Exploration of Queueing Theory's results using Discrete Events Simulations

Andrea Maestri, 14055031, andreamaestrimaestri@gmail.com
Aron Golombek, 14064103, aron.golombek@student.uva.nl

December 2021

Abstract

This report investigates how queuing theory can be studied using Discrete Event Simulations. Different systems were compared, all of them with an exponential distributed arrival time of the customers but varying distributions for service times. The major conclusions of this paper are that the number of servers in a queuing system strongly influence the mean waiting time in the queue. Furthermore *shortest job first* scheduling preforms better than FIFO scheduling for M/M/1 queues. The last important point of the report is that changing the service time distribution to a distribution with a higher variance, leads to a deficiency in the accuracy of the simulated value compared to the analytical expected value.

Introduction

Queuing theory refers to the mathematical analysis of queues which can play an essential role in resource allocation concerning operations for many distinct applications such as communication networks, computer systems, or even to simulate a queue in an airport or a shop[1]. It is mainly seen as a branch of applied probability theory and there is a strong mathematical foundation for the field allowing analytical solutions for certain parameters of specific queues. Stochastic simulation plays a vital role in the analysis and optimization of different queuing systems. The goal of such analysis is trying to estimate certain system parameters such as the mean waiting time in the queue, the queue's mean length and so on.

Some introductory books and papers from which the theory used in this report was developed are [2], [3] and[4] as well as [1] and [5]. In this material, the major focus is on the mathematical derivations of the theory.

In support of the theory, the aim of this report, is to verify the behaviour of specific queuing system using *discrete event simulations* as well as to extend the analysis to systems for which analytical solutions are less well documented. When possible the simulated answers will be compared with the analytical solutions. In particular, this report will analyse how varying certain system parameters such as the system load (i.e. the rate of arrivals divided by the service rate) and server quantity (e.g. the numbers of check-in desks in an airport) affects specific metrics of a system, including the previously mentioned average waiting time in the queue and iterations necessary to acquire statistically relevant results. Moreover, it will investigate the convergence of a queuing system to a steady state changing the number of customers.

This report consists of four sections. Section 1 introduces important aspects of general queuing theory as well as more in depth mathematical analysis of the queue systems investigated in this report. Subsequently section 2 will elaborate of the simulation methods used to model the individual queuing systems. This is followed by a presentation an analysis of the obtained results in section 3. Lastly some general conclusions about this topic will be drawn in section 4.

1 Theory

This section will begin by developing an intuition for the distinct factors influencing queuing systems and will expand on this to introduce the stochastic framework which is used for their formal mathematical description. Additionally, the notation used to differentiate different queuing systems will be introduced alongside the distinct queuing systems treated in this report. Lastly, some mathematical theory relating to queuing systems will be introduced alongside the analytical analysis of two particular systems.

1.1 Queuing Theory and Notation

Intuitively, the length of a queue depends on three simple parameters, the arrival rate of customers, the service time required per customer and the number of servers. In order to create a stochastic framework for this the arrival rate of customers as well as the service time can be represented as distinct distributions. Furthermore, different queue types can exist for different applications. Most commonly the first in first out (FIFO) ordering is used where the first task or person to arrive in a queue is treated first. However, certain applications in computing for example use last in first out (LIFO) ordering or other forms of prioritization in order to attempt to optimize the queuing system. The so called Kendall Notation describes a queuing system using the mentioned aspects and is represented as:

$$A/B/m/N - S \tag{1}$$

Where A and B represent the distribution type of customer arrivals and service times respectively. If the arrival and the service time are random variables distributed accordingly to an exponential exponential distribution with rate λ and μ, then $E[A]^{-1} = \lambda$ which is the reciprocal of the mean arrival time and $E[B]^{-1} = \mu$ which is the reciprocal of the mean service time. The number of servers is given by m. N is the maximum queue size which for all cases treated in this report will be assumed to be infinite. Lastly S represents the queuing discipline in which tasks are prioritized. Using the parameters defined above, a useful variable to introduce is the system load, defined as[1]:

$$\rho = \frac{\lambda \cdot E[B]}{m} \tag{2}$$

It can be thought of as extent to which the system is used to capacity. In the following subsections the distribution types and prioritization methods investigated in this report will be introduced.

1.1.1 Distributions

Three distinct time distributions will be analysed in this report. These will be introduced briefly below with the letter in brackets representing the abbreviation used in Kendall Notation.

Memory-less Distribution (M) The memory-less distribution is also known as an exponential or Markov distribution: $A(t) = 1 - e^{-\lambda t}$. Where $A(t)$ denotes the cumulative distribution function. It is the distribution of the inter-arrival or times of a Poisson Process. The name memory-less stems from the fact that the inter-arrival times are completely independent of each other.

[1]However, the value of the system load can be find for a single server using the Little's law, described in the following subsections

Deterministic Distribution (D) The Deterministic Distribution is characterized by no variation in the generated variables leading to arrival of customers at a constant rate as well as a constant service time for all customers.

Long-tail Distribution (L) A long-tail distribution (or hyper-exponential distribution) with k phases is defined as follow[2]:

$$a(t) = \sum_{j=1}^{k} a_{\mu i}(t) p_i. \tag{3}$$

It is essentially a combination of different exponential distributions, each occurring with a certain probability p_i.

1.1.2 Queuing Disciplines

While several more queuing disciplines exist, the report will investigate only two distinct disciplines.

First In First Out (FIFO)
As mentioned previously this merely describes that the customers or tasks are treated in the order in which they arrive. If not mentioned specifically this queuing discipline will be used as the standard discipline.

Shortest Job First
As the name suggests this discipline disregards the order in which customers or tasks arrive and rather prioritizes tasks based on the time it takes to complete them. The jobs are ordered according to the time it takes to complete them and the shortest task is preformed first.

1.2 Mathematical Theory

Here some mathematical theory used in this report relating to queuing systems will be introduced.

1.2.1 Little's Law

Little's law gives an important relation between the mean number of customers in the system and the mean sojourn time (i.e. the time spent in the system). A rigorous proof is given in [5]. It's generality allows it to be used in several ways, for example to find the average waiting time in the queue $E[W]$ from the average length of the queue $E[L^q]$:

$$E[L^q] = \lambda \cdot E[W] \tag{4}$$

[2]Where $a(t)$ denotes the probability density function

and also for a single server's load:

$$\rho = \lambda \cdot E[B].$$

1.2.2 Pasta Theorem

In this report, all the systems have a Poisson arrival rate. For this type of systems $M/\cdot/\cdot$ any arriving customer will on average find the same situation in the queuing system as an outside observer looking at the system at an arbitrary point in time. A rigorous proof is given in [5], however, as an heuristic explanation, this property is due to the absence of memory in the exponential distribution.

1.3 Analytical Analysis of M/M/1 and M/M/n queues with FIFO ordering

Given a fixed value for the system load ρ, the expected behaviours of the M/M/1 and M/M/n queues differ. In particular, when $\rho < 1$, the expected waiting time in the system will vary drastically with the number of servers n. In this section, a brief justification of this will be given followed by an analytical derivation of this behaviour.

1.3.1 Heuristic explanation

From an heuristic point of view, the natural outcome for this comparison is that the M/M/n queue will have a smaller value for the mean waiting time $\bar{W}$. The reason for this is that, since the capacity of the server is a random quantity, having more servers will guarantee that if for a particular customer the service time is bigger than $1/\mu$, that particular customer will occupy just one server while the others will remain available for the other customers. In the M/M/1 queue on the other hand, if a customer has a particular high service time, then the others must wait the end of that service. Therefore having more servers guarantee a more "regular" service.

1.3.2 Mathematical derivation

In order to derive the mean time in the system analytically, it is better to start with a time dependent probability and take the limit $t \longrightarrow \infty$ for the steady state. In a queue system, the probability of a certain state is the sum of the probability of the flow in that state minus the probability of the flow out of that state. Therefore, for a M/M/1 queue, the probability of being in the state k is:

$$\frac{dP(t)_k}{dt} = (P(t)_{k-1} - \mu P(t)_{k-1}) - (\lambda P(t)_k + \mu P(t)_k).$$

After taking the limit $t \longrightarrow \infty$ and after solving the recursive equation the solution is:

$$p_k = \left(\frac{\lambda}{\mu}\right)^k p_0.$$

Using the normalization: $\sum_{k=0}^{k=\infty} p_k = 1$, the result is:

$$p_0 = 1 - \frac{\lambda}{\mu}.$$

Using this information and the Little's law eq. (4), the mean waiting time is[3]:

$$\bar{N} = \sum_{k=0}^{k=\infty} kp_k \to \bar{T} = \frac{\bar{N}}{\lambda} = \frac{1}{\mu - \lambda}. \tag{5}$$

For a M/M/n queue, there are two cases, when the customers are less then the number of servers, and when they are more. This leads to the following recursive equation:

$$\lambda p_{k-1} = min(k, c)\mu p_k$$

and it gives:

$$p_k = \begin{cases} \frac{(n\rho)^k}{k!} p_0, & k \leq n \\ \rho^k \frac{(n\rho)^n}{n!} p_0, & k > n \end{cases}$$

with p_0 that can be found from the normalization. Using this results, the resulting mean queue length and waiting time are:

$$\bar{N} = \sum_{k=0}^{k=\infty} kp_{n+k}$$

and

$$\bar{T} = \frac{\bar{N}}{\lambda} = \frac{(n\rho)^n}{n!(1-\rho)(n\mu)} \cdot \left((1-\rho) \sum_{k=0}^{n-1} \frac{(n\rho)^k}{k!} + \frac{(n\rho)^n}{n!} \right)^{-1}. \tag{6}$$

Comparing eq. (5) and eq. (6), and using the following set of parameters

- M/M/n: $n = 2$, $\lambda = 18\ min^{-1}$, $\mu = 10\ min^{-1}$
- M/M/1: $\lambda = 9\ min^{-1}$, $\mu = 10\ min^{-1}$

the formulas above give the following results:

- $\bar{T}_{M/M/n} = 0.426\ min$
- $\bar{T}_{M/M/1} = 1\ min$

1.4 Analytical Analysis of M/G/1

This section will provide a generalisation of the mean waiting time for a system with one server when the mean service time is from a generic distribution G. The main information of this sections are from [5].

[3] A more complete proof is given in [1]

A new arriving customer, before his service has to wait for the *residual service time* of the customer in service and all the service time for the customers in the queue. Denoting the residual service time as R, the average serving time for this general system is:

$$E[W] = E[L^q] \cdot E[B] + \rho E[R].$$

It can be demonstrated that the residual service time's moments are described as follows [5]:

$$E[R^n] = \frac{E[B^{n+1}]}{(n+1)E[B]}. \tag{7}$$

Therefore, using the Little's law:

$$E[W] = \frac{\rho E[R]}{1-\rho}. \tag{8}$$

With a similar approach, it is possible to demonstrate that the variance of the mean waiting time is given by:

$$\sigma^2(W) = \frac{\rho}{(1-\rho)^2}(\rho E[R]^2 + (1-\rho)E[R^2]). \tag{9}$$

2 Methods

This section will provide some information on the python library used for the simulations preformed for this paper as well as explaining the distinctions between how the individual queuing systems are simulated in python.

2.1 SimPy Library

SimPy is a library based on standard python allowing many functionalities for process-based discrete event simulation [6]. It allows for the modelling of active component such as customers as well as the simulation of shared resources as required when several servers are to be simulated [6]. Within the SimPy environment processes can be described as functions yielding certain results. These processes can then interact with each other to simulate more complex systems.

2.2 Simulating Queuing Systems in Python

While the structure of the algorithm simulating the different queuing systems is almost identical, slight difference exist in how specific parameters are generated and used. Thus, first the general outline of the algorithm structure will be explained. Subsequently, the details of how each queue system used in this report generates and uses the central parameters will be elaborated upon.

2.2.1 Algorithm Structure

Two functions are required, one to generate customers and hence simulate the process of customer arrival and one to generate the service interaction. The latter function consists of the customer requesting service, waiting until a resource is free to serve him and subsequently being served. Both the wait and service times are recorded. The other function generates a specific number of customers at intervals dictated by the distribution type and and calls the service function to execute the process.

For all simulations in this report the customer arrival is modelled by a memory-less distribution depending on the mean rate of customer arrivals λ. In the simulation $random.expovariate(\lambda)$ is used to randomly generate the time t until the next customer arrival. This is executed by calling the timeout function in the SimPy environment with t as the argument. The service time is simulated in a similar fashion, however due to the fact that these are modelled using different distributions for the different queuing systems in this report, the way in which μ is generated varies between the experiments. This will be discussed in section 2.2.2. A short summary of how the mean waiting time of customers for a given system is found is provided below.

1. Set the number of servers n to the desired value.
2. Set the number of customers.
3. Set the system load ρ and the service rate μ leading to the customer arrival rate per server $\lambda_{server} = \rho\mu$ as well as the arrival rate into the system $\lambda_{system} = n\rho\mu$.
4. Create an instance of the SimPy environment using simpy.Environment
5. Create an empty lists for the wait times.
6. Create the servers using the $simpy.Resource()$ function with the environment and n as arguments.
7. Create a process within the environment using the customer generation function as an argument. This will:
    - Generate customer arrivals according to the specific distribution and λ_{system}.
    - Create the service process using the service function which allows the customer to request a server, wait until a server is free, be served and leave. The wait time of the customer is appended to the list of wait times.
    - Preform this process for the number of customers set previously.
8. The average of the wait times is calculated using numpy.mean().

In order to increase the confidence in the mean wait time value, this process is preformed at least 100 times and until the standard deviation of the mean wait times is less than 1%. The average mean wait time and the standard deviation are updated iteratively using eq. (10), eq. (11) and eq. (12) [2]. Here A is the global average of the mean wait times of each simulation.

$$\overline{A}_{j+1} = \overline{A}_j + \frac{A_{j+1} - \overline{A}_j}{j+1} \tag{10}$$

$$S_{j+1}^2 = \left(1 - \frac{1}{j}\right) S_j^2 + (j+1) \left(\overline{A}_{j+1} - \overline{A}_j\right)^2 \tag{11}$$

$$\sigma = 1.96 \frac{S}{\sqrt{j+1}} \tag{12}$$

This results in a 95% confidence interval in that the calculated mean wait time differs less than 1% from the real converged value.

2.2.2 Differences between Queuing Systems

As mentioned previously the only difference between the queuing systems investigated in this report is the distribution of service times. This is with once exception, namely M/M/n Queues with Shortest Job First Ordering where the queuing discipline is different. In all the different scenarios the value of the service rate is fixed to $\mu = 1/5 \; min^{-1}$ (except for the long tail distribution) and the value of λ is changed in order to obtain the desired value for the system load.

M/M/n Queues with FIFO Ordering
For this queuing system the service times are distributed using a memory-less distribution. Thus similarly as explained previously for the generation of customer arrival times, the service time for every customer is generated using $random.expovariate(\mu)$.

M/M/n Queues with Shortest Job First Ordering
While the generation of service times is identical as described for M/M/n queues with FIFO ordering, the ordering discipline is different. Because SimPy uses FIFO as the standard ordering, a different prioritization needs to be introduced. This can be achieved using a priority statement when the customer request service. The priority is set equal to the service time, leading to the shortest tasks being treated first.

M/D/n Queues with FIFO Ordering
Because the service time is deterministic, μ is constant leading to a constant service time of $\frac{1}{\mu}$.

M/L/n Queues with FIFO Ordering
To generate the service time for a customer using the long-tail distribution, a random number is generated accordingly to eq. (3)

2.3 Hypothesis testing

In order to investigate if the simulated values for the mean waiting time come from the same distribution, the Welch test will be used. In particular, this report will analyse if, after the simulations, it can be stated that for different values of the number of servers, the mean waiting time is significantly different (i.e. if the two analysed values are from two different distributions or if the difference is just due to the stochasticity). The Welch test require to compare the actual value[4]:

$$A_{val} = \frac{\bar{W_{q_1}} - \bar{W_{q_2}}}{\sqrt{\frac{S^2_{q_1}}{n_{q_1}} + \frac{S^2_{q_2}}{n_{q_2}}}} \tag{13}$$

to the critical value from a T-distribution. However, in this case, the degrees of freedom are always enough to approximate the T-distribution to a Gaussian distribution. Therefore the critical value will be:

$$C_{val} = 1.96.$$

Using this value, the hypotheses will be accepted with a confidence level of 95% if $|A_{val}| < |C_{val}|$.

In the same fashion, to compare the simulated value to the expected value from an analytical analysis, is it possible to use the T-test. In this case the actual value that needs to be compared to the critical value is the following:

$$A_{val} = \frac{E[W] - \bar{W}}{\sqrt{\frac{S^2}{n}}} \tag{14}$$

Where $E[W]$ is the value from the theory introduced in section 1.4.

3 Results

This section will firstly analyse how the number of customers influence the distribution of the mean waiting time. Subsequently an analysis of how the number of servers in a M/M/n, M/D/n and M/L/n queuing system with a FIFO scheduling system influence the behaviour of the system will be presented. In a similar fashion a comparison between the FIFO and *shortest job first scheduling* is presented. All of the results will be tested for their statistical relevance using the T-test or the Welch test. Specifically it will be tested whether the simulated values are compatible with the theoretical distributions and, for different values of the servers number, if the difference between the simulated values is only due to the stochasticity of the system.

[4]where W is the waiting time, S is the sample variance, n is the number of iterations required to have an uncertainty of $0.01 \cdot W$ with a confidence level of 95% and q_i are two different system e.g. M/M/2 and M/M/4

3.1 System behaviour with different number of customers

In order to study how a queuing system behaves, it is necessary to study it in a steady state as the real behaviour of a queue starts after the adjustment period in the beginning of the simulation. Therefore, this part of the results will investigate how different values of the system load influence the number of customers required for the system to be in such a state. All of the simulations have been preformed for a M/M/1 queue.

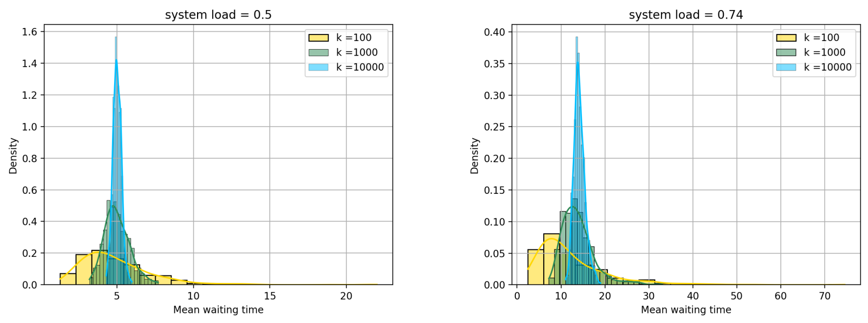


Figure 1: Average wait time for different values of the customers' number

Figure 1 shows that, after 500 simulations, for $\rho = 0.5$ and $\rho = 0.74$ the system starts to behave as expected (i.e. the distribution of the mean waiting time is approaching a normal distribution) from $k = 1000$. However, already for $k = 100$ the distribution starts to resemble a normal distribution. On the other hand, it is worth noticing that, a distribution with a smaller variance (i.e. for $k = 10000$) will lead to a smaller number of iterations to obtain the desired confidence level.

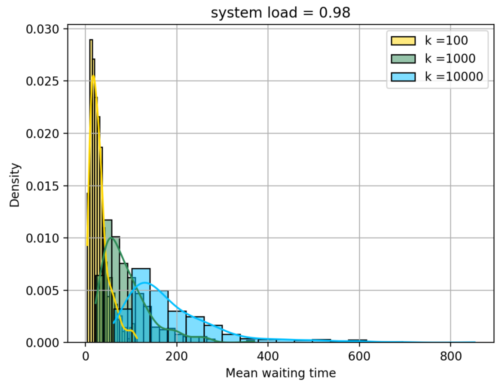


Figure 2: Average wait time for different values of the customers' number

Conversely Figure 2 shows that increasing the value of the number of customers, leads to an increase in variance. An heuristic explanation to this is that, if ρ is approaching 1, the arrival rate and the service rate are approximately the equal. Therefore, if the number of customers are higher, it is more likely to see a wider range of behaviours for the mean waiting time. Looking at these three plots, in order to balance the computational expense of a simulation, while ensuring the system will be in a steady state, the number of customers for the rest of the report will be fixed to $k = 10000$. This decision is expected to have some benefits, as well as some drawbacks. The main disadvantage is that a higher number of customers requires more computational effort. However, with a higher number of customers, the system is more likely to be in a steady state. Moreover, as it is possible to see from the plots, a large number of customers means that less repetitions of the simulation are necessary to reach the confidence interval as the variance between the individual simulations will be smaller, assuming $\rho \neq 1$.

3.2 M/M/n Queues with FIFO Scheduling

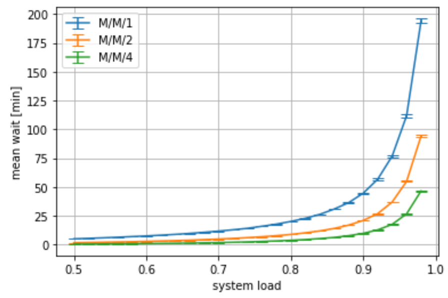
Studying a M/M/n queuing system, with $n \in [1, 2, 4]$ for different values of ρ, the mean waiting time changes. The mean waiting time decreases as the number of servers increase for a given ρ as seen in Figure 3a as well as table 1 in Appendix A.

In particular, using the Welch test (table 5, table 6 in Appendix B) it is clear that we can reject the hypothesis that the values of the waiting time comparing different numbers of servers come from the same distribution. Furthermore, with the T-test the hypothesis that the expected values and the simulated values are from the same distributions can be accepted when the system load is not close to 1. On the other hand, when ρ is approaching 1, the system starts to be more stochastic. In fact, from table 1 in Appendix A it can be seen that for $\rho = 0.98$ The hypothesis must be rejected.

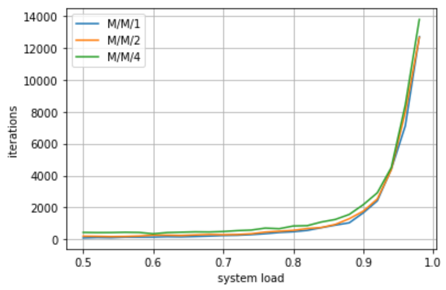
Furthermore as seen in Figure 3b, the number of the iterations required to reach the designed confidence interval increase exponentially for $\rho \to 1$. However, this number is approximately the same for all the values of n.

3.3 M/M/n Queues with Shortest Job First Scheduling

While FIFO scheduling is certainly the most common, it is not necessarily the most efficient. In fact, from fig. 3a and from table 2 in appendix A it is clear that the *Shortest job First Scheduling* results in a smaller mean waiting time. The Welch test was used to verify with a 95% confidence interval that the difference is not merely due to stochasticity. This can be seen in table 7 in appendix B.

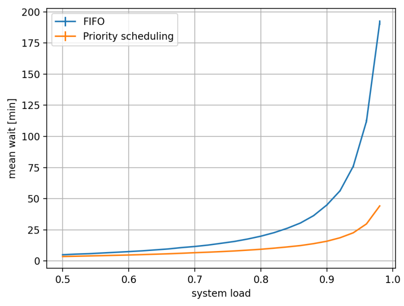


(a) Average wait time for varying system load

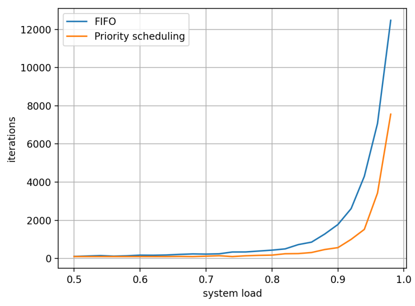


(b) Necessary Iterations for varying system load

Figure 3: M/M/n with FIFO Scheduling



(a) Average wait time for varying system load



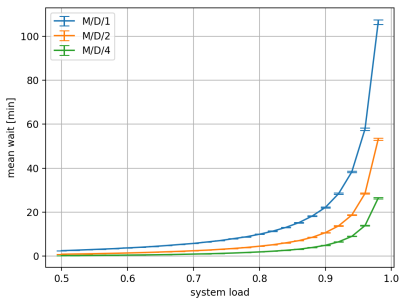
(b) Necessary Iterations for varying system load

Figure 4: M/M/n with Shortest Order First Scheduling

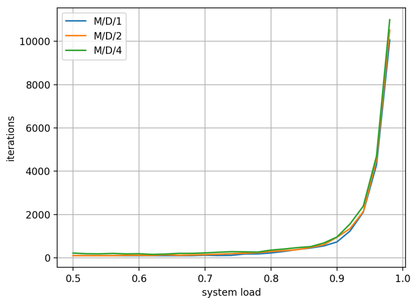
3.4 M/D/n Queues with FIFO Scheduling

The behaviour of a queuing system with a deterministic service time distribution is shown in fig. 5a. As expected, using a high number of customers and a high number of iterations, the behaviour is very similar to an M/M/n system (see table 3 in appendix A). However the difference lies in the number of iterations required to obtain this result. In fact in this case, the required number of simulation was ten times smaller than the other systems. Using the Welch test, the hypothesis that the values from different M/D/n systems come from the same distribution could also be rejected. The same is true for the comparison between the M/D/1 and the M/M/1 queues (see table 8, table 9 and table 12 in appendix B). Similarly as for M/M/n queues, using the T-test the hypothesis that the expected values and the simulated values are from the same distributions can be accepted when the system load is relatively small. As it approaches one this is not necessarily the case as can be seen in the fact that for $\rho = 0.98$

the hypothesis had to be rejected (see table 3 in appendix A).



(a) Average wait time for varying system load

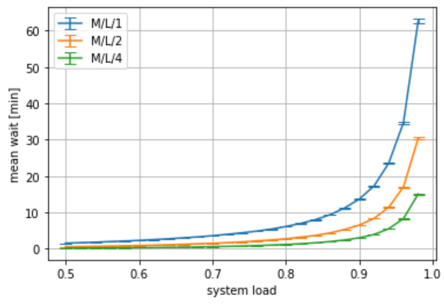


(b) Necessary Iterations for varying system load

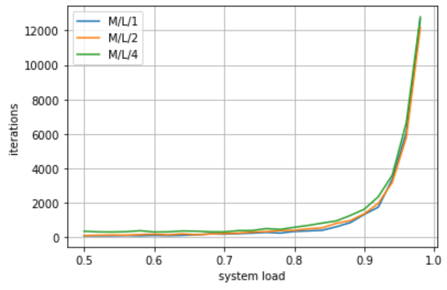
Figure 5: M/D/n with FIFO Scheduling

3.5 M/L/n Queues with FIFO Scheduling

The last system studied in this paper is an hyper-exponential distributed service time with parameters $\mu_1 = 1\ min^{-1}$ and $\mu_2 = 0.2\ min^{-1}$. The behaviour of the system is very similar to the previous ones. The Welch test confirmed that with a confidence level of 95%, it can be affirmed that the mean waiting times come from different distributions for different values of n (see table 10, table 9 in appendix B). However, in this case, as it is possible to see from table 4 in appendix B, the hypothesis that the simulated values and the values from the theory come from the same distributions must be rejected. A reason for this can be a higher variance of the service time's distribution.



(a) Average wait time for varying system load



(b) Necessary Iterations for varying system load

Figure 6: M/L/n with FIFO Scheduling

4 Conclusion

As explored in this report, using a discrete event simulations to study queuing theory, can lead to interesting conclusions. From the investigation of a systems steady state, it becomes clear that the simulations start to work if the number of customers is above a certain threshold, as otherwise the system can not be considered in a steady state (i.e. the system needs an initial adjustment period). This threshold however, can vary with the number of servers and with the system load. Moreover, after fixing the number of customers to 10000, to ensure that for most values of ρ the system was in a steady state, it was found that the analytical formulas don't always give the same result as the simulations. For example, when the system load approached to 1, the result of the simulation and the analytical formulas differ even if the number of customers $k = 10000$. However, due to the high computational cost of the simulation for such a high value of ρ, the number of the customers was not adjusted. In the second part of the report, the goal was to compare different queuing systems. It can be concluded for all systems that the differences in outcomes when tested with different numbers of serves was not due to stochasticity but due to a difference in the underlying distribution. Furthermore, the simulated mean waiting times for both M/M/n and M/D/1 systems are consistent with their analytical counterpart. However it was found that this is not true for values of ρ close to one as the system behaves in a more stochastic manner.

The same analysis was preformed for M/L/1 systems, however even for low values of ρ it could not be proven that the simulated and expected values stem from the same distribution. This is possibly due to the fact the hyper-exponential distribution has a significantly higher variance compared to a normal exponential distribution which is reflected in a broader range of values for the mean waiting time.

Additionally while M/M/n and M/D/n systems showed very similar behaviour for equal system parameters, the number of iterations necessary to reach the same level of confidence was significantly lower for M/D/n systems.

Moreover it was shown that using *shorter job first scheduling* in an M/M/1 systems leads to a shorter mean waiting time than using FIFO ordering for the same system.

A proposition for future research would be to investigate how the required number of customers to be in a steady state changes when ρ approaches 1, in order to perform the simulations with a more suitable value of k. Moreover, especially for the M/L/n queues, it will be interesting to apply a finer parameters tuning before the simulation, in order to obtain results comparable with the theoretical ones.

A Appendix A

	M/M/1				**M/M/2**				**M/M/4**			
System Load	**E[W]**	$\bar{W} \pm \sigma$	**n**	**Hypothesis**	**E[W]**	$\bar{W} \pm \sigma$	**n**	**Hypothesis**	**E[W]**	$\bar{W} \pm \sigma$	**n**	**Hypothesis**
0.5	5.000	5.020 ± 0.050	122	accept	1.667	1.675 ± 0.017	172	accept	0.435	0.430 ± 0.004	430	reject
0.74	14.231	14.154 ± 0.141	275	accept	6.052	6.082± 0.061	424	accept	2.369	2.351 ± 0.023	561	accept
0.98	245.000	192.759 ± 1.928	12477	reject	121.263	94.837 ± 0.948	13048	reject	59.750	46.603 ± 0.466	13616	reject

Table 1: Results for M/M/n Queues

	M/M/1 priority
System load	$\bar{W} \pm \sigma,\ n$
$\rho = 0.5$	$3.545 \pm 0.027,\ n = 99$
$\rho = 0.74$	$7.462 \pm 0.074,\ n = 136$
$\rho = 0.98$	$43.751 \pm 0.437,\ n = 6982$

Table 2: Results for M/M/1 Queue with priority scheduling

	M/D/1				**M/D/2**		**M/D/4**	
System Load	**E[W]**	$\bar{W} \pm \sigma$	**n**	**Hypothesis**	$\bar{W} \pm \sigma$	**n**	$\bar{W} \pm \sigma$	**n**
0.5	2.5	2.492 ± 0.018	99	accept	0.890 ± 0.009	115	0.248 ± 0.002	207
0.74	7.115	7.111 ± 0.071	166	accept	3.103 ± 0.031	187	1.235 ± 0.012	250
0.98	122.499	107.629 ± 1.076	10808	reject	53.080 ± 0.531	10603	26.090 ± 0.261	10466

Table 3: Results for M/D/n Queues

	M/L/1				**M/L/2**		**M/L/4**	
System Load	**E[W]**	$\bar{W} \pm \sigma$	**n**	**Hypothesis**	$\bar{W} \pm \sigma$	**n**	$\bar{W} \pm \sigma$	**n**
0.5	3.499	1.532 ± 0.014	99	reject	0.519 ± 0.005	139	0.139 ± 0.001	383
0.74	9.961	4.389 ± 0.044	220	reject	1.865 ± 0.019	320	0.740 ± 0.007	443
0.98	171.499	61.667 ± 0.617	11961	reject	30.265 ± 0.303	12098	14.990 ± 0.150	12687

Table 4: Results for M/L/n Queues

B Appendix B

System Load	**Degree of Freedom**	**Actual Value**	**Critical Value**	**Hypothesis**
0.5	148	123.89	1.96	Reject
0.74	376	102.69	1.96	Reject
0.98	18225	89.34	1.96	Reject

Table 5: Hypothesis Testing for M/M/1 vs M/M/2

System Load	Degree of Freedom	Actual Value	Critical Value	Hypothesis
0.5	193	141.06	1.96	Reject
0.74	549	112.15	1.96	Reject
0.98	19044	89.47	1.96	Reject

Table 6: Hypothesis Testing for M/M/2 vs M/M/4

System Load	Degree of Freedom	Actual Value	Critical Value	Hypothesis
0.5	207	47.02	1.96	Reject
0.74	386	81.98	1.96	Reject
0.98	13729	147.76	1.96	Reject

Table 7: Hypothesis Testing for M/M/1 with FIFO Scheduling and Shortest Task First Scheduling

System Load	Degree of Freedom	Actual Value	Critical Value	Hypothesis
0.5	122	118.71	1.96	Reject
0.74	226	101.26	1.96	Reject
0.98	15753	89.09	1.96	Reject

Table 8: Hypothesis Testing for M/D/1 vs M/D/2

System Load	Degree of Freedom	Actual Value	Critical Value	Hypothesis
0.5	131	136.29	1.96	Reject
0.74	244	109.63	1.96	Reject
0.98	15431	89.44	1.96	Reject

Table 9: Hypothesis Testing for M/D/2 vs M/D/4

System Load	Degree of Freedom	Actual Value	Critical Value	Hypothesis
0.5	120	122.80	1.96	Reject
0.74	298	103.76	1.96	Reject
0.98	17416	89.60	1.96	Reject

Table 10: Hypothesis Testing for M/L/1 vs M/L/2

System Load	Degree of Freedom	Actual Value	Critical Value	Hypothesis
0.5	158	138.39	1.96	Reject
0.74	419	109.83	1.96	Reject
0.98	17742	88.64	1.96	Reject

Table 11: Hypothesis Testing for M/L/2 vs M/L/4

C Appendix C

System Load	**Degree of Freedom**	**Actual Value**	**Critical Value**	**Hypothesis**
0.5	174	88.4	1.96	Reject
0.74	388	87.15	1.96	Reject
0.98	19301	75.58	1.96	Reject

Table 12: Hypothesis Testing for M/M/1 vs M/D/1

References

[1] A. Willig, *A Short Introduction to Queueing Theory.* Technical University Berlin, Telecommunication Networks Group, 1999.

[2] S. M. Ross, *Simulation.* Knovel Library., Amsterdam: Academic Press, fifth edition. ed., 2013.

[3] A. O. Allen, *Probability, Statistics, and Queueing Theory – With Computer Science Applications.* Amsterdam: Computer Science and Applied Mathematics. Academic Press, 1978.

[4] W. Feller, *An Introduction to Probability Theory and Its Applications - Volume I.* New York: John Wiley, third edition ed., 1968.

[5] I. Adan and J. Resing, *Queueing Theory.* Department of Mathematics and Computing Science Eindhoven University of Technology.

[6] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, "Sympy: symbolic computing in python," *PeerJ Computer Science*, vol. 3, p. e103, Jan. 2017.