CS 177, Homework 8

Applications of Probability in Computer Science: Winter 2007

Due Date: Friday March 15th, 12 noon (via EEE)

Suggested Reading

- Section 7.4.3 and 7.4.4 in the text.

In this week's homework you will simulate in MATLAB a single-server queueing system as discussed in class.

Problem 1: Simulating an M/M/1 Queue

You are to implement an algorithm to simulate an M/M/1 queue with infinite queue size. The inputs to this algorithm will be as follows:

1. K, the number of customers to be simulated.
2. The arrival rate λ.
3. The service rate μ.
4. An initial state value for the random number generator `rand`.

It will produce as outputs the following:

1. A list of K arrival times of the K customers (the first arrival time can be 0 without loss of generality).
2. A list of K times at which service was begun for each input event (customer).
3. A list of the K departure times for each of the K customers.

```
function [times] = MM1_simulation(K, lambda, mu, state)
%
% function to simulate K customers in an M/M/1 queueing model
%
%  INPUTS:
%     K: number of customers
%   lambda: parameter of the exponential model for arrival times
%       mu: parameter of the exponential model for service times
%    state: positive integer = initial state (seed) for the random number generator
%
% OUTPUTS:
%    times: K x 3 vector of times for each customer, where
%      times(i,1) = arrival time of customer i
%      times(i,2) = start of service time for customer i
%      times(i,3) = departure time of customer i
%
%                                              CS 177 Homework 8, 2007

% code follows below......
```

You can begin your simulation code by generating K inter-arrival times and K service times. You can re-use and modify the code you wrote in Homework 7 for simulating from an exponential density.

To figure out when the service for each customer starts, you will need to sequentially go through the inter-arrival and service times and keep track of the state of the system. When a new event happens (when a new customer arrives or when the server finishes serving a customer) your simulation code will decide, based on the state of the system, what the next state of the system is (e.g., if a customer's service finishes, and there are customers in the queue, then the next service starts). Note that to track the state you need to keep track of both how many people are in the queue and whether or not anyone is being served. The only time the state can change is either when the next person arrives or when the server finishes serving someone—in between these events the state cannot change so you only need to look at state changes when these events occur. Implementing this will require some careful thought, but is relatively straightforward once you realize that everything proceeds sequentially based on state changes.

Problem 2: Simulating an M/M/1 Queue with Finite Queue Length

Repeat the last part, but now for a model with a finite queue of size $r - 1$ (so that no more than r customers can be in the system at any time). Note that with a finite queue that if the system is full when a new customer arrives, then that customer does not join the system at all and does not ever get service.

```
function [s,times] = MM1_finite_simulation(K, r, lambda, mu,state)
%
% function to simulate K customers in an M/M/1 queueing model
%   with finite queue size r-1
%
%  INPUTS:
%     K: number of customers
%     r: maximum number of customers that can be in the system
%   lambda: parameter of the exponential model for arrival times
%       mu: parameter of the exponential model for service times
%    state: positive integer = initial state (seed) for the random number generator
%
% OUTPUTS:
%    s = K x 1 status vector:
%      s(i) = 1 if the customer entered the system, = 0 if they did not
%    times: K x 3 vector of times for each customer, where
%      times(i,1) = arrival time of customer i
%      times(i,2) = start of service time for customer i
%      times(i,3) = departure time of customer i
%     and times(i,:) = 0  if s(i) = 0 (i.e., customer did not enter system)
%
%                                              ICS 178 Homework 6, 2006

% code follows below......
```

What to Submit

Submit your 2 MATLAB function files, plus any other MATLAB functions that you wrote and that are called by your functions, and 1 text document (Word or PDF) containing the answers to the questions below. Submit all files to the EEE online homework folder for Homework 8.

1. Documented code for each of the 2 MATLAB functions above. Please use comments liberally in your code!

2. For the infinite queue, histograms of both the waiting times for each customer and the amount of time spent in the queue for each customer, for $K = 10,000$ (or larger K if you wish) with 50 bins, for each of the following cases (a) $\lambda = 1, \mu = 10$, (b) $\lambda = 2, \mu = 10$, (c) $\lambda = 9.9, \mu = 10$.

3. For each of the 4 cases in part 2, calculate the mean value of the samples in the histogram and briefly discuss why they are different across the 3 cases.

4. Now compare the mean values in part 3 with the appropriate theoretical values of W and W_Q as discussed in class, and briefly discuss (i.e., they should be close enough that they validate both the theory and the correctness of your simulator).

5. Now for a finite queue, generate histograms of both the waiting times for each customer who entered the system, and the amount of time spent in the queue for each customer who entered the system, for $K = 10,000$ (or larger K if you wish) with (for example) 50 or 100 bins, with $\lambda = 0.9, \mu = 1$ for both (a) $N = 3$ and (b) $N = 15$.

6. For each of the 2 finite queue lengths, calculate the number of customers who did not enter the system (for each simulation), compare to the theoretically-predicted values, and briefly discuss the difference between theory and empirical numbers and between the 2 queue lengths.