Management and Content Delivery for Smart Networks: Algorithms and Modeling

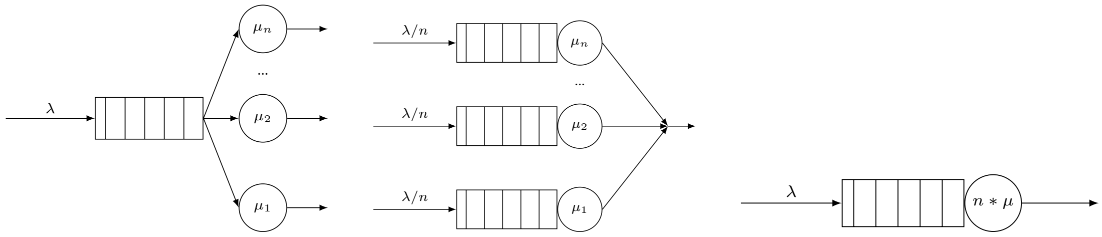
Lab. 1: Simulation of Simple Queuing Models

The objective of this laboratory is to practice essential steps for the simulation of queuing systems. You will simulate very basic queue models, investigating the effects of model parameters, e.g., arrival and service rates.

This will help you to understand the workings of a event-based simulator and prepare the ground for the more complex simulations that will follow.

Ex 1: Comparing Queue Models

Assume a set of servers handles requests for a popular service. The operator has different options to deploy the service, as modeled in Figure 1.



(a) 1 waiting line, N processors (b) N independent servers (c) 1 waiting line, 1 fast processor

Figure 1: Possible configurations for the servers.

In the first scenario (Figure 1a), the operator deploys a single solution with n processing units, all with the same service rate (Poisson process, $\mu_1, \mu_2, \ldots, \mu_n = \mu$). The memory to buffer pending requests while processors are busy is shared by all units.

In the second scenario (Figure 1b), the operator deploys n servers that operate in parallel. Each server has independent memory to buffer pending requests, with exactly the same capacity. The servers are also equal in terms of service rate (Poisson process, $\mu_1, \mu_2, \ldots, \mu_n = \mu$).

In the third scenario (Figure 1c), the operator deploys a single server that is on average n times faster than the ones in previous cases (i.e., Poisson process, $n * \mu$), but with similar buffer capacity.

In all cases, servers satisfy requests sequentially, one at a time. If a request arrives while no processing unit is free, it is put in a queue – first-come, first-served (FCFS). Requests arrive following a Poisson process with arrival rate λ. In Figure 1b arrivals are split in the different waiting lines **randomly**.

Lets assume servers have **limited buffer capacity able to hold B pending requests**. If a request arrives when there is no more buffer space, it is discarded.

Study:

- How the *response time* per request varies according to μ and B.
- How the number of requests rejected due to lack of buffer space is affected by the parameters of the models.
- Which approach is "better" in terms of response time and rejected requests?
- Compare your simulation results with theoretical performance metrics.

Ex 2: Mixing Servers

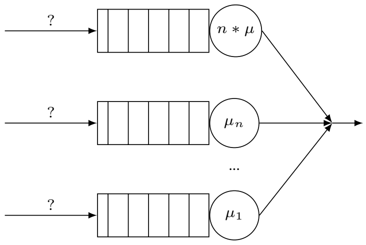


Figure 2: Mixing diverse servers in the same system.

Now imagine that a mix of previous solutions is deployed, as in Figure 2 – that is, n servers of low service rate (with $\mu_1, \mu_2, \ldots, \mu_n = \mu$) are put in parallel with a server with $n * \mu$ service rate. The overall arrival rate is still λ and all servers have the same buffer capacity B. Assume again FCFS queues, Poisson arrival and service, and assume that jobs cannot change queue during the waiting time.

- How would you distribute requests among the queues in order to maximize throughput? Support your answer with simulation results.

Groups and Final Reporting

You are expected to work on groups of up to three students. Each group is required to prepare a **single** report describing results of **all labs in the course**. This report must not exceed 10 pages.

You have as starting point the skeleton of a network simulator written in Python.

You need to delivery both the written report and your source code before the last exam date, in September (13/09/2018).

Groups delivering the report **before 16 Jun 2018** will receive feedback and have the opportunity to deliver a revised report.

References

[1] SimPy in 10 Minutes. `https://simpy.readthedocs.io/en/latest/simpy_intro/`
[2] matplotlib. `http://matplotlib.org/`