Jason Lai
204995126

Problem 1

1. C, E
2. A, B
3. B, C, D.
4. D
5. B, C, D, E
6. E
7. A, D
8. B, E
9. B, F
10. B, D, E
11. A, B, C, E
12. C
13. C, E
14. A
15. B, E

Problem 2

1. A hostname is resolved in Iterated Query as follows: the receiving host queries the local DNS server for the hostname-to-IP address mapping. The local DNS server, if it doesn't have this mapping, it queries the highest server in the hierarchy: the Root DNS server. If the Root server doesn't have the mapping either, it responds by telling the local DNS server to contact the next level on the hierarchy: Top-Level Domain servers. The local DNS server will continue contacting servers in descending order of the hierarchy, until one of them gives it the mapping instead of the response: "I don't know this name, but ask this server." This is different from recursive Query as the local DNS server is the one that has the burden of querying each server in the hierarchy, whereas, recursive places this burden on the contacted name servers i.e. local DNS asks Root, Root asks TLD, TLD asks authoritative, and responses are sent in reverse back to local DNS then receiving host.

2.

3. SMTP needs authorization to prevent strangers from sending emails from one another resulting in spam; SMTP authenticates that the sending and receiving hosts have good addresses before sending mail. Furthermore, access protocols like POP have authorization to ensure the user is the owner of the mailbox, before allowing user commands to manipulate the mailbox. HTTP does not need authorization since the interaction between client and server is just requesting web pages; the server has full control

4. $\frac{L}{R}$ = transmission delay for 1 packet

The queuing delay for the last transmitted packet is $N\frac{L}{R}$ seconds.

5. If the student group expects a large number of users eventually, then client-server paradigm should be used. If a large number of users is expected, then many factors can affect user experience like distances between peers in remote locations and the amount of data stored on each peer's device about the network. Therefore, I think a distributed client-server system would be more appropriate for handling a large database and covering large areas reliably. This way peers do not need to rely on one another for data, and instead rely on the server.

Problem 3

1st error:

```
if((server_fd = socket(AF_INET, SOCK_STREAM, 0)) > 0) {
```

this line will cause the program to exit upon successful socket creation, since socket() returns a file descriptor. This should be changed to < or == -1.

2nd error:

```
address.sin_port = ntohs(PORT)
```

The correct function call should be htons(MYPORT), since we want to convert the unsigned short from host byte order to network byte order for the port.

3rd error:

```
if(accept(server_fd, 3) < 0) {
```

This should be listen, since we have to listen for a connection request before we can accept any messages. Also incorrect number of arguments for an accept call anyways

4th error:

```
if ((new_fd = listen(server_fd, (struct sockaddr *)&address, (socklen_t*)&addrlen)) <0)
```

This should be accept, since we want to accept messages after listening. This should also be placed in a main accept loop for best effect. Incorrect number of arguments for listen anyways.

Problem 4.

1. 1st packet: $\frac{500}{2\times10^6} + 0.001 + \frac{500}{1\times10^6} + 0.001 = 0.00275s$

2nd packet: $\underset{\text{queue delay}}{\frac{500}{2\times10^6}} + 0.001 + \frac{500}{2\times10^6} + 0.001 + \frac{500}{1\times10^6} = 0.003$

$0.003 - 0.00275 = 0.00025$ Time gap: **0.00025 seconds**

2. 3rd packet:

$$2\times\underbrace{\frac{500}{2\times10^6}}_{\text{queue delay}} + \frac{500}{2\times10^6} + 0.001 + \frac{500}{1\times10^6} + 0.001$$

$$= 0.00325$$

It takes **0.00325 seconds** for C to receive all 3 packets.

Problem 5.

1. Joe's search will use DNS, TCP, UDP, and HTTP protocols.
   Application layer: DNS, HTTP
   Transport-layer: TCP, UDP

2. The client will first query the local DNS server to get the IP address of google.com server. Then the local DNS will do some iterated query, unless it has a cached IP for the hostname, and respond to the client with the IP. with the IP, the client will contact the web server for Google.com and request a TCP connection and google web server sends back ACK. Then the client sends request for Google's home page and server responds with base html file. Plus any referenced objects the client requests. the client then types the search field for "news today" and sends another HTTP request and the server responds. Then Joe clicks on cnn.com and

3. Persistent: $2(1) + 20(1) =$ 22 seconds

   Non-persistent: $2(1) + 20 \times 2(1) =$ 42 seconds

   Non-persistent w/ parallel: $2(1) + 2(1) + 2(1) + 2(1) + 2(1) =$ 10 seconds

   The faster one is Non-persistent with parallel connections.

4. The protocols used are HTTP, TCP, SMTP, IMAP/POP, DNS.

Survey

1. above my expectation

2. Average

3. Slow

4. No

5. For improvements, I think more examples of how protocols or algorithms from now on would give me a better idea on how the more complex concepts work. The slides are good, but I feel they could have better terminology used e.g. piggybacking in sample quiz, but I do not remember hearing this term in lecture