



SF2567 - QUEUING THEORY PROJECT

Rémi Lacombe

Advisor: Pr. Per Enqvist - KTH Royal Institute of Technology

1 - Introduction

Queues are a huge issue when it comes to productivity. It is estimated that every year 37×10^9 hours are wasted waiting in queues for the US alone. In this project, customers can either be of type 1 or of type 2 depending on the type of service they seek. The queuing system (QS) under study has s_1 servers that can handle type 1 customers, s_2 servers for type 2 customers and s_{12} servers that can handle both types. This situation arises in many places, such as in call centers, which motivates our interest in such QS. No analytic result for the steady-state behaviour of this sort of QS is known when they include multi-types servers. Otherwise, if there are only single-type servers, the QS can satisfactorily be described with one or several “M/M/s” queuing models which analytic results are fully known and understood. The purpose of this work is to implement a simulation of a QS which includes multi-types servers in order to compute its parameters of interest when it reaches steady-state such as the mean waiting time of the customers in the queues. The two main parts of this project were:

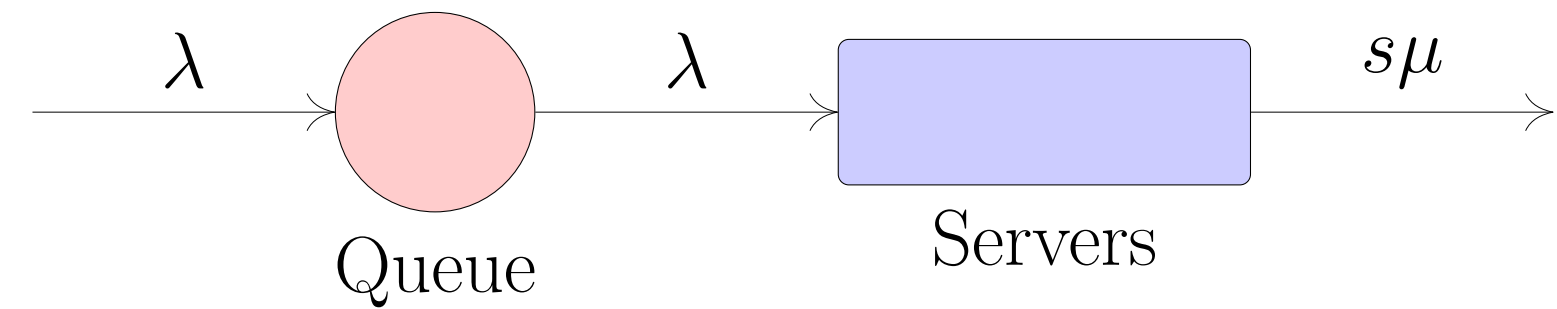
- Implementation of a simple M/M/s QS. This part aimed at verifying that the QS simulation was working properly since the results it yielded could be compared with analytical ones.
- Implementation of a QS with multi-types servers. This simulation was similar to the previous one, but had to include the new types of servers.

Both simulations have been conducted with C++ since this language combines an oriented-object aspect with excellent speed performances.

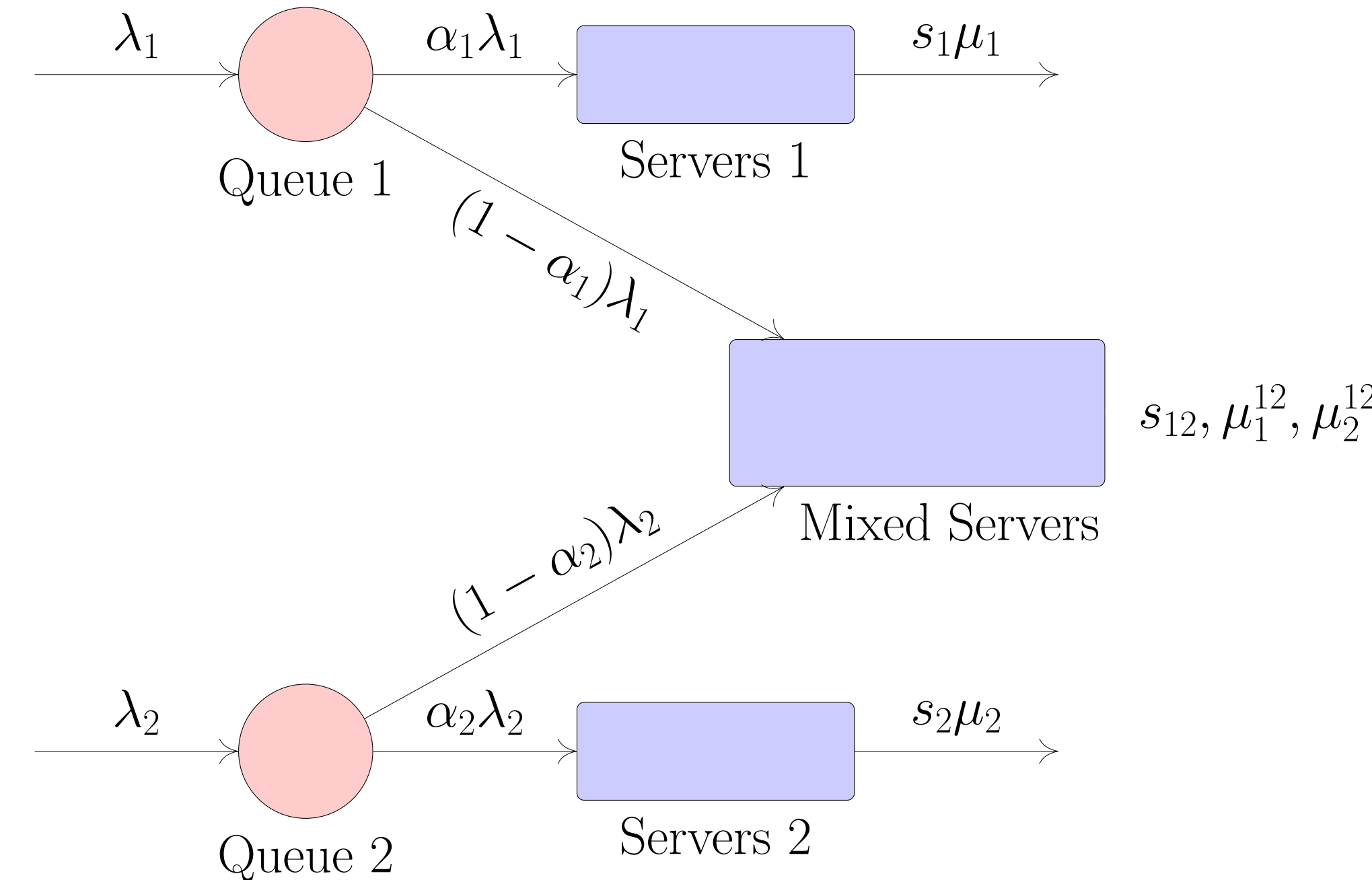
2 - Definitions

- The arrival of customers in the QS is modeled by a Poisson process, so the inter-arrival time between two customers has an exponential distribution of parameter λ_1 or λ_2 depending on their type. It can be shown that these parameters represent the mean arrival rate of the customers in the QS.
- Once a customer of type 1 (respectively of type 2) enters the QS, there is a probability α_1 (respectively α_2) that he will be assigned to a single-type server. If no adequate server is available, the customer waits in the queue of its service type until another leaves the QS. The queues are unbounded FIFOs.
- Likewise, the service of one customer handled by one server is modeled by a Poisson process of parameter μ_1 , μ_2 , μ_1^{12} or μ_2^{12} depending on the type of customer and on the type of server. Therefore, the mean leaving rates for the groups of single-type servers are $s_1 \mu_1$ and $s_2 \mu_2$ and the mean leaving rate for the multi-types servers is $\left(\frac{(1 - \alpha_1)\lambda_1}{(1 - \alpha_1)\lambda_1 + (1 - \alpha_2)\lambda_2} \mu_1^{12} + \frac{(1 - \alpha_2)\lambda_2}{(1 - \alpha_1)\lambda_1 + (1 - \alpha_2)\lambda_2} \mu_2^{12} \right) s_{12}$.

Simple M/M/s queue



Queueing system with multi-types servers



3 - C++ Modelling

Simple M/M/s Queue: In the case where there is only single-type servers, the QS is a Markov process. A class *Queue* was created, along with several member functions, in order to keep track of the number of customers in the QS and of which servers were busy. The main file includes a loop that performs a large number of iterations. Each iteration is a jump forward in time to the next event that happens in the QS: either a customer joins or one leaves a server. The time before this particular event reoccurs is then computed using the *exponential_distribution* method of C++ and stored in an array. This way, the program simulates a real-time QS which customers are constantly joining or leaving. Once a given number of iterations has been reached, the process stops and the steady-state probabilities are computed since we know the amount of time the system spent in each state.

QS with multi-types servers: A child class *DoubleQueue* is now created to keep track of the customers waiting for or being served in the multi-types servers. It has to store the order of arrival of the customers waiting in the queue since now the two types of customers can be waiting for the same servers. Apart from that the simulation is quite similar to the previous one in its working albeit it requires much more code. The steady-state parameters are also computed once the simulation comes to a stop.

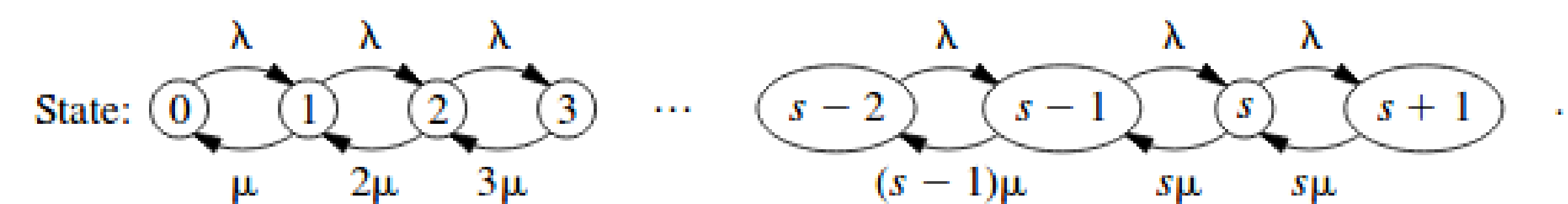
4 - Results

Since a the M/M/s queue is a Markov process, its steady state probabilities are given by the following formula:

$$P_n = \frac{(\lambda/\mu)^n}{n!} P_0 \quad \text{if } 0 \leq n \leq s, \quad \text{and} \quad P_n = \frac{(\lambda/\mu)^n}{s! s^{n-s}} P_0 \quad \text{if } n \geq s, \quad [1]$$

where P_n is the steady-state probability of having n customers in the QS and s is the number of servers. For a running time of about 70 seconds, my simulation of a M/M/s queue yields these values for the steady-state probabilities with a precision of 2×10^{-4} . My simulation of a QS which includes multi-types servers yields these same results as well (when one chooses $\alpha_1 = 1$ for instance). Therefore, both simulations seem to be satisfactory approximations of a real-life QS.

Markov process for a M/M/s queue [1]



Acknowledgement

I would like to warmly thank Pr. Per Enqvist at the Department of Mathematics at KTH for providing me with this research subject and for helping me to understand it. I am looking forward to continuing to work on the theoretical study of QS with multi-types servers with him during the second semester now that we have this computational tool.

[1] Frederick Hillier, Gerald Lieberman *Introduction to Operations Research*, 1967