ASSIGNMENT-1

NAME : Muhtasim Ahmed

COURSE : CSE421

SEC : 23

ID : 22299106

Answer to the ques No. Q1

a) Interoperability: Standards ensure devices from different different vendors can communicate correctly.

b) Compatibility and economies of scale: Standardized interfaces reduce development cost and make components reusable across products.

c) Security and reliability: Standards define tested, vetted protocols and behaviors (including security practices), improving network stability and trust.

Answer to the ques No. Q2

The causes are-

a) Multiplexing: HTTP/1.1 typically opens multiple TCP connections or pipelines requests (which suffers from HOL blocking)

b) Binary framing and header compression:

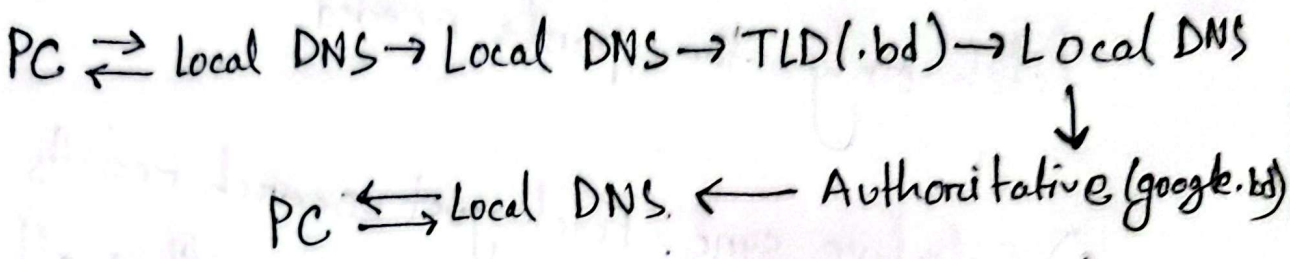
HTTP/2 uses binary framing and HPACK header compression, which reduces header overhead and parsing cost vs the textual headers of HTTP/1.1.

c) Server push: HTTP/2 can push resources and prioritize streams, improving perceived load time

d) Fewer TCP: HTTP/2 commonly runs over a single persistent TLS connection, reducing handshake overhead relative to many short HTTP/1.1 connections.

Answer to the ques No. Q3

Diagram:



So, total query-response pairs (for Local DNS to finish resolution) = 3 (plus the intial PC ⇄ local exchange which is the client's first pair)

Answer to the ques No. Q4

Scenerios:

a) Low-bandwidth clients: Download message headers and a preview but not large attachments which saves bandwidth and improves responsiveness.

b) Large attachments: User wants to read the texts but not the attachments – IMAP lets the client fetch only the parts needed.

c) Selective sync: Fetch just search results or newest messages' bodies without pulling all mail content.

Answer to the ques No. Q5

I would recommend TCP as it provides reliable, in-order delivery, retransmission of lost data, and congestion control – all important when correctness is the main priority.

Answer to the ques No. Q6

The sender retransmits a previous data segment before RTO timer expires when the sender receives multiple duplicate ACKs (typically 3 duplicate ACKs). This triggers fast retransmit which means the sender infers a packet was lost (receiving repeated ACKs for the same byte) and retransmits immediately without waiting for the RTO.

Answer to the ques No. Q7

i)

Given,

Total objects = 24

One way small packet propagation = 27ms

Server link speed = 25 Mbps

Entire page load time = 4.345 Seconds

Total RTT = 27 + 27 = 54ms = 0.054 s

11) Per obj fixed overhead = 27+27+5 = 59ms
= 0.059 s

Total time = 24×0.059
= 1.416s

Remaining time = 4.345 - 1.416
= 2.929 s

Per obj transmission time = 2.929/24
= 0.122 s

Per obj (bits) size = 0.122×25000000
= 3051041.667 bits

Per obj (bytes) size = 3051041.667/8
= 381380.21 bytes

So each obj ≈ 381380 bytes (372.4 KiB).

Answer to the Ques No. Q8

1) S1 seq = 1910

Length = 232

S2 seq = 1910 + 232 = 2142

S2 length = 365

S3 seq = 2142 + 365

= 2507

Ack = 1532 [As its was the ack when S1 was sent and also the server didnot receive any additional data]

∴ S3: seq: 2507

Ack: 1532 →

ii) Client FIN:

Seq will be 1532 as it didnot receive any additional data

Ack: (1910+232) = 2142.

∴ Seq = 1532, ack = 2142

iii) Client receive-window (rwnd) while sending FIN

rwnd before S1 transmitted = 7000

Total occupied = 232+365+421
= 1018 bytes

Client buffer = (7000 - 1018) = 5982 bytes

Client then processes S1 and frees up 232 bytes,

Available buffer = 5982+232
= 6214 bytes

∴ Client rwnd while sending FIN: 6214 bytes