Wyatt Kormick
4932481

1.
    a. False
    b. True
    c. False
    d. True
    e. False
2.
    a. The iterative DNS query mechanism lets the root server follow the path of authoritative DNS servers to find the best one, the one most likely to meet the client's needs, for its query, and hand the job over to them, so they can do the same thing, or have a response that they can then send directly back to the client's resolver.

    b. Advantages
        i. If the client doesn't know where to get a response, a recursive query will get the response for it, whereas an iterative query would just get referrals to servers that might have an answer, but may just have another referral instead.
        ii. A recursive DNS query can speed up the process by using 'root hints' (sections of the host name that match up with DNS servers) to find the correct response quicker.

    Disadvantage

        Recursive dns queries can allow for a long chain of recursive queries through many different servers. When a server finds an answer it will have to go back up through the long chain of servers to get back to the client, potentially taking quite a bit of time, whereas with an iterative request, the server with the answer would immediately send the answer back to the client.

c. The DNS message keeps an identifier in its header that the server can then match to the identifier in the query sent by the client, and then send the message to the client matching the matched query.

d. DNS messages can fit within one UDP message, so it is faster to just send one UDP segment, and receive another one, than it is to set up the three way handshake with TCP (at least three messages), and then start sending just a few messages. If a DNS query is lost, the client will timeout, and send either a request to a different DNS server, or tell the first DNS server that it didn't receive a reply.

e. 25 msec. 15 msec to do the three-way handshake to set up the TCP, 5 msec to send the request to the server, and 5 msec to receive the reply from the server.

3.

a. The server wouldn't receive its ack message for its synack, so it would eventually resend its SYNACK message. The client would receive this duplicate SYNACK message, realize that it's ACK was lost, and resend it.
b. This scenario isn't possible. The SYN message is used for establishing a connection from the client to the server. A client would not send a SYN message after it has already established the connection, and we know the connection did get established. IF it did happen, the server would attempt to establish a connection with the client again by sending a SYNACK.
c. The server receives a SYN, sent back a SYNACK for the sequence number, and receives an ACK back for the same sequence number as its SYNACK. These make up all of the parts of the TCP three way handshake. According to the server side, the TCP connection is reestablished at this point.
d. The purpose of the TIME_WAIT state on the client side is to ensure the final ACK goes through. If the ACK packet doesn't make, it is fine because the client is still waiting, and will be able to resend it. Without it the client would close, the packet could get lost, and the server wouldn't close.

4.

a. If the server receives a packet with a sequence number (ex: 1), it will send an ACK for that same sequence number back. If it doesn't receive that packet however, it will think that its previous ACK (ex: 0) with was never received by the client, so it will send that ACK to the client again. The client will see that the ACK with this old sequence number (0) has been sent instead of the ACK for its latest packet (1) and realize that it needs to resend it. The data transfer wouldn't be reliable without it.
b. The receiver will receive packet three as normal, sending an ACK back for it. It will buffer packets five and size and buffer them, as well as sending the ACKs for them back. It will set window to start at four. Once the sender resends packet four successfully, the other packets will be received as normal, and the connection can go on.
c. The sender's window will move so the first entry in the window is packet four's. Since this is only a move of one, the sender can only send one more packet.
d. RTT = 0.072 msec, L = 80000 bits/packet, Data Rate R = 10^10 bits / sec,
   Window (W) = 4 packets d_trans = L/R = 0.008 msec, RTT+d_trans = 0.08 msec
   U = (W*(L/R))/(RTT+D_trans) = 0.4
   i. The maximum utilization of the link (sending 4 packets) is 40%
   ii. 1 = (W*(0.008))/(0.08), W = 10 packet window
   iii. We need 4 bits to represent a window size of 10 packets

Selective Repeat Protocol Animation used for problem 4:
https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/selective-repeat-protocol/index.html

Information on Iterative and Recursive DNS queries for problem 2:
http://www.slashroot.in/difference-between-iterative-and-recursive-dns-query