Priority Queue Lab 11

Stefano Gioda, *s294781*

I. Problem Statement

Write the simulator for a queue with k servers, and waiting line of size N, which implements a strict priority service discipline (with preemption). Customers are partitioned into two classes: High Priority (HP) customers and Low Priority (LP) Both arrival processes are Poisson with rate λ_{HP} and λ_{LP} respectively. LP customers are served only when no HP customers are in the waiting line. The service of a LP customer is potentially interrupted upon the arrival of a HP customer if no servers are idle. Furthermore, upon the arrival of a HP customer, to accommodate the arriving customers, a LP customer is potentially dropped if there is not room in the waiting line. Plot of the average delay for HP customers, LP customers, end the aggregate average delay (i.e. the average delay on all customers) when $\lambda_{HP} = \lambda_{LP} = 0.2, 0.4, 0.8, 1.4, 2.0, 2.4, 2.8$, k=2, N=1000 and

a) $E[S]_{HP} = E[S]_{LP} = 1.0$

b) $E[S]_{HP} = 1/2$ and $E[S]_{LP} = 3/2$

Consider the three ”usual” distributions for the service time:

EXP: exponentially distributed with mean= $E[S]_{*P}$

DET: deterministic = $E[S]_{*P}$

HYP: distributed according to a hyper-exponential distribution with mean=$E[S]_{*P}$ standard deviation=$10 * E[S]_{*P}$

II. Proposed Approach

A. Preemption

When an high priority customer arrives and there is no server available, but some low priority customers are being served, then the last low priority users being served is preempted, so its service time is decreased by the amount of time already spent in the server and it is sent back in the queue, and the high priority customer takes its place in the server.

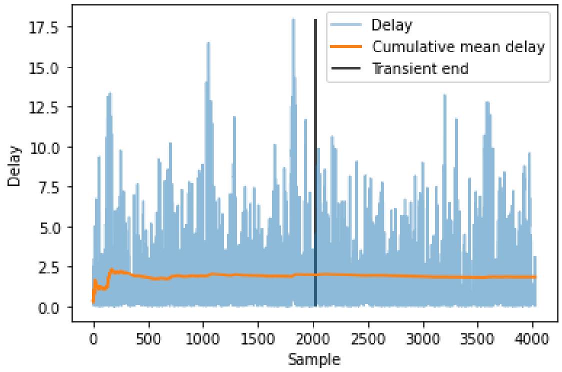
B. Detect and remove transient

The analysis of the transient is performed on the delay which is the metric of interest for this simulation. In particular on the delay of all the customers, both low and high priority.

In order to detect the transient end in an automated way, the following step are used whenever there is a new delay:

- compute the new cumulative average
- if the absolute difference between the previous and the current cumulative average is less than a difference threshold:
  - increase the counter of steady state samples
  - if the counter is bigger than a length threshold, then the transient is finished
- else: reset the counter of steady state samples to 0

Fig. 1: Delay of all customers using exponential service time with $E[S]_{HP} = 1/2$ and $E[S]_{LP} = 3/2$ and $\lambda_{HP} = \lambda_{LP} = 0.8$. Detection of the transient using the cumulative average delay.



The algorithm requires two thresholds: one for the absolute difference of consecutive averages and one for the number of consecutive samples with stable average. The requirement of having stability for a certain number of samples is due to the fact that sometimes it looks stable for a bit, but then it starts again oscillating. Some experiments have been performed using the relative difference instead of the absolute one, but it led to too short transient for high utilisation, since the average delay in this case is high, bigger differences are allowed, while increasing too much the transient for low utilisation, since the average delay is quite small.

A result of this algorithm is illustrated in Fig. 1, where the threshold was set to $1e^{-3}$ and the length of the sequence to 50.

C. Hyper-exponential generation

The problem ask to repeat the simulation using exponential, deterministic and hyper-exponential distributions for the service time. While the first one is already provided in Python and the second one is trivial to implement, for the last one an ad-hoc solution was needed.

The hyper-exponential distribution is composed by n exponentially distributed random variables, each of them has probability p_i of being chosen. It is simulated by choosing one of the random variables according to the probabilities p_i and extract a value from it.

The mean and the variance of an hyper-exponential, given the means μ_i of the exponential variables, are

$$E[X] = \sum_{i-1}^{n} p_i \mu_i$$

$$Var[X] = \left(\sum_{i=1}^{n} p_i \mu_i \right)^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} p_i p_j \left(\mu_i - \mu_j \right)^2$$

By setting n, the number of exponential random variables, to 2, this system must be solved

$$\begin{cases} p_1 \cdot \mu_1 + p_2 \cdot \mu_2 = \mu_H \\ p_1 + p_2 = 1 \\ 2 \cdot p_1 \cdot p_2 \cdot (\mu_1 - \mu_2)^2 = \sigma_H^2 - \mu_H^2 \end{cases}$$

In order to have $\mu_H = 1$ and $\sigma_H = 10$, one of the possible solutions is

- $p_1 = 0.99$
- $p_2 = 0.01$
- $\mu_1 = \frac{\sqrt{2}-1}{\sqrt{2}}$
- $\mu_2 = \frac{99+\sqrt{2}}{\sqrt{2}}$

A solution for the $\mu_H = \frac{1}{2}$ and $\sigma_H = 10 \cdot \frac{1}{2} = 5$ is

- $p_1 = 0.99$
- $p_2 = 0.01$
- $\mu_1 = \frac{2-\sqrt{2}}{4}$
- $\mu_2 = \frac{2+99\sqrt{2}}{4}$

A solution for the $\mu_H = \frac{3}{2}$ and $\sigma_H = 10 \cdot \frac{3}{2} = 15$ is

- $p_1 = 0.99$
- $p_2 = 0.01$
- $\mu_1 = \frac{6-3\sqrt{2}}{4}$
- $\mu_2 = \frac{6+297\sqrt{2}}{4}$

III. Results

All the reported results used

- threshold for the comparison of the cumulative averages in the transient analysis equal to $1e^{-3}$
- number of samples with stable cumulative average in the transient analysis equal to 50
- number of delays collected equal to 1000

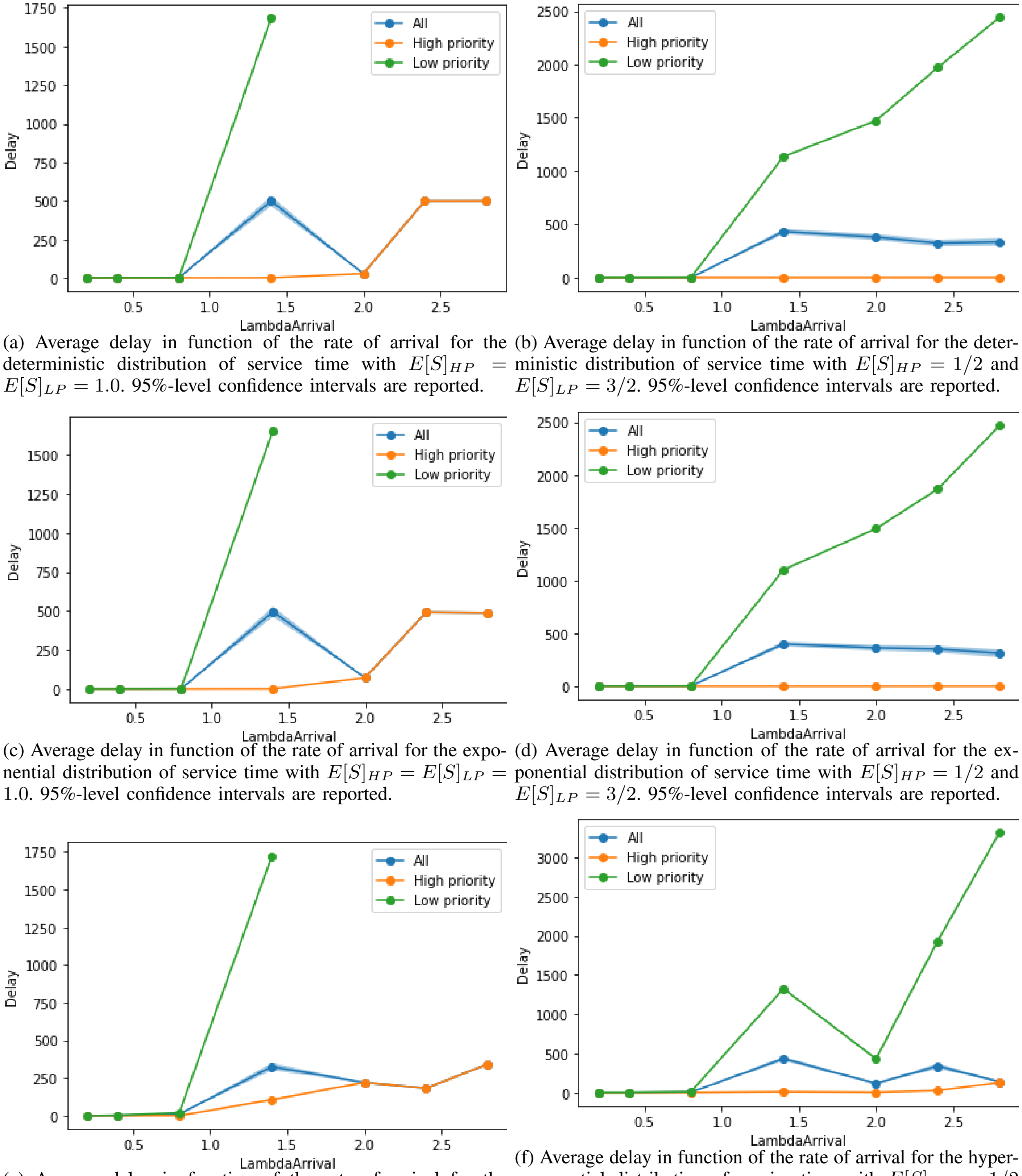
A. *Average delay in function of the rate of arrival*

Fig. 2 shows the average delays for all the distributions of service time in function of the rate of arrival $\lambda = \lambda_{HP} = \lambda_{LP}$.

In all the figures the average delay increases as the rate of arrival increases, in particular for low priority customers, since the fact that more high priority customers are arriving, implies that low priority customers need to wait longer before being served. Instead, for the high priority customers the increase of average delay is smaller.

An interesting fact to point out is that for the graphs on the left column (Fig. 2a, Fig. 2c, Fig. 2e), which are characterized by having $E[S]_{HP} = E[S]_{LP} = 1.0$, the average delay for low priority customers stops at rate 1.5 and this is because with higher rates they are not served at all. On the other hand, for the graphs on the left column (Fig. 2b, Fig. 2d, Fig. 2f), which are characterized by having $E[S]_{HP} = 1/2$ and $E[S]_{LP} = 3/2$, the average delay for low priority customers is present for all rates, since in this case the service time of high priority customers is lower $(1/2)$, so there is time also for serving low priority customers.

Fig. 2: Plots of the average delay for high priority customers, low priority customers and aggregate average delay of all customers for the different distributions of service time.



(a) Average delay in function of the rate of arrival for the deterministic distribution of service time with $E[S]_{HP} = E[S]_{LP} = 1.0$. 95%-level confidence intervals are reported.

(b) Average delay in function of the rate of arrival for the deterministic distribution of service time with $E[S]_{HP} = 1/2$ and $E[S]_{LP} = 3/2$. 95%-level confidence intervals are reported.

(c) Average delay in function of the rate of arrival for the exponential distribution of service time with $E[S]_{HP} = E[S]_{LP} = 1.0$. 95%-level confidence intervals are reported.

(d) Average delay in function of the rate of arrival for the exponential distribution of service time with $E[S]_{HP} = 1/2$ and $E[S]_{LP} = 3/2$. 95%-level confidence intervals are reported.

(e) Average delay in function of the rate of arrival for the hyper-exponential distribution of service time with $E[S]_{HP} = E[S]_{LP} = 1.0$. 95%-level confidence intervals are reported.

(f) Average delay in function of the rate of arrival for the hyper-exponential distribution of service time with $E[S]_{HP} = 1/2$ and $E[S]_{LP} = 3/2$. 95%-level confidence intervals are reported.