1.
(a) Propagation delay is the amount of time to transmit a signal from one place to another over a link. Propagation delay is dependent on the distance of physical link and propagation speed in medium, which is about 2*10 meters per second.

(b) Transmission delay is the amount of time required for the router to push out all the packet's bits into the wire. Therefore, transmission delay is dependent on the length of the packet and transmission rate between routers.

(c) A protocol is a set of rules and conventions about how to format, transmit and receive data between network devices.

(d) One key advantage of a layered network architecture is that it eases maintenance and updating by modularization, which means we can update a layer's implementation without affecting its lower layers.

(e) False. HTTP/1.1 is persistent because it allows multiple objects to be sent over a single TCP connection, not because it makes HTTP stateful. HTTP is stateless because it does not maintain past histroy.

(f) False. First of all, UDP is a protocol for communications, though it does not establish connection between two machines. It still requires port number to identify which process is being used. Otherwise, UDP can not make two machines communicate with each other.

2.
(a) It can support 200 users simultaneously. 200Mbps/1Mbps = 200

(b) The probability that a given user is transmitting is 10%, because each user will transmit randomly only 10% of time.

(c)
The probability that exactly half of the users are transmitting simultaneously is:

$$\binom{150}{300} 0.1^{150} * 0.9^{150}$$

(d) Since up to 200 users are able to transmit data with 1Mbps simultaneously, if there are more than 200 users are transmitting simultaneously, congestion will happen. The probability that more than 200 users are transmitting data simultaneously is:

$$1 - \sum_{n=0}^{200} \binom{200}{n} 0.1^{200} * 0.9^{200-n}$$

3.
(a) The time it takes to move the message from the source host to the first packet switch is:
(8 * 10^6) / (2 * 10^6) = 4 seconds

(b) The time it takes to move the first packet from source host to the first switch is:
10000 / (2 * 10^6) = 0.005 seconds

The time the second packet will be fully received at the first switch is:
0.005 * 2 = 0.01 seconds

(c) The time it takes to move the first packet to the destination host is:
10000 / (2 * 10^6) * 3 = 0.015 seconds
After that, one packet will be moved to the destination host every 10000 / (2 * 10^6) seconds.
Therefore, the time it takes to move the file from source host to destination when message segmentation is used is:
0.015 + 799 * 10000 / (2 * 10^6) = 4.010 seconds

(d) When we use message segmentation, each packet can be switched within a small amount of time, which means that it can handle larger volumes of traffic even if there are many computers sending packets at the same time. Otherwise, some computer needs to wait longer for other computers to finish sending the package. In addition, if an error occurs during the transmission of an unsegmented package, the package may need to be retransmitted, which will waste a lot of time.

(e) With message segmentation, the destination has to rearrange packets in the correct order, which means more work needs to be done in the destination. In addition, since headers are added to packets when passing through layers, more amount of bytes are needed to be transmitted because we have more packets produced by message segmentation.

4.
1) The time it takes local host to obtain IP address is $RTT_1 + RTT_2 + \cdots + RTT_n$. Then the time it takes local host to establish TCP connection with the server is RTT_0 and finally it takes another RTT_0 for local host to request and receive the object. Therefore, the total amount of time is: $2RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n$

2)
(a) According to the previous problem, the time it takes to receive the HTML file is $2RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n$. After that, local host will go on to fetch another eight files and each one will take $2RTT_0$ since a single TCP connection can only send one HTTP request. Therefore, the total time elapses with Non-persistent HTTP with no parallel TCP connections is:

$$2RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n + 2RTT_0 * 8$$
$$= 18RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n$$

(b) Since the browser can have 5 parallel connections at a time, it only takes $2RTT_0$ to obtain 5 objects once local host have the IP address of the server. Therefore, the total time elapses with non-persistent HTTP with the browser configured for 5 parallel connections is:

$$2RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n + 2RTT_0 * 2$$
$$= 6RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n$$

(c) For persistent HTTP, once local host has established a TCP connection with the server, local host can obtain multiple objects in a single request without initiating a new TCP connection. Therefore, the total time elapses with persistent HTTP is:

$$2RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n + RTT_0$$

$$= 3RTT_0 + RTT_1 + RTT_2 + \cdots + RTT_n$$

5.
1) The destination port for the DNS query message is 53. The source port of DNS response message is also 53.

2) The DNS query message is sent to 8.8.8.8 and this IP address is my default local DNS server (Also known as Google DNS).

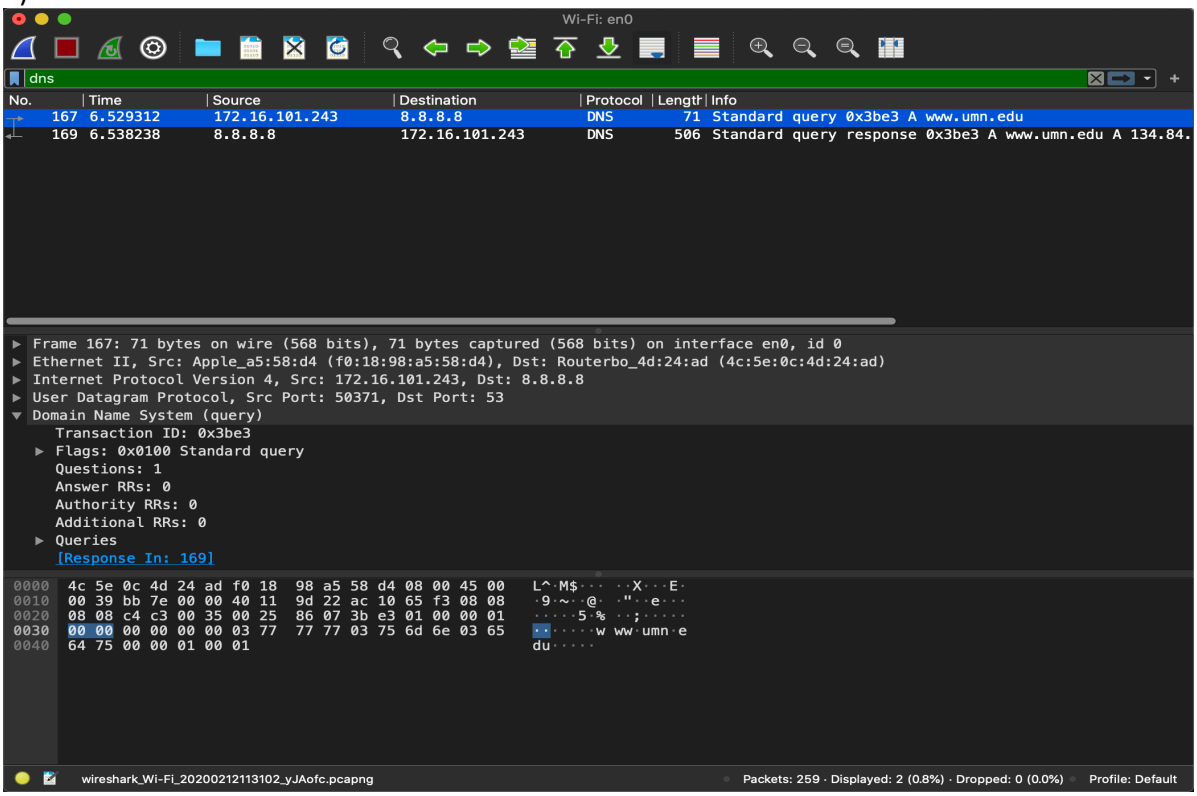
3) The responses also come from 8.8.8.8, which is my default local DNS server.

4) In the third DNS query message in Fig. 2, the “type” if DNS query should be “0x0100 Standard query” and “Type A”, which means that it requests for IPv4 address. The query message does not contain any “answers”.

5) Two answers are provided and each answer contains a IP address of the domain name “www.umn.edu”.

6) From the fourth DNS response message in Fig. 2, we can know that the authoritative name server of the university of Minnesota is “ns-auth-1.umn.edu”.

7)



6.

a. On the Client-side, it may still go on sending packets to the server since the client has been in the established state. If the server still does not receive any packet before some timeout, the server will resend SYNACK(y,x).

b. Yes, this may happen because of some network latency. The server may know the duplicate SYN(x) is from a previously closed connection. The server may consider the client to initiate a three-way handshake and send SYNACK(y',x) to the client, where y' is the sequence number randomly chosen by the server.

c. After sending SYNACK(y',x) to the client in question b, the server will expect a package with the ACK number: y' + n, where n is the size of the data segment and y' is the sequence number randomly picked by the server. If the duplicate message ACK(x,y) has a y value not equal to y' + n, the server will reject to establish the connection. If y is equal to y' + n by chance, the server may consider the connection is established but this not very likely to happen because the server may not randomly pick the same sequence number two times.