

MAKEUP Midterm Exam
CMPSCI 591: Computer Networks
Fall 2000
Prof. Jim Kurose

Instructions:

- **Please use two exam blue books** – answer questions 1, 2 in one book, and the remaining questions in the second blue book.
- Put your name and student number on the exam books NOW!
- The exam is closed book.
- **You have 80 minutes** to complete the exam. **Be a smart exam taker** - if you get stuck on one problem go on to another problem. Also, don't waste your time giving irrelevant (or not requested) details.
- The total number of points for each question is given in parenthesis. There are 100 points total. An approximate amount of time that would be reasonable to spend on each question is also given; if you follow the suggested time guidelines, you should finish with 10 minutes to spare.
- Show all your work. Partial credit is possible for an answer, but only if you show the intermediate steps in obtaining the answer.
- Good luck.

Question 1: "Quickies" (25 points, 15 minutes)

- What is meant by the term *statistical multiplexing*?
- What are two key differences between a *datagram network* and a *virtual-circuit network*?
- Consider an http client that wants to retrieve a WWW document at a given URL. The IP address of the http server is initially unknown. The WWW object at the URL has one embedded GIF image that resides at the same server as the original object.
 - What transport and application layer protocols besides http are needed in this scenario?
 - Suppose that the time needed to contact and receive a reply from any server (for any protocol) is RTT. How many RTT's are needed from when the user first enters the URL until the complete document is displayed? Assume that non-persistent http is used. Consider the delays of all protocols in your answer, not just those of http.
- How are ports numbers used by UDP in demultiplexing incoming segments?
- Which protocol – Go-Back-N or Selective-Repeat - makes more efficient use of network bandwidth? Why?

Question 2: Transport Layer Potpourri (24 points, 20 minutes)

- Suppose the sender and receiver in a pipelined reliable data transfer protocol have a window of size N . Suppose the sequence number of the segment at the base of the window at the receiver is x . What is the possible range of sequence numbers in the sender's window? Justify your answer.
- Suppose that a reliable data transfer protocol such as TCP has an RTT estimate that is far too small. What is the consequence of having too small an estimate of the RTT? Now suppose the RTT estimate is far too large. What is the consequence?
- What is the role of the *threshold* in TCP congestion control? How does it change over time?
- Consider two TCP senders. They are at different sending hosts and go to different destinations, but pass through a *common* bottleneck link (that is the only bottleneck link on either of their paths). What does it mean to say that TCP provides fair sharing of bandwidth at the bottleneck link? Suppose the RTTs of the two connections are very different. Is TCP "fair" in this case? Justify your answer.

Question 3: A distributed transaction processing system (26 points, 20 minutes)

Consider a distributed transaction processing system of a client and a remote server. The client receives transaction requests from local users. These transaction requests must be communicated to the server, which will execute the transaction request and return the result of the transaction request. (You can think of a transaction as requesting or updating an account balance from the server database, and the response containing the resulting balance. *The client and server communicate over a medium that can lose and delay messages; the maximum delay in the medium is not known.* The medium will *not* corrupt or reorder messages.

The client should receive requests from local users (via the event *callbyuser(request)*) and return results to users (via the event *returndatatouser(data)*) in the order in which the requests were generated. The server receives messages from the client via the *messagefromclient(clientmsg)* event , executes a transaction via a call:

result = execute(clientmsg)

and sends messages to the client via the *mesagetoclient(servermsg)* event, where *clientmsg* and *servermsg* are messages (that you define) sent from the client and server, respectively. The client receives messages from the server via the *messagefromserver(servermsg)* event.

Give a FSM description of the client and server. Describe the format of the messages sent from client-to-server and from server-to-client. Your protocol should be minimalist in the sense that it should not contain any functionality that is not strictly needed to meet the above requirements.

Question 4: Routing Algorithms (25 points, 15 minutes)

Consider the network shown below.

- Show the operation of Dijkstra's (Link State) algorithm for computing the least cost path from **F** (the *rightmost* node in the figure below!) to all destinations. Also, explicitly list the shortest path routes from E to all destinations that are the result of the algorithm's computation.
- Show the distance table that would be computed by the distance vector algorithm in B. Note: you do *not* have to run the distance vector algorithm; you should be able to compute the table by inspection.

