HW #6

ELEN E4710 - Intro to Network Engineering
Fall 2004
Prof. Rubenstein

Due 12/6/2004

Homework must be turned in at the beginning of class on the due date indicated above. CVN students have one additional day. Late assignments will not be accepted.

1. 3 flows, \( f_1, f_2, \) and \( f_3 \) all pass through the same router \( R \). Packets from flows \( f_1 \) and \( f_2 \) arrive at \( R \) in bursts of 10 with a period of one second, packets from \( f_3 \) arrive at the router one at a time with a period of .1 seconds. Packets from all flows contain the the same number of bytes). Suppose packets from \( f_1 \) start arriving at \( R \) at time \( t = 0 \), flow \( f_2 \) at time \( t = 5 \), and flow \( f_3 \) at time \( t = 0.5 \). If each flow transmits a total of 100 packets, and the router processes a packet every 0.05 seconds, what are the completion times of the various flows when

(a) \( R \) uses FIFO (FCFS) queueing?
(b) \( R \) uses round robin?
(c) \( R \) uses weighted fair queueing where the slack (i.e., the maximum reserve that a flow can build up) is 20 packets?

2. Construct a FSM for a router that implements weighted fair queueing for two flows \( f_1 \) and \( f_2 \) where \( w_1 = 3, w_2 = 2 \), and the slack is 0.5 (where the processing of a packet from \( f_1 \) adds 1/3 to its virtual clock, and the processing of a packet from \( f_2 \) adds 1/2 to its virtual clock). Your state machine can depend on the following functions and events to simplify its design:

- \( IE() \): the queue is empty (prior to an arrival)
- \( H(i) = H(i) \) equals 0 when no packets from \( f_i \) are queued in the system, and equals 1 otherwise
- \( D() \): triggered when the router completes processing its current packet.
- \( A() \): triggered when an arrival to the router from \( f_i \) occurs (only needs to be used when the router is not processing any packets).
- \( P(i) \): process a packet from flow \( f_i \)

With these functions and events, you should build your FSM so that it indicates clearly whose packet should be processed next whenever such a decision needs to be made. (Hint: each state should indicate the current difference between the two flow’s clocks.) Suggestion: You should create functions that combine the above functions, e.g., let \( G(i) = (IE(), A(i))|D(), H(i) = 1) \).

3. Construct a Markov model for a round-robin queueing system that can store up to 4 packets for processing, where no more than 2 packets from a single flow is ever stored. Assume there are two flows in the system, where both flow’s packets are processed at rate \( \mu \), and flow \( f_i \)’s packets arrive at rate \( \lambda_i \). Label transitions with their transition probabilities.
4. Construct Markov models for each of two priority queueing systems, each processing packets from two flows. Each system can hold a total of 4 packets in total. A packet from \( f_1 \) should always be processed before any packets from \( f_2 \) in the queue. Assume both flow’s packets are processed at rate \( \mu \), and flow \( f_1 \)'s packets arrive at rate \( \lambda_1 \).

(a) In the first system, a packet from \( f_1 \) arriving to a full queue will replace a packet in the queue from \( f_2 \) (if one exists).

(b) In the second system, packets in the queue are not removed except when their processing is complete. Assume that a packet arriving from \( f_1 \) to a full queue will be turned away, even if there are packets from flow \( f_2 \) in the queue.