1a.

S.IP	D.IP	S.port	D.port	S.Mac	D.Mac
0	14	49152	80	e	11

b. A well known server port

2. The field/value is the HTTP status code in the response message like 4xx or 5xx indicates the problem and the HEAD method can be used for troubleshooting as it retrieves only the header without

the body, allowing verification of server response.

3. Iterative DNS lookup can be faster than recursive DNS lookup when the recursive resolver is overloaded or has high latency. In iterative lookup, the client directly sends queries to the authoritative servers, thus eliminating any possible delays that might arise from the intermidiate resolvers. This method avoids the bottlenecks and may take advantage of the quicker responses from the authoritative servers if the client has a good connection.

4. Web-based email systems like Gmail combine SMTP and HTTPS. The web interface with HTTPS is protected between the client and the email server but SMTP is deployed for the transmission of emails from the email server to common servers. This protection gives the users security while accessing accessing email serveres and email transmission from server to server is also reliable.

5.
The sender transmits the data segment and start the RTO timers (70 ms). When the segment is lost, the reciever does not send an Acknowledgement. The senders RTO time expires after 70 ms, prompting the sender to retransmit the segment. Then the receiver successfully

recieves the retransmitted segment and send an acknowledgement to the sender.

6. IP add: 175.172.122.75

10101111 . 10101100 . 01111010 . 01001011

Subnet mask: 255.255.128.0

11111111 . 11111111 . 10000000 . 00000000

here No of 1's = 17

Bitwise AND result = 10101111 . 10101100 . 00000000 . 00000000

In decimal = 175.172.0.0

network add = 10101111.10101100.00000000.00000000

brodcast add = 10101111.10101100.11111111.11111111

in decimal = 175.172.255.255

∴ prefix mask : /17

7.

a.

Total RTT = 2*N*RTT | N= no of obj

⇒ 480 = 2*30*N

∴ N = 8

b. obj size= 10MB = 80 ~~Mbps~~ Mbits

speed= 80Mbps

$$\text{Transmission time} = \frac{\text{obj size}}{\text{speed}} = \frac{80}{80} \text{ s}$$

$= 1$ s

8. Gt. c_1= 192 B, c2=250B, S1 = 350B, S2=127B

S3= 412B, S4= 387 B

C2 seq=5044, ack= 2024

S1 seq = 2024, len=350. so next = 2374

S2 " = 2374, len=127 so " = 2501

S3 " = 2501, len = 412 so " = 2913

S4 " = 2913 len = 387

S4 ack = 5044+250 = 5294

b.

From diagram, s2 was lost and s3 and s4 arrived out of order before the gap was filled. Using TCP cumulative ACKs, the client continuous to acknowledge the last byte that has been received in order. After s1, only in order, the next expected is 2024 + 350 = 2374.

c. s1 ; 2024 - 2373

s2 ; 2374 - 2500

s3 ; 2501 - 2912

s4 ; 2913 - 3299

∴ next expected server byte = 3300

∴ ACK-3 = 3300.

with Go-Back-N, after the sender learns that s2 was not received in order. it retransmit.