CS 421: Computer Networks **SPRING 2009**

MIDTERM
April 9, 2009
120 minutes

Name: ______________________________

Student No: ______________________________

Show all your work very clearly. Partial credits will only be given if you carefully state your answer with a reasonable justification.

Q1	
Q2	
Q3	
TOT	

1)

a) (5 pts) Give at least three reasons why DNS is not implemented as a centralized database system, i.e., with a single centralized name server.

b) (5 pts) Suppose that you are developing a new application for the Internet which requires reliability but does not care about out-of-order delivery. Which transport layer protocol is more appropriate for this application? Fully justify your answer.

c) (5 pts) Suppose that an application uses the **Go-Back-N** receiver algorithm and wants to communicate with another application which uses the TCP sender algorithm. Is it possible that these two applications can reliably communicate with each other? Fully justify your answer.

d) (5 pts) Suppose that an application uses the **Selective Repeat** receiver algorithm and wants to communicate with another application which uses the TCP sender algorithm. Is it possible that these two applications can reliably communicate with each other? Fully justify your answer.

e) (5 pts) Why does the performance of the TCP Congestion Control algorithm degrade over wireless links where packets frequently experience bit errors?

f) (5 pts) Assume that two applications have a very large file to transfer, and they decide to open $n > 1$ parallel TCP connections for the file transfer. List one advantage and one disadvantage of using n parallel TCP connections from these two applications' point of view.

2)

a) A TCP sender is transmitting over a 1 Gbps ($1x10^9$ bits/sec) channel which has a 10 msec one-way delay. Assume further that a very large amount of data is transferred and no packers are lost or errored. The sender utilization (U_{sender}) is defined as the percentage of time the sender is busy transmitting bits (same as defined in your textbook). Remember that the receive window field in the TCP header is 2 Bytes long.

   i. (6 pts) Assume that window scaling is not used for this connection. What is the maximum value of U_{sender} possible for this connection?

   ii. (6 pts) Now suppose that the TCP connection uses a window scaling factor of 64. What is the maximum value of U_{sender} possible for this connection?

c) Consider a 1 Mbits/sec channel with a 10 msec one-way propagation delay, i.e., 20 msec roundtrip propagation delay. We want to transfer a file of size 6000 Bytes. Each packet has a total size of 625 Bytes including a 25 Bytes header, i.e., each packet contains 600 Bytes of data. When there is data to be transmitted, each packet contains the maximum number of bytes. Assume that ACK packets have negligible transmission times and there is a processing delay of 5 msec after a packet is fully received at the receiver until the transmission of the corresponding ACK is started. Assume that **every 7th data packet transmitted by the sender is lost, whereas no ACK packets are errored or lost**.

   i. (10 pts) Assume that **Go-Back-N** protocol is used with a window size of N = 8 segments. The timeout for each window is set to 40 msec starting from the beginning of each window. How much time is required to complete the transfer of the whole file and receive the final ACK at the sender?

   ii. (10 pts) Assume that **Selective Repeat** protocol is used with a window size of N = 8 segments. The timeout for each packet is set to 40 msec starting from the end of the transmission of the packet. How much time is required to complete the transfer of the whole file and receive the final ACK at the sender?

3)

a) An application running on Host A wants to send a very large file to another application running on Host B over a TCP connection. Assume that there are no packet losses or errors over this connection. The transmission rate of the link connecting host A to the Internet is given by R bps. Suppose that the application process running at Host A can send data into its TCP socket at a rate of S bps, and the application process running at Host B can read data from its TCP socket at a rate of T bps. Further suppose that both the TCP send buffer at Host A and the TCP receive buffer at Host B can hold only 1% of the file.

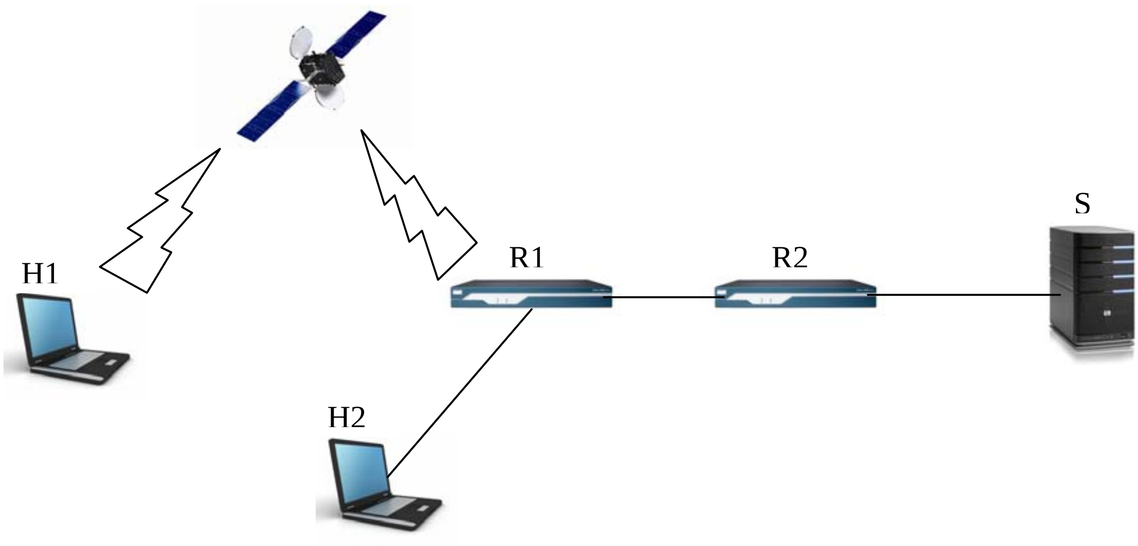
   i. (6 pts) Assume that S = 10 x R and T = R. What will prevent the application process running at Host A from continuously passing data to its TCP socket at rate S bps? Is it TCP Flow control algorithm, or TCP Congestion control algorithm, or something else? Fully justify your answer.

   ii. (6 pts) Assume now that S = R and T = R / 2. What will prevent the application process running at Host A from continuously passing data to its TCP socket at rate S bps? Is it TCP Flow control algorithm, or TCP Congestion control algorithm, or something else? Fully justify your answer.

b) (8 pts) Consider the following network. Assume that two separate applications running on hosts H1 and H2 want to transfer two very large files over two different TCP connections to server S. Assume there is no other traffic than these two TCP connections. The roundtrip delay between H1 and S is 200 ms and the roundtrip delay between H2 and S is 50 ms The transmission speed of the links are:

- H1-R1: 10 Mbps in each direction
- H2-R1: 100 Mbps in each direction
- R1-R2: 5 Mbps in each direction
- R2-S: 100 Mbps in each direction

Assume that packet losses occur only on the link R1-R2 and that the two TCP connections can, together, fully utilize the bandwidth on link R1-R2, i.e., $T_1 + T_2 = 5$ Mbps, where T_i is the throughput achieved by the TCP connection between Hi and S, i = 1, 2. Calculate the throughputs, T_1 and T_2, achieved by each TCP connection.



c) Consider the TCP round-trip time and timeout estimation algorithm as we discussed in class:

$$\text{EstimatedRTT} = (0.875 \times \text{EstimatedRTT}) + (0.125 \times \text{SampleRTT})$$

$$\text{DevRTT} = (0.75 \times \text{DevRTT}) + (0.25 \times |\text{SampleRTT} - \text{EstimatedRTT}|)$$

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 \times \text{DevRTT}$$

Suppose that currently EstimatedRTT = 20 ms and DevRTT = 10 ms. Assume that the next measured SampleRTT = 30 ms. Immediately after calculating the updated values, TCP transmits a new segment, and this segment experiences an RTT of 50 ms.

i. (5 pts) Assuming that neither this new data segment nor its ACK are lost, will this new segment timeout?

ii. (5 pts) An earlier version of TCP did not use DevRTT, and simply set TimeoutInterval = 2 x EstimatedRTT. Now answer the above question in i.

d) (8 pts) Assume a congestion feedback model for a system composed of two flows sharing a bottleneck link with bandwidth R bits/sec where both connections have the same RTT. Each flow gets binary synchronous feedback in discrete time steps of one RTT. If the aggregate consumption of the two flows is above the bottleneck bandwidth R, both senders receive a congestion notification signal (**CN**), otherwise they receive no CN. The flows use the following congestion control scheme: When no CN is received in a time step, each sender increases its window to βW, where W is the current window size and $\beta > 1$, e.g., $\beta = 1.01$. On the other hand, for each congested time step, i.e., when senders receive CN, each sender decreases its window by a fixed number of segments, e.g., by 5 segments. We can call this algorithm as *Multiplicative Increase-Additive Decrease* (**MIAD**). **Prove or disprove** that MIAD achieves a fair allocation of bandwidth between the flows, i.e., each flow getting R/2, by using graphical arguments similar to the one we made in class in showing that TCP's AIMD algorithm is fair. Use the following figure for showing the evolution of the throughputs. Assume that the initial throughputs achieved by the two connections correspond to "A".