CS 421: Computer Networks **SPRING 2006**

MIDTERM I
March 23, 2006
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.

1)

a) (6 pts) Explain why an application developer would prefer a reliable transport layer service over a transport layer service that provides no reliability. For what type of applications would a developer prefer an unreliable transport layer service? Why?

b) (6 pts) Explain how the byte range retrieval requests in HTTP 1.1 are used. Give two examples of practical applications you saw in this course that use byte range retrieval requests.

c) (5 pts) Suppose a byte range header is used in an unconditional GET request. What is the status code of the HTTP response (either the name or number of the status code will be accepted as an answer) contained in the HTTP reply corresponding to this request?

d) (6 pts) Suppose a DNS resolver, e.g., a browser, at the host windows.microsoft.com tries to get the IP address of a non-existing host, e.g., nonexist.bilkent.edu.tr. How can the local DNS server for the domain microsoft.com inform the DNS resolver that the given host is non-existing? Under what conditions can this resolver get the IP address of an existing host, e.g., firat.bilkent.edu.tr, faster than getting the information about the above-mentioned non-existing host?

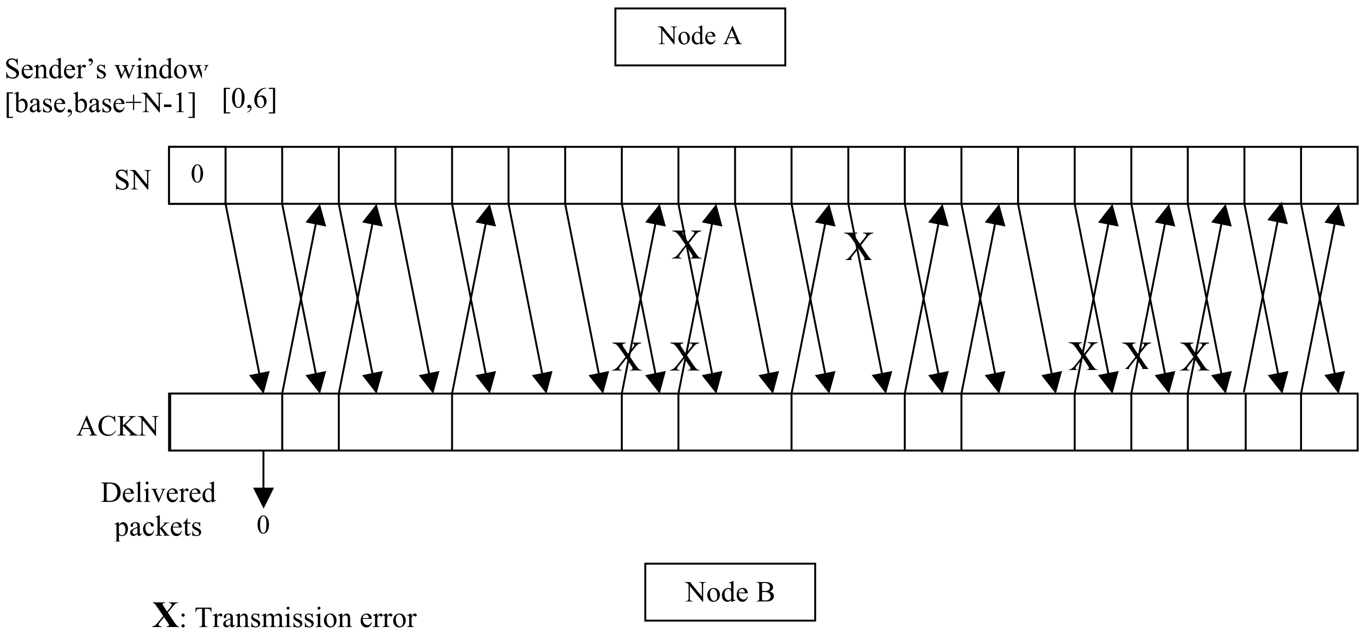
e) (6 pts) What is an overlay network in a P2P file sharing system such as Gnutella? What are the endpoints of the edges in the overlay network?

f) (6 pts) Suppose there are N active peers in the Gnutella network, and each pair of peers has an active TCP connection. Suppose that these TCP connections pass through a network with a total number of M routers. How many nodes and edges are there in the corresponding overlay network?

2)

a) (10 pts) 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 13500 Bytes. Each packet has a total size of 1625 Bytes including the 125 Bytes header, i.e., each packet contains 1500 Bytes of data. When there is data to be transmitted, each packet contains the maximum number of bytes. Assume that ACK packets are of 125 Bytes long and there is a processing delay of 1 msec after a packet is fully received at the receiver until the transmission of the corresponding ACK is started. **Go-Back-N** protocol is used with a window size of N = 4 packets. Assume that every 6^{th} packet crossing the forward channel is lost while ACKs are not lost or corrupted. Assume 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 packet is set to 3msec larger than the **total** round-trip delay (including transmission, propagation and processing delays).

b) (15 pts) Consider the use of Go-Back-7 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 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.**



c) (10 pts) You are hired to design a reliable transport protocol that uses Selective Repeat. This protocol will run over a 100 Mbits/s network. The distance between the two end systems that will use this protocol is 1,500 km and the speed of propagation is 3×10^5 km/s. All other delays such as processing and store-and forward delays at intermediate nodes can be ignored. Assume that the segments are 10,000 Bytes long and the maximum segment lifetime, i.e., the maximum delay that a segment can experience in the network, is 60 seconds. What should be the size of the Receive Buffer (in Bytes) that you need to allocate in your protocol so that the available bandwidth can be fully utilized? How many bits should you allocate for the Sequence Number field of your protocol header so that the protocol will run correctly under any circumstances?

3)

a) (10 pts) Normally if a TCP host receives a segment with sequence number 1000 and length 1460, it will respond with an ACK of 2460, since that is the sequence number of the next byte anticipated. Suppose instead that the receiver sent two ACKs, one with the ACK number field being 2000 and the second with it being 2460. Is there any harm in this "divided ACK"? What would the impact of the divided ACK be on the TCP congestion control algorithm?

b) (10 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, 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 both 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.

c) (10 pts) Assume that the congestion window of a TCP flow was 22 segments long when a *timeout* occurred. Assume that there are no segments or acknowledgments of this flow that were in transit when the timeout occurred. The round trip delay for the flow is fixed and is equal to 60 msec and the transmission time for a segment is 5 msec. The receive window is fixed at 100 segments for the entire duration of the connection. What is the minimum time the flow will spend in the Slow Start phase after the timeout before reaching the Congestion Avoidance phase?