A Close Analysis of the M/M/n, M/D/n and M/LT/n Queueing Systems

Vrije Universiteit, University of Amsterdam, Netherlands

November 2019

Abstract

In this paper, we examine the queueing behaviour of M/M/n, M/D/n and M/LT/n systems, where the latter is a long-tail distribution, which consists for 75% of servicing times with an average of 1 and the remaining 25% of servicing times with an average of 5. To this extent, the batch means method was utilised to examine the average time spent in a queue. Comparing the results for different numbers of servers, rendered the conclusion that employing more servers leads to shorter waiting times for the M/M/n, while the workload (ρ) was kept constant. In addition to this, it was found that applying a Shortest Job First discipline reduced the queueing times for M/M/n in comparison to the First In First Out discipline. The M/D/n model yielded short average queueing times which were similar to the M/M/n SJN discipline. Finally, for the M/LT/n system it was clear that the limiting factor on the queueing time is the long average servicing time, which instigates long waiting times for relatively low workloads.

1. Introduction

In 1909, Agner Krarup Erlang firstly formulated the queueing theory for a Danish telecommunication company (Erlang, 1909). The aim of this study was to understand the daily use of the network during peak- or off-peak hours. Understanding the process of demand, a company can provide the necessary infrastructure to ensure an adequate service. The queueing theory can be applied to all sorts of domains incorporating a waiting time due to queuing processes, such as server systems or retail businesses.

In this paper, the queuing theory is applied to server systems. The average waiting and occupancy times of different systems are analysed to enlighten the capacity of the servers (Willig, 1999). The paper will discuss M/M/n, M/D/n and M/LT/n[1] distributions exploring the effect of using multiple servers. Additionally, the dynamics of the *first in, first out* will be compared to the results yielded by the *shortest job first* discipline. We hypothesise that using multiple servers will decrease the waiting times for similar workloads, due to avoided congestion, which would have otherwise resulted from substantial tasks. Similar reasons argument in favour of the *shortest job first* over *first in, first out* with respect to minimising average queueing times.

[1]LT: long-tail distribution

2. Background & Theory

2.1. Queueing Theory

Queueing theory is a statistical approach of defining the waiting time, queue length and service time in a given environment. The theory distinguishes between an infinite population sampling of clients and the queueing node, also referred to as service centre. Based on the continuous-time Markov process, one client arrives at the service centre in a random fashion and requests a job. While the first client is served, a second client comes in and waits for his turn. Hereby, the number of jobs in the queue increases by 1. Subsequently, the first client is served and leaves the service centre and the number of jobs decreases by 1. This model is based on the service discipline *first in, first out (FIFO)*, which due to a buffer induces a waiting time. Another service discipline could be *shortest job next* (SJF), where the client in the queue (if present) with the smallest task is served first.

The average length of a queue depends on the capacity of the service centre, i.e. the service speed and the number of queue nodes (servers), the rate of the arrivals and the service discipline. In the case of n servers in a service centre, the model can be described by the Kendall's notation (Kendall, 1953):

$$A/B/n - S$$

where A defines the distribution of the interarrival times, B the distribution of the service times, n

the number of servers available in a service centre and S the implemented service discipline. The general notation varies according to the nature of the system studied. In case of an exponential distribution of the interarrival times $A(t) = 1 - e^{-\lambda t}$ and an exponential distribution of the service time $S(t) = 1 - e^{-\mu t}$, the letter M replaces both A and B:

$$M/M/n$$

It must be noted that M stands for the Markov property of being memoryless. In other words, the probability distribution of a present interarrival time would not be affected by the previous interarrival times.

If the interarrival times are memoryless and distributed in the same manner, they are independent and identically distributed (iid) random variables. Additionally, if the interarrival times are exponentially distributed, the process of a client coming in can be described by the Poisson process.

A different variation of $M/M/n$ system is the long-tail distribution where the average servicing times μ can take on different, predefined, discrete values. The servicing times are still exponentially distributed for the different μ's.

The environment can also be described by the following notation

$$M/D/n$$

where M defines the exponential distribution of the interarrival times. In this case, D refers to a deterministic service time. The time needed to serve a client is constant. Consequently, all the jobs have the same load.

In case of a Markov Distribution, the average waiting time in the queue W_q and the system total waiting time W (i.e. W_q plus the servicing time) can be found by a continuous-time Markov chain (CTMC). The assumption must be set that no events can happen at the same time and that the arrival rate λ and service rate μ are not state-dependent. Figure 1 shows the CTMC flow where an event is generated increasing the number of clients in the queue with a rate λ or decreasing the queue length with a rate μ.

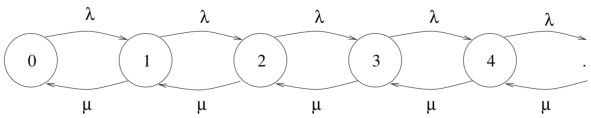

Figure 1: CMTC model where λ defines the arriaval rate and μ the capacity of the server. Each number represent a state with the correspond clients in the queue (Willig, 1999).

Whether arrival or servicing happens first is determined by a Poisson process. An environment is said to have a steady-state if after a certain time the distribution of the costumers does not change. To satisfy the latter statement, the probability p of a transfer from state k to state $k+1$ at a certain time t must be the same as transitioning from $k+1$ back to k to have a steady-state. Therefore, the following expression can be stated

$$\lambda P_k(t) = \mu P_{k+1}(t)$$

with t being independent of the state. When rewritten, it renders

$$P_{k+1} = \frac{\lambda}{\mu} P_k$$
$$P_{k+1} = \rho P_k$$

with ρ being the workload. P_k can be reduced to the number of clients in the queue. P_{k+1}, in this case, is associated with the event of a client leaving the service centre which leads to $N-1$ clients in comparison to P_k. Consequently, we find for a steady-state that $\rho < 1$, since $P_{k+1} < P_k$. Generalising the previous equation results in:

$$P_k = \rho^k P(0)$$

The normalisation of P leads to

$$\sum_{k=0}^{\infty} P_k = \rho^k P(0) = 1$$

$$P(0) \sum_{k=1}^{\infty} \rho^k = 1$$

Thus the mean number of clients in the service centre at a certain time t can be expressed as the expectation value $E[k]$:

$$E[k] = P(0) \sum_{k=1}^{\infty} k\rho^k$$

To evaluate $E[k]$, one first computes the average of $\sum_{k=1}^{\infty} \rho^k$, which can be expressed by the geometric series decomposition where is S=1 (total probability):

$$S = P(0) + P(0)\rho^2 + ...P(0)\rho^{k-1};$$
$$-\rho S = -P(0)\rho^2 - ... - P(0)\rho^k,$$

to find an expression for P(0).

Subsequently, we take the difference $S - \rho S$

$$S - \rho S = P(0) - P(0)\rho^k$$
$$S(1-\rho) = P(0)(1-\rho^k)$$
$$S = \frac{P(0)(1-\rho^n)}{1-\rho}$$

Because $\rho < 1$, ρ^k with $k \to \infty, \rho^k \to 0$. Rearranging the previous equation yields

$$P(0) = 1 - \rho, \tag{1}$$

which defines the probability of having zero jobs during the steady-state.

The average of

$$\sum_{k=1}^{\infty} k\rho^k$$

can be expressed by the geometric series decomposition.

$$\begin{aligned} S &= P(0) + 2P(0)\rho^2 + ... + kP(0)\rho^{k-1} \\ -\rho S &= -P(0)\rho^2 - ... - kP(0)\rho^k \\ S - \rho S &= P(0) + P(0)\rho^k \end{aligned}$$

where $P(0) + ... + P(0)\rho^k$ corresponds to the Maclaurin Series and can be written with $\rho < 1$ to $\frac{\rho}{1-\rho}$. Thus resulting in:

$$S = \frac{\rho}{(1-\rho)^2}$$

Plugging this back into the expression for $E[k]$ gives:

$$E[k] = P(0)(\sum_{k=0}^{\infty} k\rho^k) = P(0)\frac{\rho}{(1-\rho)^2}$$

Since we know that $P(0) = 1 - \rho$, it follows that

$$E[k] = \frac{\rho}{1-\rho} \tag{2}$$

which corresponds to the mean number of clients in the service centre (Adan & Resing, 2001).

An important concept is Little's law, which describes the relationship between the average number of clients in the system, the mean arrival time λ and the mean time W spent in the system by the following equation (Little, 1961):

$$W = \frac{1}{\lambda}E[k], \tag{3}$$

which in the case of an M/M/1 queue can be written as

$$W_{M/M/1} = \frac{\rho}{\lambda(1-\rho)}. \tag{4}$$

In case of ρ tending to unity, $E[k]$ and W grow to infinity. Thus, the average waiting time in the queue can be expressed by

$$W_{M/M/1_q} = W - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}, \tag{5}$$

where $\frac{1}{\mu}$ is the average of the exponential distributed service time.

For an M/M/2 queue, the system load ρ is assumed to be the same. However, a second server is provided. The following situation is schematised with m being the number of servers in figure 2.

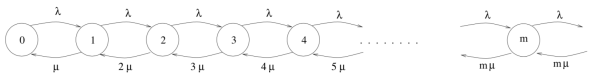

Figure 2.5: CTMC for the M/M/m queue

Figure 2: CMTC model where λ defines the arrival rate and μ the capacity of the server. Each number represent a state with the corresponding clients in the queue and m the number of servers (Willig, 1999).

Since the M/M/2 has the same memoryless properties as M/M/1, the system can be described by the number of clients present and the CMTC can be used to solve for the probability P_k. However, due to the fact that 2 servers are present, we have $\rho = \frac{\lambda}{2\mu}$, i.e. $\frac{\lambda}{\mu} = 2\rho$. Based on the general formula derived by Willing (1999), $E[k]$ can be found via the following steps:

$$\sum_{k=0}^{\infty} P(k) = 2\rho^k P(0) = 1$$

$$P(0) + 2P(0)\rho\sum_{k=1}^{\infty}\rho^{k-1} = 1$$

$$P(0) + 2P(0)\rho\sum_{k=0}^{\infty}\rho^{k} = 1$$

Using the previous results of the geometric series

$$P(0) + 2P(0)\rho\frac{1}{1-\rho} = 1$$

$$P(0) = \frac{1-\rho}{1+\rho} \tag{6}$$

Next, the expectation value of the number of clients in the system at state k can be expressed as

$$E[k] = 2P(0)(\sum_{k=0}^{\infty} k\rho^k) = 2\frac{(1-\rho)}{(1+\rho)}\frac{\rho}{((1-\rho))^2}$$

From which we get

$$E[k] = \frac{2\rho}{1-\rho^2} \tag{7}$$

Using Little's law again, the average waiting time in the system corresponds to

$$W_{M/M/2} = \frac{2\rho}{\lambda(1-\rho^2)} \tag{8}$$

Thus, for 2 servers the average waiting time in the queue is

$$W_{M/M/2_q} = W_{M/M/2} - \frac{1}{\mu} = \frac{\rho^2}{\mu(1-\rho^2)} \quad (9)$$

Comparing the two different waiting times, $W_{M/M/2}$ is by a factor of $\frac{1-\rho}{1-\rho^2}$ shorter than a system with only one server. In case of similar ρ, naively, one would expect that the two systems ($M/M/1$ & $M/M/2$) would have the same waiting time, which is shown not to be the case.

For a more general formula with n number of servers, formulas 10 and 11 can be used. However, the derivation of this formula goes beyond the scope of this report.

In a general fashion, one has to compute the probability that a job has to wait

$$\Pi = \frac{(np)^n}{n!}\left((1-\rho)\sum_{c=0}^{n-1}\frac{(c\rho)^c}{c!} + \frac{(c\rho)^n}{n!}\right)^{-1} \quad (10)$$

Using Π, one can express the mean waiting time in queue $W_{M/M/n_q}$ with the following equation

$$W_{M/M/n_q} = \Pi\frac{1}{1-\rho}\frac{1}{n\mu} \quad (11)$$

3. Methodology

The goal of this paper is to examine the average time spent in a queue for different queueing set-ups. The first set-up that was explored, is the M/M/n queue. Of this set-up, the average queueing time as a function of ρ was examined for one server. To this end, multiple simulations were done with varying values of μ, i.e. varying server capacities. Secondly, the average queueing time for different numbers of servers was investigated. This effectively also changes the value of ρ, where $\rho = \frac{\lambda}{n\cdot\mu}$, n being the number of servers. To keep the workload constant, λ was increased linearly with the number of servers. Thirdly, instead of using FIFO as a serving discipline, as was done in the aforementioned experiments, we compared the results to the SJF service discipline.

Additionally, the dynamics of different service rate distributions were explored. The M/D/n set-up was one of the service rate distributions that was looked at, as well as a long tail distribution where 75% of the exponentially distributed service rates had an average service rate of 1, while the other 25% had an average service rate of 5. For these service rate distributions, we looked at the average waiting times for different numbers of servers. Furthermore, we investigated how various average service rates affect the waiting time in the queue.

In order to model the queuing simulations, the SimPy framework was used extensively (Simpy, 2017). The process-based, discrete-event simulation framework allows a simple implementation of the desired environment incorporating the different types of events and the numbers of servers present. In addition, it allows for the implementation of requests to a service centre, with the possibility of priority based serving, which suits the goal of simulating queues.

To make the events Poisson distributed, the random generator of the NumPy library was used (McKinney, 2012). To generate the pseudo-random numbers, the NumPy library makes use of the Mersenne Twister algorithm, producing 53-bit precision floats (Matsumoto & Nishimura, 1998). For visualisation of the plots, the MatPlotLib package is used (Hunter, 2007). Figure 3 and 4 show a Poisson distribution of the interarrival and servicing time with $\mu = 0.8$ and $\lambda = 0.64$.

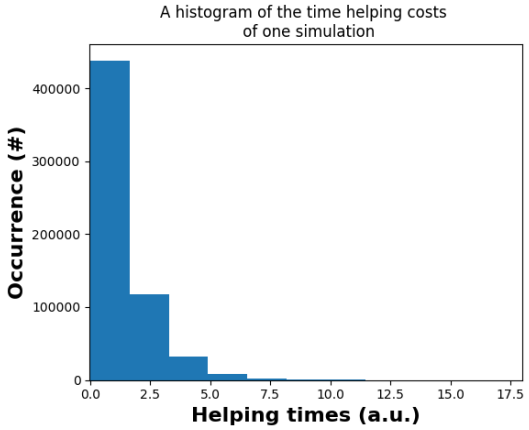

Figure 3: Exponential distribution of the servicing time with $\mu = 0.8$ for a simulation of 6×10^5 services executed.

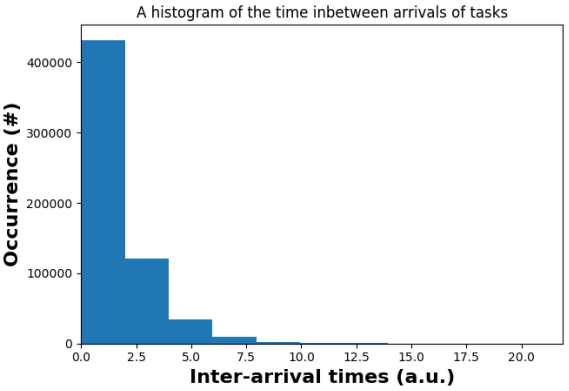

Figure 4: Exponential distribution of the interarrival time with $\lambda = 0.64$ for a simulation of 6×10^5 services executed.

In order to quantify the significance of the results in the steady-state simulation, the variance of the waiting time in the queue was calculated by means of the Batch Means (BM) method (Andradóttir &

Argon, 2001). In order to obtain σ^2, one divides a single long run in N number of non-overlapping batches of the same size m. Subsequently, every second batch is used in order to avoid correlation between the batches, resulting in $kb = \frac{1}{2}N$ batches used to determine the overall mean value. For each chosen batch the mean $\overline{X}_j$ is computed. The size of one simulation was defined by 6×10^5 services executed.

$$\overline{X}_j = \frac{1}{m} \sum_{i=1}^{m} X(j-1)m + i$$

Subsequently, the variance is computed by taking the difference of the mean from each batch with respect to the mean $\overline{X}_{BM}$ of the batches combined.

$$V_{BM} = \frac{1}{kb-1} \sum_{j=1}^{kb} (\overline{X}_j - \overline{X}_{BM})$$

The standard deviation can then be calculated with $\sigma = \sqrt{V_{BM}}$ (The Network Lab, 1970). By applying the Batches Method, the batch means are decorrelated, which makes makes them iid (Andradóttir & Argon, 2001). Of every simulation, the first 10.000 tasks were removed to avoid the influence of the initialisation parameters, i.e. the queue being initialised at length 0.

Figure 5 shows the distribution of the batch means of size 8000 in an M/M/1 simulation where the first 10×10^3 services executed out of 3.6×10^7 were ignored in order to avoid any initialisation bias. The distribution approximately resembles a Gaussian curve which allows the application of the central limit theorem in order to determine the confidence interval (C.I.) (Le Cam, 1986).

$$\overline{X}_{BM} \pm z_\alpha \frac{\sigma}{\sqrt{kb}} \tag{12}$$

Equation 12 states that given a number of batches, the probability of a value not deviating more than $z_\alpha \frac{\sigma}{\sqrt{kb}}$ from the true value $\overline{X}_{BM}$ is given by a confidence interval of $1 - 2\alpha$. In this report $\alpha = 0.025$ to get an C.I. of 95%.

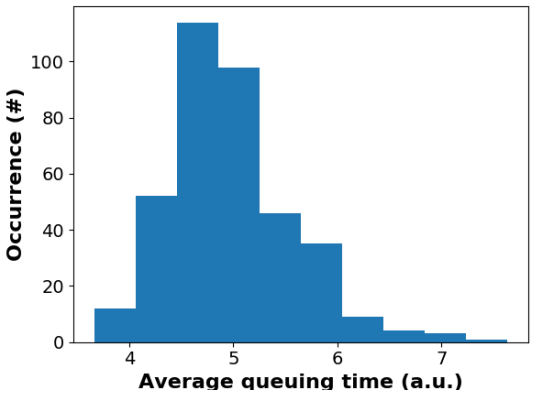

Figure 5: Gaussian distribution of the batches' means of size 8000 in an M/M/1 simulation with an initialisation period of 10000 services executed. The size of the simulation is defined by 3.6×10^7 services executed and the service discipline is FIFO. As parameters, $\mu = 0.8$ and $\lambda = 0.64$ were used.

4. Results & Discussion

The intrinsic nature of the studied service centre requires that the interarrival and servicing time have a large probability of being short. The relative shortness is defined by the respective paramters μ and λ. Figure 3 and 4 show the exponential distributions of the two events. Although these parameters are exponentially distributed, the mean waiting time in the queue is normal distributed as shown by figure 5. Based on the following premise, the central limit theorem was applied to define the 95% C.I.

To minimise the variance on the average queueing time, the best size of the batches was determined to minimise the standard deviation σ. In order to avoid any initialisation bias, the first 10×10^3 services were excluded and, additionally, the simulation was repeated 30 times to account for stochastic effects. Figure 6 shows how the standard deviation varies with the size of the batches.

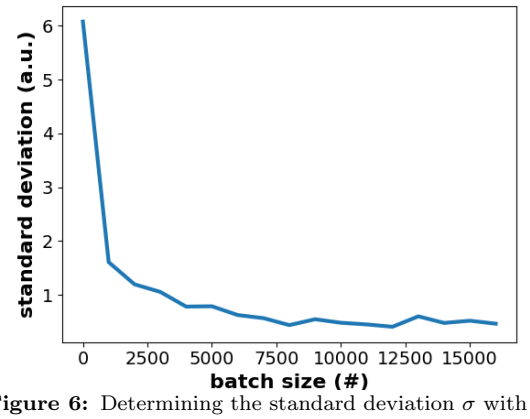

Figure 6: Determining the standard deviation σ with different batch sizes in an M/M/1 simulation with FIFO as a service discipline. To reduce the stochastic effects, the simulation was repeated 30 times. Thereby, the initialisation period was set to 10×10^3, $\mu = 0.8$ and $\lambda = 0.64$.

The size of the batches influences the standard deviation. Batches of small sizes are subjective to correlation error which induces high standard deviation whereas bigger batch sizes average out the fluctuation in the data. Based on figure 6, 8000 as number of batches was judged to generate adequate standard deviation without compromising the running time.

Figures 9, 10 and 11 show the distribution of the means of the different batches for the M/M/1, M/M/2 and M/M/4 FIFO simulations. The figures are taken up in the appendix. The working load ρ was kept constant to allow a comparison by increasing λ. The gaps between the various bins are due to the number of batch averages available which limits a smooth histogram. Due to computational constraints, the number of batches was set to 8000 for simulations of 600000 simulations which leads to 38 data points.

Table 1 summarises the previously mentioned histograms in terms of the total mean and the 95% C.I. What stands out in the table is the decreasing mean waiting time in the queue with an increase number of servers with the same working load ρ.

Table 1: Mean value of the waiting times in the queue for a M/M/n environments of 6×10^5 services using FIFO disciplines.

	Mean Value	95%C.I
M/M/1	4.956	+- 0.184
M/M/2	2.233	+- 0.027
M/M/4	0.900	+- 0.006

Using the equation 5, one gets a theoretical mean waiting time of 4.999 for an M/M/1 system using the FIFO discipline. For the M/M/2 system using the FIFO discipline, the equation generates a mean waiting time in the queue of 2.22. For the M/M/4 the general equation 10 was used which generated a value of 0.931. Comparing the theoretical results with the experiments, one can observe an overlap of the values for M/M/1 and M/M/2. However, the theoretical value of M/M/4 lies outside the confidence interval of the experimental values.

Figures 15, 16, 17 show the distribution of the mean of the different batch for the M/M/1, M/M/2 and M/M/4 for a SJF simulations. Table 2 condenses in information of the histogram distributions in terms of their mean and 95% C.I. The results follow the same trend as in table 1 where the mean waiting time in the queue decreases with an increase in the number of servers available. Interestingly, the SJF data is observed to be smaller than the respective FIFO data.

Table 2: Mean value of the waiting time in the queue for a M/M/n environments of 6×10^5 services using SJF disciplines.

	Mean Value	95%C.I.
M/M/1	2.36	+- 0.010
M/M/2	1.14	+- 0.003
M/M/4	0.48	+- 0.001

One can explain this phenomena by considering the fact that the algorithm services the shortest task in the queue first. The long tasks are postponed until a server is free and, therefore, don't delay the queue. Thus the queue is minimised and the average waiting time in the queue decreases.

After investigating the impact of the number of servers on the mean waiting in the queue, the influence of the working load ρ was analysed. Table 3 presents the experimental mean waiting time for an M/M/1 set-up with a changing ρ. The smaller the ρ, the lower the mean waiting time in the queue and the variance of the value. Due to a high capacity compared to the interarrival time, tasks can be serviced faster than they are likely to appear. Thus, the mean waiting time is short. If ρ approaches unity, the waiting time increases exponentially with a high variance. In order words, the mean average time in the queue varies extraordinarily with ± 2062.83. This value would imply negative waiting time which is not possible. However, it shows the great volatility of the queue length.

Table 3: Comparison of different ρ in a M/M/1 simulation of 6×10^5 services long.

ρ	**Theory**	**Experiment**	**95% C.I.**
0.98	97.95	103.24	2062.83
0.88	14.81	14.481	3.24
0.81	7.39	7.29	0.28
0.75	4.68	4.65	0.059
0.69	3.31	3.27	0.015
0.64	2.49	2.46	0.010
0.61	1.95	1.94	0.007

These findings show that ρ is a key factor in average waiting time. If ρ approaches unity, the queue length is highly influenced by the subsequent events, which induces high variance. Since the servicing time and the interarrival time are similar, one would expect a system with a really short queue where the event of someone leaving and entering the system would happen almost simultaneously. However, one longer task can provoke a queue that accumulates. Based on this reasoning the queue can also decrease rapidly if a task is finished before a new one gets added to the queue. A possible way of reducing the variance could be increasing the duration of the simulation. Another possibility would be that another batch size would better fit higher values of ρ.

Another queueing model that was analysed in this report is the M/D/n model with a FIFO and SJF service discipline. In the appendix, figure 12, 13 and 14 show the distribution of the mean queueing time for the M/D/n model in FIFO. Table 4 summarises the mean values with the respective C.I. of the FIFO discipline.

Table 4: Mean value of the waiting time in the queue for a M/D/n environments of 6×10^5 services using FIFO disciplines. The service time was $\mu = 0.8$

	Mean Value	95%C.I.
M/D/1	2.48	+- 0.023
M/D/2	1.15	+- 0.003
M/D/4	0.47	+- 0.001

As expected from previous results, the mean queueing time decreases with the number of servers. The analysis was taken further by using the M/D/n with SJF environments (figure 18, 19 and 20), which yields the results shown in table 5.

Table 5: Mean value of the waiting time in the queue for a M/D/n environments of 6×10^5 services using SJF disciplines.The service time was $\mu = 0.8$

	Mean Value	95%C.I.
M/D/1	2.45	+- 0.0129
M/D/2	1.12	+- 0.0036
M/D/4	0.47	+- 0.0005

The similarity between the FIFO and SJF of an M/D/n can be explained by the fact that in the deterministic model the service time is constant. Consequently, the tasks require the same amount of service and therefore the SJF is ineffective. Surprisingly, the M/D/n model yields results similar to M/M/n with SJF service discipline. A possible explanation could be that both experiments have the same mean value for μ. and same interarrival time λ.

Finally, the long-tail distribution model was analysed in this report. Thereby, the servicing time μ was switching between two values during the run. The common value was set to 1 with a probability of 75%. Figure 7 shows a long-tail distribution of 5 runs with as the second value of mu 1, 1.5, 3, 4 and 5 with a probability of 25%.

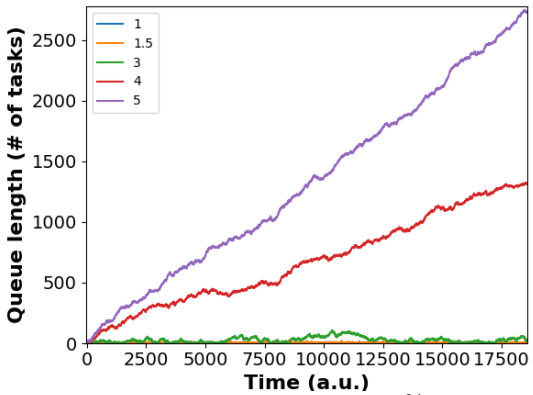

Figure 7: Long-tail distribution with 75% of the jobs requiring 1 seconds of servicing time. 25% of the jobs requiring respectively 1, 1.5, 3, 4, 5 a.u. of servicing time.

For a value of 1, 1.5 and 3 one can see a fluctuating queue length over the time. However, the value of 3 induces slightly more volatility. The values 4 and 5 show a linearly increasing queue length. The higher the value for μ, the steeper the slope. In other words, for small servicing times switching from 1 to 3 a.u., the service centre can handle the arrival tasks in a reasonable fashion by keeping the queue length small enough. However, if the servicing times can take the value of 4 or 5 with a probability of 25% of each state, the service centre immediately is overwhelmed by the tasks and the longer servicing times induce a linear increase of the queue length.

In addition to different values of μ, the long-tail distribution was analysed with different numbers of servers. Figure 8 shows the results where the values for μ were 1 and 5. The data was averaged over 30 runs.

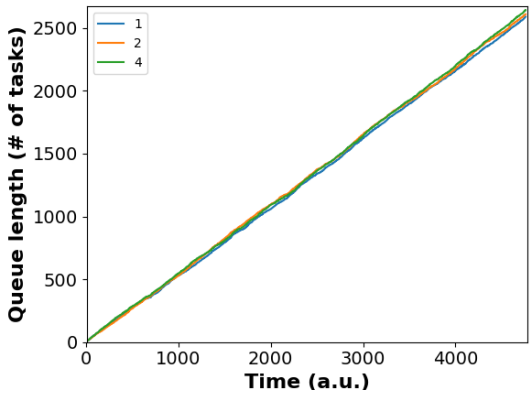

Figure 8: Long-tail distribution with 75% of the jobs requiring 1 a.u. of servicing time and 25% requiring 5 a.u. of servicing time. The simulation was run for 1, 2 and 4 servers. The data was averaged over 30 runs.

One can see that for different servers, the lines are overlapping, which means that the queue lengths increase in the same fashion for all the different number of servers. This implies the number of servers has no impact on the the queuing length in case of a long-tail distribution with with 75% probability of having an average 1 a.u. service time and a 25% change of having an average of 5 a.u. of service time. To complete this picture one could investigate at what ratio of the long-distribution, the queue length starts to increase linearly.

5. Conclusion

The aim of this report was to analyse the different queueing models in terms of number of servers and the workload ρ. The hypothesis that more servers would decrease the average queueing time was experimentally confirmed. Additionally, we identified that for an M/M/n model, the queuing time can be reduced if the *Shortest Job First* discipline is used. Despite the outstanding performance, the SJF would in many practical applications be infeasible, because the operating system may not know before hand what the duration of the tasks in the queue is. Furthermore, it would also imply that long tasks would have a longer average waiting time, which could be impractical depending on your application. As the workload ρ approaches unity, the queueing time is subjective to high variance, which is due to highly sensitive environment. Finally, the simulations have identified that a long-tail distribution can rapidly induce an unstable system in which the queue length increases indefinitely. Further research might explore the dynamics brought about by different arrival rate distributions.

6. Appendix

This section shows the raw data of the experiments discussed in the report.

6.1. FIFO service discipline M/M/n

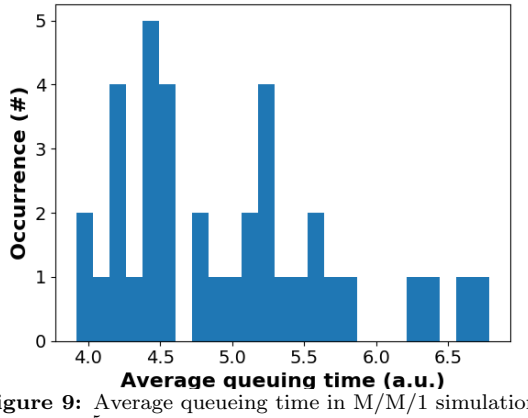

Figure 9: Average queueing time in M/M/1 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was FIFO with $\mu = 0.8$ and $\lambda = 0.64$.

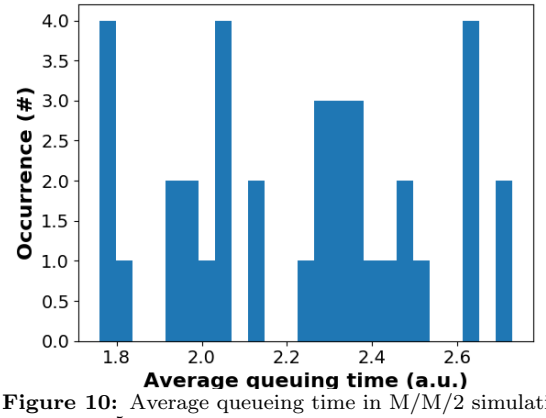

Figure 10: Average queueing time in M/M/2 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was FIFO with $\mu = 0.8$ and $\lambda = 0.128$.

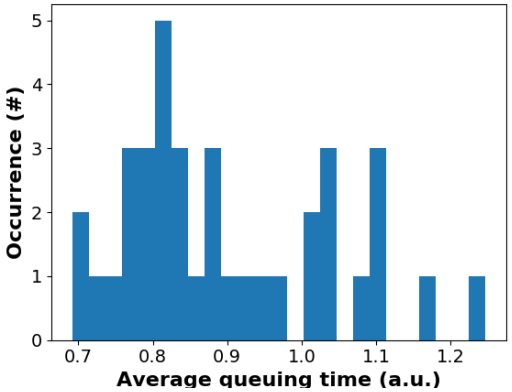

Figure 11: Average queueing time in M/M/4 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was FIFO with $\mu = 0.8$ and $\lambda = 0.256$.

6.2. FIFO discipline M/D/n

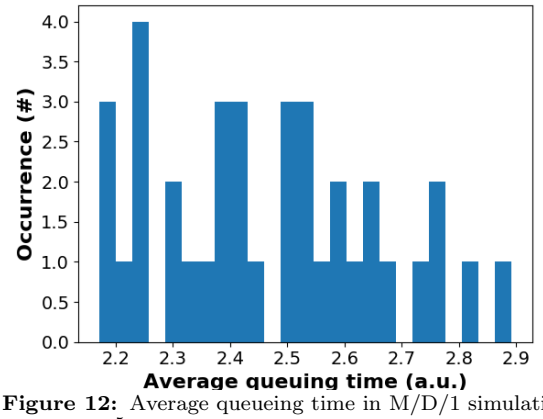

Figure 12: Average queueing time in M/D/1 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was FIFO with $\mu = 0.8$ and $\lambda = 0.64$.

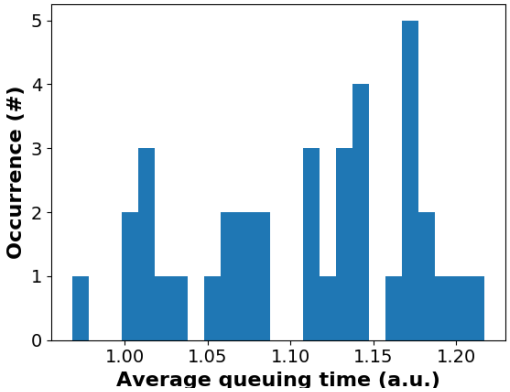

Figure 13: Average queueing time in M/D/2 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was FIFO with $\mu = 0.8$ and $\lambda = 0.128$.

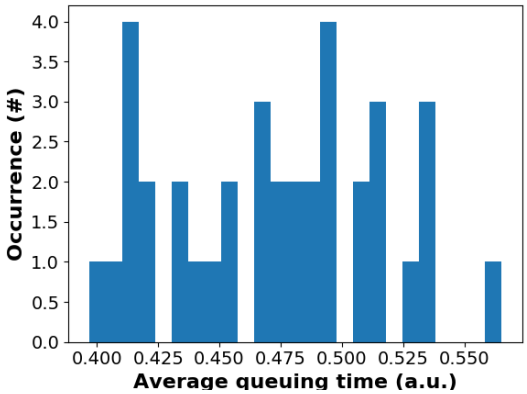

Figure 14: Average queueing time in M/D/4 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was FIFO with $\mu = 0.8$ and $\lambda = 0.256$.

6.3. SJF service discipline M/M/n

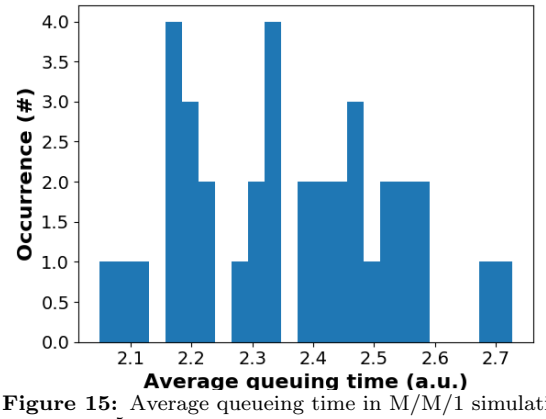

Figure 15: Average queueing time in M/M/1 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was SJF with $\mu = 0.8$ and $\lambda = 0.64$

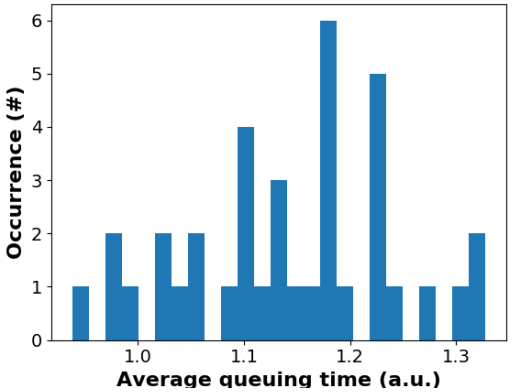

Figure 16: Average queueing time in M/M/2 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was SJF with $\mu = 0.8$ and $\lambda = 0.128$

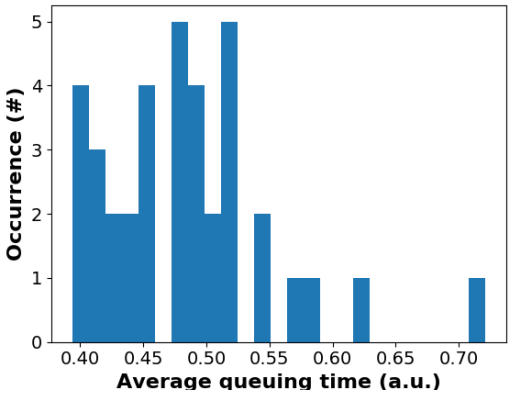

Figure 17: Average queueing time in M/M/4 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was SJF with $\mu = 0.8$ and $\lambda = 0.256$

6.4. SJF service discipline M/D/n

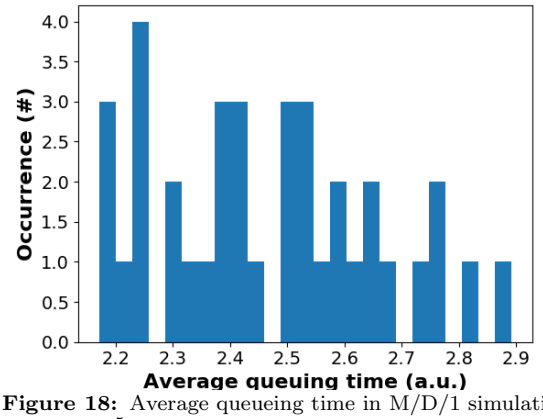

Figure 18: Average queueing time in M/D/1 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was SJF with $\mu = 0.8$ and $\lambda = 0.64$.

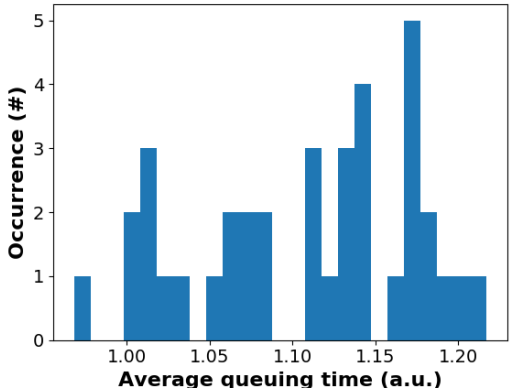

Figure 19: Average queueing time in M/D/2 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was SJF with $\mu = 0.8$ and $\lambda = 0.128$.

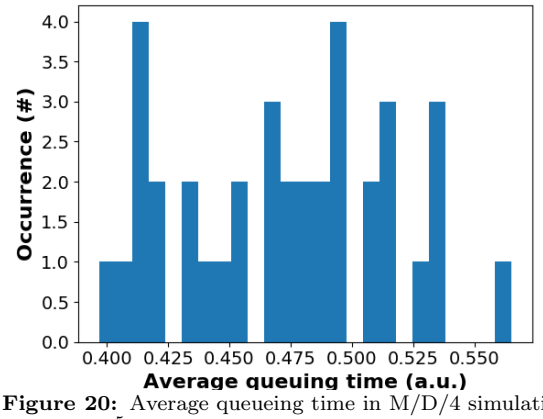

Figure 20: Average queueing time in M/D/4 simulation with 6×10^5 services long and a batch size of 8000. The service discipline was SJF with $\mu = 0.8$ and $\lambda = 0.256$.

References

Adan, I., & Resing, J. (2001). *Queueing theory: Ivo adan and jacques resing.* Eindhoven University of Technology. Department of Mathematics and Computing ....

Andradóttir, S., & Argon, N. T. (2001). Standardized time series methods: variance estimation using replicated batch means. In *Proceedings of the 33nd conference on winter simulation* (pp. 338–343).

Erlang, A. K. (1909). The theory of probabilities and telephone conversations. *Nyt. Tidsskr. Mat. Ser. B*, *20*, 33–39.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, *9*(3), 90.

Kendall, D. G. (1953). Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, 338–354.

Le Cam, L. (1986). The central limit theorem around 1935. *Statistical science*, 78–91.

Little, J. D. C. (1961, jun). A proof for the queuing formula:l= w. *Operations Research*, *9*(3), 383–387. doi: 10.1287/opre.9.3.383

Matsumoto, M., & Nishimura, T. (1998, January). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, *8*(1), 3–30. Retrieved from `http://doi.acm.org/10.1145/272991.272995` doi: 10.1145/272991.272995

McKinney, W. (2012). *Python for data analysis: Data wrangling with pandas, numpy, and ipython.* " O'Reilly Media, Inc.".

Simpy, T. (2017). Overview—simpy 3.0. 10 documentation. *Simpy documentation*.

The Network Lab. (1970, January). *Batch means method.* Retrieved from `http://www.netlab.tkk.fi/opetus/s38148/s02/luennot/E_statanal.pdf`

Willig, A. (1999). A short introduction to queueing theory. *Technical University Berlin, Telecommunication Networks Group*, *21*.