1) a) Source IP address = 16

Destination " " = 12

Source MAC address = H

Destination " " = J

Source Port = Dynamic (1024 to 49151)

Destination Port = Registered (49152 to 65535)

b) Registered Port

2) Ans: Web cookies store small pieces of user data (like login info, preferences and session ID's, in the ~~brower~~ browsers, allowing websites to recognize returning users. They help maintain login sessions, remember user settings and personalize content. Cookies also enable shopping carts and save browsing history for smoother navigation. Overall, they make browsing more convenient and customized by preserving user specific information across visits.

3) Ans: This happened because John's email ~~cilent~~ client is using the POP3. By default, POP3 downloads messages from the mail server to the local computer and then deletes them from server. As a result, the emails are no longer available online or any other devices. If John wants to access his email from multiple devices he should use the IMAP instead, which keeps emails stored on the server.

4) Ans: This happens because the domain "gogle.com" has likely been registered by Google and configured in DNS to point to the same IP address a "google.com". A CNAME or a record can map 'www.gogle.com' to the same web server as "www.google.com". When a user types the misspelled address, DNS resolves it to that IP and the server redirects the user to the correct site. This ensures users still reach Google even if they make a small typing mistake.

5) Ans. Flow control in the transport layer ensures that the sender doesn't overwhelm the receiver with too much data at once. In TCP, the receiver advertises a window size that tells the sender how much data it can handle at a time. The sender then limits it's transmission to stay within this window. If receiver's buffer is full, it can temporarily reduce the window size or set it to zero until it is ready again. This mechanism prevents buffer overflow and ensures that no data segments are lost at the receiver side.

6) i) Subnet Mask:

/19 → into Binary Mask is = 11111111.11111111.11100000.
00000000

converting each 8bit octet to decimal ; 11111111 = 225

11111111 = 225

11100000 = 224

∴ subnet mask = 225.225.224.0

ii) Network Address:

Given, Host IP = 173.192.221.54

converting to Binary = 10101101 11000000 11011101
00110110

Doing 'AND' between Host IP and Subnet Mask = (3rd & 4th Octet only)

~~10101101 11000000 11011101 00110110~~

~~11111111 11111111 11100000 00000000~~

~~11111111 11111111 11111101 00110110~~

3rd Octet = 11100000

11011101

11000000 → 192

∴ Network address is = 173.192.192.0

Broadcast Address:

Network Address (Binary of 3rd & 4th octet) = 11000000 00000000

flip host bits to 1 = 11011111 11111111

Converting to decimal = 223 255

∴ Broadcast Address = 173.192.223.255

(Ans)

7) a) total RTT = 480 ms

One way delay = 15 ms

⇒ RTT = 30 ms

Total RTT cost = 1 RTT for TCP + n × 1 RTT per object

~~4800~~ 480ms = (1+n) . RTT 48ms

480ms = (1+n) . 30 ms

$$\Rightarrow \frac{480}{30} ms = (1+n)$$

∴ n = 15 objects

(Ans)

b) each object size = 10MB | 10MB = 10×8 ~~b~~

server speed = 80 Mbps | = 80 Mb

$\therefore$ transmission time = $\frac{80Mb}{80Mbps}$

= 1 sec (for one object)

final

$\therefore$ transmission time for 15 objects = 15 seconds

(Ans)

8) a) given,

c1 seq = 1024

c1 ack = 5044

client byte after c1 = 1024 + 125

= 1149

after c_2 = 1149 + 244

= 1393

Server byte space starts at 5044 (from c1's ack)

after s1 = 5044+399 = 5443

" s2 = 5443+120 = 5563

" s3 = 5563+410 = 5973

" s4 = 5973+350 = 6323

$\therefore$ S3 segment's ; seq = 5563

ack = 1393

(Ans)

b) the client misses S2 with seq = 5443

the client receives S3 (seq = 5563 ; which is out of order.

Ack.1 starts just after receiving S3.

$\therefore$ next expected byte = seq of S1 + Data of S1

= 5044 + 399

= 5443

c) In selective repeat, the client buffers the out of order segments S3 and S4, even though its missing S2. When re-transmitted S2 finally arrives, the client fills the gap and immediately passes S2, S3 and S4. from its buffer. Ack.3 is sent after all this data is processed, so it's ack number will be for the next byte expected after S4.

$\therefore$ ack = seq of S4 + Data of S4

= 5973 + 350 = 6323

(Ans)