Bandwidth : Max. data transfer rate

Throughput : Actual data transfer rate

Router : ① Connects two networks.
② Routes the data packet based upon IP addresses.

Data transfer : ① Generate the message
② Divide the data
③ IP address of the reciever, Routing.
④ Using MAC address a packet is forwarded to different devices.

① TCP/IP : 5 layers (Practical Model)

Application
↓
Transmission
↓
Network
↓
data link layer
↓
Physical

② OSI (Open System Interconnection) : 7 layers.

Application → Presentation layer → session layer → Transport
↓
Physical ← DLL ← Network

(Reference Model)

Hypertext Transfer Protocol

Date ___/___/______ MAC address : Physical address (48 bits)

Saathi

→ TCP/IP model :

Application → msg communication, services - http, https, DNS, FTP, Telnet

DNS : protocol used to convert link into IP address.

Transport → (main function) segmentation, Port address, Process to process communication or end to end communication, functionality of error control, flow control (if recieving speed is 100 MB/S then sending speed must also be 100 MBps).

Network → Router, routing of packets, IP address, host to host communication.

data	Port	IP	MAC

data link layer → framing, flow control, error control.

Physical layer → Physical medium, cables, RJ connector, electric signal.

Date __ / __ / ____

switch : forwards the data packet using MAC addresses.

→ OSI model :

Application
Presentation
Session

Session → Dialogue control and synchronisation

- simplex (only 1 computer is allowed)
- half duplex (At a time only one comp. can send data. Total 2 computers are allowed) A ⇄ B
- full duplex

synchronisation → It resends only that data which has error in it. It is identified through checkpoints

Presentation:

→ It handles the encoding / decoding of the data (e.g- data is sent in ASCII codes format and the system to which it is sent supports UNICODE method)

→ encryption / decryption

→ compression

☆ In TCP/IP Model, the function of all these three layers are done only by Application layer

Saathi

Date __/__/__

Switching Techniques in Computer Networks:

1. Circuit Switching (only data, no packets)
2. Packet Switching (in packets)
   (Used Normally)

- First of all, establish a path, then reserve the resources and then transmit.

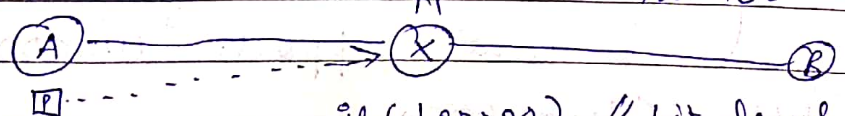
- Network controls where to send the data packet.

3. Virtual circuit switching:
   A packet switching technique in which we establish a path.

Different types of delay in Network:

1. Processing delay
2. Queuing delay
3. Transmission delay
4. Propagation delay.

→ Processing delay: Delay to process header info. by a router.



if (!error) // bit-level
then check IP address // in headers
and check network address based on IP address
then where to send the packet i.e. lookup to routing table.
if router is free then it is sent otherwise it is stored in queue.

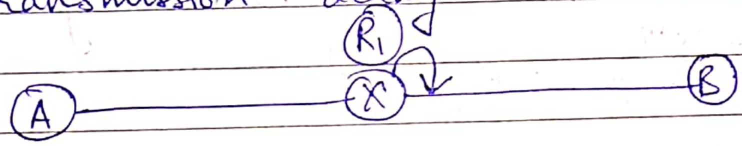
Page No.

Scanned by CamScanner

→ Queuing delay :

Based upon the traffic on the network, the time interval in which the router ~~stores~~ has stored the packet in the queue.

→ Transmission delay :



time required to forward the packet into the link.

$$\boxed{\begin{array}{l} \text{Transmission time} \\ (L/R \text{ seconds}) \end{array} = \frac{\text{Packet size } (L \text{ Bits})}{\text{Bandwidth (link capacity)} \quad (R \text{ Bit/s})}}$$

→ Propagation delay

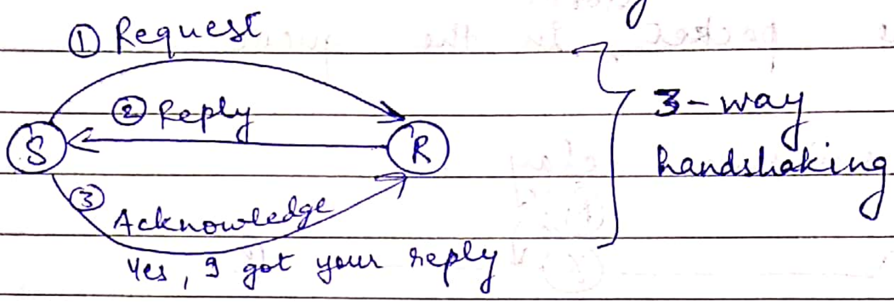
Time ~~required~~ taken by the packet to travel ~~fro~~ from the ~~bit~~ link to the destination.

$$\boxed{\text{Propagation delay} = \frac{\text{Distance}}{\text{Speed}}}$$

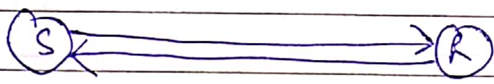
disadvantage : A bit lengthy ~~had~~ header ~~size~~ size.

→ TCP : Transmission control protocol

- Reliable
- Connection oriented Protocol
  - Handshaking



connection is established



Both can send the data simultaneously

→ UDP : User Datagram Protocol

- Connection less protocol
- No need of handshaking
- A header size is also less. (around 8 bytes)
- Unreliable but fast.

---

① domain name into IP address through DNS Query history or DNS query for the DNS server.

② Request is sent to the server

③ A message is generated through the server.

encrypted data = e.d.

Date ___ /___ /______

$\boxed{e.d.}$

(4) Application layer encrypts it (msg.).
(5) Transport layer appends transport layer header

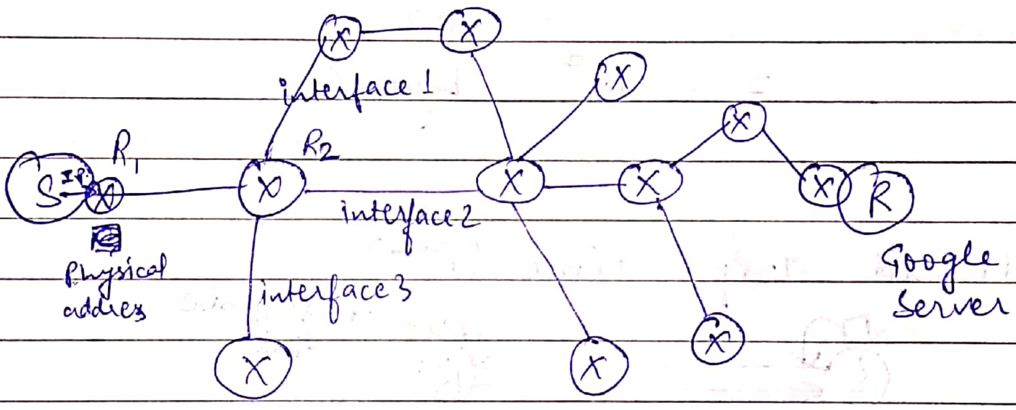
e.d.	t.h.

(6) Network layer appends network layer header.

e.d.	t.h.	n.h.

(7) Data link layer appends its header. (Physical address) or MAC address

e.d.	t.h.	n.h.	~~MAC~~ d.h.



R_2 uses I.P. address to find the network address ~~&~~ and ~~address~~. & ~~therefore~~ then forwards it to another interface. suppose R_3 and this process continues until packet reaches the destination:

This packet contains I.P. address of the destination

172.
192.
10.
} Private IP address.

Scanned by CamScanner

Date ___ /___ /______ Tut

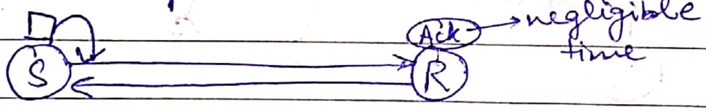
Q- 2 km length
Bandwidth ~~at~~ of propagation speed
& = transmission delay
100 byte.
2×10^8 m/s. = speed of signal.

$$\frac{100 \text{ byte}}{x} = \frac{\cancel{2000}\ 10^3 \text{ m}}{2 \times 10^8 \text{ m/s}}$$

$$\Rightarrow x = \frac{10^2 \times 10^8}{10^3}$$

$$= 10^7 \text{ byte/s}.$$

→ Round Trip time (RTT)



$$RTT = T_{trans} + T_{prop} + T_{prop}$$

$$\Rightarrow \boxed{RTT = T_{trans} + 2 * T_{prop}}$$

$$\Rightarrow \boxed{\text{for } T_{trans} \ll T_{prop}}$$

$$\boxed{RTT = 2 \times T_{prop}}$$

→ Bandwidth delay product :
= Bandwidth * delay (RTT) [bytes]

↔ At any time, how much data is present in the network is called bandwidth delay product.

Handshaking : Connection establishment

Q- Calculate the total time required to transfer a 1.5 MB file

RTT = 80 ms , Propagation delay = 40 ms.

Packet size = 1 KB

handshaking = 2 * RTT (before sending the data)

i- Suppose bandwidth = 10 Mbps and data packets can be sent continuously.

ii- bandwidth = 10 Mbps but after we finish sending each data packet we must (reached or not reached) wait for 2 * RTT before sending the next packet.

Time required?

Ans.

i- Transmission delay = $\frac{10^3 \text{ B}}{10 \text{ Mbps}}$ = 0.8 ms.

$= \frac{1 \times 10^3 \times 8}{10 \times 10^6}$

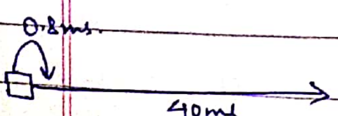
~~Total time = [(2 * RTT)] + [RTT * 1536] ms.~~

~~= [160] + [80] * 1536 ms.~~

~~= 160 * 120000 ms.~~

~~= 120160 ms.~~

Total time = [(Total no. of packets) * transmission delay]



+ Propagation delay.

+ Handshaking time.

= 1228.8 + 40 + 160 ms.

= 1428.8 ms.

Date ___/___/______

Saathi

iii - $2\left[(2 * RTT) + \frac{RTT}{2} + 0.8\right] * 1536$ ms.

$= 2\left[160 + 40 + 0.8\right] * 1536$

$= 2 \; 200.8 * 1536$

$= \cancel{308428.8 \text{ ms}}$

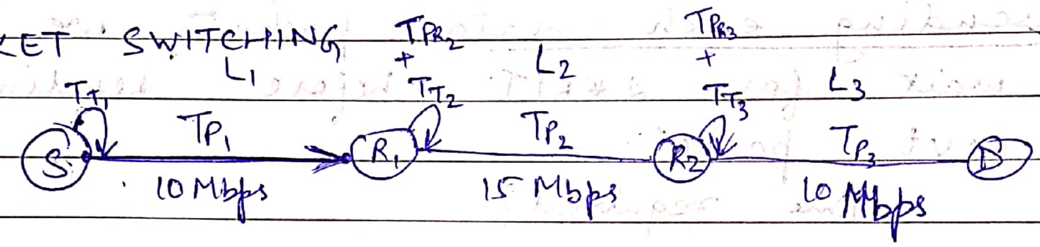
$= 308228$

iv - $[2 * RTT + 0.8] * 1536 + 40$

$= 160.8 * 1536 + 40$

$= 247028.8$ ms.

→ PACKET SWITCHING



* Do not calculate processing delay of sender.
- But calculate transmission delay of sender.

* The capacity with which data is to be sent is more than that to be recieved then there is no queuing delay.

→ Queuing delay = (For R_2)

$$\frac{\text{Packet size}}{\text{Bandwidth}(L_3)} - \frac{\text{Packet size}}{\text{Bandwidth}(L_2)}$$

T_{P_R} : Processing delay

T_P : Propagation delay.

→ If we send packets back to back, then only queuing delay occurs.

Scanned by CamScanner

Date __/__/__

→ Circuit Switching

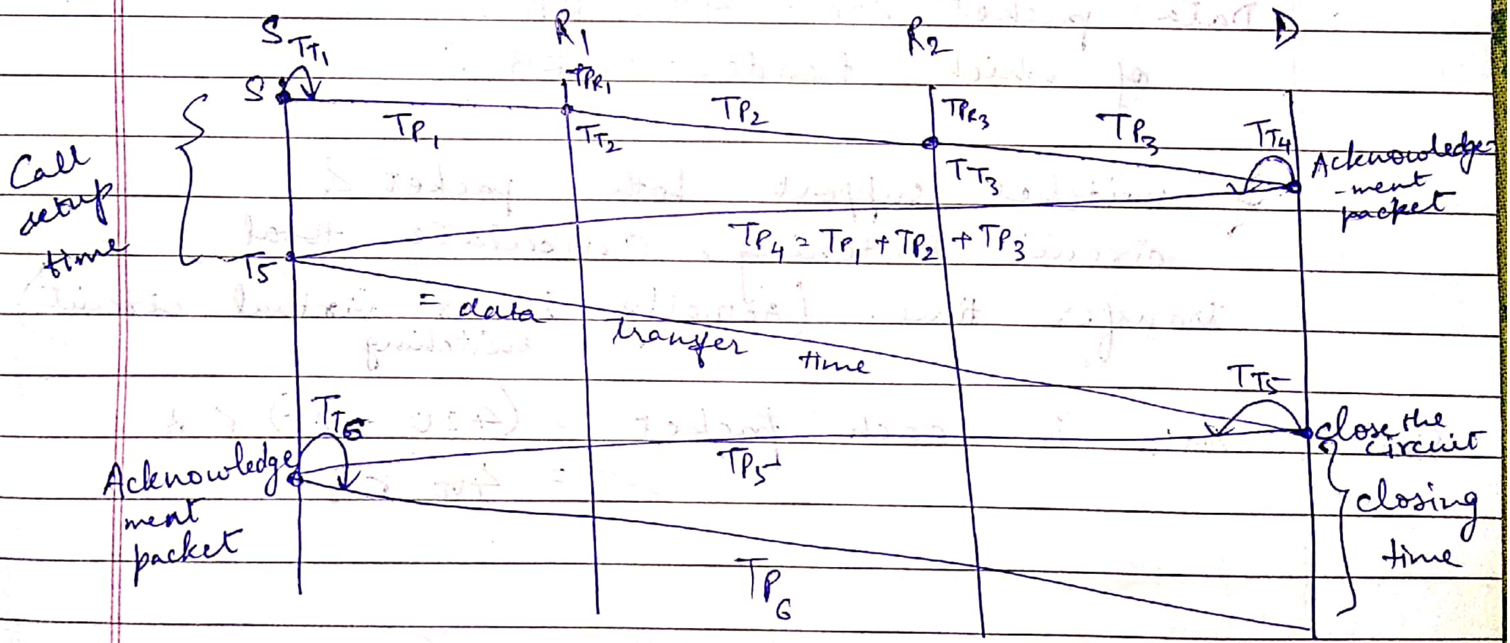
Virtual circuit
→ ~~Packet~~ Switching → actually a packet switching but connection is established just like ~~switching~~ circuit switching.

① — Establish a path (Call setup)

② — Data Transfer

③ — Closing of circuit

~~There are~~
☆ three packets that are used for connection establishment.



data transfer ~~to~~ do not have queuing delay.

NOTE: At call setup, queuing delay = 0

Circuit Switching → data transfer time
↓
bandwidth = min bandwidth of the links

Date ___ / ___ / ___

Q. Consider a file size of 1500 KB being sent from S to D along the path composed of S, D, 3 links and 2 store and forward switches. Consider L_1, L_2, L_3 having (d, R).

dist. (KM)

Bandwidth. (Mbps).

$L_1 = (10, 15)$
$L_2 = (15, 20)$
$L_3 = (15, 15)$

- Propagation speed = 2×10^8 m/s.
  $T_{Pr} = 1$ ms
- Call setup request & ack. packet size = 2 KB.
- Data packet size = 420 KB
  of which header = 20 KB.
- If switches support both packet & circuit switching, Calculate total transfer time. (Actually it is virtual circuit switching)

Sol-: data in each packet = (420 − 20) KB
= 400 KB

$$T_{t_1} = \frac{2 \times 1000 \times 8}{15 \times 10^6} = \cancel{1.06}\ 1066.6\ \mu s.$$

$$T_{p_1} = \frac{10 \times 10^3}{2 \times 10^8} = 50\ \mu s.$$

Processing delay (R₁) = 1 ms.

Date __/__/____

$$T_{t_2} = \frac{2 \times 1000 \times 8}{\cancel{20}\ 10 \times 10^6} = 800\ \mu s.$$

$$T_{p_2} = \frac{15 \times 10^3}{2 \times 10^8} = 75\ \mu s.$$

Processing delay (R_2) = 1 ms.

$$T_{t_3} = \frac{2 \times 1000 \times 8}{15 \times 10^6} = 1066.6\ \mu s.$$

$$T_{p_3} = \frac{15 \times 10^3}{2 \times 10^8} = 75\ \mu s.$$

$$T_{p_4} = T_{p_1} + T_{p_2} + T_{p_3} = 200\ \mu s.$$

$$T_{t_4} = \frac{2 \times 1000 \times 8}{15 \times 10^6} = 1066.6\ \mu s.$$

Call setup time = $(T_{t_1} + T_{t_2} + T_{t_3} + T_{t_4}) + (T_{p_1} + T_{p_2} + T_{p_3} + T_{p_4}) + 2ms.$

$= 400\ \mu s + 1066.6 + 1066.6 + 800 + 1066.6\ \mu s + 2ms.$

$= 400\ \mu s + 3999.8 + 2ms.$

$= 4399.8\ \mu s + 2ms.$

$= 6399.8\ \mu s.$

Circuit switching :

data transfer time $= \frac{\cancel{1500}\ 100 \times 1000 \times 8}{\cancel{15} \times 10^6} + 200\ \mu s.$

$= 800.2 ms.$

Page No.

Total data transfer time = 800.2 + 6.4 ms.
≃ 806.6 ms.

Saathi

Date ___/___/___

→ Virtual circuit switching :

4 packets : $\left(\begin{Bmatrix}400\\+20\ \text{header}\end{Bmatrix} + \begin{Bmatrix}400\\+\ \text{header}\end{Bmatrix} + \begin{Bmatrix}400\\+\ \text{header}\end{Bmatrix} + \begin{Bmatrix}300\\+\ \text{header}\end{Bmatrix}\right)$

$= \{420, 420, 420, 320\}$

$= \frac{3 \times 420 \times 1000 \times 8}{15 \times 10^{6}} + \frac{320 \times 1000 \times 8}{15 \times 10^{6}}$

data transfer time = 842.67 ms. + 200 μs.

= 842.87 ms.

∴ Total time = 849.27 ms.

→ Packet switching :

transmission time $= \frac{3 \times 420 \times 1000 \times 8}{15 \times 10^{6}} + \frac{320 \times 1000 \times 8}{15 \times 10^{6}}$

= 842.67 ms.

$T_{P_1} = 50$ μs

$T_{P_{R_1}} = 1$ ms.

$T_{t_{L_1}} = \frac{420 \times 1000 \times 8}{20 \times 10^{6}} = 168$ ms.

$T_{P_2} = 75$ μs

Date ___/___/_____

$T_{P_{R_2}} = 1$ ms.

$T_{T_{L_3}} = \frac{420 \times 1000 \times 8}{15 \times 10^6} \therefore = 224$ ms.

$T_{Queuing} = \frac{420 \times 10^3 \times 8}{10^6} \left[\frac{1}{15} - \frac{1}{20} \right]$

$\frac{4 - 3}{60}$

$= \frac{420 \times 10^3 \times 8}{60 \times 10^6}$

$= 56$ ms.

$T_{P_3} = 75\ \mu s$

Tut

Q- Consider sending a large file of f bits from host A to host B. There are 3 links b/w A & B with two ~~suitable~~ switches & the links are uncongested. Host A segments the file into segment of 'S' bits ~~beats~~ and adds 80 bits of header to each segment ($L = 80 + S$ bits). Each link has transmission rate of r bits/s. Find value of 'S', that minimizes the delay of moving the file from A to B. Disregard propagation delay.

(no processing delay is given ⇒ disregard
& no queuing delay is given

50 packets $A \overset{3\,ms}{} \quad L_1 \quad S_1 \quad L_2 \quad S_2 \quad L_3 \quad B$

Date ___/___/______

Packet size = $80 + S$ bits

Bandwidth = r bits/s

No. of packets = f/S

$\Rightarrow$ $\cancel{T_T = 3 * \left[\frac{80+S}{r}\right] = \left[\frac{240+3S}{r}\right]}$

$\cancel{\frac{dT_T}{dS} = \frac{3\,d(S/r)}{dS} = \frac{3}{r}}$

$\Rightarrow$ Total time $= 3 * \left(\frac{80+S}{r}\right) + \left(\frac{f}{S} - 1\right)\left(\frac{80+S}{r}\right)$

$= \left(\frac{3S + 240}{r}\right) + \left(\frac{f}{S} - 1\right)\left(\frac{80+S}{r}\right)$

$\Rightarrow \frac{dT}{dS} = \frac{3}{r} + \left[\frac{-f}{S^2}\right]\left[\frac{80+S}{r}\right] + \frac{f}{Sr} - 1 = 0$

$\Rightarrow 2 - \frac{f}{S^2}(80+S) + \frac{f}{S} = 0$

$\Rightarrow \frac{2S^2 - f(S+80) + fS}{S^2} = 0$

$\Rightarrow S \neq 0$

$\Rightarrow 2S^2 - f(S+80) + fS = 0$

$\Rightarrow 2S^2 - 80f = 0$

$\Rightarrow 3S^2 = 80f$

$\Rightarrow$ ~~$S = 80f$~~

Scanned by CamScanner

Date ___/___/_____

$$\Rightarrow S = \sqrt{\frac{\cancel{2\times2}\times\cancel{2\times2}\times5}{3}f}$$

$$= 4\sqrt{\frac{5}{3}f}$$

$$T = \frac{3S}{r} + \frac{3\times80}{r} + \left(\frac{f}{S}-1\right)\left(\frac{S+80}{r}\right)$$

$$\frac{dT}{dS} = \frac{3}{r} + \left(-\frac{f}{S^2}\right)\left(\frac{S+80}{r}\right) + \frac{1}{r}\left(\frac{f}{S}-1\right) = 0$$

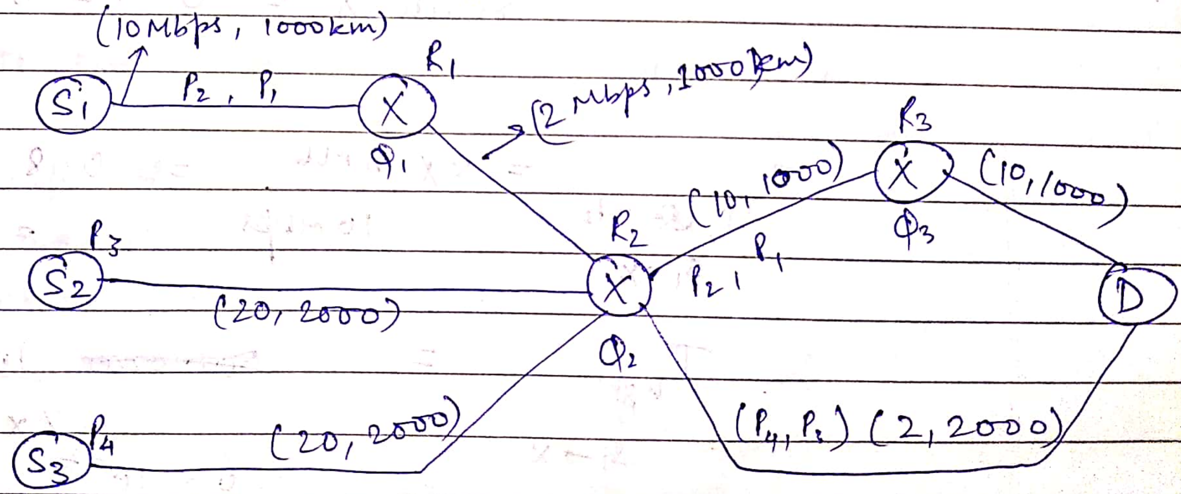
$$\Rightarrow 3 - \left[\frac{fS+80f}{S^2}\right] + \left(\frac{f}{S}-1\right) = 0$$

$$\Rightarrow 2S^2 - fS - 80f + fS = 0$$

$$\Rightarrow 2S^2 = 80f$$

$$\Rightarrow S = \sqrt{40f}$$

Ques –



Signal speed $= 2\times10^8$ m/s, Packet size $=1000$ KB, ignore processing delay. ~~At what time~~ all packets will reach to destination D.

Page No.

Date ___/___/______

$$Q.D = \text{Packet size}\left[\frac{1}{BW_{L_3}} - \frac{1}{BW_{L_2}}\right]$$

$$\text{Packet size} = \frac{1000 \times 8}{1000} = 8 \text{ Mb}.$$

$$T_{T_{P_3,P_4} \text{ at } S_2, S_3} = \frac{\cancel{2}^{4}\,\cancel{8} \text{ Mb}}{\cancel{20}_{\cancel{10}_{5}} \text{ Mbps}} = 0.4 \text{ s}.$$

~~Queuing delay~~ P_3, P_4 ~~= 16 Mb [...]~~

$$T_{\text{Propagation } P_3, P_4 \text{ at } S_2, S_3} = \frac{\cancel{2000000} \times 10^6 \text{ m}}{2 \times 10^8 \text{ m/s}}$$

$$= 10^{-2} \text{ s}.$$

$$= 0.01 \text{ s}.$$

2) Total time ~~at~~ for P_3, P_4 to reach X = 0.41 s.

$$T_{T_{P_1} + P_2,\ S_1 \to X} = 2 \times \frac{8 \text{ Mb}}{10 \text{ Mbps}} = 0.8 \text{ s} \times 2$$

$$= \cancel{0.4}\ 1.6 \text{ s}.$$

$$T_{\text{prop } P_1,\ S_1 \to X} = \cancel{1000000}\ \frac{10^6}{2 \times 10^8} \text{ s}.$$

$$= \frac{10 \times 10^{-3}}{2} = 5 \times 10^{-3} \text{ s}$$

$$= 0.005 \text{ s}.$$

Scanned by CamScanner

Date __/__/__ for P_1 & P_2

$\therefore$ Total time to reach x = 1.605 s

Queuing delay at R_1 for P_1 & P_2 $= 2 \times 8 \left[\frac{1}{2} - \frac{1}{10}\right]$ s.

$= 2 \times 8 \left[\frac{5-1}{10}\right]$ s.

$= 2 \times \frac{8 \times \cancel{4}2}{\cancel{10}5} = 3.2$ s $\times 2$

$= 6.4$ s.

Transmission delay

Port no. 80 (standard port)

→ HTTP (Hypertext Transmission Protocol)

☆ First of all we need to download base file.

(HTTP)

→ State less vs. connection less

- doesn't store any info. of any request. (what request was done from a particular user.)

- no connection is established.
- no handshaking.

no parallelism parallel without pipelining with pipelining

Date

→ non-persistant HTTP vs persistant HTTP

- at most one obj. can be sent over a TCP connection, then connection is closed.

multiple objects can be sent over one TCP connection, ~~then~~ between client & server.

1- Base ~~file~~ file is requested
2- ~~Base file is requested~~ Connection is established
3- ~~Server sends base file~~
4-
5- HTML file i.e- base file is passed (read)

Q- If TCP uses 3-way handshaking, then why do we uses only 1 RTT for request, response, acknowledgment?

→ Non-persistant like

- No parallism
  We can't establish parallel connections.

Parallel

After recieving the base file we can get parallel connection. (In 2 RTT multiple objects can come simultaneously.)

Non-pipelining

each obj. has 1 RTT

Pipelining

1 RTT for all the obj's.

cr : carriage return

Date ___ /___ /______

→ If not ~~written~~ written, non-persistant condition with / without pipelining, then we would consider with pipelining.

- Keep Alive : A type of times which gives us timeout for the connection.

- Head : When we only need the header info. then it is used.

- Put : method used to upload data.

- 200 OK : status code for successful request and response.

- 301 Moved permanently : requested obj. is moved to another server.

- 400 Bad Request : request msg. is not understo-od by server.

- 404 Not Found : requested document not found on this server.

→ Proxy server : satisfies client request without involving origin server
hotstar

→ Client - Server : Connections are intermittent, temporary

Page No.

Saathi

Date __ / __ / ____

Ques- ⇒ Suppose within your web browser, you click on a link to obtain a webpage. The IP address for the associated URL is not cached in your localhost, so a DNS lookup is necessary to obtain the IP address. Suppose that n-DNS servers are visited with time $RTT_1 + RTT_2 + \ldots + RTT_n$

Suppose that webpage associated with the link contains exactly one obj. Consisting of a small amount of HTML text. Let RTT_0 denote the RTT b/w localhost & server. Assuming 0 transmission time of the obj. How much time elapsed from when the client clicks on the link until the client recieves ~~the~~ the obj.

$$= \sum_{i=1}^{n} RTT_i + 2RTT_0$$

Ques- In above problem, suppose HTML file references 8 very small objs. on the same server. Neglecting Transmission time how much time elapsed with:

i- Non-persistant HTTP with no parallel TCP connection.

ii- Non-persistant HTTP with the browser configured for 5 parallel connections.

iii- Persistant HTTP.

i- DNS + $18RTT_0$

ii- DNS + $6RTT_0$

iii- DNS + $3RTT_0$: ~~basefile~~ connection, basefile, all objs

Page No.

Non-persistant = Base file (Connection setup + basefile) + DNS Query + objects (2* No. of connections)

Saathi

Date ___/___/___

Persistant = Basefile + DNS Query + (No of objects)

NOTE: if pipeline is there, for all obj. then only 1 RTT, would be needed instead of total no. of obj.

→ Summary of HTTP:

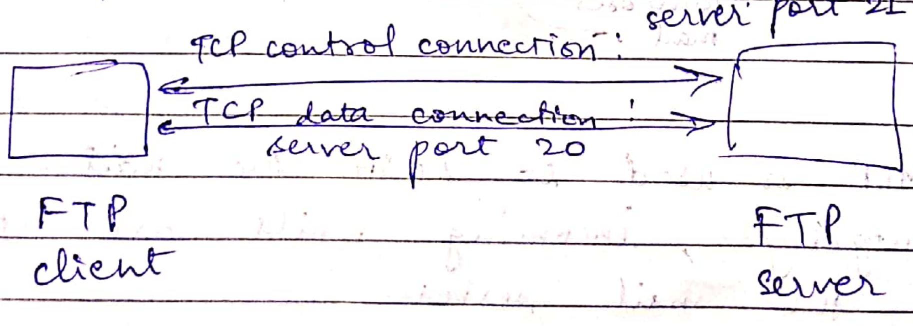
- TCP/IP connection
- Stateless protocol
- Port No. 80
- Persistant , Non-persistant

Persistant: ~~Parallel~~ Pipeline (back to back), Non-pipeline

Non-persistant: Parallel, Non-parallel

* File Transfer Protocol

- ftp server : port 21
- ftp : RFC 959



- For each & every file, you need a different data connection, but control connection is same/permanent.

Page No. ___

Date __ / __ / ____

- control connection : "out of band" because it is not used to transfer data.

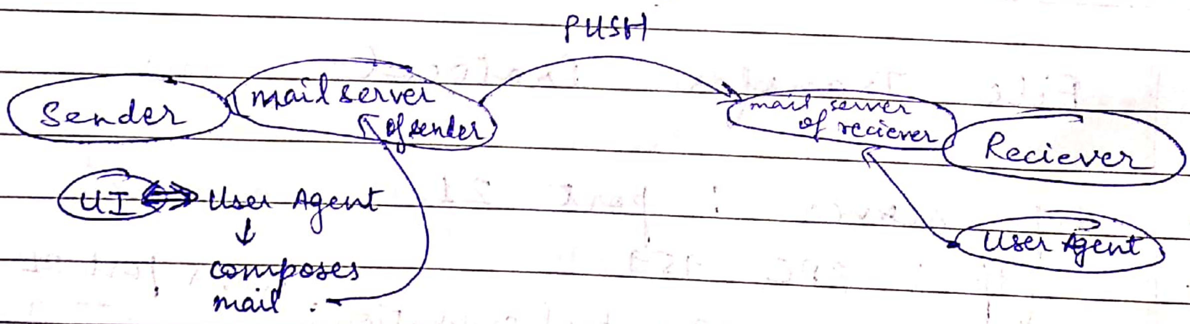
- FTP server maintains "state" : (Stores info.) 'current directory'.

→ Summary of TCP/IP :

- client - server model
- 2 TCP connections - control (permanent), data
- each & every file requires different data connection.
- Port No. 21 (control connection)

★ SMTP : ~~Simple~~ ~~[illegible]~~ Simple Mail Transfer Protocol

→ Mechanism of mail exchange :



→ SMTP is used to PUSH the mail.

→ outgoing, incoming mails are stored in the mail server.

→ works b/w mail server.

→ mail server : mailbox or storage dedicated to you.

→ uses TCP, port ~~[illegible]~~ 25.
RFC 2821.

TCP - reliable (ack packet comes) whereas UDP does not send any ack packet.

RFC : Request for comment

- 3 phases - i- handshaking
  ii- transfer of messages
  iii- closure.

- interaction : commands - ASCII text (not secure)
  response - status code & phrase

Q- Consider a short 10 m. link over which a sender can transmit at a rate of 150 bits/sec. Suppose sent packets containing data are 1 lakh bits long and packet containing only control are only 200 bits long.

Assume that 10 parallel connections. Now consider at the HTTP protocol. Suppose that each ~~domain~~ obj. is 100 K bits and that the initial downloaded obj. contains 10 reference objs. from the same sender. Determine delay for -

i- non-persistant (parallel)

ii- Persistant

i- ~~Control~~ length = 10 m.
rate = 150 bits/sec.
sent packets = ~~100000~~ 10^5 bits.
control packets = ~~40~~ 200 bits.

~~Non-persistant (parallel) :~~
~~2 RTT (initial downloaded obj.) + 2RTT (data packets)~~

Scanned by CamScanner

Date ___ / ___ / ____

connection establishment + data transfer time
(control packets) (data packets)

conn. establishment

SYN SYNACK ACK base file

$$\left[\left(\frac{200}{150} + \frac{200}{150} + \frac{200}{150}\right)\right] + \left(\frac{100000}{150}\right)$$

$$+ \left(\frac{200}{150/10} + \frac{200}{150/10} + \frac{200}{150/10}\right) + \left(\frac{100,000}{150/10}\right)$$

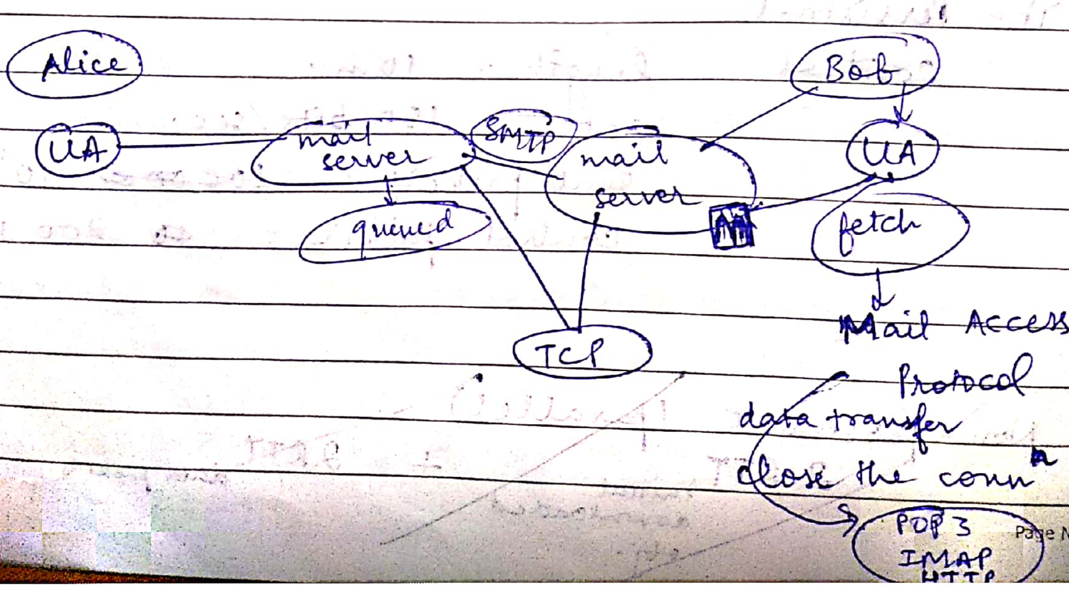
10 reference objs data packets

ii - Persistant

SYN SYNACK ACK Basefile

$$\left(\frac{200}{150} + \frac{200}{150} + \frac{200}{150}\right) + \left(\frac{100000}{150}\right)$$

$$+ 10 \times \left(\frac{100,000}{150}\right)$$



Page No.

CR : carriage return
LF : line field

Date ___/___/___

Saathi

NOTE : HTTP & SMTP interaction, both occurs in ASCII thus, we can see and understand the interaction.

→ SMTP – • persistant
• head & body in 7-bit ASCII
• CRLF . CRLF ~~to~~ is used to end connection

* Mail – Access Protocol

1. POP3 : Post office Protocol 3
2. IMAP : Internet Mail Access Protocol.
3. HTTP : Hypertext Transmission Protocol.

↳ organized at the server itself
↳ download then organize at personal level

1. POP3 :

1- Authorization phase
2- Transaction phase
3- Updation phase

→ Authorization phase commands :
client : ~~[illegible]~~ username, password
server : responses +OK , –ERR

Saathi

Date ___/___/___

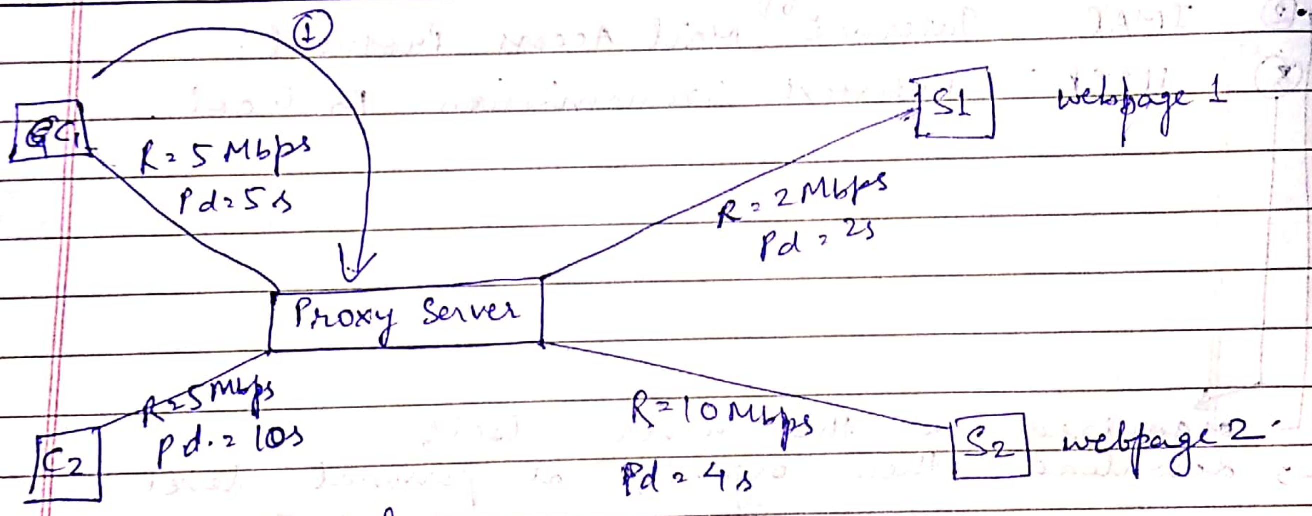
(2) Transaction phase :
list , retr , dele , quit

Modes of POP 3

NOTE : (1) download and delete mode :
does not allow re-read.

(2) download and keep :
opening mail in other comp. , we do not get it organized , but it allows re-read.

Q-



Pd : Prop. delay

In the network dig. given , at time t = 0
C_1 wants to connect to S_1 and C_2 ~~wand~~
wants to connect to S_2.
S_1 : Persistant HTTP connⁿ — non-pipelining
S_2 : Non-persistant HTTP connⁿ

C_1 requests for webpage 1
& C_2 " " " 2 .

W1 : consists of 5 objs .
W2 : Consists of 3 objs .

Page No.

Non-pipelining : a packet is sent first then another packet is sent after first ~~one~~ reaches.

Date ___/___/___

Saathi

each obj size = 10^7 bits.
Initially proxy server is empty. It gets refreshed after 250 s. Assume T.D. & P.D. only.

i- Calculate the time required after which C_1 & C_2 will recieve their respective webpages.

Sol. →

~~C_1 total delay = $\left(\frac{10^7 \times 5}{5 \times 10^6} s + 5 \times 5\right) s.$~~

~~= 250 s.~~

~~C_2 total delay = 3~~

C_1 total delay = $\left(\frac{10^7 \times 8}{10^6 \times 8} + 5 \times 5\right) s.$

= 35 s.

~~C_2 total delay~~

C_1 : request proxy server ⟶ proxy server establishes conn^n requests base file fetches objs.

Page No.

Date ___/___/_____

$$2.5 \quad \frac{5\times10^7}{2\times10^6} + 5\times(2\times2)$$

$$C_1: \boxed{5s + \cancel{20}\ 2\times(2\times2) + \cancel{28}\ \frac{5\times10^7}{2\times10^6}} + \frac{5\times10^7}{5\times10^6} + 5$$

$$= \cancel{44}\ \cancel{98} \quad 13 + 25 + 20 + 15$$

$$= 73\ s.$$

~~delay~~

$$\cancel{C_2: 10s + (2\times4)\times3 + 3\times\left[10 + \frac{10^7}{10^6}\right]}$$

$$\cancel{= 10s + 24 + 60\ s}$$

$$\cancel{= 102\ s}$$

$$C_2: 10s + (2\times8) + (2\times8)\times3 + 3\times\left[\frac{10^7}{10^7} + 4\right]$$

$$+ 3\times\frac{10^7}{5\times10^8} + 10$$

$$= \underbrace{10}_{C_2\to\text{Proxy server req.}} + \underbrace{16}_{\text{Base file}} + \underbrace{48}_{\text{Conn}^n} + \underbrace{15}_{\text{Obj. from } S_2\to P.S} + \underbrace{6 + 10}_{\text{Obj from } P.S\to C_2}\ s.$$

$$\cancel{73} = 105\ s.$$

Date __/__/____

ii- At time $t = 50$ s. , C_1 demands for W2. Calculate the time after which C_1 will get the webpage 2.

iii - At time $t = 260$ s. , C_2 demands for W1. Calculate the time after which C_2 will get W2.

→ Domain Name System :

- IP address (32-bit) : used for addressing ~~datagram~~ datagrams
- uses UDP so to reduce response time.

Services :

1- hostname to IP address translation
2 - host aliasing
   - canonical, alias names

3- mail server aliasing.
4- load distribution
   - replicated web servers :
     many IP addresses correspond to one name
     which server to choose from a pool of different servers (duplicate servers) is decided by DNS.
     (the company of which the server is, also has its own load balancing technique.)

Scanned by CamScanner

Date ___/___/___

→ why not centralized DNS ?

- single point of failure
  e.g. if USA has centralized DNS, what if failure happens, whole internet of the world would fail.
- traffic volume.
- distance centralized database

→ Distributed, hierarchial database

Root server

|

Top level Domain server (7 in the world)

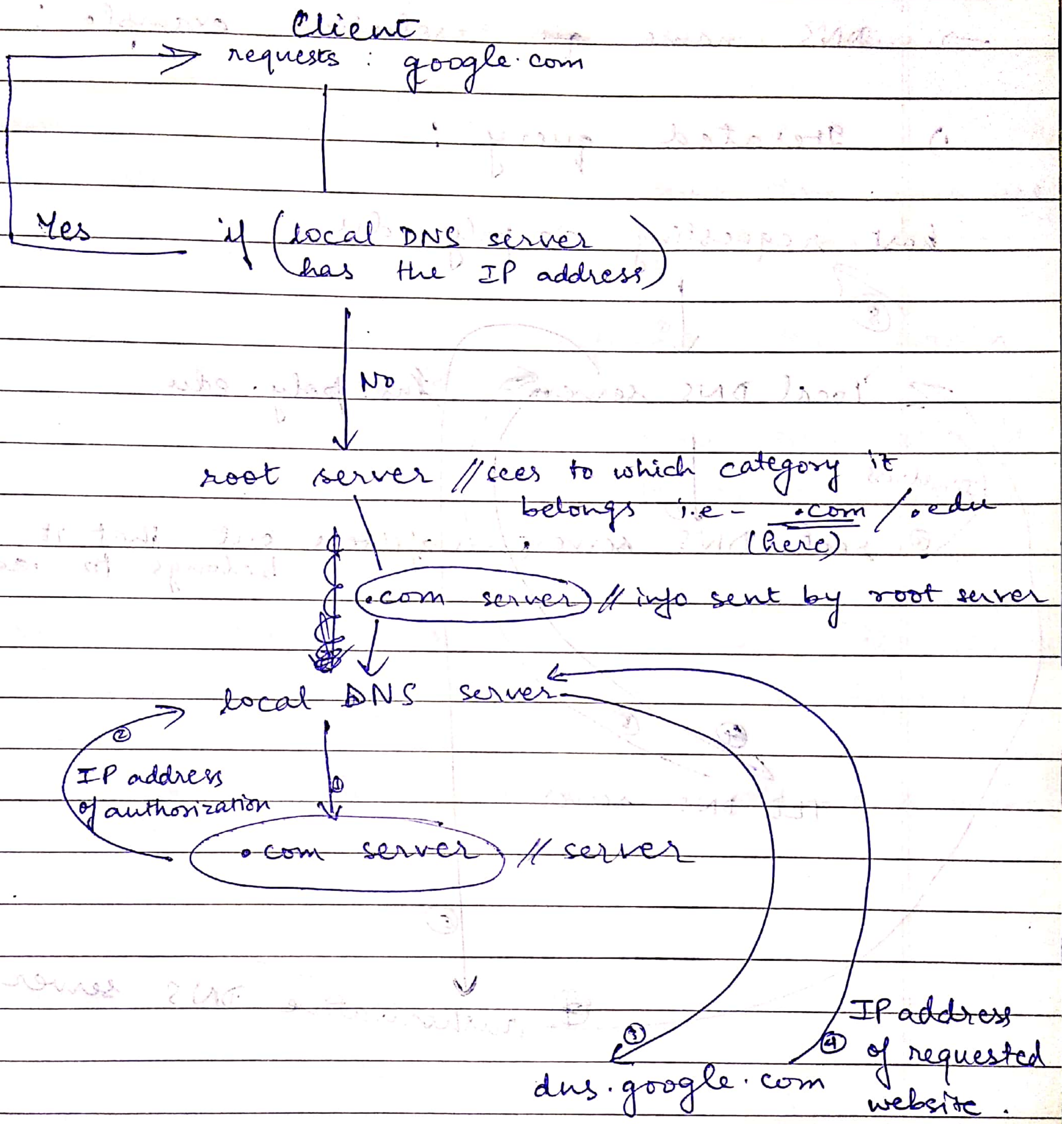
|

Authoritative server
e.g - yahoo.com DNS server / youtube.com DNS server

|

Queries

Date __/__/__ flow chart

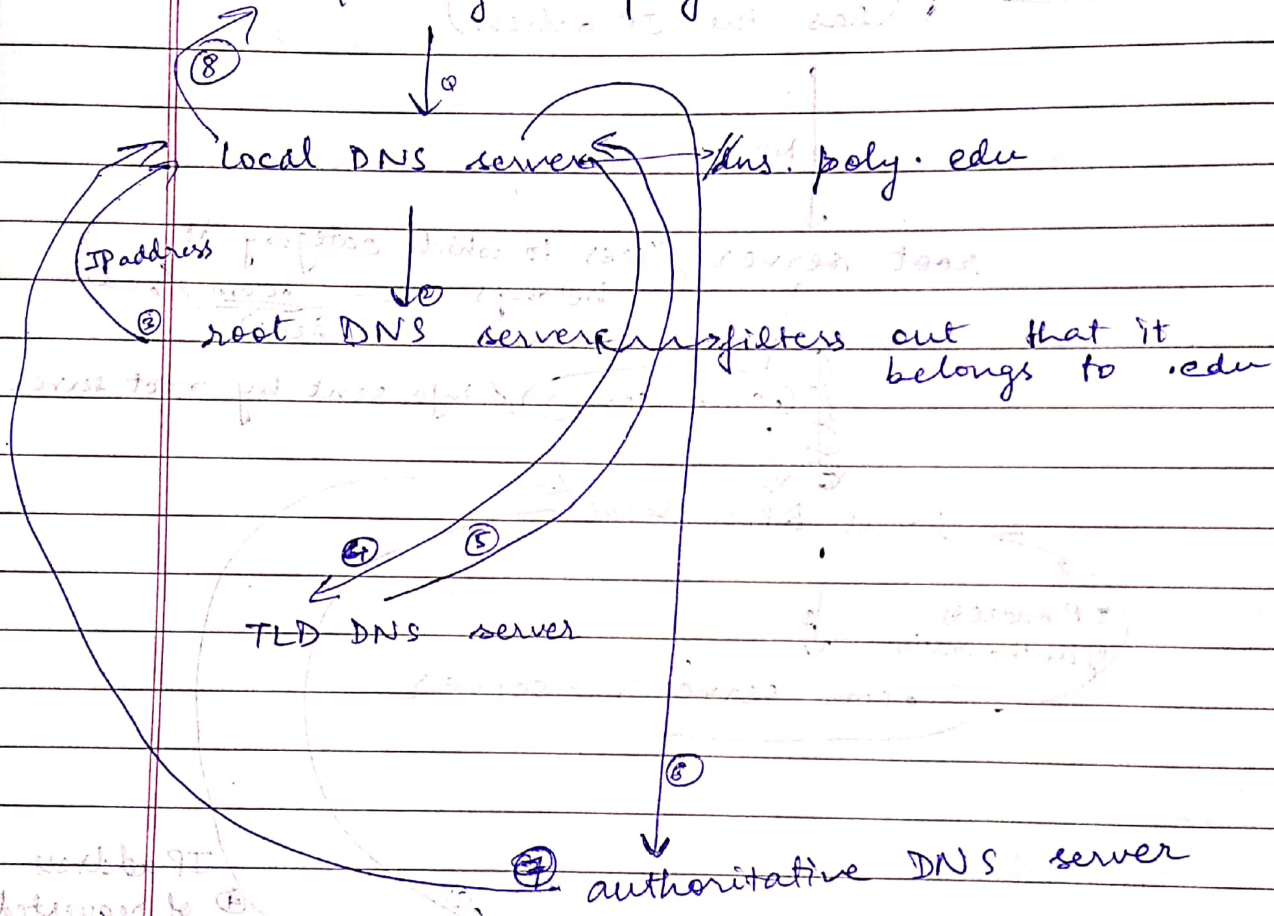


ISP : Internet Service Provider.

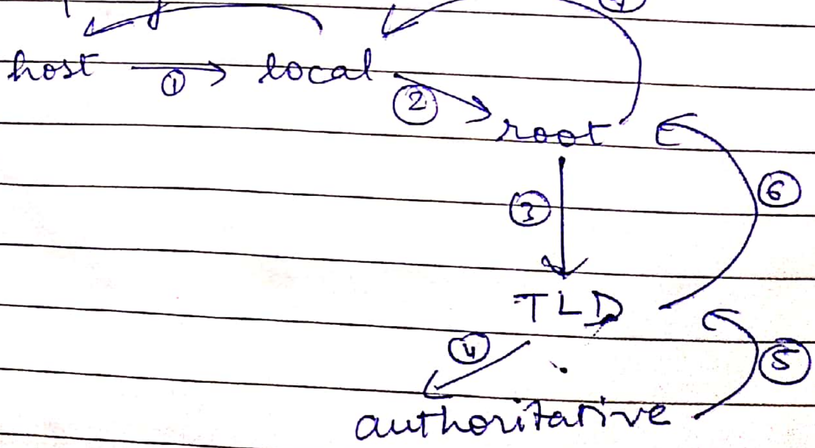
→ DNS name ~~qu~~ resolution example :

1) Iterated query :

host requesting cis.poly.edu



② recursive query



Date ___ /___ /______

NOTE ! - Iterated query method :
local DNS server iterates queries

~~Rec~~

TTL : Time to live : how much time you are going to store the info in the cache.

→ In case if cache has expired, then again you need to query.
- suppose, if ans is in local DNS server's ~~a~~ cache, it answers from its cache.

- If the query ans. is in cache at any level, it gets answered by there only, no further ~~also~~ query required.
- If not then further queries are done.