Spring 25 — set A 18221052

Q.1 Ans IP addresses remains same at the network layer, at link layer, MAC addresses changes per hop. App endpoints identifies ports.

here, source port : 443 (HTTPS, server side)

destination port : client's ephemeral port

Source IP : Web server's IP

Destination IP = PC 1's IP

Source MAC = Robten 3's IP

Destination MAC = Next hop MAC.

Q.2 Ans': → POST, upload new assignment

→ update an existing assignment

II. PUT or PATCH

III. HEAD (HEAD returns headers only): 200k → resources, 404 → no resources

check existance without downloading

Server returns : status + headers

so if status = 200k, ⇒ exists

404 ⇒ not found

saves payload bytes vs GET

Scanned with CamScanner

Q.3 | Ans: Feild : HTTP Cookie (Set-Cookie → browser stores → sends back Cookie :- on requests).

Privacy risks :- Long-lived identifiers enable cross-site tracking/profile linkage.

Mitigations : Blocks 3nd-party cookies, clear cookies, uses private mode, tracking protection, privacy extensions, use per-site permission.

Q-4 | Ans Given, TTL = T sec

changes each Δ = 1 day = 86400 s

Cache law : a resolver that cached the old A/AAAA record keeps it for up to TTL. If T >> time since change, users hit stale IPs.

Propagation : if records are changed at 00:00 and users cached at 23:59 yesterday with TTL = 3600 s, they'll still use the yesterday IP until 00:59.

even after refreshing the browser, because refresh does not bypass DNS caches.

Fix: quantitively: set TTL << Δ; T = 300s (5min)

ensures cutovers occur after old TTL's expire,

on front with CDN/anycast that stays stable

while origin changes.

Q-5 | Ans PSH (Push flag) instructs the receiver to

deliver data to the app immidiately which does not

wait for more bytes.

URG : urgent pointer ; which marks urgent region

so the receiving TCP prioritize those bytes for the

application

Q.6 | Ans Access delay at 91% router utilization

$D = I/850$; $Δr = I/350$;

$IR = 80/23 = 34.73$ Δ ms.

here, M/M/1 queue for congested access routers

If λ = arrival rate, μ = service rate,

utilization, $\rho = \lambda/\mu = 0.91$

Mean service time $T_s = 1/\mu$

$\therefore$ Mean system time, $W = \frac{1}{\mu - \lambda} = \frac{1}{\mu(1-\rho)} = \frac{T_s}{1-\rho}$

Mean queueing time $W_q = W - T_s = \frac{\rho}{1-\rho} T_s$

plog in, $\rho = 0.91$;

$\frac{1}{1-\rho} = \frac{1}{0.09} = 11.1 \Rightarrow W = 11.1\, T_s$

$W_q = \frac{0.91}{0.09} T_s = 10.1\, T_s$

Interpretation: average waiting is about 10.11x the service time, total time $\approx 11.11x$, that's why access delay explodes near saturation.

if serialization time per packet, $T_s = 0.3ms$

$\Rightarrow W_q \simeq 10.111 \times 0.3 = 3.033\, ms$

Q.7 | Ans: RTT = R ms

(i) one persistent connection

1 RTT per object

Total $\simeq$ DNS + 23R = 850

23R = 800

R = 800 / 23 = 34.78 ms

f) File transmission time:

computing total size = 5×6 = 30 MB

19×2 = 38 MB = 68 MB

= 68 × 8 = 544 Megabits

Throughput = 200 Mb/s

$$t = \frac{500\ Mb}{200\ Mb/s} = 2.72 = 2720s$$

Q. 8 | Ans: RWND = 10,000 B

Server RWND = 20,000 B

ISN : client 5678, server 1234

we know, first data byte after 3-way handshake uses ISN + 1

here, segments: C1 (client → server):

seq = 5679 ; len 546

bytes ⇒ [5679, 5679 + 546 − 1] = [5679, 6224]

S1 (server → client) : seq = 1235; len = 786

[1235, 2020]

S2 [server lost initially] seq = 2021, len 685

[2021, 2705]

S_3 : seq = 2706, len 256 ⇒ [2706, 2961]

selective repeats : ISN 5678

Senven c = 10,000, RWND: C = 10,000, S = 20,000

ii) After S_3 receive 2962

Ack number : ~~SEQ~~ seq 8725

iii) Receiving window size of Ack - 1.8958 bytes

heavy connection → long queues ⇒ high access/ queuing delay

After 000 arrival, Ack-1: ACK = 2021, SEQ = 6225

window = 8958

After ~~nt~~ retransmit received, Ack-2: ACK = 2962, SEQ = 6225

During retransmision phase of Sf = 2021.