2P)
Eg: Bit Torrent, Shareit

DH-03 (D-discover, O-offer, R-request, A-acknowledgement)
DHCP follows UDP Protocol.
DHCP discover is broadcast in nature so it sends request to all of the DHCP server and client machine
Inside UDP MAC is put.
If we don't configure DHCP server 67 is closed

Packets enter till transfer layer all other client ignore because 67 is closed but DHCP server accepts it as 67 is open and provides an offer for IP address from pool of IP along with subnet mask, default gateway

P2P applications:

Peers communicate with each other and server is not in the picture

File dist'n : where the appl'n distributes a file from a single source to a large no. of peers
Ex: Bit-torrent, ShareIt.

Here dist'n time reduces as peers assits in distributing the file.

→ Servers is burdened to distribution
→ Wastage of bandwidth on server side

u_s : upload rate of server
u_i : upload rate of i^{th} peer
d_i : download " " " "
size of file : F bits
Client - Server Architecture

S $\xrightarrow[\text{copy}]{\text{one}}$ N peers

S $\xrightarrow[\text{bits.}]{\text{Total}}$ $\frac{NF \text{ bits}}{u_s}$

$d_{\min}_{\text{peers}} = \{d_1, d_2 - - - d_N\}$

$\frac{F}{d_{min}}$ } → peer with least download rate

It depends on:
Server upload rate & minimum download among the peers

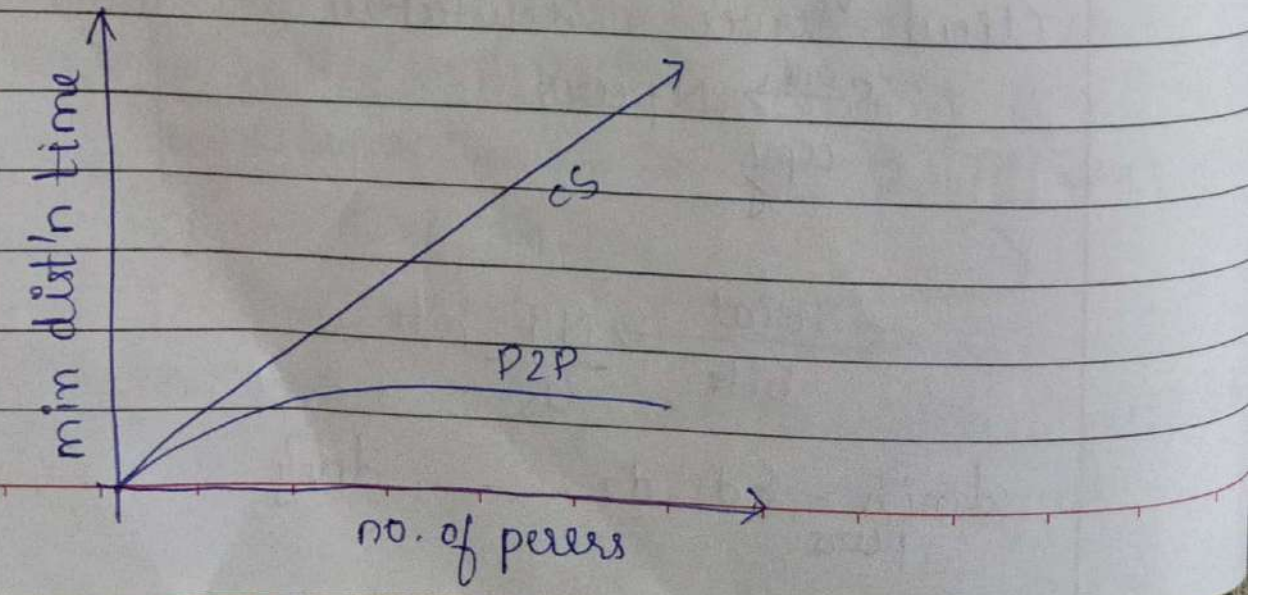
$$D_{max}^{cs} \left\{ \frac{NFbits}{u_s}, \frac{F}{d_{min}} \right\}$$

Distribution Time Comparision

Peer to Peer architecture

$$D_{P2P} \geq \max \left\{ \frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i} \right\}$$

Client servers v/s P2P



DATE: / /

ch-03 Transport layer Services and Principles [Part-02]

→ The sequence no of each segment is the number of the first byte carried in that segment

→ An application is sending some data that is broken down into segments. The segment size is defined using MSS (maximum segment size)

→ In TCP header the sequence no and the ACK numbers are the fields

401 - 600
↓
sequence number

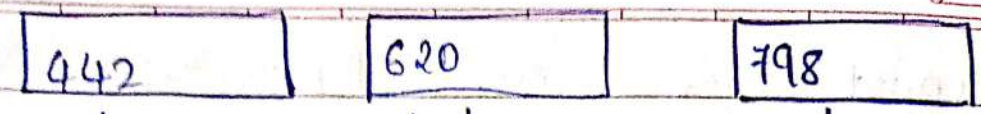
→ ACK no → next expected byte 501

→ The TCP send & recieve buffer may impact the performance of WAN data transfers

→ TCP is in-order delivery example
Out-of-order segments are not delivered.

Ex: 442 initial sequence no & first three segments each contain 178 bytes

DATE : / /



S1 S2 ACK1 S3 ACK2 ACK3 - x

1 2 3

But second is not reach

ACK-1 620

ACK-2 x

ACK-3 620

RTT estimation and time out

What to do if segment transmission fails? Till what time to wait?

→ Once timeout occurs we re-transmit

Timeout: * Not too long // delay
* Not too short // retransmissions degrade performance of network

RTT is total time taken to reach destination and return back to source

Timeout is dependent on RTT

DATE: / /

* It should be longer than RTT value
  → but RTT varies
* Too short : premature timeout, unnecessary retransmissions.
* Too long : slow reaction to segment loss

So we use exponential moving average

EMA
→ EMA is used to evaluate the trending direction over a period of time
→ EMA gives more weight to data that is more current

how to estimate RTT?
Sample RTT: Measured time from segment transmission until ACK receipt
→ ignore retransmissions

Sample RTT will vary, want estimated RTT "smoothie"
average several recent measurements, not just current sample RTT

PAGE NO.:
DATE: / /

Estimated RTT = (1-α) * Estimated RTT + α * SampleRTT

α = 0.125

Timeout interval = Estimated RTT + 4 * DevRTT

Estimated RTT — Safety margin

DevRTT: Is a measure of the variability of the RTT

DEVRTT: (1-β) * DevRTT + β * |sampleRTT - EstimatedRTT|

β = 0.25

Congestion Control

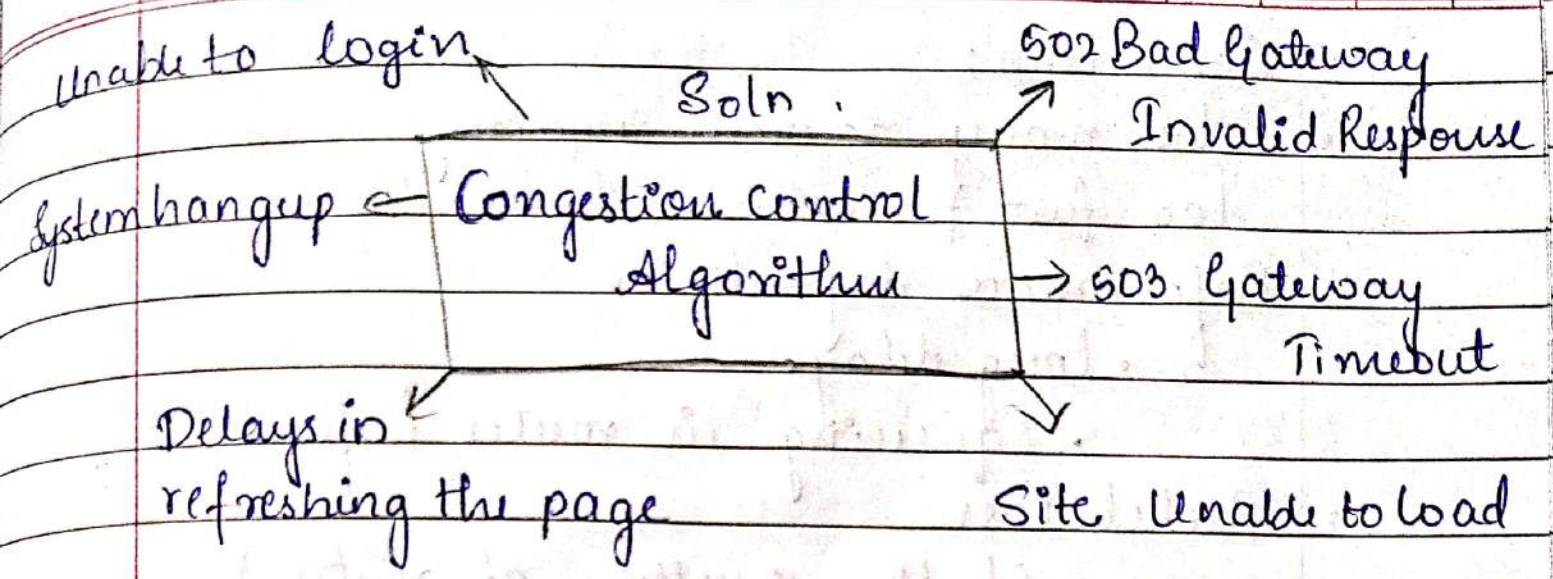
Reason:

load > Capacity

No. of packets sent to network > No. of packets a network can handle

Quiz: In fast retransmission the no of ACK messages is 3

PAGE NO.:
DATE: / /



Sender rate $\propto$ cwnd

$\propto \frac{1}{RTT}$

In slow start,
→ double cwnd every RTT
→ done by incrementing cwnd for every ACK recieved

DATE : / /

Congestion:

- "too many sources sending too much data too fast for network to handle
- Performance degradation
  - long delays
    - (queuing in router buffers)
- Packet loss
  - (buffer overflow at routers)
- top-10 problem

→ feedback via → Receiver
→ Direct network feedback

* Sender limits transmission:

LastByteSent - LastByteAcked ≤ cwnd

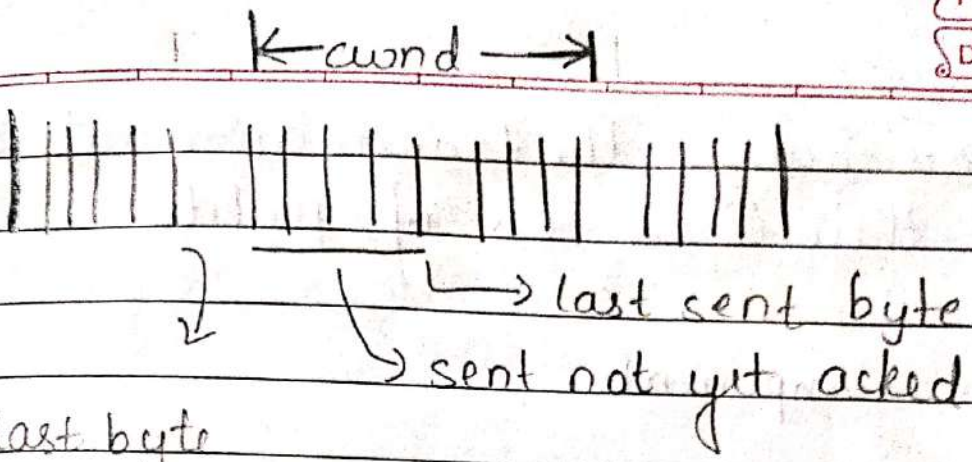
cwnd → congestion window

- cwnd is dynamic, function of perceived network congestion

Sender rate ∝ cwnd

∝ 1/RTT

sender sequence number space



$$\text{rate} \approx \frac{\text{cwnd}}{\text{RTT}}$$

TCP congestion control : Additive Increase Multiplicative Decrease (AIMD)

Approach : Sender increases transmission rate (window size), until loss occurs.

Additive Increase — cwnd - Increase by 1 MSS

AIMD multiplicative decrease — cwnd - Decrease cut by 1/2 after loss

This gives the saw tooth behaviour

DATE : / /

Sender changes the transmission rate such that there is no loss of packets.

3 Major Components

1) Slow Start (SS)

Increase the rate exponentially

cwnd = 1.

ssthresh = $\frac{cwnd}{2}$

start cwnd = 1 $\rightarrow 2^0$

After 1 RTT = 2 $\rightarrow 2^1$

2 " = 4 $\rightarrow 2^2$

3 " = 8 $\rightarrow 2^3$

2) Congestion Avoidance (Additive Increase)

cwnd = ssthresh

start cwnd = i

After 1 - RTT = i+1

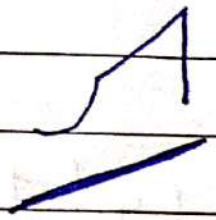
2 - " = i+2

3 - " = i+3

DATE: / / 6

3. Congestion Detection / Fast Recovery
   3. Duplicate ACK's (multiplicate decrease)
   Time Out (cwnd = 1MSS)

TCP Tahoe
TCP Reno



i) SS
→ double cwnd every RTT
→ done by incrementing cwnd for every ACK received

Network layer

1) IP addressing / logical addressing
2) Routing.

IP - 32 bit address (4 bytes)

8.8.8.8
0-255

192.168.35.198

192.168.35 → Network Address
198 → host name / host address

In hierarchy

* The IPv4 addresses are unique and universal
* The address space of IPv4 is 2^{32} or 4,294,967,296

PAGE NO.:
DATE: / /

	Byte 1	Byte 2	Byte 3	Byte 4	Type
class A	0-127	-	-	-	End hosts
class B	128-191	-	-	-	End hosts
class C	192-223	-	-	-	End hosts
class D	224-299	-	-	-	Multicast Address
class E	240-255	-	-	-	Research / future space

Dotted-decimal notation.

	Net Id	Host Id
class A	8	24
" B	16	16
" C	24	8
" D		
" E		

Ex: 192.168.35.254
Net
class C
1 host

Net Host
10.2.0.10
class A

Special LAN Addresses

IP	Address Range
class A	10
class B	172
class C	192

⟶ Rdt 1.0

Assume No bit errors
No loss of packets

Sender: Hello Reciever: Hello

⟶ Rdt 2.0
with bit errors

→ Need Error detection
→ Reciever feedback
→ Retransmission

Sender: Hello Reciever: Hellᴖ

Control msg:
ACK -> +ve acknowledgement
NAK -> -ve "

If error retransmission happens but dupes may occur so we do for seq no added to packet no (1-bit info 0 or 1)

Protocol: Stop & Wait
↓
Sender stop sending data & wait for response from reciever

→ Rdt 3.0 ARQ - Automatic Repeat Request

Packet loss.

Sender: Hello Reciever: Hell

Retransmission

↳ How long to wait? Timeout

↳ Dupes received?

RDT:

FSM any

Machine can be host which performs finite no. of actions on occurence of an event causing the transition in the state

Rdt 1.0

Two separate FSM for sender & reciever each with only one state

Application layer

→ Voice over IP and video conferencing - Skype, Facetime, Google

→ User generated video such as YT and movies on demand such as Netflix

→ Road Traffic forecasting apps (Yelp, Tinder, Waz and Yik-Yak)

→ Web applications there are 2 distinct programs
- one the browser program running in the user's host
- Web server program running in the Web server host.

→ In P2P applications
- A program in each host that participate in file sharing community

The two significant architecture paradigms
1) Client Server architecture
2) Peer to Peer (P2P) application

Client Server Architecture
→ Always on host called server serves requests for many other hosts, called clients
Ex: Web application, FTP, Telnet, e-mail.

P2P architecture : There is minimal or no reliance on dedicated servers in data center. Instead the appl'n exploits direct communication b/w pairs of intermittently connected hosts, called peers.
Best suited for traffic intensive apps
Ex: File sharing (bit torrent)

peer-assisted download acceleration (Xunlei)
Internet telephony and video conference (Skype)
→ Self scalability
3) Hybrid architecture in instant messaging app
→ Client-server - track IP addresses of users
→ P2P - for user to user messages are sent directly b/w the host (without passing through intermediate servers)

P2P merits:
1) Self-scalability
2) Each peer generates workload by requesting files they also add service capacity to the system by distributing files to other peers
3) Cost effective (as they don't require server infrastructure and server bandwidth)

Demerits:
Security, performance, reliability due to this highly decentralized structure

→ In the context of a communication session b/w a pair of processes, the process that initiates the comm'n is called the client. The process that waits to be contacted to begin the comm/session is the server

Socket -> A process sends messages into, and receives messages from, the network through software interface called a socket. It is an interface b/w appl'n & transport layer within a host. It is an API b/w the appl'n & network

The application developer can decide
(i) transport protocol
(ii) transport layer parameters - max buffer & segment size
(iii) Once the appln developer chooses a transport protocol the appln is built using the transport layer service provided by that protocol.

→ Receiving host as well as receiving identifier are required for addressing

To identify of a process
(i) the address of the host (IP addrs (32 bit - unique)
(ii) the identifier specifying the receiving process in the destination host (Socket no)

↓ : In general the host may be running many network applications

Web Server - 80
Mail Server - 25 (using SMTP)

The appln at sending side pushes msgs through the socket the transport layer protocol has the responsibility of getting the msgs to the socket of receiving proce.

The services from transport layer protocol:
(i) reliable data transfer
(ii) Security
(iii) throughput
(iv) timing

Reliable data transfer

→ The data sent by the sending process is delivered correctly and completely to the other end of appl'n.

If there is no reliable data transfer - it is acceptable for loss-tolerant appl'ns Ex: Multimedia appls

Throughput: In context of comm session b/w 2 processes along a network path, is the rate at which the sending process can deliver bits to the receiving process.

Ex: Bandwidth-sensitive appl'ns (Multimedia)

Elastic appl'ns:- Electronic mail, file transfers, web transfers

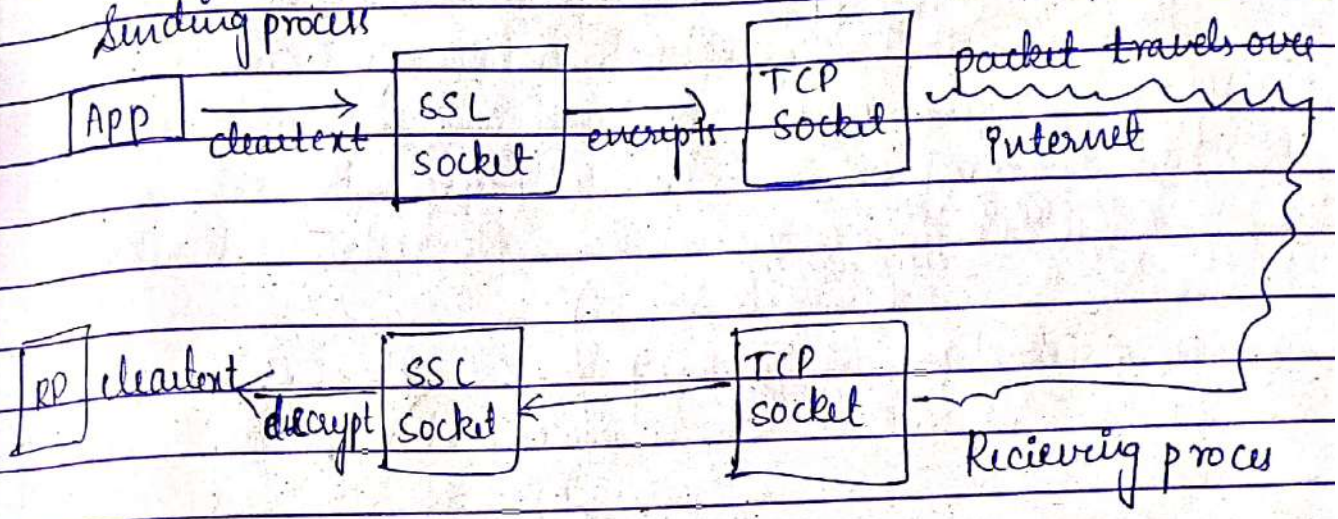
Timing - Interactive real time applications such as Internet telephony, tele conferencing, multiplayer games virtual environments.

TCP: 1) connection oriented service
2) Reliable data transfer
3) Congestion control mechanism (throttles sender)

UDP: 1) connection less
2) non-reliable

SSL: enhancement of TCP → Security Socket layer That provides data integrity, end to end security, encryption, process to process security, and end-point authentication It is implemented at appl'n layer

SSL code must be in both servers and client sides of appl'n
SSL also has its own socket API



UDP : 1) no frills
2) lightweight
3) connectionless // no handshaking
4) Unreliable data transfer
5) minimal services.
6) No congestion control mechanism.

No timing or throughput guarantees

TCP → Email, remote terminal access, Web, FTP
UDP → Internet telephony (Skype)
↘ designed for TCP with UDP as backup as firewalls block UDP traffic.

Appl'n	Appl'n layer Protocol	underlying
Electronic Mail	SMTP	TCP
Remote terminal access	Telnet	"
Web	HTTP	"
File Transfer	FTP	"
Streaming multimedia	HTTP (YT)	"
Internet telephony	SIP, RTP, proprietary (Skype)	TCP/UDP

HTTP

* Run on TCP
* Request & response headers are within the same TCP connection
* Send's its control info in-baud
* Server is stateless

FTP

* Run on TCP
* 2 parallel TCP connections for file transfer i) control connection & //P-21 ii) data connection //P-20
* Send's its control info out-of-band
* It maintain state about user
  → associate control connection with specific user account
  → Keeps track of user's current directory

Commands in FTP are sent as 7 bit ascii format
Replies are 3 digit numbers

SMTP : Components

i) User Agents ii) SMTP iii) Mail Servers

MS outlook, Apple Mail

Uses TCP & persistent connection

both client (send) & server (receive)

Body of all mail messages - 7 bit ascii

Commands in FTP: USER, PASS, LIST, RETR fn, STOR fn

331 - Username OK, password required
125 - Data connection already open
425 - Can't open data connection
452 - Error while writing to file

SMTP

* Port no - 25
* persistent connections
* Push protocol
* Each msg, including the body of msg must be in 7-bit ASCII. if not it should be encoded while sending & decoded while receiving
* It places all the msg objects into one msg.

HTTP

* Port no - 80
* persistent connections
* Pull protocol
* Each msg, including the body of msg to be 7-bit ASCII else - No restrictions as such.
* Encapsulates each object requested in its own response msg

Chapter-02: Application Layer

DHCP

1) Host dynamically obtain IP address from network servers when it "joins" network
2) DHCP discover is broadcast in nature (both to DHCP server & client)
3) DHCP uses UDP protocol
4) DORA process.
5) Port 67 is used for sending data to serve
6) Pool of IP addresses can be reused
7) Host is any machine/device connected to internet.

8) DHCP broadcasts work :: DHP server accepts the data as 67 port is open whereas in other clients machine 67 port is closed :: DHCP server is configured with 67 port as open.
9) DHCP along with IP address also provide:
   1) Subnet mask
   2) address of first-hop router for client (default gateway)
   3) Name and IP of DNS server.
10) No of IP addresses the IP pool can take is dependent on mask.

11) To automate configuration of host device we use DHCP.

12) Client-68

HTTPS

1) Hyper text transfer Protcol secure.
2) Extension of HTTP.
3) Used for secure com
4) Encrypted using Transport layer Security / SSL
5) Ruponse and requests are encrypted
6) Port number - 443 by default
7) It runs on top of TCP
8) Client / Server model

SSH

1) Security Shell
2) Default port - 22.
3) It allows tunneling to safe network / server.
4) Encryptic network protocol.
5) Client server model.
6) Remote administration protocol that allows users to control & modify their remote serves over Internet.
7) It uses authentication to remot user
8) A public key - for encryption
9) A private key - for decryption.
10) data integrity, encryption, end to end security

SSL

1) Security Socket layer
2) Encryption based internet security protocol
3) Ensures privacy, authentication, and data integrity
4) Port No - 443
5) Client - Server.

Bandwidth of one user = Total Bandwidth / No of users.

SMTP

1) Port no - 25
2) Components = User agents, mail servers, SMTP
3) Principal application layer protocol.
4) TCP is used for reliable data transfer.
5) Three phases:
   i) Handshaking → Command "HELO" from client
   ii) Transfer of msg → Command "DATA"
   iii) Closure → Command "QUIT"
6) Body of all mail msgs = 7 bit ASCII
7) Mail servers act both as client & server

IMAP

1) Internet Mail Access protocol
2) Port - 143
3) More features.
4) Complex

POP3 (+OK, -ERR replies)

1) Post Office Protocol
2) Port - 110
3) Less features
4) Simple → 3 phases: Authorization, Transaction, Update

8) Handshaking → Sender address, Recipient address.
9) 220 - Server name, 250 - intro, 250 - OK

SMTP	HTTP
* Port - 25	* Port - 80
* Persistent connections	* Persistent Connections
* Push protocol	* Pull protocol
* 7-bit ASCII else encode	* No restrictions
* All msg objects in one msg	* Encapsulates each requested object in its own response msg
* HELO, DATA, QUIT	* GET, POST, PUSH

POP3

1) Port - 110
2) Uses TCP and simple
3) Client Server
4) Three phases:- i) Authorization ii) Transaction iii) Update
5) Post office Protocol
6) Commands: User agents give commands
   i) list
   ii) retr
   iii) dele
   iv) quit
7) After quit server enters update phase
   b) If download & delete mode then delete in server
8) Server replies : +OK, -ERR
9) Maintains state of user during session not across sessions

IMAP

1) Port - 143 Internet mail access protocol, Complex
2) It provides means to create remote folders and assign messages to folders
3) delete msgs, move msgs to folders, search msgs
4) Commands to ↑
5) Maintains user state info across IMAP sessions
   Ex: names of folders & msgs associated.
6) Commands to obtain components of msgs.

FTP

1) Port No- 20 (control) 21 (data)

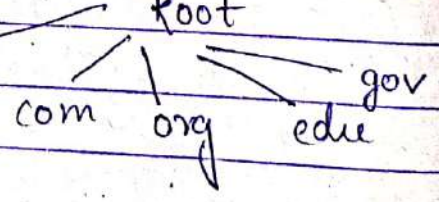
2) TCP

DNS

1) Port - 53 Domain name service.
2) translate domain name to IP address, CS
3) UDP, Appl'n layer protocol
4) they are UNIX machines running BIND software.
5) people - host name Routers - IP
6) Services provided:-
   (i) A distributed database implemented in hierarchy of DNS servers
   (ii) appl'n layer protocol that allows hosts to query the distributed db.
   (iii) Host aliasing
   (iv) Mail server aliasing → obtain canonical name & IP
   (v) load distn.

Problems with centralized DNS
   (i) Single point of failure
   (ii) Traffic volume
   (iii) Distant centralized db
   (iv) Maintainence
   (v) Doesn't scale

Three classes of DNS server
   (i) Root DNS servers
   (ii) TLD " "
   (iii) Authoritative " "

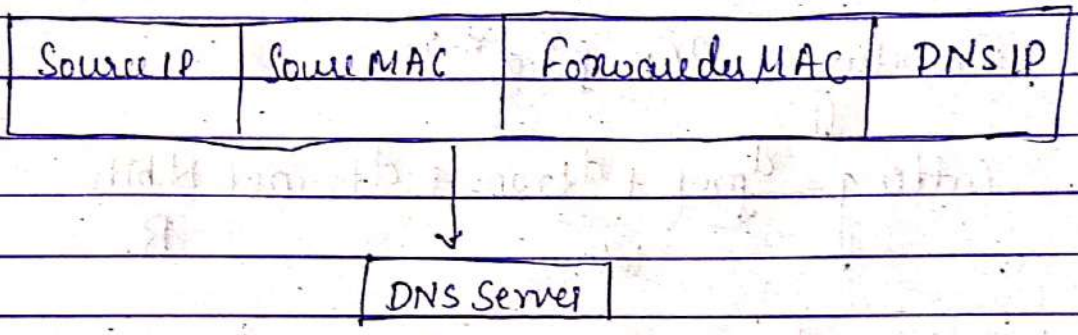


Local DNS server - Unnecessary bandwidth connections are reduced
DNS cache.
Resource Record - Name, Value, Type, TTL
   ↳ RR provides hostname - IP address mappings

Type	name	value
* If Type = A	hostname	IP address
* If Type = NS	domain	hostname of an authoritative DNS server.
* If Type = CNAME	aliashost	canonical name/hostname
* If Type = MX	alias host	" " of mail server

A - authoritative

* Initial UDP query then TCP query



If UDP packet crashes then reliable TCP

* TTL - No of hops / TTL
<RD length> <RD data>

* NS - subdomain
* A - parent domain
* CNAME - aliasing
* MX - mail server

* Distribution time

$$D_{CS} = \max\left\{\frac{NF_{bits}}{u_s}, \frac{F_{bits}}{d_{min}}\right\}$$

$$D_{P2P} = \max\left\{\frac{NF_{bits}}{u_s + \sum_{i=1} u_i}, \frac{F_{bits}}{d_{min}}, \frac{F_{bits}}{u_s}\right\}$$

Transmission delay = $\frac{L}{R}$ $\frac{\text{bits}}{\text{b/s}}$

Propagation delay = $\frac{d}{s}$ $\frac{m}{m/s}$

Bandwidth per user = $\frac{\text{Total bandwidth}}{\text{No of users}}$

Probelity = ${}^{n}C_{n}\, p^{n} q^{n-r}$

Cutting = $\underbrace{d_{prop} + d_{trans}}_{\text{latency}} + d_{trans} + \frac{N \text{ bits}}{R}$

Delay = d proc + d queue + dtrans + d prog

Queuing delay = $\frac{La}{R}$

a - average packet arrival rate
R - link bandwidth

* Delay does not depend on packet length / transmission rate

* No. of packets transmitted = R/L

* End to End delay = N (d proc + dtrans + d prog)
↳ Links

* Link utilization = $\frac{R \text{ bottleneck}}{R \text{ link}}$

* Width of bit in link = $\left(\frac{1 \text{ bit}}{R}\right) * s$