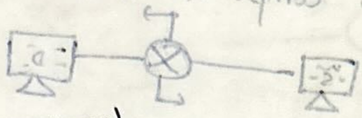
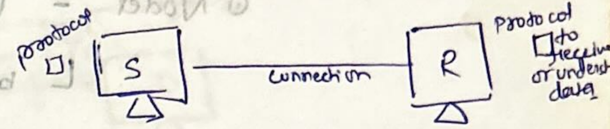
In an OS machine OS makes the connection of communication smoother so, with CN we had to make smooth communication b/w two other devices.

System Carrier Information | commodity among two or more entities.





Transport — Computer

Vechile — packet | payload

dest. ~~Add~~ — IP ~~Address~~

path — Routing Algo (software)

Intersection — switches | Router

Traffic post — flow control

~~Accident~~ — collision of packet

Routing Algo is a procedure that lays down the route or path to ~~transfer~~ transfer data packets from source to the destination

router is a device that connects two or more packet-switched networks or subnetworks.

flow control regulate the data transfer b/w comp & other nodes in a network.

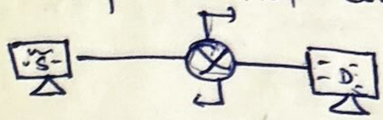
Basic Requirement for Communication.

- Enddevices — a sender device, a receiver device
- Medium | Links — wired | wireless
  - wire — (Ethernet card) ← co-axial cable, Twisted pair, Optical fiber
  - wireless — (wificard)
- NIC (Network Interface ~~Controller~~ card) (port)

(without NIC commuction is possible only in one PC itself)

◉ Nodes - (Router | Switches)

L bcz two comp. do not connect directly.



◉ I Application to generate & send data and one Application to receive the data.

Goals of Computer Network.

-> Efficient — Less cost, delay, Energy Consumption less

-> Robust - Toward failure | error

{If my one side link got down so my total network should not be corrupted. It must be strong. robust}

i) Scalable - If my no. of user Increasing or data transfer among Increase. Still my network provide the service. It had to be sustain

* Way's to Through Which CN is going to acheive the goal.

i) Innovation of H/w

ii) Innovation of S/w

Measurement (whether the goal is acheived or not)

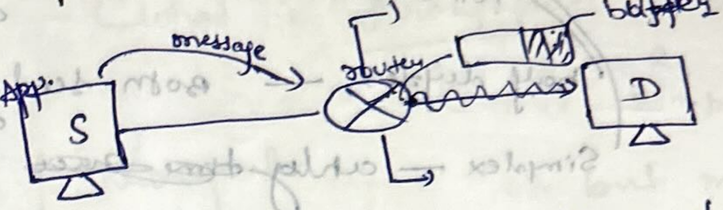
i) Throughput | Bandwith | dataRate

L bps, kbps, mbps, Gbps

Amount of data that Can transfer per second.

Delay in CN

Delay | Latency (~~Check whether it is corrupted or not~~)



Step 1:- Processing Delay :- Time elapsed in Inspecting the packet, Correct | Corrupted

Step 2:- Queuing delay :- If my processing is going on already and new packet come to it had to wait in buffer for it's turn (waiting in the Queue for it's turn to process)

Step 3:- Transmission delay (delivering the whole packet into the channel)

$$Td = \frac{\text{Size of Packet}}{\text{Bandwidth}}$$

Step 4:- Propagation ~~Transmission~~ delay :- Amount of time elapsed in delivering the whole packet in the communication link.

factor — Size of packet
- Throughput | Bandwidth of channel
- propagation delay. depend on medium is it twisted or co-axial cable and the distance also.

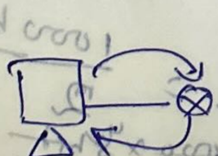
{ Time to travel any bit from one end of the channel to another. }

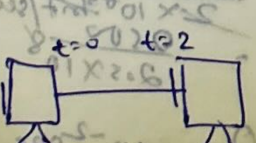
$$Pd = \frac{\text{distance}}{\text{speed of signal in medium}}$$

(1 node to 2 link)

End - End delay $= 2$ ($Proc_d$ + que_d + $Trans_d$ + $Propo_d$)

Round Trip time $= 2 \times Prop_d$
(RTT)





Max^m. no. of bit in Transit in a given channel $= \underbrace{\text{Bandwidth} \times P_d}_{\text{These are factor to depend on.}} \times 2$ full duplex

Bandwidth delay product $=$ Bandwidth $\times$ RTT

A channel.

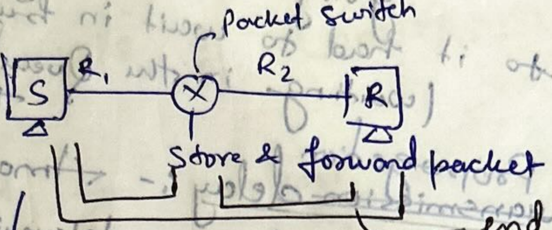
Full duplex — Both send or Receive at the same channel

half duplex — Both send or Receive but not at the same time.

Simplex — only ~~……~~ Sender or only receive

Delay Calculation Question

Q



Bandwidth — R_1 & R_2

end to end delay

100mb
$t=0$

$t = 1$ - 50m
$t = 2$ - 100 mb
it had to wait for whole data
due to store & forward packet ~~……~~ architecture

calculate ~~total~~ end-end delay

packet size = L.

If other delay we not able to find assume it is zero

$T_d = \dfrac{\text{Size of packet}}{\text{Bandwidth}}$

$$\text{Transmission delay} = \frac{L}{R_1} + \frac{L}{R_2}$$

Q Packet size = 1000 bytes distance = 2500 km

propagation Speed 2.5×10^8 m/s and transmission rate

→ bandwidth = 2 mbps.

$T_d = \dfrac{\text{Size}}{\text{Bandwidth}} \Rightarrow \dfrac{1000 \text{ byte}}{2 \text{ mbps}}$

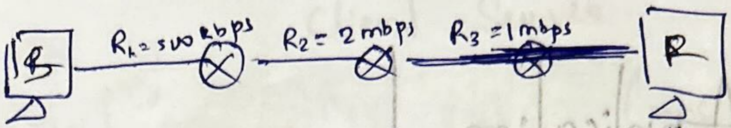
$= \dfrac{1000 \times 8 \text{ bit}}{2 \times 10^6 \text{ bit/sec}} = 4/10^3 \text{ sec} = 4\text{ms}$

$P_d = \dfrac{\text{Distance}}{\text{Speed of signal}} = \dfrac{2500}{2.5 \times 10^8}$

$P_d = 10^{-2}$ sec

$= \dfrac{2500 \times 10^3 \text{ m}}{2.5 \times 10^8 \text{ m/s}} = 10^{-2}$ sec $= 10$ ms

Q



— What is throughput = 500 kbps Ans

My R_2 & R_3 bandwidth has 2 mbps or 1mbps capability but my Applic has a capability of 500 kbps

So, 500 kbps is the data rate

Bandwidth | Throughput

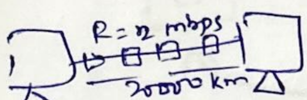
↓ Max^m rate possible

↓ current data rate

Q . Suppose the ~~Only~~ file is 4 million byte

Transmission delay = $\frac{\text{Size}}{\text{BW (current rate)}} = \frac{\text{4 million bytes}}{\cancel{\text{2 mbps}}\ \text{500 kbps}}$

Q



what is the width of a bit in a link

20000/2

5/8/23

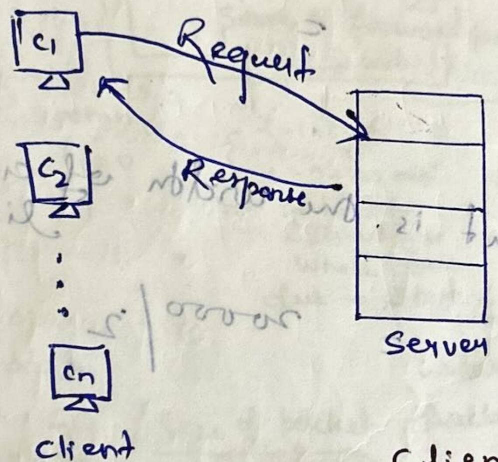
Application Layer

└ Generate Message (provide service to User)

so, Depend on Artitecture

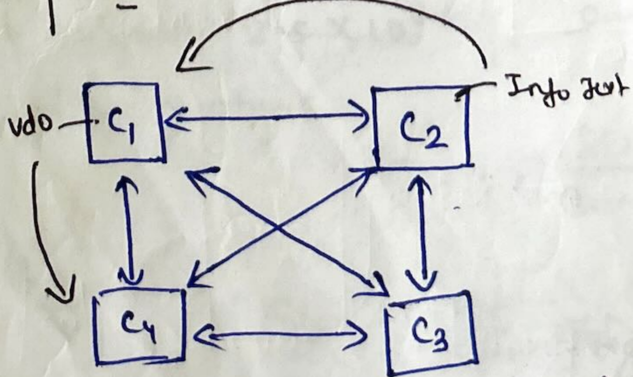
- client - Server
- pear to pear
- Hybrid

Client - Server



Client	Server
• Sent's request to get the service	• provide certain specific service to the client
• Intermentiently ON/OFF	• Always On
• dynamic ip address	• Fixed Ip Address

peer to peer



No Designated Server

- Same Computer Behave as a Server Or Same as a client

Ex :- Big Torrent, Shareit

Client - Server

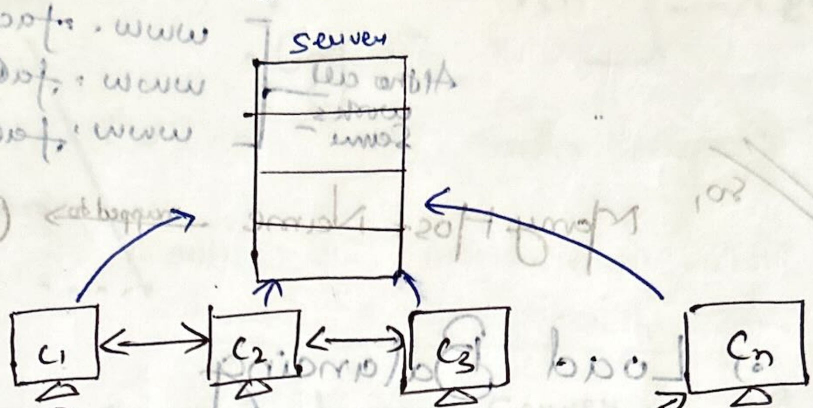
Pros

- Security
- Simple Architecture

Cons

- Single point of failure
- Infrastructure Intensive based

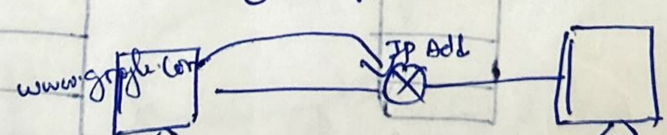
Hybrid

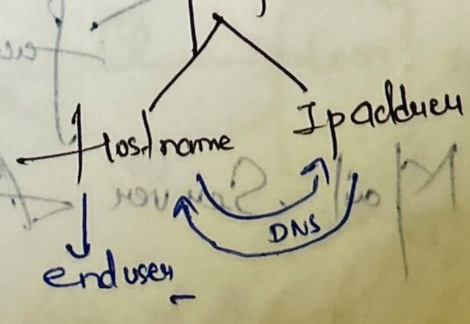


•) Initially Every client must Contact the Server.

DNS (Domain Name System)

•) DNS is the protocol used to convert hostname (www.google.com) to Ip address or Vice-Versa





Functionality provided by DNs.

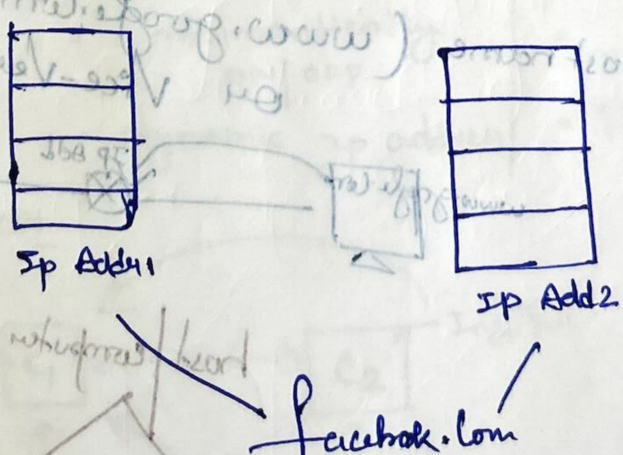
- Host $\underset{}{\overset{N}{\rightleftharpoons}}$ Ip Addrew.
- Host Aliasing (Another Name)

Altho all works Same
[www. facebook.com
[www. facebook.com
[www. facbok. com

So, Many Host Name $\xrightarrow{\text{mapped to}}$ One Ip Addrew

- Load Balancing
  - Balance ~~load~~ across the different Server distributed globally.

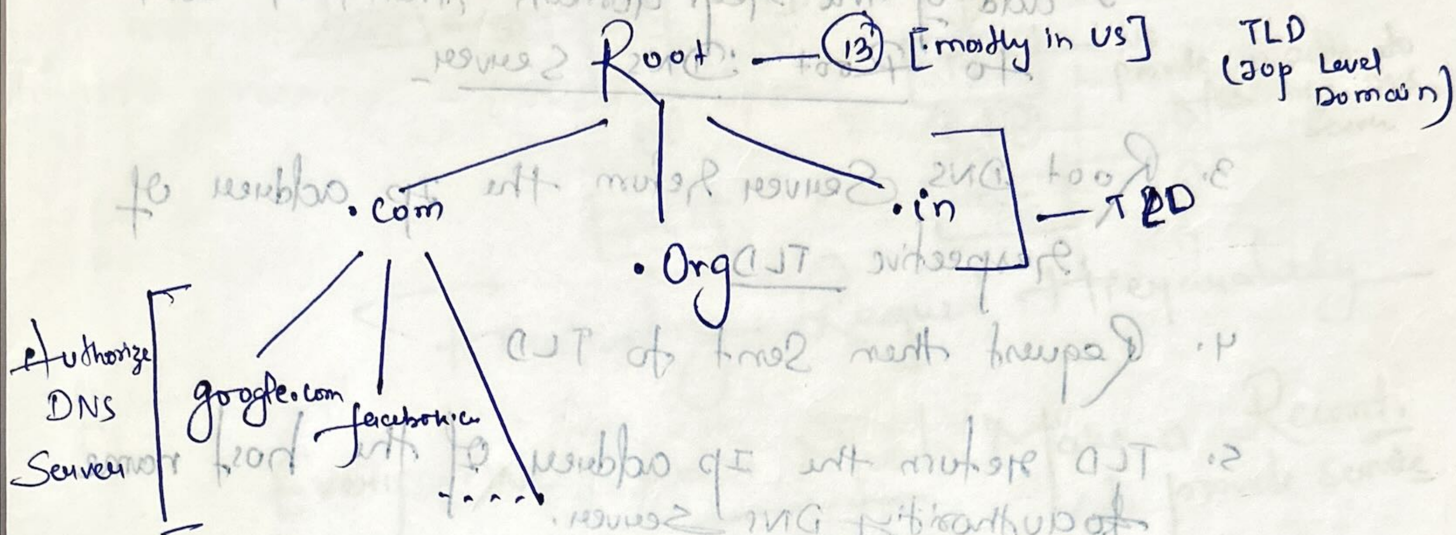
So, Many Ip Addrew ⟶ One host name



- Mail Server Aliasing

Architecture of DNS...

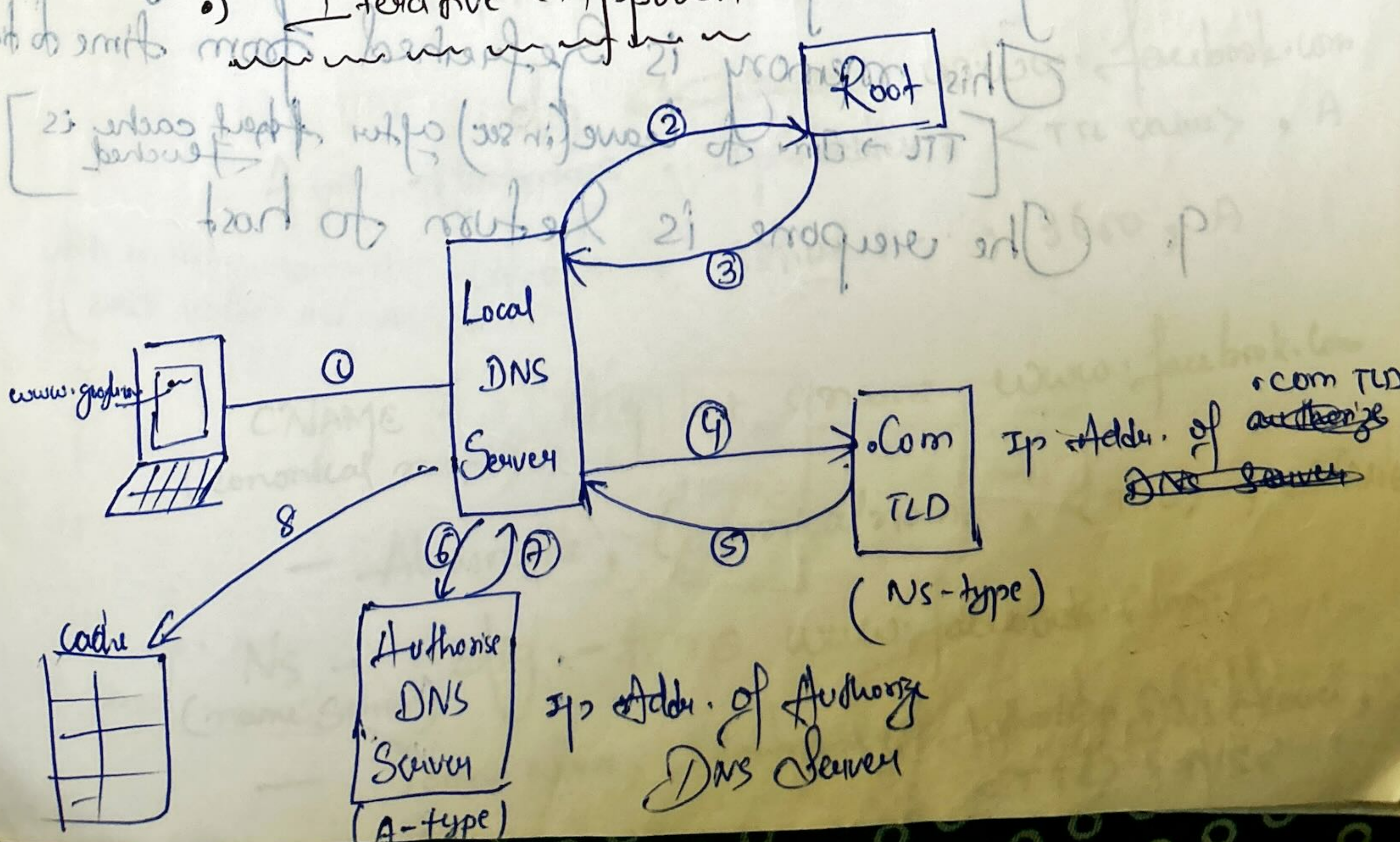
↳ Hierarchial & Distributed



Authorized DNS Server → Ip add. to Host name
maintain mapping b/w them.

How DNS Work

•) Iterative Approch



1. The DNS ~~Reqest~~ Request is Sent to the local DNS Server.

2. If ~~DNS~~ Local DNS Server does not have the IP add of the req. domain, then, it sent to Root DNS Server

3. Root DNS Server Return the IP address of respective TLD.

4. Request then Sent to TLD

5. TLD return the IP address of the host name ~~to~~ authority DNS Server.

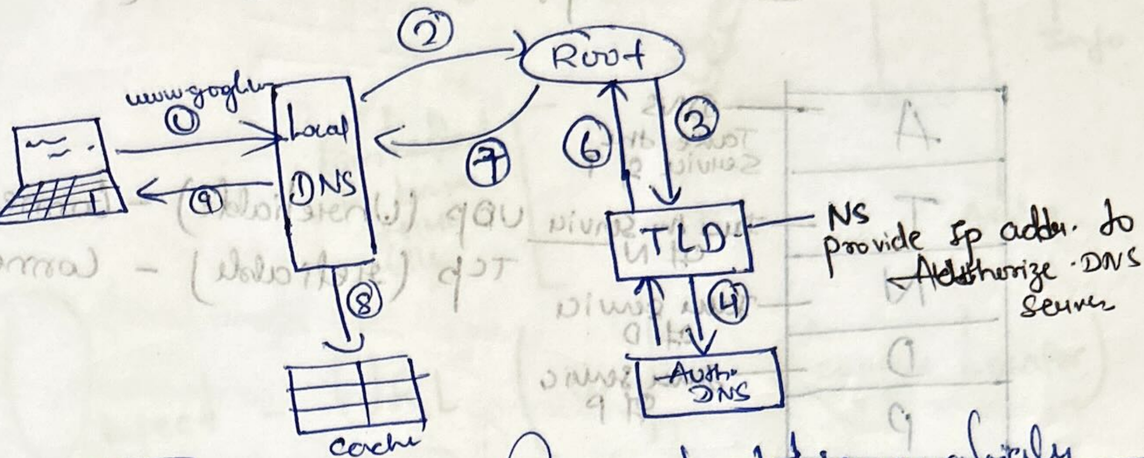
6. Request is Sent to Authorize DNS Server

7. Authorize DNS Server Return the IP address of the Desire Website

8. Local DNS Server Saves this IP address for future reference in a cache memory
   This memory is Refreshed from time to time
   [TTL → Time to Leave (in sec) after that cache is fetched]

9. The response is Return to host

Recursive Approach



Forwarding Request Hieyarahicaly

Every DNS Server Must Make a Record. To provide service

i) A - Mandatory for Auth. DNS Server

ii) CNAME - (cononical name)

iii) NS - Mandatory for TDL server.
(Name server)

iv) MX -
(Mail Exchange)

dig ~~-t cname~~ www.facebook.com

A → Hostname , Ip Address , <TTL value> , A

google , 187.121.6 . . , 3600 , A

CNAME → dig -t cname www.facebook.com
(cononical name)

— Aliasname , Conomicalname , <TTL> , CNAME

NS → dig -t ns www.facebook.com
(name server)

— Domain Name , Hostname of Authority DNS Server , <TTL> , NS.

DNS - Application Layer (Client-Server Architecture)

A	— DNS Take the Service of T
T	— Take the service of N — UDP (Unreliable) - Connection less; TCP (reliable) - Connection-Oriented
N	— Take service of D
D	— Take service of P
P	

Q. Whether DNS will take the Service of TCP or UDP?

→ DNS

- Amount of data transfer is less then UDP is ~~used~~ prefered mostly
- Amount of data transfer is more or huge to, TCP is prefered.

So, DNS take UDP Service Commonly
└ www.google.com (less data)

but also DNS Takes TCP
└ Zone Transfer (Send data transfer from the one DNS to other DNS)

www.google.com — DNS Server

port no must know of DNS to Send packet (53)
└ Every server has well known port

→ Client - Server.
→ UDP
→ 53 port no. of DNS

Web & http protocol

A web page is Consisting of a base HTML file

Objects — Another HTML file, Video, Audio, Info

Embedded to base HTML file

Object - URL (uniform Resource Locator)

URL ex:- https:// www.kiit.ac.in / images / logo.jpg

$\underbrace{\text{https://}}_{\text{prototype}}$ $\underbrace{\text{www.kiit.ac.in}}_{\text{Host name}}$ $\underbrace{\text{/images/logo.jpg}}_{\text{path}}$

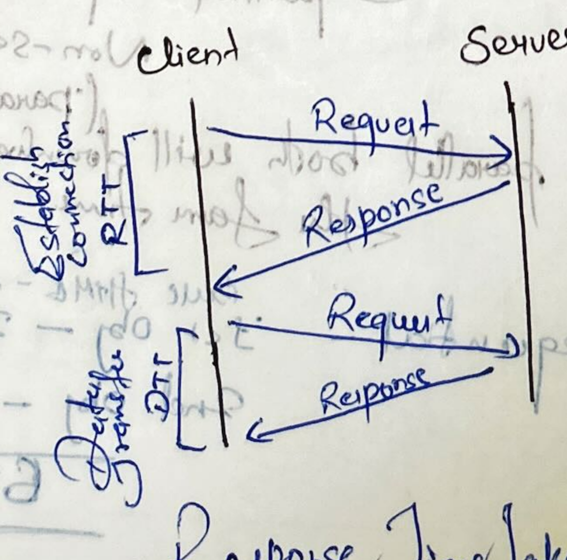
Three Components URL

http
- http 1.0 → non-persistant in nature
- http 1.1 → persistant in nature or Both — Default

Arch - client - Server

Transport Layer - TCP (while downloding large file lot of audio, video

port no - 80 | 433 - https

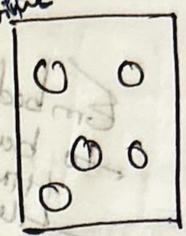


Response Time Taken = 2RTT

suppose,

webpage

base html



(Two types of connection used in HTTP)

2 way's to download the Object

persistent – [i) By making one connection download all the file Or Obj using Same Connection.

Non-persistent – [ii) Different Connection for Downloading the object.

Q

Base html

- 2 obj is embedded with base HTML
- Calculate Response Time Taken.
- For Downloading,

BaseHTML – 2RTT

Follow Non-persistent

Sequential

Non-sequential (parallel)

In parallel both will download at the same time.

In Sequential –

Base HTML – 2RTT
1st obj – 2RTT
2nd obj – 2RTT

6RTT

Non-sequential Base-HTML – 2RTT
all obj – 2RTT
4RTT ✓

Follow persistent

In Sequential – So, connection is established
Base HTML – 2RTT
1st obj – 1RTT
2nd obj – 1RTT
4 RTT ✓

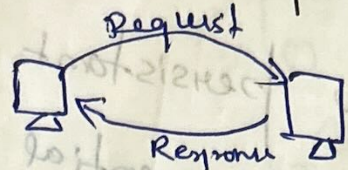
In Non-Sequential Base HTML – 2RTT
all obj – 1 RTT
3RTT ✓

Non-persistent

Corn – Time taken dely is more

- Ex:- Bank is Non-persistent
- No longer connection establish
So, it is more secure but it take more time for tranction.
- facebook is persistent

Https Message Formate.



Two type of Message are there :-

- Request
- Response.

HTTP — 1.0 - non-persistent
— 1.1 - Both but by default persistent.

Request Line

Get/~chebroul/HTTP/1.1 —

Method	sp	URL	sp	Version	cr	lf

Now add some extra info to travel this message by Transport Layer or Data Link... but some Header info must we have to add

Host: www.cse.iitb.ac.in —

Header field name:	sp	value	cr	lf

User-agent: Mozilla/5.0 —

Header field name:	sp	value	cr	lf

Connection: close — bcz we want a Non-persistent data but HTTP 1.1 is by default persistent so we had to break to make it non-persistent

~~Accept-lang~~

Accept-language: fr —

Header field name:	sp	value	cr	lf

To distinguish b/w header and body by making two line gaps

cr	lf

Entity Body

content is null if we use get

* Type of Method in HTTP Request :-

Get → Request a document from the Server

Head → Request Info about a document

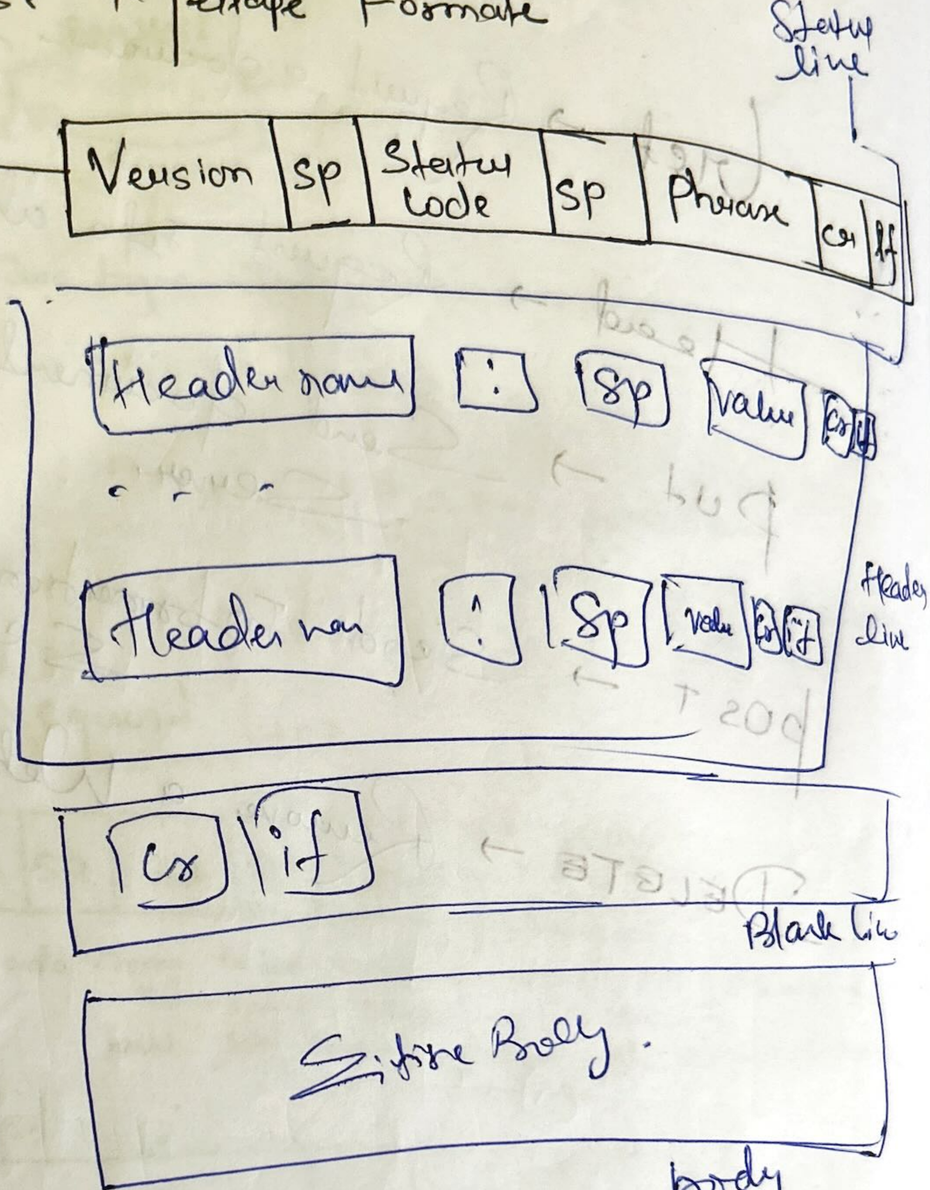
put → Send document from client to Server.

post → Send Information from client to Server.

DELETE → Remove a Webpage.

Response Message Formate

HTTP/1.1 200 OK



Additional info from server to client
ex:- data/document

CN LAB

Our Goal of a Computer is to be efficient, Robust & Scalable

so, How to design a Computer Architecture To acheive these functionality.

postal Network

component — functionality

Hostel — Accomodate student

Student — Generate Letter

Letter — Contain Information

office boy — Multiplexing | Demultiplexing

postman — (i) Decide the path of letter (ii) hop-hop communication

Infrastructure — physical transfer of letter

Computer Network (TCP|IP)

Component — functionality

Computer — Accomodate Application process] Application Layer

Appl. process — generate message] Application Layer

Message — Information] Application Layer

Transport software — Multiple | Demulti., process-process comm. (port no)] Transport Layer

Network | Routing software — Decide the route, host-host | end-end comm (IP Address)] Network Layer

Switch | Router — hop-hop comm (mac-address)] data-link layer

Infrastructure cable/air — physical Info. Carry (signal)] Physical Layer

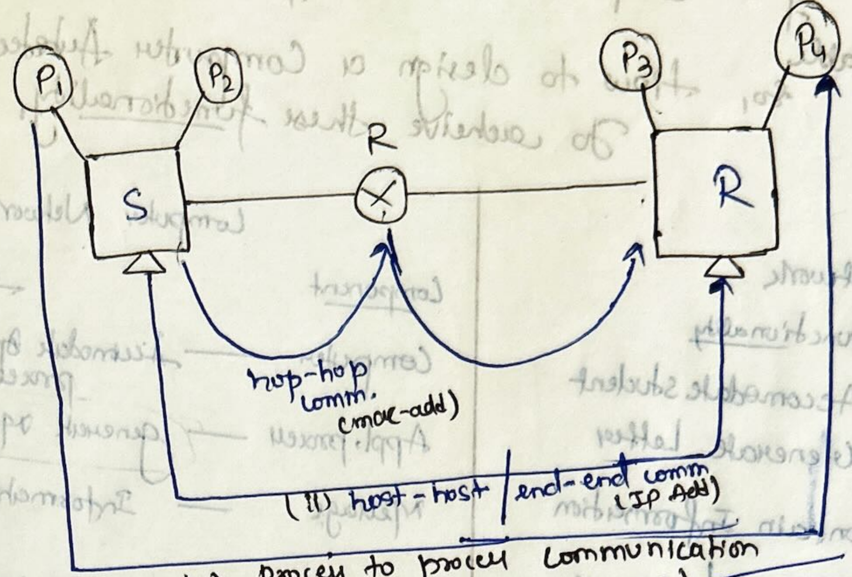
To make more Complex program easier we use divide & concor. same, In Computer Network we use Layer Architecture

5 layer are there and this layer Architecture is called TCP|IP Architecture

→ Application Layer · Transport Layer · Network Layer

· DataLink Layer · Physical Layer

① TCP|IP is a set of protocol Organised in different layer



(i) process to process communication — src port no. / Dest port no.

- by the help of (port-no)
- Done by Transport Layer

port no - To rember the destination Address. It is used to Identify the process uniquely.

[In Order to support process-process communication we had to support host-host]

ii) host-host / end-end communication

- by the help of (IP-Address)
- Done by Network Layer

iii) [In Order to support host-host communication we had to support hop-hop comm.].

iii) hop-hop comm.

- by the help of (mac-Address)
- Done by ~~Switch/Router~~ data link Layer

OSI Model

It support two more layer, altho with 5 layer

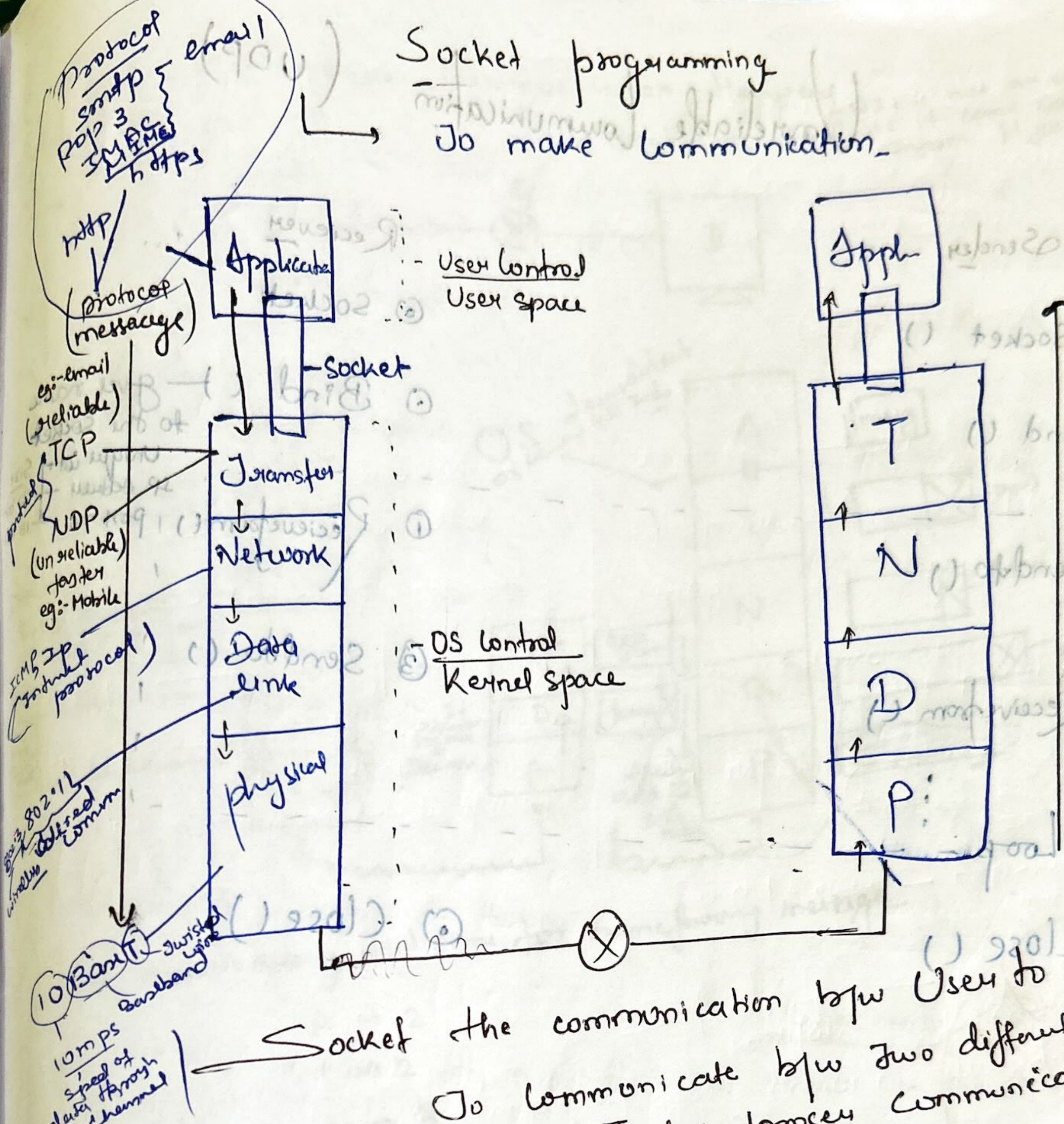
- Session Layer - Authentation
- presentation Layer - Look & feel

But we use 5 Layer Architecture TCP/IP; Here these 2 Layer are not present but altho we use these functionality

Bcz these 2 Layer are clustred in Application Layer

Socket programming

To make communication



Socket the communication b/w User to Kernel space

To communicate b/w two different process

Inter process communication

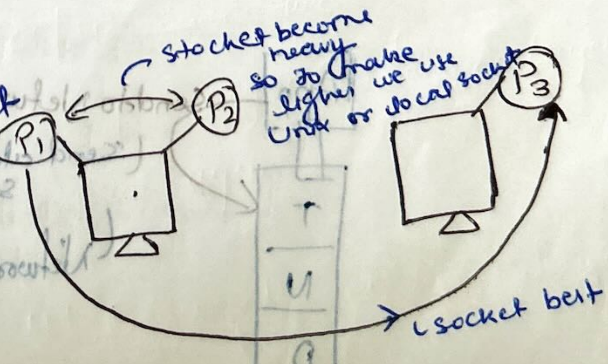
- pipe
- shared memory
- socket

Type of socket

→ Internet Socket

→ Unix or local Socket

→ X.25 Socket

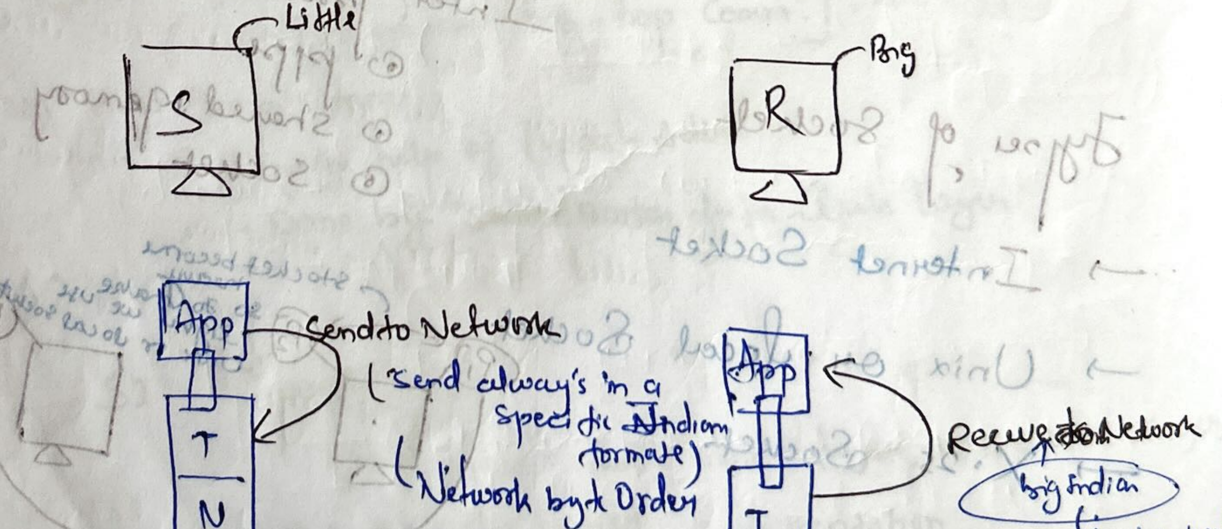


Type of Internet Socket

- Sock - STREAM — Reliable
- Sock - DGRAM — Unreliable

Unreliable Communication (UDP)

Sender	Reciever
0 Socket ()	0 Socket
0 Bind ()	0 Bind () — gives name to the socket
	Unique id — IP Address — 32 bit, port no — 16 bit (Size)
2 Sendto ()	1 Recievefrom ()
4 Receivefrom ()	3 Sendto ()
Loop	
0 close ()	0 Close ()



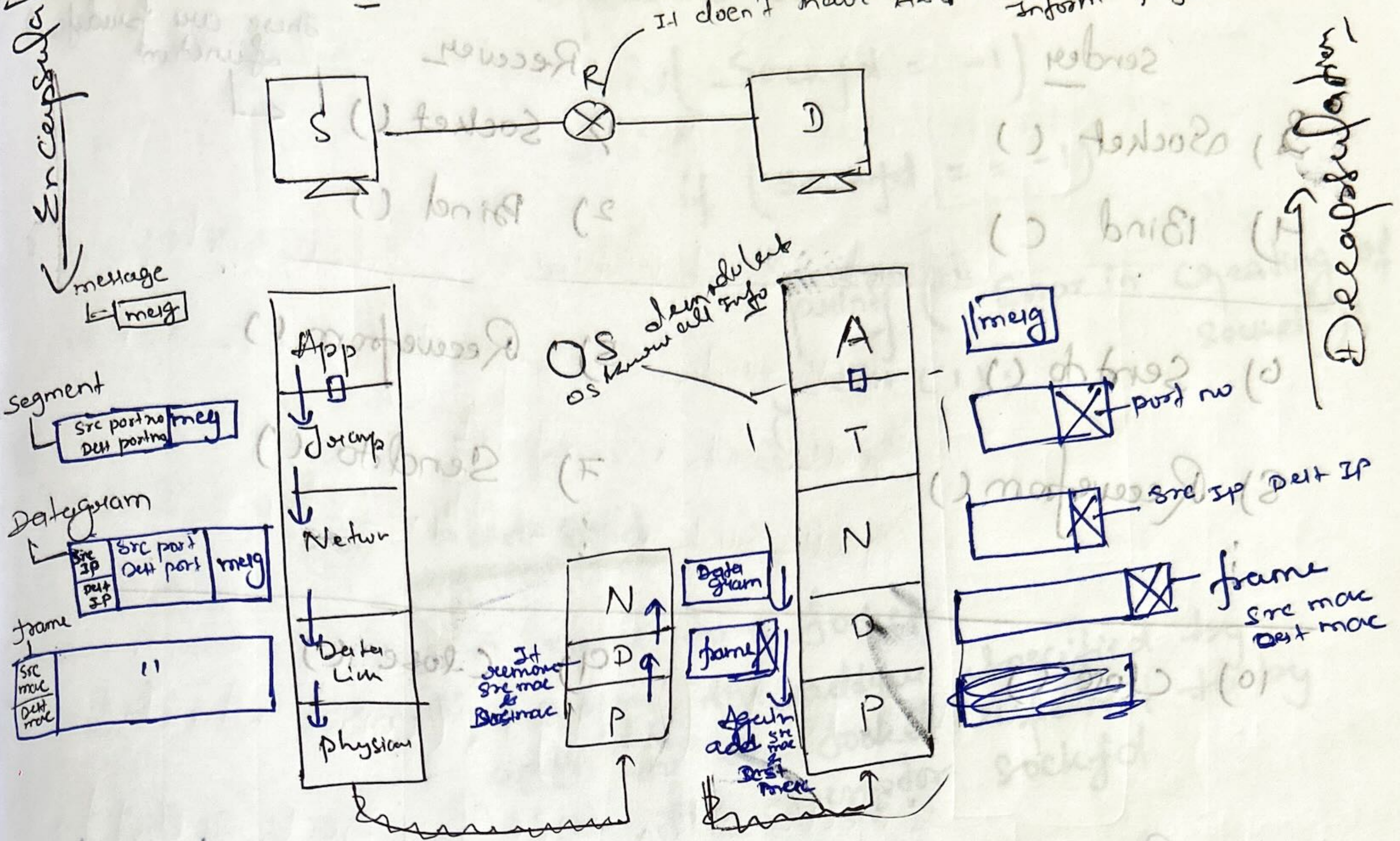
Recieve from — n to hs, n to hl

For this conversion these functionality are use

sendto — h to ns - host to network short
h to nl - host to network long

How Communication Happens

- It is not an end device bcz it cant consume the inform it just routing/forward
- It doesn't have A & T



& How many Hit time for transferring message

A → 2

T → 2

N → 3

D → 4

P → 4

5/8/23

LAB

(API)
These are predefined function

Sender	Receiver
3) Socket ()	1) Socket ()
4) Bind ()	2) Bind ()
6) Sendto ()	5) Receivefrom ()
8) Receivefrom ()	7) Sendto ()
10) Close ()	9) Close ()

Reciever

Socket → special file behind the scene.

man socket

Socket → create an end point for communication.

int socket (int domain, int type, int protocol)

domain - Specifies a communication Domain
└ AEINET

Type - Specifies communication Semantics
└ Reliable (TCP) - Sock-STREAM
└ Unreliable (UDP) - SOCK-DRAM

Protocol
└ put only '0' - OS will decide the UDP bcz other no. option are there

It Return
└ file Descriptor

int sockfd = Socket (AF-INET, SOCK-DRAM, 0) case :- Suppose file is created but no memory space is avail. then — error

if (sockfd = -1)

{ if (sockfd == -1)

{ printf("Error in creating of socket");

exit(1);

}

⊙ man Bind

Bind → name to a socket

bind() → assign the address specified by addr to the socket referred to by the file descriptor sockfd

int bind (int sockfd, const struct sockaddr *addr, socklen_t addrlen);

addr ↑ IP, Port no } — for Uniqueness

```
struct sockaddr {
    sa_family_t  sa_family;
    char         sa_data[14];
}
```

⊙ man 7 ip } To populate with generic Structure

```
struct sockaddr_in {
    sa_family_t     sin_family;
    in_port_t       sin_port;
    struct in_addr  sin_addr;
};
```

```
struct in_addr {
    uint32_t    s_addr;
};

struct sockaddr_in recv_sock;
    recv_sock.sin_family = AF_INET
    recv_sock.sin_port = 5000;
    recv_sock.sin_addr.s_addr = INADDR_ANY
        { Typecasting bcz for it is of
          sock_addr in type }
int recv = bind (sockfd, (const struct sockaddr*)
    & recv_sock, sizeof (recv_sock));

    if (recv == -1)
    {
        printf ("Error in Binding");
        exit (0);
    }
    else
    {
        printf ("no Error");
    }
}
```

① Man Recievefrom.

SSize_t recvfrom (int sockfd, Void * buf, size_t len,
int flags, struct sockaddr * src_addr,
Socklen_t * addrlen)

char recv_buf [100];
~~struct~~ int len = sizeof (src_addr);
int ~~ret~~ = recvfrom (sockfd, recv_buf, Sizeof (recv_buf),
0, for receiving the data to buffer

Size - strlen (recv_buf); — 0
sizeof (recv_buf); — 100
(It inform the OS this is the Size of buff which I mean recieve

(while writing the prog we don't know who is going to send the data to us before hand so, Donot need to populate
Os - will populate it

0, (struct socket addr *) (&src), (&len);

if (ret == -1)
{
printf (" Error statement");
exit (1);
}

Sender.c

a.out gcc sender
./a.out

Receiver.c Binary Executable
① gcc receiver.c → a.out
./a.out

Both Binary Executable are same so, It create problem

Therefore,

gcc sender. -o send
./send

gcc receive.c -o recv
./recv

TCP Communication

Client

1) Socket

Bind is optional but OS will call it

2) ~~...~~

2) connect
3) Send
4) Recv
5) close

Server

1) Socket
2) Bind
3) Listen
4) Accept

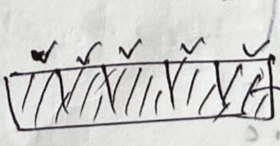
Mandatory for connection

5) Recv
6) Send
7) close

Man listen

Listen - Listen for connection on a socket (Make passive socket)

int listen(int sockfd, int Backlog);



5 connection req can be store in buffer at single point of time Otherwise it is refuse

Accept → It check in listen queue is there is any pending connection in that then it will take one and accept it

Man connect

↳ send a conn req to the server.

Ip Add & port no. of the server must be populated here

int connect (int

Man accept

int accept (int sockfd, struct sockaddr *addr, socklen_t *addrlen);

populate the Ip Add & port no. from where I get the Conn. Req.

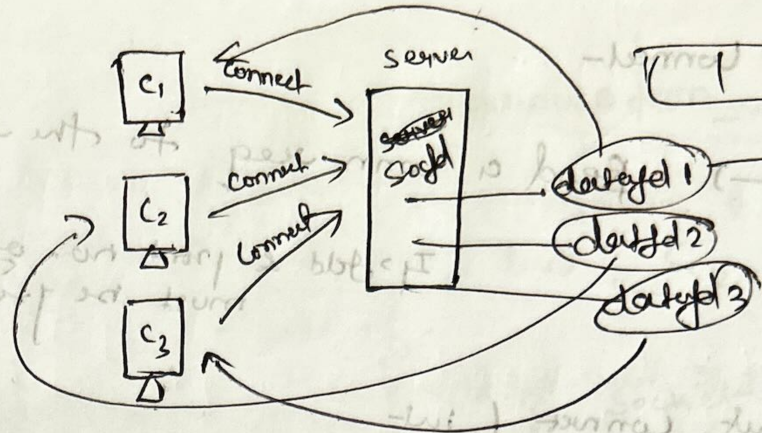
Return file descriptor

[# Socket also return file descriptor (sockfd)]

sockfd — (passive socket job is to accept conn req but not capable of communicating data)

datafd = accept (. , . . .)

(new file descriptor (It will make communicating data possible))



ip & port of client & server
So, in accept
other info not need

so, 1 sockfd & 3 datafd.

☆ Man recv

ssize_t recv (int sockfd, void *buf, size_t len, int flag);

☆ Man send

ssize_t send (int sockfd, const void *buf, size_t len, int flags);

Computer Network Type

PAN, LAN, WAN, MAN

PAN (personal Area Network)
↳ Smallest network | personal to a user
ex:- wireless comp. keyboard & mouse
. Bluetooth - headphone

LAN (Local Area Network)
↳ . provided Inside a building
. privately owned
. Connect host in office, building.
Useful for sharing Resource b/w end users
like:- printer, scanner etc.

MAN (Metropolitan Area Network)
↳ expanded throughout the city
ex:- TV cable N/w
↳ Helps an organisation to connect all it's offices in the city.

WAN (wide Area Network)
↳ May span across whole country
↳ It provide connectivity to LAN & MAN
ex:- Telecommunication N/w

Internet:- when two or more N/w are connected to one another it called an Internet.
(Network Of Network)

Internet connect all WAN's

Computer Network Topologies

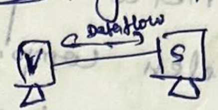
↳ It is a way computer system and network equipment connected to each other.

⊙ point to point, ⊙ BUS ⊙ STAR ⊙ RING

⊙ MESH ⊙ TREE

point to point

↳ connecting two end devices back to back using a single piece of cable



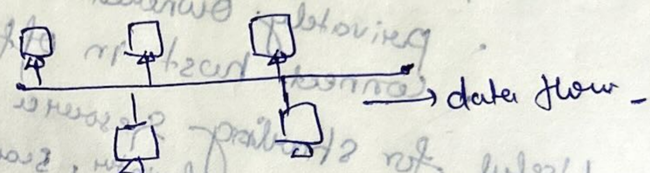
Bus Topologies

All devices share single communication line or cable

Failure of one device doesn't affect other.

Failure of communication line make all device fail

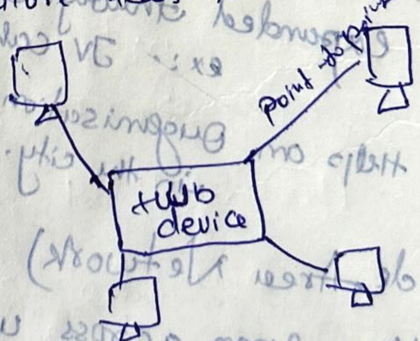
Data is sent in only one direction



Star Topologies

All devices are connected to a central device (Hub device) using point to point connection

⊙ Hub failure leads to all host failure

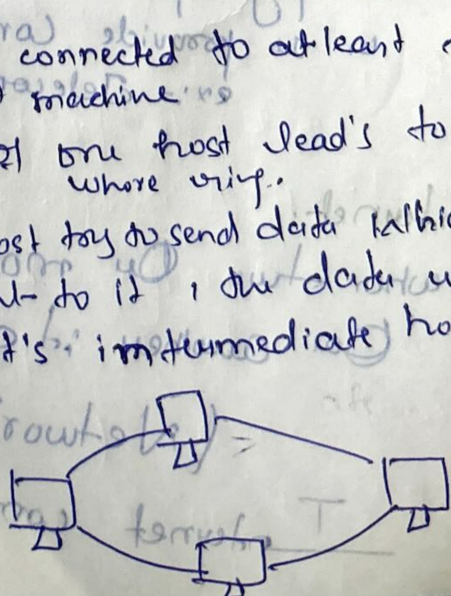


Ring Topologies

Each host connected to atleast two other host machine

⊙ Failure of one host lead's to failure of whole ring.

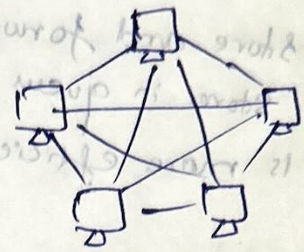
⊙ If a host try to send data which is not adjacent to it, the data will travel to all it's intermediate host.



Mesh Topologies.

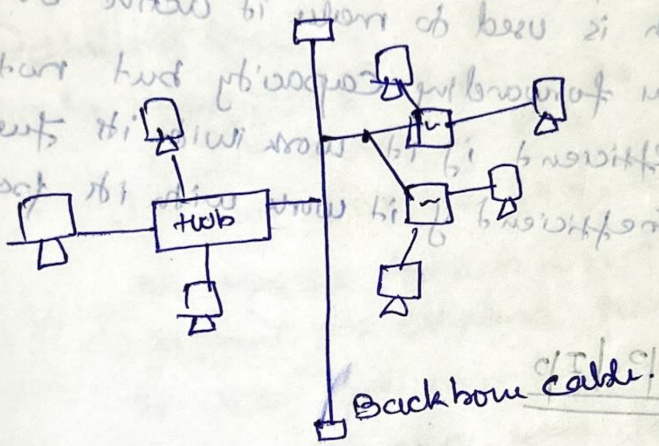
- Host may have point-to-point connection to every other host or may with few.
- provide more reliable network structure

$$\text{no. of link} = \frac{n \times (n-1)}{2}$$



Tree Topologies

- Also known as Hierarchical Topology.
- used mostly in present day
- It is the combination of ~~...~~ bus & star



Switching In CN

↳ process of forwarding packet coming from one port (to another port leading toward the destination

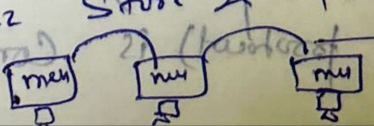
Types :-

- packet switching
- Circuit Switching
- Message Switching.

Message Switching.

- Recieve the whole ~~data~~ message, buffer it until the whole resoure are available to transfer it to next hop
- Every switch in transit path need enough storage to accomodate the entire message.

bcz Store & forward technique is used (Store full message then forward)



packet switching

- Entire message breaks down into smaller chunks called packet

Internet use packet switching Technique
It has both store and forwarding capacity
Router help to store in queue and then forward it
packet switch is more efficient than circuit switch

Circuit switching

when two nodes communicate with each other over a dedicated communication path.

ex:- Telephone

Switch is used to make it active or Inactive
It has forwarding capacity but not storing
It is efficient if it work with its full capacity
Inefficient if it work with its partial capacity.

<u>TCP/IP</u>

Application Layer

- → Communication in application layer in b/w 2 process
- protocol - DNS, HTTP, SMTP, FTP.
  - HTTP (Hyper Text transfer protocol) is a vechicle for accessing www.
  - SMTP is used in Email Service
  - FTP is used for transferring file from one host to another

Transport layer

→ It take/get message from Application layer
→ It encapsulate that message with a transport layer Header
→ Message + Header → Segment or Uses datagram
→ It given Service to Application layer
→ (Transmission control protocol) is Connection Oriented protocol
→ (User Datagram protocol) is Connection less protocol

→ TCP provide flow control & congestion control

→ UDP doesn't provide flow control & congestion control

Network layer

↳ Responsible for creating connection b/w src to dest.

→ communication to Network layer is host to host comm.

→ Router in the path chooses the best route

protocol - ICMP, IP, IGMP

IP define the format of packet

IP is connectionless protocol so, it doesn't have flow control & congestion control

Datalink layer

↳ Responsible for taking the datagram and moving it across the link.

link can be wired LAN, wired WAN or switch

No specific protocol is used here

support all standard protocol

It take Datagram encapsulate with the header of link layer ⟶ frame

This layer provide error detection & correction.

physical Layer

Responsible for carry the Individual bit in a frame across the link.

It contain a hidden layer called Transmission Media so that bit can convert in Signal before transmission.

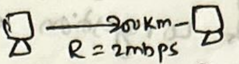
Delay Calc

$$Pd = \frac{D}{\text{Prop speed}}$$

$$Td = \frac{\text{Size of pkt}}{BW\ (DR)}$$

$$\text{Throughput} = \text{Min}^m (R_1 \ldots R_s)$$

Delay Calculation

Q 

Prop speed = $2.5 \times 10^8 m/s$

$D = 200 \times 10^3$ m

$R = 2$ mps . P speed = 2.5×10^8 m/s

a) $P_{delay} = \frac{D}{P.R} = \frac{200 \times 10^3}{2.5 \times 10^8} = 8 \times 10^{-3}$ s

b) Maxm no. of bit in the link $= P_d \times R$

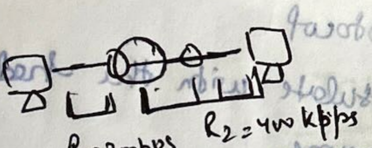
$= 8 \times 10^{-3} \times 2 \times 10^6 = 16000$ Bits

c) File size = 80000 bits
Time to propagate from [host] - [host]

Total time = $n+1 (P_d + T_d + P_d + Q_d)$

$= P_d + P_r$

End to End delay $= 8 \times 10^{-3} + \frac{80000}{2 \times 10^6} = 48 \times 10^{-3}$ s

Q 

Throughput = ?

$R_1 \to 1/2 \to$ Throughput = ?

Throughput = Maximum (R_1, R_2)

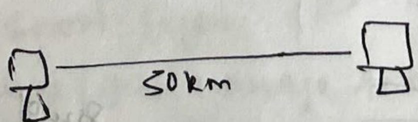
= Maxm (400, 3)

400 kbps

$R_1 = \frac{3 mbps}{2} = 1.5$ mbps

then, also Throughput - 400 kbps

Q point-to-point line of 50 km length.



$D = 50 \times 10^3$ m

Pkt (size) = 100 Byte

Speed = 2×10^6 m/s

i) what BW → $P_d = = T_d$

$P_d = = T_d$

$\frac{Dist}{Speed}$ $\frac{size}{R}$

$R = 3.2$ mbps

$\frac{50 \times 10^3}{2 \times 10^6} = \frac{100}{R}$

Q. N - pkt arrive to a link at which no pkt are currently being transmitted or queued
Pkt of length - L bit, Transmission Rate of R bps.
Avg queuing delay of N pkt?

After N pkt arrive at link

Delay for 1st pkt = 0

2nd pkt = L/R

3rd pkt = $2L/R$

$\vdots$

$(N-1) L/R$

Total delay $= 0 + L/R + 2L/R \cdots + (N-1) L/R$

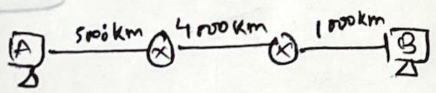
$= L/R \, [1 + 2 + \cdots (N-1)]$

$= L/R \left(\frac{N(N-1)}{2}\right)$

Avg Delay = $\frac{\text{Total Delay}}{N}$

$= L/R \, \frac{(N-1)}{2}$

Q. size = 1500 byte pkt



Prop speed = 2.5×10^8 m/s
Transmission Rate = 2mps (BW)

P_{rd} = 3m/s

End to end: $[18 + 40 + 6] \times 10^{-3}$
$= 64 \times 10^{-3}$ Ans

End-to-end delay $= n+1 \, (P_d + P_d + T_d + Q_d)$

Total Transmission delay $= (n+1) \, T_d$

$= 3 \times \frac{1500 \times 8 \text{ bit}}{2 \times 10^6}$

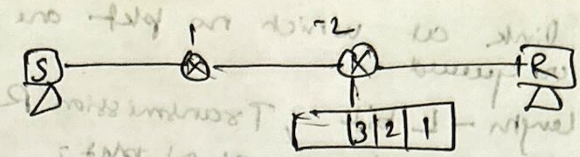
$= 18 \times 10^{-3}$ sec

Total prop delay $= (n+1) P_d$

$= 3 \times \frac{(5000 + 4000 + 1000) \times 10^3}{2.5 \times 10^8}$

$= 4 \times 10^{-2}$ sec

Total processing delay $= n \times P_d$
$= 2 \times 3 \times 10^{-3} = 6 \times 10^{-3}$ sec



5 km all link

$5 \times 3 = 15 \times 10^3$ m

R = 500 mbps

$= 500 \times 10^6$

Prop speed $= 10^8$ m/s

Proc delay = 0 size pkt = 2000 bits.

Queuing delay at Router 2

Qd for 4^{th} pkt $= \frac{3 \times S}{R} = \frac{3 \times 2000}{500 \times 10^6}$

$= 12 \times 10^{-6}$ sec

$Td = (n+1) \times \frac{L}{R} = 3 \times \frac{2000}{500 \times 10^6} = 12 \times 10^{-6}$ sec

$Pd = (n+1) \times \frac{D}{P\,speed} = \frac{15 \times 10^3}{10^8} = 15 \times 10^{-5}$ s

End to end delay $= 0 + 12 \times 10^{-6} + 12 \times 10^{-6} + 15 \times 10^{-5}$

$= 174\ \mu s$

200 Ok :-

301 Moved permanently

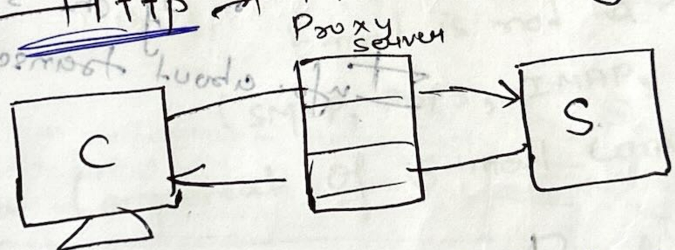
400 Bad Request - If you send request in bad format so, it can't understand

404 Not found - Request obj not found on server

503 Service Unavailable Currently your server is busy or down.

505 HTTP Version not Supported - Server does not support the HTTP version.

HTTP support proxy server



Advantages:

- when we send req. first it go to proxy server so, it check the req is avail to my end or not
  If yes avail then it go to my main server otherwise no.
  So, load is reduced.

- Traffic in the Internet is reduced
  If you search the kitt.ac.in and our proxy server had that data then it respone after to forward with main server.

- It is faster bcz it is in our local network
  reduce latency.

Or if we send 500 req and 400 is not available so, it became faster bcz they will not goes to main server.

Disadvantages

- More cost
- Stale cache

Two type of server are there

Stateless
:- If the server does not remember about the transaction done by client

Stateful
:- If the server rembers the info. about transection done by client

HTTP → (Default) Stateless

by help of cookies

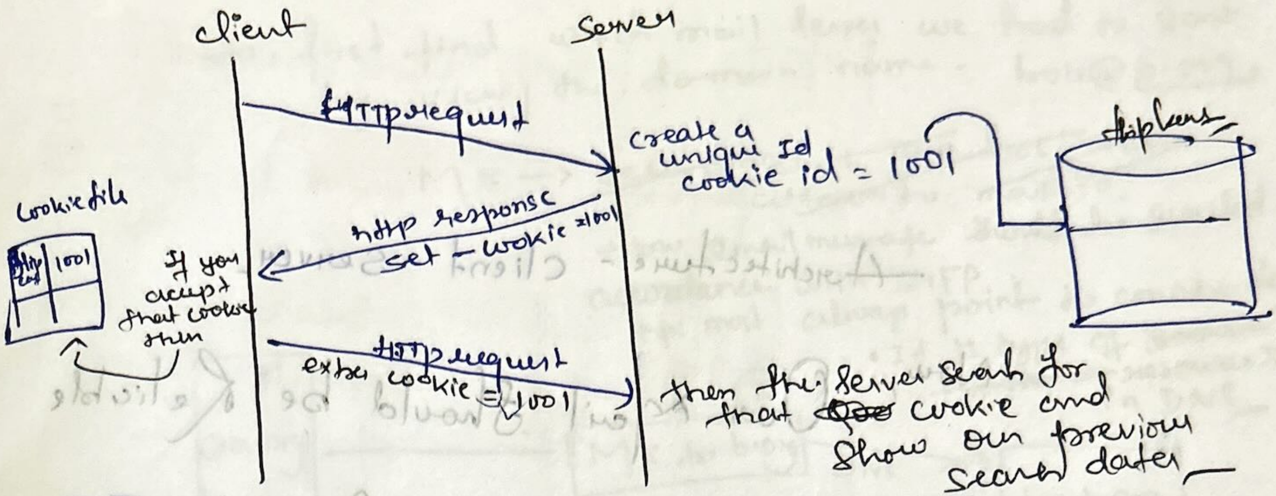
but sometime it behave

Stateful

Proxy Server

→ A proxy server is a computer, that keeps the copies of the response to recent request when HTTP client ~~server~~ send request to proxy server, the proxy server checks it's cache, if the response is not stored in cache, then the proxy server send the req. to Original server. Proxy server acts as both client & server.

It reduce the load on Original server



E-mail

DNS used to convert the hostnam ⇄ Ip Address

web service single protocol.

But, E-mail is not a single protocol

(SMTP, POP3, IMAP, MIME)

Various Component of E-mail comm-

1→ User Agent (Mail Reader)

Functionality — (Compose, Send, Receive, Organize)

User Agent

form:- Browser

Email - client Application (Microsoft Outlook, Mozilla Thunderbird)

↦ Mail Server

↦ SMTP

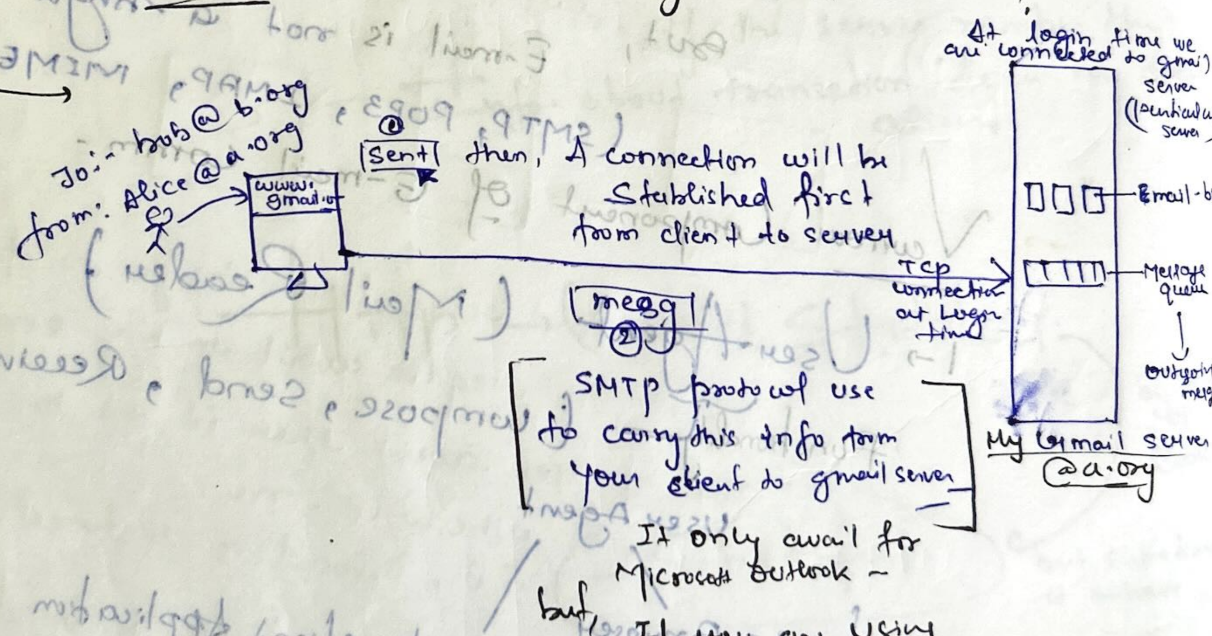
~~Architecture~~ - client - Server

Now, Our Mail Should be Reliable

So, In Transport Layer protocol → TCP

port no - 25 (email server run In 25 port)

How These Things Interact ?



SMTP protocol use to carry this info from your client to gmail server

It only avail for Microsoft Outlook -

but, If you are Using browser then - http protocol use

- Every mail Server Maintain One One or separate Mail box ~~of~~ or particular person to receive those ~~mail box~~. message.

When Message come from and it has to go Somewhere else so then are ~~som~~ message which is going On so, it had to wait in buffer (Message queue for Outgoing mail.

But bob is not connected to other mail server it is somewhere else so, →

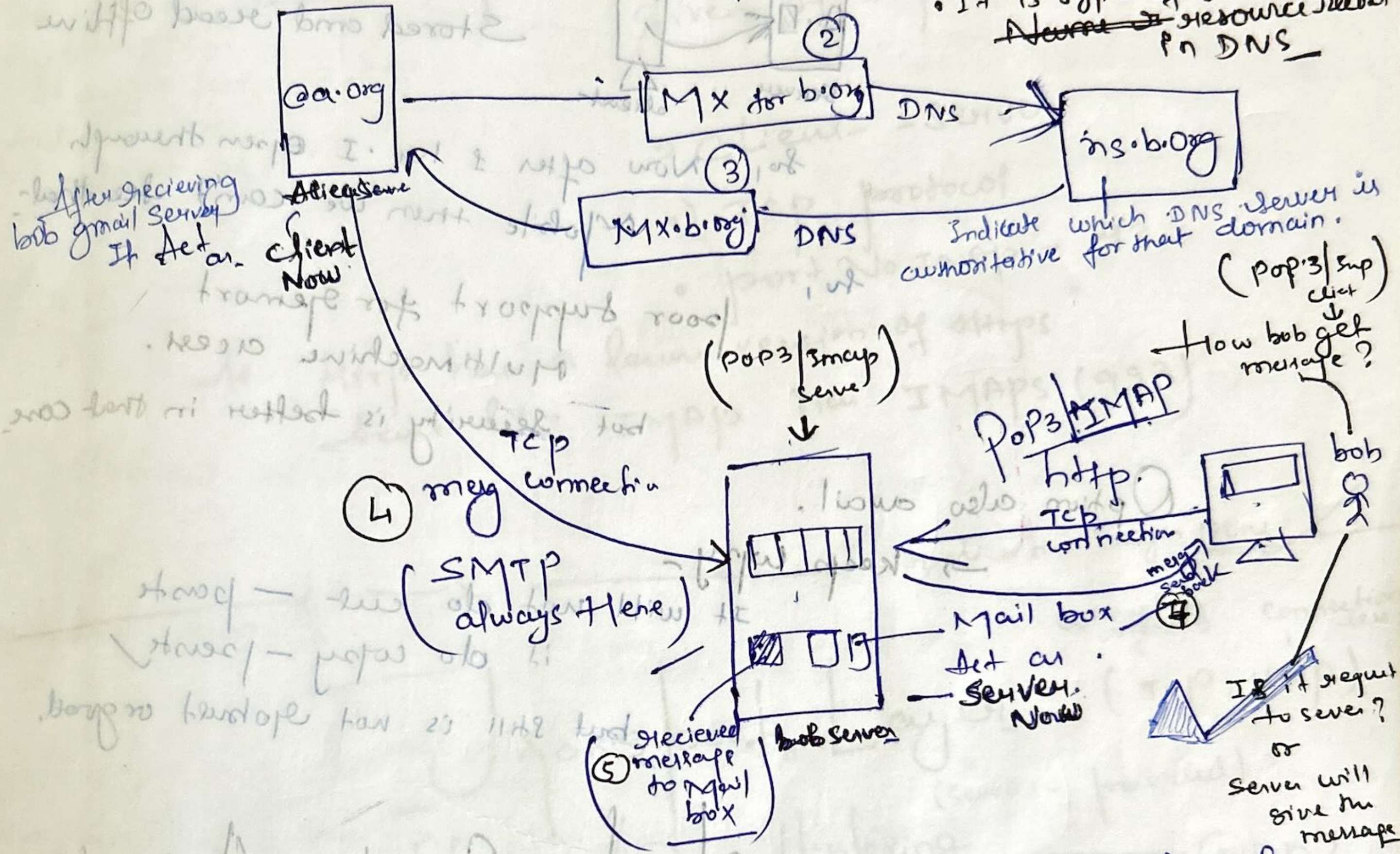
So, first find which mail server we had to sent by viewing the domain name - bob@b.org

MX → ~~Tell what is the host name~~ ~~attach~~ to mail id

How email message should be routed in accordance with SMTP

MX must always point to another domain.

• It is type of ~~Domain Name &~~ resource record in DNS



5 Whenever bob login to his server then connection established and it can see that—

Protocol

push - SMTP / http

pull - http / pop3 | IMAP (Mail Access protocol)

** Q what is push & pull what is needs of that?

Q w.r.t to specific protocol what are the various component to run

⇒ Eqnory browser here?

bob client / Aliece — Smtp client, pop3 | IMAP client

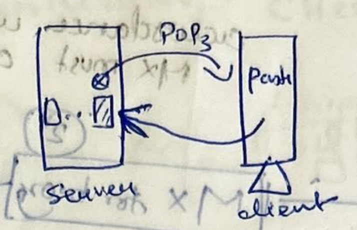
bob Serv / Aliece serve — Smtp server, Smtp client, POP3 | IMAP server.

MIME ⟹ Whenever, We want to attach Vdo, audio then, MIME protocol use.

(Congest) Message queue — Whenever My message queue is full then message will not recieve to bob.

POP3 (Post office protocol)

⊙ Very simple protocol with limited functionality.



• Emails retrieved, Stored and read offline

So, (3) Now after 1 hr. I open through mobile then we can't see that.

So, poor support for remort Multimachine acess.

but Security is better in that case.

Option also avail.

→ keep copy.

It will not do cut - paste
it do copy - paste

but still is not robust or good.

•) Client - Server Architecture

•) Tcp protocol

•) Port No - 110 Pop3

IMAP

- It is more Robust
- It acces by multiple device

Q Searching the mail at server side done due given by IMAP

- Client - Server
- TCP protocol
- port No IMAP - 143

As Http has a secure version of Https
same like IMAP has IMAPS (993)

~~x~~ Application Layer Done

Transport Layer

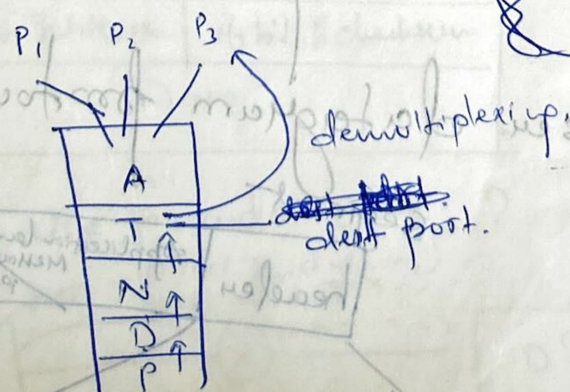
Connection oriented, Connection less
(TCP, UDP)

(service - provided)

service:

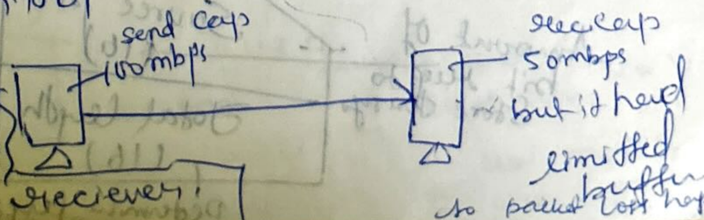
- Multiplexing / Demultiplexing ← {UDP}
- process to process Communication.

{support ← TCP
All feature}



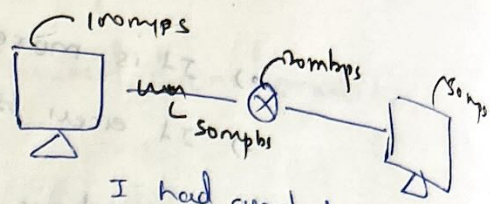
- flow, Control

Controlling the data Rate of sender
(Acc. to the caps of reciever)



• Congestion Control

↳ It take care the Capability of Network.



I had applied the flow control can I say no packet loss happen?

→ No, bcz my cap of Router is only 20mbps so, data loss may happen.

• Error Control

→ Even if I apply congestion control & flow control but I can not still ensure packet loss

① due to collision of packet)

② packet may be {corrupted | Loss | Receive multiple time}

UDP

(User datagram protocol)

Segment

header	Application layer Message / payload

} → Max length of segment
{$2^{16}-1$} bytes
= 65535 bytes

Source port (16)	Dest port (16)
Total length (16)	Checksum (16)

} — 32 bit

Amount of bit need to store this info.

Determine length

Max length of payload - 65535 - 8
= 65527

Segment

Header	PayLoad

port no

Range $(0 - 65535)$

$(2^{16} - 1)$

→ Well known → $(0 - 1023)$ bcz there are reserved for well known server. (Reserved by ICANN organisation.)

like - 80 for STTP, 53 for DNS.

→ Registered → $(1024 - 49151)$ we had to specify some time (www.server.com:8080)

→ Dynamic → $(49152 - 65535)$ Temporary

Checksum

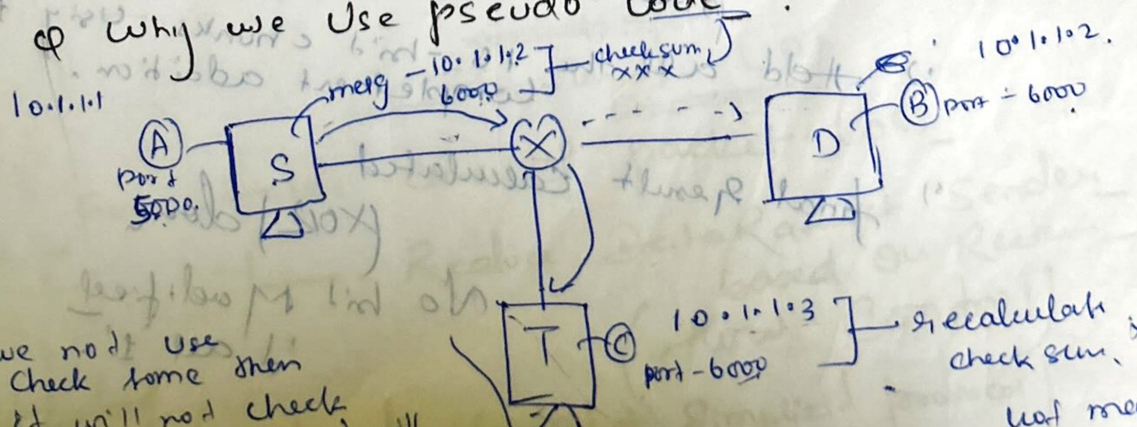
→ Though it we would check correctness or Integrety of the packet (error detection) (processing delay)

How it is check?

→ ⊙ Header
⊙ payload
⊙ pseudo Header Added

pseudo header	32 bit IP Add of Src	
	32 bit IP Add. of Dest	
	All 0s | 8-bit protocol	16-bit UDP length
Header	16 bit Src port no	16 bit of Dest port no
	16 bit Total Length	16 bit of checksum
payload	Data	

Q Why we use pseudo code?



recalculate check sum. xxy not match so corrupted

If we not use check some then it will not check Ip Add only it will check port no. to any 6000 so correct

UDP - checksum - Optional

Put 0 at checksum

— IPV4 (0)

but in IPV6 → Mandatory

(other them '0' put in checksum)

Sender

i) Divide the whole segment in to 16 bit chunks

Initialy checksum = 0

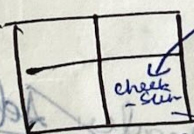
ii) Add all the 16-bit chunks using 1's complement addition

$$\begin{array}{r} 1110 \\ 0110 \\ \hline \cancel{1}0100 \\ 1 \\ \hline 0101 \end{array}$$

0101 — final sum after adding all 16 bit chunks (suppose)

(modify) final result has to be complemented to get the final checksum.

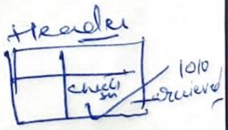
1010



If we get finaly '0' means sender is not Intrested in checksum so, make again complete to brough '1'.

Reciever

* Reciever will recieved whole segment



- Divide the whole segment Header, payload, pseduo header in to 16 bit chunks.
- Add all the 16-bit chunks using 1's complement addition.
- final result calculated

(XOR) done

1010 — If my checksum is 0

$$\begin{array}{r} 1010 \\ 0101 \\ \hline 1111 \end{array}$$

No bit Modified if all '1'.

Reliable Data Transfer
(RDT)

Channel

Noiseless

↳ There is no possibility of error no duplication, no Corruption.

Noisy

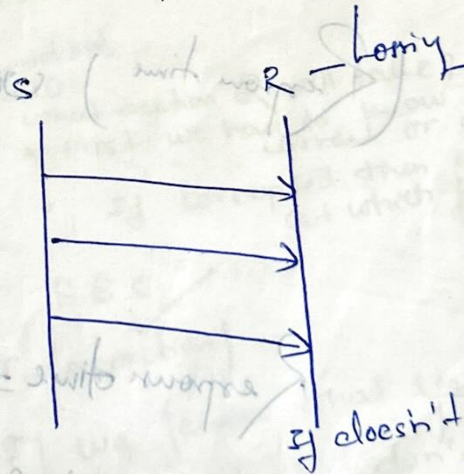
↳ packet may be loss, corrupted, Duplicated

protocol
→ Simplest
→ Stop & wait protocol.

- Stop & wait ARQ
- Go-Back-N ARQ
- Selective Repeat ARQ

(ARQ → Automatic repeat Request)

Simplest.



It just go and send the packet no worry about Reciever.

Here In noiseless there is no loss in channel but In Reciever may packet loss happen.

It doesn't have the capability more then Sender

Sender Data Rate > Recieven data rate

= Cause packet loss

~~ex: UDP~~

Similar to UDP

For ~~take~~ care of packet Loss

└ Reduce Data Rate at Sender based on Reciever

(flow Control)

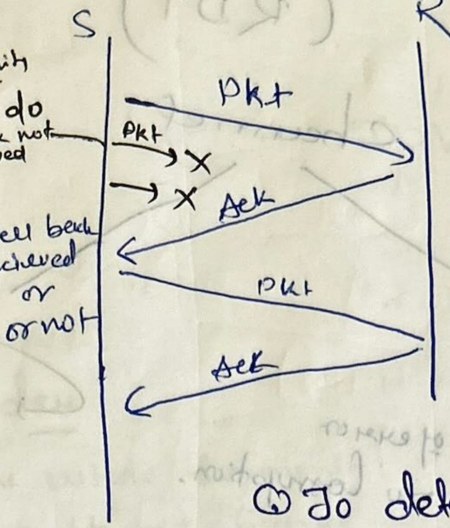
Stop & wait protocol = (Apply flow Control + Simplest protocol)

Stop & Wait protocol

└ sliding window size = 1
means 1 pkt send then without Ack it can't send another.
After Ack our new packet will send only
(flow control)

Check reliability
Not do unless Ack not recieved

to sender is it recieved the pkt or reliable packet or not
It tell back

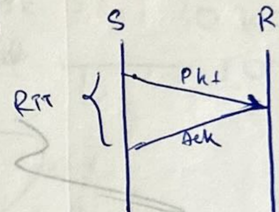


① To detect correct pkt checksum is added

problem ?

First

Efficiency | link Utilization | channel Utilization



(No Query or problem bcz no router bcz it is reliable)

(Response time) Total delay $= T_d + P_d + T_{d\,ack} + P_{d\,ack}$

$= T_d + 2 \times P_d + T_{d\,ack}$

($T_{d\,ack} <<$ very less so Ignore)

Response time $= T_d + 2 \times P_d$

Useful time of sender is only T_d

So, Efficiency $= \dfrac{\text{Useful time}}{\text{Total time}}$

$= \dfrac{T_d}{T_d + 2 \times P_d}$

Efficiency $= \dfrac{1}{1 + 2 \times a}$ ✓ $a = \dfrac{P_d}{T_d}$

$= \dfrac{1}{1 + 2 \times P_d / T_d}$

Efficiency of Stop & wait protocol is poor bcz it has to wait more time after performing Td. It is more poor when our Pd is more.

but In Reality All the channel is Noisy

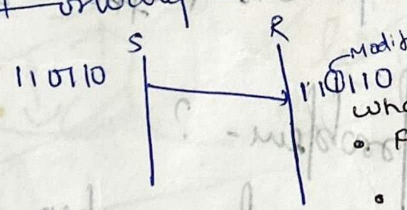
Stop & wait ARQ (Stop & wait protocol + Error Control)

Stop & wait protocol ARQ — Retransmission

deal with
- packet loss
- packet corruption (silent) — Forward error correction Mechanism / Backward error correction Mechanism
- packet duplication (sequence no) 0, 1

Sender ~~start~~ a timer ~~before sending~~ packet

Forward error Correction



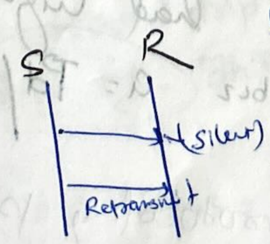
what action R will take?
- First we had to know whether it is correct or corrupted
- If corrupted then we had to know at which pos it is corrupted

then FEC

(Backward error Correction)

But If we know that yes it is corrupted but we don't ~~also~~ know at which pos.

→ ~~As~~ a result Reciever (silent)



Retransmission happen.

- May be this time we acheive correct packet

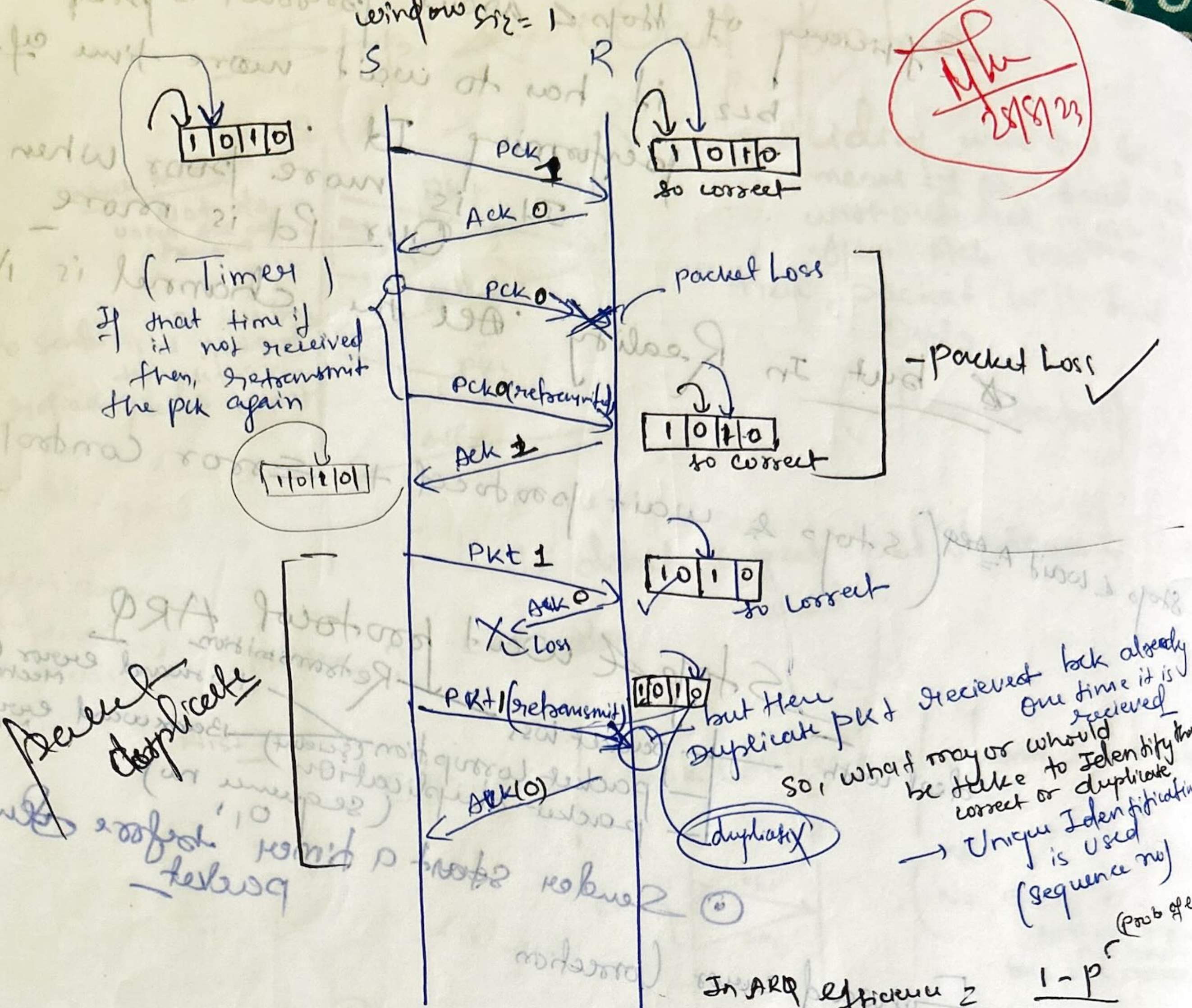
If Not again Retransmit

(Forward error correction is complex ~~to~~ Nature)

(BEC — Is easier but very ~~take~~ time (In TCP also))

28/8/23

window size = 1



In ARQ efficiency = $\frac{1-p}{1+2a}$ (Prob of error)

(P = 0) If not given

Now what is the problem - ?

→ Efficiency is less when Pd is Very high

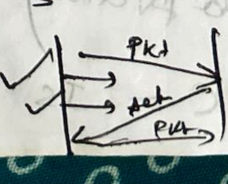
~~that~~ → Leads to poor Efficiency. (Satelite Comm.)

Or when my Td is Very less where we had high bandwidth

bcz $a = Pd / Td$.

Sliding Window protocol, Pipling Use to Improve the Efficiency

GBN, SR (protocol)



Go-Back-N ARQ (SW) - (Improve the efficiency)

Send Window - $(2^m - 1)$ - How many pkt should be send before receiving ack.

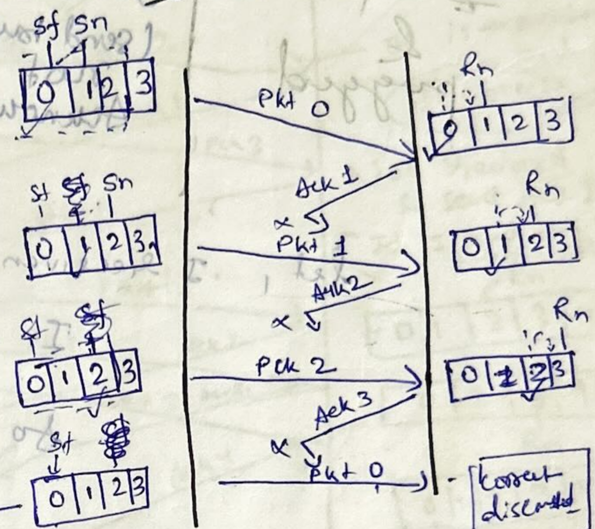
Recv Window - 1

Sf → first outstanding pkt
Sn → point to send next packet
(from m=2 then Range = 0 - 3)
Send window size - 3

Sequence no (m) = ~~(0, 1)~~
~~Can be sent. Correct~~
~~accepted / even from receiving~~
$(0 - 2^m - 1)$

If m=3 then Seq no = 0 - 7

0 1 2 3 0



(Timer Out) Retransmit - 0 1 2 3

~~Q If we change from $2^m - 1$ to 2^m?~~
~~let, Now we take 2^m = Window size 4~~
~~Or why we can use $2^m - 1$ send window size?~~

Bandwidth Delay product

Stop & wait protocol is very inefficient if our channel is thick and long

Thick → our channel has large Bandwidth
long → Round trip delay is long.
product of Thick & long → Bandwidth Delay product

Q Bandwidth line = 1 Mbps = 1×10^6
1 bit take 20 millisec to make round trip = 20×10^{-3}

Bandwidth Delay = $10^6 \times 20 \times 10^{-3}$
= 20000 bits

pkt → 1000 bit in length.

link utilization = $\frac{1000}{20000}$

$m = 3 \rightarrow 2^m - 1 = 7$ (0-7)

Automatic Repeat Request

Already send and ack recieved

S_f — S_n — Going to send — Not accepted

6	7	0	1	2	3	4	5	6	7	0	...

Send ack & purged	Outstanding (send but not Acknowledged)	Can be send when accepted from process	Cannot be accepted from process

let, I recieve an ack of 1 means I recieved pkt '0'

so, our S_f will now on 1.

Two Acknowledgment Scheme

(i) cumulative Ack

Ack - 3 — every thing is recieved

Pkt 2 ✗ — Pkt 3 — Ack 3

Ack - 3 means sequence 2 is also recieved

(ii) selective Ack

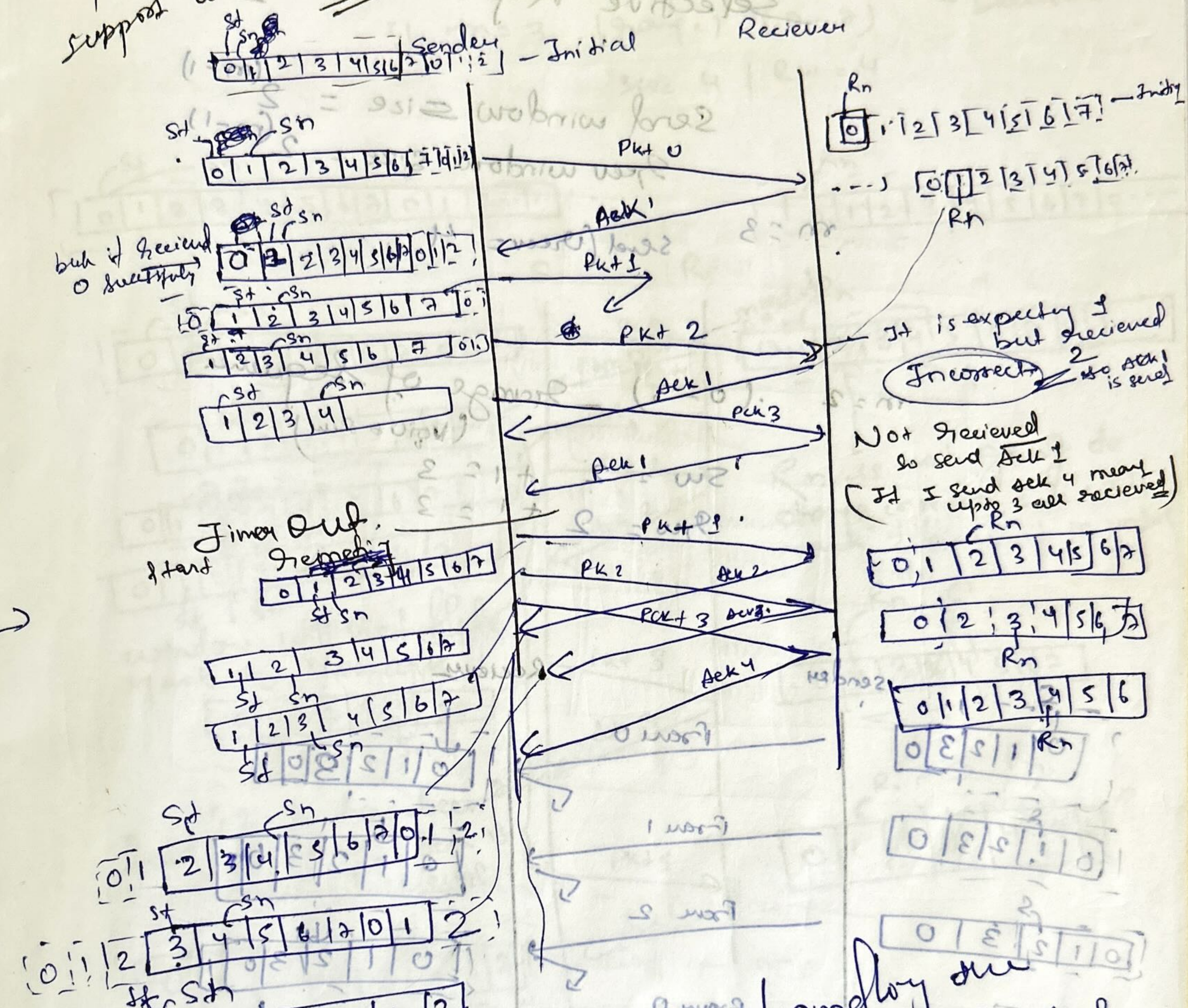
Ack - 3

Only 3 has been recieved, not check for any other

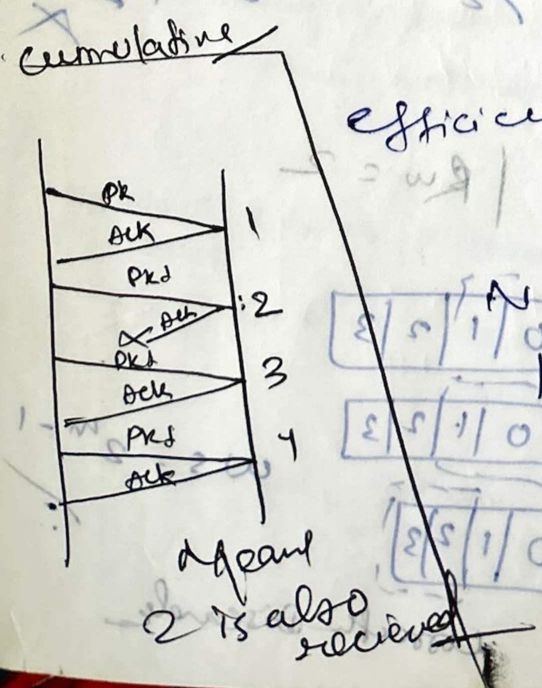
Go-Back-N ARQ

support cumulative Ack

$m = 3$ $2^m - 1 = 7$ | send window



Reciever is not handling out of order packet — Selective Repeat



$$\text{efficiency} = \frac{N(1-p)}{(1+2\times a)(1-p+Np)}$$

$N \to$ window size

$p \to$ probability of error

Selective Repeat ARQ

Send window size = $2^{(m-1)}$

Recv window size = $2^{(m-1)}$.

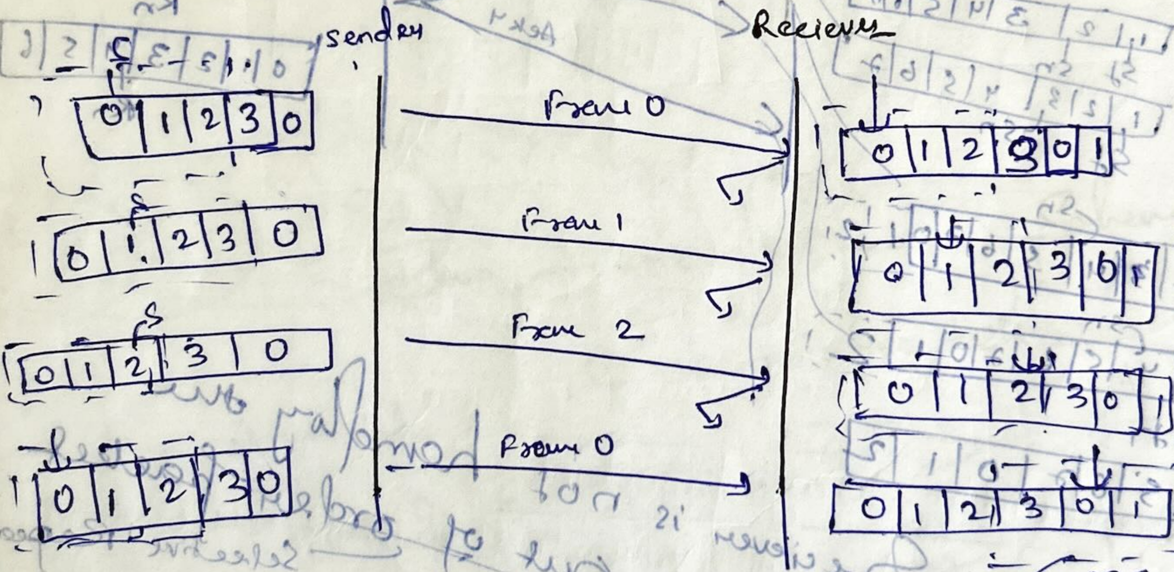
$m = 3$

send / recv = 4

$m = 2$: (0 – 3) – range of sequence (Sender & Recv)

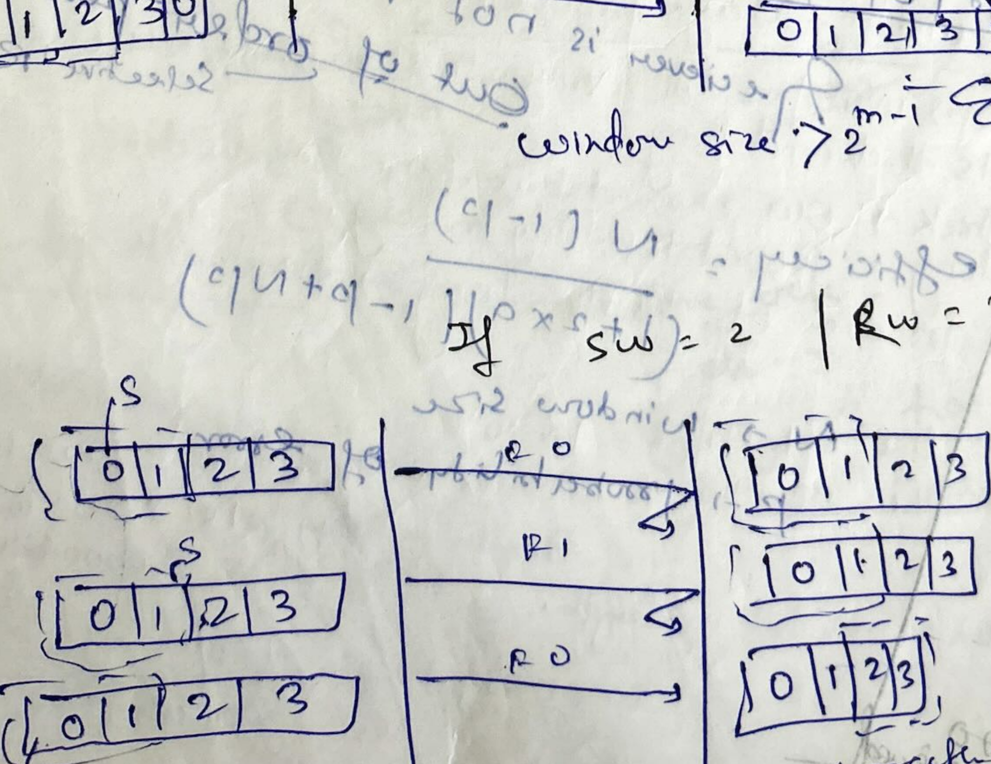
SW = 2 + 1 = 3

Rw = 2 + 1 = 3



window size > 2^{m-1} Erroneously accepted

If SW = 2 | Rw = 2



Correctly Discarded

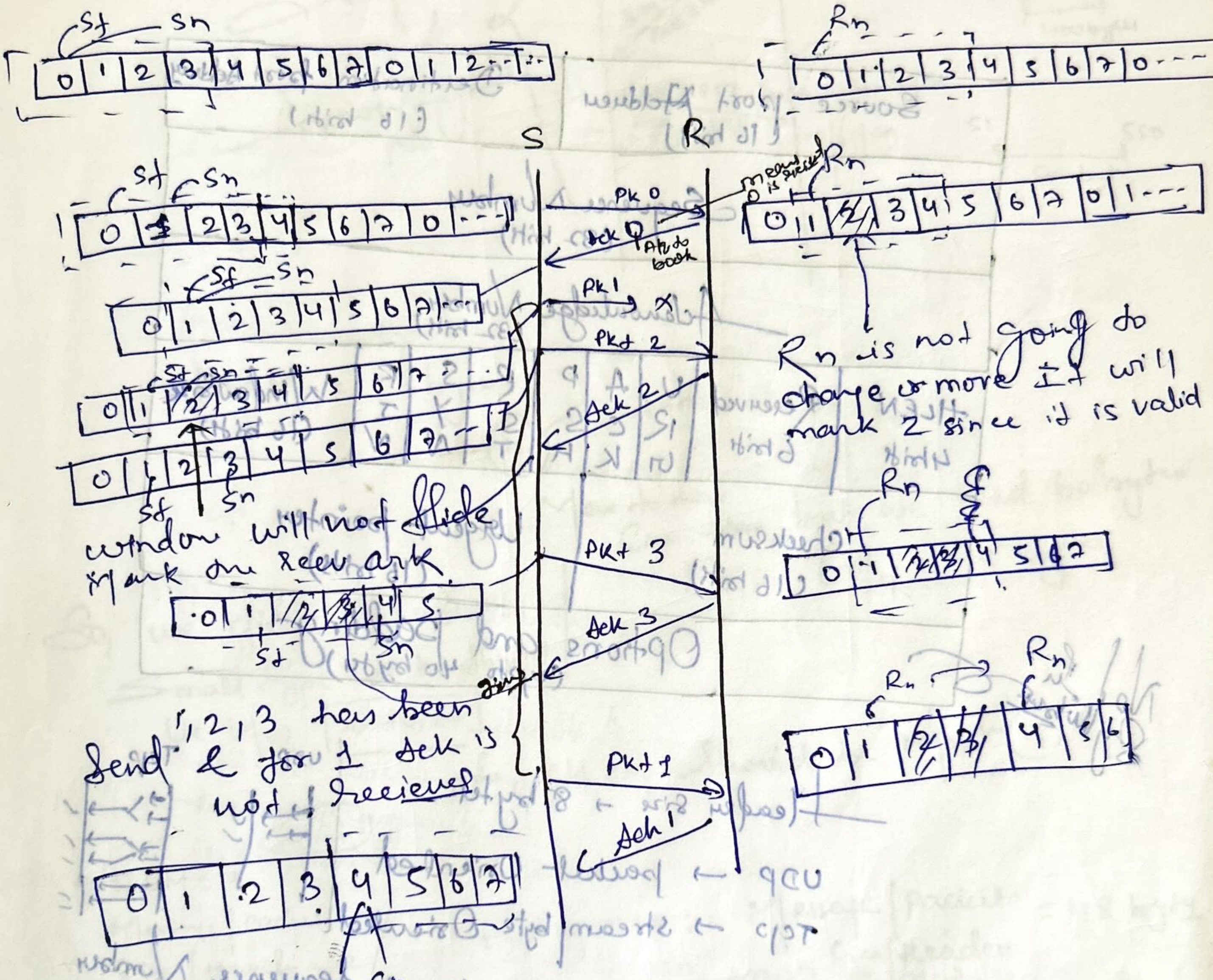
WS = 2^{m-1}

Selective Repeat ARQ

Support Selective Acks

If $m = 3$ (range) $(0-7)$

$S_w = 4$ | $R_w = 4$



TCP :-

20 to 60 byte	
Header	Data

a segment

Source port Address (16 bit)				Destination port Address (16 bit)				
Sequence Number (32 bit)								
Acknowledge Number (32 bit)								
HLEN 4 bit	Reserved 6 bit	URG	ACK	PSH	RST	SYN	FIN	Window size (16 bit)
Checksum (16 bit)				Urgent pointer (16 bits)				
Options and padding (upto 40 bytes)								

No of Syllabus

Header size → 8 bytes

UDP → packet Oriented

TCP → stream/byte Oriented

In TCP is byte e's given in a sequence Number (0 — 49 · 50 byte)

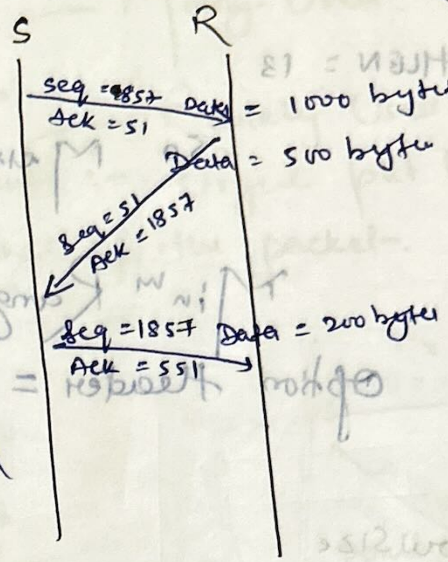
Here, Every Message Or packet doesn't have sequence number because it may combine or break down into Multiple fragement.

- TCP generally support accumulative Ack (It is always present) but with present of Cumulative Ack it also support Selective Ack (Additional features)

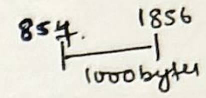
↑

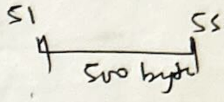
In Advance Version it generally supported

Sequence No & Acknowledgement No



Every byte assigned sequence no.
So, It is this much





HLEN = 4 bytes.

15 — Can store

Max to max Can store but we need 60 bytes.

So, 60/4 = 15 bytes.

So, we divide the size in small type which should lie b/w $\boxed{20 \text{ to } 60}$

So, HLEN should be Minm 5

HLEN = 3 — Invalid

Message | packet Size | Header = 12

HLEN = 12 : Message | Packet Size | Header = 48 bytes

HLEN = 10 : Message | packet Size | Header = 40 bytes,

(The size of HLEN is Multiple of 4.)

∴ Message | packet size | Header = 50 (Not valid)

To validate it do padding

∴ Near greater value divided by 4

= 52

∴ +2 is padding.

HLEN = 13

So Message | Header = 52

Min^m Range of Header

Option Header = 52 - 20 = 32

length of header

Window size

with the help of window size the reciever control the data

window size (16) = 65595 bits.

Reciever window

Window size = 1000 bytes

Send them it will

Advertise Window (hoong to tell the sender frequently about size) whenever reciever will send.

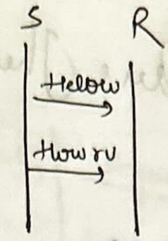
* Checksum - same as UDP
  - Header, payload, pseudo Header
    - Src Ip
    - Dst Ip
    - protocol
    - UDP length

Flags significant

- URG (urgent)
- Ack (Acknowledgement)
- Psh (Push)
- Rst (Reset)
- SYN (synchronization)
- FIN (finish)

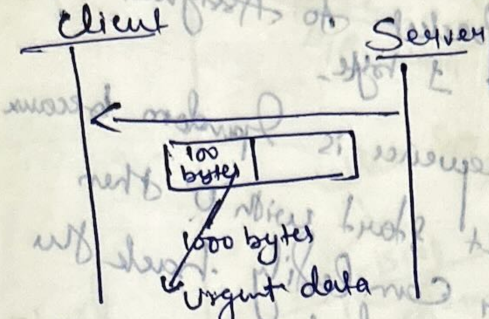
Data Communication

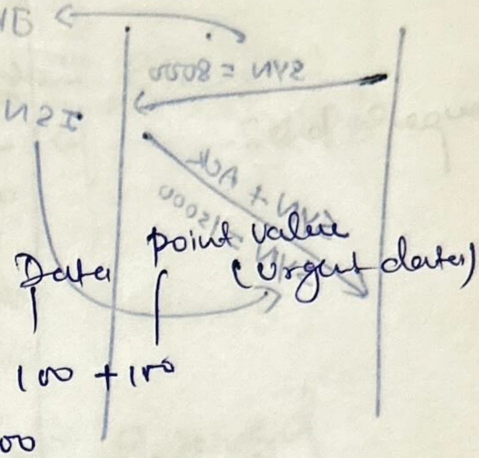
Connection Management



Push :- Mostly Used / Rarely Used

Urgent :- Urgent pkt comes at the begining of the packet.



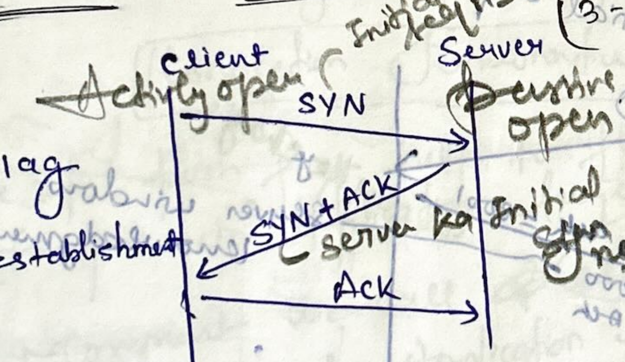


Urgent point = 100

Last byte of the Urgent data = 100 + 100 = 200

Connection Establishment

(3-way Handshaking)



SYN Flag

↓

Connection Establishment

4 way Handshaking.

Jb ek array full nhi hoga ya send nhi hoga best usko decide krna padega even if client is closed

~~Graceful~~

(FIN) Graceful & Forceful (RST)

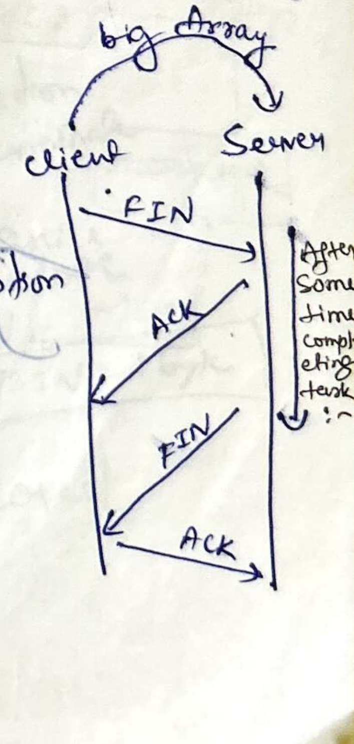
Teardown

↓

(3-way Handshaking)

Exception

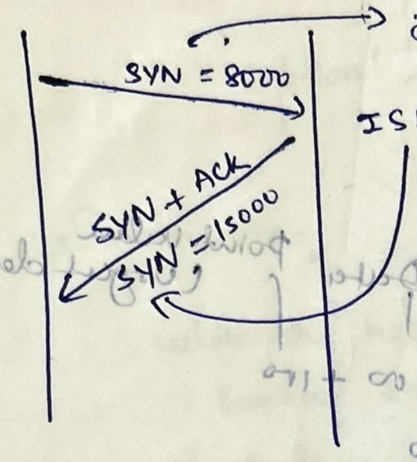
FIN



Two type of Teardown Are There

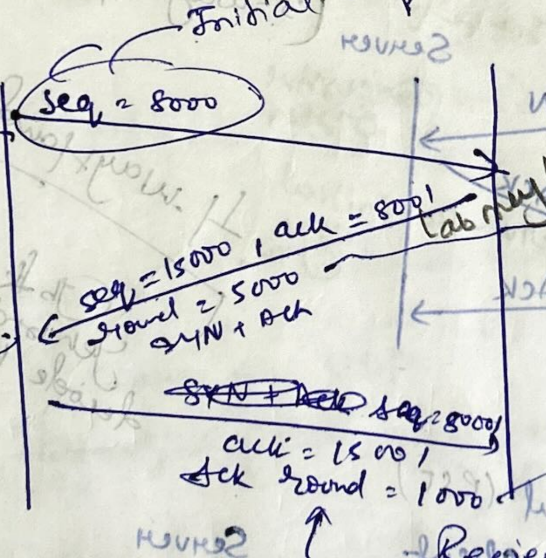
① Graceful (FIN)
② Forceful shutdown (RST)

This is Control packet to Assigned 1 byte.



The sequence is random because of it start with '0' then Hacker can easily hack the data

Control packet & Read
Data Packet



Two Major things exchange after Connection is

① Window size
② sequence number (initial)

Client

Connection Establishment - Done

Now to transfer data whole part first seq. no. of this ~~segment~~ is fixed

Send request --> Sequence = 8001, ack = 15001

~~Ack~~ A/P Data bytes = 8001 - 9000 --> Receive

PSH flag (urgent flag)

Send Request --> Sequence = 9001, ack = 15001

A/P Data bytes = 9001 - 10000 --> Receive

uska seq no

Send request

seq = 15001, ack = 10001

A/rwnd = 3000

Data byte = 15001 - 17000

Received <--

Send Request --> seq = 10001, ack = 17001

A/rwnd = 10000 --> Received

Received and given to Application Layer (Flushed)

Connection Termination

P - Use of push flag helps to flush the message from transport to Application layer.

- The requirement we call ~~@~~ user function then the data is flush to Application layer of push flag is not set.

Active close --> (finish) FIN A/P, Seq = 10001, Ack = 17001 --> Connection ~~Sever~~ terminate ~~close~~

~~Red~~

FIN + Ack, Seq = 17001, Ack = 10002 <-- - - passive close

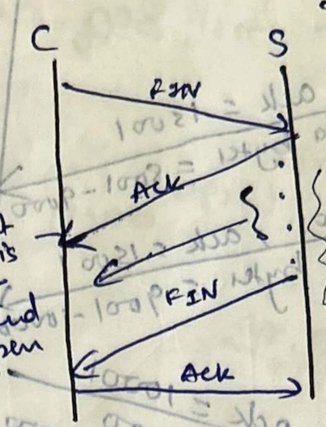
Connection closed 17001 + 1 byte

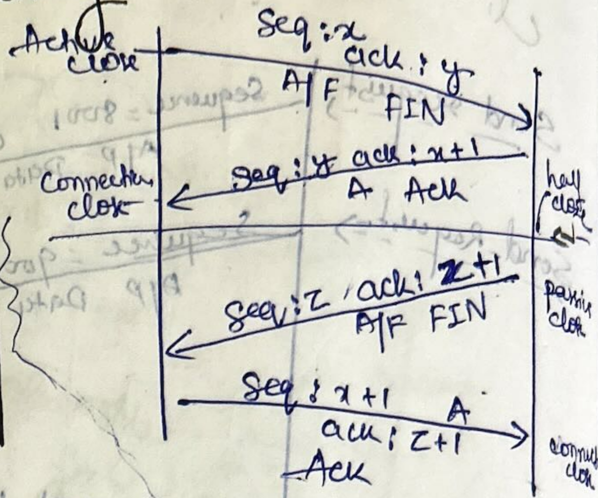
becoz FIN = 1 byte

Ack = 17002, Seq = 10002 --> - - Closed

4 - way Handshaking

Called as Half close — client end is close while Server end is open



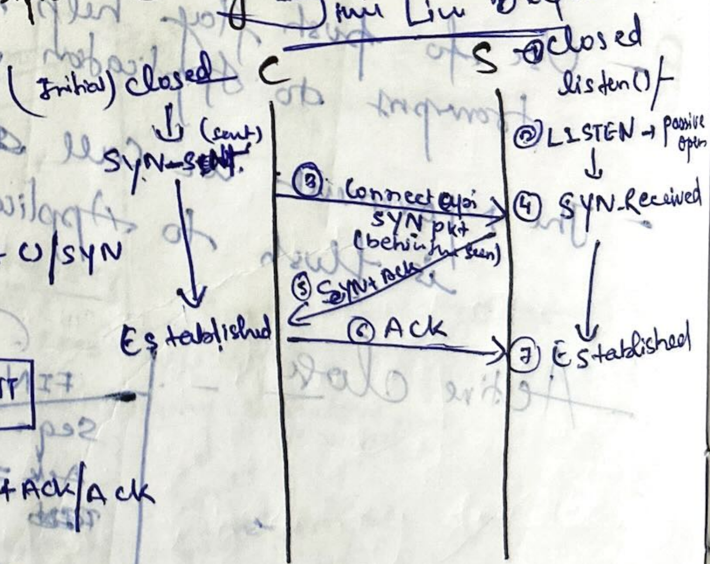


- Sending pkt is no more allowed but server can send.
- Being in Half close (client) but can able to recieve the data is permissible ✓
- Being in Half close allow to send Ack ✓ but not the pkt.

State Transition Diagram for TCP.

Time Line Diagram

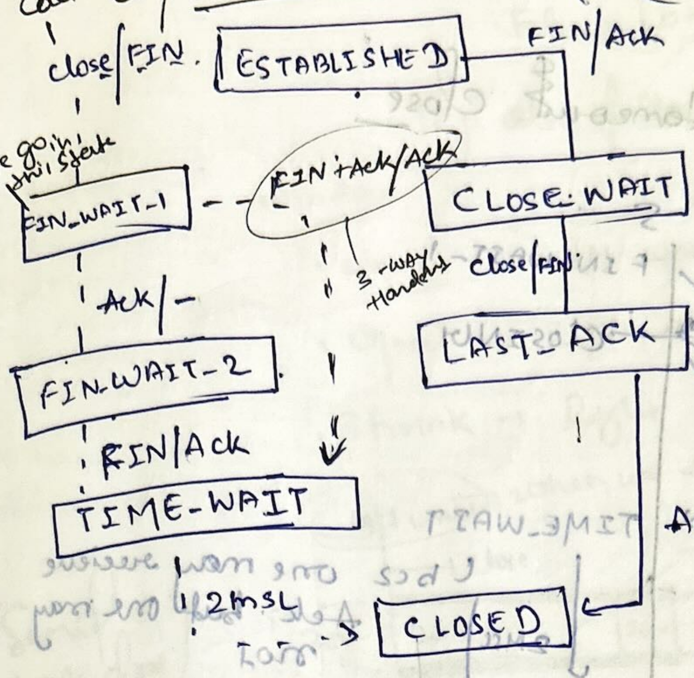
Connection Establishment

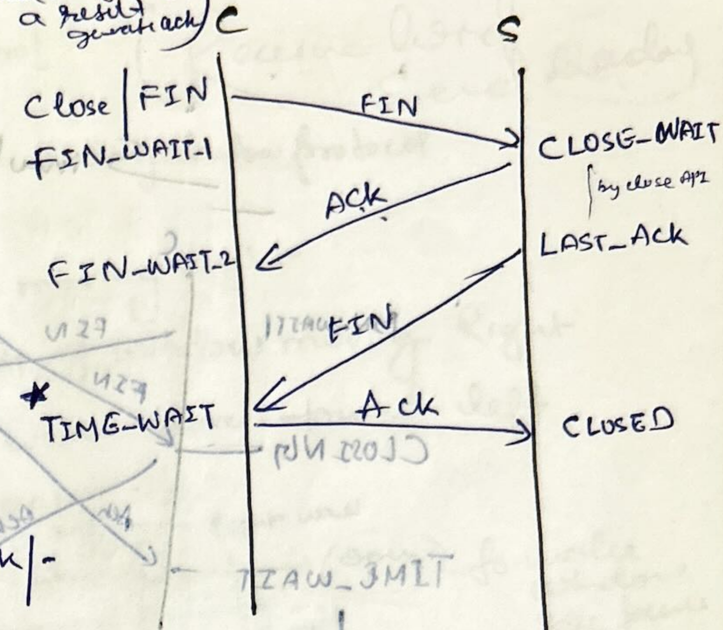


Now Our Connection is Established for send pkt client call Connection Teardown

As a result FIN is send

received the fin as a result generate ack





NOTE :- In 3-way Handshaking FIN-WAIT-2 Not Encounter

2 msl - max^m segment Lifetime.
L wait in Time-wait

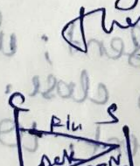
what is need why we wait?

→ ① Reason

To confirm my Ack is recieved to server. end ~~or~~ not

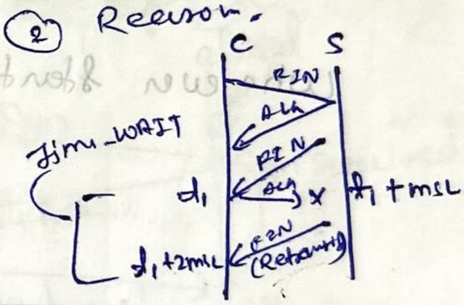
So, that's why my Server will be go In close state then client will go

In case my Ack is lost then, (my) Server never go to close that's way Client wait for 2 MSL to confirm that my Ack is recieved then client also closed



(If not ~~then~~ again FIN (retransmit) from server)

② Reason.



MSL -

→ A segment can stay in the network

If it not recieved on that time then it had to discarded

Suppose It recieved FIN, but Ack ~~X~~.

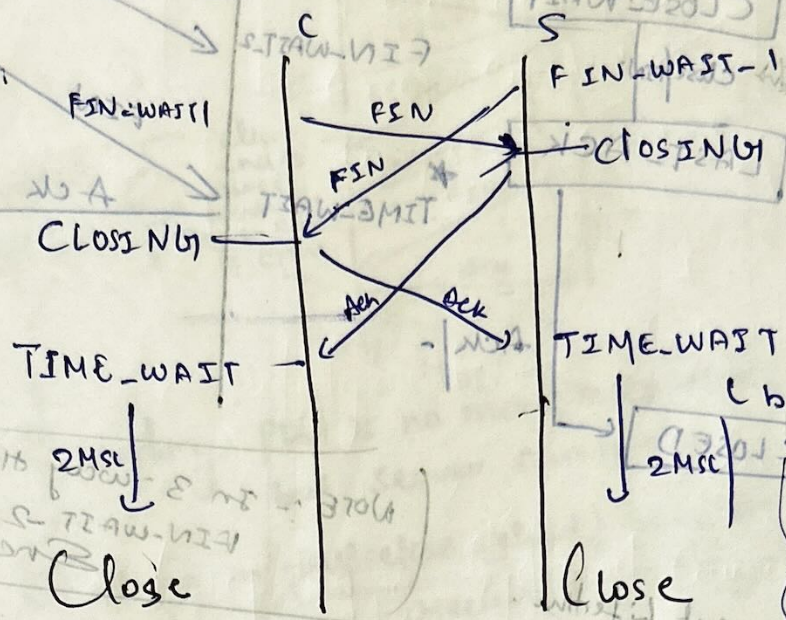
then retransmit happen

(~~t_1~~ d_1+msL, d_1+2msL)

If after 2 MSL ~~I~~ did not recieved retransmit

~~they~~ think mean our Ack is recieved

Simultaneous close



bcz one may recieve Ack but one may not

If we Not use then Other will never go in close state

Whoever started the teardown first will close first —

S - CLOSED - No Connection exist

S - LISTEN - passive open received; waiting for SYN.

C - SYN_SENT - SYN sent; waiting for Ack.

S - SYN-RCVD - SYN+Ack send; waiting for Ack.

C,S - ESTABLISHED - Connection established; Data transfer in progress.

C - FIN_WAIT_1 - First FIN sent; waiting for Ack.

(AC) C - FIN_WAIT_2 - Ack to first FIN received; waiting for 2nd FIN

S - CLOSE WAIT - First FIN received, Ack sent; waiting for application to close.

C - TIME-WAIT - Second FIN received, Ack sent, waiting for 2MSL timer-out.

S - LAST-Ack - Second FIN send, waiting for Ack

C,S - CLOSING - Both side decided to close simultaneously.

Flow Control

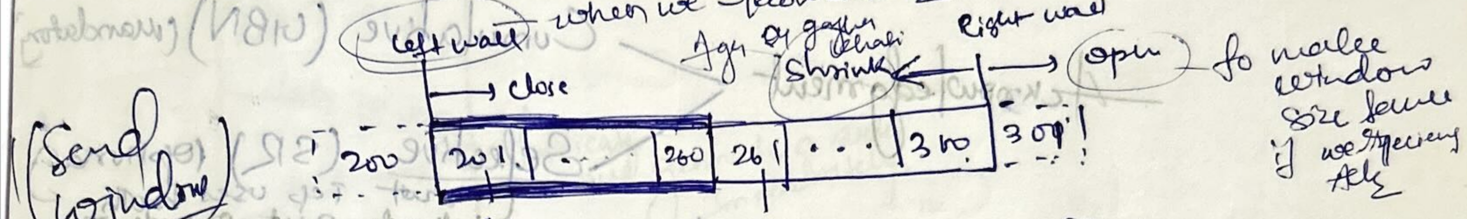
(Receive Window
Send window)

Tcp also follow sliding window protocol

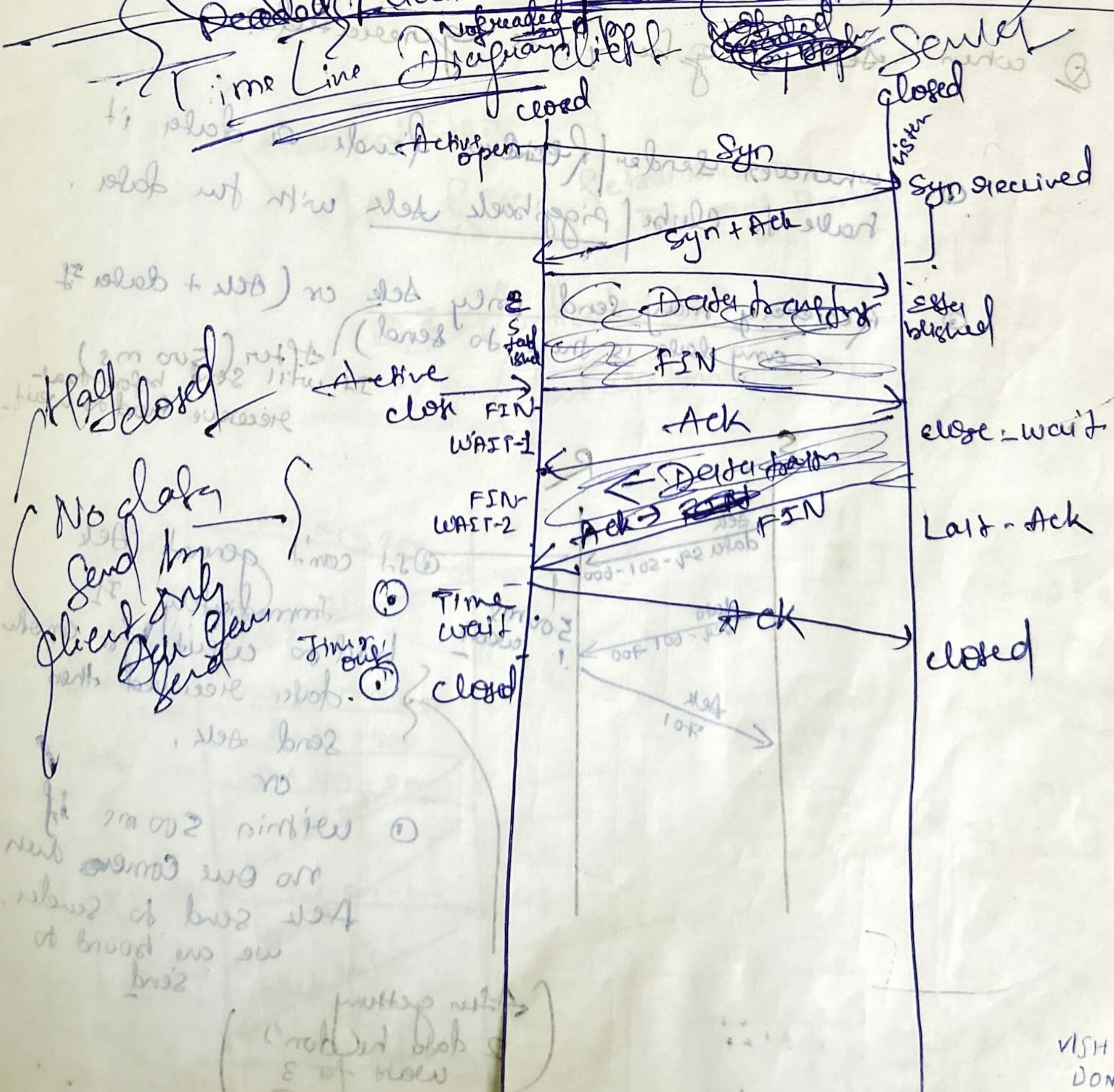
window

- close → Left wall moving Right
- Open → Right wall of window moving Right
- Shrink → Right wall moves toward left

when we receive an Ack



(Recieve window doesn't have shrink because same)
to not comming
~~Decided~~
buffer
Not readed



VISHAL
DONGANDU

As a part of Reliability

Functionality occur in TCP.

① Flow control ✓

② Error control
- Packet corruption
  - └ Checksum
- packet duplication
  - └ seq no (0,1)
- packet loss
  - └ Acknowledgment
  - └ Retransmission
    - i) RTO — Retransmission timer out time
    - ii) 3 duplicate Ack — (fast retransmission)

③ Congestion control

Acknowledgment
- Cumulative (GBN) (Mandatory)
- Selective (SR) (Optional)
  (but TCP uses both
  help for out of order pkt.)

★ (TCP support by default Cumulative Ack)

Q when Ack is g being is generated?

① whenever sender/Receiver sends a data it have to Club/Piggyback Ack with the data.

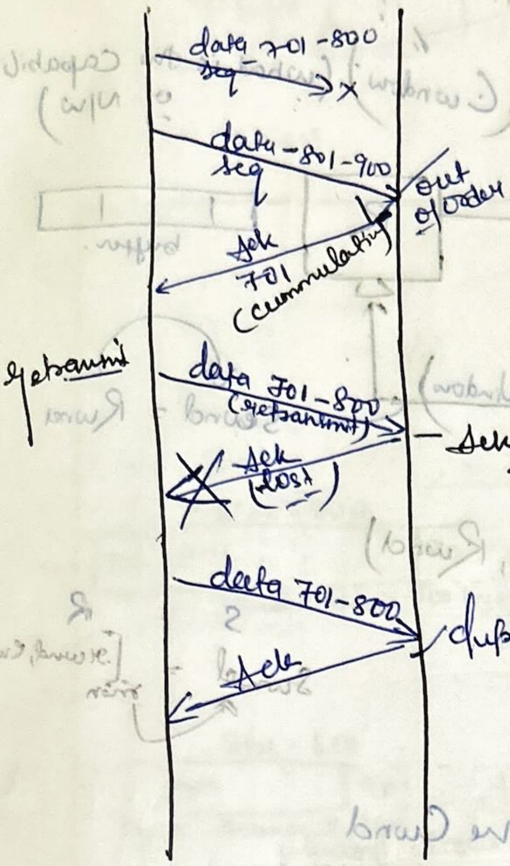
② Receiver may send only Ack or (Ack + data if any data is there to send) After (500 ms) It will send before that receive had to wait.

① It can't generate Ack Immediately It had to wait for another data recieved then send Ack.

or

① within 500 ms if no one comes then Ack send to sender. we are bound to send

(After getting 2 data he don't wait for 3



① If the receiver is recieving out of order pkt it don't wait immediately it will send an Ack.

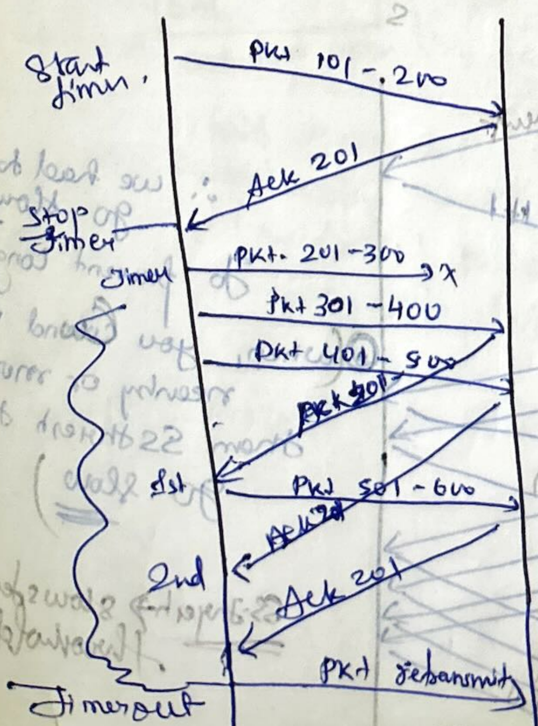
Ack here also send immediately

duplicate and generate a ack Immediately.

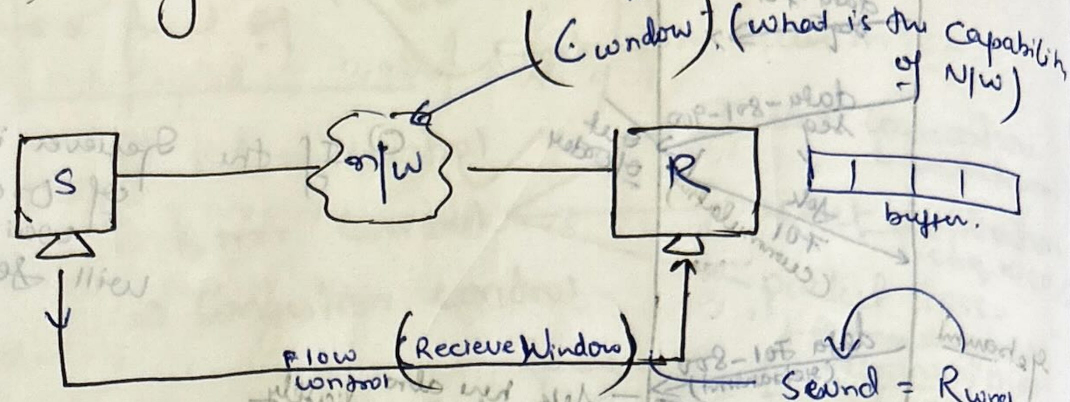
Retransmission

① RTO - Retransmission Timeout Timer

② 3-duplicate Ack (fast retransmission)



Congestion Control.



so, Swnd = Min (Cwnd, Rwnd)

S R

Swnd = min [Rwnd, Cwnd]

let Cwnd = X

If congenstion occur
then, decrease the Cwnd

If no congenstion
then, Increase the Cwnd to get more efficiency.

⊙ How to find the Congestion Window Size?

↳ Initially Cwnd = 1 MSS

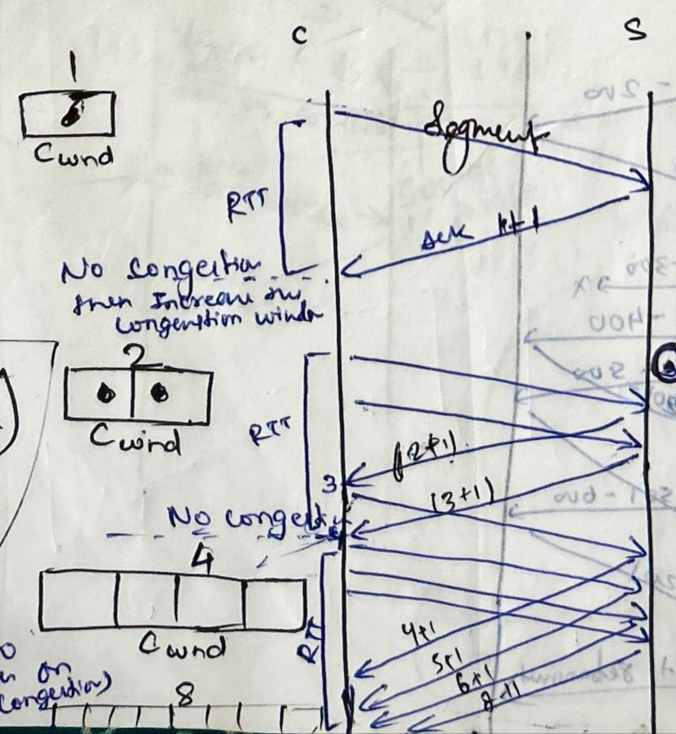
1 phase :- slow start

Cwnd = Cwnd + 1

Every time I recieve an Ack

Every RTT, my (Cwnd = Cwnd * 2)

Exponential Increase



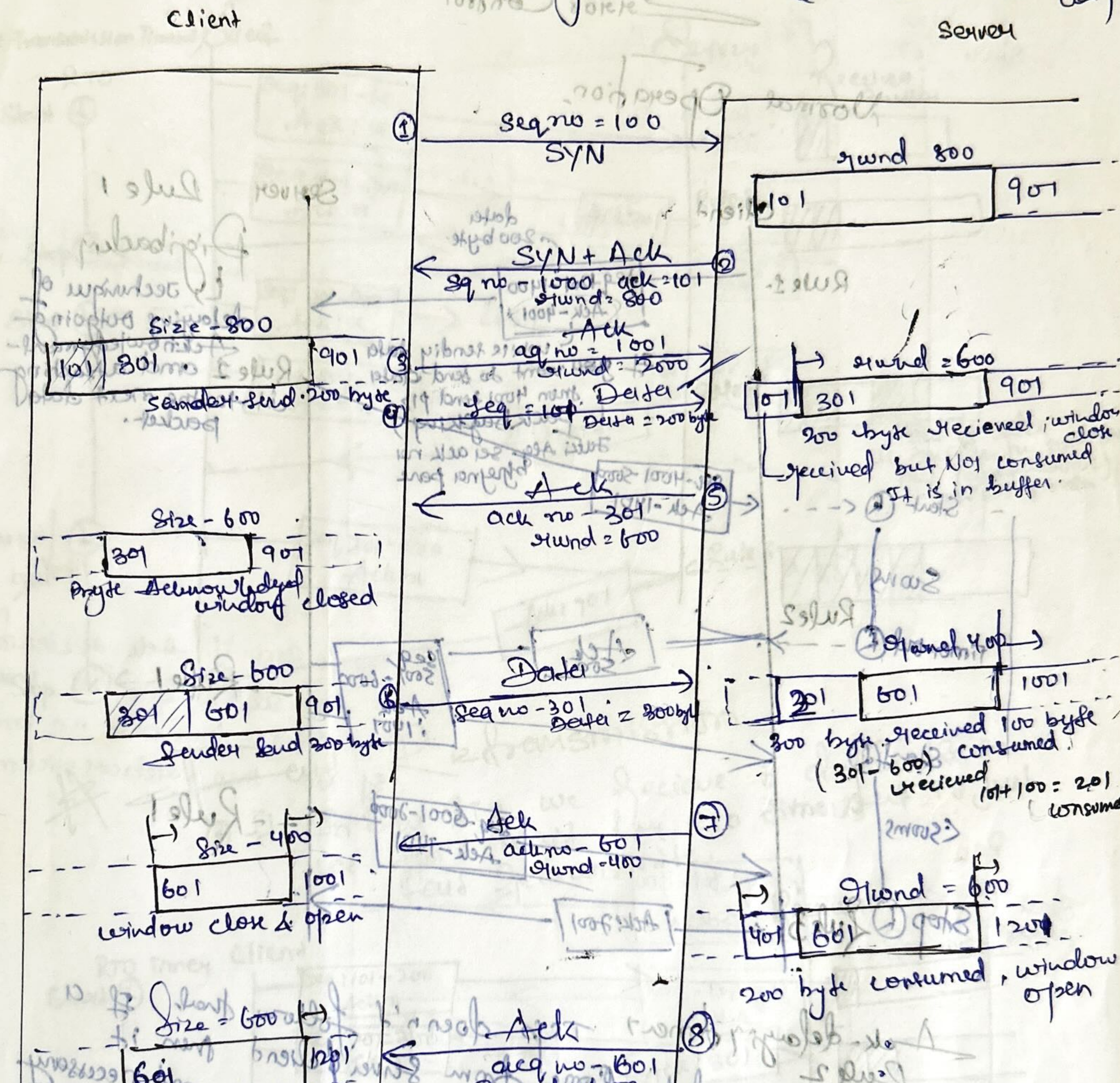
∴ we had to go slow to prevent congestion

⊙ (when, you Cwnd is nearby or more than SSthresh then go slow)

SS-Thresh → slow start threshold

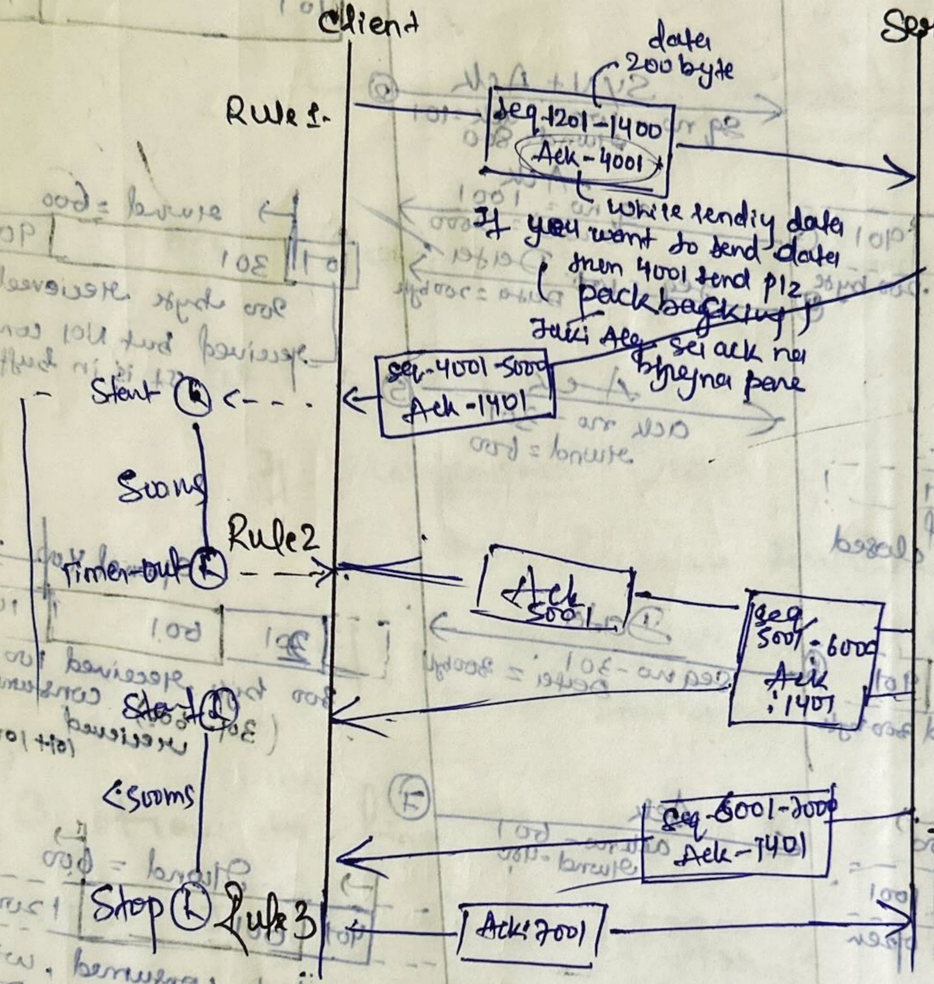
(TCP) Flow control

By direction. (~~Send~~ Send or Receive slows or faster (sw))



Error Control

Normal Operation.



Rule 1
Piggybacking
→ Technique of delaying outgoing Acknowledgement and attaching it to the next data packet.

Rule 1 and attaching

— Rule 1

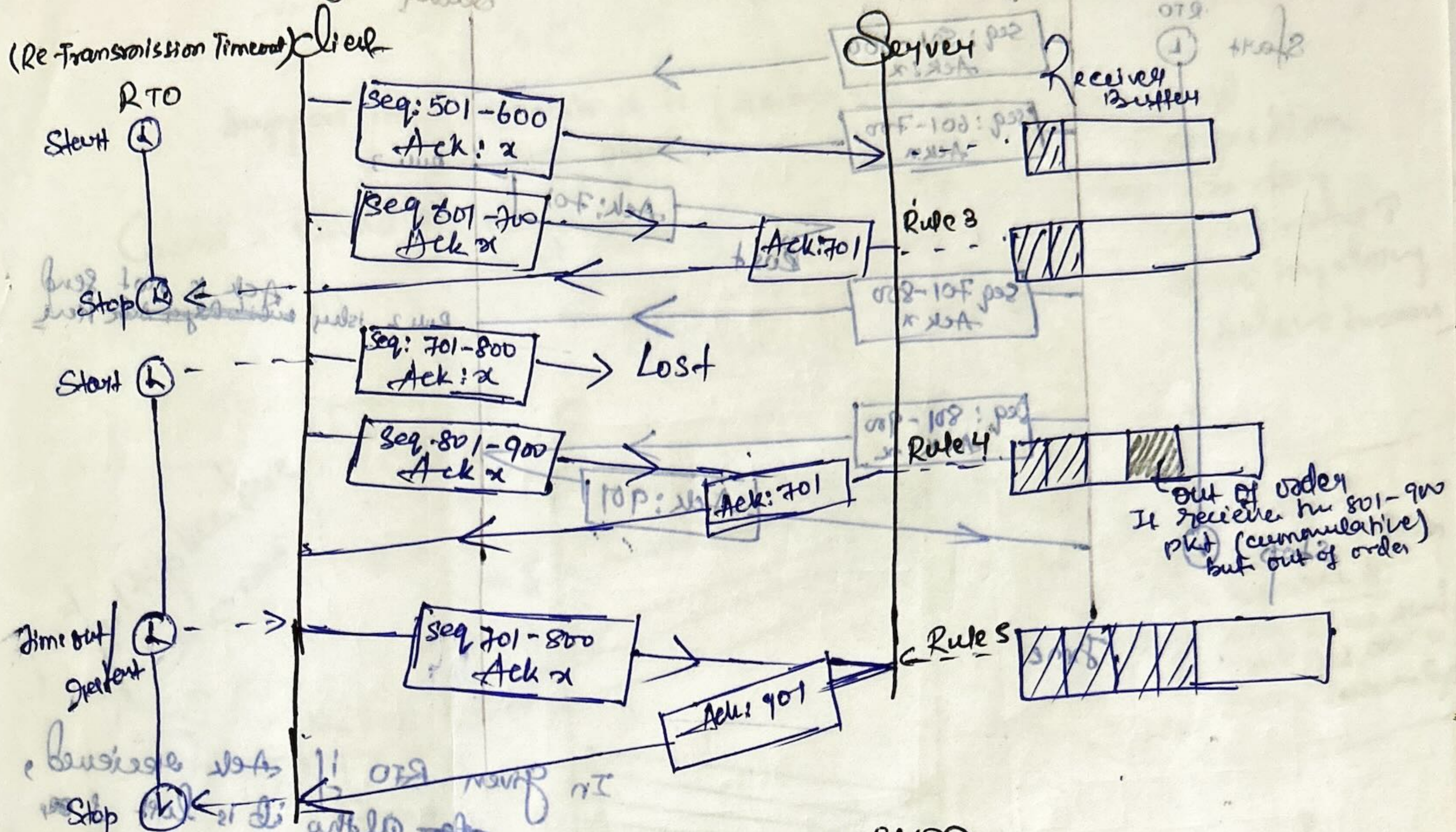
— Rule 1

~~Ack-delaying timer~~
Rule 2

TCP doesn't follow that if a data come from server/client then it is not necessary Immediately send Ack. If it send Ack immediately then too many Ack were made. So, it wait for some more pkt to come ~~after time~~ then it send Ack with that packet (combine) if not then send only Ack after timer-out.

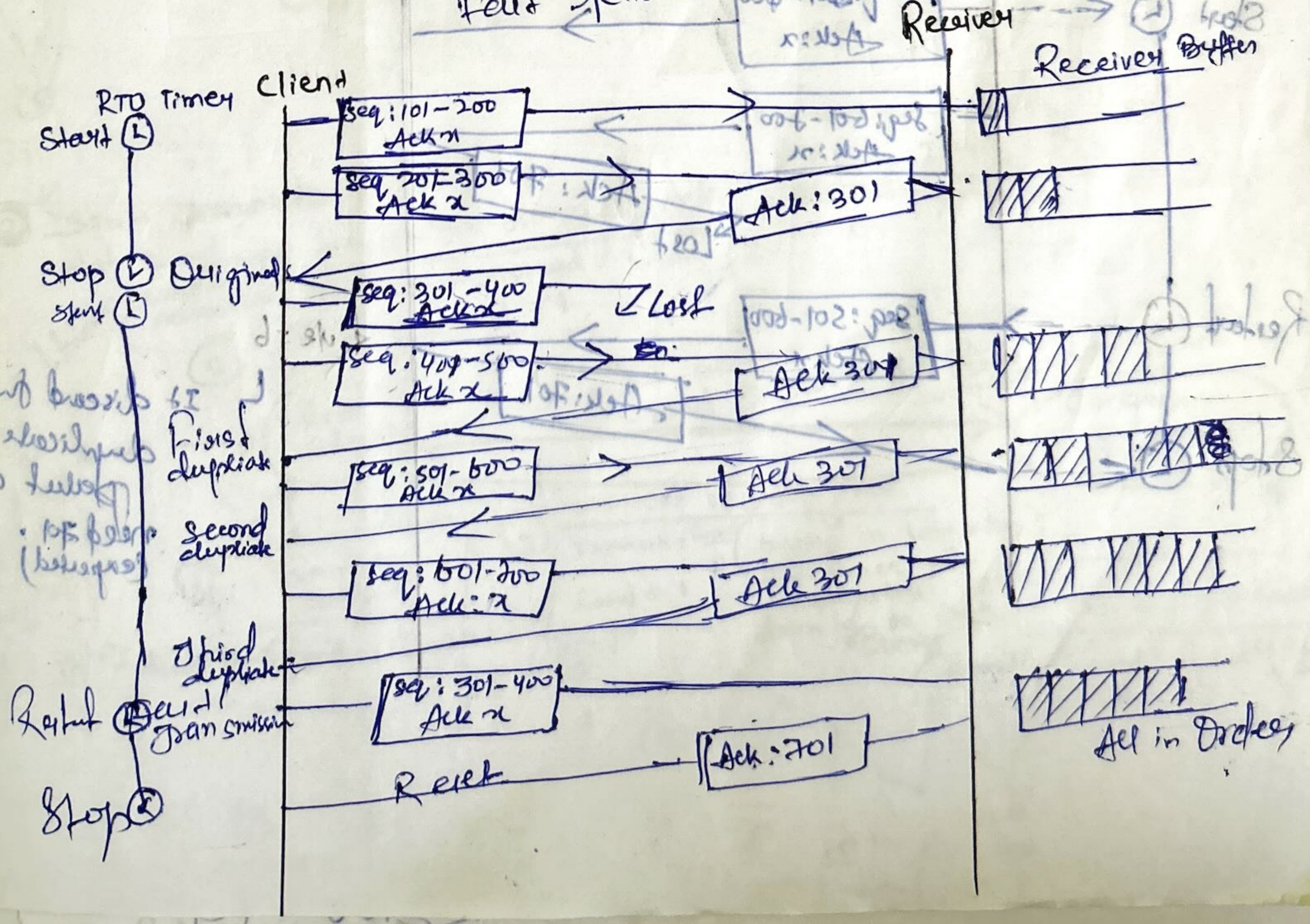
Rule 3:- If you get 2 pkt In Order don't wait then for 3rd one It send Ack.

Lost Segment.

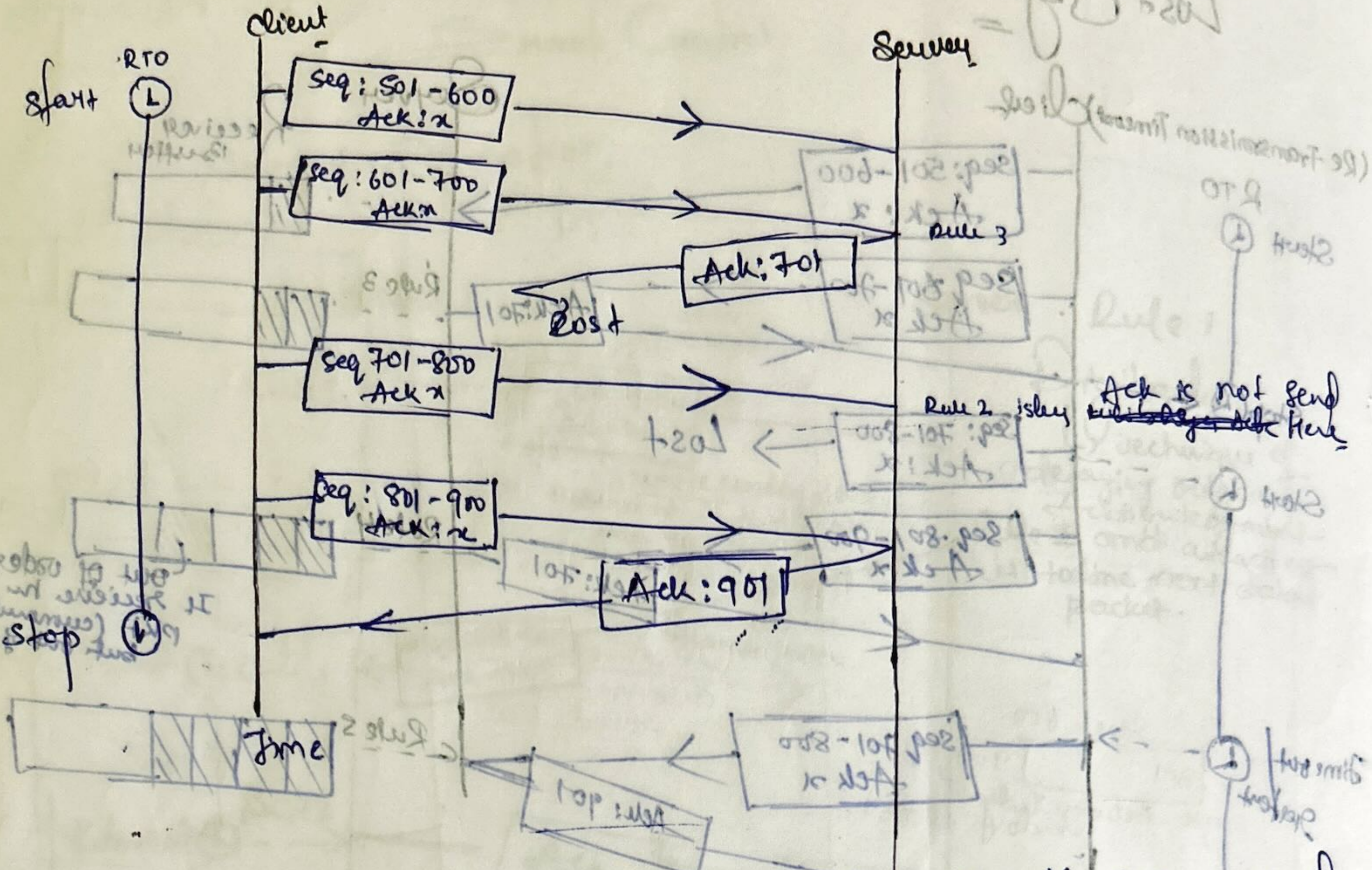


Fast Retransmission.

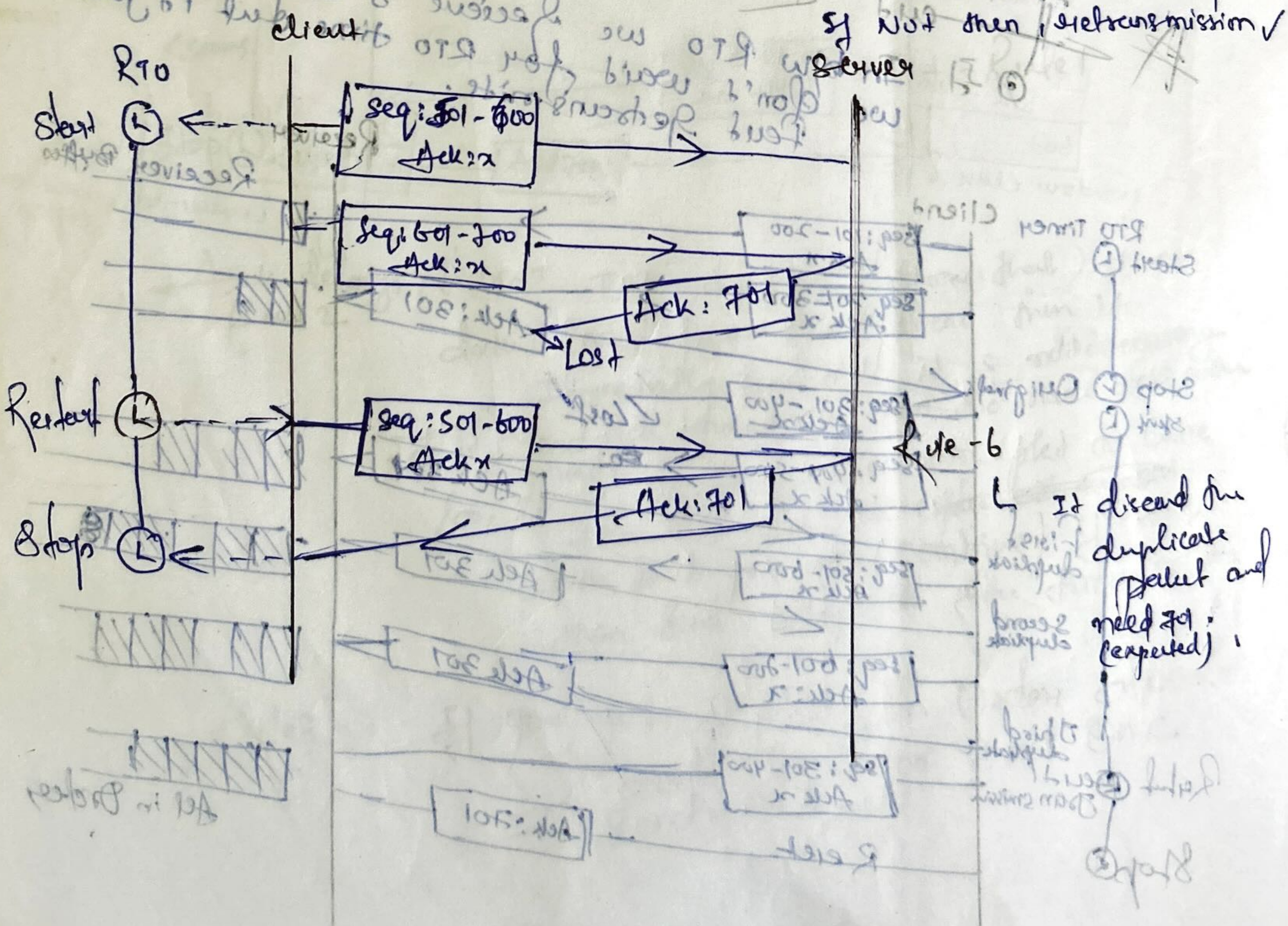
① If In b/w RTO we recieve 3 duplicates then we don't wait for RTO time out it just fast retransmits.



Lost Acknowledgment



In given RTO if Ack recieved, altho it is lost for time then also no problem
If Not then, retransmission ✓



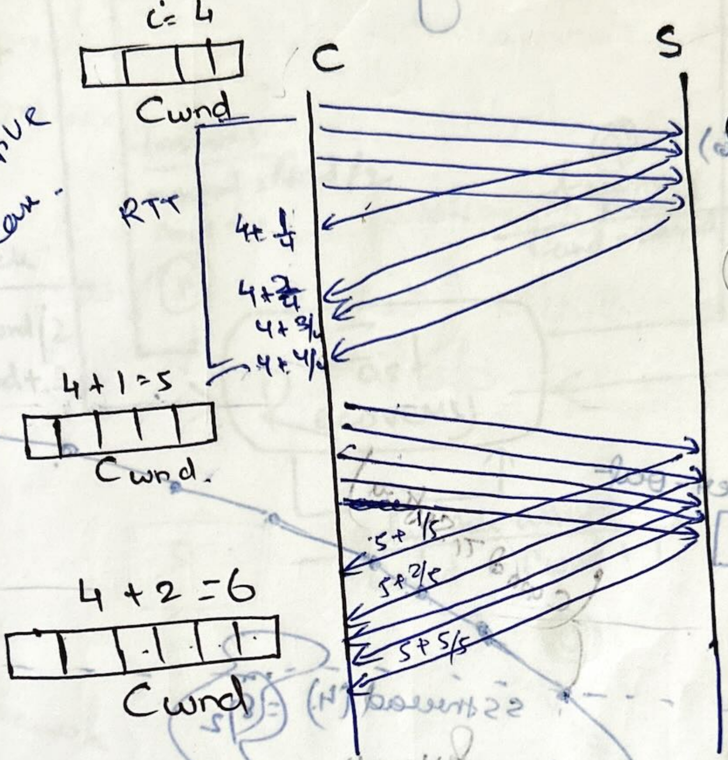
2nd phase :- Congestion Avoidance
(Additive Increase)

To avoid congestion by not doing exponential slow by doing Additive Increase_

Suppose my ssthresh is 4 (reached)
So go slow_

$$Cwnd = Cwnd + \frac{1}{Cwnd}$$

Multiplicative Increase.

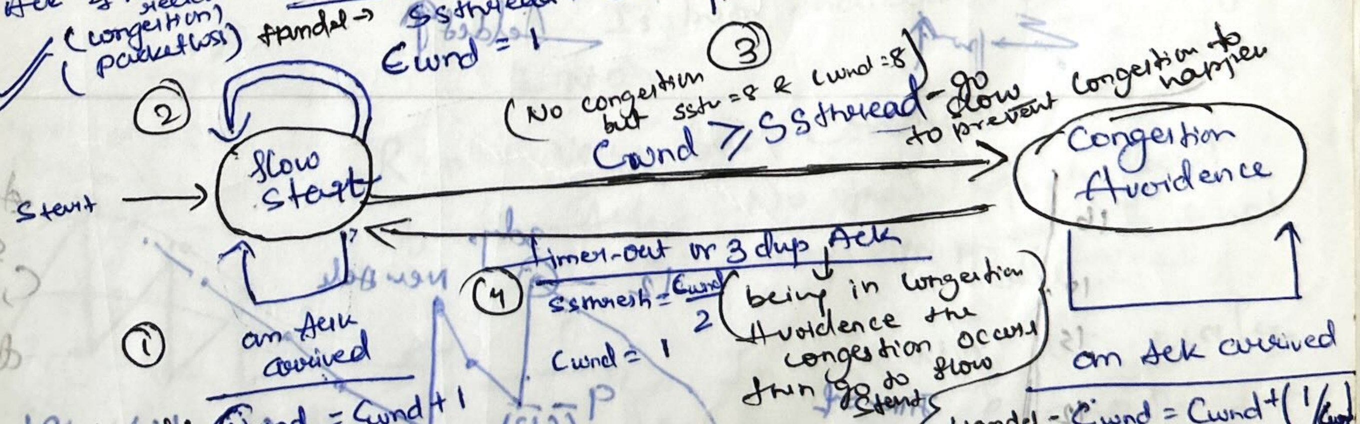


In slow start If we send 4 then It become 8 but Here If we send 4 and got all 4 then Increase by 1

TCP TAHO

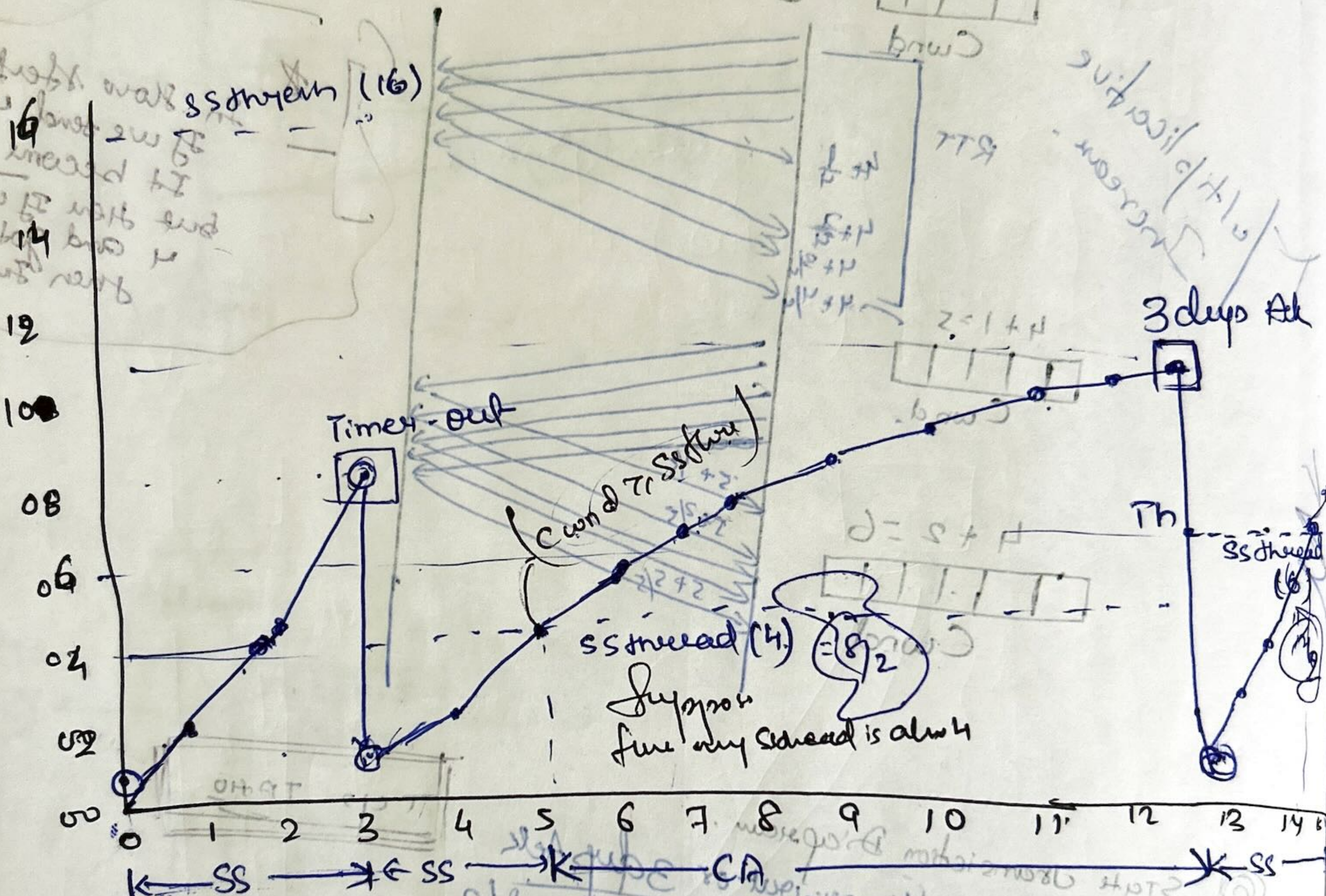
③ State Transition Diagram.

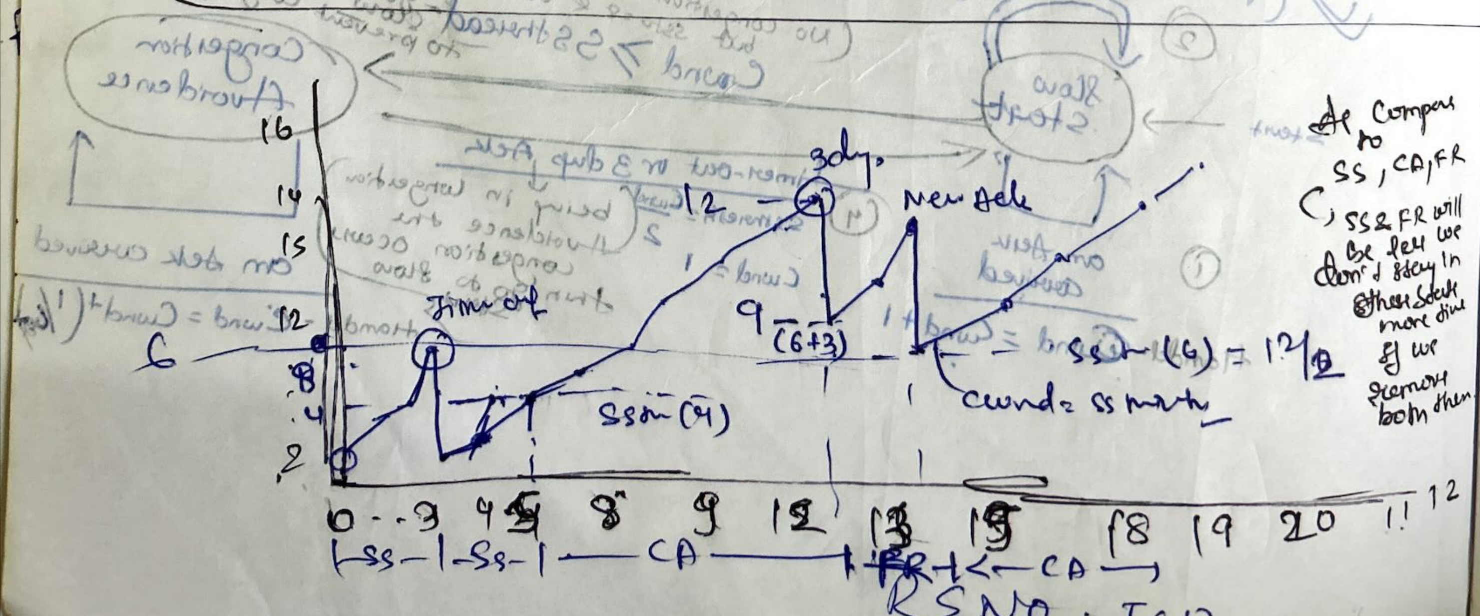
Ack If not recieved (congestion) (packet loss) Handel → timer-out or 3 dup Ack
ssthread = Cwnd/2
Cwnd = 1



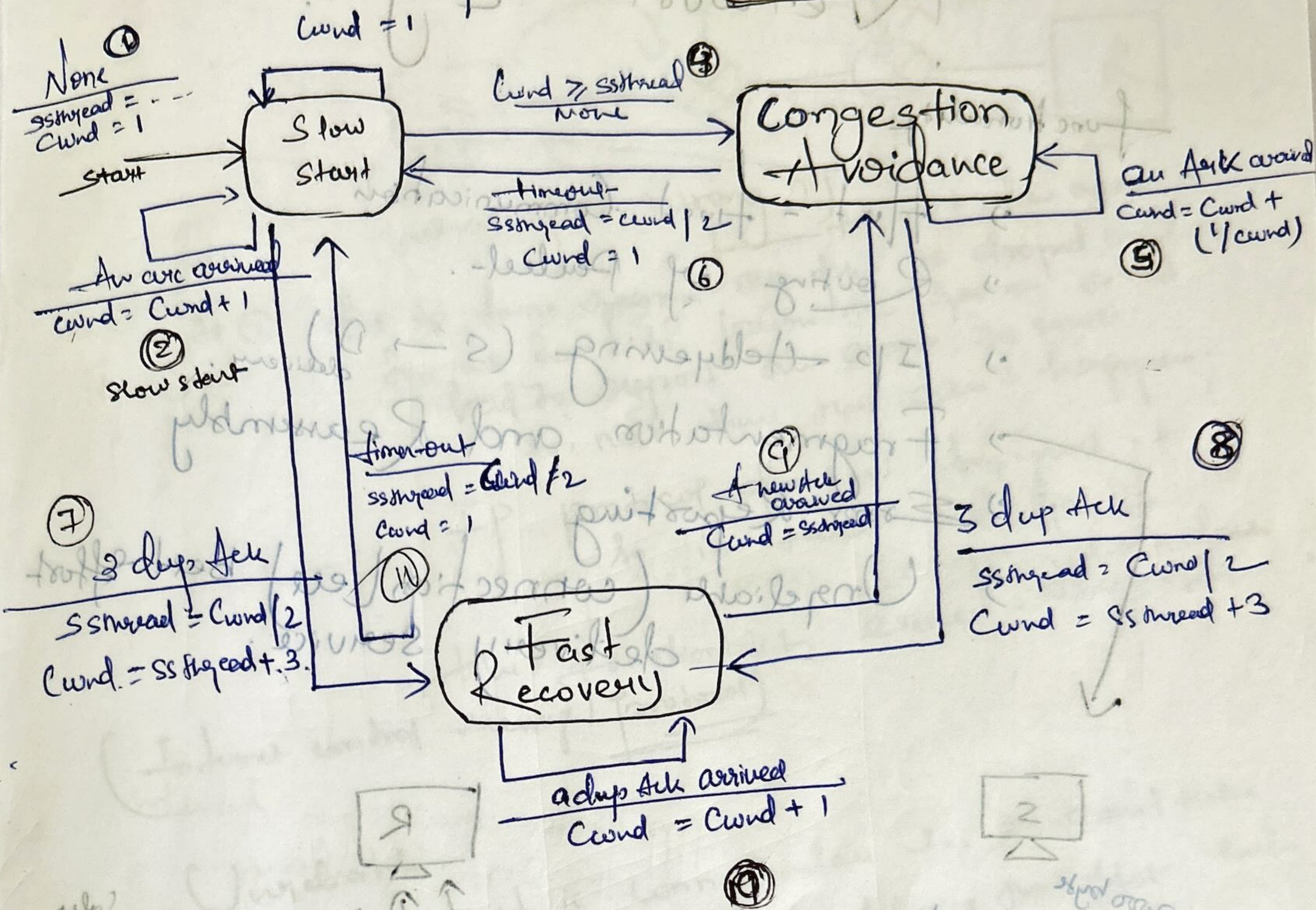
Timer out | 3 duplicate Ack

Heavier | Indicates Congestion | Lighter.





RENO TCP



Q1 As we know congestion happen by

RTO 3-duplicate Ack

but Is both indicate same congestion

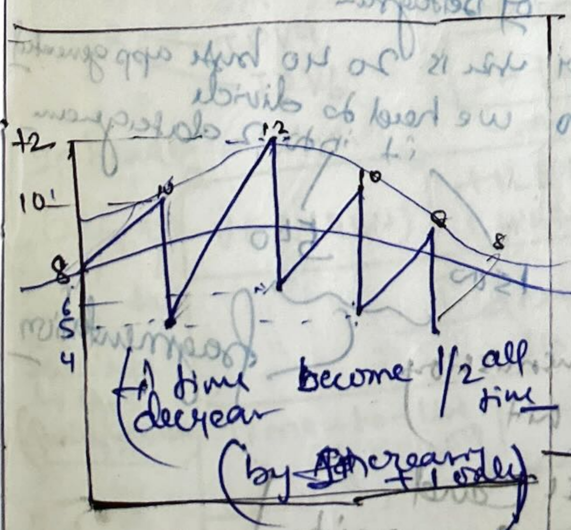
↳ NO

RTO → very bad congestion.

3-dups Ack - No that much level
It may drop p... but not all.

Avg.

So, In RENO we make one state for RTO & another for 3-dups Ack
To differenciate them

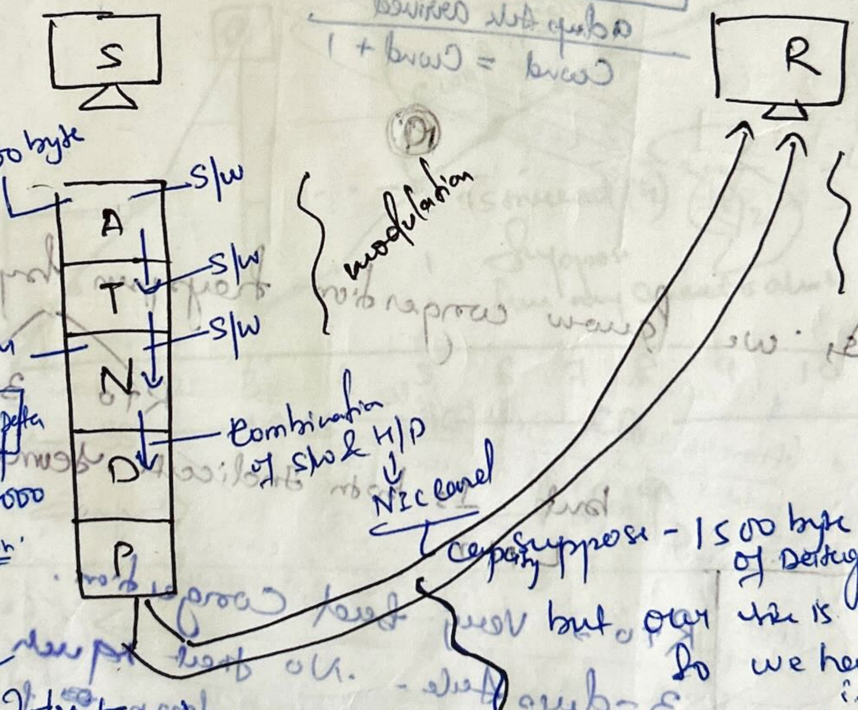


$BW = \frac{10+12+10+8+8}{5}$ It always not 9. it also

Network Layer

functionality

- Host - host Communication
- Routing of packet.
- I/p Addressing (S → D) delivery
- Fragmentation and Reassembly
- Error Reporting
- Unreliable (connection Less) best effort delivery service.



2000 byte

modulation

Datagram

Net Data	T Data	App Data
20	(20) TCP / (8) UDP / 60 max	2000

min

Combination of S/W & H/P
NIC card

capacity

demodulation takes place
So, Here Reassembly occurs done by Network layer based on characteristic of pkt

suppose - 1500 byte of Datagram

but our size is 2040 byte app generated
So we had to divide it into 2 datagram

2040 (TCP)

2028 byte UDP

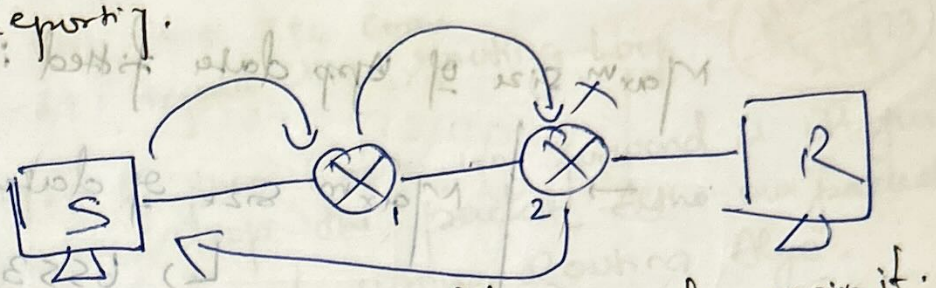
1500

540

fragmentation

(Maxm Transmission Unit
(NIC card capacity.)

Error Reporting.



(what goes wrong ?) for not done again it.

At ② due to some reason our pkt is droped or lost
So, we had to know the reason or it had to report an error to server.
so, next time this can't happen.

IP protocol doesnot support it
so it take a help of ICMP which done error reporting.
the functionality
(Internet control message protocol)

Unreliable.
As it is connectionless so, It can't take the guarantee to delivery the pkt but it can try its best.
best effort delivery it do

Different field of Header of Network layer.

Header	Payload

–20–60 byte– i.e. Datagram.
In some IP pkt — vdo stream (Data)

VER — IPV4
— IPV6

Ip header — 20 byte
20/4 = 5
HLEN

Service type
↳ QoS (Quality of service)

0, 4, 8, 16, 31

VER (4bit)	HLEN (4 bit)	Service type 8 bit	Total length 16 bit (leng of datagram)
Identification 16 bits (In fragmentation)		Flag 3 bit	Fragmentation Offset 13 bit
Time-to-Live 8 bit — loop routing	Protocol 8 bit	Header Checksum 16 bits	
Source IP address (32 bit)			
Destination IP Address (32 bits)			
Option + padding (0 to 40 bytes)			

Max^m size of App data fitted into a Datagram.

→ Max^m size of datagram

↳ 65535 bytes

↳ 65535 − 20 − 20 = TCP

65535 − 20 − 8 = UDP

Max^m size of app data generated by an App

↳ No limit.

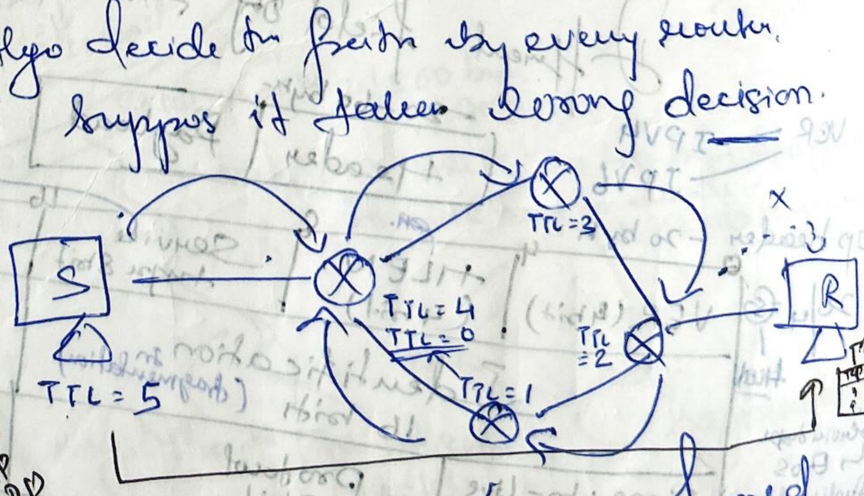
Max^m size of App data that can transfer to Data link layer.

MTU − 40 (TCP)

Time to Live (TTL): no. of hop count.

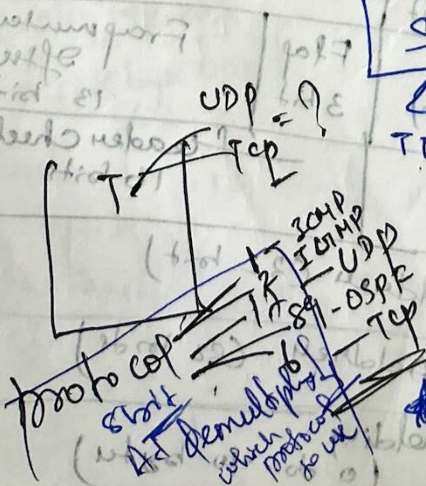
Routing Algo decide the path by every router.

Suppose it takes wrong decision.



Loop is formed

it never deliver the pkt



* Routing loop

So, To stop this TTL come
It break this routing loop.

If TTL = 0 then Router not forward it it just drop the packet. Altho our packet is correct due to wrong Routing Algo.

Not to make congestion this is done

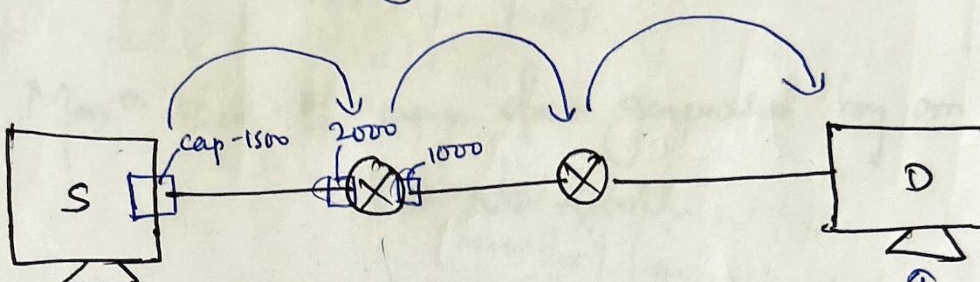
Mhn 5/10/23

Fragmentation... & Reassembly..

Maxm size of datagram = Network Header + T_H + App Data.

Under which Condn fragment Happen.

↳ datagram size > MTU (Maxm Transmission limit)



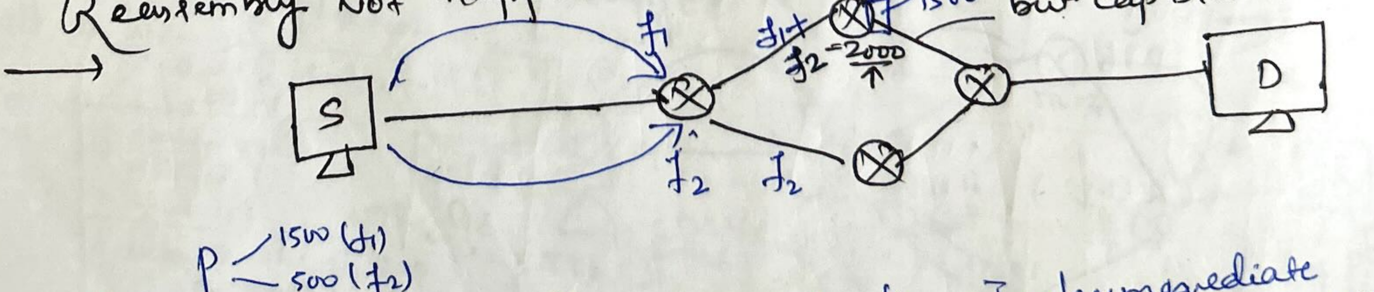
fragmentation May (P) Happen
- f_1 — 1500 — 1000, 500
- f_2 — 500

* Every system has its own capability. One NIC cap. may vary from another.

Reassembly at Destination only.

* fragmentation may happen at source end and intermediate router, but never at the destination.

Reassembly Not happen at router?



P < 1500 (f_1), 500 (f_2)

Reassembly doesn't happen at the Intermediate router as it ~~Compare~~ Consumes a lot of resource and other router can have a lower MTU value which results in fragmentation again.

→ Always fragments of the packet must be reassembled to reduce the error and to recover the Original message.

Responsibility for doing fragmentation and Reassembly.

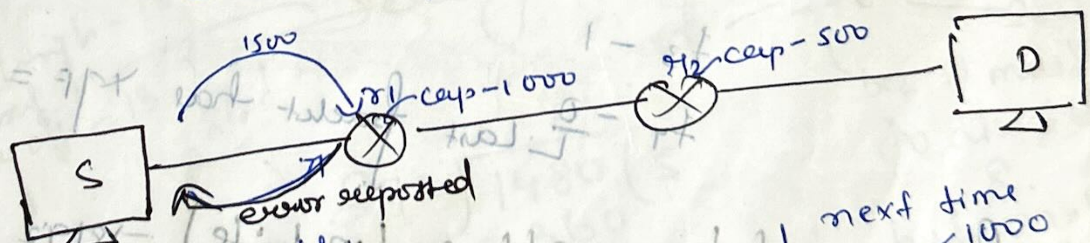
- Identification (16)
- Flags (3) → | Res 0 | DF | MF |
- Fragmentation Offset (13)

= Identification field helps in Identifying the fragment of the same packet.

P_1 — f_1, f_2 — x is populated

P_2 — f_1, f_2 — y is populated

= Flags
- 1st bit - reserved
- 2nd bit DF (Don't fragment)
- 3rd bit MF (More fragment)



P — 1500 (f1), 500 (f2)

2000

DF = 1 - Means do not fragment.

so it can move nex	
as,	1500 > 1000 (MTU)

so, It drop some packet and error must report back to server.

So Next time It Can't happen Again →

next time P — 1000, 1000

so, Intelligently It can know the capacity of r1 i.e 1000

again the r_2 drop so, by error report
It can again know the cap of r_2 so,

P.(2000) → 500, 500, 500, 500

So, the whole process called PMTU
(path MTU Discovery)
when path is discovered so, DF = 0.

Since, fragmentation is a costly process we try to avoid it at the Intermediate node with the help of DF flag set to 1 and this process is called as PMTU

MF (More fragment)

It help to Identify the last fragment arrived to start the Reassembly.

f_1 - 1
f_2 - 1
f_3 - 1
f_4 - 0 ← Last packet has MF = 0

fragmentation Offset (13 bits) → 8192 (2^{13}) bytes.

Search the Order of fragment to assemble.

0	500	1000	1500
500 f_1	500 f_2	500 f_3	500 f_4

f_1 → 0 (frag offset)
f_2 ⇒ 500
f_3 → 1000
f_4 → 1500

pos. of that fragment with respect to Original datagram.
f_4 → 1500, f_2 → 500, f_3 — 1000, f_1 → 0

~~So, we find the Scaling factor,~~ Directly.

Q. Suppose $p = 300$ and it recieved to D, So, we don't want any reassembly and no fragmentation happen so, How we know that

If Offset = MF = 0 at same time.

So, we find the Scaling factor.

ex:- 0 \ $\frac{4000}{8}$ ~~2500~~ Store in frag. Offset (13 bit) $2^{13} = 8192$

4000	25535

65535 datagram

So, It can't store $2^{16} / 2^{13} = 2^3 = 8$

Q. Datagram = 1500 byte.

MTU = 296 byte

How many fragment = ?

Details of each fragment ?

1500 byte.

20	Payload

segment 1480 byte

Every fragment should contain the network layer header for further transportation and to reach the destination.

MTU - 296 byte (Ph Layer)

payload = 296 - 20 = 276 byte.

Total length - 16, frag offset - 13 bit, 2^3

Scaling factor is 8

payload - 276 - 272 ✓ - 280

(bcz it must had to divisible by 8. but max^m value is 276 so we take 272)

$$272 \overline{)1480}(5$$
$$\underline{1360}$$
$$120$$

272 + 272 + 272 + 272

272 + 272 + 272 + 272 + 120

Identification	Flag	Offset	Payload / Datagram
f_1 x	MF = 1	0 — kullo aujkoi pagmu korbo 0 - 272	272 / 296
f_2 x	"	34 — 272/8 — 272 - 544	"
f_3 x	"	68 — 544/8	"
f_4 x	"	102	"
f_5 x	"	136	"
f_6 x	MF = 0	170	120 / 140

0, 34, 68, 102, 136, 170

f1	f2	f3	f4	f5	f6

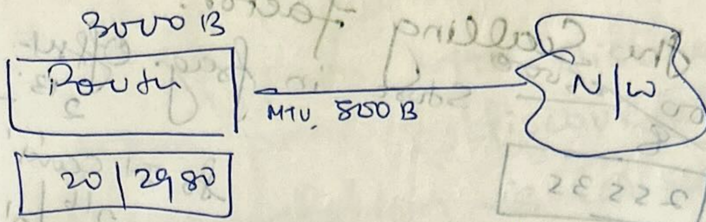
Q. $\boxed{20 \,|\, 2980 \text{ payload}}$ MTU = 500 B

Datagram $\to$ 3000

How many fragment

MF = ? offset = ?

Total length value of all



so, fragmentation happen =

500 B ⇒ 20 + 480 B (Header, Data)

payload = 500 - 20 = 480

Scaling factor ~~480/8~~ - 480

$$480 \overline{)\,2980\,}\,(6$$
$$\underline{2880}$$
$$100$$

2980

= 480 + 480 + 480 + 480 + 480 + 480 + 100

	MF	Payload	Datagram
f1	1	480	500
⋮	⋮	⋮	⋮
f7	0	100	120

It is not imp that all packet follow the same route it may follow different route. ∴ header added to all.

Ip Addressing :-

Total number of Ip address = 2^{32} = 4 bign

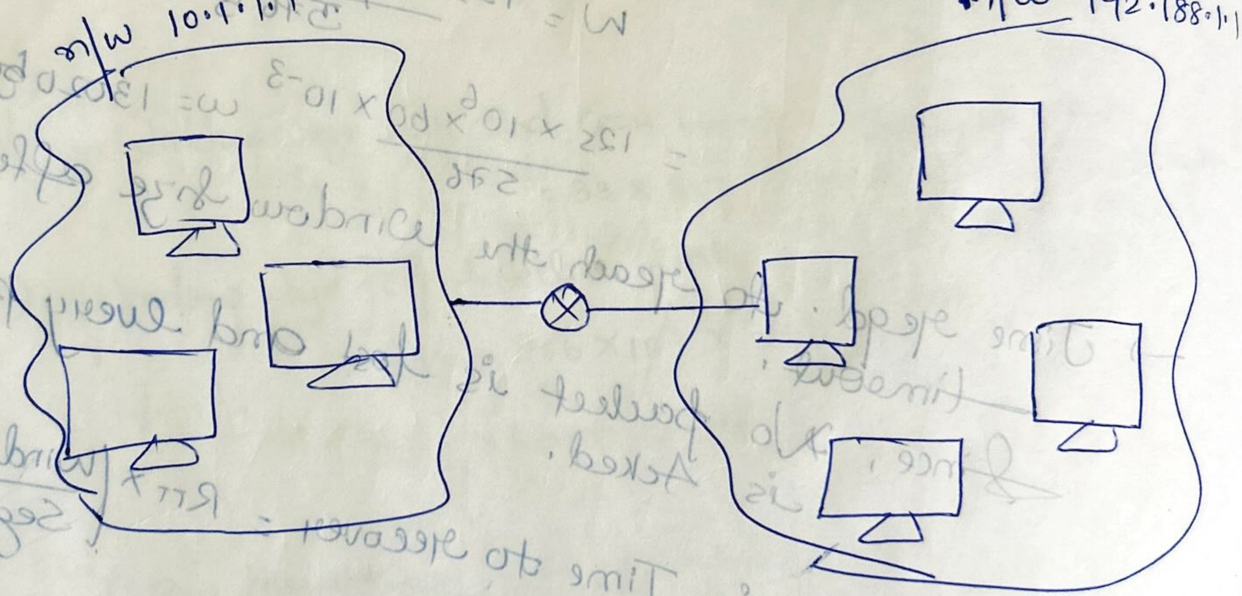
|← 32 bit →|

IPV4

0.0.0.0 → Lowest Ip Address

255.255.255.255 → Highest Ip Address

Prefix	Suffix
network id	Host id



* If we need to communicate in the same network we don't need routers but to communicate to a device on another network we use router.
The Network Id tells whether the Ip address belongs to same or the other network and host id tells the machine where the packet should reach within that network.

Classes of Ip Address:- (In 1980)

five class are there A, B, C, D, E

Before 1980.

Ip Address is of 32 bit

8 bit	24 bit
network bit	Host bit

fixed

At that time no that much need of It so, It is ok but after 1980 - It will not scalable

2^8 - 256 n/w possible
 └ less

2^{24} - too must host

∴ classes come

So,

0------- . ------- . ------- . -------
8 bit . 8 bit . 8 bit . 8 bit

1 octate = 1 byte

← 32 bit →

so, 2^{32} possible Ip Address

So, to find there class we had to check :—

In class A

0----	
8 bit	24 bit Host id

N/w id

so, 1 bit is fixed to 0

so, 2^{31} - total Ip Address of class A

Default Mask = 255.0.0.0

Ip Add - 64.0.0.8 ← 8th host
 └ This belong to which N/w

So, 01000000.0.0.00001000

Default Mask 11111111.0.0.00000000

01000000.0.0.0
 └ 64.0.0
 └ This is N/w

So, | 0---- | 8 bit → 2^7 possible permutation

[0 — 127]

but, 00000000 → 0
01111111 → 127 } we can't use
 └ reserved by IANA

So, Total Number of ~~IP~~ Used by us is

128 - 2 = 126

No. of Host possible is 2^{24} - 2

64.0.0.0 - whole google
64.255.255.255 - Broadcast

Classes of Ip Address:- (In 1980)

five class are there A, B, C, D, E

Class B

←—— 32 ——→

10 - - - - - | | |

8 bit | 8 bit | 8 bit | 8 bit

so, Range of class B - 2^6 = (128 — 191)

64 { 10 000000 — 128
:
10 111111 — 191

2^{30} - Total Ip Address possible

10 - - -	
16 bit	16 bit
N/w id	Host id

2^{16} Host possible but we use $2^{16} - 2$

2^{14} N/w possible

ex:- 130 . 28 . 4

128 - 191

so, class B

Belong to which N/w

130 . 2 . 3 . 4

so, Default Mask ? 255 . 255 . 0 . 0

130 . 2 . 0 . 0

This N/w.

Brodcast

130 . 2 . 255 . 255

130 . 2 . 0 . 0

:

130 . 2 . 255 . 255

$2^{16} - 2$

FID

class C

← 32 →

110 . . .

2^5 - (192 - 223)

110 0 0 0 0 0

Total no. of Ip Address - 2^{29}

110 1 1 1 1 1

110 . . .	
24 bit N/W	8 bit Host

2^8 Host possible

used → $2^8 - 2$ ⇒ 254

2^{21} N/W position.

ex:- 194.2.3.4

(192 - 223)

class C

N/W Id - 194.2.3.4

255.255.255.0

194.2.3.0 - N/W Id. Tnw 2-wins

194.2.3.255 - Brodcast Address.

class D

1110 . . .

← 32 bit →

2^4 - (224 - 239)

Total no. of Ip Address - 2^{28} = 6.25%.

No Host | No N/W

Reserved for Multitasking

Email Group | Brodcast

class E

← 32 bit →

1111 . . .

2^{28} - Total no. of Ip Address

Reserved for Military purpose

(240 - 255)

Wed and Identify Host / N/W

Type of Ip Address :-

Given by

i. public → paid (always Unique) → ICANN

ii. private → It's Range is A - 10.0.0.0 — 10.255.255.255

B - 172.16.0.0 — 172.31.255.255

C - 192.168.0.0 — 192.168.255.255

iii. loopback → It loops the n/w and It is used for testing { 127.0.0.0 - 127.255.255.255

Default Loopback Ip Address → 127.0.0.1

Communication

Unicast (one-one)

Multicast (one-Many)

Broadcast (one-all)

Problem In Classful Ip Addressing

let, an Organisation

100 Computer

If Use class 'C' addressing → 256

U = 100/256 i.e. less than 50%

50% Ip addresses are wasted.

Ip Address :- 201.20.30.40

Class 'C'

Broadcast Id = 201.20.30.255

Total host = 256 but 2 is not in use

so, N/w Id

so, Default Mask = 255.255.255.0

so, and operation

201.20.30.4

Host Id

Class N/w Id → 201.20.30.0

L Kis ~~class~~ N/w ka hai

Last host = 201.20.30.255 is last but not in use

so = 201.20.30.254

Subnetting

ex:- 192.168.10.25 (class C; N/W | host)

Rule 1:- N/W Id - All the N/W bit kept intact and all the host bit are set to '0'.

192.168.10.0 — N/W

Rule 2:- Broadcast Id - All N/W bit kept intact and all the host bit set to 1.

192.168.10.255

Rule 3:- No. of subnets/Networks, depends on no. of bits dedicated for the subnet

Subnet (n)

No. of Host in the N/W or

$(2^n - 2)$

by fixing this we divided a N/W in two parts - one is '0' and another is '1'

N/W Id

No. of Host : 100

Let, only 7 bit req for 100 host

192.168.10.0 0000000

this 7 bit is enough to represent 100 — $2^7 = 128$ so, less wastage happen

N/W subnet bit 1

0 1111111

Subnet Id

192.168.10.0

192.168.10.127 Broadcast Id

range - 192.168.10.1 to 192.168.10.126

126 — Host

N/W Broadcast add - 192.168.10.255

192.168.10.1 0000000

2nd Subnet bit

1 1111111

192.168.10.128

192.168.10.255 Broadcast

range - 192.168.10.129 to 192.168.10.254

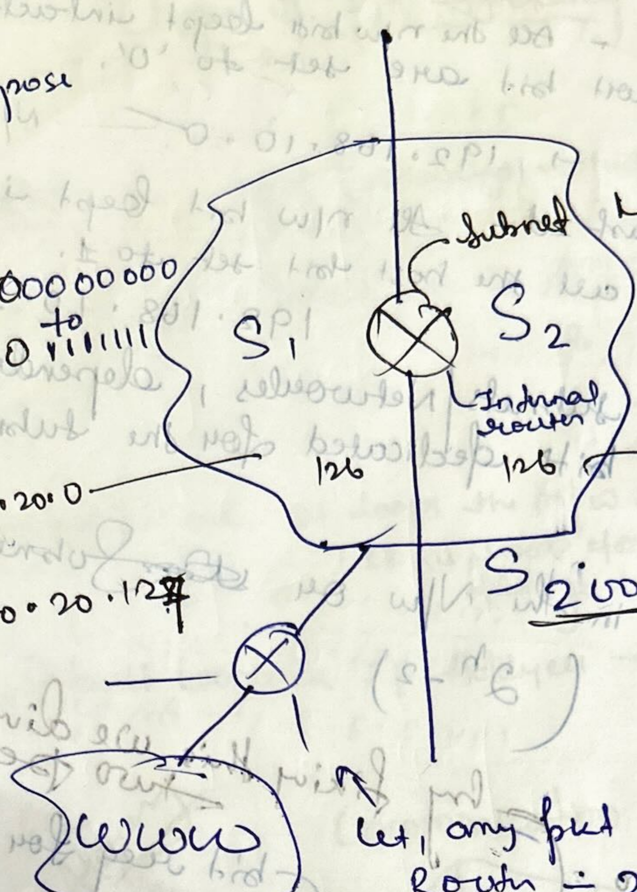
Subnet Mask of 'C' - 255.255.255.128

Here Only 26 wastage is there for compare to 156. too much less

Subnet Mask ?

Suppose

Mask = 255.255.255.128



200.10.20.00000000 to 200.10.20.01111111

200.10.20.10000000 to 200.10.20.1111111

200.10.20.128 to 200.10.20.255

200.10.20.0 | 200.10.20.127

S 200.10.20.0

Let, any pkt come at this Router - 200.10.20.10 - IP

the Default Mask 255.255.255.0

(And) operation

200.10.20.0

Now It had to known that It will go to S_1 or S_2 done by Internally Router.

Done by Subnet Mask

255.255.255.10000000

$\underbrace{255.255.255}_{\text{Class C}}$. 1 $\rightarrow$ No. of bit reserve In subnetting ; 128

Let, a packet come with dest. addr. 200.10.20.15

Now by performing And operation It come Internal router by the help of Subnet Mask It will decide to go to which S_1 or S_2

$$\begin{array}{r} 200.10.20.15 \\ 255.255.255.128 \\ \hline 200.10.20.0 \end{array} \longrightarrow S_1$$

If

$$\begin{array}{r} 200.10.20.130 \\ 255.255.255.128 \\ \hline 200.10.20.128 \end{array} \longrightarrow S_2$$

Q. Suppose you have a host on this N/w with an ip of 154·71·150·42 and the subnet mask is 255·255·248·0 then findout the subnet address.

154·71·150·42

↳ class 'B'

Default Mask = 255·255·0·0

~~Brodcast Id = 154·71·150·255~~

Subnet Mask - 255·255·248·0

8 + 8 + $\underbrace{11111}_{5}$000

N/w → 21

H/st - 11

No change

154 · 71 · 1001|0110 · 42

255 · 255 · 1111|1000 · 0

154 · 71 · 10010000 · 0

↳ 154·71·144·0

↳ Subnet Address

154·71·10010000·0

As 154·71·1111111·11111111

N/w bit one Intact, set to '1'

Brodcast Id 154·71·151·255

Q. Given the prefix 192.168.1.0/24, find the N/w Mask, N/w Id, what should be the length of subnet mask allowing up to 4 subnets, 1 subnet means 1 sub n/w

~~CIDR~~

Q

Classless Addressing (CIDR)

→ No class
→ Only Blocks

32

Block ID	Host ID

→ Notation

x.y.z.w/n → mask (11111) | No. of bit represent block (N/w)

ex:- 200.10.20.40/28
↳ means 28 no. of 1's
N/w Id is reps by 28 no. of bit
so, 32 − 28 = 4 Host Id

28	4

←32→

so, 2^4 − 16 host

No of host → 16

200.10.20.40

Default Mask — but No class concept is here
but n → mask

11111111 . 11111111 . 11111111 . 1111 0000

$\underbrace{11111111.11111111.11111111.1111}_{28\ N/w}\ \underbrace{0000}_{4\ Host}$

Mask — 255.255.255.240

And op.

N/w Id — 200.10.20.32/28

or.

$\underbrace{200.10.20.0010}_{28\ bit\ as\ it\ is}\ \underbrace{1000}_{make\ all\ host\ \to\ 0}$

↳ 200.10.20.32/28

01000
10000
00000

Rules → ① Address should be contiguous
② No. of addresses in a block must be in power of '2'
③ First address of every block must be evenly divisible with size of block.

Q. Given the prefix 192.168.1.0/(24), find the N/w Mask, N/w Id, what should be the length of the subnet mask allowing up to 9 subnet, subnet mask & subnet Id.

segment N/w bit.

so, $\underbrace{11111111.11111111.11111111}_{24}.00000000$

N/w Mask = 255.255.255.0

N/w Id →
192.168.1.0
255.255.255.0
192.168.1.0

24	8
Block	Host

$2^8 = 256 - 2 = 254$ host

Q. Given the host address 192.168.1.177/29, what are the subnet & broadcast address?

N/w bit − 29

192.168.1.177

Mask $\boxed{255.255.255.11111000}$

192.168.1.10110001
255.255.255.11111000
192.168.1.$\underbrace{10110000}_{176}$

Subnet Id — 192.168.1.176

Subnet Mask − 255.255.255.248

$\underbrace{192.168.1.10110}_{29\text{ bit}}\underbrace{000}_{111}$

Broadcast Id − 192.168.1.183

Regular Subnetting :- The size of cost of subnet is always fixed

Original Network :- 192.168.1.0/24

$\underbrace{00}_{\text{subnet bit}}\ \underbrace{000000}_{\text{host bit}}$

00
01
10
11 } → 4 subnet can we prepare

$2^6 = 64$ bit host in each.

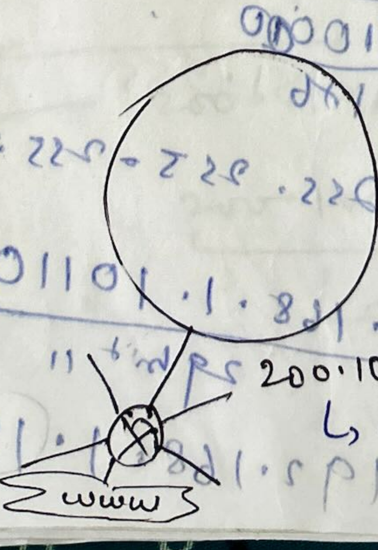
→ future we want to prepare 4 subnet → 1st – 16, 2nd – 16, 3rd – 50, 4th – 100 } Req. not satisfied bcz max no. of host is 64 only.

→ So, we can't do the previous subnet so we don't use i.e. it does not fit in real time. So, we don't use this instead of this we use $\boxed{\text{VLSM}}$

VLSM (Variable length subnet masking)

↳ convert large network into small network but I should have this much flexibility that I can make one subnet bigger, and other smaller, A/q to our req.

o Make different size subnet.



200.10.20.0

↳ class C

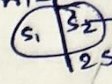
so, N/w bit – 24

host bit – 2^8 – 256 but 254 are usable bcz last 255 is Direct broadcast and 1st is N/w bit.

Only host bit to change

S_1 200.10.20.1 - - - - - - -

Reserved means.

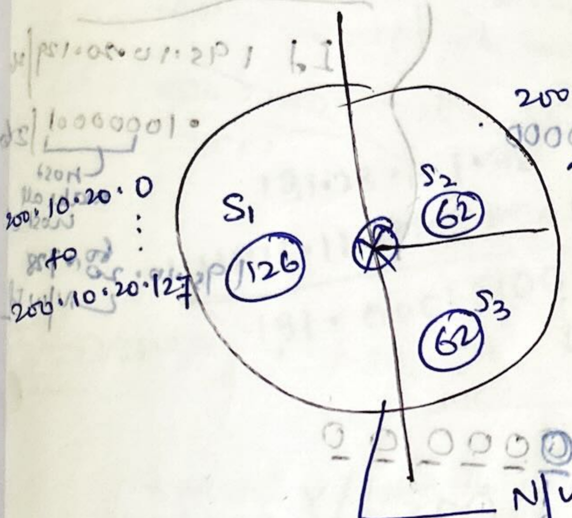


200.10.20.1 - - - but we want again this in two part then again reserve one bit

S_2 200.10.20.10 - - - Reserved

S_3 200.10.20.11 - - -

Variably subnet want: 100, 50, 50



200.10.20.128 to 200.10.20.191

200.10.20.192 — Subnet N/W Id
to
200.10.20.255 — Subnet direct brodcast Address

N/W Id — 200.10.20.0
Direct brodcast Address — 200.10.20.255

but due to 3 subnet
256 − 6 = 250
we not use.

let, suppose a packet come and Id know that N/W Id

Now, Id had to know which subnet — so, done by Internal router by Subnet Mask.

255.255.255.10000000 — S_1
128

class 'C' — 255.255.255.11000000 — S_2, S_3
192

Subnet Mask.

Subnetting in Classless Inter-domain Routing
(Subnetting in CIDR)

26 - N/w Bit → 195.10.20.10000000

N/w | Host

2^6 — host (64) 62 - usable

195.10.20.128 | 26

Id 195.10.20.129/26

.10000001 | 26

host

make all host 0

to 195.10.20.128

N/w Id

Mask - 11111111.11111111.11111111.11000000

255.255.255.192

For Subnetting

↳ 195.10.20.10000000

N/w bit

fix

Subnet Id
195.10.20.128 | 27

to

195.10.20.159 | 27

Subnet broadcast

10011111.

195.10.20.10100000

to

195.10.20.10111111

fix 30

195.10.20.160 | 27

to

S2 → 195.10.20.191 | 27

CIDR receive a packet with address 131.23.151.76.
The Router routing table shows following

131.23.151.76
131.16.0.0/12

11111111.11110000.0.0
And

131.0001 0000.0.0

131.16.0.0 It is match
we send.

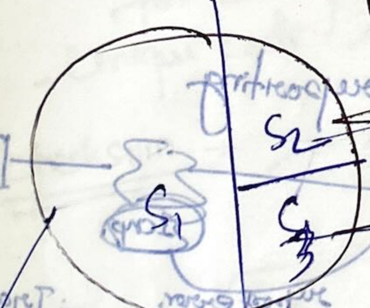
131.23.151.76 131.28.0.0/14

11111111.11111100.0.0 ← 14

131.00010100.0.0

131.20.0.0 X

VLSM CIDR



245.248.128.0/20

12-bit host

255.255.11110000.0

Route

245.248.1000|0000.00000000

N/W fixed for subnet

245.248.10000111.11111111

245.248.128.0/21
to
245.248.135.255/21

S1

245.248.10001000.0
to
245.248.10001011.255

245.248.136.0/22
to
245.248.139.255/22 S2

245.248.10001100.0
to
245.248.10001111.255

245.248.140.0/22
to
245.248.143.255/22 S3