Question 1: Client-Server vs. P2P

Microsoft distributes OS updates regularly. Suppose it has to release an update to the operating system that is 100 MB in size. The number of users it has to distribute the update to is N. The bandwidth coming out of Microsoft is unlimited, however the server pool they have can support a maximum of 1,000 simultaneous downloads. The clients are mixed, 50% of them have: 1Mbps download and 500 Kbps upload capacity. The remaining 50% of them have 5Mbps download and 500Kbps upload capacity. Assume that the P2P system is "perfect", i.e. all nodes can immediately start uploading at full speed.

For values of N being (i) 100,000

(a) The total time it takes until the last client has received the patch using a client-server solution residing at Microsoft,

(b) The total time it takes assuming a perfect Bittorrent style P2P distribution system

Question 2: Assume that a Circular DHT is designed to keep track of where content chunks are stored in a P2P network. If it is designed to support two failures at a time, including the failure of two adjacent neighbors in the overlay network, then specify the steps taken for inserting a new peer into the DHT. What are the different peers that a given node has to keep track of? What is the procedure to be taken by a node when its two adjacent peers fail?

Question 3: Reliable Data Transfer

A sender wishes to perform reliable communication with two receivers simultaneously. Upon receiving a packet, a receiver sends an ACK if the packet is received uncorrupted. Otherwise, it sends a NAK if the packet is corrupted. A receiver's acknowledgment can be corrupted as well. Each receiver's acknowledgment arrives at the sender as an uncorrupted ACK independently with probability p (i.e., After sending a packet to a receiver, the sender "knows" with probability p that the receiver successfully received the packet.) Assume that the sender can determine from which receiver an arriving acknowledgment belongs to (e.g., it can make the determination from the source address
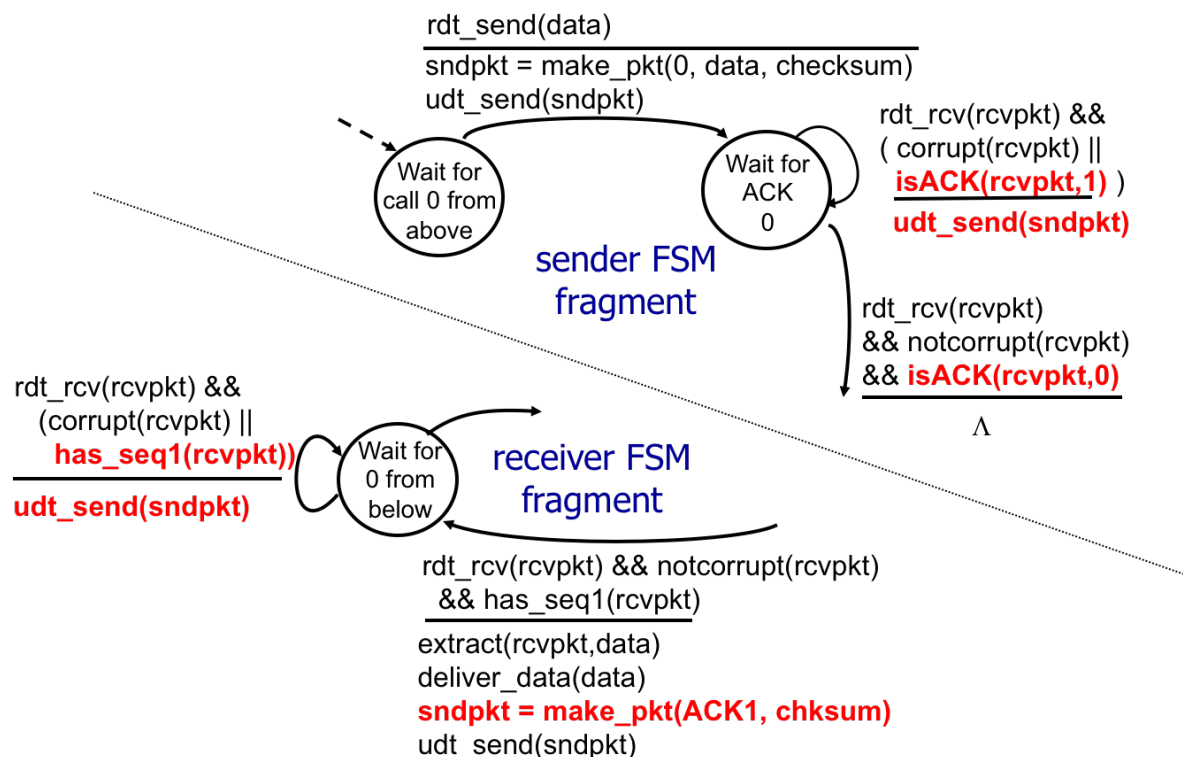
of the packet.)

Is it sufficient to have just two sequence numbers (i.e., just using sequence numbers 0 and 1 for packets) just like in the single-receiver case? Explain how two sequence numbers can be used or give a counter-example showing that two is not sufficient.

Question 4:

Shown below is the partial state machine for the sender and receiver for RDT 2.2. Please complete both the sender and receiver state machines for RDT 2.2. Note that the state machine is symmetrical, and therefore the remainder of the state machine would like somewhat like what you see here, except for the sequence number value being changed.

Good Luck!