Assignment-1

Dipta Mazumder

ID-24141264

Section-23 (CSE421)

Ans. to the Que no - 1

(a) Correct order from Application Layer (7) to physical Layer (1)

1. Application Layer (L7) - Step-6
2. Presentation Layer (L6) - step-5
3. Session (L5) - Step-3
4. Transport (L4) - Step-1
5. Network (L3) - step-2
6. Data Link (L2) - Step-7
7. Physical (L1) - step-4

(b) Yes.

In BitTorrent, peers exchange different pieces of the file. The protocol is designed so that there is at least one seeder → (a peer with complete file) collectively the swarm has all pieces distributed among peers, downloading is possible.

(c) No.

The proxy server won't send a conditional Get request since it is dependent on the freshness of cache. proxy server checks if the cached copy is valid and fresh. In question, the proxy has a copy but doesn't mention client validation headers.

(d) This possible because of;

1. Video Compression & Efficient Encoding

2. Content delivery networks

3. Buffering

* access link bandwidth is lower than 100 mbps but higher than actual average bitrate of the compressed stream. Thats why it can stream 8k without buffering.

Ans. to the Que no -2

(a) Websites track even without sign up by;

1. cookies
2. Ip address
3. Browser fingerprinting
4. Local storage

Tracking helps the HTTP protocol

1. managing sessions
2. provide personalized contents or ads
3. Enable analytics
4. Caching ~~optima~~ optimization

(6) Recursive DNS lookup is better overall for the typical internet ecosystem.

Justification:

1. End-user experience - client get faster response

2. Widely deployed - Google, cloudflare use recursive resolution for clients

3. Reduces complexity - don't need to handle multi-step iterative process

4. Caching efficiency - Recursive resolves cache results for many users.

2] (c)

(i) PC to Local DNS = 23 ms

RTT total = 46 ms

Total recursive resolution = $(3 \times 46) = 138$ ms

∴ Total RTT = local DNS + recursive resolution

= 46 + 138

= 184 ms.

Ans.

(ii) Here,

DNS time = 184 ms

1 RTT between client and server = 78 ms

TCP handshake to open persistent connection;

for 30 objects = 30 × 78 = 2340 ms.

∴ Total = DNS + TCP + 30 objects

= 184 + 78 + 2340

= 2602 ms Ans.

(iii) Here,

Total time = RTT time + Data transfer time

$\Rightarrow 5344 = 2602 + \text{Data transfer time}$

$\Rightarrow$ Data transfer time $= 5344 - 2602 = 2742$ ms.

Now,

each object = 12 MB = 96 Mb

Total data = $30 \times 96 = 2880$ Mb

Now,

Data transfer time = $\frac{2880}{x}$ MBps

$\Rightarrow \frac{2742}{1000} = \frac{2880}{x}$

$\Rightarrow x = 1050.69 \approx 1.05$ Gbps

Ans to the Ques no-3

(a) Yes, It is possible for a web server to receive multiple HTTP request that contain the same port number. However, each request does not arrive over the same TCP connection. TCP connection is uniquely identify by;

Source IP, Source Port, Destination IP and destinatio port.

The server differentiates them based on their unique combination of source IP and source port number.

(b) UDP doesn't provide reliability because it lacks fields such as sequence Number, Acknowledgement number and Window size in its header.

However, reliability is achieved at the application layer, where applications implement their own mechanism like ACKs, sequence numbering or error correction to ensure data is recieved correctly when UDP is used.

(c) (i)

Server ISN = 203

1st byte = 203 + 1 = 204

Sequence number of 1st byte of nth server data;

$$Seq_n = 204 + (n-1) \times 889$$

For 4th;

$$Seq_4 = 204 + (3 \times 889) = 204 + 2667$$

$$= 2871 \quad \text{Ans.}$$

Now,

Client ISN = 8924

1st byte = 8924 + 1 = 8925

4 HTTP request for each 234 bytes = 4 × 234 = 936

$\therefore$ Ack = 8925 + 936

= 9861

Ans

(ii)

$$\text{Client Seq}_k = 8925 + (k-1) \times 234$$

$$\text{clientseq}_{13} = 8925 + (13-1) \times 234$$

$$= 8925 + 2808$$

$$= 11733$$ Ans.

for Ack,

10th server data was lost, only received segments (1-9) in order;

$$\text{ACK} = 204 + (9 \times 889)$$

$$= 204 + 8001$$

$$= 8205$$ Ans.

(iii) Total bytes recieved = 12 x 889 [1-9 and 11-13]

= 10668 bytes Total = 12

processed 5 segments bytes = 5 x 889

= 4445 bytes

Unprocesed bytes = (10668 - 4445) = 6223 bytes

Advertised RWND = reciever butter size - unprocessed byte

= (10000 - 6223)

= 3777 bytes

Ans.