Conner Mattingly
11282717
Exam 1 Makeup

1. a)
Reason: I did not have enough knowledge of the three protocols, nor did I have a deep enough understanding of the question in order to give an answer that would recieve full points.
Answer: All three of the protocols are used in different situations for different reasons. However, I think the most essential protocol for use in the internet is SMTP. This is because SMTP allows for relaying and forwarding emails between multiple mail servers and sending emails from a client to a server. The other two protocols have to do more with the transfer of mail from a server to a client.

2. b)
Reason: I made a conversion mistake of megabits to bits. Understood the formula to get the correct answer, just used an incorrect value in my calculations by mistake.
Answer: Transmission time = Packet size/ Bit Rate
(2000 bits) / (1 megabit per second) = 2000/1,000,000 = .002 seconds
d)
Reason: It looks like I had the correct formula and used the wrong numbers which resulted in an incorrect answer.
Answer: (TransmissionTime) / (RoundTripTime + TransmissionTime) = Utilization
.002seconds / (.002seconds + (5000kilometers /200,000kilometersPerSecond))
= .074 (Multiply by 100 to get percent) = 7.4 % Utilization.

5.
Reason: I didn't have enough experience with the recv() system call.
Answer: B. recv() returns the number of bytes read in, which will be anywhere from 0 to the value of the size parameter that was passed in, or -1 if there were any errors.
10.
a)
Reason: I didn't know how to calculate the setup time or at least did not think long enough about how to calculate it. Also forgot to add RTL to calculate the total time taken to send a packet.
Answer: (3-way handshakeTime)*(# of packets)*(Dtrans for a packet + RTL)
= (1.5 * RTL) * (11) * (RTL + 50)
= (1.5 * 100) * (11) * (100 + 50)
= (1.5 * 100) * (11) * (150)
= 150ms * 11 * 150ms
= 247500 ms or 247.5 seconds
b)
Reason: Same reason as 10.a. Didn't calculate the setup time for TCP.
Answer: (3-way handshakeTime) + (# of packets)*(Dtrans for a packet + RTL)
= (1.5 * RTL) + (11) * (RTL + 50)
= (1.5 * 100) + (11) * (100 + 50)
= (1.5 * 100) + (11) * (150)
= 150ms + 11 * 150ms
= 1800 ms or 1.8 seconds

c)
Reason: Forgot to add RoundTripTime to main page packet time and forgot to add initial setup time. Also, which HTTP version does this question ask for? (It asks for minimum time so I will assume HTTP 1.1)
Answer: (setupTime)+(Time for main page)+(time for other pages)
= (1.5 * RTL)+(OriginalRTT + OriginalDtrans) + (NewRTT + NewTrans)*(# of packets)
= (100 * 1.5) + (100 + 50) + ((1+1)*(10))
= 150 + 150 + 20
= 320 ms or .32 seconds

11.
a)
Reason: Understood the difference of endianess but could not figure out how to calculate from the way it was stored.
Answer:
b)
Reason:
Answer:
c)
Reason:
Answer:

12.
b)
Reason: It looks like I thought the Acknowledgment numbers were suppose to increase based off of what the sender is sending but in reality they increase in response to what was received.
Answer: The Ack field of the packets from A to B will increase according to the payload size of packets coming from B to A. Therefore the acks sent from A should be as follows.
Ack1: 1
Ack2: 51
Ack3: 101
Or the following if not taking into account the 3-way handshake.
Ack1: 0
Ack2: 50
Ack3: 100
c)
Reason: Just had an incorrect understanding of Acknowledgement fields.
Answer: The Ack number should from B should match the next sequence number that A will send. For the Acknowledgement of the first packet the Ack would be the original sequence number plus the size of the packets payload(length of the packet). That would result in an Ack of 1050.
e)
Reason: Looks like I forgot to mention what A would do.
Answer: A would perform a retransmit of the lost packet after the duplicate acknowledgements are received or a timeout occurs. Whatever comes first. (When a sender retransmits before the timeout, due to dup acks, it is known as a fast retransmit)
f)
Reason: I’m unsure which sender is meant by the question and am unsure what B would do
Answer: I think that A would just retransmit the second packet when the packets timeout occurs.