Name : Gulshan Shriram Hatzade

Roll No. CS24MTECH14006

(1)

Q.1.) a) (A) — (X) — (X) ...... (X) n (B)

$t_f = \frac{L}{R}$, $t_p = \frac{nm}{s}$, $t_{ack} = \frac{A}{R}$ bits/s

a) (i) SAW :-

As we have to wait for ack, we will calculate for 1 packet & add same for p packets.

$$\text{Total time} = P\left[2N\frac{m}{S} + N\frac{L}{R} + \frac{NA}{R}\right]$$

$$= PN\left[\frac{2M}{S} + \frac{L}{R} + \frac{A}{R}\right] \text{ sec}$$

(b) BTB :-

We have to send p packets continuously without waiting for ack.

$$\text{Total time} = \frac{pL}{R} + \frac{2NM}{S} + \frac{NA}{R} + (N-1)\frac{L}{R} \text{ sec}$$

Q. 1) b) (1) SAW :-

utilization, $\mu_{SAW} = \dfrac{\frac{L}{R}}{PN\left(2\frac{L}{R} + \frac{2m}{S} + \frac{A}{R}\right)}$

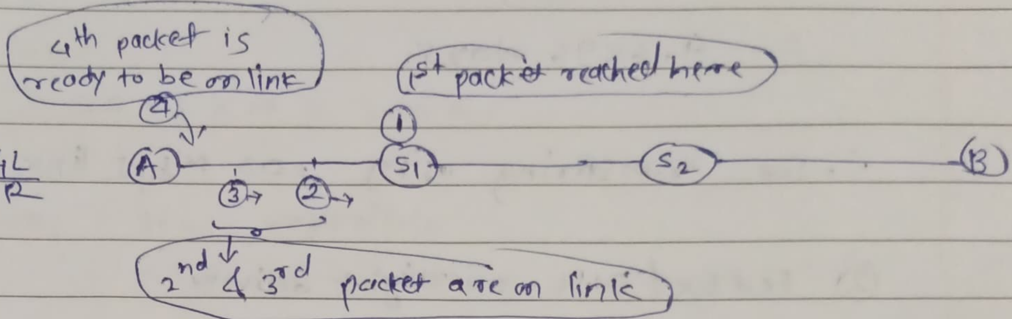
(2) BTB :-

utilization, $\mu_{BTB} = \dfrac{P \cdot \frac{L}{R}}{\frac{PL}{R} + \frac{2Nm}{S} + \frac{NA}{R} + (N-1)\frac{L}{R}}$

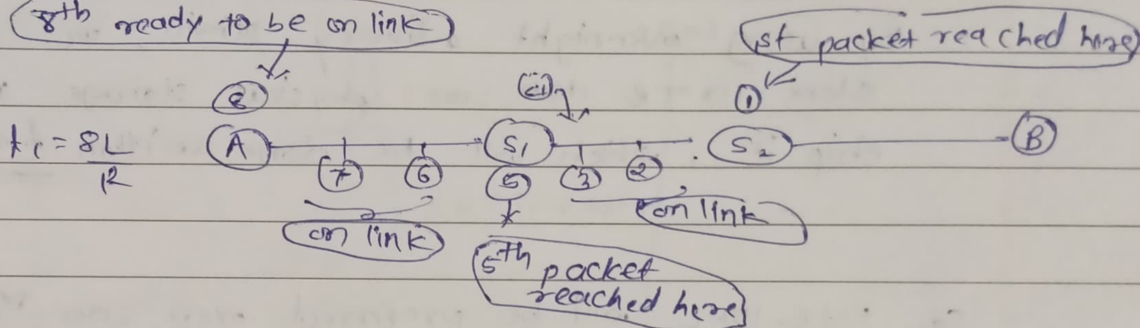
$\mu_{BTB} = \dfrac{PL}{PL + 2RNm + NA + (N-1)L}$

Q.1) c)

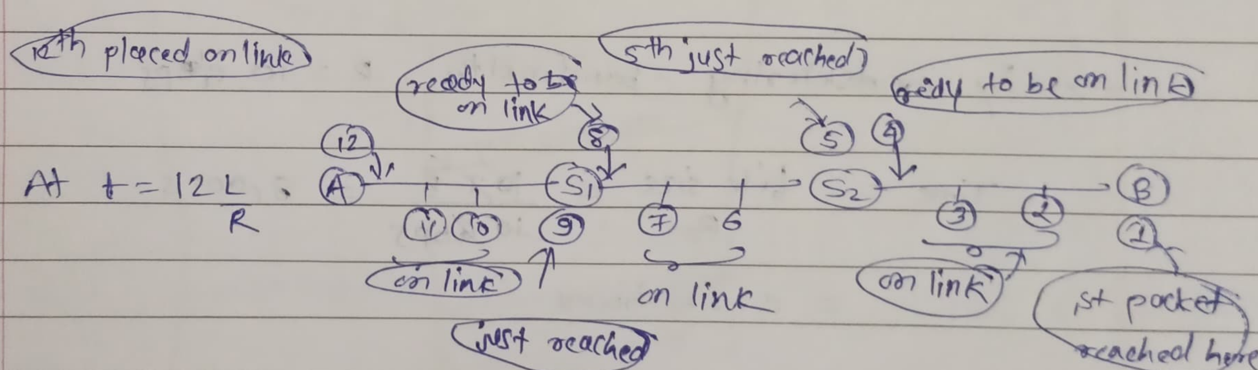
At $t_1 = \frac{4L}{R}$



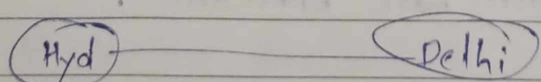
At $t_1 = \frac{8L}{R}$



At $t = \frac{12L}{R}$



Q.2)



For transmitting data from Hyd to Delhi,

(i) using 200 Mbps link,

$$\text{Time} = \frac{\text{file size}}{R} = \frac{10\,TB}{200\,Mbps} = \frac{10\,M \times 8}{200} = 4 \times 10^5\ s$$

$$= \frac{400{,}000}{3600}\ hr$$

= 111.11 hours

= 4.6296 days

∴ For transfering using 200 Mbps link, take 4.62 days.

(2) Fed Ex / DHL overnight delivery

By using overnight delivery service, we can store 10 TB data on physical storage media & ship it, which will be done within 1 day.

So, Fed Ex / DHL will be preferred over 200 Mbps link.

→ By condsidering bandwidth, R = 10 Gbps

$$\text{Time} = \frac{\text{file size}}{R} = \frac{\cancel{10}\,\cancel{T}^{K}B}{\cancel{10}\,\cancel{G}bps} = 8{,}000\ s$$

$$= 2.22 \text{ hours}$$

For internal transfer between servers within IIT H, the network is much more efficient. internal data centre

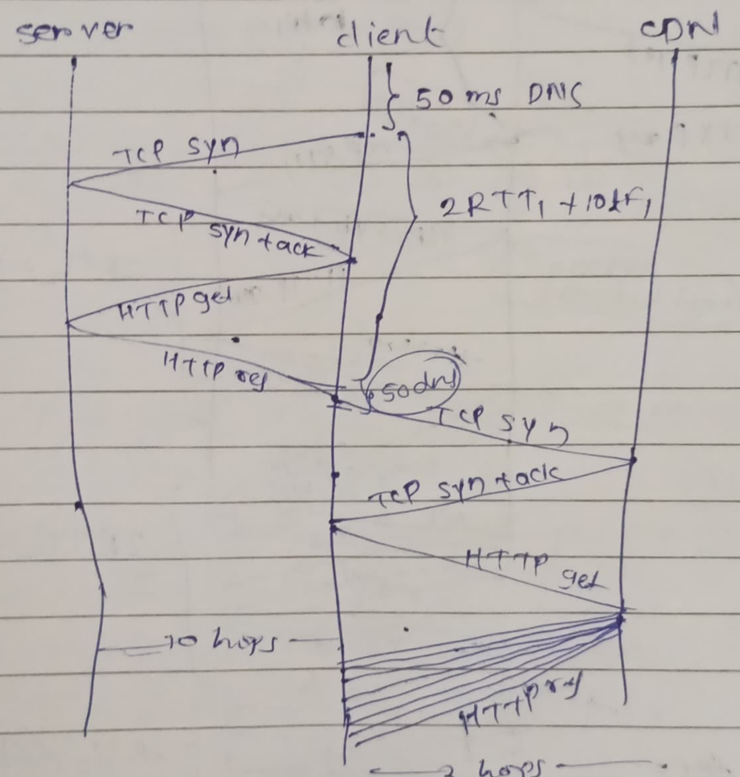
3) Solution a)

$t_{f_1} = 1\ ms$

$RTT_1 = 20\ ms \times 10 = 200\ ms$

For CDN, t_{f_2} = negligible

$RTT_2 = 20\ ms \times 3 = 60\ ms$



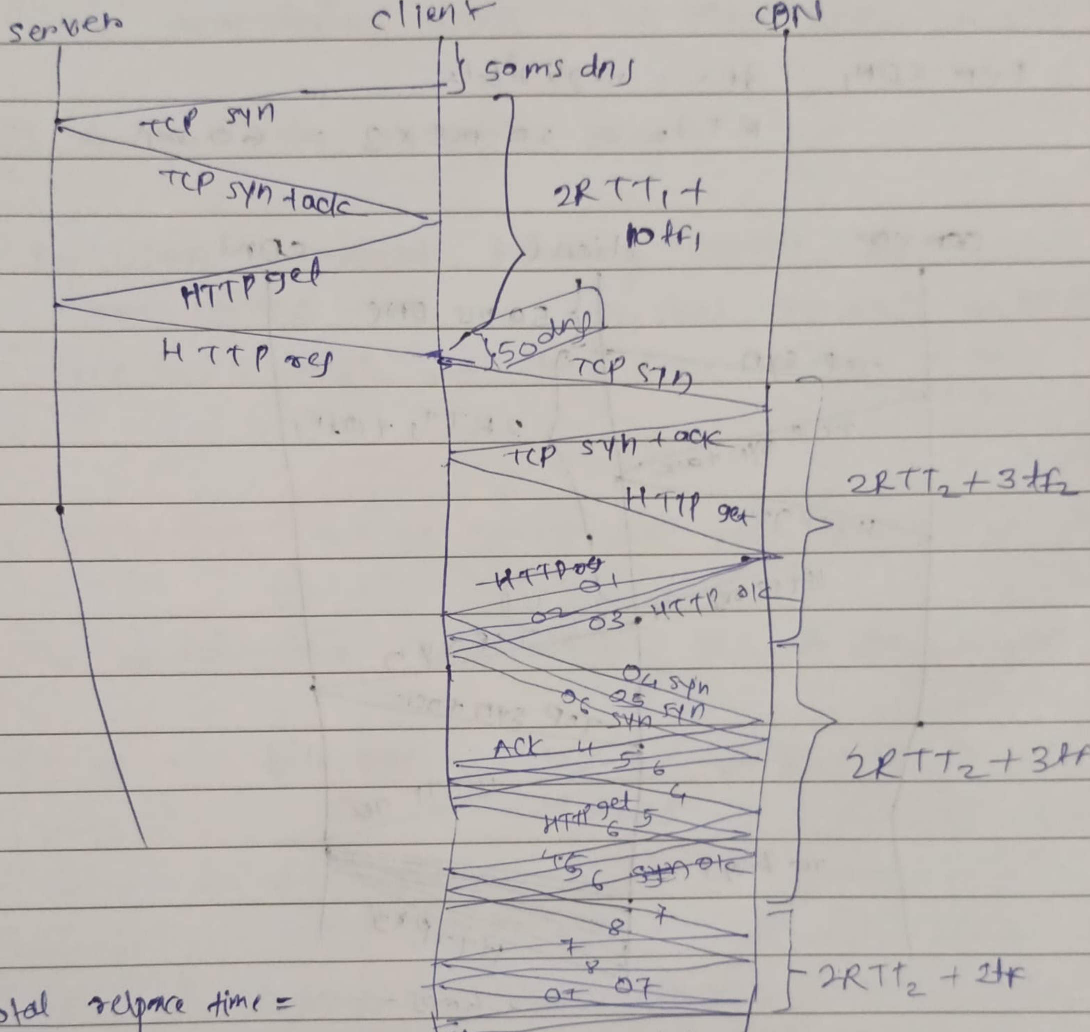
Total responce time will be

$= \underbrace{50}_{dns} + \underbrace{(2RTT_1 + 10t_{f_1})}_{\text{establish connection to serve}} + \underbrace{(1RTT_2)}_{\text{establish conn}^n \text{ to CDN}} + \underbrace{(1RTT_2)}_{\text{accessing 8 objects from CDN}} + \underbrace{(8t_f)}_{8\ obj}$

$= (50 + 400 + 10) + 50 + (120 + 10)$

$= 640\ ms$

b) HTTP/1.0 : Non persistent with 3 parallel conn^n :



Total relponce time =

$$\underbrace{50}_{dns_1} + \underbrace{(2RTT_1 + 10tf_1)}_{\text{establish connect}^n \text{ to server}} + \underbrace{50}_{dns_2} + (2RTT_2 + 3tf) \times 2 + (2RTT_2 + 2tf + 2t$$

$= 50 + (400 + 10) + 50 + 2(120 + 3) + (120 + 4)$

$= 880$ ms

classmate

Date

Page

7

c) DNS resolution time is not required here, as it is cached.

$RTT_{eff} = 20ms \times 2 = 40\ ms$

Total responce time $= \underbrace{(1\ RTT_{eff})}_{\text{TCP connection}} + \underbrace{8\ tf}_{\text{8 objects}} + \underbrace{(1\ RTT_{eff})}_{\text{fetch HTML}}$

Total Responce time = Time for HTML file + Time for objects data

$= 40 + 42 + 40 + 7$

$= 131\ ms$

Q.4) $N = a \times d$, transmission delay $= \frac{1}{200}$ sec

N = average no. of packets = 50.

~~d = average queing delay = 20 ms = 0.02 s~~

Total delay $= \frac{1}{200} + 20 = 0.005 + 0.02 = 0.025$ sec

∴ Average incoming traffic rate,

$$a = \frac{N}{d}$$

$$= \frac{50}{0.025}$$

$a = 2000$ packets/s

∵ packet size, $L = 1\ KB =$ ~~1024 × 8 = 8192 bits~~

∴ $a = 2000$ packets/s

$= 2000 \times$ ~~8192 bits/s~~ $10^3 \times 8$

~~= 20480000~~

$a =$ ~~20.48 Mbps~~ $\simeq 16\ MB$

∴ Average incoming traffic rate = ~~20.48 Mbps~~ 16 MBps classmate

Date
Page

8

Solution 5

Option A:

In HTTP/1.1 pipelining with parallel TCP connection, we can open multiple TCP connection to retrieve resource concurrently, but managing these TCP connection is costly.

Each parallel TCP session requires additional connection setup, which will increase latency.

Managing parallel TCP sessions leads to high overhead, hence option A can help in reducing bottlenecks but introduces significant overload which doesn't fully solve HOL blocking problem.

Option B:

HTTP/2 introduces multiplexing over a single TCP connection which allows multiple requests & response to be in flight simultaneously without need of ordering, which eliminate HOL blocking at HTTP layer. It reduces connection overhead with single TCP connection & it uses header compression to reduce overhead, which further improves performance.

Option C;

HTTP/2 with parallel TCP session will reduce impact of HOL blocking problem, but it adds connection management overhead similar to option A.

Hence, option B is most efficient solution, it solves HOL blocking, reduces overhead, minimizes latency which makes it ideal for most high traffic websites.

Q.6) a) $R = 1.5$ Mbps

frame duration $= 25$ ms

Time slot per user (Tslot) $= \frac{25ms}{50} = \frac{1}{2}$ ms $= 0.5 \times 10^{-3}$ s

Amount of data user can send in one slot is:

Throughput for 1 user per slot $= R \times (T_{slot})$

$= 1.5$ Mbps $\times 0.5 \times 10^{-3}$ s

$= 0.75 \times 10^{3}$.

$= 750$ bits per slot

Since, each user gets 1 slot per frame,
the throughput of one user is $\boxed{750 \text{ bits per frame.}}$

Throughput of Network :-

We have 50 slots, so,

Total throughput $= 50 \times 750$ bits per frame

$= 37,500$ bits per frame.

1 frame → 25 ms

1 second → $\frac{1}{25 \times 10^{-3}}$

1 second → 40 frames

∴ Frame rate $= 40$ frames / second

Total Network throughput (bps) $= 37,500 \times 40 = 1.5$ Mbps.

b) throughput = 512 Kbps (desired)

Throughput per slot = $750 \times 40 = 30,000$ bps per slot

For acheiving 512 Kbps, ∴ the no.s of slot required is

Number of slots = $\frac{512 \times 10^3}{30,000} = 17.07 \approx 18$ slots

Min. number of slots required are 18 slots

Q.7) a) For LCP : Average and 90th percentile both needs improvement.

For FID :- Needs improvement in 90th percentile.

For CLS :- Both Average & 90th percentile needs improvement.

b) → For LCP

① Optimize server responce time.
Using CDN to deliver content from server close to users.
Implementing cacheing strategies to store frequently requested assets.

② Lazy load images & prioritize critical content :
Large image should be lazy loaded, means they are only fetched when user scroll near them.

→ For CLS :- ① Reserving space before for ads or other ele. which can be displayed later. ② Specify diamentions of image

→ For FID ① Using preload / prefetching to prioritize critical assets
② Define 3rd party scripts from executing first.

(c) Google collect core web vitals metrics using field data, typically gathered from real users interactions, through various method.

1) Chrome user Experience Report (CrUX):
This dataset captures & aggregates anonymized performance data from real users on chrome browser worldwide.

(2) Lighthouse -
It is an automated tool used for auditing web pages. & it can simulate loading scenarios to metrics.

(3) Google search Console & PageSpeed Insights :-
these tools also provided data from CrUX & use metrics LCP, FID & CLS to give performance scores, along with suggestions for improvement.

Q.8) a) Disabling third party working does not ~~effects~~ blocks any network request, resulting 0 blocked request.

(b) 30% request are ad rebted.

Total # network req ad blocker prevent = $150 \times \frac{30}{100} = 45$ request

(c)

10

(c) 3rd party cookies related req = $150 \times \frac{40}{100} = 60$ req

(d) 60 req are related to tracking cookies,
45 req are related to ads & tracking scripts

(1) unique req blocked by ad blocker = 45 - 15 = 30

(2) unique req blocked by disabling cookies = 60 - 15 = 45.

Sol^n 9) a) Email communication is designed to be asynchronous because it allows users to ~~allow to~~ send & receive messages without requiring both sender & receipt to be online at same time.

b) Email is reliable because of

1. Message store & forward : Email server store messages until they can be ~~deliv~~ delivered.
2. Receipt confirmation :- SMTP server provide delivery confirmation when msg has been successfully handed over receipient email server.

~~Guaran~~

Guaranteed end to end delivery is not always ensured as

~~1. Receipt side issue - Recients~~ .

1. Recipient side issue - Recipient inbox might be full or receipient may have filters that block certain msg.
2. There is no direct conformation from receipient.

10) a) For ensuring user requests are served by CDN edge servers, a series of DNS resource records are configured across different DNS servers, both company authoritative name server & CDN's DNS infrastructure.

The authoritative ~~name~~ DNS server of company redirects traffic to CDN edge server

1. CName record

(ipl-stream.example.com, cdn-edge.example-cdn.com, CName)

② A record

(cdn-edge.example-cdn.com, 201.1.114.5, A)

③ NS record

(cdn-edge.example-cdn.com, ns1.cdnprovider.com, NS)

b) After opening the app or website, following step occurs -

① The user opens streaming app or website which triggers a DNS resolution process.

② The users device queries its local DNS resolver to resolve domain ipl-stream.example.com
If local DNS cache doesn't have record, query is forwarded to root DNS server
Root server refers query to ANS of company which contain cname record that point to CDN (cdn-edge.example-cdn.com).

③ ~~CDN~~ CDN's DNS server return IP of nearest edge server using A/AAAA record.

④ User device now makes connection with nearest CDN edge server using protocols like TCP, TLS.

⑤ If requested content is cached in edge server, it is served, if not req. forwarded to origin server. & content will be delivered to user.

c) Following can be adopted :-

① HTTP/2 or HTTP/3 :-

HTTP/2 and http/3 support multiplexing & reduced latency. These can handle multiple streams over a single connection, which reduces HOL blocking, making content delivery smoother & faster

(2) Edge caching :- CDN should aggresively cache both live & recorded content at edge servers to minimize load on origin server & reduce latency for users.

(3) Adaptive Bitrate Streaming :-
Implement ABR to automatically adjust video quality based on user's network conditions, which ensures that users with slower connections receive lower resolution streams without excessive buffering.