Homework 4

Simulation and Performance Evaluation – University of Trento

DEADLINE: May 18, 2023

You can solve the following assignments using any programming language. You are allowed to use any utility functions you wish.

Exercise 1

Implement a discrete-event simulator for an M/M/1 queue-server system that manages at least the stated below

- Start of the simulation
- End of the simulation
- Arrival of a packet
- Departure of a packet

To do this, create first an ordered queue/list of events where:

- Every event always links to the one that immediately follows it in time;
- When you insert an event in the queue, you always insert it in order of increasing time; (i.e., say that the queue contains three events: event 1 taking place at time t_1 and linking to event 2, which takes place at time t_2 and which links to event 3 at time t_3; if another event 4 taking place at time t_4 is inserted in the queue, and $t_2 < t_4 < t_3$, then you have to make event 2 link to event 4, and event 4 link to event 3.

Finally implement the system behavior as seen in class, namely:

- When a packet arrives: if the server is free seize the server and schedule the departure of the packet; if the server is busy, increase the number of packets in queue;
- When a departure event is triggered: if the queue is empty, release the server; otherwise keep the server busy and schedule the next departure event.

Use your simulator to do the following:

1. Show how the number of packets in the system (those in queue plus those currently in service) varies over time. Compare your results with the theoretical average number of packets in the system in stationary conditions, $\rho/(1-\rho)$, where $\rho = \lambda/\mu$.
2. Play with λ and μ, and discuss how their values impact the convergence of the system to the theoretical value.
3. Use your simulator to measure the average time that a packet has to wait in queue, on average, and compare it against the theoretical value, $\rho^2/\big(\lambda(1-\rho)\big)$ (*Hint*: you will need to run your simulator several times to do the above. Remember the contents of the class on output analysis.)

Exercise 2

1. Repeat exercise 1 for an M/M/c/K system where
   - there are c servers (start with $c = 2$, then test for larger values if time allows);
   - the queue of each server can hold up to K packets (start with some easy number such as $K = 10$): this means that a server having K packets in queue would discard any other packets assigned to it.
2. Experiment with simple packet assignment policies (round-robin, least-loaded servers first...) as well as with policies that consider the occupancy of the queue of the servers (e.g., avoid sending packets to servers that have a full queue, or to a queue that is more than x% full, for some x).
3. For each policy, plot a histogram showing the number of packets served by each of the c servers, as well as the average distribution of the queuing delay experienced by each packet. You can show other metrics as needed.