Homework Assignment 2

1. (1pt) Under what circumstances can the HTTP response have empty body?

2. (1pt) Suppose Alice, with a Web-based e-mail account (such as Hotmail or Gmail), sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice's host to Bob's host. List the series of application-layer protocols that are used to move the message between the two hosts.

3. (2pt) List and explain the two different approaches for DNS resolving.

4. (2pt) Is it possible for an organization's Web server and mail server to have exactly the same alias for a host name (e.g., *foo.com*)? What would be the types for the RRs that contain the hostnames of the web and the mail servers?

5. (3pt) Suppose TCP operates over a 1 Gbps link.

    a. Assume TCP could utilize the full bandwidth continuously and the average packet size is 60 bytes, how long does it take for the TCP sequence numbers to wrap around completely?

    b. Suppose an added 32-bit timestamp field increments 1000 times during the wraparound time you found above. How long would it take for the timestamp to wrap around?

6. (3pt) Suppose you have the following 8-bit bytes: 01010011 01100110 01110100.

    a. What is the 1s complement of the sum of these 8-bit bytes? (Although TCP and UDP use 16-bit words in computing the checksum, you only need to consider 8-bit based checksum for this problem)

    b. Why use 1s complement of the sum instead of the sum itself as the checksum?

7. (6pt – including 2 extra pts) Consider two network entities, A and B, which are connected by a perfect bidirectional channel (i.e., any message sent will be received correctly; the channel will not corrupt, lose, or re-order packets). A and B are to deliver data messages to each other in an alternating manner: First, A must deliver a message to B, then B must deliver a message to A, then A must deliver a message to B and so on. If an entity is in a state where it should not attempt to deliver a message to the other side, and there is an event like `rdt_send(data)` call from above that attempts to pass data down for transmission to the other side, this call from above can simply be ignored with a call to `rdt_unable_to_send(data)`, which informs the higher layer that it is currently not able to send data. [Note: This

simplifying assumption is made so you don't have to worry about buffering data.]

Draw a FSM specification for this protocol (one FSM for A, and one FSM for B). Note that you do not have to worry about a reliability mechanism here; the main point of this question is to create a FSM specification that reflects the synchronized behavior of the two entities. You should use the following events and actions that have the same meaning as protocol rdt1.0 in Figure 3.9: `rdt_send(data)`, `packet = make_pkt(data)`, `udt_send(packet)`, `rdt_rcv(packet)`, `extract(packet,data)`, `deliver_data(data)`. Make sure your protocol reflects the strict alternation of sending between A and B. Also, make sure to indicate the initial states for A and B in your FSM descriptions.