Prepared and Written By

RAZEEN AHMED

CSE-421

[COMPUTER NETWORKS]

HANDWRITTEN NOTE

Computer Science and Engineering
BRAC University

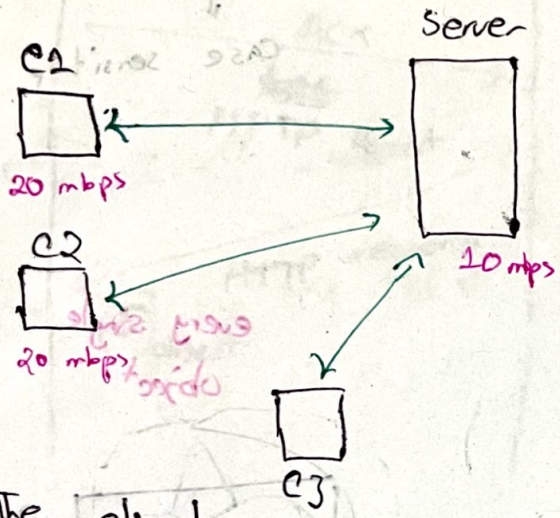
Github: github.com/razeen
LinkedIn: linkedin.com/in/razeenahmed

#APPLICATION LAYER:-

(i) It provides an interface.
It prepares the data for the next layer.

The PC communicates by following two architecture or methods:-

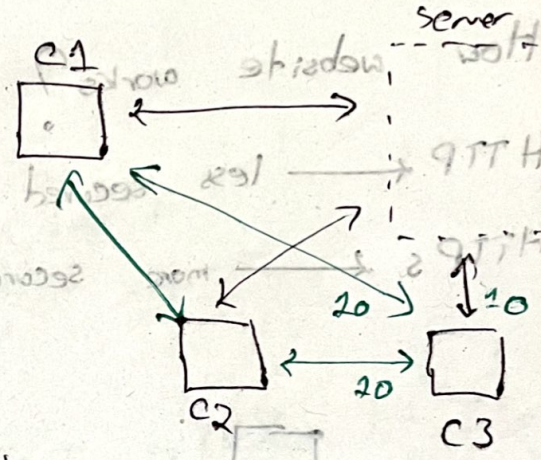
(i) CLIENT SERVER



The clients always communicates with a single server.

→ The client can't download m

(ii) Peer-to-Peer P2P



For client communicates with the server and the clients.

→ The download speed of different clients gets added up to its limit.
So, C3 download = 30 mbps

★ RISK :-
This te connection is less secure or it has some security issue.

The WEBSITE

URL : http :// www.nytimes.com /tech /index.html

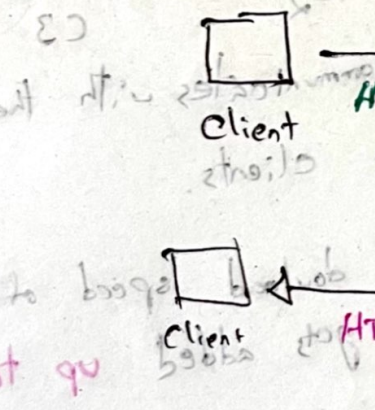
- http → AP transfer protocol
- www → host name
- nytimes → domain name
- .com → top level domain
- /tech → file path
- /index.html → file name

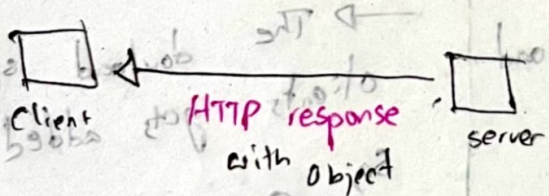
Case sensitive

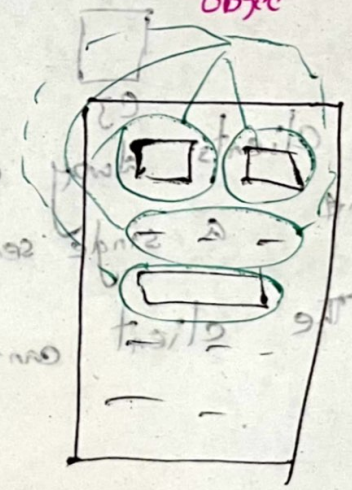
★ How website works?

HTTP ← less secured

HTTPS ← more secured







Requests for every single object at a time

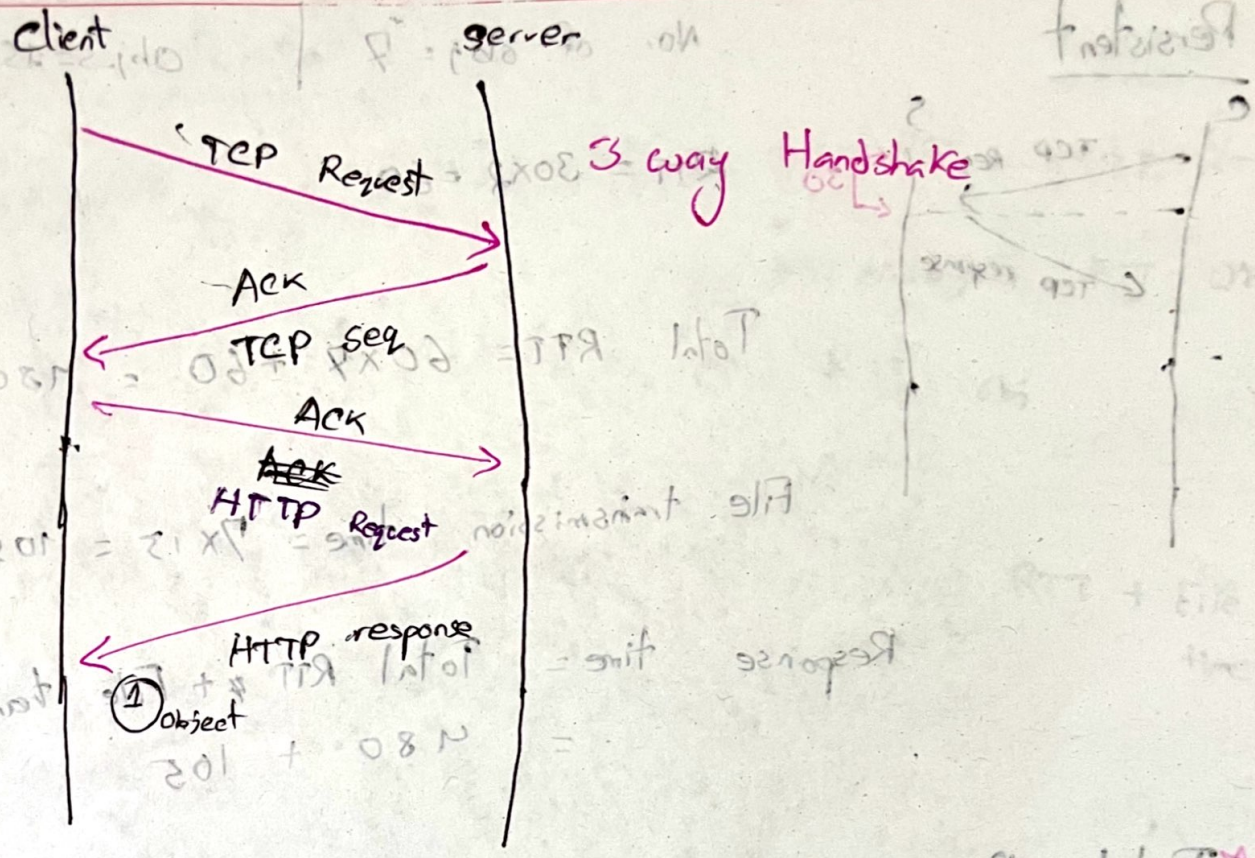
<P> </P>
<a>
<img> </img>

every tag is every object

★ Application layer uses TCP protocol.

↓ why?

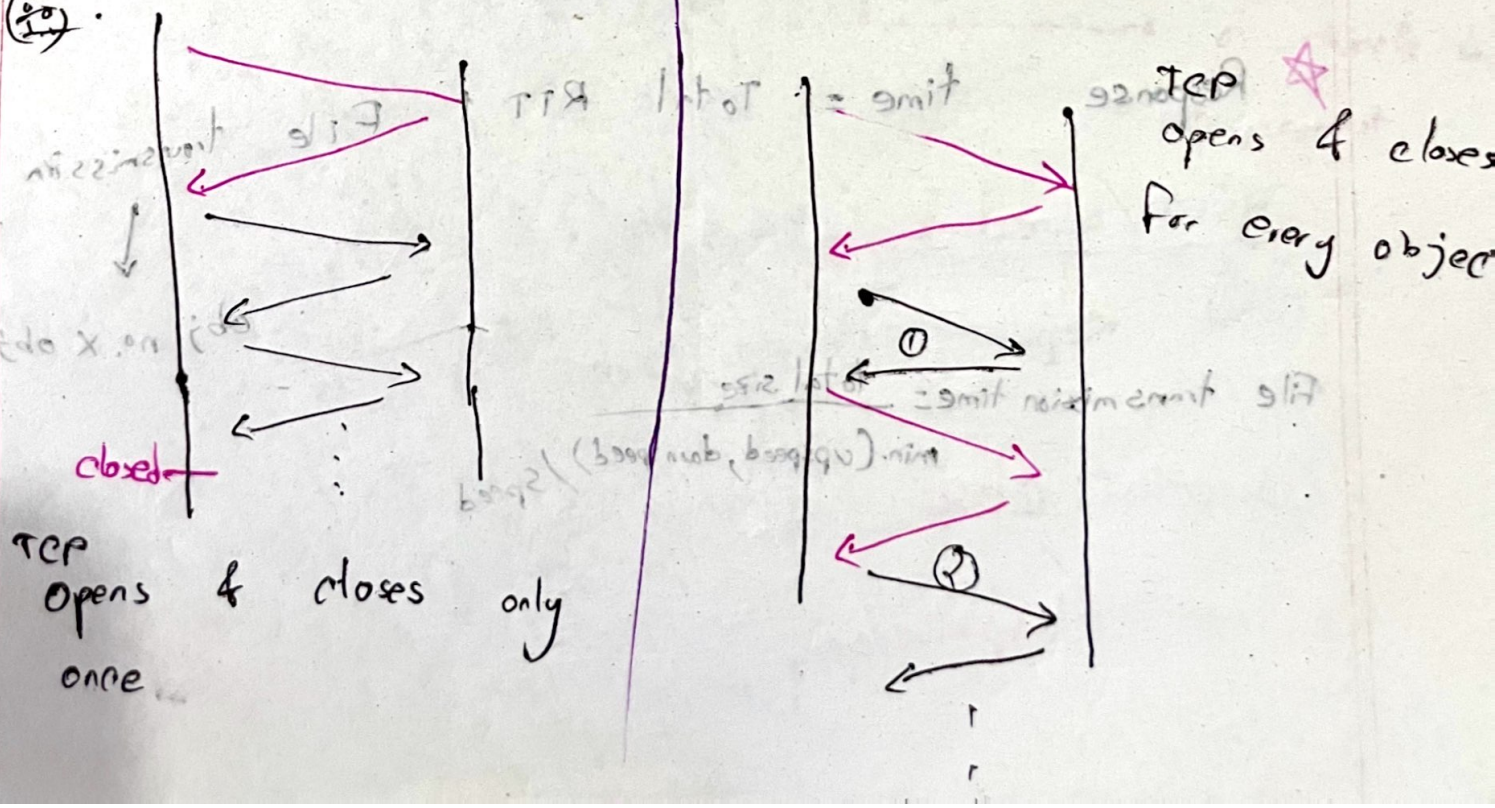
Because it encrypts the data Creates a secured line.



HTTP Connection two types:-

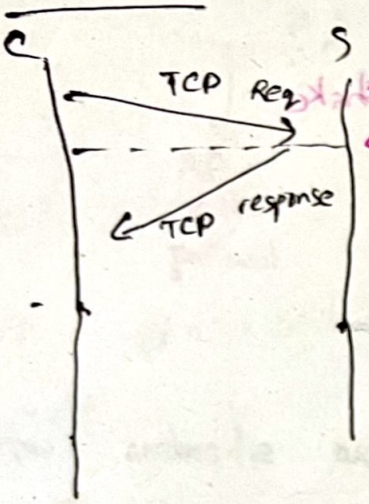
(i) Persistent

(ii) Non persistent



Persistent

No. of obj = 7 | Obj_S = 15 ms



RTT = 30 × 2 = 60

Total RTT = 60 × 7 + 60 = 480

File transmission time = 7 × 15 = 105

Response time = Total RTT + File transmission
= 480 + 105

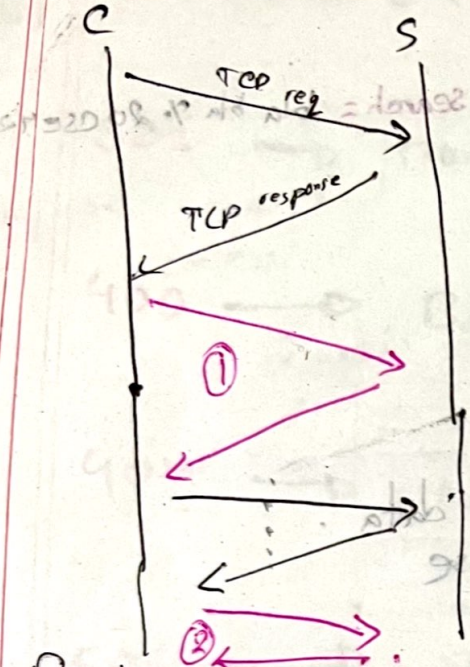
☆ Total RTT = TCP RTT + HTTP RTT
= RTT + RTT × obj

☆ Response time = Total RTT + File transmission time

obj no. × obj_S

File transmission time = $\frac{\text{total size}}{\min(\text{upspeed}, \text{downspeed}) / \text{Speed}}$

Non Persistent



Total RTT = TCP RTT + HTTP RTT

= RTT x obj + RTT obj

= 2 RTT x obj

Response Time = Total RTT + File transmission time.

Pipelining :-

~~obj~~

Effect objects, $Obj_e = \left\lceil \frac{obj}{p} \right\rceil$

Total no. of object

no of objects send per request.

HTTP REQUEST

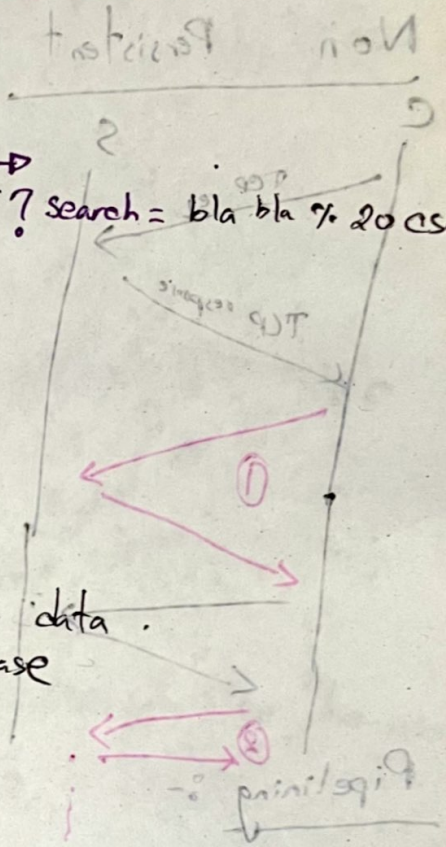
METHODS:-

(i) GET → To fetch information ex→ ?search=blabla%20cse4210,

* Not secured.

(ii) POST → To insert data.
→ when we entry new data in database

* It is secured



(iii) PUT → To update an existing information.

(iv) DELETE → To delete the data.

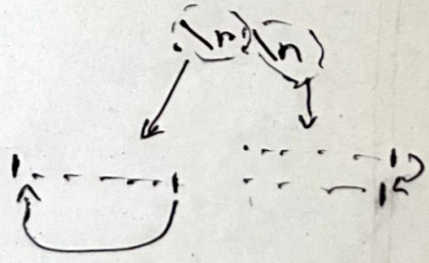
→

GET /somedir/index.html HTTP/1.1 \r \n

Method | file path | file name | Protocol | Protocol version

Connection: close [non persistent connection]

: Keep alive/open [persistent connection]



HTTP RESPONSE

200 → OK

301 → Moved parmanently (to a new location)

400 → Bad Request (The request is corrupted or not properly formatted)

404 → Not found

505 → VERSion not supported, engine not supported

protocol ↓ HTTP/1.1 status code ↓ 200 OK

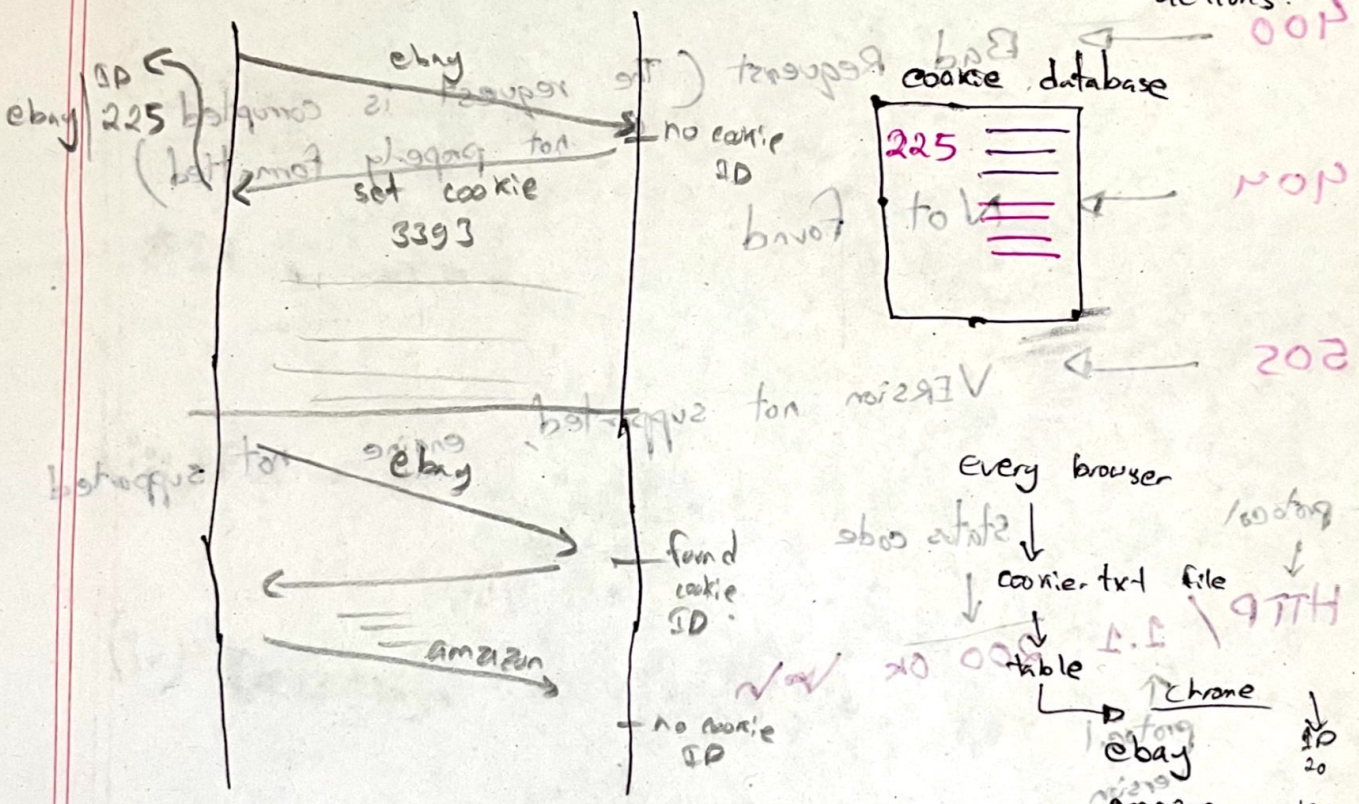
protocol version

#Cookies

NOTE:- HTTP IS A STATELESS PROTOCOL

Cookies + HTTP

↳ saves the previous actions.



Every browser

↓

cookie.txt file

↓

table

↓

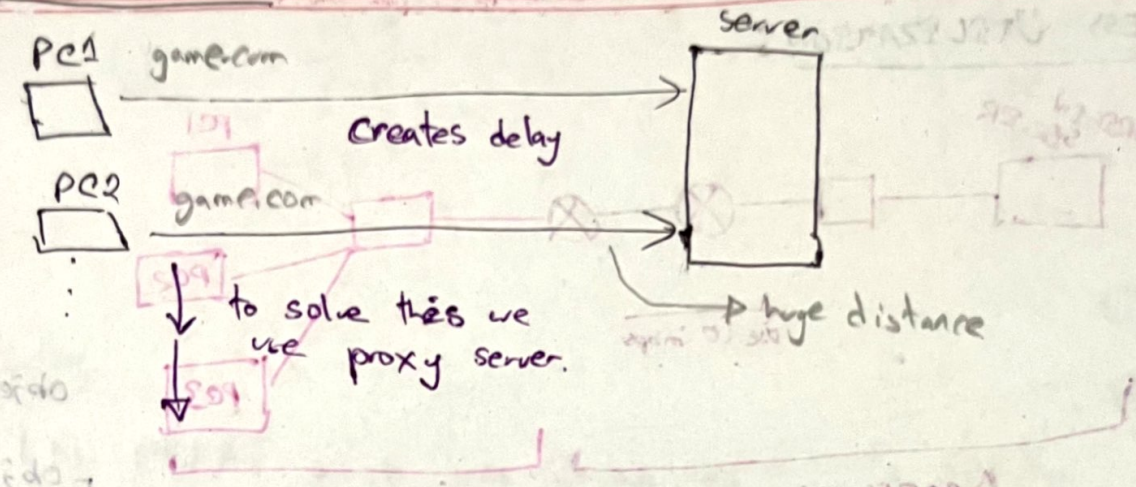
Chrome

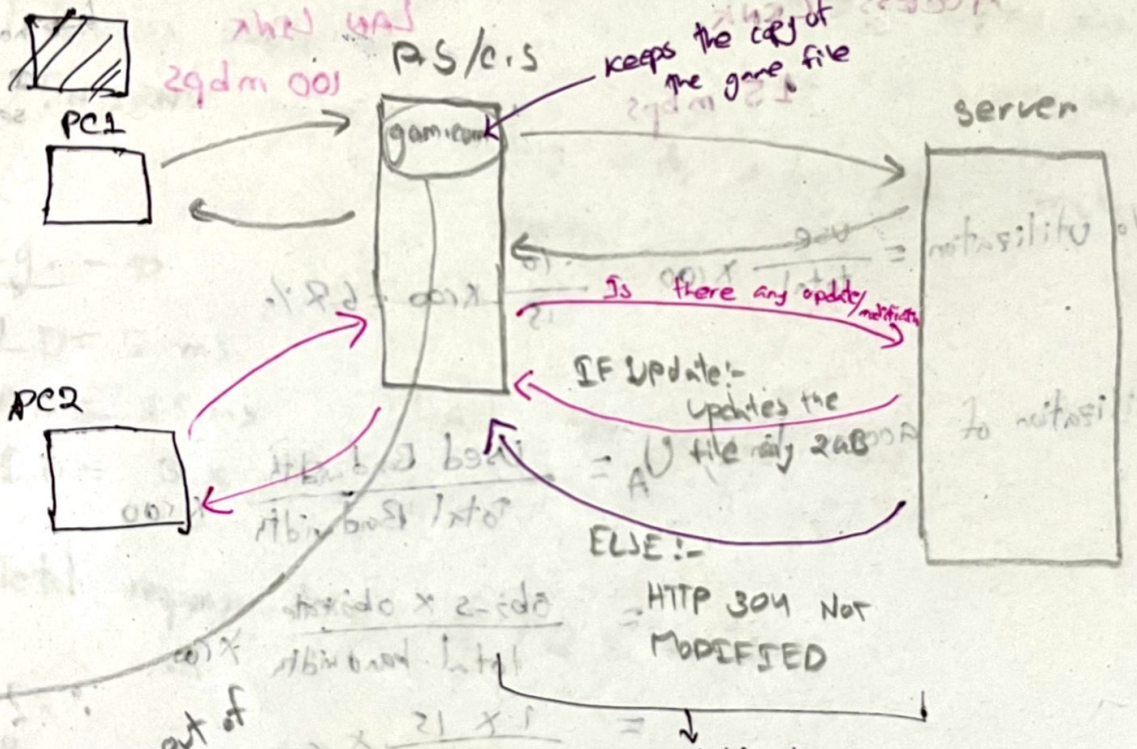
	ID
ebay	20
amazon	10
Avast	102

Firefox ← records

	ID
ebay	30
amazon	50

PROXY SERVER



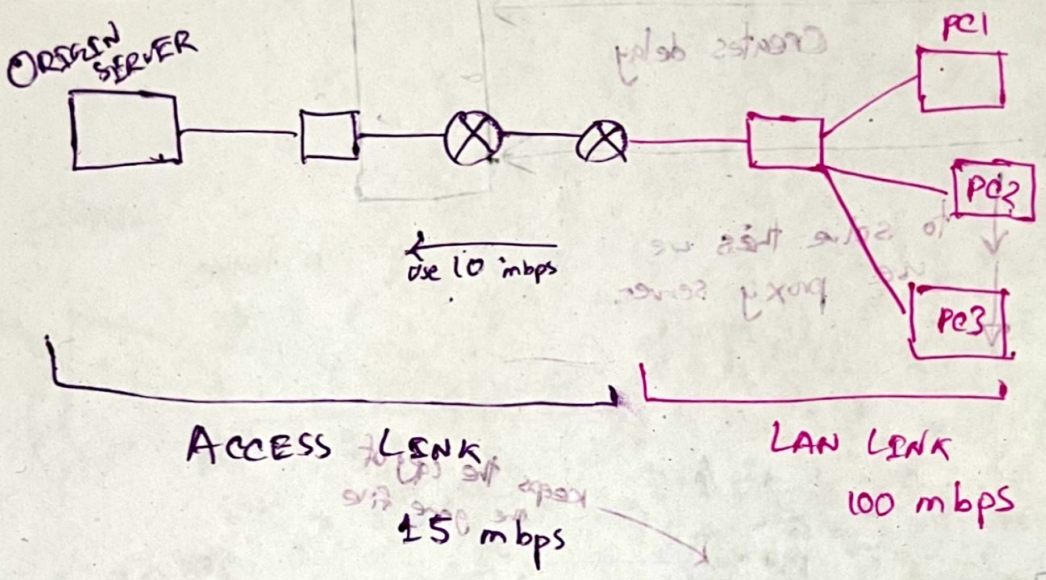


IF (If date) → STALE DATA

out of

(i) Track actions
(ii) speed control
(iii) Act as a filter

ACCESS UTILIZATION :



object size = 1 M
object rate = 15 /
how many object I can req every sec.

$$\% \text{ utilization} = \frac{\text{use}}{\text{total}} \times 100 = \frac{10}{15} \times 100 = 67\%$$

$$\%\text{ utilization of access link, } U_A = \frac{\text{Used Bandwidth}}{\text{Total Bandwidth}} \times 100$$

$$= \frac{\text{obj-s} \times \text{objrate}}{\text{total bandwidth}} \times 100$$

$$= \frac{1 \times 15}{15} \times 100$$

$$= 100\%$$

[good for pc1 bad for others]

solution

(i) Add proxy server ✓

(ii) cap speed ✗

(iii) speed ↑ ✗ (costly)

$$\% U_L = \frac{1 \times 15}{100} \times 100$$

$$= 15\%$$

Access link can be - but we have to select the less delay server

$U_A = 30\%$

$U_L = 9\%$

DELAY

(i) LAN DELAY → the time to bring the file from proxy server (processing in P.S)

(ii) ACCESS DELAY → " " " " " " origin server (processing in 0.5)

(iii) INTERNET " → routing inside every router creates delay.

e.g. →

LD = 5 ms

AD = 25 ms

ID = 3 S

LAN HIT RATE = 30%.

Total response time = 5 + 25 + 3000

" " " with CHR = 0.3 × 5 + (25+3000) × 0.7

= 2119

= 2.119 ← actual RTT

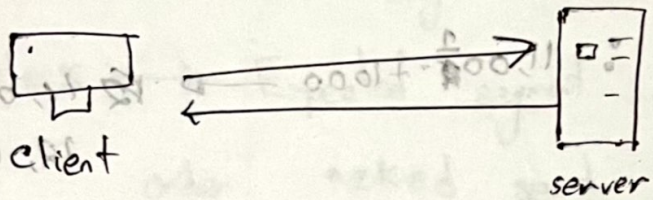
Average response time = LAN delay + Access delay + Internet delay

" " " = (hit ratio) x (LAN delay)

+ (Miss ratio) (Access delay + internet delay)

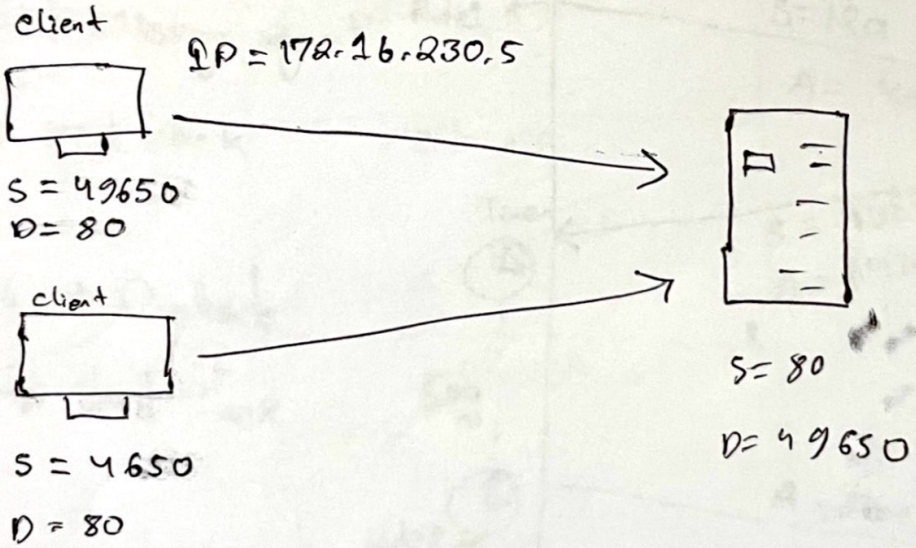
PORT NUMBERS

Well known	Registered		Private/ Dynamic	
1023	1024	49151	49152	65535



client
S = 49650
D = 80

server
S = 80
D = 49650



IP = 172.16.230.6

~~Socket~~.
Socket => IP : Port

In TCP

only Source Port & dest port is not enough. IP is also required.

Source dest.

BYTE Number :-

~~from 500~~

SEQUENCE NUMBER :-

F.B →

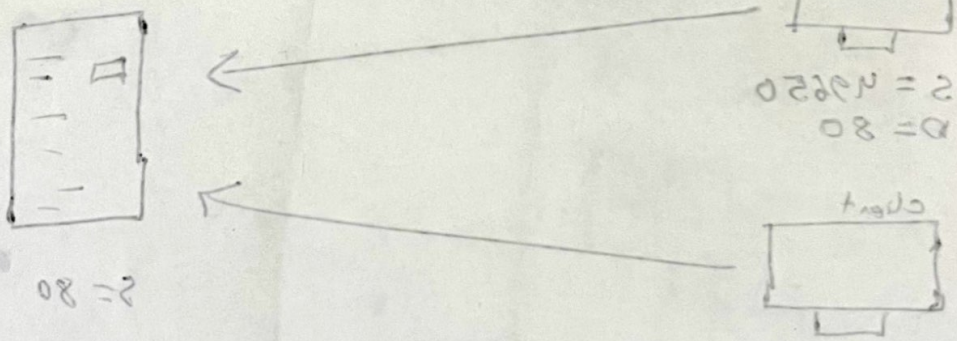
10, 001

Range

1st segment: 10,001 + 1000 → 10, 000 to 11, 000

2nd " : 11,001 + 1000 → 11, 001 to 12, 000

ACKNOWLEDGEMENT NUMBER :-



10/03/25

GENERATING ACKNOWLEDGEMENT RULES :-

Rule 1 → Data with Ack

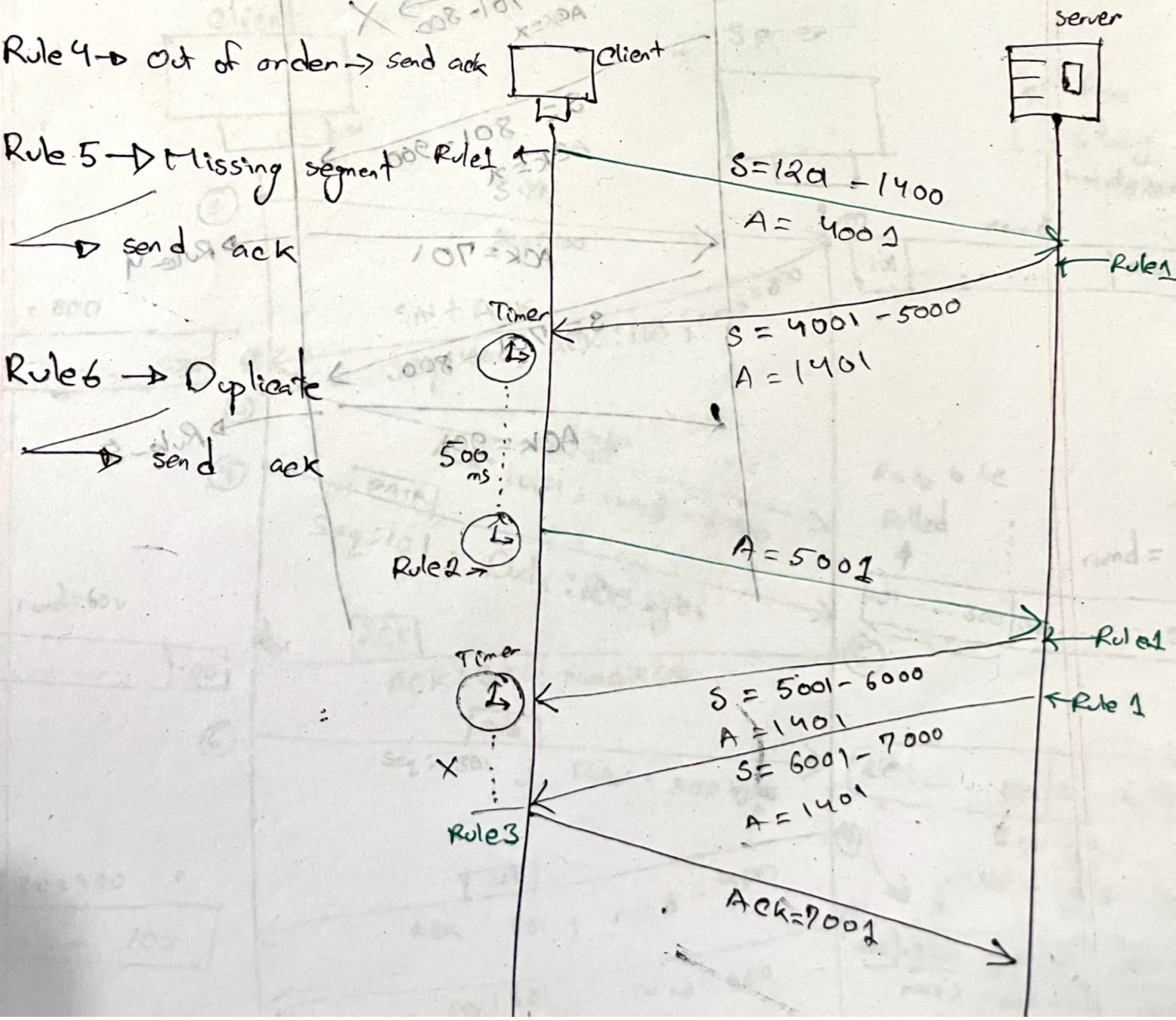
Rule 2 → One segment received and then wait / delay

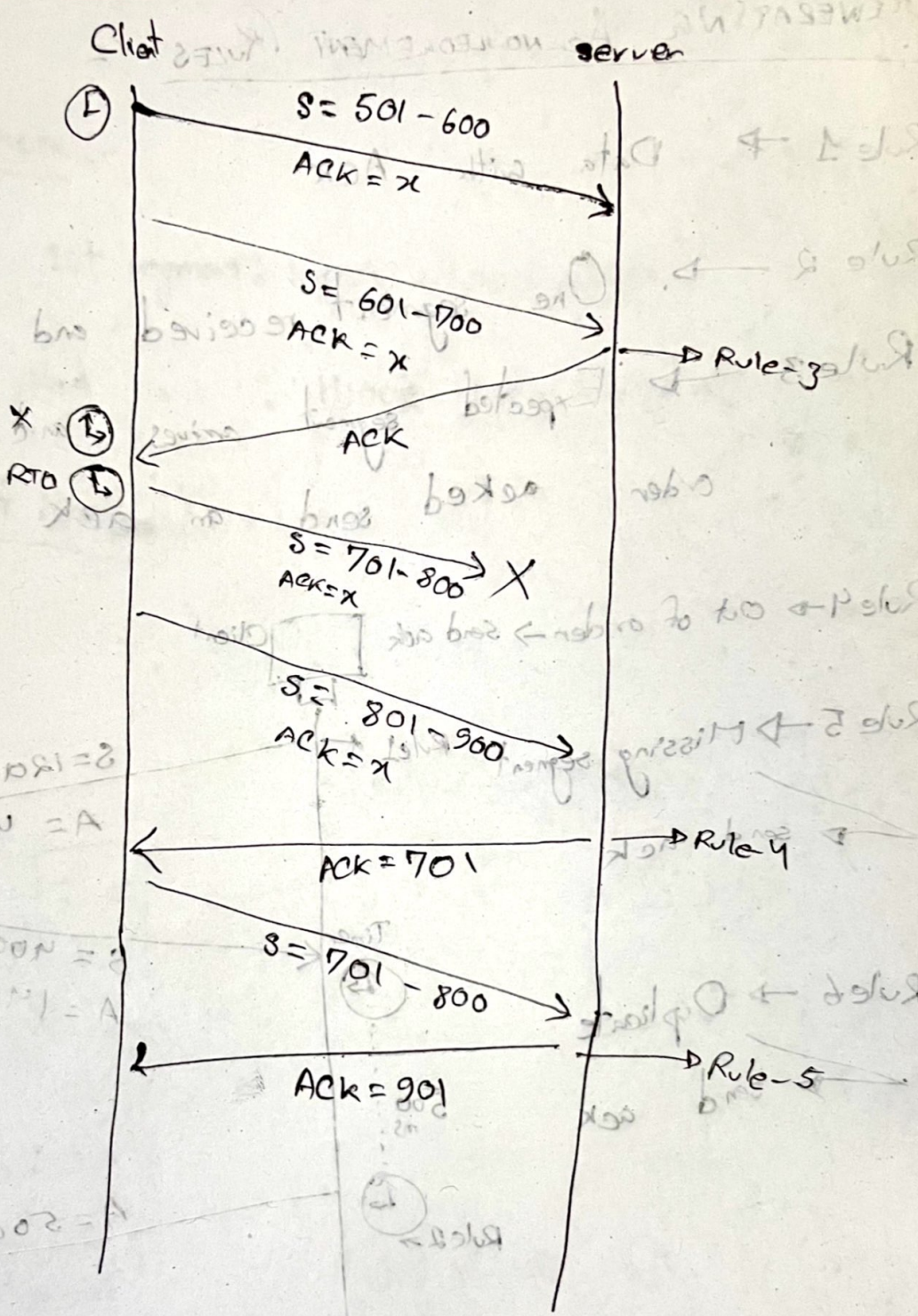
Rule 3 → Expected segment arrives and previous in-order acked send an ack

Rule 4 → Out of order → send ack

Rule 5 → Missing segment → send ack

Rule 6 → Duplicate → send ack




Client
server
S= 501 - 600
ACK = x
S= 601 - 700
ACK = x
Rule-3
X
RTO
ACK
S= 701 - 800
X
ACK=x
S= 801 - 900
ACK=x
Rule 4
ACK = 701
S= 701 - 800
Rule-5
ACk = 901

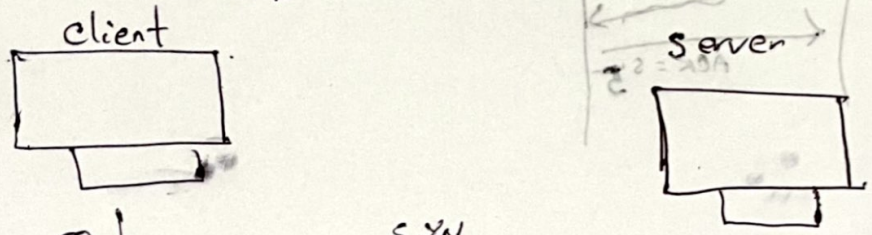
Flow Control Using Sliding Window

Window ⟶ size of device's buffer.

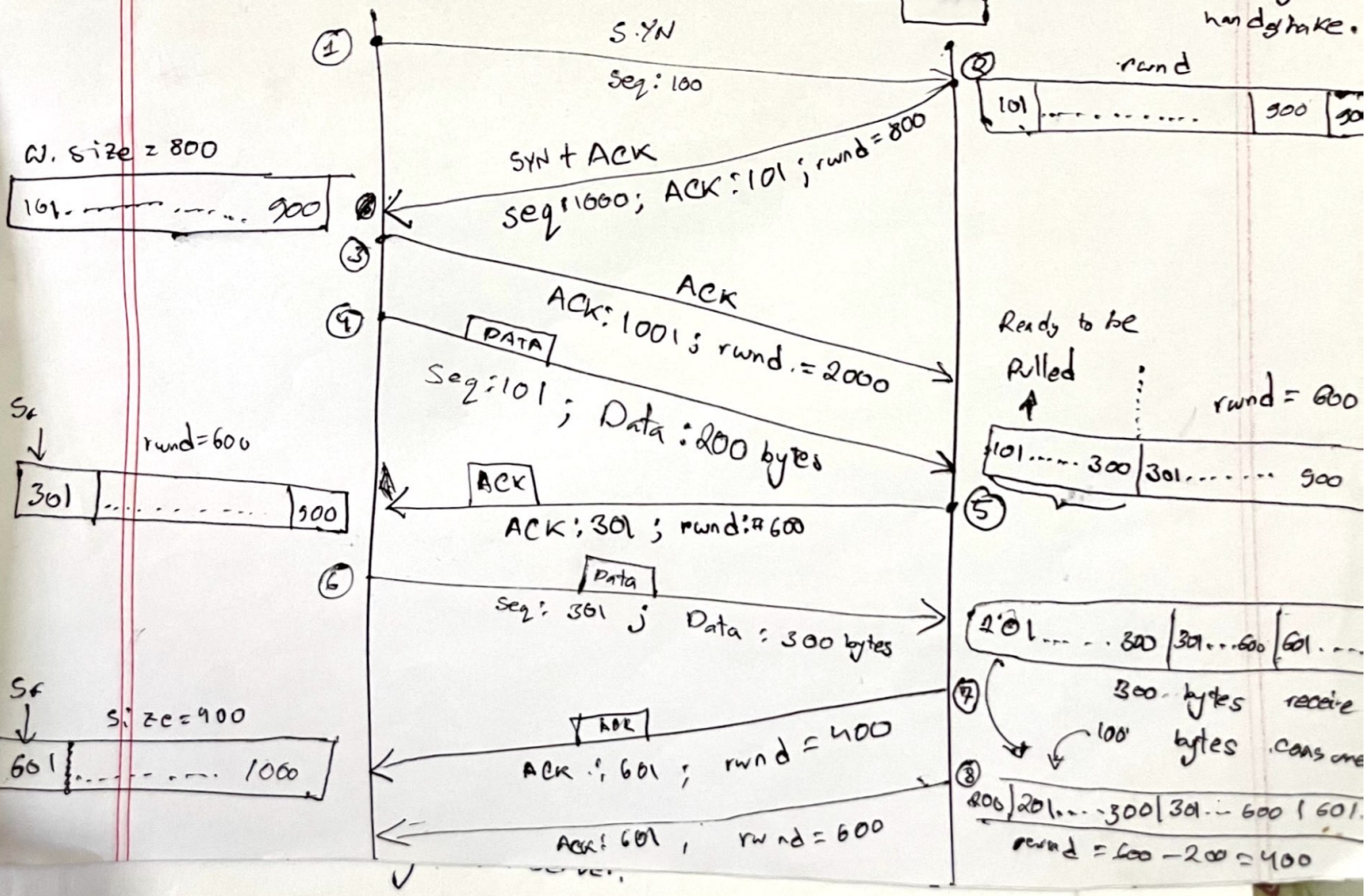
sliding window ⟶ (i) sender
(ii) receiver.

rwnd = buffer size − number of bytes to be pulled

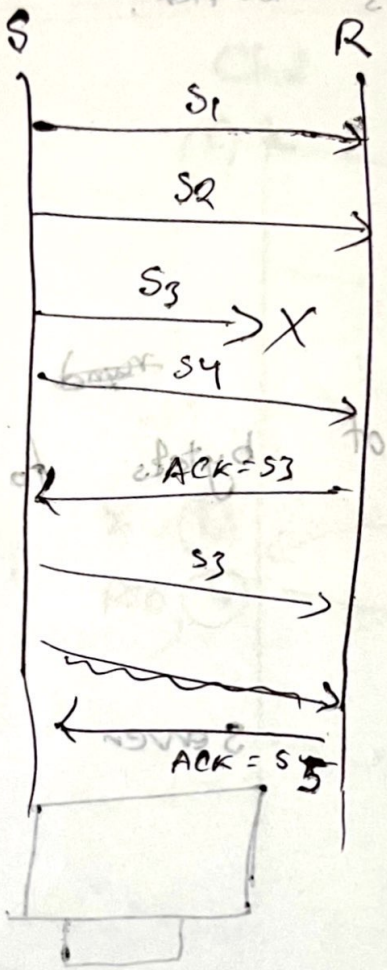
Flow Control Example



1st flow
↳ 3 way handshake.



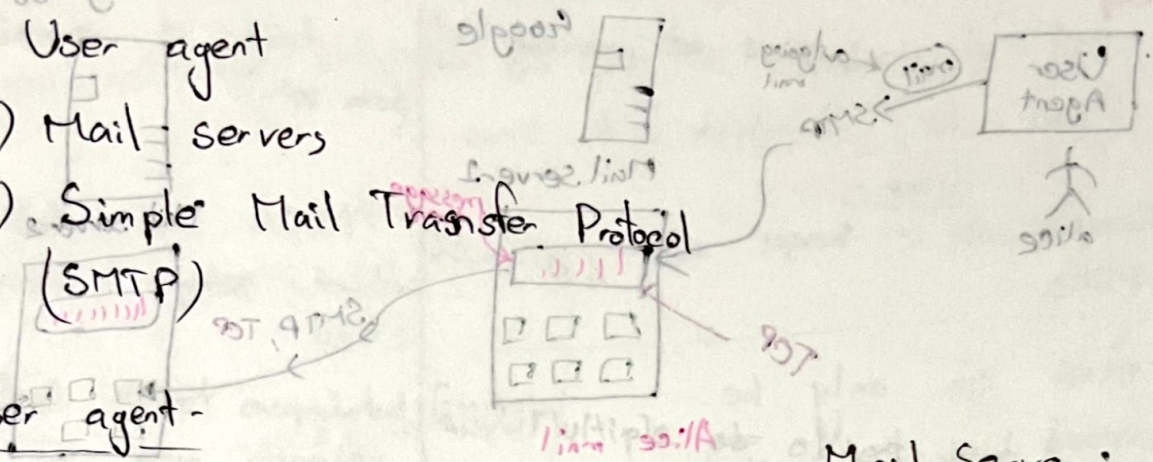
Go back N protocol:



APPLICATION LAYER: (Email & DNS)

MAJOR COMPONENTS :-

(i) User agent
(ii) Mail servers
(iii) Simple Mail Transfer Protocol (SMTP)



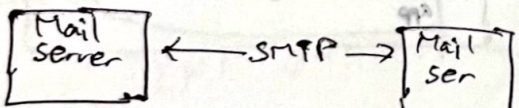
User agent:-

Software program that is used for composing, editing, reading, forwarding mail messages.

e.g. → Outlook, iPhone mail client, Web browser

SMTP Protocol:-

We sent emails using SMTP protocol.



From one mail server to another mail server the mails are transferring through SMTP.

Client :- sending mail server
Server :- ~~receive~~ receiving mail server.

Mail Server :-

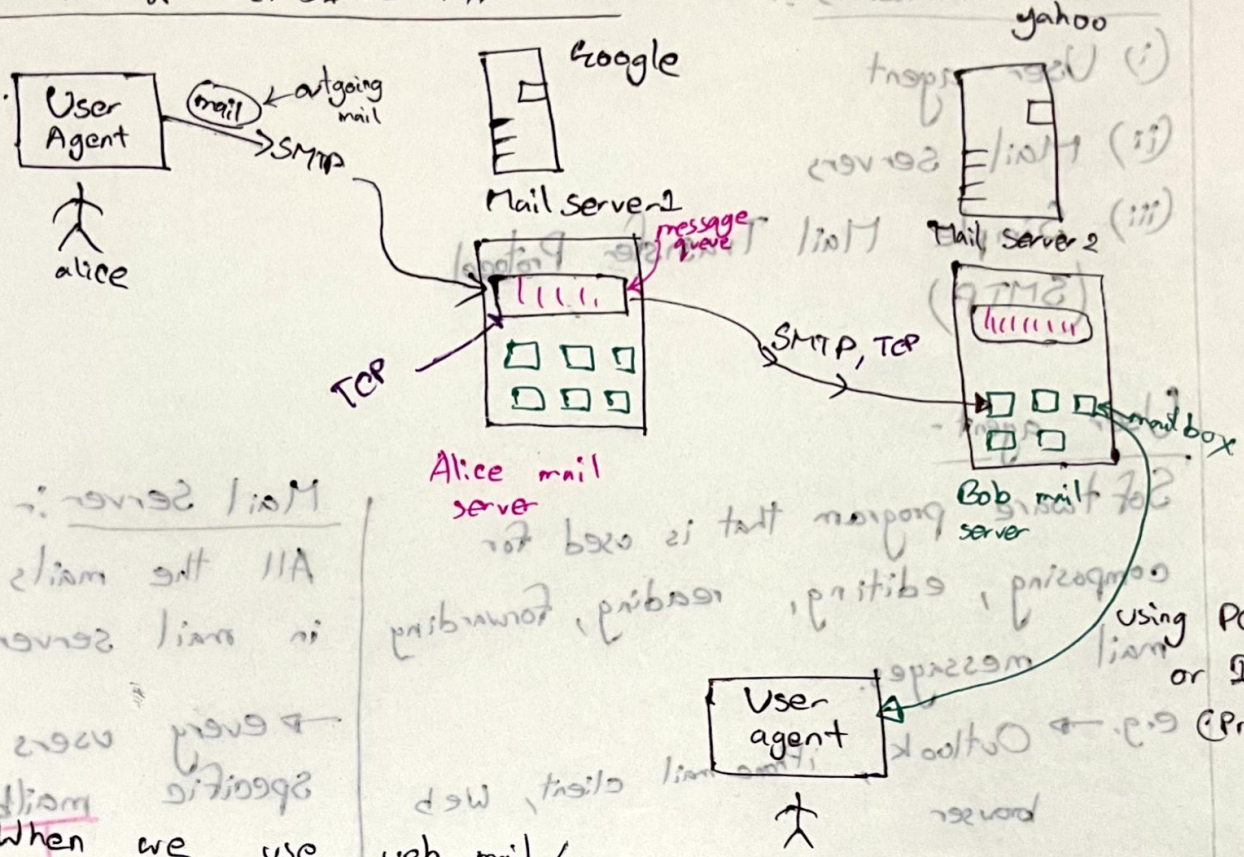
All the mails are stored in mail servers.

→ every users have their specific mailbox

→ It contains the incoming messages for user

→ The outgoing messages stays in message queue

TIMELINE of SENDING AN EMAIL:



→ When we use web mail/browser we send the email using https/HTTPS protocol. If we use software/mail app then it's use SMTP.

→ Server to server communication always uses SMTP.

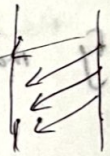
→ SMTP Protocol · three phases of transfer

(i) SMTP handshaking

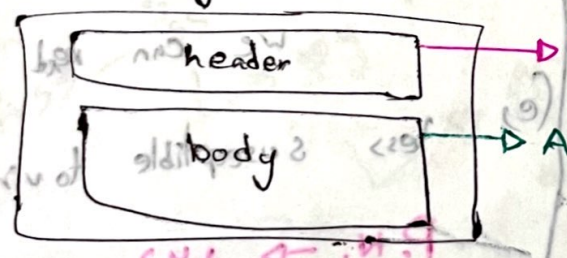
(ii) SMTP transfer of messages

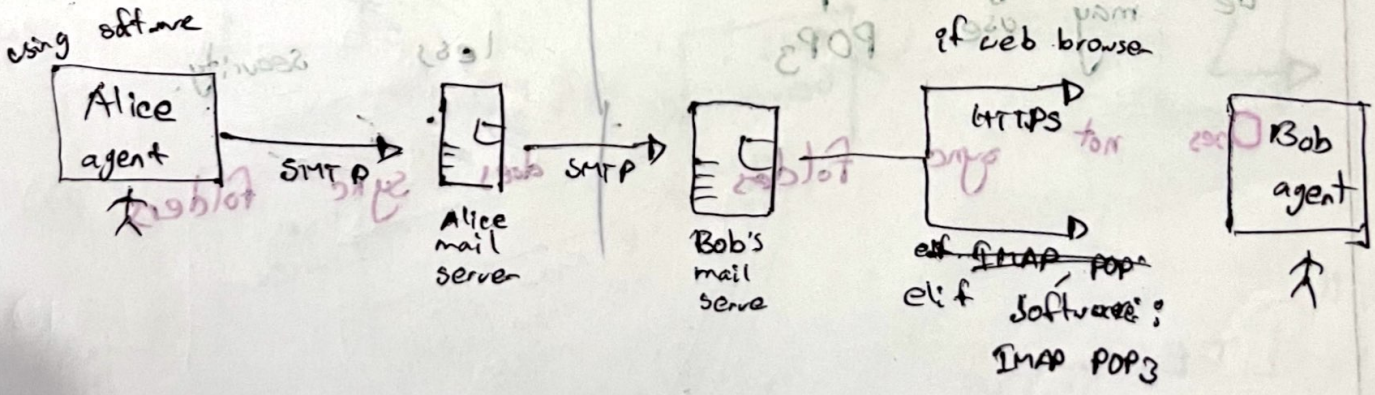
(iii) SMTP closure

TCP closure.

HTTP	SMTP
(i) Server to client (vice versa)	(i) server to server ★ SMTP (uses persistent connection)
(ii) Both have ASCII command / response interaction	(ii)
(iii) Each object encapsulated in its own response message	(iii) Multiple object sent in multiple part message.
(HTTP pull) 	(SMTP Push) 

Mail message format:





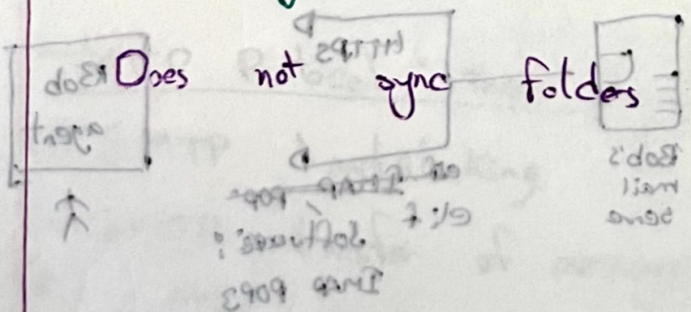
(i) POP3 :- ~~Mail~~

(a) Mail downloaded at PC and deleted from the server.
(can be download and keep in server too)

IMAP :-

(a) keeps all mail at the server

POP3	IMAP
(b) Mail can only be accessed using a single device.	(b) Mesgyes can be accessed on multiple devices.
(c) does not allow — create, delete, modify mail boxes	(c) Allows → delete, modify, update mail boxes
(d) Once downloaded — can only read it	(d) Before download — we can read the msg par
(e) vulnerable to virus	(e) less susceptible to virus
P.N → 110	P.N. → 143
For security purpose we may use POP3	less security.
Does not sync folders	does sync folders



#DNS → DNS operates as a distributed database that maps domain names to IP Addresses.

SERVICES →
- To translate user supplied hostnames to IP addresses. (DNS Cache)
- Host aliasing: hides the server's original name.
- Mail Server aliasing
- load distribution.

⇒ ebay → 101, 102, 103, 104 → Rand Robin → distributes the load to 5 server

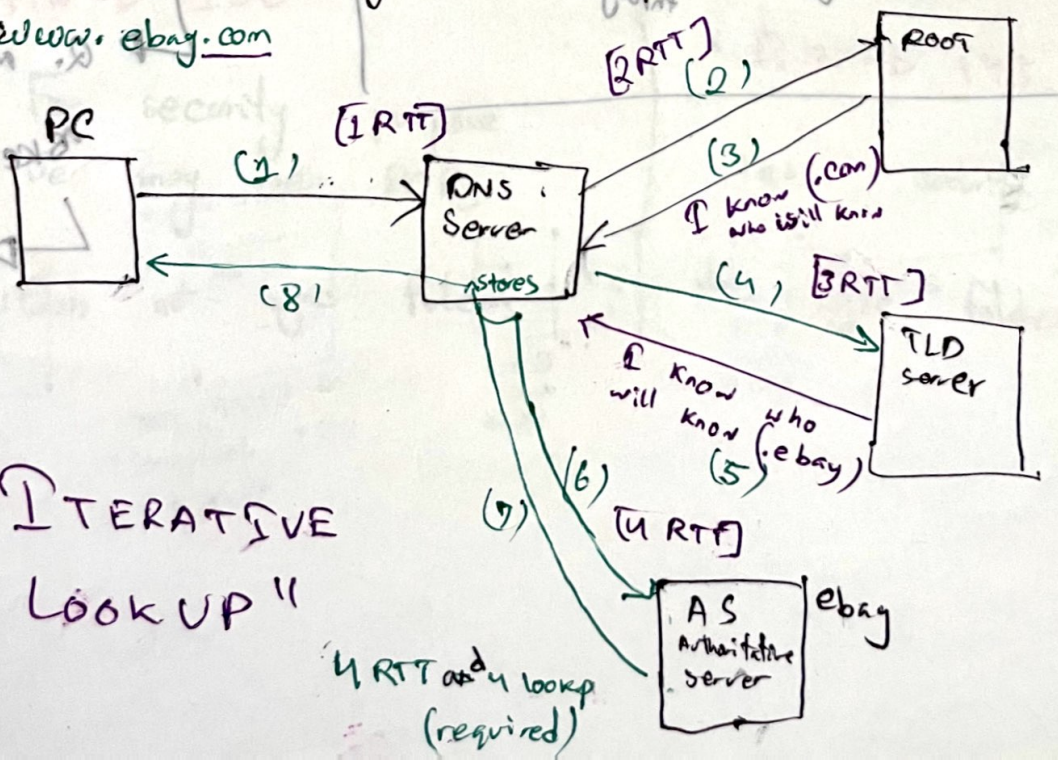
★ It is employed by other application layer protocols
→ HTTP, SMTP, FTP

★ How a request goes from PC?
www.ebay.com

DNS Server to other server = lookup

Top level domain
Root → points to
TLD: .com, .org, .net, .ac

AS:-

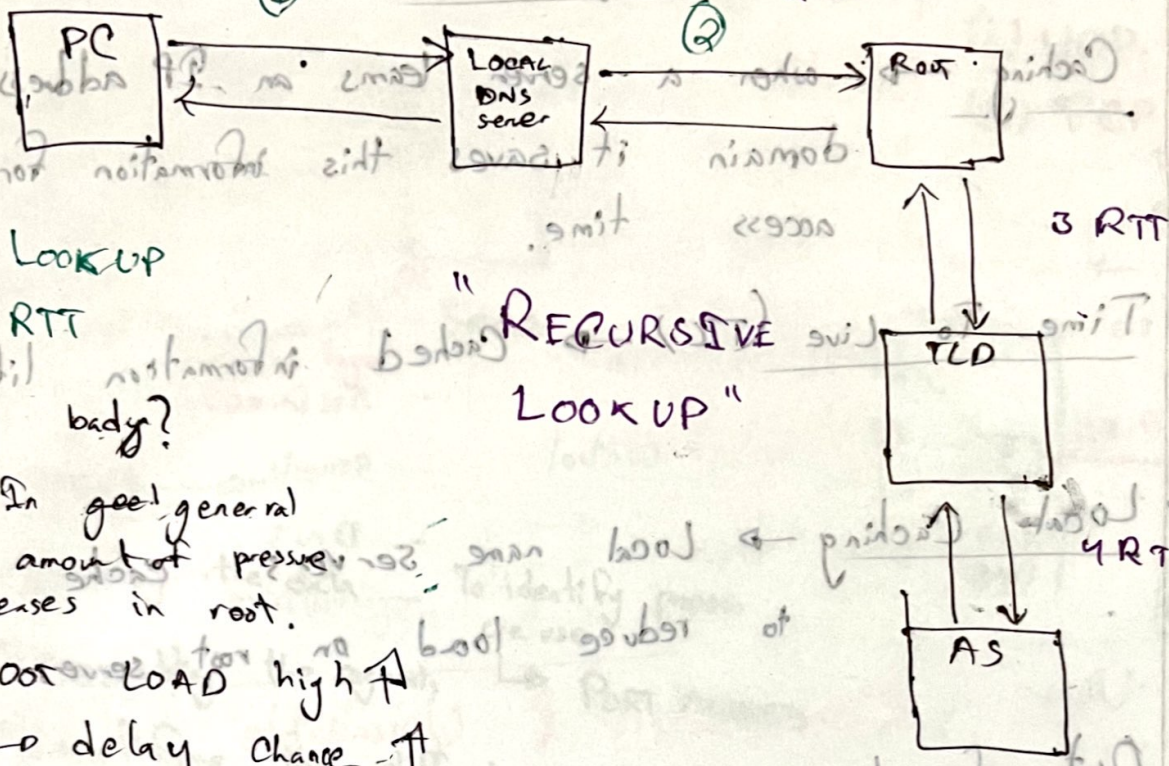


"ITERATIVE LOOKUP"

4 RTT and 4 lookup (required)

★ There might be multiple level of TLD server
brac.u.ac.bd

→ How many RTT required?
→ Is there Local DNS server?



2 Lookup
4 RTT

"RECURSIVE LOOKUP"

→ Why bady?
→ In general general the amount of pressure increases in root.
→ Root LOAD high ↑
↳ delay chance ↑

DNS RECORDS :-

DNS request format → (name, value, type, ttl)

type=A → provides HN (name)
returns IP addr (value)
value → IP address

type = NS → Name ⇒ domain
returns website IP addresses location
(google.com, dns.google.com, NS) value → DNS server IN Fo

type = CNAME
↳ provides alias name
returns original name
value → the name webpages diff name

type = MX
gmail.com → type=A → web
gmail.com → type=MX → mail
value → mail server info

~~ATTACKING DNS~~

DNS → Caching / Updating Records

Caching → when a server learns an IP address for a domain it saves this information for faster access time.

Time To Live (TTL) → Cached information lifespan.

Local Caching → Local name servers cache TLD records to reduce load on root servers

Out of date Cache → Cached entries might be outdated if a domain's IP address changes.

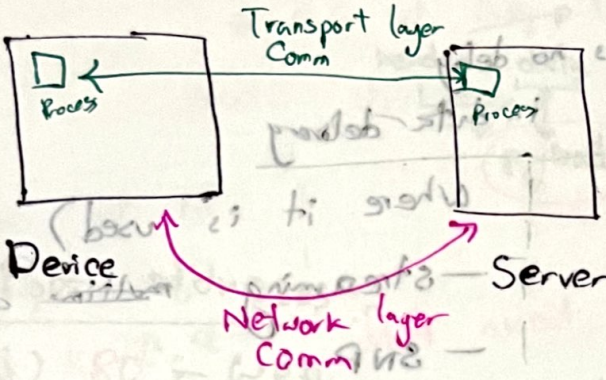
problem overcome by

Updating standard :- Updates the DNS records to (IETF standard) update the updated domain's IP address.

TRANSPORT LAYER . PDU → Segments

PORT ADDRESS

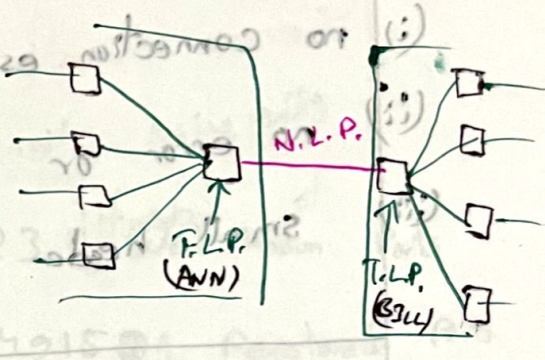
→ Process to Process message delivery.



TRANSPORT LAYER PROTOCOLS:

(i) UDP User Datagram Protocol

(ii) TCP Transmission Control Protocol



FUNCTIONS:-

(i) Segmenting the data

(ii) Reassembling the segments

(iii) Identifying the different applications

(iv) Multiplexing.

To identify process we use → PORT ADDRESS

TCP & UDP

For Reliable transmission → TCP

" Fast " → UDP

Reliability Functions:-

(i) Initiation & termination

(ii) Perform flow control

(iii) Enable Error recovery.

→ TCP (only)

UDP → TRANSPORT LAYER PROTOCOL

(i) Best Effort service.

(ii) Used by application that requires no delay in data delivery

	where it is used)
	— streaming apps
	— SNP
	sometimes —
	— DNS
	— HTTP/3

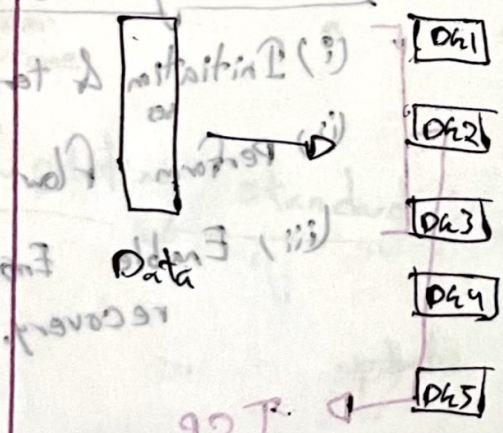
Why it is FAST?

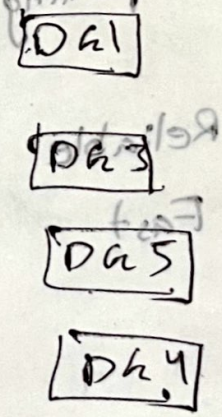
(i) no connection esstablishment

(ii) no error or flow control

(iii) small header size

UDP → Connectionless & Unrealiable





DG → datagram.

→ does not reassembles
→ out of order
→ lost Datagrams).

UDP header →

Source Port No. | Dest. Port No.

Total length → header + data → payload

Checksum → small error checking → optional

PORT NUMBERS :-

(i) 80 – Web

(ii) 25 – SMTP

(iii) 0 - 65535 – (Range)
- 0 - 1023 → Well Known Ports
- 1024 - 49151 → Registered Ports
- 49152 - 65535 → Private / Dynamic Ports

(iv) 20 – FTP data

Well Known ports → Reserved for common services
Preassigned by (ICANN

Registered " → Companies and users register with ICANN / For applications that communicate using any 1 T.L.P

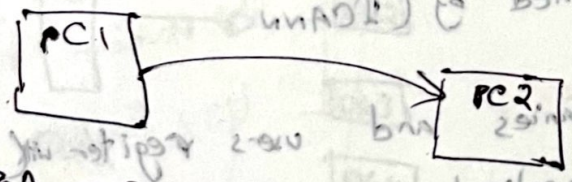
Dynamic " → Assigned to a user application at
time.

IMP Notes on P.No.'s

The client sets the dest port to 80 & uses a dynamic port as its source (Always dynamic on client side)

Servers must use well known P.N. otherwise clients wont be able to identify servers process.

FOR PC to PC :



P.A. → Dynamic P.A. → registered

source

dest

If the server uses Dynamic port ?

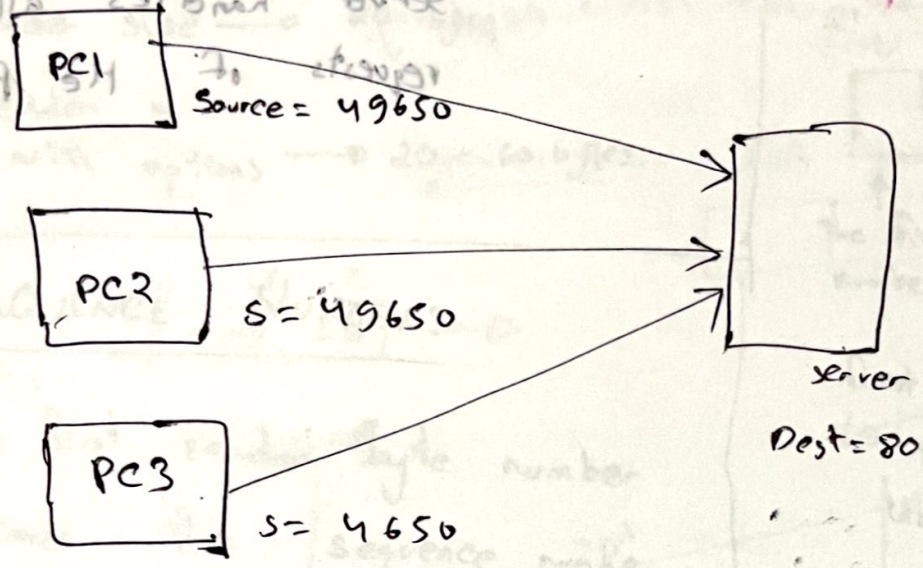
→ If it is dynamic then how I set or know the dynamic port address ?

[When ~~some~~ everyone trying to access ~~some~~ something we try to keep that at a fixed location

~~Servers~~ ~~on~~

[In this case we will use registered P. A.]

What if 3 PC generates the same Dynamic Port Address?

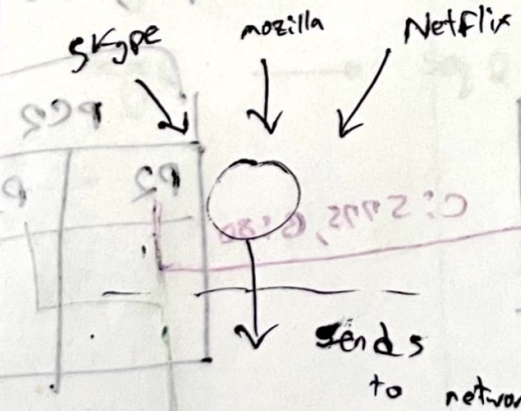


Then how will it identify the PC differently? →

using → IP Address

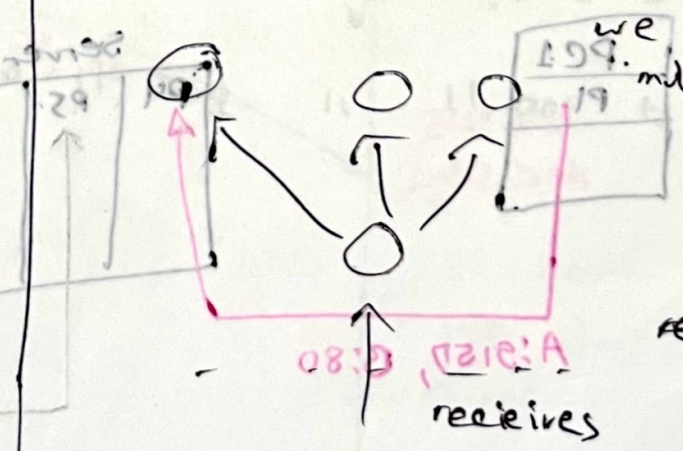
keep them seperate by using → Socket (IP Address, Port no)

MULTIPLEXING



process of combining multiple application layer data streams.

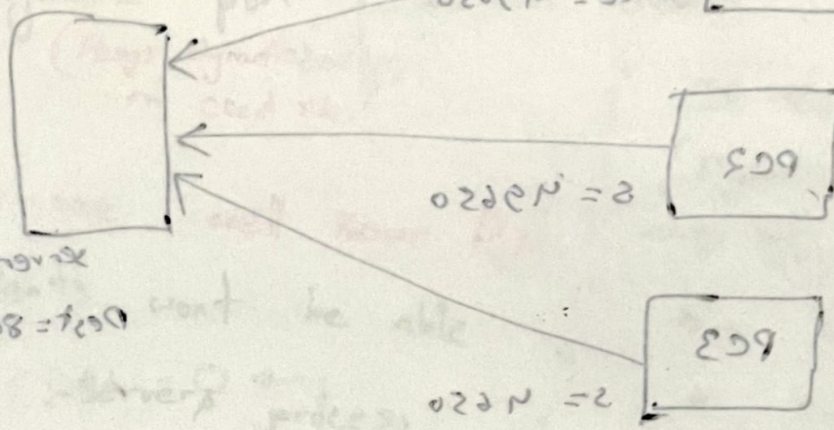
DEMULTIPLEXING



receive data to the correct application process

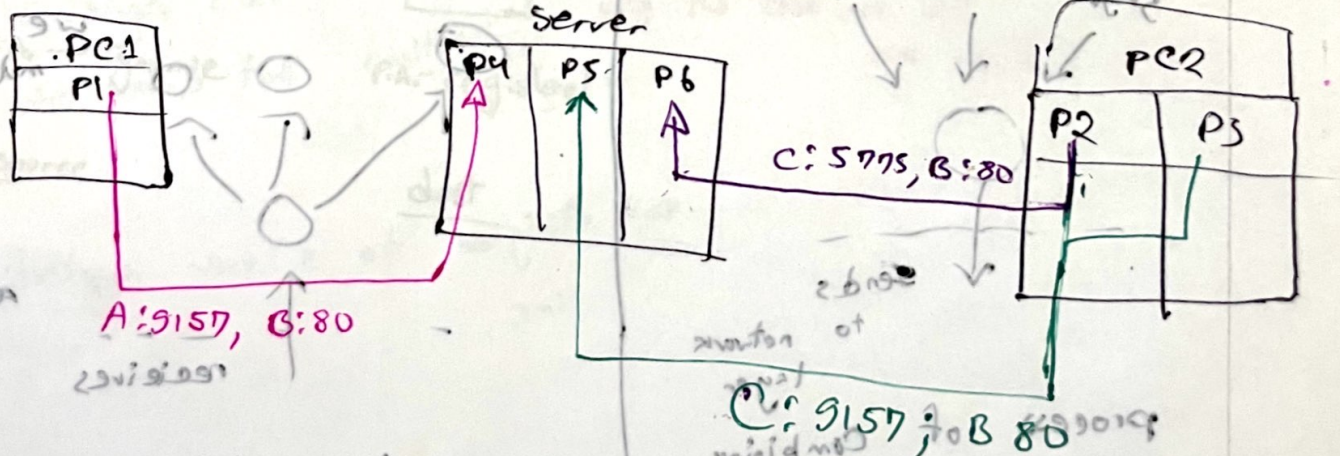
UDP Connectionless Demultiplexing: (Only 1 socket of the server handles all the requests of PC)

dest IP, dest port



Connection Demultiplexing:

Creates a socket using tuple (source IP, source port, dest IP, dest port)



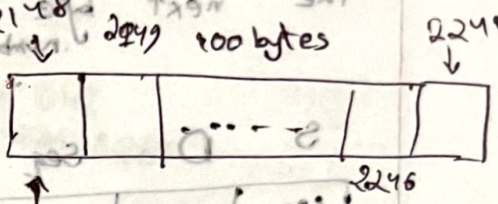
#TCP

TCP segment header:-

header size → 20 bytes

header " with options → 20 to 60 bytes.

BYTE NUMBER



The first starts with an arbitrary number $(0 - 2^{32} - 1)$

SEQUENCE NUMBER:-

The first random byte number becomes the sequence number

or [The seq no. of 1st segment is ISN]

first byte → 1067

size → 3000 byte

last byte → 4066

ex →

Total file → 5000 bytes

first byte → 10,001

Each segment → 1000 bytes Seq no. ?

seg 1 → seq 1 → 10,001 Range 10,001 to 11,000

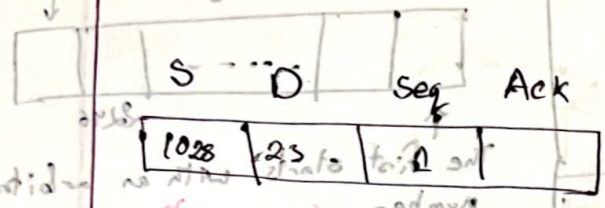
seg 2 → seq 2 → 11,001 " 11,001 to 12,000

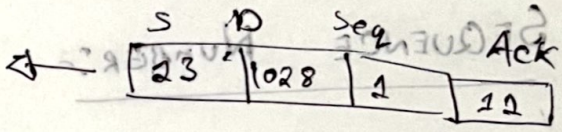
⋮

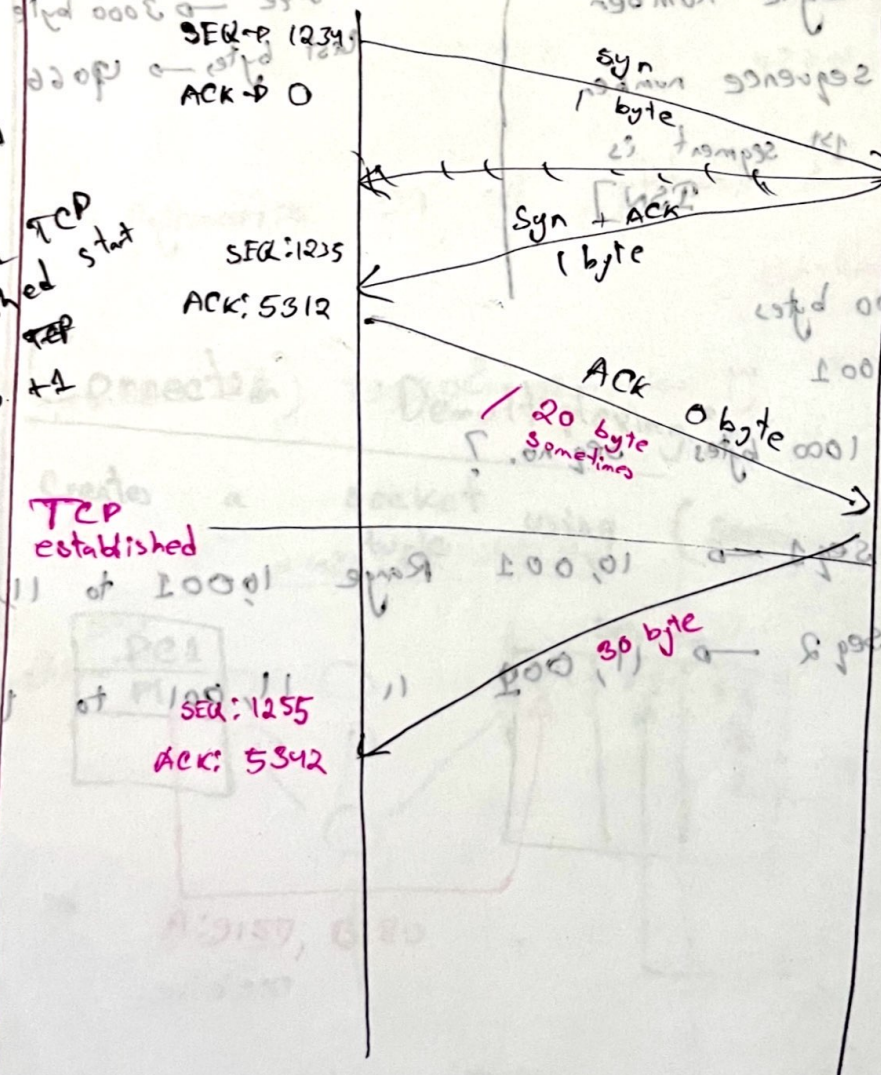
TCP

ACKNOWLEDGMENT No. :-

The next byte number which receiver expecting.

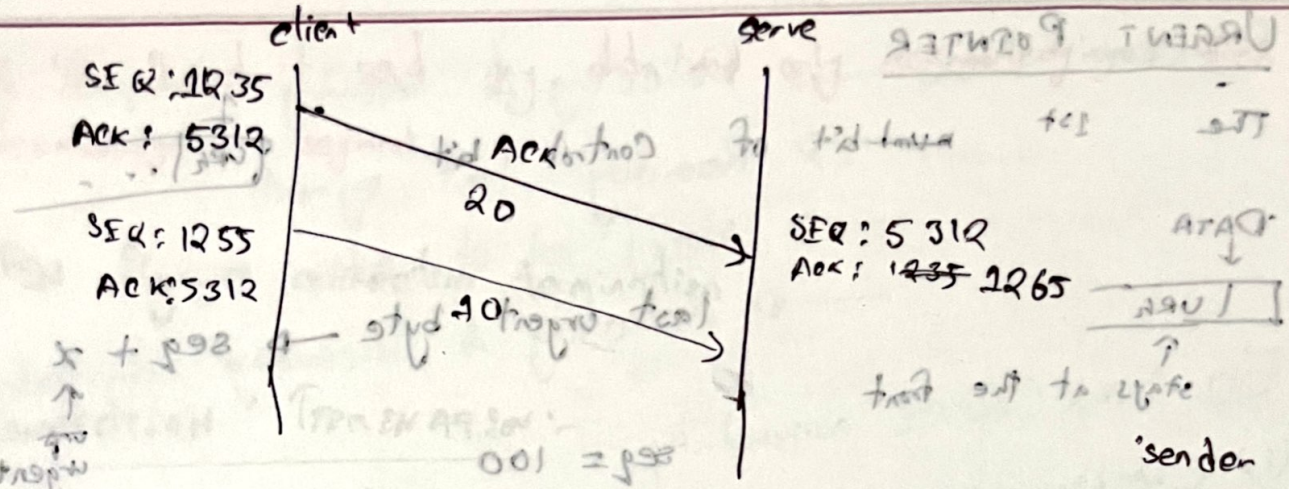






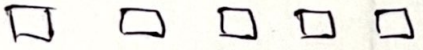
If ISN given and said TCP established with seq. no. +1 that

SEQ: 5312
ACK: 1255



HEADER LENGTH:

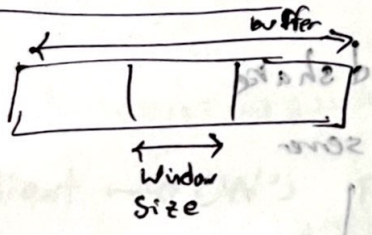
Indicates the no. of 4 byte words



→ 20 bytes → HLEN = 5 bits

60 " → HLEN = 15 bits

WINDOW SIZE:-



The maximum amount of data which I can take before sending an ACK.

Receiver sets the WS

CHECKSUM:-

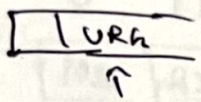
To check if the segment got corrupted while segment was travelling to reach the dest.

uses → TCP Header
→ TCP Body
→ PSEUDO IP Header

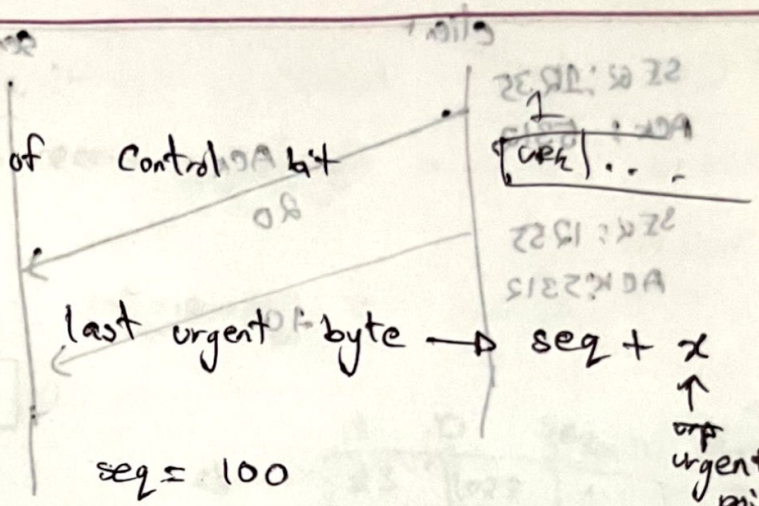
URGENT POINTER

The 1st bit of Control bit

DATA



stays at the front



seq = 100

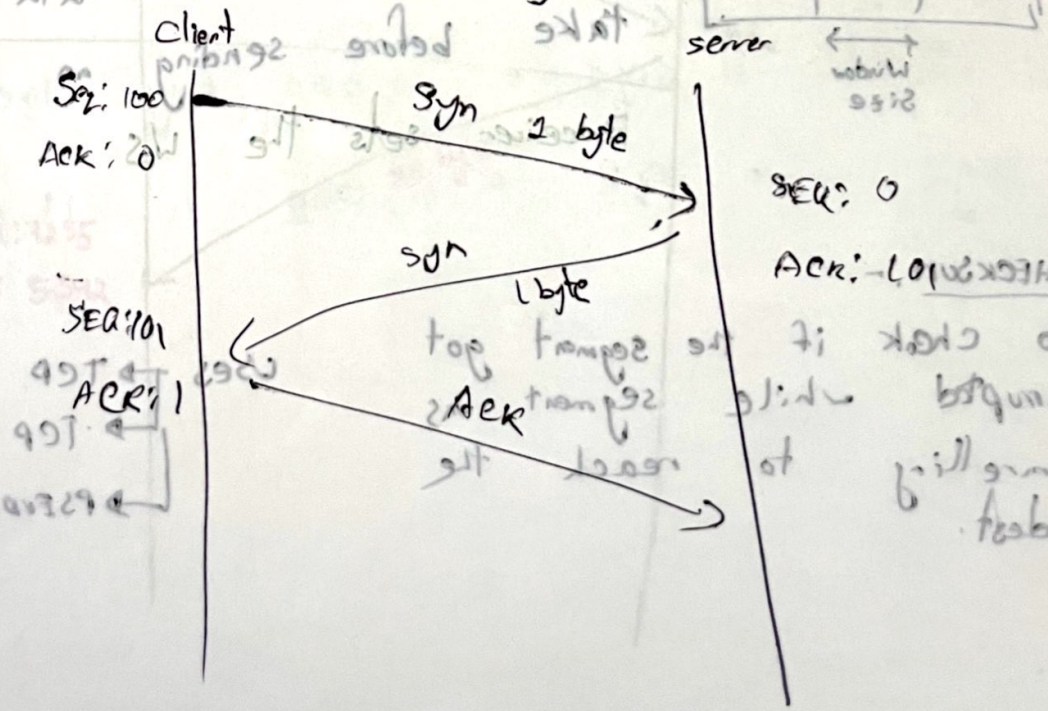
x = 200

last urgent byte = 100 + 200 = 300

Total urgent data = 300 - 100 + 1 = 201

OPTIONS :

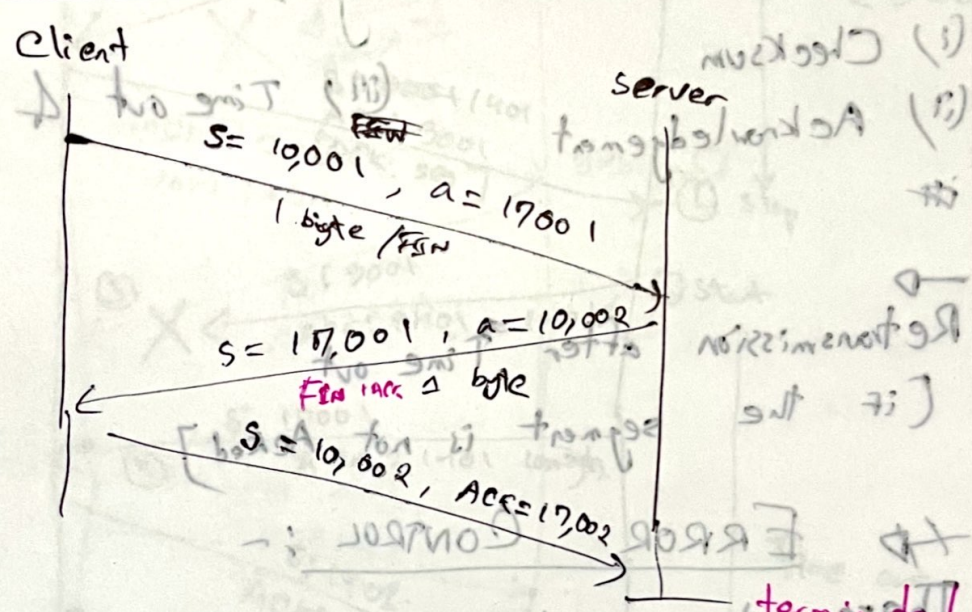
Connection Establishment → 3 way handshake



★ If I don't send any data and only acknowledging then its acknowledgement segment.

FIN flag → connection termination

CONNECTION TERMINATION :-



Client → ON's FIN flag

server → doesn't on FIN flag

↓

Client - server both wants to close

as server didn't finished to send all / couldn't data.

So, when client turns ON the FIN flag and server doesn't turns ON the FIN immediatly.

the its it is HALF CLOSE.

Ack segments don't consume seq no's and are not acknowledged

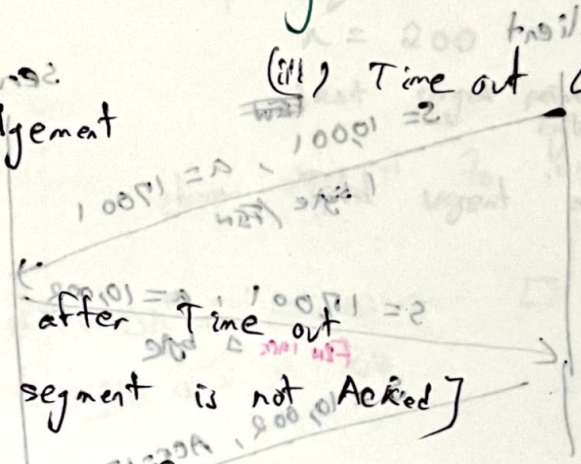
RELIABILITY IN TCP

PROBLEMS/CASES → Detecting & resending corrupted segments.
→ Resending lost segments
→ storing out of order segments.
→ detecting & discarding duplicated segments.

SO ERROR CONTROL achieved by →

(i) Checksum
(ii) Acknowledgement
(iii) Time out & retransmission.

→ Retransmission after Time out
(if the segment is not Acked)



→ ERROR CONTROL :-

There two Criteria to send ACKNOWLEDGEMENT :-

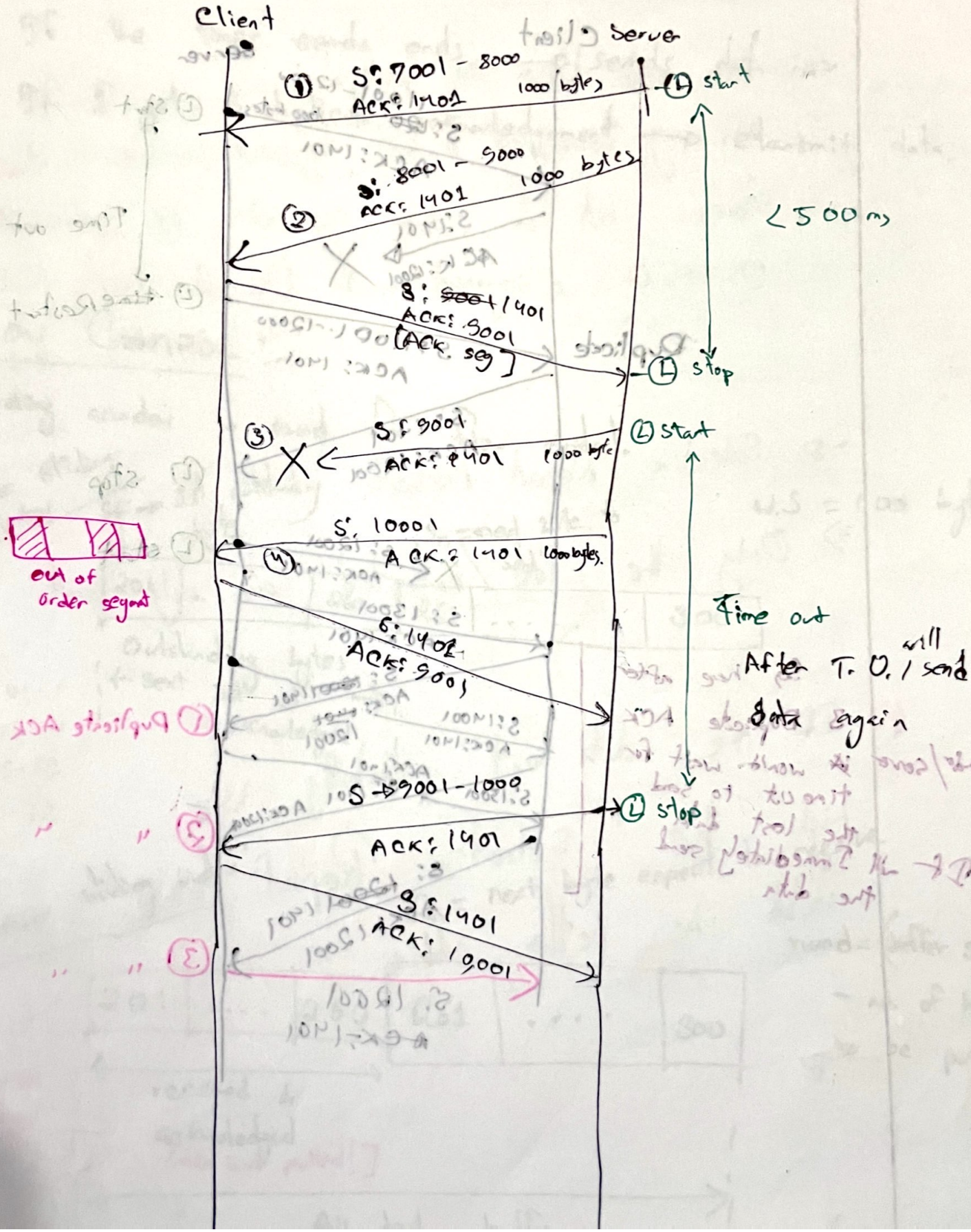
(i) If the timer ends → sends → Ack
(ii) If I get 2 segment consecutively then sends → ACK
(iii) When I get out of order segment. sends → ACK
(iv) When I "missing/expected" (when I get something later than the previous)
(v) If I get duplicate segment sends → ACK

exception case to send ack when the timer does not end

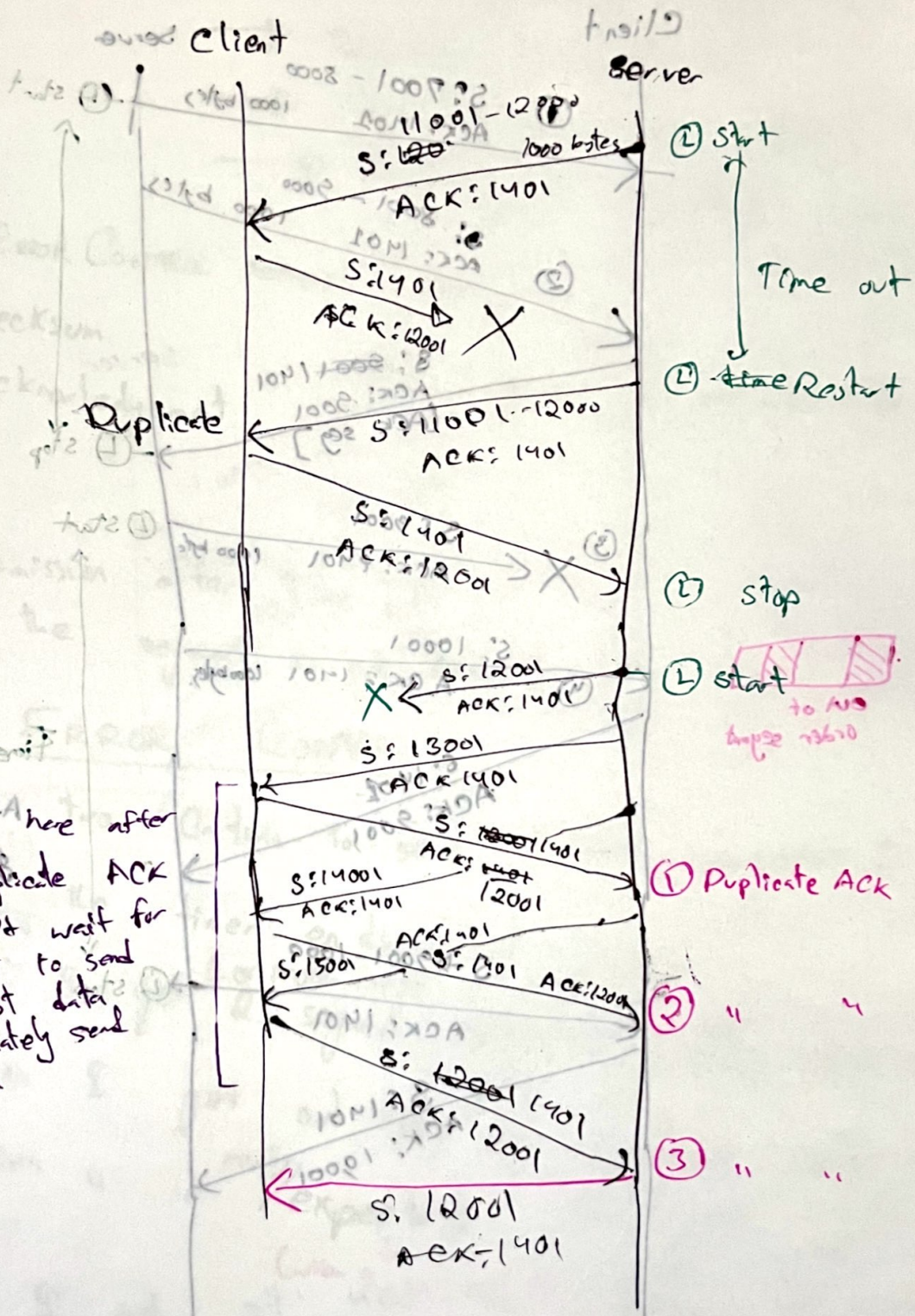
or before the timer ends

→ Detecting and ~~and~~ discarding duplicated segments

Segment lost transmission :-



Lost Ack



3 Duplicate ACK here after

3 Duplicate ACK would wait for timeout to send the lost data

If will Immediately send the data

Sf, Sn

Go back N protocol & selective repeat protocol are same.

Retransmission Rule :

(i) If the timer ~~sends~~ ends → sends data

(ii) If I get 3 duplicate acknowledgement → retransmit data.

exception to send for retransmission before timer ends

FLOW CONTROL :

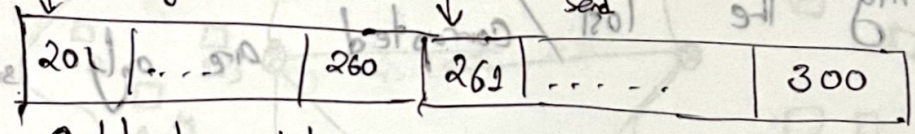
sliding window is used for flow control.

Sender sliding window

Sf → 1st outstanding byte.

Sn = next byte to send

w.S = 100 byte

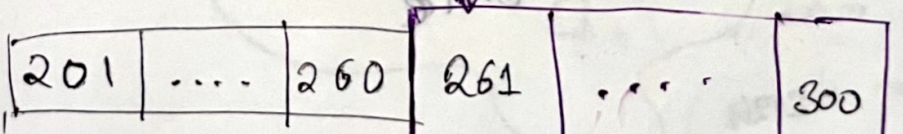


Outstanding bytes ← sent but not acknowledged →

Receiver sliding window :

Rn = next byte expect to receive

rwnd = (buffer size - no. of bytes to be pulled)



received & acknowledged [data isn't pulled!]

Allocated buffer.

If the pulled isn't even if its acknowledged then the buffer doesn't get empty

DNS $\xrightarrow{\text{fetch}}$ IP $\longrightarrow$ TCP $\xrightarrow{\text{connection established}}$

Different Sliding window Protocols for [out of order data]

→ Go Back N protocol.

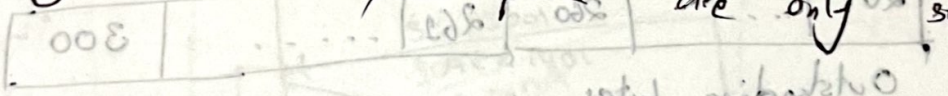
⇒ If any corrupted or lost segment arrives it doesn't stores out of order segment it discard those O. OF. O of segment → Do not Keep track of out of order segments.

1-20
11-20 ×
21-30
31-40

buffer = 30 empty

→ Selective Repeat Protocol.

⇒ Only the lost / corrupted are only sent.

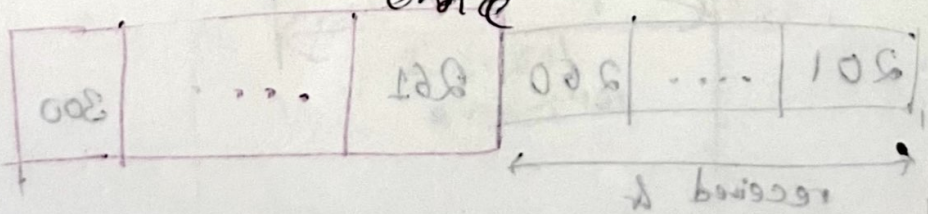


1-10
11-20
21-30
31-40

buffer = 10 empty

If the data lost probability is high the Selective Repeat Protocol would be better choice

→



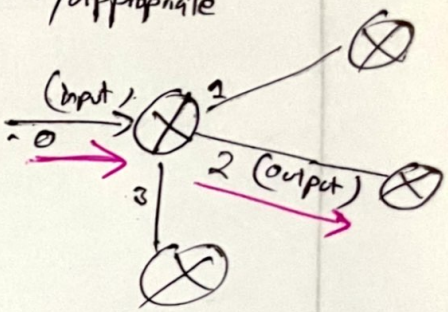
NETWORK LAYER

IPV4 FUNCTIONS

Functions of Network layer → 2 types

(i) Forwarding :-

→ Transfer packets from router's input to specific router's output /appropriate



(ii) Routing :- [Helps to take the decision which path (from-inp to-out) a certain packet should take]

→ Determine the route taken by packets from saree to destination.

How? → by taking using a "Routing Table"

Two types of network based on Packet switching :-

☆ Datagram Network – connectionless service of Network layer ex- (Internet)

☆ VC Network – connection service of network layer.

DATAGRAM NETWORKS :-

There is no fixed path for the transfer of different packets.

☆ no end to end connections created.

VIRTUAL CIRCUITS

→ A fixed frequency/path is set between end to end devices.

ex- ATM, Frame Relay.

Functions of Router:

(i) Run routing algorithms/protocol.

(ii) Forwarding packets from input to output.

(TIMELINE)

Can we use?

TCP → Transport

Datagram → Network

Yes it is possible

IP Protocol:- sends a packet and doesn't track if it arrived at the destination.

Network layer well known protocol →

☆ Internet protocol

↳ Best effort protocol

he uses helping protocol **ICMP** → (error reporting) if the message does not arrives.

Why didn't we used ~~ICMP~~ error reporting in IP?

→ we want the data for fast transmission. → that's why error reporting hasn't been incorporated.

Fragmentation math → Query offset calc

IPv4 Datagram Format

Version
↓
V4 or V6

Header length
size of header (4-7 bits)

Type of Service
provides precedence to the prefered service [multimedia]

Total length
Header + data length

Header checksum
- taking header & data
- making sure IP protocol and everything working
- If the address working
- check any error in header

Port Protocol
TCP or UDP

S A
source IP address

D A
destination IP add

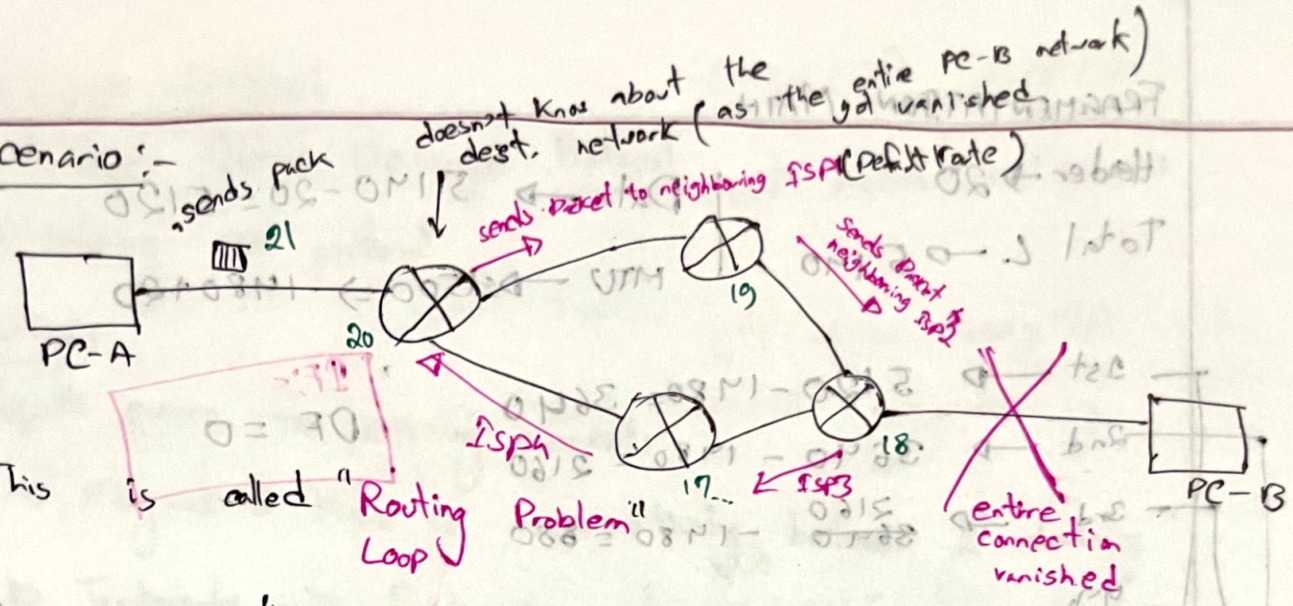
TTL
sets a time limit. (max no. of hops)

Flag's
MF
DF

Identification → every segment has byte number in transport layer
every packet has a number
↳ helps to reassemble the fragments

Offset
13 bits

Scenario :-



This is called "Routing Loop" Problem"

To overcome this problem we use TTL.

IP Fragmentation & Reassembly

To divides data into multiple parts

IP header bits used to identify, fragments order.

Packet / Datagram → Data + header

Data → data

Fragment offset = Initial byte / 8

example → next page

Why we use fragmentation?

Router stays connected with different links.
- different links has different capacity & BW
  - so it has different MTU (Max transfer Unit)

As a result, if 4000 byte data packet is sent and the MTU is 1500 bytes the packet can't get through the link as it crosses the MTU.

→ To overcome this we use fragmentation

FRAGMENTATION MATH:

Header → 20

Total L → 5140

Data → 5140 - 20 = 5120

MTU → 1500 ⇒ 1480 + 20

DF:-
DF = 0

1st → 5120 - 1480 = 3640

2nd → 3640 - 1480 = 2160

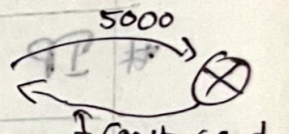
3rd → 2160 - 1480 = 680

4th → 680 - 680 = 0

Data	offset
0 - 1479	0/8 = 0
1480 - 2959	1480/8 = 185
2960 - 4439	2960/8 = 370
4440 - 5119	4440/8 = 555

DF:- DF = 1

MTU = 2000



$$n^{th}\ \text{offset} = \frac{\text{Initial Byte \#} - (\text{MTU} - \text{Header}) \times (n-1)}{8}$$

ICMP Protocol

→ Internet Control Message Protocol

→ helping protocol.

Functions

→ reports errors for missing packets.

→ Ping → if there is connectivity between 2 devices

→ Traceroute → unable to connect with a device can identify using traceroute which router was problematic [Trace the source to destination path]

PING :-

* Test the reachability of a host → by sending/pinging certain url.
* Records any packet loss or situation of the network
  - sends 4 ping → 2 ping lost 50% packet loss
    - To ensure it reaches the host
    - 0 " " 0% " " good connection

(i) Check connectivity
(ii) Measure network speed
(iii) Troubleshoot issues

ICMP Message :-

ICMP ☆☆

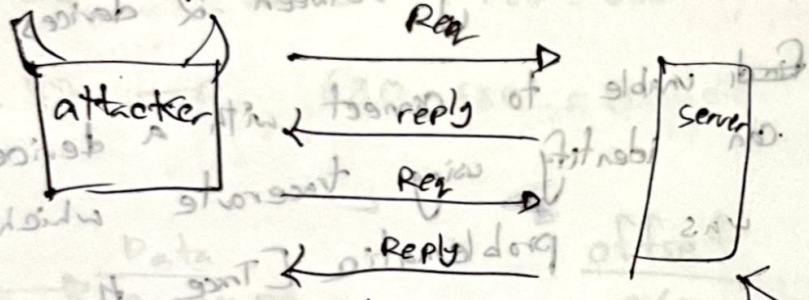
Type	code	Message
ping req → 8	0	
ping reply → 0	0	

Type	Code	message
3	3	port unreachable

#PING ATTACKS:-

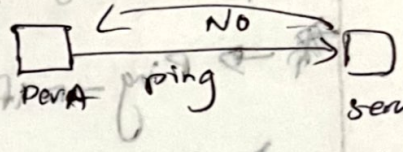
PING Flood Attack / DOS Attack

Attacker sends ping multiple times by changing source address every time!



sends thousands of ping and the server ultimately crashes.

Sometimes, If we ping a server it doesn't replies. But we can visit the website perfectly



then what will I load the server?

can't access the server as the server crashed

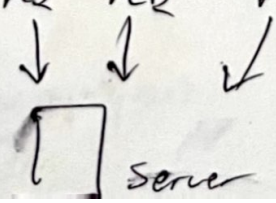
DDOS Attack / Zombie Attack:-

Attacker sends mail to multiple users and sends a ~~trojan / virus~~ trojan / virus as an attachment.

clicks ——— virus ——— gets installed.

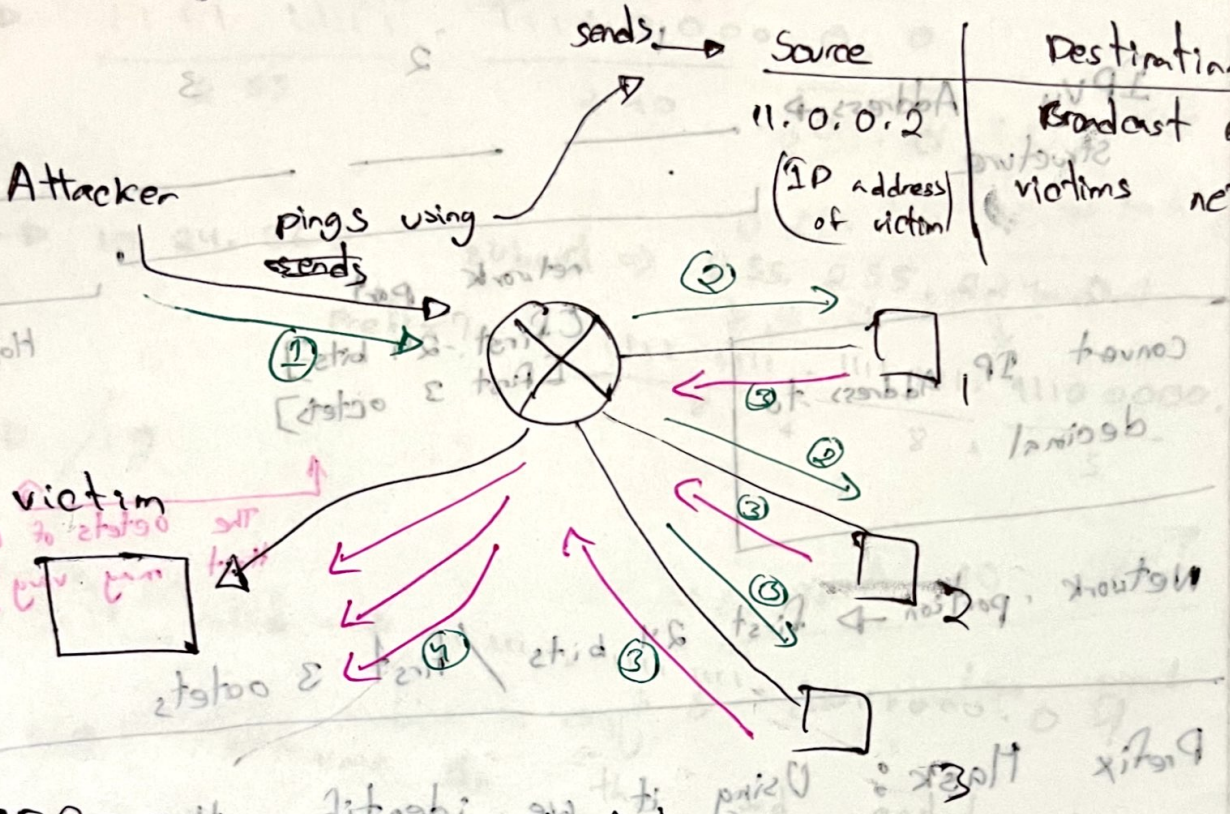
multiple the n users

pings the server again & again



Therefore, the server crashes.

Packet magnification / ICMP Smurf



TRACEROUTE :-

TTL = 1 => 1 hop
2 => 2 hop

STEPS

- Source address → own device IP
- dest address → the one device I want to reach
- TTL → increaments ~~over~~ by 1 [starts from 1] every time ~~STARTS~~ by incrementing ~~from 1~~
- returns → destination ~~with~~ address where ~~destination~~ TTL expried

Will continue until I get dest addres

There is no "Tranport layer" in Intermediate devices

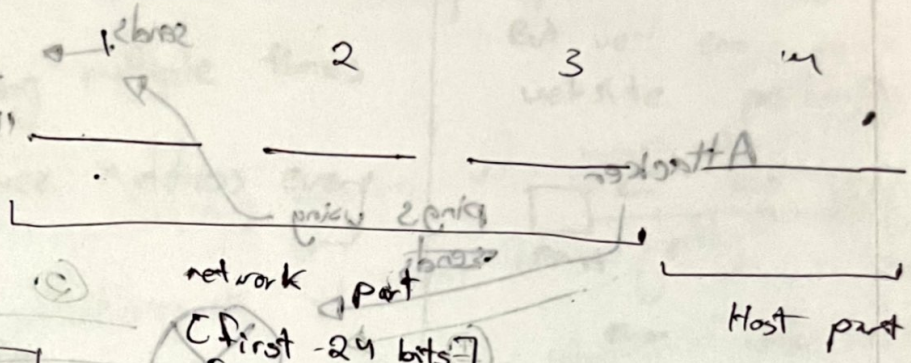
Network tool used to trace the path a data packet takes from a computer to a target server

[MOSTLY FOR TROUBLESHOOTIN

IPV4 Addressing

IPV4 size → 4 x 8 = 32 bits
byte

IPV4 Address structure



Convert IP Address to decimal

The octets of Network and Host may vary!

Network portion → first 24 bits / first 3 octets

Prefix Mask: Using it we identify the network portion and host portion.

Subnet Mask: The bits which are 1 they are network and 0 bits are host part.

example:

Prefix → /24 ⟹ Subnet 255.255.255.0

/16 ⟹ " 255.255.0.0

/8 ⟹ " 255.0.0.0

EXAMPLE

IP → 10. 24. 36. 2 / 12 subnet ?

11111111 . 11110000 . 0 . 0

255 . 240

IP → 10. 24. 36. 2 subnet ⇒ 255. 255. 240. 0

Prefix?

1111 1111 . 1111 1111 . 1110 0000 . 0

8 + 8 + 3

→ /19

Is it valid → 1111 1111 . 1111 1111 . 1011 0000 . 0 ?

→ No! because on the 3rd octets the 1's are not consecutive, or Also, the only the first part can be the network part the 0's can't be in between.

AND OPERATION

IP Address AND Subnet Mask → Network Address.

IP → 133. 15. 2. 1

Subnet → 255. 255. 255. 0

network → 133. 15. 2. 0

1 and 0 = 0
0 " 1 = 0
0 " 0 = 0
1 " 1 = 1

TYPES OF ADDRESS :-

Every network has 3 types of address → (i) Network Address → first
(ii) Broadcast " → last
(iii) Host "

All of host in the network have the same network bit

10.0.0.0/24
[First 24 bits are network part]

In case of Broadcast address all host bit's are 1.

MAX No. POSSIBLE Host WITH IP Address :- $2^8 - 2 \Rightarrow 254$
(N+B)

Example :-
192.168.5.30/25

32 - 25 = 7 → $2^7 - 2 = 126$

EXAMPLE →

IP → 192. 168. 10. 193 /24 Find S.A. B.A N.A.

IP => 192. 168. 10. 193 → 1100 0000 1010 1000 0000 1010 1100 0001

Subnet => 255. 255. 255. 0

Network => 192. 168. 10. 0

Broadcast => 192. 168. 10. 255

1st available IP => 192. 168. 10. 1

last available IP => 192. 168. 10. 254

SPECIAL ADDRESS :-

(i) Unicast → To send a particular host

(ii) Broadcast Address → To send all hosts of a network

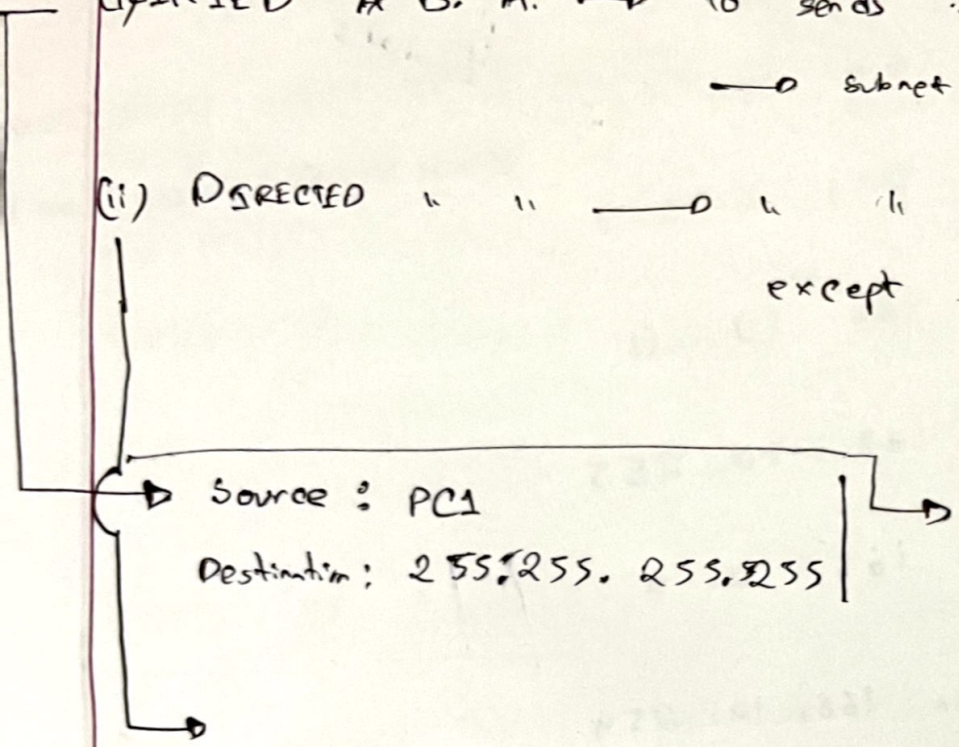
(iii) Multicast → To send a group of hosts.

BROADCAST ADDRESS :-

(i) LIMITED B. A. → To sends own networks all hosts.

→ Subnet 255.255.255.255

(ii) DIRECTED " " → " " other networks host except its own networks "



MULTICAST ADDRESS :-

The software assigns a particular multicast address to the hosts who have installed.

Source: 172.16.4.1

Dest: 224.10.10.5

192.1

Given,

IP → 192.168.170.13/20 Find → S.M.; N.A.; B.A.; 1st U.A.; 2nd last U.A.

Network bit → 20
host " → 12

Subnet → 255.255.240.0

Network → 192.168.160.0

Broadcast → 192.168.175.255
(All host will be 1)

240 → 1111 0000
170 → 1010 1010
1010 0000
160

STARTS from ...160.

1st usable IP → 192.168.160.1

2nd " IP → 192.168.160.2

last " IP → 192.168.175.254

1010 1010
1010 1010
1010 1111
175
All host

IPV6 & IP Addressing

Header length → 40 byte (fixed)

→ no fragmentation allowed.

IPv6 follow string notation → ":"

IPv6 Representation

Rule 1: All 0 → 1 0

0000 → 0

preceding '0' will be discarded →

0 9 c 0 → 9 c 0

0 0 0 5 → 5

Rule-2: One or more 16 bit (consecutive) segments consisting of all zeroes can be represented using "::"

~~we can~~

But we can use this only once "::" in the IP address

Example

1080 : 0 : 0 : 0 : 800 : _ : _ : _ ⟼ 1080 :: 800 : _ : _ : _

0 : 0 : 0 : 0 : 0 : 0 : 0 : 1 ⟼ ::1

DHCPv4

IPV6 & IP address representation

Network → Prefix Mask (present) /64
→ NO Subnet Mask present

Normally → first half → Network part
last half → Host part

WHY IPV6?

→ 32 bit address space soon to be completely allocated.
→ simple header format → speed processing.

1080 : 0 : 0 : 0 : 0 : 0 : 0 : 0

DHCP OPERATION:

(i) Using router
(ii) Using dedicated server

DHCPV4

When a certain host connects to a network, the DHCP server dynamically assigns an IP address to the host.

"It is a Protocol"

Why DHCP needed?

→ If there was no DHCP then we had to assign all the IP addresses, DNS server, and default gateway manually. Also, there may arise some problems while assigning those things.

Timeline:

(i) Step-1 device connects to a network and it requests an IP address.

(ii) A DHCP server on the network listens for these requests and assigns/leases an available IP address for a limited period of time

→ If a DNS server changes its IP address has changed

The DHCP server provides the changed IP address, DNS, subnet Mask and default gateway to the host

#DHCP OPERATION:

two way (i) Using router →

(ii) Using dedicated server → If the organization is large

*DHCPv4 works in client/server mode

When DHCP IP address returns the IP address to the pool

→ Step-1 → When a client communicates with a DHCPv4 server the server assigns/leases an IP address to that client along with default gateway, DNS server, subnet mask, domain name for a short period of time.

when lease expires

step2 → when a lease is about to expire the client requests the DHCP server to extend the time limit of the IP address.

step 3:- → Then the DHCP server extends the lease time. Else:- If lease expires and doesn't requests IP address is returned to the DHCP pool.

(ii) when DHCP server doesn't get any connectivity reply.

Step 1:- DHCP server constantly checks the clients connectivity if it is still connected to the address

If yes:- do nothing

else:- Returns the IP address to the DHCP pool.

DORA [How to obtain an IP address]

(i) Discover the DHCP server by Broadcasting. the request.

why Broadcast?
→ the client doesn't know where the DHCP server is

(ii) Offers an available IP address to lease (Unicast)

(iii) DHCP request, ~~acceptance~~ Broadcast the acceptance of IP address to the selected server and Implicit decline to other server.

why again Broadcast
→ If there are more server in the network.

(iv) ~~verifies~~ DHCP Acknowledgement or verifies the lease. Info.

Lease RENEW :-

DHCP REQUEST ⟼ Before lease expires client sends a request to DHCPv4 server.

If IP Add already offered to other client :- If the due to some sorts of delay DHCP request arrives late and the a IP leaase gets expired then the DHCP may assign that IP address from the pool to another client. In this situation, DHCP will reply ⟶ DHCP NACK.

NOT OFFERED :- After receiving the DHCP REQ (Before the lease gets expired) message the server extends the lease by returning DHCP Ac

DHCP RELAY

☆ When a client requests for IP addres to the Router does not fowards the broadcast packet (drop) rather it drops it.

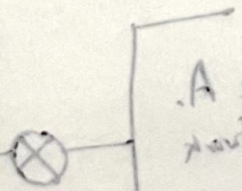
→ To solve this problem the router is setted in sveh a way that it sends the request to DHCP server using TAN #

STEPS TO CONFIGURE

Step 1 → Exclude the specifying IPv4 address because we use those address for printer, server, default gateway, as static address.

Step 2 → Name the pool

step-3 → set network address and subnet address



COMMANDS

Show running config | section DHCP ⇒ displays the DHCPv4 configured on R1

Show ip dhcp binding ⇒ displays a list of all IPv4 addres to MAC address binding provided

show ip dhcp server statistics ⇒

no service dhcp ⇒ the router won't work as dhcp

NAT [network address translation]

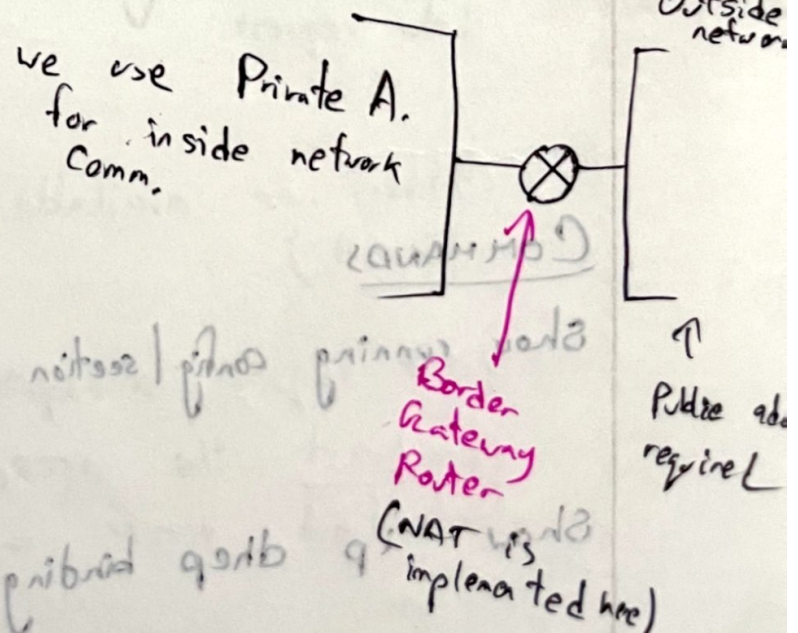
uses two address types →

(i) Private → can't access internet using Pr. add

(ii) Public
↳ can access internet using Public address

Private address → NAT [translates] → Public address

How NAT Works?



→ (i) Inside
(ii) Outside
(iii) Local → Private address of the inside of the Network
(iv) Global → Public

" Outside " "

Inside local address

Inside global address → Private converted to Public address

Outside L. A. → the one which I am communicated with (his public address)

Outside G. A. → "Port" " " " with (his private address)

NAT SENDING

PAT [used when Not Overloaded] → Port Address Translation

combination of IP Address & Port Address

↳ IP Add. : Port Add.

Private Address → PAT Translates (IP : P.A.) → Public Address (Internet)

↳ Source Port Address

~~Next~~ Problem :-

There might be a scenario when the Port Address will become the same for two devices.

Solve :-

Router changes the "port" number of the next device (who tried to access the Internet and faced the scenario) with the next available port.

PROBLEM 2 →

~~Port Address is av.~~

~~Bat~~ PAT works with Port Address and it is available in transport layer. But there are some protocols like ICMP which does not have transport layer. Therefore, the packets do not contain layer 4 ~~No. of Number~~ port number

Solve :-

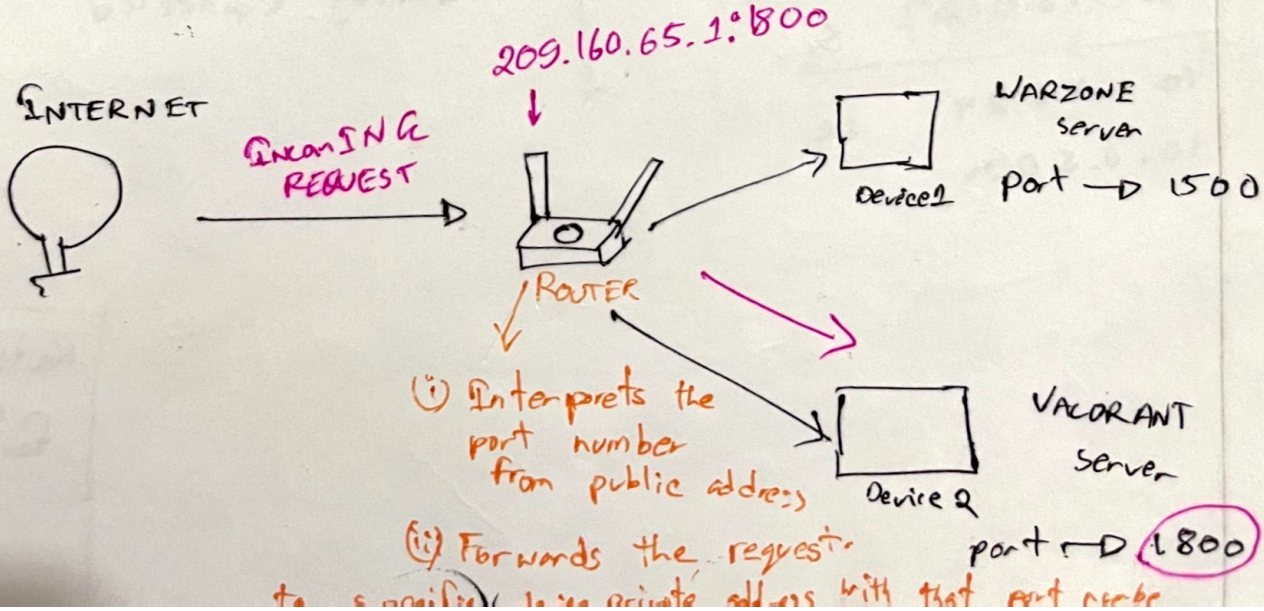
In this case PAT provide an ID from itself and ~~and using it~~, PAT differentiate between IP addresses.

ADVANTAGE :

1 public is enough to send 4000 or more private address to internet

PORT FORWARDING :-

If someone wants to access a device/IP address from outside the network then we use "port forwarding" using the port number.

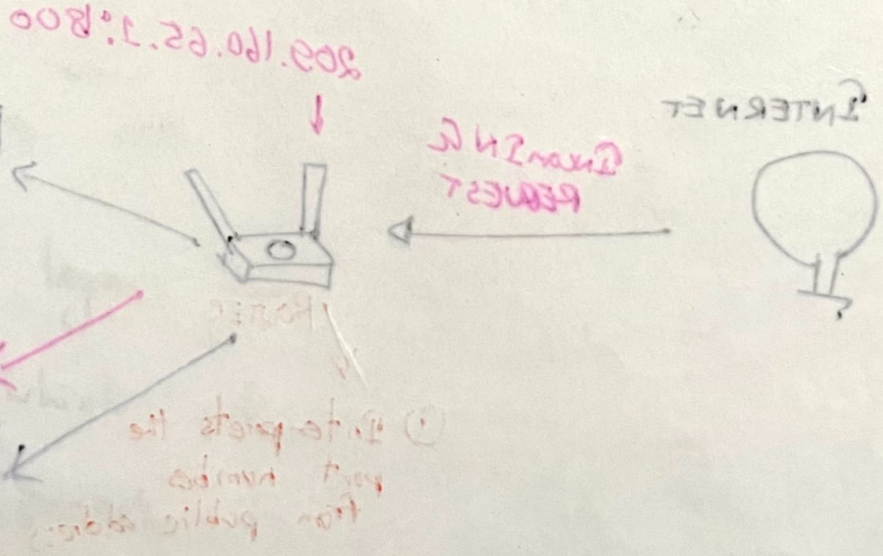


COMMAND

→

ip nat inside source static tcp 192.168.10.254 80

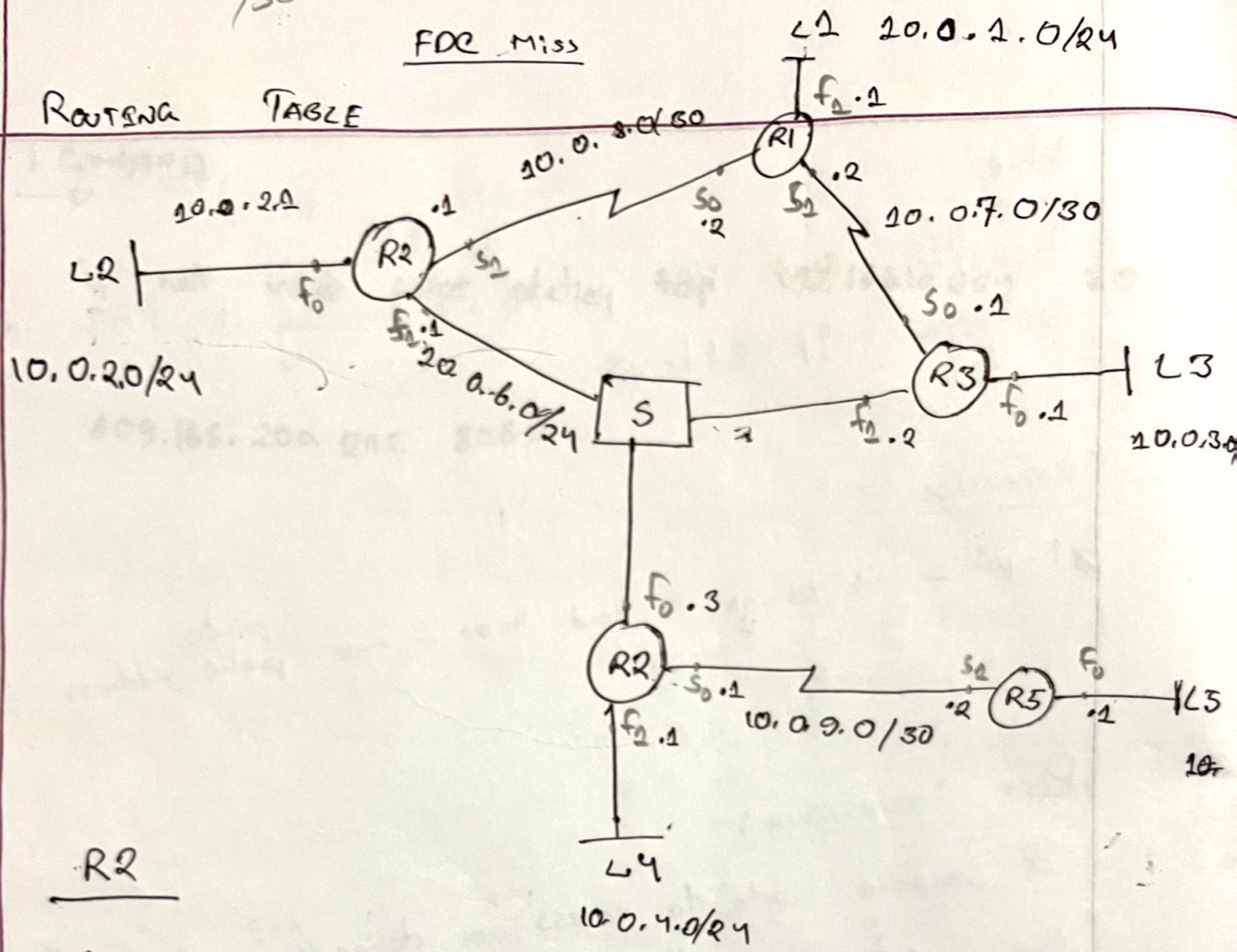
209.165.200.225 8080



/30

FDC Miss

ROUTING TABLE



R2

Network Add.	Exit Interface / Port
10.0.2.0/24	f0
10.0.6.0/29	f1
10.0.8.0/30	S1
10.0.1.0/24	S1
10.0.5.0/24	

→ directly connected.

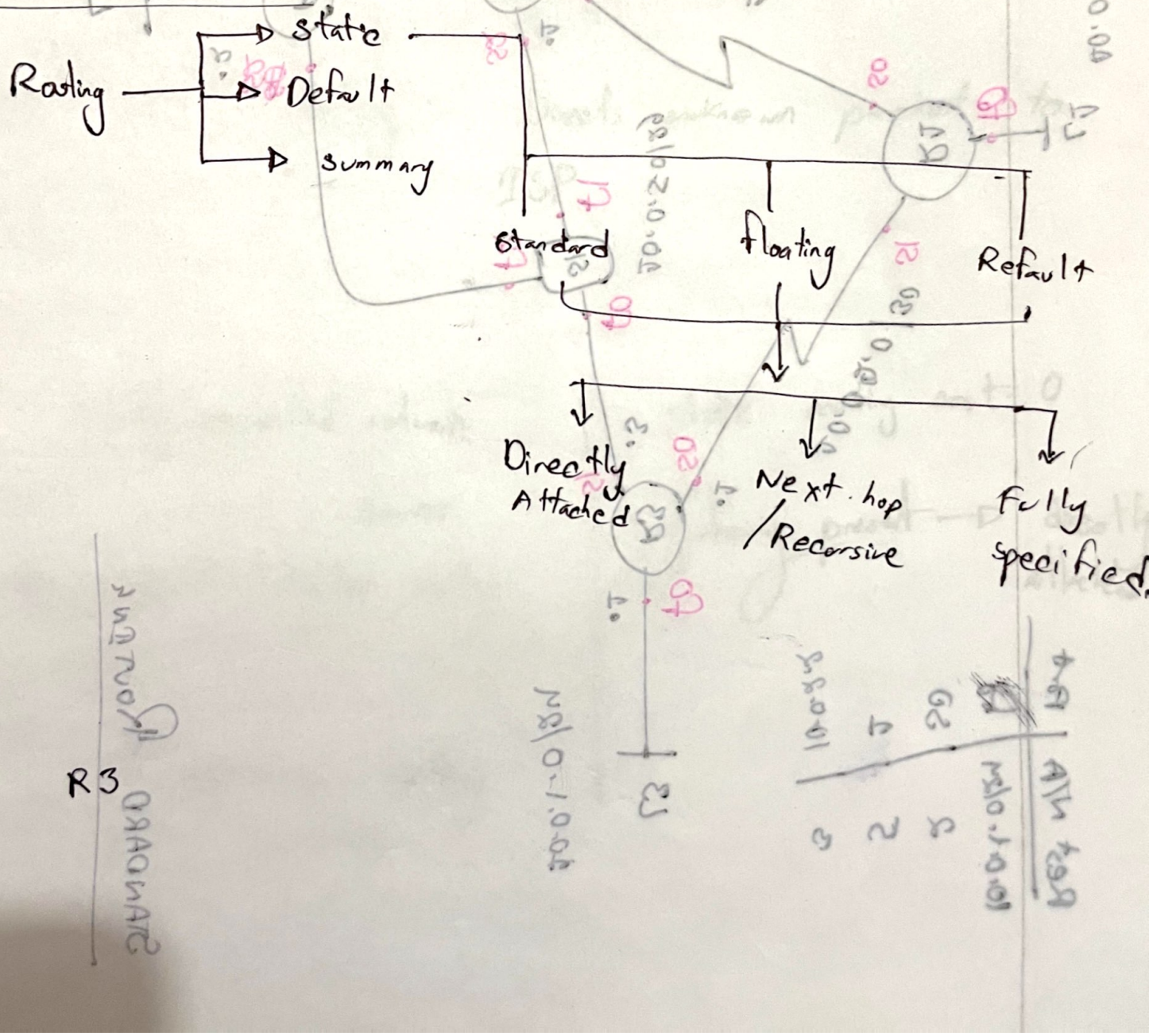
Routing command for exam!

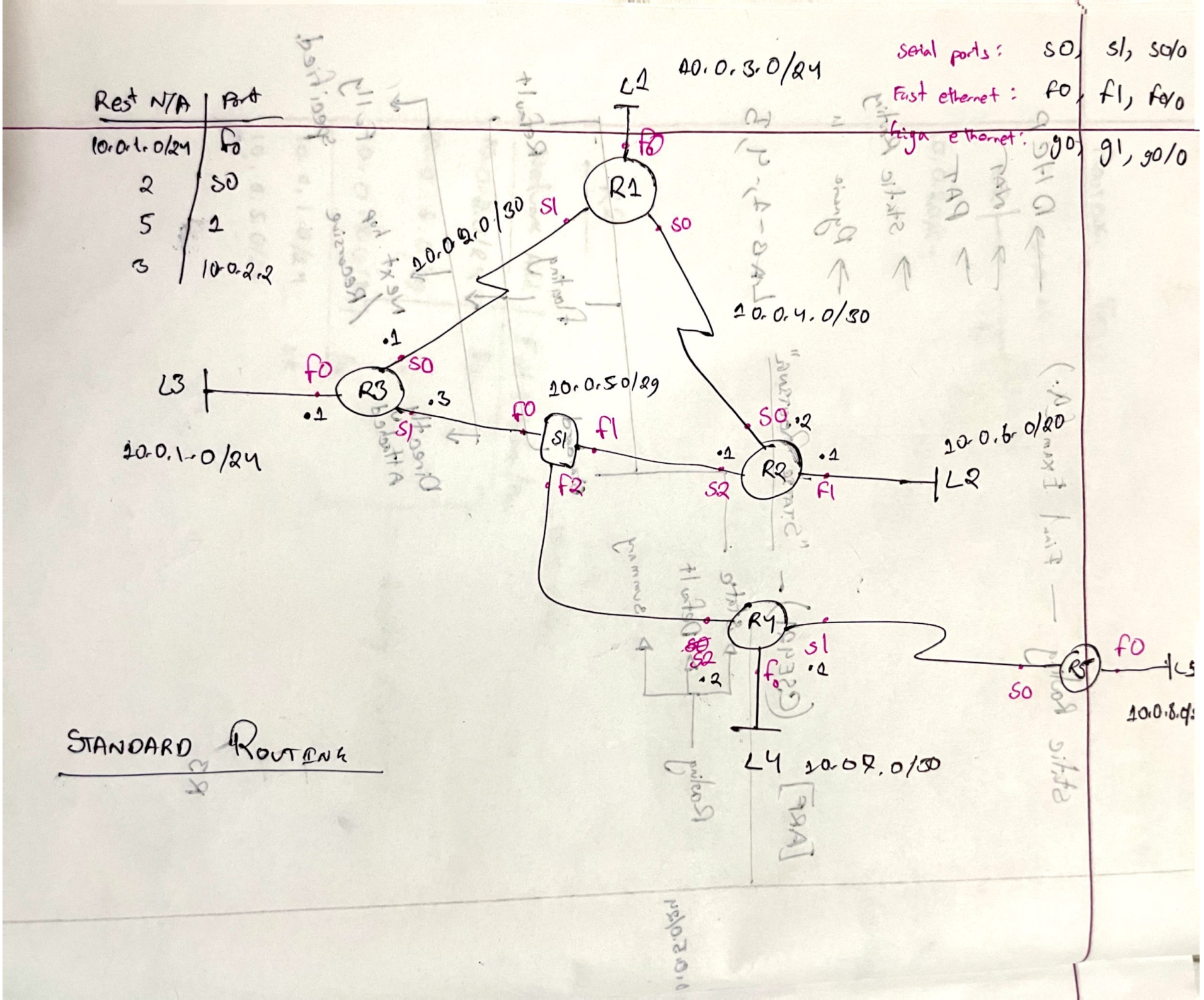
Static Routing — Final Exam (Q.)

→ DHCP
→ NAT
→ PAT
→ Static Routing
→ Dynamic "

[ARF] (CSE421) — "STATIC ROUTING"

LAB - 1, - 4, 5




Serial ports: S0, S1, S0/0
Fast ethernet: F0, F1, F0/0
Giga ethernet: G0, G1, G0/0
Rest N/A | Port
10.0.1.0/24 | F0
2 | S0
5 | 1
3 | 10.0.2.2
L1
10.0.3.0/24
F0
R1
S1
S0
10.0.2.0/30
10.0.4.0/30
.1
S0
F0
L3
R3
.3
10.0.5.0/29
.1
S1
F0
F1
S0
.2
10.0.1.0/24
.1
.1
R2
10.0.6.0/20
L2
F2
S2
F1
R4
S1
.2
S2
.2
F0
S0
R5
F0
10.0.8.0/
L4 10.0.7.0/30
STANDARD ROUTING

#SUBNETTING

All IPV4 addresses are occopied

Soln

Long term :-

change to IPV6

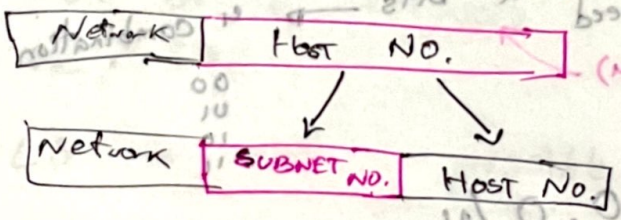
Short term

→ Subnetting (Reduces the wastage of IP address)

→ DHCP

GOAL → to partition a single physical network into more than 1 smaller logical sub network

↔ how we do that?



SUBNETTING METHODS → 3

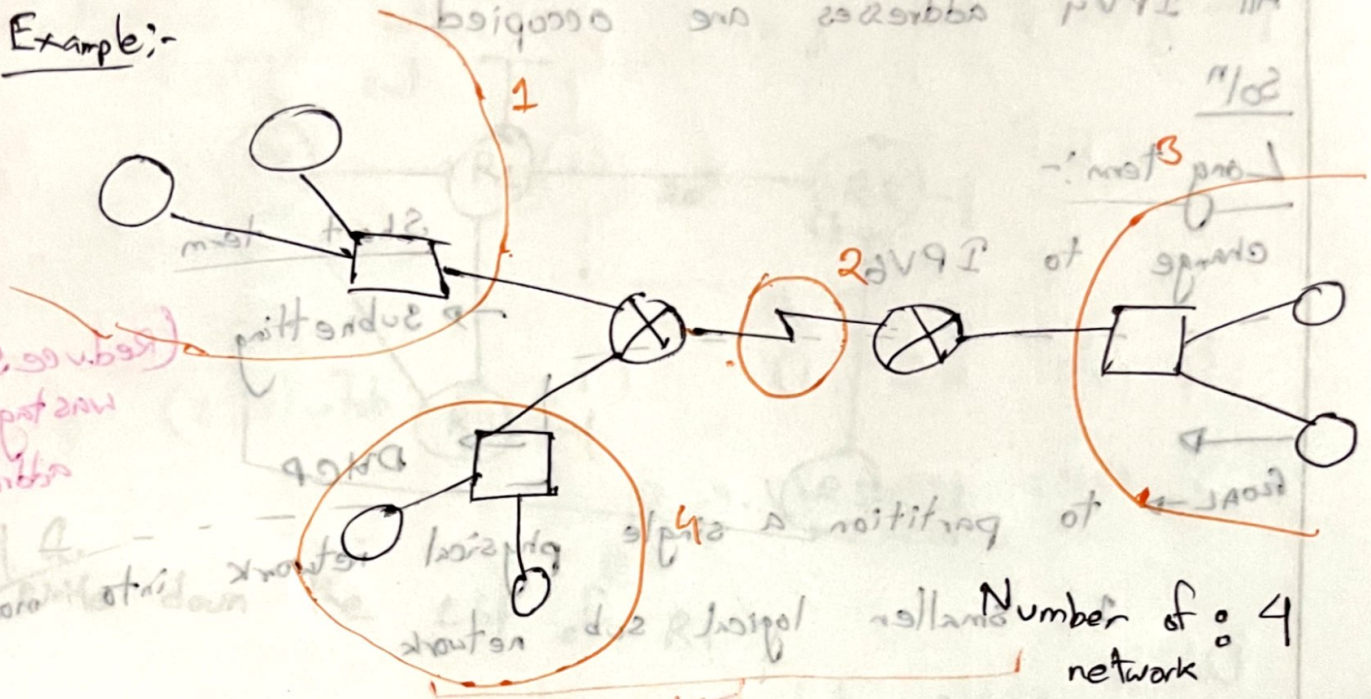
(i) classful IP addressing

(ii) Fixed length subnet Masking

(iii) Variable Length Masking

FIXED LENGTH SUBNETTING

Example;-



Number of network : 4

4 networks → need → 2 bits → 4 combination → 4 address

$\log_2(4)$

00
01
10
11

Given,

192.168.0.0/16

→ 192.168.0000 0000.0000 0000

⇒ 11.11.000.....

⇒ 11.11.010.....

⇒ 11.11.100.....

⇒ 11.11.110.....

why 18? → network bit +

dec

1 → 192.168.0.0/18

2 → 192.168.64.0/18

3 → 192.168.128.0/18

4 → 192.168.192.0/18

VARIABLE LENGTH SUBNETTING (VLSM)

EXAMPLE

192.168.16.0/24 | Host part → 8

→ ⇒ 192.168.16.0000 0000

5 network → 3 Network bit

Remaining bit = Host bit = 8 - 3 = 5

No. of IP address = $2^5 - 2 = 30$

There are → 120 + 60 + 20 = 200 hosts

and the no. of IP addresses available are only 30

So, FLS is not possible

→ VARIABLE SUBNETTING :-

192.168.16.0000 0000

0000 0000 (1)

1000 0000

1000 0000 (2)

1100 0000

1100 0000 (5)

1110 0000

1110 0100

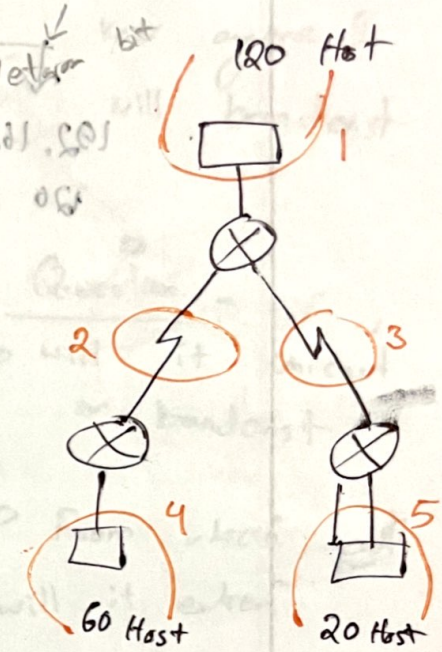
1110 1000

1110 1010

1110 1011

FOR HOST

FOR NETWORK



5 networks ≈ 3 bits required

For 120 hosts → (Network 1)

120 + 2 = 122 ≈ 7 bits required

For 60 hosts →

60 + 2 = 62 ≈ 6 bits required

For 20 hosts →

20 + 2 → 22 ≈ 5 bits required

For 2 hosts (WAN) →

2 + 2 → 4 ≈ 2 bits

TREE

192.168.16.0/24

192.168.16.0/25
120 Hosts

192.168.16.128/25

192.168.16.128/26
60 Hosts

192.168.16.192/26

192.168.16.192/27
20 Hosts

192.168.16.224/22

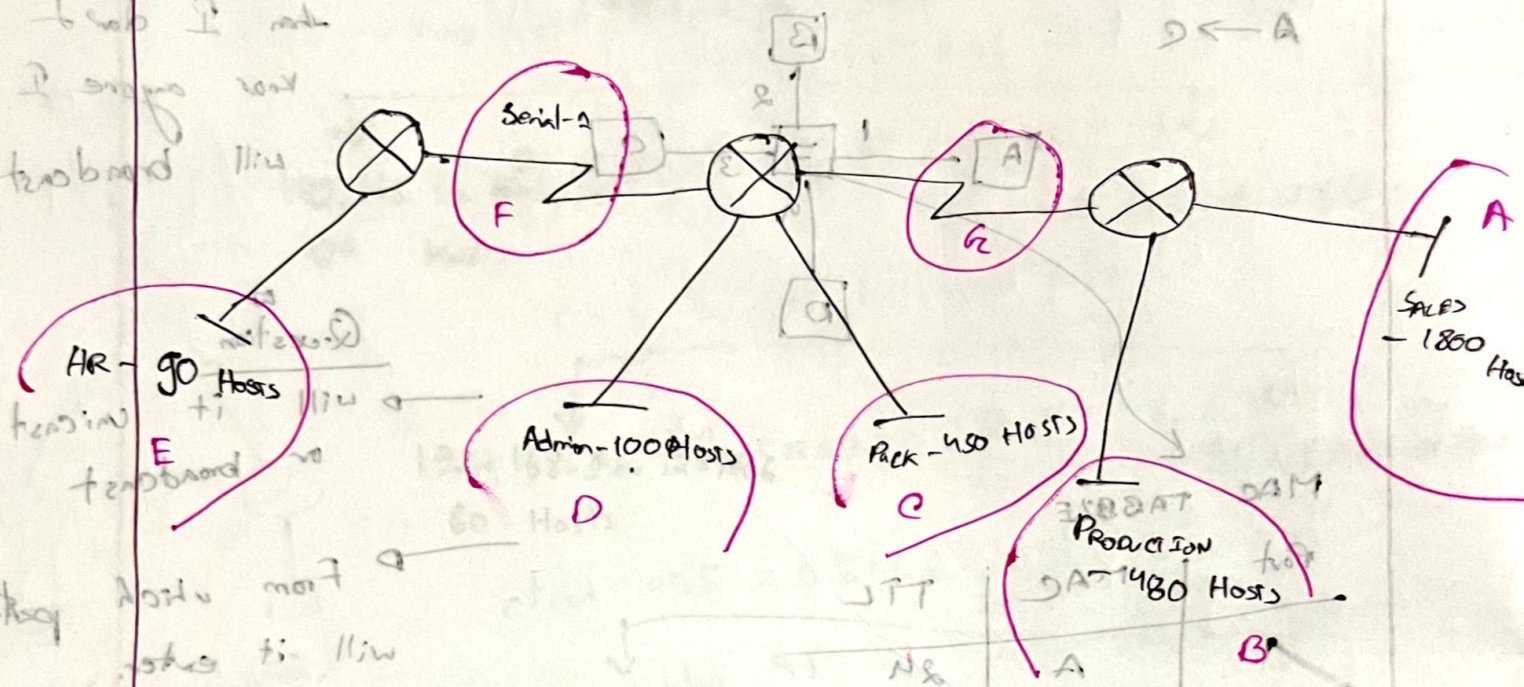
192.168.16.224/30
(WAN-1)

192.168.16.228/30
(WAN-2)

that this does need VLS

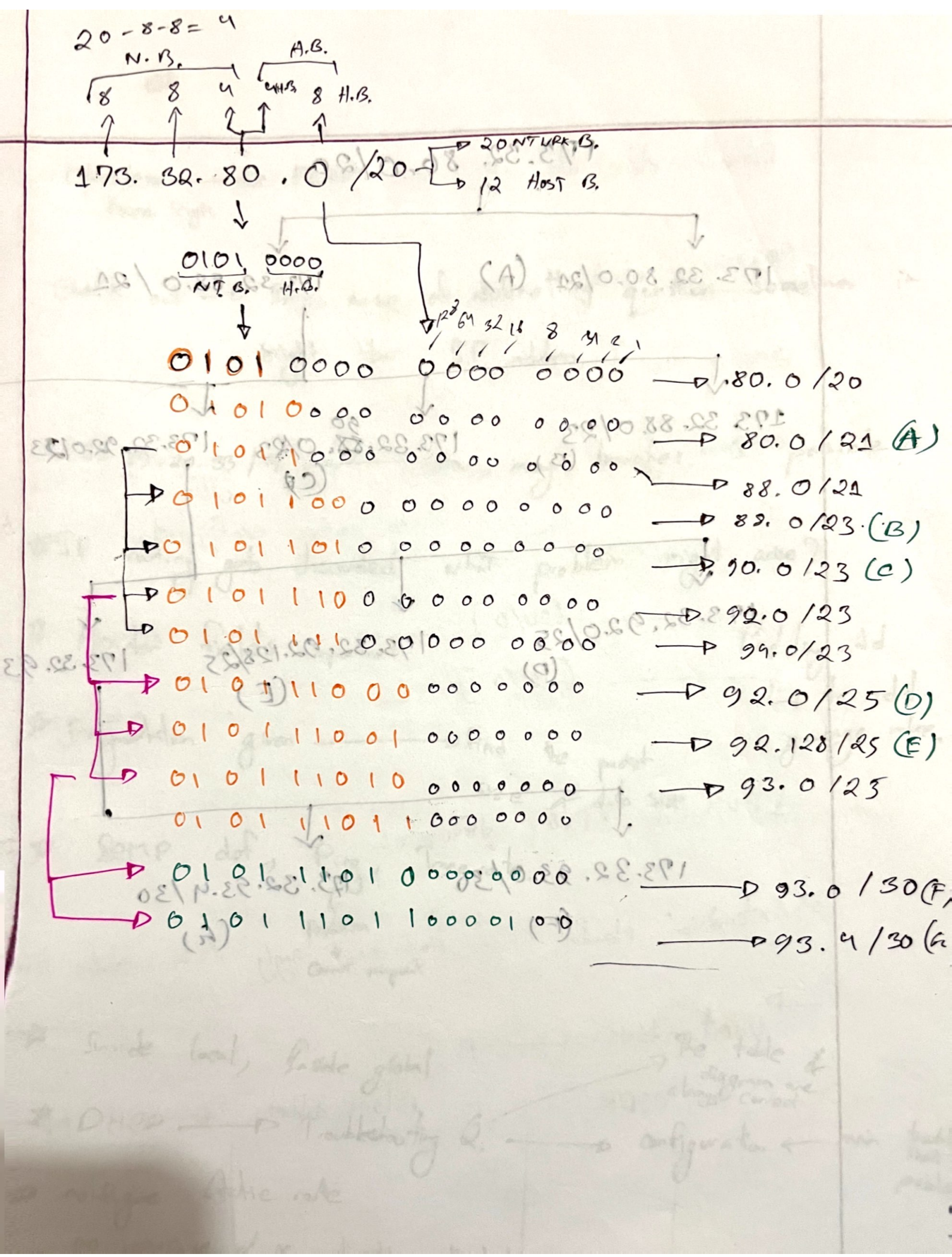
SUBNETTING (EXAMPLE)

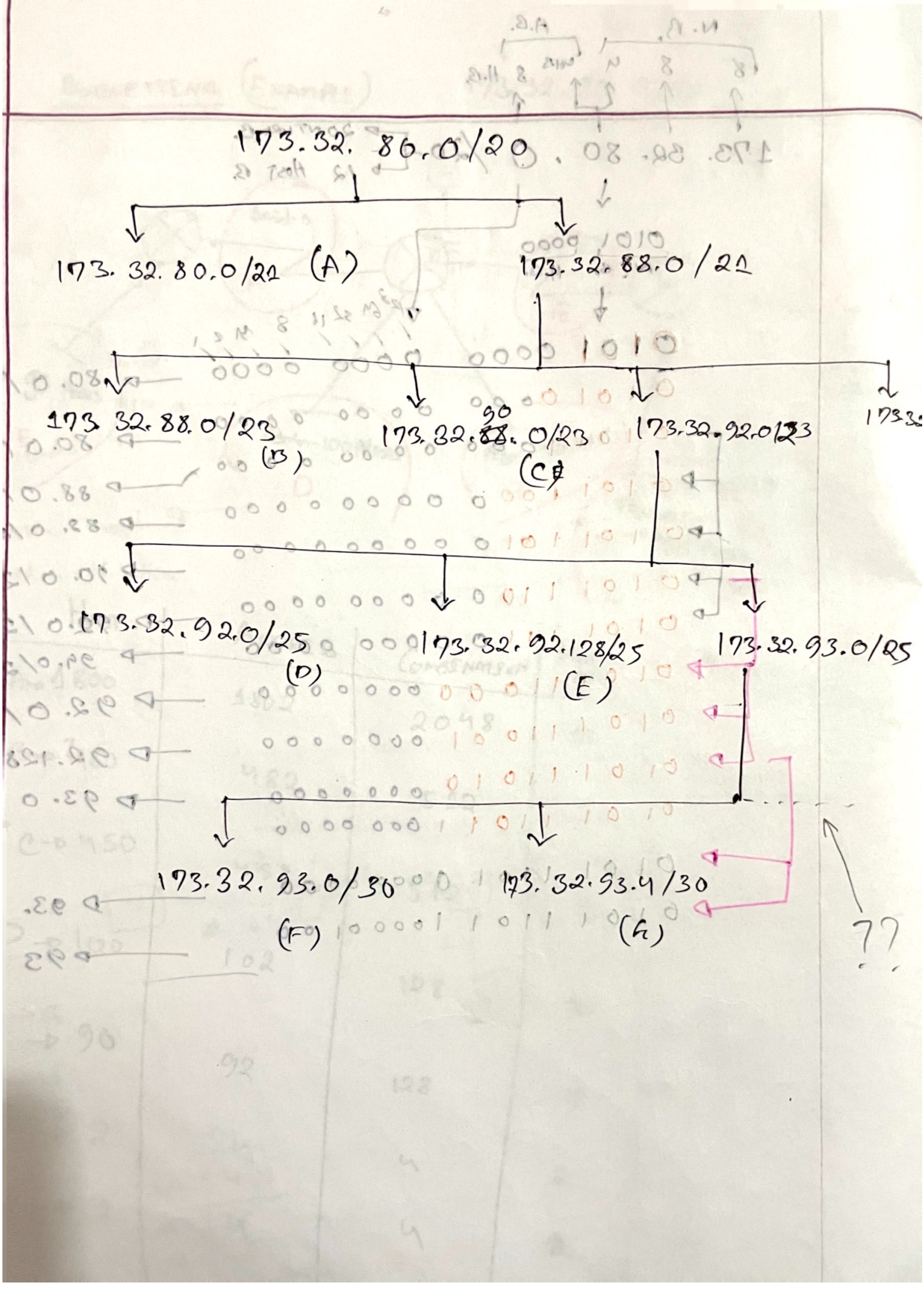
173.32.80.0/20



bit	comb
8	256
9	512
10	1024
11	2048

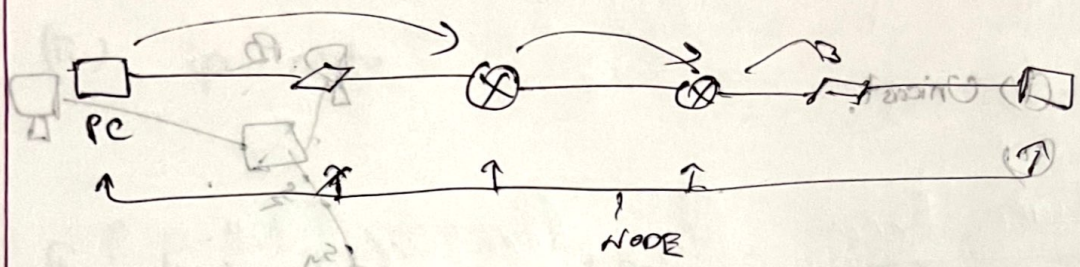
Host	ADDRESS	COMBINATION	HOST BIT	NETWORK BIT / PREFIX MASK
A → 1800	1802	2048	11	21
B → 480	482	512	9	23
C → 450	452	512	9	23
D → 100	102	128	7	25
E → 90	92	128	7	25
F → 2	4	4	2	30
G → 2	4	4	2	30


20 - 8 - 8 = 4
N.B.
A.B.
8
8
4
SUBNET B.
8 H.B.
173. 32. 80 . 0 /20
20 NTWRK B.
12 HOST B.
0101 0000
NT B.
H.B.
128 64 32 16 8 4 2 1
0101 0000 0000 0000 → 80.0/20
0101 0000 0000 0000 → 80.0/21 (A)
0101 1000 0000 0000 → 88.0/21
0101 1000 0000 0000 → 88.0/23 (B)
0101 1010 0000 0000 → 90.0/23 (C)
0101 1100 0000 0000 → 92.0/23
0101 1100 0000 0000 → 92.0/25 (D)
0101 1100 0100 0000 → 92.128/25 (E)
0101 1101 0000 0000 → 93.0/23
0101 1101 0000 0000 → 93.0/30 (F)
0101 1101 0000 0100 → 93.4/30 (G)

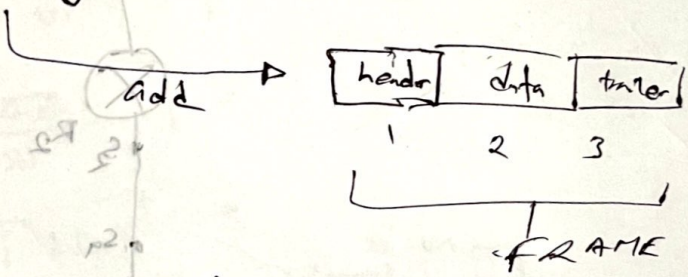

173.32.80.0/20
173.32.80.0/21 (A)
173.32.88.0/21
173.32.88.0/23 (B)
173.32.90.0/23 (C)
173.32.92.0/23
173.32.92.0/25 (D)
173.32.92.128/25 (E)
173.32.93.0/25
173.32.93.0/30 (F)
173.32.93.4/30 (G)
??

DATA LINK LAYER

Data link layer ensures hop to hop connection.



Dat link layer ⟶ PDU ⇒ FRAME



DATAGRAM IS TRANSFERRED BY DIFFERENT LINK PROTOCOLS OVER DIFFERENT LINKS.

LINKS: copper cable,
coaxial cable.
wireless ⟶ 802.11 protocol

[Protocol will be different according to links]

HERE,

TOURIST ⟶ DATAGRAM

TRANSPORT SEGMENT ⟶ COMM. LINK

.11 tation MODE ⟶ LINK LAYER

bus, uber, plane

TRAVEL AGENT ⟶ Routing algorithm.

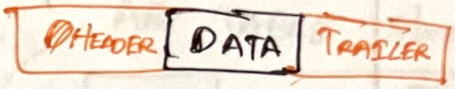
Je amake kothe dibe kivabe jabe kivabe jabe

Flat ← does not have any hierarchical structure.

LINK LAYER FUNCTION :-

(i) FRAMING :- Adding data with header & trailer.

→ MAC ADDRESS USED IN FRAME HEADERS TO IDENTIFY SOURCE & DEST.

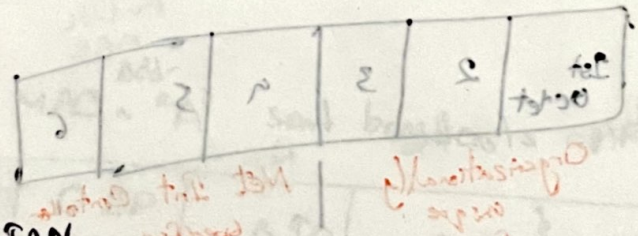


(ii) LINK ACCESS :-

(i) How to send a frame to the link.

(iii) RELIABLE DELIVERY BETWEEN ADJACENT NODES

→ In wireless link there are high error rates → so much reliability is needed.



(iv) ERROR DETECTION.

(v) ERROR CORRECTION

→ Corrects bit errors

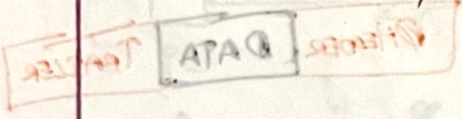
(vi) Flow control

(vii) Half duplex & Full duplex

LINK LAYER (HELPS TO) DELIVERY → HOP TO HOP

Where link layer is implemented?

→ Adapter / NIC (network interface card)



★ LINK LAYER ADDRESSING → MAC ADDRESS

→ 48 bits

→ portable

example → NIC card no.

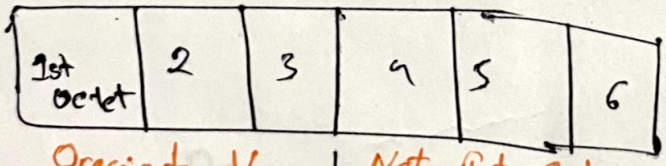
same in DHK

same in CHTG

→

48 bits MAC Add burned in NIC Room.

FRAME STRUCTURE



1 octet → 8 bits

2 digits

00

UNICAST

To send a single device

MULTICAST

To send grap of devices

BROADCAST

ARP → Address Resolution Protocol

→ Finds the MAC address using the IP address

* Every device has ARP table

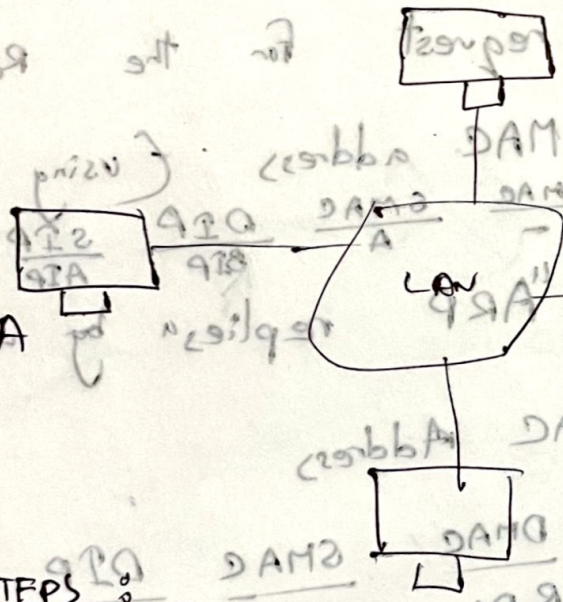
ARP TABLE stores →

IP	MAC	TTL

for 20 min

How ARP works?

→



A → B

sends

STEPS :

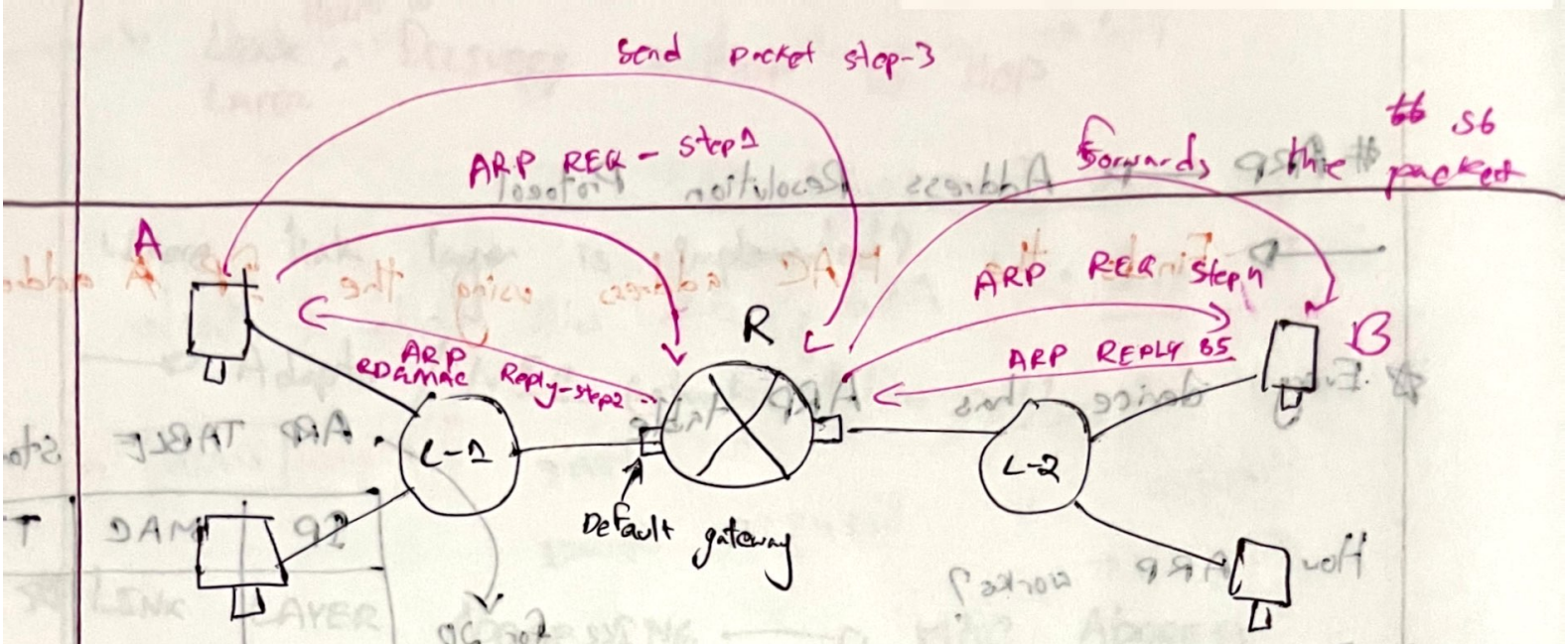
(i) To know the MAC Add. of "A" sent broadcasts ARP request

sends →

S MAC	D IP	S IP	ARP REQ

(ii) The specific device (B) matches the IP Address with itself and finds out it is an ARP req.

(iii) B sends ARP Reply with DMAC ; SMAC ; DIP ; SIP ; ARP REPLY



(i) Device 'A' ar. ARP request for the Router's default gateway MAC address (using the

DMAC	SMAC	DIP	SIP	Type
—	A	BIP	AIP	A to B

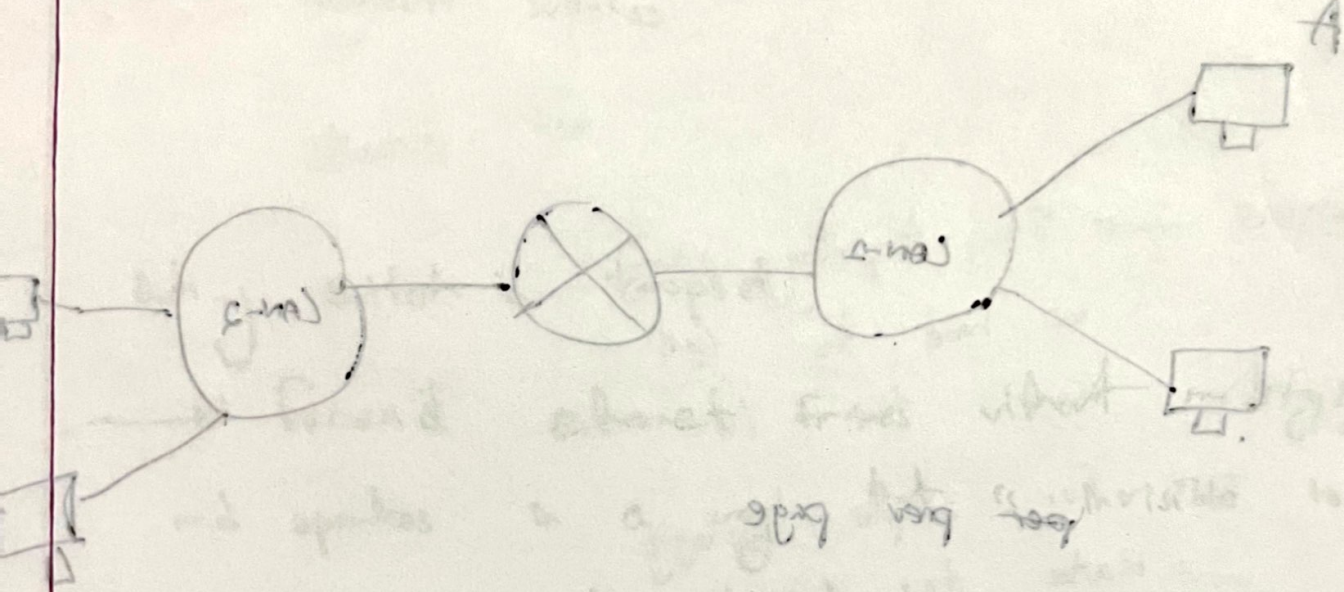
(ii) Then Router R "ARP replies" by with default gateway MAC Address

DMAC	SMAC	DIP	SIP	Type
R DR MAC	A	BIP	AIP	A to B

(iii) Then the Router's find out that the D ~~MAC~~ IP address is of Device 'B'. which is directly connected with.

(iv) Router ARP request for Device B's MAC address. ARP reply with B's MAC

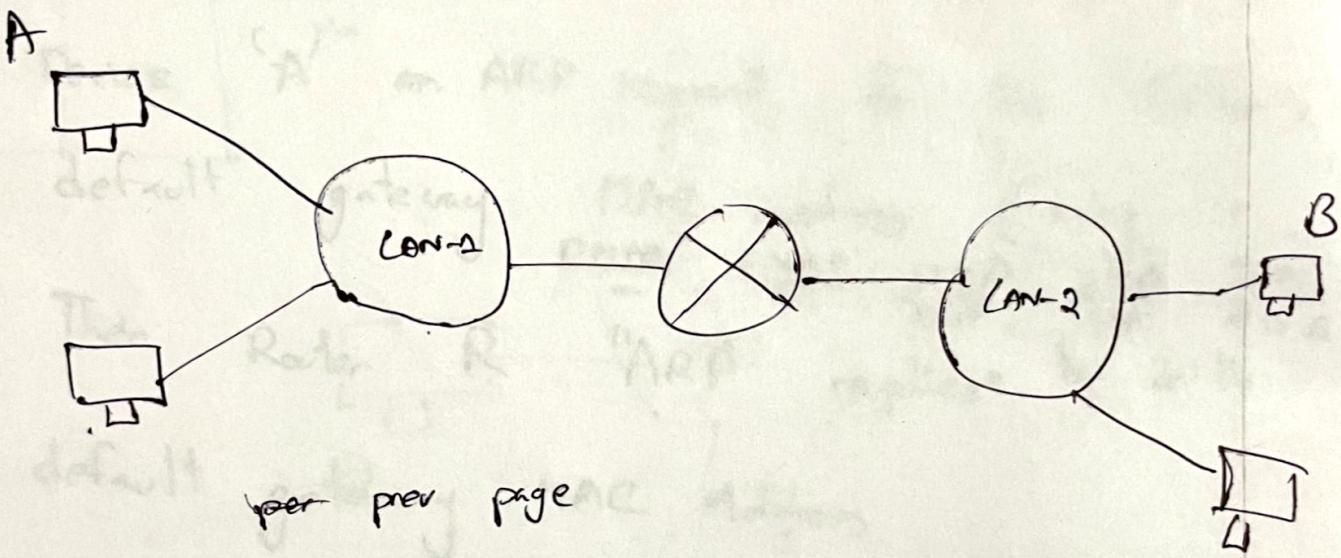
(v) The Router forwards the packet of device A to device B using MAC of B.



ARP is to Plug & Play without the intervention of admin

ARP table is created at the installation of software

Different LAN: working mechanism :-



per prev page

SWITCH

Functions :-

(i) Store & forward Ethernet frames.

(ii) Transparent → the device doesn't have any switches information

(iii) Plug & play → switches do not need to be configured.

→ In between switches

Why switch is transparent

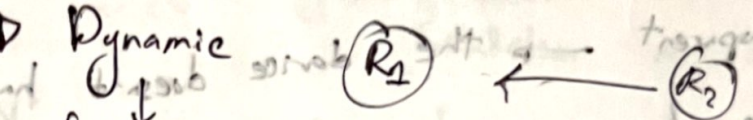
→ forward ethernet frames without modifying them and operates in a way that is invisible to the devices on the network.

→ No need for host awareness

STATIC ROUTING

Routing → Static (configures inside the router)

→ Dynamic (Remote routes are automatically learned via D.R.)



Routers communicate by themselves.

We don't perform static routing for directly connected networks.

#ve " " " for distant "

STATIC → If topology changes we have to let know the 100 routers

→ for simple topology

→ Route to destination always same

There are 4 types of STATIC ROUTING:-

(i) STANDARD S. R.

(ii) FLOATING " "

(iii) DEFAULT " "

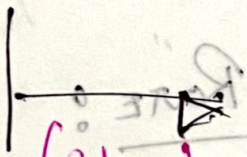
(iv) SUMMARY " "

ip route → targetted Network address → targetted Network s/m → exit interface / next hop IP address / fully specified

#In static routing we set the cost to 0 as the doesn't get calculated and we can select the path by our choice

★ Administrative distance :- Trustworthiness of the source of the rate.

PROBLEM OF NEXT HOP IP ADDRESS



the routing table lookup has to be performed two time.

↳ This is Recursive Rate lookup

PROBLEM OF EXIT INTERFACE

If it is multi access interface then exit interface will create problem.

In case of multi access interface the router doesn't know which will be the next hop.

If we observe that a switch is connected to multiple router.

DEFAULT STATIC ROUTE :-

Customer goes to ISP, then internet

If you found any packet whose source IP address is unknown to you then send the packet to the next IP address

⇒ IN SHORT, the router doesn't know how to reach the destination then send the packet through the default static router

Why default route?

→ It is not possible to configure S.R. for the huge no. of networks.

→ Even if we configure the situation of the routing T. would be too complex to handle. └ might tends to infinity

FLOATING STATIC ROUTE :

→ We create a backup route for a router

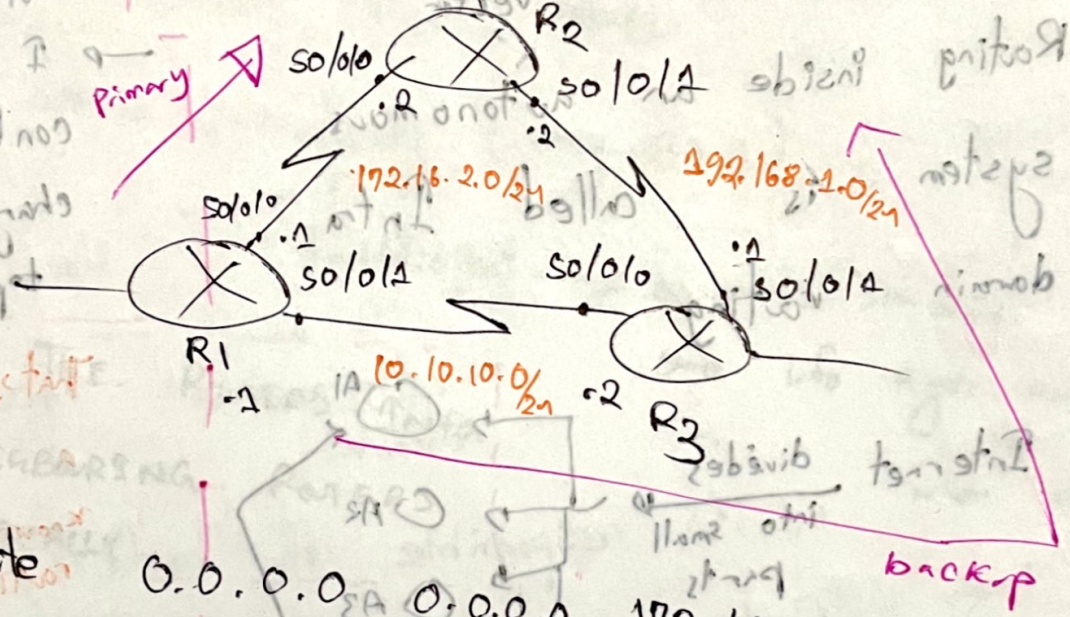
→ If the primary route for a router somehow gets down then some of the packet might get dropped due to the absence of a backup route. As a result the router would be able to use the backup route in case the primary router gets down.

STANDARD STATIC ROUTE AD is 1

" " " " 0 (when directly connected)

we perform floating static route with the help of AD. AD value shows the amount of trustworthiness of network

Configure floating static route :- default



ip route 0.0.0.0 0.0.0.0 172.16.2.2 ← next hop IP address

ip route " " 10.10.10.2 5 ← AD should be greater than 1.

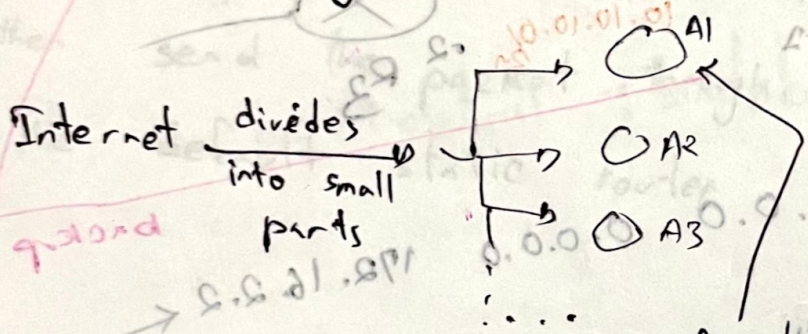
DYNAMIC ROUTING

Routing Protocol

- Inter Domain
- Intra

- Distance vector
- link state

★ Routing inside an autonomous system is called Intra domain routing.



chg #?

→ I have to know about all the networks and let know the route the huge amount of networks

→ I also have to configure if there is change on the topology.

That's why we follow a automated process known as dynamic routing

An autonomous system: group of networks & routers under single admin tion

Routing Algorithm:

R. A. finds the least cost path from the source to dest.

→ 2 types

- Global
  [gathers all the information first then applies the routing Algo.] takes idea of the full topology
  ⇒ Link state
- Decentralized
  [Gets partial info then applies routing algo. immediately]
  ⇒ Distance vector

DISTANCE VECTOR

Uses hop count

based on Bellford Algorithm

- estimate the cost to neighbours (by sending its own distance vector)
- Then estimate the cost to destination
- This occurs periodically (every 30 s)

↓

Inefficient → as it sending the same info again and again.

SHARES THE ROUTERS INFO TO NEIGHBOURING ROUTERS PERIODICALLY

↓ when all routers

*When every single router of the topology come to know about all the router's information Then we can say → The network has converged.

Amount of time it takes to converge ⇒ All routers know about all of the networks attached to all of the neighbouring routers

& size of the network

If the network doesn't converge I am not getting the optimal path.

Sometimes ↑ suboptimal path ↓ get the

LINK STATE ROUTING :-

- Uses Dijkstra Algorithm
  - shortest path first
- Exchange Hello packet

PROCESS

1. Each Router learns about its own directly connected network.
2. " " is responsible for contacting its neighbour (Exch Hello Packet) on D.C.N.
3. Each router builds a LSP containing state of each D.C. Link.
4. Each router floods LSP to all router.
5. " " uses all the LSP recieved to construct database.

Distance Vector
- protocols are like road signs

Link state
- protocols are more like road map

Does not uses hop count

Accumulate costs

Link state Packet

No Response of Hello sent
No LSP sent.

Neighbours discover
keep alive function

Hello packets: serve as a keepalive functions to monitor the state of the neighbour.

Link state routing protocol: reach convergence much faster than distance vector routing because each router floods the LSP to all of the neighbouring ~~route~~ router.

LSP only sends →

(i) Initial P startup
(ii) whenever there is a change in the topology (link going down, Neighbour Adj broken. or coming up

Build Routing Table
→ builds LSP
→ Flood it
→ Construct a database
→ Formation of S.P. tree for each node
→ calculation of a routing table.
shortest path

→ Fast Network converge
→ Map of Topology
→ Hierarchial design
↳ multiple area allows better route summarisation
→ Event driven updates.

IPv6

IPv6 : fields :-

(i) Traffic class :- Same as IPv4 - Type of service
↳ provides Quality of service.

(ii) Payload length :- The amount of data. (Total length in IPv4)

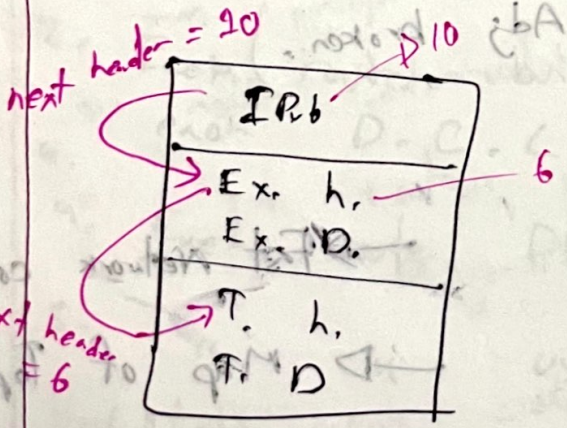
(iii) Hop LIMIT : TTL

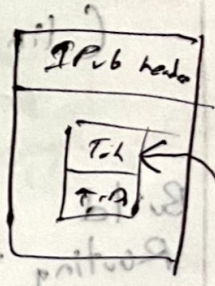
(iv) Source Add :- 128 bit

6 - header

EXTENSION HEADER :-

Next header : This field indicates which header the data has started with





Protocol is known as Next header in IPv6

INFO CARRIED IN NEXT HEADER

- Hop by Hop Options
- Routing header
- Fragment header
- ESP header

FLOW LABEL :-

If I want all packet to go with the same flow.
All ... will go to dest with same flow label

IPv6 Address type :-

- Unicast
- multicast (No broadcast in IPV6)
- ~~broadcast~~ using
- any cast
  - will send the packet to everyone but will receive only → the one who will be the closest.

Public Address :
is now unicast global Address

CIDR	Block Assign
0000 ::/8	special Add.
2000 ::/3	global unicast
FC00 ::/7	Unique local unicast
FE 80::/10	link local add.
FF 00 ::/ 8	Multicast "

Unspecified Addr.

When software need IP ~~but Add.~~ but don't have IP add
::/128 (Just for communicate with itself)

LOOPBACK ADDRESS

→ ff. The protocal stack working perfectly
::1/128

LINK LOCAL UNICAST:

→ Inside network ← Communicate → UNIQUE LOCAL
Globally unique

MULTICAST ADD.

→ starts with "FF"

ANYCAST

uses unicast

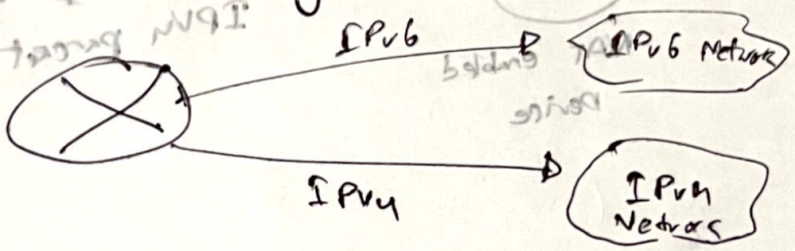
TRANSITION OF IPv4 to IPv6

3 techniques:-

DUAL STACK:-

Router is configured with both IPv4 & IPv6

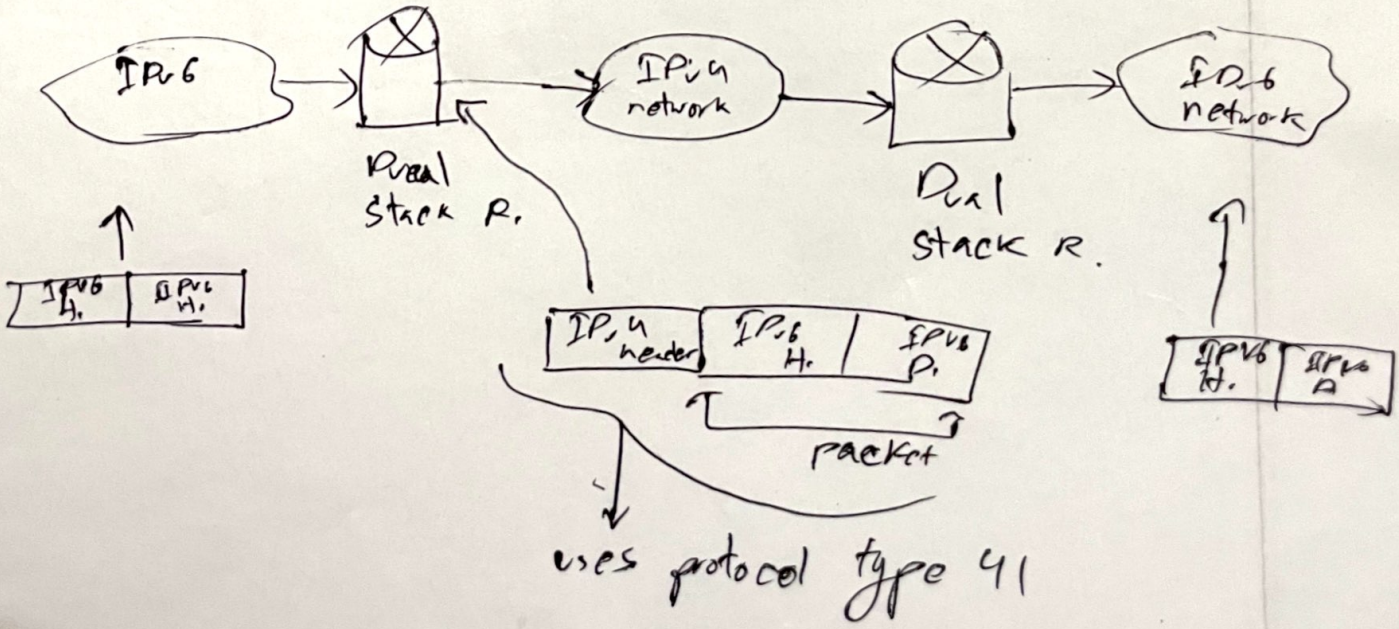
Involves running IPv4 & IPv6 at the same time



(Recommended)

IPv6 TUNNELING:-

Adds IPv4 header with IPv6 packet and sends through IPv4 network. Encapsulates the IPv6 packet in the IPv4 packet.



NAT PROTOCOL TRANSLATION:-