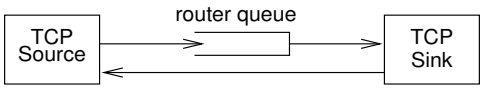
cmsc 417 F02 **Exam 1** **Name:** _____

If you want your exam/project scores displayed on class page by last five digits of your SID, sign below and give the last five digits of your SID: _____

Total points 30. Total time 70 mins. 3 problems over 3 pages. No book, no notes, no calculator.

1. [15 pts]



This problem deals with TCP congestion control using the window-based model in Note 5 and the scenario shown above. Assume the following:

- Source uses the standard TCP congestion control (slow-start, additive increase, multiplicative decrease) except that timeout value is constant at 40 ms.
- Packet payload is 1 KB. Source has initial cw (congestion window size) of 1 packet and initial sst (slow-start threshold) of 8 packets. The router queue holds at most 3 packets and serves a packet in 10 ms.
- Roundtrip time consists entirely of the delay encountered in the router queue (i.e., zero delay encountered by acks). Queue overflow is the only cause of packet loss. No other traffic enters the router queue.

Determine the following for a 26 KB file transfer: the transfer time in ms (from start to everything acked); the number of windows sent; and, for each window send, which 1 KB chunks are sent assuming the file consists of chunks 0, 1, $\cdots$, 25) (answers for the first two window sends are shown below:

```
window send    file chunks sent
  1              0
  2              1, 2
  3
  .
  .
```

2. [**10 pts**] Consider the following variation of the sliding-window data transfer protocol in Note 4:

- Sequence numbering is done per byte (as in TCP).
- Data messages have the form $(\mathsf{D}, \mathsf{data}, \mathsf{len}, \mathsf{cj})$, where data is an array of bytes, len is the number of bytes in data, and cj is the modulo-N sequence number of the start byte in data. (We omit source id and sink id fields in messages.)
- Cumulative ack messages have the form $(\mathsf{ACK}, \mathsf{cj}, \mathsf{w})$, where cj is the modulo-N sequence number of the next insequence byte expected at the sink and w is the receive window size.
- Selective ack messages have the form $(\mathsf{SACK}, \mathsf{cj}, \mathsf{len})$, where cj is a modulo-N sequence number and len is a positive integer. They are sent by the sink to indicate that it has received and is buffering the out-of-sequence data segment starting at cyclic sequence number cj and of length len.
- The source entity maintains the variables na, ns, ng, sw, sbuf as usual, except that each entry sbuf[i] has two fields: sbuf[i].data, containing a data byte; and sbuf[i].ackd, a boolean that is true if and only if sbuf[i].data has been selectively acked (i.e., by a SACK message).

Give the code for the receive SACK event at the source entity. Specifically, indicate how the source variables are updated upon reception of $(\mathsf{SACK}, \mathsf{cj}, \mathsf{len})$.

Your code must not exceed 15 lines in total. Program elegance counts.

3. [**5 pts**] Consider a client-server application that runs on TCP and involves users 1, 2, and 3, where user 1 is a client and users 2 and 3 are servers. Assume the following:

- For any two users i and j, IP provides a perfect channel from i to j of very high bandwidth and constant end-to-end message delay $T_{i,j}$. (E.g., $T_{2,1}$ is the delay from user 2 to user 1.)
- The users execute at very high speed.

User 1 does the following:

- connect to user 2
- get the address of user 3 (a few octets) from user 2
- connect to user 3
- send a few octets of data to user 3
- ...

Determine the elapsed time from the instant when user 1 issues the connect to user 2 to the instant when user 1 gets the ack for the data sent to user 3. Your answer is a formula in the $\{T_{i,j}\}$.