CS 421: Computer Networks **FALL 2008**

MIDTERM
November 20, 2008
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) In the TCPServer.java code that we discussed in class, why the server does not close the connection socket immediately after it sends the reply message?

b) (5 pts) Is it possible to have two UDP segments with the same destination port numbers to be delivered to two different application processes? Why or why not?

c) (5 pts) Is it possible to have two TCP segments with the same destination port numbers to be delivered to two different application processes? Why or why not?

d) (5 pts) Suppose that we have a communication system with a high loss rate and a large round trip time. Which one will you prefer: Go-Back-N or Selective Repeat? Why? Explain your reasoning.

e) (5 pts) List at least two disadvantages of using extremely large window sizes in Go-Back-N and Selective Repeat.

f) (5 pts) Give a communication scenario under which Go-Back-N is more efficient than Selective Repeat, i.e., Go-Back-N requires less time to transfer a given file compared to Selective Repeat. In the scenario, assume that the same window size is used by both protocols and exactly the same set of data and ACK packets are lost or errored for both schemes.

2)

a) Assume that the bandwidth of a connection is 100 Mbps ($100x10^6$ bits/sec). Each data packet is 2,500 Bytes long and the ACK packets have negligible lengths. Assume that you use the Selective Repeat protocol with a window size of 100 segments.

   i. (6 pts) What should be the maximum round-trip delay in order to achieve a bandwidth utilization of 100%, i.e., $U_{sender} = 1$?

   ii. (4 pts) What is the minimum number of bits necessary to represent the sequence number for proper operation of the Selective Repeat protocol?

   iii. (4 pts) Assume that the round-trip delay is equal to the round-trip propagation delay, i.e., all other delays are negligible. Using the round-trip delay computed in part i., compute the maximum distance between the two ends of the connection in order to achieve 100% utilization.

b) Consider a 1 Mbits/sec channel with a 20 msec one-way propagation delay, i.e., 40 msec roundtrip propagation delay. We want to transfer a file of size 9600 Bytes. Each packet has a total size of 1250 Bytes including a 50 Bytes header, i.e., each packet contains 1200 Bytes of data. When there is data to be transmitted, each packet contains the maximum number of bytes. Assume that ACK packets have negligible lengths and there is a processing delay of 2 msec after a packet is fully received at the receiver until the transmission of the corresponding ACK is started. Assume that **no data packets are lost or errored, but every 4^{th} ACK crossing the reverse channel, i.e., receiver to sender, is lost**.

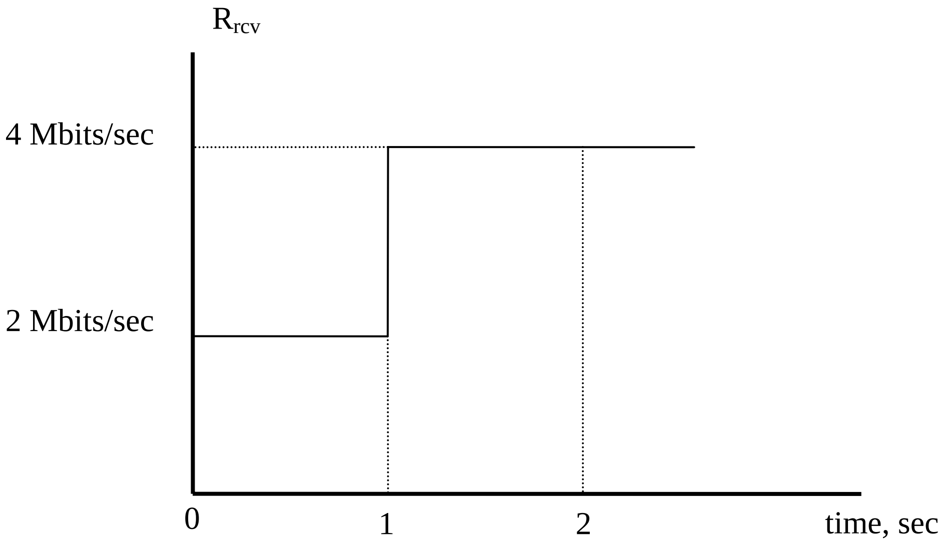
   i. (13 pts) Assume that **Go-Back-N** protocol is used with a window size of N = 6 segments. The timeout for each window is set to 60 msec starting from the beginning of each window. How much time is required to complete the transfer of the whole file and receive the final ACK at the sender?

   ii. (13 pts) Assume that **Selective Repeat** protocol is used with a window size of N = 6 segments. The timeout for each packet is set to 60 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) (6 pts) Assume that a TCP sender A first measures the actual round trip time as 20 msec for the first packet, and A thus sets its estimated round trip time to 20 msec. The next actual round trip time that A measures is 100 ms. In response, A increases its estimated round trip to 40 msec. The next actual round trip time that A measures is 20 msec. What is the next estimated round trip computed by A?

b) (8 pts) Suppose that R_{rcv}, the rate at which bits are arriving to a TCP receive buffer, is given in the following figure as a function of time. The application process at the receiver is removing bits from the receive buffer at the constant rate of 3 Mbits/sec ($3x10^6$ bits/sec). Assume that the receive buffer is initially empty and it has a fixed size of 150,000 Bytes. What is the value of the Receive Window advertised by the receiver at t = 2 sec?



c) (8 pts) Assume that a TCP connection uses window scaling that allows receive window of up to 1 MBytes. Suppose that the TCP connection runs over a 1 Gbits/sec ($1x10^9$ bits/sec) link with a round-trip propagation delay of 100 msec and there is no other traffic on the link. We want to transfer a 255 KByte file over this TCP connection. Assume that TCP receive buffer is 1 MBytes, and the application process removes the data as soon as it is placed in the receive buffer. Assume further that TCP connection uses 1 KByte segments and no loss event occurs during the entire file transfer (for simplicity, assume that the whole file is transferred using 255 segments). Further assume that the slow start threshold (*ssthresh*) at the beginning of the connection is infinitely large. Ignore all processing and queueing delays. What is the total time from the beginning of the transmission of the first data segment until the sender receives the ACK for the last data segment?

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 a simple 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 $\beta > 1$. On the other hand, for each congested time step, i.e., when senders receive CN, each sender decreases its window to αW, where $0 < \alpha < 1$ (typical values can be: $\alpha = 0.5$, $\beta = 1.01$). We can call this algorithm as *Multiplicative Increase-Multiplicative Decrease* (**MIMD**). **Prove or disprove** that MIMD 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".