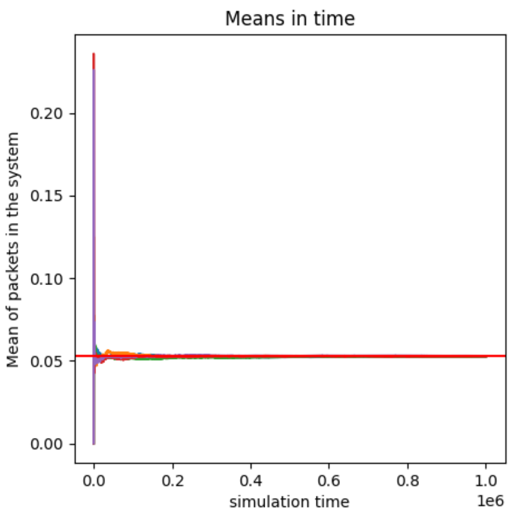
# Homework 3
# Simulation and Performance Evaluation
# University of Trento

Davide Pasetto (239941)
Deborah Aloisi (249948)

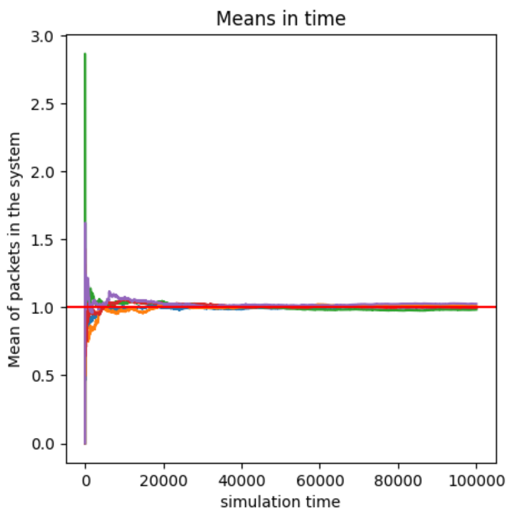
May 2024

## Exercise 1

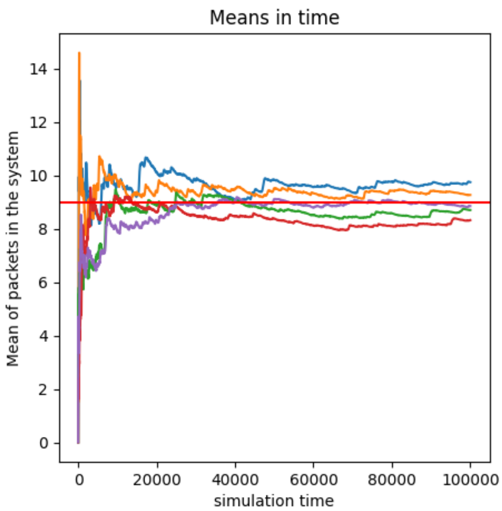
To analyze the performance of a M/M/1 machine we ran 5 simulations where both the arrival and the service time followed a Poisson process, varying the value of mu and lambda. We used the results to visually identify the warm-up time in order to remove the initialisation bias form the batch means calculation. The results of this process can be seen in figure 1



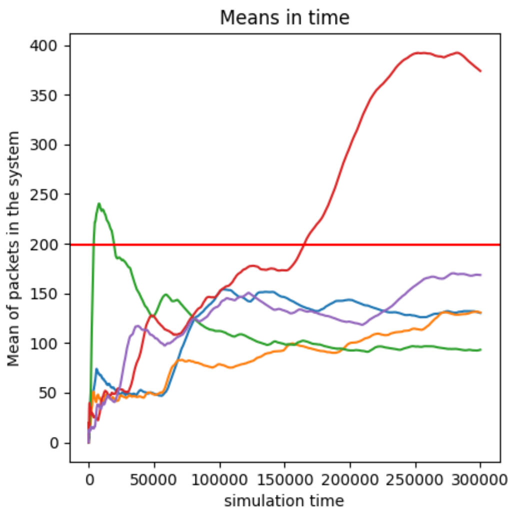

(a) Results using mu=0.1, lambda=2




(b) Results using mu=1, lambda=2




(c) Results using mu=1.8, lambda=2

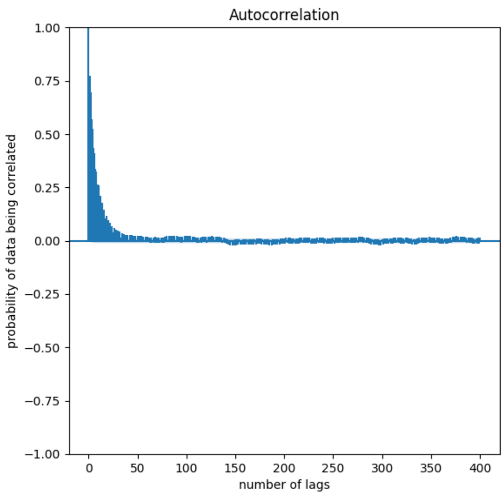



(d) Results using mu=1.99, lambda=2

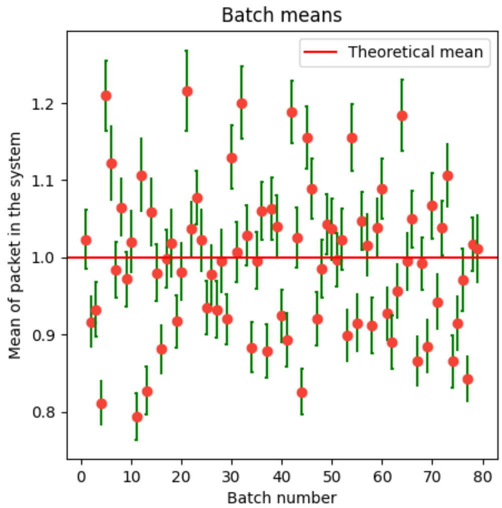
Figure 1: The average of different runs of the simulation using different values for mu.

As it is possible to see, even this crude plot already shows how the mean tends to converge pretty closely to the theoretical mean when the fraction $\rho = \lambda/\mu$ is much smaller than 1. On the other hand, when the fraction tends to 1 (like in figure 1d) the simulation converges much slower, and it was not feasible for us to run a simulation long enough for convergence to be reached., therefore making the simulation mean imprecise.

After a proper warm-up time is identified for every value of lambda and mu, we take a single simulation from every run to apply the batch mean.
For this simulation we plot the autocorrelation (figure 2a) to identify a proper cut-off lag that is also used to determine the batch size.




(a) Batch auto-correlation




(b) batch means

Figure 2: The autocorrelation and the batch means of a single simulation: simulation with l=1 and mu=2 taken as an example

After dividing the samples into batches of the proper size and calculating their mean and confidence intervals (figure 2b), it is possible to calculate the grand mean and its confidence interval. The results are as following:

| **lambda** | **mu** | **theoretical mean** | **simulation mean** |
|---|---|---|---|
| 0.1 | 2 | 0,0526 | 0.0524 +- 0.0005 |
| 1 | 2 | 1 | 0.9965 +- 0.0213 |
| 1.8 | 2 | 9 | 8.7487 +- 0.8513 |
| 1.99 | 2 | 199 | 145.0034 +- 28.1547 |

Table 1

## Exercise 2

In this exercise we used a Pareto distribution to generate the service time instead of a Poisson process. The peculiarity of the Pareto distribution is that, although most of the values are still close to the mean, it is still possible to draw extremely big variates. These values create a spike in the number of packets in the system. To counteract this effect, it is necessary to lower the arrival rate of packets. In figure 3 it is possible to see how the mean of 5 different simulations vary over time, while using a value of 0.2 for the arrival rate.

Computing the mean of the packet in the system result more difficult than in the previous exercise that while taking some time all simulations converged to one line. In this case we cannot find a point to consider as the end of the warm-up time. We are obligated to take values that variate a lot during the simulation.

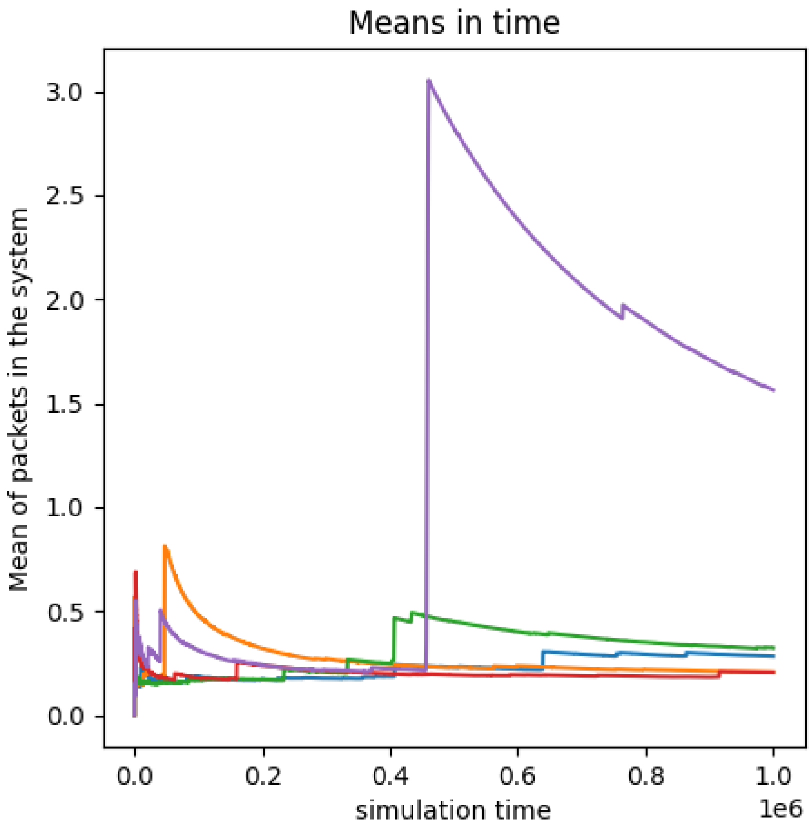
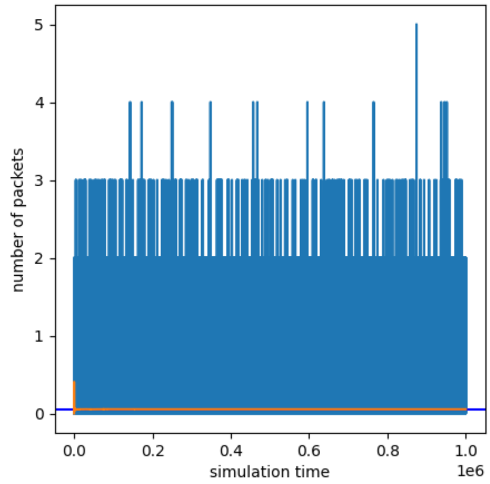
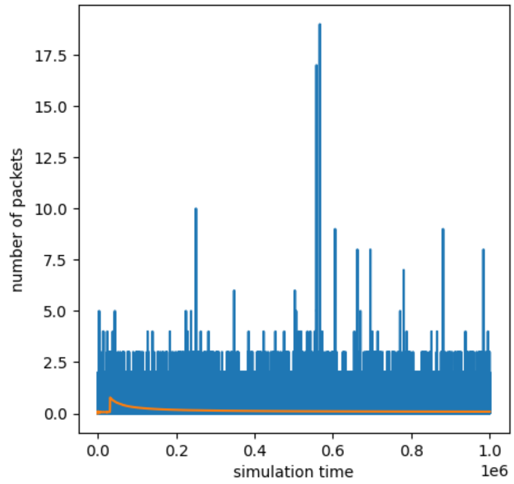



Figure 3: Results using a pareto distribution and lambda=0.2

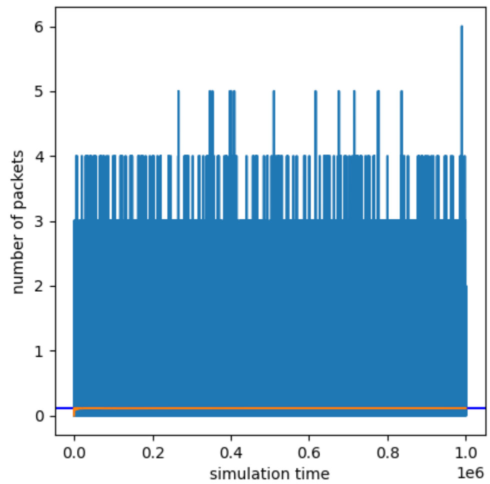
If figure 4 we can see a comparison between some runs of the system while using Poisson processes for both the arrival time and the service time and using a Pareto distribution for the service time. As it is possible to see, in figure 4a and ref 4c the number of packets in the system tends to stay pretty stable at low numbers. On the contrary, in figures 4b and 4d the number of packets have huge spikes, and the number is overall higher. This is also due to the fact that the average service time for the Pareto distribution is 1,5 seconds, which is significantly higher than the average service time of an the exponential distribution used for the service time, whose mean is 0,5. To assess this difference we ran a simulation using as service time an exponential distribution whose mean is the same as Pareto (i.e. with mu=0.66). It is still possible to see how in figure 4e the number of packets is much more stable towards lower values than the simulation using Pareto.



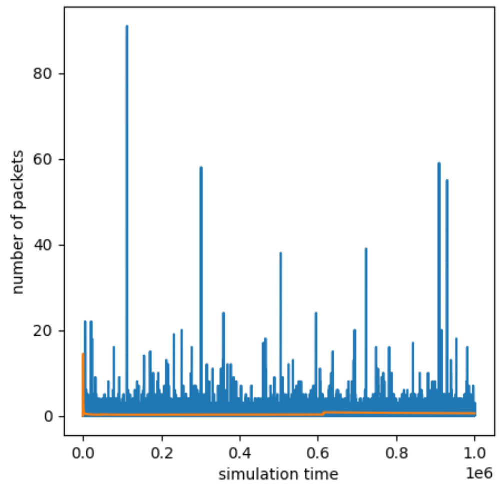

(a) lambda = 0.1, mu = 2



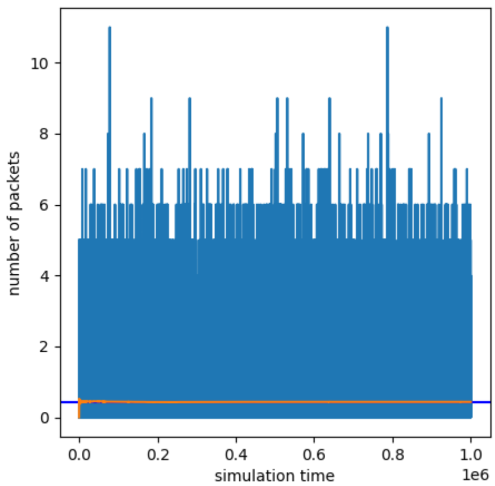

(b) Pareto with lambda = 0.1



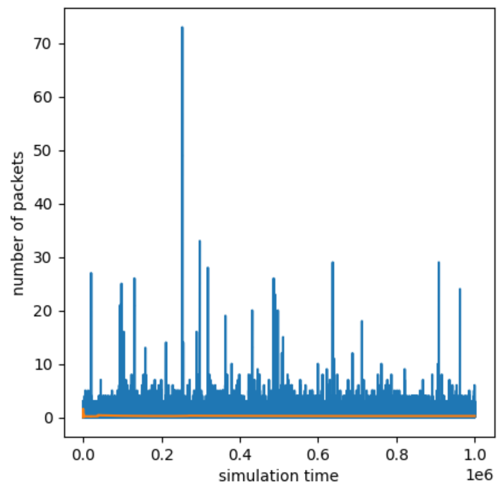

(c) lambda = 0.2, mu = 2




(d) Pareto with lambda = 0.2




(e) lambda = 0.2, mu = 0.66




(f) Pareto with lambda = 0.2

Figure 4: Comparison of system occupation with the same lambda and to the left with a Poisson process for service time and on the right with a Pareto distribution