

Simulation - Assignment 1

Anton Roth

April 21, 2017

1 Task 1

The system with two queues was implemented with the *Event-Scheduling* approach. The states for the system were; Length of Q2, number of arrivals to Q1 and number of rejected arrivals to Q1. Four events were implemented: Arrival to Q1, Departure of Q1 (combined with arrival to Q2), Departure of Q2 and Measure.

As model verifications the following was controlled numerically and graphically:

- Inter-arrival times $\rightarrow 0$:
 - Length of Q1 $\rightarrow 10$.
 - Rejection ratio of Q1 $\rightarrow 1$.
- Inter-arrival times $\rightarrow \infty$:
 - Length of Q1 $\rightarrow 0$.
 - Rejection ratio of Q1 $\rightarrow 0$.

Measurements were made with time-differences exponentially distributed with a mean of 5 s. 10000 measurements were taken. The results of task 1 is presented in Table 1.

Table 1: Results of task 1. For the three different inter-arrival times the mean value and the corresponding standard deviation (StdDev) is presented for length of Q2 and the rejection ratio of Q1.

Inter-arrival times Q1 (s)	Mean length Q2	StdDev length Q2	Mean rejection ratio Q1	StdDev rejection ratio Q1
1	11	13	0.52	0.02
2	4.4	3.0	0.070	0.009
5	0.43	0.55	0	0

From the results of length Q2 presented in Table 1 it is impossible to draw proper conclusions as the uncertainties is of the same order as the mean. The rejection ratios are more significant and shows an expected behaviour, i.e. the shorter inter-arrival time the higher rejection rate.

Is it ok to assume that mean is normal distributed? This is *Central Limit Theorem*, right?

2 Task 2

The *Event-Scheduling* approach was also used in this task and the code written for task 1 was used as a template. The event structure was changed and a specific method was implemented for the addition of a job to the buffer. The following events were used: AddJobA, AddJobB, ServeJobA, ServeJobB and Measure.

The states were simply the number of job type A, denoted NA and B, denoted NB , in the buffer. Due to bad planning, an ugly solution was implemented for the case of adding a job from the buffer to serve (see code).

As a verification step the following was controlled:

- Delay times $\rightarrow \infty$:
 - $NA + NB \rightarrow 0$. This is due to that the serve time of A, $x_A = 0.002$ s is shorter than the average arrival time ~ 0.0067 .
- Serve time for job A and B $x_A, x_B \rightarrow \infty$:
 - $NA + NB \rightarrow \infty$.

Table 2: Results of task 2 for the first three questions/simulation runs presented in the task description. For all runs the mean and standard deviation (StdDev) of the buffer length is presented.

“Run”	Mean length of buffer	StdDev length of buffer
1	130	100
2	7.2	7.9
3	3.6	3.8

The result of run 1 and 2 differs substantially. As job B is prioritised and feeded to a system with a delay of 1 s one obtains a periodicity. Consider a constant delay of 1 s and the situation when the system starts up, the periodic behaviour can be explained with the following chain of events:

1. Job A:s are added to the buffer and served efficiently. The buffer length is kept minimal.
2. After 1 s job B:s are added to the buffer and since they are prioritised they are served immediately. The serving time is $x_B > x_A$ and together with the fact that A jobs are continuously added to the buffer this implies that the buffer length will increase.
3. After some time all B jobs have been served and A jobs are again served. The buffer length decreases.
4. \rightarrow point 2.

Run 2 exhibits a weaker periodic behaviour since delay is now sampled from an exponential distribution. This has the effect that all B jobs will not arrive at the server at the same time and the buffer length will not increase as much.

The mean length of the buffer for the third run is the lowest. The buffer length no longer shows a clear periodic behaviour. As A jobs now are prioritised and have a shorter serving time the buffer length can be efficiently be kept minimal. When there are no A jobs, B jobs which have a longer serving time will be processed. The pile-up in the buffer is not possible due to the efficient processing of the jobs. Better description?

It should also be mentioned that there is a “start-up” period needed for the first run (although not implemented), as it takes some time for the buffer to stabilise in its periodic behaviour.

3 Task 3

A similar system compared to in task 1 was implemented. One difference was the measure of the mean time a customer spends in the system. The time spent by every customer in the system was stored in a container and used to calculate the mean. Hence, it was not “Measured” as an event as this was considered the most efficient implementation and is not very computationally costly for the simulation.

Measurements were made with time-differences exponentially distributed with a mean of 5 s. 10000 measurements were taken. The results of task 1 is presented in Table 1.

Table 3: Results of task 1. For the three different inter-arrival times the mean value and the corresponding standard deviation (StdDev) is presented for length of Q2 and the rejection ratio of Q1.

Inter-arrival times Q1 (s)	Mean length Q2	StdDev length Q2	Mean rejection ratio Q1	StdDev rejection ratio Q1
1	11	13	0.52	0.02
2	4.4	3.0	0.070	0.009
5	0.43	0.55	0	0