ID

(a) Assignment-1 (20341014) Sec-23
Spring '24 Set A

| D.MAC | S.MAC | D.IP | S.IP | D.Port | S.Port |
|---|---|---|---|---|---|
| J | H | 12 | 16 | wellknown | dynamic n>1023 |

(b) Well-known / Registered

→ Question 1 ends

* Session manage
* Personalize
* make it convenient

→ Question 2 ends

POP3 behavior (more secured)

↓↓

connect → download → delete

Solution :- Using IMAP (less secured but more accessible)

→ Question-3 ends

google bought gogle(dns name) so that even with the typo it goes to google .com

→ Question-4 ends

TCP uses sliding window protocol for flow control.

The receiver has
* receive buffer
* window size

window size tells exact how much buffer space left and receive buffer receive the incoming data.

The sender agrees to send data but

there's an interval for the acknowledgement for it.

→ Question 3 ends

**Subnet mask**

A/19 means first 19 bits are 1's (network) (CIDR)
remaining 13 bits are 0's (hosts)

So, it goes like :-

11111111 . 11111111 . 11100000 . 00000000

255 . 255 . 224 . 0

**Network Address**

Host IP 173.192.221.54
Subnet mask 255.255.224.0

54 in octet :- 00110110
0 " " :- 00000000
And operation 00100000 (0 in decimal)

221 in octet :- 11011101

224 " " :- 11100000

11000000 (192 in decimal)

by this way I get 173. and (AND) 255=173,

192 and (AND) 255, 221 AND 224 = 192

54 (AND) 0 = 0 (all are in decimals)

So, the network address will be:-

173. 192. 192. 0

Broadcast Address

Now, network address :-

173. 192. 192. 0

in binary 10101101. 11000000. 11000000. 00000000

(octet)

Setting the last 13 bits to 1 will be:-

10101101. 11000000. 11011111. 11111111

converting it to decimals :-

173. 192. 223. 255

→ question 6 ends

RTT required is 480 ms

One-way delay is 15 ms (only sender to receiver)

We know RTT means Round trip time

So, RTT = (2 × 15) ms

= 30 ms

In Persistent HTTP connection, delay sum will be :-

- 1 RTT establishes TCP connection
- N RTTs to request all N objects

∴ Total RTT Delay = (1 × RTT) + (N × RTT)

= (N+1) × RTT

now, from given

(N+1) RTT = 480

⇒ (N+1) × 30 = 480

N+1 = 16

⇒ N = 16-1

∴ N = 15

→ question 7a ends

File size (L) 10 MB

Server Speed (R) 80 mbps

Transmission Time = $L/R$

$$= \frac{10\ MB}{80\ mbps}$$

$$= \frac{10 \times 8\ mbits}{80\ mbps}$$

$$= \frac{80\ mbits}{80\ mbps}$$

$$= 1s$$

→ question 7b ends

Client – Server :-

C1

Seq 1024, Data 125 bytes (1024-1148)

C2

seq 1024 + 125 = 1149, data 244 bytes (1149 - 1392)

Server – Client :-

S1

Seq 5044 (Ack of C1), data = 399 bytes (5044-5442)

S2

seq 5044 + 399 = 5443, data = 120 bytes (5443-5562)

S3

seq 5443 + 120 = 5563, data = 410 bytes (5563-5972)

S4

seq 5563 + 410 = 5973, data = 350 bytes (5973-6322)

(a) $S_3$ seq = 5563

$C_2$ seq = 1149

$C_2$ Data = 244

The last byte received from the client was 1149 + 244 - 1 = 1392

∴ Next expected byte is 1392+1 = 1393

$S_3$ Ack = 1393

(b) $S_2$ segment lost

$S_1$ received successfully (seq 5044)

$S_3$ received but it is out-of-order packet as it's still waits for 5443

With cumulative ack, the client discards the out-of-order $S_3$ segment and sends an ack for the last in-order byte pt received.

The last in-order byte was end of $S_1$. The next expected byte is 5443.

The ACK-1 segment will have an Ack no of 5443

(c)

In Selective repeat, the client buffers out-of order packets.

Client receives $S_1$ (seq 5044) → OK

Client misses $S_2$ (seq 5443)

Client receives $S_3$ (seq 5563) → Out of order

→ Client buffers S3

Client receives $S_4$ (seq 5973) → Out of order

→ Client buffers $S_4$

Diagram shows server resends $S_2$ as it's missing, and client receives it.

when $S_2$ was received it "fills the gap" in its buffer.

Now it has sequential stream of $S_1$-$S_4$

Client now can deliver $S_2$, $S_3$, $S_4$ to the application

ACK-3 segment is sent after this happens. Its acknowlegment number will indicate the new next expected byte.

The last contiguous byte received and delieved in the end of $S_4$

$S_4$ seq = 5973, data = 350 last byte of $S_4$ = 5973+350

∴ next expected byte is (6322+1) = 6322

= 6323