CS 421: Computer Networks **FALL 2007**

MIDTERM I
November 7, 2007
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) An application can choose UDP as the transport layer protocol instead of TCP because UDP provides finer application layer control than TCP for determining what data is sent in a segment and when. In answering the following questions, recall the socket programming examples using TCP and UDP that we discussed in class.

   i) (5 pts) Why does an application using UDP have more control (compared to TCP) of what data is sent in a segment?

   ii) (5 pts) Why does an application using UDP have more control (compared to TCP) on when the segment is sent?

b) (6 pts) HTTP uses TCP as the transport layer protocol. Suppose HTTP was configured to use UDP instead. Assume that within your Web browser you click on a link to obtain a very small base HTML file from some server. This file references 5 very small objects each located on that same server. Let RTT be the round-trip propagation delay between your host and the server. Ignoring transmission times and all other network delays, what is the minimum number of concurrent parallel HTTP requests that **non-persistent HTTP using UDP** would need so that it retrieves the base HTML file and all the objects faster than **persistent HTTP with pipelining under TCP**? **Briefly explain/justify your answer for full credit.**

c) (6 pts) List one advantage and one disadvantage **each** of using Cumulative ACKs, Selective ACKs and NAKs.

d) (6 pts) Give two possible scenarios under which a TCP segment is retransmitted although there are no data or acknowledgement packet losses at all.

e) (6 pts) Suppose an application program running on a host wants to establish the reliability of a link by sending packets and measuring the percentage that are received. Explain the difficulty of doing this if the application program is performing these measurements over a TCP connection.

2)

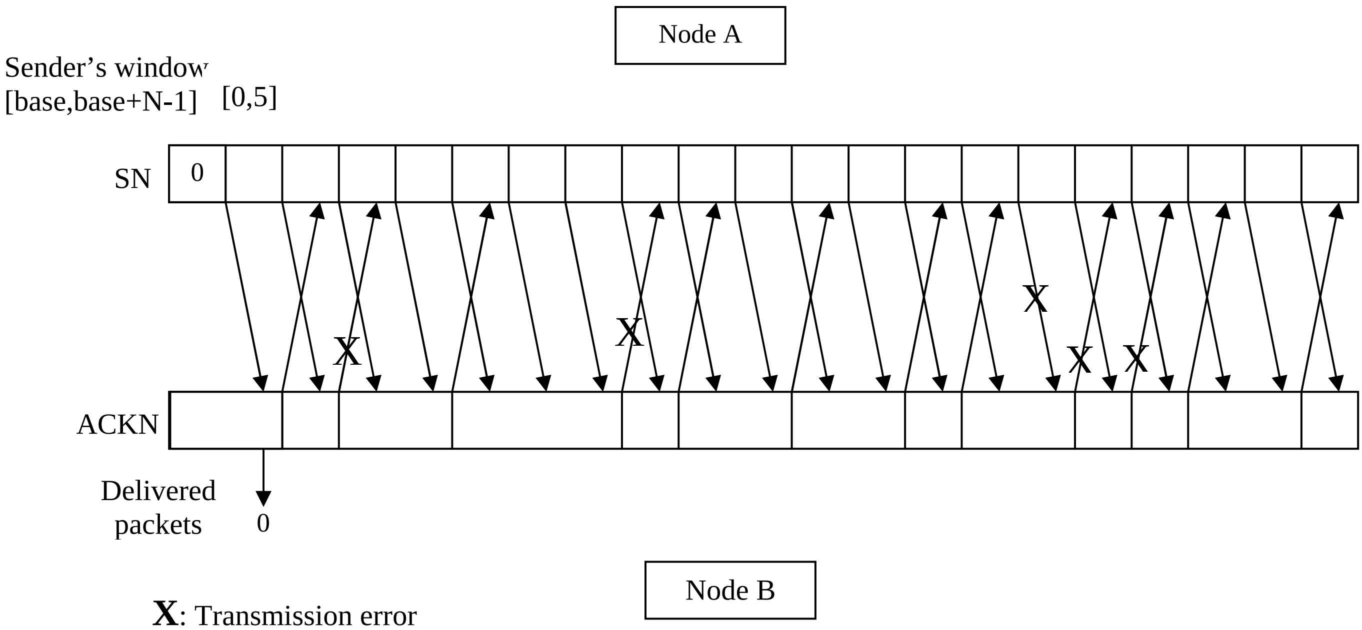
a) Assume that the bandwidth of a connection is 10 Mbps ($10x10^6$ bits/sec) and the round-trip propagation delay for the connection is 30 msec. Assume that each data packet is 2,500 Bytes long and the ACK packets have negligible lengths. **Separately answer the following questions for both Selective Repeat and Go-Back-N.**

   i. (8 pts) Assuming that no packets are lost, what should be the minimum window size (in packets) in order to achieve a bandwidth utilization of 100%, i.e., $U_{sender} = 1$?

   ii. (6 pts) What is the minimum number of bits necessary to represent the sequence numbers for proper operation using the window size that you calculated above?

b) (15 pts) Consider a 10 Mbps channel with a 20 msec roundtrip propagation delay. We want to transfer a file of size 18000 Bytes. Each segment has a total size of 1250 Bytes including the 125 Bytes header, i.e., each segment contains 1125 Bytes of data. When there is data to be transmitted, each segment contains the maximum number of bytes. Assume that ACK segments have negligible lengths and there is a processing delay of 1 msec after a segment is fully received at the receiver until the transmission of the corresponding ACK is started. We use **Selective Repeat** protocol with a window size of N = 8 segments. Assume that **every 8th segment** crossing the forward channel is lost while ACKs are not lost or corrupted. Assume further that the processing delay at the sender after an ACK is received is negligible. How much time is required to complete the transfer of the whole file and receive the final ACK at the sender? Assume that the timeout for each segment is set to 30 msec starting from the end of the transmission of the segment.

c) (15 pts) Consider the use of Go-Back-6 protocol for communication from Node A to Node B. Assume that when the sender reaches at the end of the window, i.e., all packets in the window are sent but not ACK'd, the sender goes into timeout, i.e., it goes to the beginning of the window and retransmits all packets in the window, as if the timeout occurred. In the following figure, indicate the sequence number (SN) for packets sent from A to B, the ACK number (ACKN) for packets sent from B to A, the times and SN of the packets at B delivered to the application layer, and the window kept at A. **Note that packets received during the transmission of another packet will be immediately processed, but the corresponding action, e.g., update of SN/ACKN, will take effect with the start of transmission of the next packet.**



3)

a) Suppose a TCP connection experiences round-trip times (RTT) of 20 msec for 30% of its packets, 40 msec for 30% of its packets, 100 msec for 20% of its packets and no ACK is received for 20% of its packets. Suppose no packets are actually lost. Assume that the estimated RTT (according to the exponential weighted moving average) is equal to the true (ensemble) average of RTT, i.e., EstimatedRTT = average value of RTT = E[SampleRTT].

   i) (7 pts) Assume that the timeout is set to **2 (two)** times the estimated RTT (as in the original version of TCP), i.e., TimeOut = 2 x EstimatedRTT. What fraction of the packets will be assumed lost by the TCP sender?

   ii) (7 pts) Currently used versions of TCP estimate both the mean and the mean deviation as we discussed in the class (the mean deviation is the average absolute distance of RTT samples from the estimated RTT), and sets the timeout to the estimated mean (estimatedRTT) plus **3 (three)** times the estimated deviation (devRTT), i.e., TimeOut = EstimatedRTT + 3 x devRTT. Assume that the TCP connection uses this new method, and the estimated RTT and deviation are equal to their true (ensemble) values, i.e., EstimatedRTT = E[SampleRTT] and devRTT = E[|SampleRTT-EstimatedRTT|]. What fraction of the packets will be assumed lost by the TCP sender?

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 200 Kbits/sec (2×10^6 bits/sec). Assume that the receive buffer is initially empty at t = 0, and it has a fixed size of 50,000 Bytes. Considering the TCP flow control algorithm, what is the value of the Receive Window advertised by the receiver at t = 2 sec?