BRAC UNIVERSITY

Inspiring Excellence

CSE421

COMPUTER NETWORKS

ASSIGNMENT 1

Name: Basit Ibrahim

ID: 22241081

Section: 22

Faculty: BIJS

Answer to the question number 1

The Application, Presentation, and Session layers (Layers 7, 6, 5) of the OSI model were combined into a single Application layer in the TCP/IP model for the following reasons:

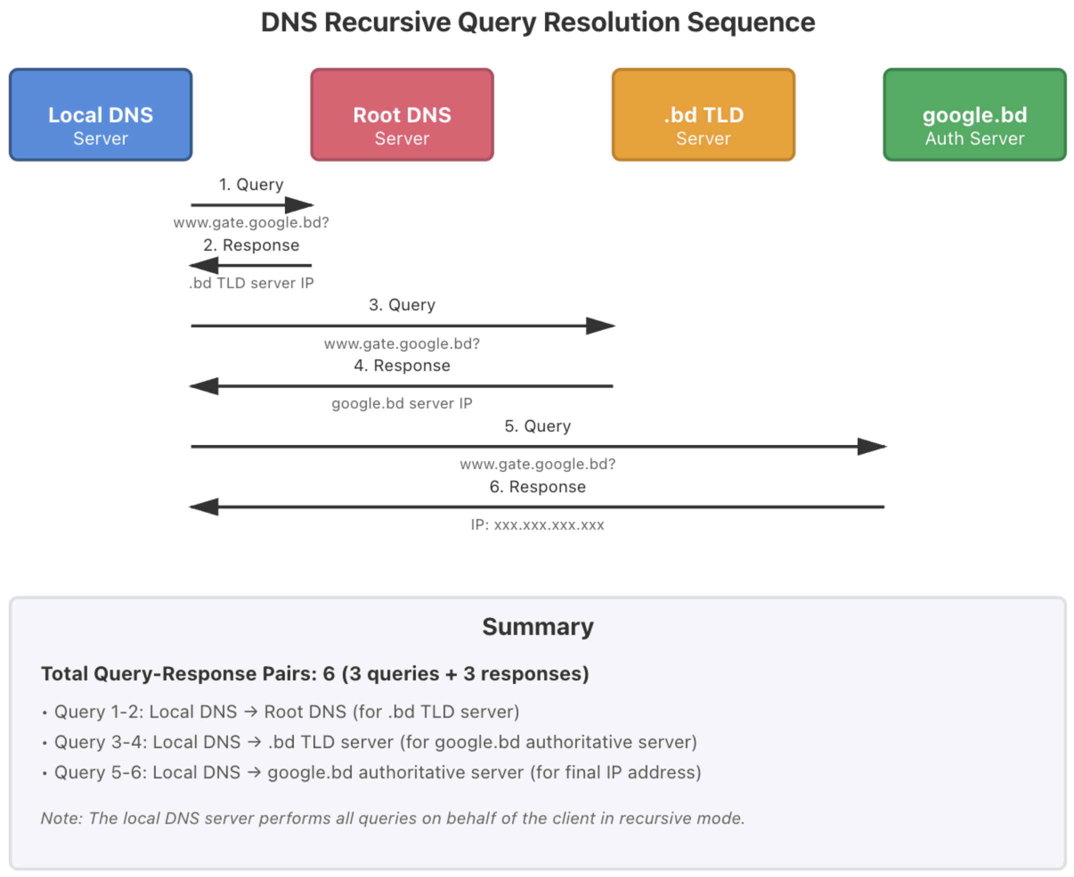
1. Practical Implementation: In real world protocols these three layers functionalities are often connected to each other and not distinctly separated. Applications typically handle their own data formatting, encryption, and session management.
2. Simplicity and Efficiency: The TCP/IP model was designed to be more practical. Combining these layers reduced complexity while maintaining all necessary functionality.
3. Flexibility: By combining these layers the TCP/IP model allows applications to implement only the features they need rather than forcing a rigid three layer structure that may be unnecessary for many applications.

Answer to the question number 2

Thursday Batabd website knew that I was a first time visitor because my browser sent an HTTP request without any cookies for batabd.com domain. The absence of cookies indicated a new user. Therefore on that day the server responded with a set cookie header containing a unique session identifier. MY browser stored this cookie locally and items added to cart were associated with this session ID on the server

On friday my browser automatically included the previously stored cookie in the HTTP request header. The server recognized my session ID from the cookie. The server retrieved cart items associated with that session ID from its database. This allowed the cart to persist without requiring an account.

Answer to the question number 3



Summary

Total Query-Response Pairs: 6 (3 queries + 3 responses)

- Query 1-2: Local DNS → Root DNS (for .bd TLD server)
- Query 3-4: Local DNS → .bd TLD server (for google.bd authoritative server)
- Query 5-6: Local DNS → google.bd authoritative server (for final IP address)

Note: The local DNS server performs all queries on behalf of the client in recursive mode.

Answer to the question number 4

Yes there are scenarios where POP3 can be more useful:

1. **Limited Server Storage**: When the email server has strict storage capacity POP3 downloads emails to the local device and typically deletes them from the server freeing up server space.
2. **Single Device Access**: Users who access email from only one device don't need synchronization and can benefit from POP3's simplicity and lower server resource usage.
3. **Offline Access**: POP3 stores all emails locally providing complete offline access without requiring server connectivity after initial download.
4. **Privacy**: Some users prefer not to keep emails on remote servers. POP3 allows complete local storage with server deletion.

5. **Slow**: Once emails are downloaded users can read and compose replies offline which is useful in areas with poor connectivity.

Answer to the question number 5

The source port numbers differ because each TCP connection requires a unique socket identifier. A socket is defined by the combination of Source IP, Source Port, Destination IP, Destination Port. Even though both tabs originate from the same device same source IP the operating system assigns different ports to distinguish between the two separate TCP connections.

Destination ports are 80 (for HTTP) or 443 (for HTTPS). The destination port is a well known port while the source ports are dynamic ports which are temporary ports assigned by the OS typically in the range 49152-65535.

Answer to the question number 6

The server does not send the FIN bit immediately in the half close scenario when the server has data still to send. When the client sends a FIN segment the server responds with an ACK to acknowledge the client's FIN. If the server still has data to transmit to the client it will:

1. Send ACK to acknowledge the client's FIN
2. Continue sending its remaining data segments
3. Only send its own FIN segment after all data transmission is complete

This creates a half closed connection state where:

- Client → Server direction is closed
- Server → Client direction remains open

The server sends its FIN bit only when it has finished transmitting all pending data and is ready to fully close the connection.

<u>Answer to the question number 7</u>

Given,

X objects total, each 1.5 Mb

Waiting delay: 7 ms

RTT for small packet: 34 ms

Server speed: 60 Mbps

Total time excluding RTT: 1.184 seconds

I. Calculate Number of Objects (X)

For non-persistent HTTP:

1 HTML base page + (X-1) embedded objects

Each object requires: waiting delay + transmission time

Transmission time per object = 1.5 Mb / 60 Mbps = 0.025 seconds = 25 ms

Time for X objects (excluding RTT):

Total = X × (waiting delay + transmission time)

1.184 s = X × (7 ms + 25 ms)

1184 ms = X × 32 ms

X = 37 objects(Answer)

II. Calculate Total RTT

For non-persistent HTTP with X = 37 objects:

Initial TCP connection: 1 RTT

HTTP request for base page: 1 RTT

For each of 36 remaining objects: 1 RTT (TCP) + 1 RTT (HTTP request) = 2 RTT each

Total RTT = $1 + 1 + (36 \times 2) = 2 + 72 = 74$ RTT

Total RTT in seconds = 74×34 ms = 2516 ms = **2.516 seconds (Answer)**

Answer to the question number 8

Given,
S1: seq = 8742, ack = 4531, size = 532 bytes

C1 size: 191 bytes

S2 size: 320 bytes, S3 size: 160 bytes

Client rwnd: 12000 bytes, Server rwnd: 14000 bytes (before S1)

I. FIN Segment Sequence and Acknowledgment Numbers

The sequence shown indicates a connection termination. The FIN segment from the server would be

Server's FIN segment,

Sequence number: 8742 + 532 + 320 + 160 = **9754** (After sending S1, S2, S3, the next sequence number is used for FIN)

Acknowledgment number: 4531 + 191 = **4722** (Acknowledging the C1 segment received from client)

II. ACK Segment from Client

Client's ACK for server's FIN,

Sequence number: **4722** (Next byte client will send - this is same as previous ack number since no data in ACK)

Acknowledgment number: 9754 + 1 = **9755** (Acknowledging FIN; FIN consumes one sequence number)

III. Server's rwnd After Receiving ACK

Server's rwnd calculation,

Initial server rwnd: 14000 bytes

Client sent C1: 191 bytes (received by server)

Server's available buffer = 14000 - 191 = 13809 bytes

After client's application reads the data, rwnd would increase, but with given information:
Server's rwnd after receiving ACK = **13809** bytes