Homework 4

Simulation and Performance Evaluation – University of Trento

DEADLINE: 17:29 on June 10, 2025

You can solve the following assignments using any programming language. Try to do the homework by yourselves, without help from AI tools. Unless otherwise stated, can use utility functions made available by the programming language of your choice, including functions to extract random numbers.

You will see a facility on Moodle to upload your homework. Please upload your code, and separately upload a short report where you describe your findings (no more than 3 pages). Upload two versions of the report: one with your names, and a second, fully anonymized one.

Exercise 1

Consider an M/M/1 queue-server system. Remember that this means that arrivals follow a Poisson process of rate λ, and services (or departures) follow a Poisson process of rate μ, where you should have $\mu > \lambda$, e.g., $\lambda = 1$ and $\mu = 2$. Moreover, there is a single server and the queue is managed according to a FIFO policy.

Implement a discrete-event simulator to evaluate the performance of the M/M/1 system. The simulator should manage at least the below events:

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

To do this, create first an ordered queue 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 affect the convergence of the system to the theoretical value.

(*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

Use the simulator built in the previous Exercise 1 to estimate the average amount of time it takes for a packet to traverse the system. Namely, compute the average time elapsed between the arrival and the departure of the packet. Do this using both the naive estimator (that computes the sample mean of packet traversal times) and by using one of the following variance reduction techniques:

- use the number of packets in queue as a control variate;
- post-stratify, where a stratum is defined in terms of the number of packets in queue when a packet enters the system.

and discuss if the variance reduction strategy helped reduce the variance of your estimate of the traversal time.

It may be useful to remember that the probability distribution of the number of packets ahead of an arriving packet (including those in queue and those currently in service) is $\pi_k = \rho^k(1-\rho)$, and that the average traversal time for any packet is $1/(\mu - \lambda)$.