CSE 123A Final Exam

August 3, 2006

Name: ______________________________ Email: _____________________________

Be sure to read questions carefully and answer all parts. Use complete sentences and explain your answers. One word responses will not be given credit. If you use any specific terms or acronyms, give a brief description of what they mean. You should be able to answer the questions in a few sentences or less. There is no need to write an essay. Remember to flip the pages over (there are questions on the back of most pages.) Good luck!

Question 1	**/ 20**
Question 2	**/ 20**
Question 3	**/ 20**
Question 4	**/ 20**
Question 5	**/ 20**
Question 6	**/ 20**
TOTAL	**/ 120**

1. Layering, Coding, and Error Detection (20 points)

a) List the layers in the Internet layering model. Describe what the purpose of each layer is. For layers 1-4, give an example of a protocol that we studied in class. (7 points)

1. Physical layer (e.g Manchester, NRZ, NRZI, 4B/5B): bit transmission and encoding
2. Data-link layer (e.g Ethernet): message framing, error detection, shared media access
3. Network layer (IP): routing and addressing
4. Transport layer (e.g. TCP, UDP): reliable in-order transmission, flow control, congestion control, application demultiplexing
5. Presentation (ok if skipped)
6. Session (ok if skipped)
7. Application (e.g. HTTP): application logic and protocols

b) We looked at several different algorithms for doing synchronous coding. Choose one of them, and briefly describe how it works. If there are any limitations to the scheme you choose, be sure to mention those as well. (7 points)

Acceptable answers include descriptions of NRZ, NRZI, Manchester, or 4B/5B. In all cases, there should be a discussion of the weaknesses (such as strings of 0s or 1s for NRZ, or inefficiency for Manchester).

c) Briefly describe how two-dimensional parity is used for error detection. Does this error detection technique catch all bit errors? (6 points)

First use 1-dimensional (simple) parity to add one bit to a 7-bit code to ensure an even/odd number of 1s. Then add the 2nd dimension of parity by adding an extra byte to each frame. This byte will have bits that are set to ensure even/odd number of 1s in that position across all bytes in frame. This approach catches all 1-, 2- and 3-bit and most 4-bit errors.

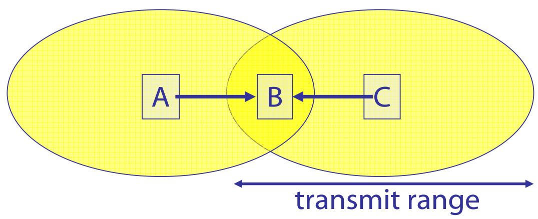
Name: ___________________________________

2. Media access, Hubs, Bridges (20 points)

a) What is channel partitioning? Give an example of one way to do channel partitioning. Why is it not well suited for random access usage? (7 points)

Channel partitioning involves sharing a channel among different hosts. We looked at 3 different approaches in class, including FDMA, TDMA, and CDMA. These techniques are not well suited for random access usage because it is unlikely that a single host is sending or receiving data at all times. So even though FDMA, TDMA, and CDMA share the channel evenly, they do not fully utilize the available link capacity when the system is not fully loaded, and therefore is inefficient for random access usage.

b) Briefly describe the hidden terminal problem in wireless media access. How does RTS/CTS overcome the hidden/exposed terminal problem? (7 points)



In this diagram, A and C can both send to B, but they can't detect each other. A is called a hidden terminal for C (and vice versa). This makes it impossible to detect collisions in all cases. RTS/CTS overcomes this problem because C will first transmit a RTS to B, and B will respond with a CTS. Although A can't detect the RTS, A can detect the CTS, and therefore A will backoff and wait before transmitting to B to avoid the collision.

c) In a LAN constructed using learning bridges, it is possible for there to be multiple paths to a single LAN. When this occurs, describe (at a high level) the technique that bridges use to determine which path to use when forwarding packets. (6 points)

Learning bridges use the spanning tree algorithm to create a virtual tree over the topology. The tree includes the shortest path to all LANs, and does not include any cycles. Learning bridges use this tree to generate a forwarding table which indicates how to forward packets efficiently. The bridges also implement selective forwarding.

3. SRMP, TCP, and UDP quickies (20 points)

a) An Internet Mail server accepts incoming TCP packets destined to port 25. How does it distinguish between different packets belonging to *different* TCP sessions (remember that different sessions can originate from the same host)? (3 points)

The combination of a client's IP address and its source port is enough to uniquely discriminate between different sessions arriving at the same Mail server.

b) How does TCP distinguish between different packets in the *same* session? (3 points)

Within the same session, each byte is represented by a unique sequence number.

c) Why is TCP well suited for applications like HTTP, IMAP, and POP (email)? (3 points)

These applications require reliability and do not have any limitations with respect to timeliness of data delivery. Thus the overhead (or delay) incurred from possible retransmissions that help ensure reliability do not degrade the performance of the application significantly.

d) For what type of applications is UDP best suited? (3 points)

UDP is best suited for applications that do not require reliability, and therefore do not need to "pay" the overhead for retransmissions. Examples include DNS, NTP, and streaming media applications.

Name: ______________________________

e) In Project 2, you implemented half of a reliable transport protocol, srmp. In it, the principal API was srmp_send, which had three parameters – the data packet being sent, the length of the data packet, and the srmp_send_ctrl_blk. In turn, the srmp_send_ctrl_blk structure contained a number of fields, including, NBE (Next Byte Expected by Client), LFS (Last Frame Sent), ESWS (Effective Send Window Size), LAR (Last ACK Received) and AdvWin (Advertised Window). When an acknowledgement packet is received by the sender, LAR is set to equal the packet's sequence number field and AdvWin is set to equal its advertised window field. However, the other srmp_send_ctrl_blk fields are calculated dynamically. Using only the variables described above, explain (these answers should be very short):

i. When srmp_send() is called, how is the sequence number of the packet calculated? (2 points)

   Seqno = NBE

ii. How is LFS updated? (2 points)

   LFS = Seqno + length

iii. How is NBE updated? (2 points)

   NBE = NBE + length

iv. How is ESWS updated? (2 points)

   ESWS = AdvWin - (LFS – LAR)

4. Congestion Control (20 points)

a) A sender and a receiver are communicating using a TCP-based protocol in which data is sent immediately after connection. The bandwidth of the link is 100Mbps, the round-trip time is 10ms and the maximum packet/segment size is 1500 bytes (ignore header overhead). Draw a packet exchange diagram showing how long it takes to transfer 15000 bytes. (10 points)

This is a TCP based protocol, so we use slow-start.

1st RTT: SYN + SYN/ACK
2nd RTT: 1500 bytes of data + ACK
3rd RTT: 3000 bytes of data + two ACKs
4th RTT: 6000 bytes of data + four ACKs
5th RTT: 3500 bytes of data + three ACKs

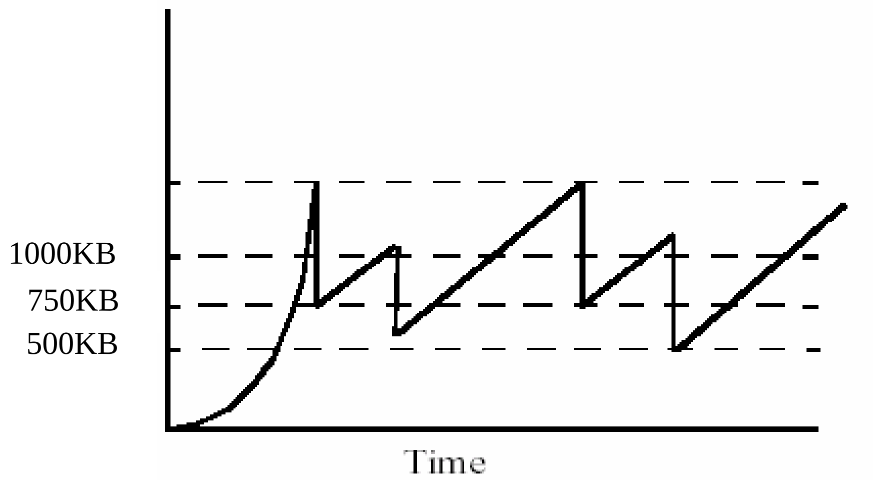
[optional] 6th RTT: FIN handshake

b) The fast recovery optimization is designed to allow TCP to keep packets flowing after a loss. Consider the TCP transfer in the previous question. If packets were lost during that transfer could fast recovery come into play? Explain why or why not. (5 points)

[This was the hardest question on the test. I was VERY generous with partial credit.] Fast recovery will not come into play in this example because the window size is never larger than 4 packets. Thus, even if the first packet is lost, there will only be enough duplicate acknowledgements to trigger fast retransmit, not enough to help with fast recovery.

Name: ________________________________

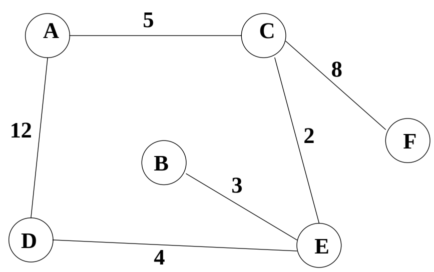
c) The graph below plots the congestion window for a particular TCP connection over its lifetime. This TCP implements slow start, fast retransmit and fast recovery and the receiver always has ample buffer (i.e. flow control isn't an issue). Mark an X on the Time axis of the congestion window plot whenever the sender believes that a packet has been lost. If necessary explain how you determined those points. (5 points)



There should be an "X" at each vertical line.

5. Routing (20 points)

a) For the network shown, complete the table below showing how the link-state algorithm builds the routing table for node D. (15 points)



	Confirmed	Tentative
1.	(D, 0, -)	(A, 12, A) (E, 4, E)
2.	(D, 0, -) (E, 4, E)	(A, 12, A) (B, 7, E) (C, 6, E)
3.	(D, 0, -) (E, 4, E) (C, 6, E)	(A, 11, E) (B, 7, E) (F, 14, E)
4.	(D, 0, -) (E, 4, E) (C, 6, E) (B, 7, E)	(A, 11, E) (F, 14, E)
5.	(D, 0, -) (E, 4, E) (C, 6, E) (B, 7, E) (A, 11, E)	(F, 14, E)
6.	(D, 0, -) (E, 4, E) (C, 6, E) (B, 7, E) (A, 11, E) (F, 14, E)	

b) How does poison reverse with split horizon solve the count-to-infinity problem in the distance vector algorithm? Are there any limitations to this approach? (5 points)

Split horizon stops a host from advertising a destination through its next hop. Poison reverse takes this one step further and requires the host to sends negative information (distance of infinity) when advertising a destination through its next hop. The limitation is that it doesn't always solve the count-to-infinity problem in all cases.

Name: ________________________________

6. Routing, Router design, and QoS (20 points)

a) Why is path vector better suited for inter-domain routing than distance vector or link state? (5 points)

It is more scalable, and it only exchanges reachability info, so the different domains do not need to agree on a global metric.

b) Briefly discuss the tradeoff between source-based tree routing and shared-tree routing in multicast. (5 points)

Source-based tree routing is more efficient, but required the routers to store state based on every source and every group. Shared-tree routing is less efficient, but requires the routers to store less state (just state for each group) and thus is more scalable.

c) What problem does virtual output queuing within input queues solve in router design? (5 points)

Virtual output queuing solves the head of line blocking problem.

d) When discussing algorithms for QoS, why does DiffServ provide better scalability than IntServ? (5 points)

DiffServ requires the routers to store less state than IntServ. IntServ requires per-flow state, while DiffServ required per-aggregate state.