UNIVERSITY OF AMSTERDAM

STOCHASTIC SIMULATION ASSIGNMENT 2

Nobody Likes to Wait: An Analysis of Queueing System Waiting Times

December 6, 2021

Students:
Steven Oud (steven.oud2@student.uva.nl)
13688650

Malou Bastiaanse
(malou.bastiaanse2@student.uva.nl)
11426934

Lecturer:
Gábor Závodszky

Course:
Stochastic Simulation

Course code:
5284STSI6Y

Abstract

The mathematical study of queues, so-called queueing theory, dates back to the early 20th century, yet is still relevant today with application in transportation systems, communication networks, and most recently, as a tool for resource management of intensive care beds and ventilators amidst a global pandemic. This study aimed to provide a brief overview of some of the important concepts within queueing theory. Firstly, we provided both theoretical and experimental proof that waiting times are reduced for multi-server systems ($M/M/n$) compared to single-server systems ($M/M/1$). Secondly, it was shown that the service disciplines of SPTF scheduling in comparison with FIFO scheduling showed significantly lower mean waiting time with less variation for $M/M/1$ systems. Lastly, we compared different service time distributions for both single and multi-server queues. It was found that the use of a deterministic distribution of the service time lead to the lowest mean waiting times, followed by the Markovian distribution, with the highest mean waiting time when using a hyper-exponential distribution. Although this report has applied a large variety of models and theorems, it is just a mere glimpse into the field of queueing theory. Possible extension upon this work could be the implementation of techniques to reduce the initialization bias, or, experiment with various other inter-arrival and service time distributions.

Contents

1 Introduction

It is common that scientific principles developed within academia are later found to be of use for industrial application, yet interestingly, for the study of queueing theory, this process is reversed with its origins laying at the Telephone Company of Copenhagen. Queueing theory was first introduced by Agner Krarup Erlang, whose work focused on optimizing the telephone operations within the company, such as minimizing the waiting time for a caller to be on hold (Erlang, 1909, 1920). Unbeknownst to the later impact of his work, Erlang mentioned within one of his earliest papers, that "*it is my belief some point or other from this work may be of interest*"(Erlang, 1909). This expectation was well exceeded, as his work laid the foundations for queueing theory, which as of currently has many applications such as within communication networks, machine plants, and transportation systems (Adan and Resing, 2015; Willig, 1999). As of recently, this field of study has gained additional importance during the COVID-19 pandemic, with queueing models being used for resource management of intensive care beds (Meares and Jones, 2020), ventilator capacity (Zimmerman et al., 2021), and even as far as optimization of queue lengths and waiting times within stores to reduce the risk of infection for customers and staff (Perlman and Yechiali, 2020).

From the first application within the telephone company to the latest applications for COVID-19 management, queueing theory has naturally evolved, yet at its core is the study of queues and waiting times within a system, in which a population requests a particular service of limited capacity (Willig, 1999). With these three basic elements (a population, a queue, and a service), one can try to answer questions such as what type of systems may increase or decrease the mean waiting time, customers in the queue, or the total time spend within the system (Willig, 1999). Queueing models can be designed in a variety of ways, such as by changing the distribution of the inter-arrival times and service times, or, by adding more or fewer servers to the system. Another possibility is changing the service discipline, with common choices being shortest processing time first (SPTF) scheduling[1] or first in first out (FIFO) scheduling.

The aim of this study is to provide a brief overview of some of the important concepts within queueing theory, by implementing experiments with a variety of different queueing models. Following this, we set out to answer three research objectives: (1) derive from both theory and experimental data that with the same load characteristics the waiting times are longer for a single-server system compared to a multi-server system, (2) assess to what extent the service discipline of SPTF scheduling impacts the waiting time compared to FIFO scheduling of a single-server system, and lastly, (3) compare the waiting times using different service time distributions, experimenting with exponential, deterministic, and hyper-exponential distributed service times for both single- and multi-server systems. For both the first and the third research objectives experiments will be done with multi-server systems, which in this study will be limited to two and four equal servers within a system.

This report is structured as follows: within the methodology we will first describe background information on queueing theory (Section 2.1), introducing notation and properties, followed by a description of how the queueing systems (Section 2.2) are implemented and the tools used for statistical analysis and visualization (Section 2.3). The results section will first provide plots that highlight the required simulation length for different system loads (Section 3.1), after which the results of the three research objectives are discussed in their respective order (Section 3.2-Section 3.4). Lastly,

[1] This is the same as shortest job first scheduling per the assignment

within the discussion (Section 4), the findings for the research objectives are discussed together with potential future research.

2 Methods

2.1 Basic Model and Notation

A basic queueing model is visualized in Figure 1. Customers arrive into the system starting from the left arrow according to some arrival rate distribution. They get placed into a queue (visualized as a row of rectangles), and get served one at a time by the server (visualized as a circle) according to a service discipline. The service time (the time taken by the server to service the customer) is also sampled from some distribution, which can be different from the distribution for arrivals.

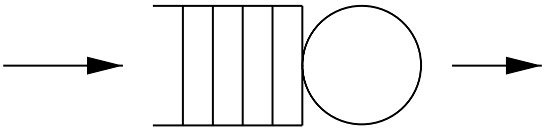


Figure 1: *Basic queueing model with one server. The server is represented by the circle, the queue by a row of rectangles, and the arriving and outgoing customers by the left and right arrows respectively.*

Formally, a queueing model is often described using Kendall notation (Kendall, 1953):

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

where

- A is the inter-arrival time distribution.
- B is the service time distribution.
- m is the number of servers.
- N is the maximum length of the queue (the queue length is often assumed to be infinite, in which case this value is omitted).
- S is the service discipline (optional, defaults to FIFO if omitted).

Throughout this report, we will experiment with different combinations of inter-arrival and service time distributions and also different service disciplines. For time distributions (A and B), the following are common choices (Willig, 1999):

- M (Markov): exponential distribution (memoryless).
- D (Deterministic): deterministic distribution with constant value (no randomness).
- H (Hyper-exponential): sum of multiple exponential distributions with weighted probability.
- G (General): specific distribution is not specified (but mean and variance is known in most cases).

It is usually assumed that the inter-arrival times are independent and identically distributed according to some inter-arrival rate λ, and that the service times are independent and identically distributed according to some service rate μ. For service disciplines, we will mostly use the FIFO discipline where the customers are served in the order they arrived in. In Section 3.3 we will compare the FIFO discipline with the SPTF discipline, where jobs with the shortest processing time are prioritized. Meaning that after a service is completed, the customer with the shortest service time is served next.

We are often interested in how occupied the system is on average. The system load for a $G/G/1$ system is defined as follows:

$$\rho = \lambda E(B), \tag{2}$$

where $E(B)$ is the mean service time. To avoid the queue from blowing up, we require $\rho < 1$. When we introduce n servers ($G/G/n$), the system load is defined as

$$\rho = \frac{\lambda E(B)}{n}. \tag{3}$$

2.1.1 Little's Law

An important theorem within queueing theory is Little's law. Little's law states the relationship between the mean number of customers in the system with the mean sojourn time and arrival rate. This was generally always assumed within queueing theory, yet its first proof was provided by John Little (Little, 1961), after whom Little's Law is named. Following the notations provided by Adan and Resing (2015), Little's law states the following:

$$E(L) = \lambda E(S), \tag{4}$$

where $E(L)$ is the mean number of customers in the system, $E(S)$ the mean sojourn time, and λ is the inter-arrival rate. Although Little's law (Equation 4) seems like a rather trivial statement, it shows that the mean number of customers within the system is independent of many aspects of the queueing system such as the inter-arrival time or service time distribution, service discipline, or service order. Hence, Little's law is quite a generally applicable theorem, as it can be applied to all $G/G/n$-queues and service disciplines (Willig, 1999). The only constraints are that a steady-state should have been reached and that the system's capacity needs to be large enough for the number of customers (Adan and Resing, 2015), meaning the system load ρ should be below 1.

Little's law can also be applied to just components of the system, such as the queue itself:

$$E(L^q) = \lambda E(W), \tag{5}$$

where $E(L^q)$ is the mean queue length and $E(W)$ the meaning waiting time within the queue. Another application of Little's law is on just the server part, for which it can be found that:

$$\rho = \lambda E(B), \tag{6}$$

where ρ is the system load, which can be interpreted as either the fraction of the time the server is working, or the mean number of customers at the server.

2.1.2 PASTA Property

The following theorem starts with the question: if a queueing system is a fraction of the time in some state A, would the fraction of arriving customers finding the system in that same state A be the same? Taking the example of a $D/D/1$-queue (as described by Willig (1999)); if the arrival rate λ is less than the mean service time μ, new customers would always find the system empty, as customers are served quicker than they arrive. Hence, although the system is only half the time empty, the customer will always find an empty system, hence the answer to our question would be no. The reason for this is that the arrival times within this system are not random times (Willig, 1999).

However, if the arrival times are sampled from a Poisson process, and thus come from an exponential distribution, the samples are random enough for the arriving customers on average to find the same state of the queueing system as an outside observer looking at the system at random intervals. This property of $M/\cdot/\cdot$ systems, is called the Poisson Arrivals See Time Averages (PASTA) property.

2.1.3 Queueing Systems

A common and mathematically appealing family of queueing systems are the $M/M/n$ systems. $M/M/n$ systems have n servers, FIFO service discipline, an infinite queue, Markovian (exponentially) distributed inter-arrival rates according to some λ, and Markovian distributed service times according to some μ. As the inter-arrival rates and service times are exponentially distributed, the mean inter-arrival time and mean service time are $E(A) = 1/\lambda$ and $E(B) = 1/\mu$ respectively. For $M/M/n$ systems, the system load is defined as

$$\rho = \frac{\lambda}{n\mu}. \tag{7}$$

By studying the limiting behaviour of $M/M/n$ systems (see Adan and Resing (2015, Chapter 5)), we can derive equilibrium expressions for the mean queue length:

$$E(L^q) = \Pi_W \frac{\rho}{1-\rho}, \tag{8}$$

where Π_W is the probability that a job has to wait (derived by PASTA):

$$\Pi_W = \frac{(n\rho)^n}{n!}\left[(1-\rho)\sum_{k=0}^{k-1}\frac{n\rho^k}{k!} + \frac{(n\rho)^n}{n!}\right]^{-1}. \tag{9}$$

Then, by applying Little's law on the queue of the system (Equation 5), we can calculate the mean waiting time:

$$E(W) = \Pi_W \frac{1}{1-\rho}\frac{1}{n\mu}. \tag{10}$$

This quantity will be used extensively in this report to analyze the performance of different queueing systems.

Queues whose inter-arrival times and service times are Markovian are easy to analyze. When one of these distributions is non-Markovian however, the analysis gets more complicated. In this report, we will consider queueing systems with non-Markovian service time distributions ($M/D/n$ and $M/H/n$ systems). Analytical methods for analyzing $M/G/1$ systems are available (Nelson, 2013), however, this report will focus on analyzing these non-Markovian systems through simulation and statistical analysis.

The analytical methods described so far assume that the queueing system is in its steady state (when the probability distribution of the number of customers in the system is stable). When simulating queues, there is a warming-up period before the queue reaches its steady-state where the probability distribution of the number of customers in the system is unstable.

2.2 Queueing Model Implementation

2.2.1 Software

The implementation of the queueing models within the experiments is done by making use of the `simpy` library (version 4.0.1) ("SimPy", 2020). The `simpy` library can be used for process-based discrete-event simulation within the Python programming language. This framework makes use of generator functions to define processes, which allows the user to model active components such as vehicles, agents, or customers. Within this study the aim is to simulate customers in a queue, which upon arrival will request access to the counter and depending on the number of customers within the system, will have to wait in the queue or not. This counter is represented with the `simpy.Resource` class or `simpy.PriorityResource` class, depending on the service discipline used. These classes are used to represent a resource that can be used by a limited number of processes (e.g. customers) at the time. If the resource is fully in use, a process (e.g. a customer) will be queued up. As this study implements a large variety of queueing models, the implementation of each will be described in order of research objective:

- *Research objective 1*: The queueing models to be implemented for the first research objective are the $M/M/1$-*FIFO* queue and $M/M/n$-*FIFO* queue, in which this latter multi-server system both two and four equal servers will be considered. For the FIFO service discipline the `simpy.Resource` class is used, where the *capacity* parameter can be set to the number of servers within the system (e.g. 1, 2, or 4). Both the inter-arrival times and service times are exponentially distributed, hence will be sampled using the `random.exponential` function of the `numpy` library. The parameter *scale* will be given the values of $1/\lambda$ or $1/\mu$ based on the chosen inter-arrival rate (λ) and service rate (μ).

- *Research objective 2*: The queueing models to be implemented for the second research objective are the $M/M/1$-$FIFO$ queue and $M/M/1$-$SPTF$ queue. The former queue is implemented the same as for research objective 1, however, the latter queue requires assigning priorities to customers waiting in the queue, where priority is given to customers with the shortest processing time. The `simpy.PriorityResource` class allows to give a priority to a customer using the parameter *priority*, where pending requests in the queue are sorted by priority, with lower values representing higher priority. As this value can be float, the sampled service time B will thus be input for the *priority* parameter.
- *Research objective 3*: The queueing models to be implemented for this objective are the $M/D/1$, $M/D/n$, $M/H/1$ and $M/H/n$ queues, all with the FIFO service discipline. The models are implemented as before, except for the newly introduced deterministic and hyper-exponential distributions for sampling the service times. For the deterministic distribution, the mean service time is simply $1/\mu$. For the hyper-exponential distribution, 75% of the jobs will have the mean service time of $1/\mu$, and the remaining 25% of the jobs will have a service time of $5/\mu$. To sample from the exponential distribution, again the `random.exponential` function will be used, scaled with the appropriate μ values. To assign which customer gets which distribution, random samples are drawn using the `random.uniform` function of `numpy`, with values lower than 0.75 leading to the regular exponential distribution.

For random number generation, we use the Mersenne Twister algorithm as given by Matsumoto and Nishimura (1998). Because we run our experiments in parallel, we use multiple instances of Mersenne Twister pseudo-random number generators. The state of each subsequent instance is advanced as if 2^{128} samples have been drawn as described by Haramoto et al. (2008) to ensure independent random numbers.

2.2.2 Notation and Parameter Choices

Using the previously described queueing models, every time a customer leaves the queue a waiting time is recorded and stored. From these values, a mean waiting time of the queue can be calculated for each run, which will be referred to as $\overline{W}$. This should not be confused with the theoretically derived mean waiting time, which for clarity will be referred to as $E(W)$. As can be seen from the previous section, the queueing models within this study all rely to a certain extent on random sampling, and thus are inherently stochastic, hence each experiment requires multiple runs to generate statistically meaningful results. To refer to the sample mean of the $\overline{W}$ values of the collective runs, the notation of $\langle\overline{W}\rangle$ is used.

When comparing between queueing models, it is of importance the system load ρ is the same for each model to ensure valid comparison. For the system to reach a steady-state, it is essential for ρ to be below one, or else the queue will grow infinitely ($t \to \infty$). The exact value ρ that will be used within the experiments, is discussed within the results (section 3.2.2), after analyzing the impact of varying values of ρ on the mean waiting time ($\langle\overline{W}\rangle$).

Both within research objective 1 and 3 queueing models are compared with different amounts of servers. As ρ is a function of the arrival rate, service rate, and number of servers, with $\rho = \lambda/(n\mu)$, the arrival rate will be scaled to ensure equal system load between the models. This is done by multiplying the arrival rate by the number of servers (n), hence $\lambda^* = \lambda n$, where λ^* is the adjusted arrival rate. Another complication is regarding research objective 3, where service times are sampled from two different exponential distributions ($\exp(\lambda)$ and $\exp(\lambda/5)$) for the $M/H/n$ queues with a 75:25 ratio. Fixing a value of μ for the other models, the adjusted service rate μ^* can be found by solving $\mu = 0.75\mu^* + 0.25(\mu^*/5)$, finding that $\mu = 0.8\mu^*$.

A single simulation is run for a fixed amount of time, which is done by setting the parameter *until* within the initialized environment (`simpy.Environment`) to a fixed value. The simulation time is an important parameter, as for too short of a simulation the steady-state might not be reached, but longer simulations are computationally intensive. The choice for this parameter was based on for which simulation time, the sample mean waiting time ($\langle\overline{W}\rangle$) would converge towards the theoretically derived mean waiting time ($E(W)$), with a relatively small confidence interval. As this is such an important parameter, an additional section is added to the results to analyse this behaviour (Section 3.1).

2.2.3 Numerically Stable Computation of $E(W)$ for $M/M/n$ Systems

To verify our simulations, we would like to numerically calculate $E(W)$ for comparisons when possible. As discussed in Section 2.1.3, the definition of $E(W)$ for $M/M/n$ systems contains a term called the delay probability:

$$\Pi_W = \frac{(n\rho)^n/n!}{(1-\rho)\sum_{k=0}^{k-1} n\rho^k/k! + (n\rho)^n/n!}. \tag{11}$$

Due to the $n!$ term in $(n\rho)^n/n!$, computing Π_W in this form can lead to numerical problems. We can use a relation between the delay probability Π_W and the blocking probability $B(n,\rho)$. The blocking probability is used in $M/G/n/n$ systems to denote the probability that a customer gets rejected (when the queue is full), and is defined as follows:

$$B(n,\rho) = \frac{\rho^n/n!}{\sum_{k=0}^{n} \rho^k/k!}. \tag{12}$$

A numerically stable recursion for $B(n,\rho)$ can be derived by rewriting Equation 12 as

$$B(n,\rho) = \frac{\rho^n/n!}{\sum_{k=0}^{n-1} \rho^k/k! + \rho^n/n!} \tag{13}$$

and dividing the numerator and denominator by $\sum_{k=0}^{n-1} \rho^k/k!$:

$$\begin{aligned} B(n,\rho) &= \frac{\rho B(n-1,\rho)/n}{1+\rho B(n-1,\rho)/n} \qquad (14) \\ &= \frac{\rho B(n-1,\rho)}{n+\rho B(n-1,\rho)}. \qquad (15) \end{aligned}$$

Then, we add a stopping criterion to give the following recursive function for calculating $B(n,\rho)$:

$$B(n,\rho) = \begin{cases} 1, & \text{if } n = 0 \\ \frac{\rho B(n-1,\rho)}{n+\rho B(n-1,\rho)}, & \text{otherwise.} \end{cases} \tag{16}$$

By the definition of Π_W (Equation 11), we have

$$\Pi_W = \frac{\rho B(n-1,n\rho)}{1-\rho+\rho B(n-1,n\rho)}. \tag{17}$$

We can thus use Equation 16 to compute Π_W, and in term compute $E(W)$. This recursive function is implemented and used to verify our results.

2.3 Analysis Experimental Results

2.3.1 Statistical Methods

As the implemented queueing models all rely on random sampling to a certain extent, multiple simulations will be run to generate results that are statistically meaningful. As mentioned under section section 2.2.2, each individual simulation has a calculated mean waiting time ($\overline{W}$). The collective of mean waiting times for all simulations can be described by its sample mean ($\langle\overline{W}\rangle$) and sample variances ($S^2$) (Ross, 2013). These statistics were calculated as follows:

$$\overline{W} = \sum_{i=1}^{m} \frac{W_i}{m}, \quad \langle\overline{W}\rangle = \sum_{i=1}^{n} \frac{\overline{W}_i}{n}, \quad \text{and} \quad S^2 = \frac{\sum_{i=1}^{n}\left(\overline{W}_i - \langle\overline{W}\rangle\right)^2}{n-1}, \tag{18}$$

where m is the number of recorded waiting times and n is the number of simulations. To calculate the sample mean and variance we will be making using of the `numpy` library, with the `mean` function to calculate the sample mean and the `var` to calculate the sample variance with parameter *ddof = 1* to to ensure an unbiased estimator of the variance. Within each experiment the sample mean is provided with a confidence interval with a confidence level of $p = 95\%$. This is done using the `stats.t.interval` function from `scipy` with parameter values: *df = n - 1*, *loc* = $\overline{X}$, and *scale* = $\sqrt{S^2}/\sqrt{n}$ (calculated with the `scipy.stats.sem` function).

The different queueing models will be compared in terms of the sample mean and sample variance of the collective mean waiting times. To accept or reject the hypothesis of equal sample variances between two populations ($S_x^2 = S_y^2$), an F-test can be performed (Heumann et al., 2017) for which we use the `stats.f.cdf` function of the `scipy` library. With parameter values: $x = \max(S_x^2/S_y^2, S_y^2/S_x^2)$, *dfn* the sample size of the population with the largest variance - 1, and *dfd* the sample size of the population with the smallest variance - 1.

To accept or reject the hypothesis of equal sample means between two population ($\langle\overline{W}\rangle_x = \langle\overline{W}\rangle_y$), the `stats.ttest_ind` function of the `scipy` library is used. This function can both perform a T-test and Welch-test (Heumann et al., 2017), by setting the *equal_var* to *True* or *False* respectively. The decision for this will be based on the results of the F-test.

The resulting p-values of each test will scaled using a Bonferroni correction (Bell, 2018; Bonferroni, 1936) depending how often the same test is performed. This is done using the `stats.multitest.multipletests` function of the `statsmodel` library. The acceptance threshold for each test is set to $\alpha = 0.05$, meaning the hypotheses will be rejected when $p < \alpha$.

2.3.2 Visualization results

All basic visualization is done using the `matplotlib` library. Boxplots are generated using the `boxplot` function of the `seaborn` library (Waskom, 2021). For basic histograms the `histplot` functions of the `seaborn` library (Waskom, 2021) is used. To these histograms a line of the kernel density estimate is added, by setting *kde = True*. In the case where many different data distribution are compared, a ridge line plot was used. This was fone with the `joyplot` function of that `joypy` library.

3 Results

3.1 Time Parameter Choice

In order to analyze the steady-state of queueing systems, we investigate the impact of the number of measurements taken during a simulation. We do this by looking at the impact of various simulation lengths on the confidence interval. Figure 2 shows the results of simulations of the $M/M/1$ system with varying simulation length and system loads. We see that for ρ closer to 1, more time (and thus number of measurements) is required to achieve the same confidence level as lower values of ρ. Systems with higher loads take longer to reach their steady-state, leading to high confidence radii at lower simulation lengths.

For the experiments conducted in Sections 3.3 and 3.4 we pick a fixed system load ρ and simulation length t based on Figure 2. For system load, we choose $\rho = 0.9$, as it is large enough to demonstrate interesting behaviour. Balancing computation time and accuracy, we choose to fix $t = 10^5$ for our experiments, which has a confidence radius of approximately 0.318 with $p = 95\%$ and approximates $E(W)$ reasonably well for $\rho = 0.9$ ($|\langle\overline{W}\rangle - E(W)| \approx 0.115$).

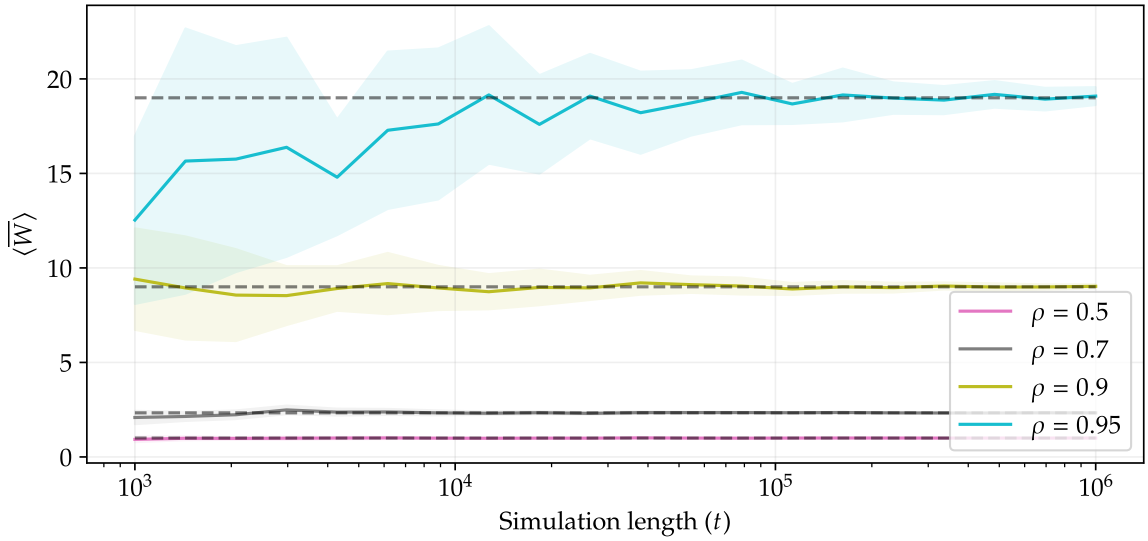


***Figure 2:** Sample mean waiting times of the $M/M/1$ system for varying time steps between 10^3 and 10^6 for different ρ. The mean waiting time of 50 runs is shown with its confidence interval ($p = 95\%$). Theoretical $E(W)$ for every ρ is shown as a dashed line.*

3.2 Comparing Single- and Multi-Server Systems

In this section, the average waiting times for $M/M/1$ systems and $M/M/2$ systems are compared. We will show that for the same system load ρ, the average waiting time of a $M/M/2$ system is shorter than for a $M/M/1$ system. To ensure fair comparison and equal load characteristics (equal ρ), the $M/M/2$ system is given a 2-fold higher arrival rate compared to the $M/M/1$ system:

$$\lambda_2 = 2\lambda_1, \tag{19}$$

where λ_1 is the arrival rate for the $M/M/1$ system, and λ_2 the arrival rate for the $M/M/2$ system.

3.2.1 Theoretical Result

We first compare the average waiting times of the $M/M/1$ and $M/M/2$ systems analytically. The average waiting time for a $M/M/n$ system is given in Equation 10, which depends on the delay probability (Equation 9). We first consider the $M/M/1$ system ($n = 1$). The delay probability for $M/M/1$ is

$$\Pi_{W_1} = \rho\left[(1-\rho)+\rho\right]^{-1} = \rho. \tag{20}$$

Then,

$$E(W_1) = \rho\frac{1}{1-\rho}\frac{1}{\mu}. \tag{21}$$

For the $M/M/2$ system, the delay probability is

$$\begin{aligned}\Pi_{W_2} &= \frac{(2\rho)^2}{2}\left[(1-\rho)(1+2\rho)+\frac{(2\rho)^2}{2}\right]^{-1} &(22)\\ &= 2\rho^2\left[(1-\rho)(1+2\rho)+2\rho^2\right]^{-1} &(23)\\ &= 2\rho^2\frac{1}{1+\rho}, &(24)\end{aligned}$$

with average waiting time

$$\begin{aligned} E(W_2) &= 2\rho^2 \frac{1}{1+\rho} \frac{1}{1-\rho} \frac{1}{2\mu} && (25) \\ &= 2\rho^2 \frac{1}{(1-\rho^2)2\mu} && (26) \\ &= \rho^2 \frac{1}{1-\rho^2} \frac{1}{\mu}. && (27) \end{aligned}$$

Given that $0 < \rho < 1$, $\rho^2 < \rho$ and $1/(1-\rho^2) < 1/(1-\rho)$. Then, by Equation 21 and Equation 27:

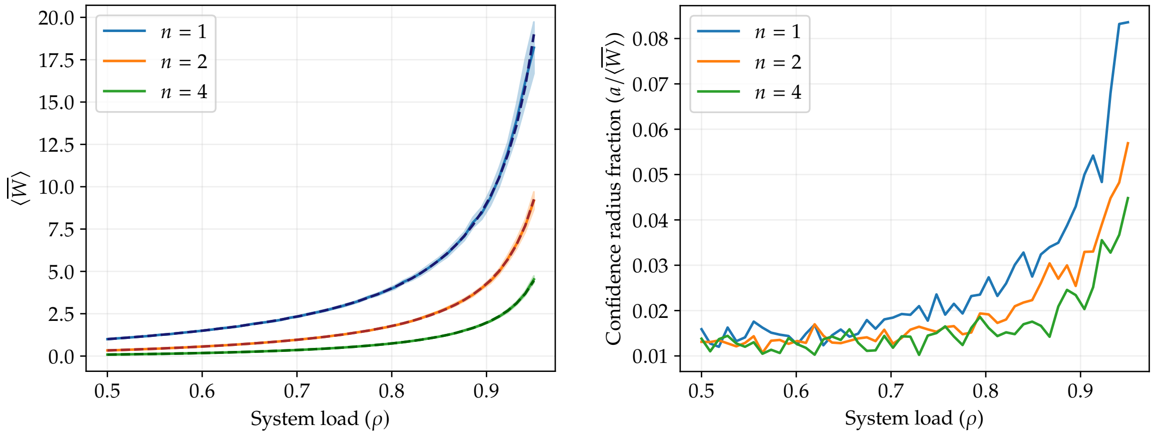
$$\begin{aligned} \rho^2 \frac{1}{1-\rho^2} \frac{1}{\mu} &< \rho \frac{1}{1-\rho} \frac{1}{\mu} && (28) \\ \iff E(W_2) &< E(W_1). && (29) \end{aligned}$$

This makes sense intuitively. First, consider what happens in a $M/M/1$ queue when a customer with a high service time occurs, relative to the other customers in the queue. This high service time customer, which will be referred to as Karen, blocks the system for all customers in the queue behind her. The waiting time for all customers behind her is then at least as long as the service time of Karen. With a $M/M/2$ system, a customer such as Karen would not block the whole system, but just one server. Customers can continue to be served by the other server while Karen is handled. Of course, the same proportion of Karens will occur in the $M/M/1$ and $M/M/2$ system if the system load is kept equal, but the probability of two Karens appearing at the same time in the $M/M/2$ system and blocking the whole system is less likely. Thus, the mean waiting time for $M/M/2$ systems is overall less than the mean waiting time for $M/M/1$ systems. This same argument holds for $M/M/n$ systems.

3.2.2 Experimental Result

To verify the results from Section 3.2.1, we write a discrete event simulation program to simulate and compare $M/M/1$, $M/M/2$, and $M/M/4$ systems. Figure 3a shows the results of different system loads each system against both the theoretically derived mean waiting time $E(W)$ and the experimentally found sample mean waiting time $\langle \overline{W} \rangle$. As explained in the previous section, the arrival rates are scaled to ensure ρ is equal ($\lambda_1 = 0.9$, $\lambda_2 = 2\lambda_1$, $\lambda_4 = 4\lambda_1$). From Figure 3a we can verify our theoretical results: for any ρ, $E(W_2) < E(W_1)$. Furthermore, we see $E(W_4) < E(W_2) < E(W_1)$. When looking at the experimentally derived $\langle \overline{W} \rangle$, the results closely resemble the theoretically derived $E(W)$ values, and thus have the same patterns in regards to the three different systems, with more servers leading to lower mean waiting times.

Besides this clear difference in mean waiting time between different n, we can also see that the confidence interval increases as ρ gets close to 1. Thus, more measurements will be required to achieve statistical significance for experiments with ρ close to 1. This is further visualized in Figure 3b, where the confidence radius as a fraction of $\langle \overline{W} \rangle$ is plotted as ρ increases. For all n, the confidence radius fraction increases as ρ approaches 1. Based on the results in Figure 3 and Figure 2, a ρ value of 0.9 was chosen to be the fixed value for all further experiments. As can be seen from these two figures, ρ values close to 1 require a sizable simulation time to reach steady-state, thus resulting in relatively large confidence intervals. As longer simulations come with a certain computational intensity, the choice of the fixed ρ value was based on balancing computational intensity, yet, keeping ρ relatively close to 1 to create interesting results.

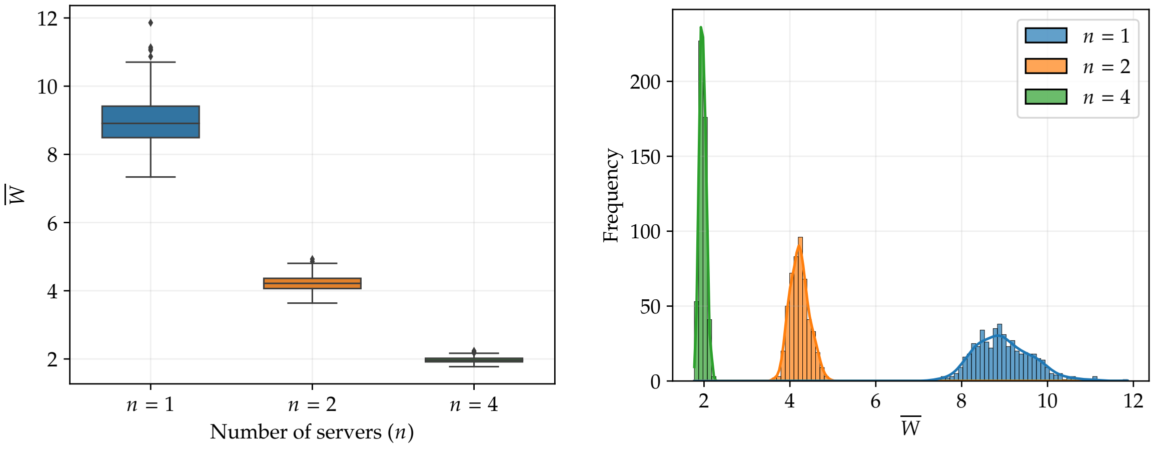


***(a)** Varying ρ on mean sample waiting time.*

***(b)** Varying ρ on confidence radius fraction.*

***Figure 3:** For $M/M/n$ systems with $n \in \{1, 2, 4\}$ with varying ρ the sample mean waiting times ($\langle\overline{W}\rangle$) is shown (a) together with the confidence radius as a fraction of $\langle\overline{W}\rangle$ (b). The parameter μ was kept at a value of 1, with λ having adjusted values of $\lambda_1 = 0.9$, $\lambda_2 = 2\lambda_1$ and $\lambda_4 = 4\lambda_1$. The confidence interval and radius have a confidence level of $p = 95\%$. 50 simulations were run for 10^5 time steps. Besides the experimentally derived results, the theoretical derived mean waiting time $E(W)$ is shown as a dashed line for every system in figure a.*

Figure 4 provides a closer looks on the data distributions of the $M/M/1$, $M/M/2$, and $M/M/4$ systems for a fixed value of ρ of 0.9, shown using both boxplots (fig. 4a) and histograms (fig. 4b). As can be seen from Figure 4a, outliers can be observed within the data distributions for all systems. These outliers are mostly present for higher values of $\overline{W}$. The difference between the minimum and maximum values (and to the same extent the first and third quartile) becomes increasingly smaller with larger values of n. Furthermore, the median values of $\overline{W}$ decreases also for larger values of n.



***(a)** Box plot of mean waiting times.*

***(b)** Histogram of mean waiting times.*

***Figure 4:** Data distributions of mean waiting times in $M/M/n$ systems with $n \in \{1, 2, 4\}$ for $\rho = 0.9$. Shown data is collected from 500 runs with 10^5 time steps. The parameter μ was fixed at a value of 1, with λ having adjusted values of $\lambda_1 = 0.9$, $\lambda_2 = 2\lambda_1$ and $\lambda_4 = 4\lambda_1$.*

These same patterns are reflected within the histograms of the data distributions in Figure 4b. Overall, more servers in a system lead to lower mean values of $\overline{W}$, with more data points centered around the mean and thus less variation. Although, most prominently for $M/M/1$, the distributions are slightly skewed to the right, they generally resemble a normal distribution.

The results of Figure 4 are backed up with Table 1 showing the in-depth statistics of the experiments. The sample mean waiting time with added confidence interval ($p = 95\%$) are: $8.976 \pm 1.17 \cdot 10^{-1}$ for $M/M/1$, $4.2270 \pm 3.93 \cdot 10^{-2}$ for $M/M/2$, and, $1.9700 \pm 1.38 \cdot 10^{-2}$ for $M/M/4$. The sample variance was found to be: $4.40 \cdot 10^{-1}$ for $M/M/1$, $5.00 \cdot 10^{-2}$ for $M/M/2$, and, $6.14 \cdot 10^{-3}$ for $M/M/4$. Between all combinations of systems, both the sample mean and variance were shown to be significantly different, based on the results of F-tests and Welch-tests ($\alpha = 0.05$). As these two types of tests were performed a total of 13 times, which includes the tests performed for the next sections, all p-values were adjusted using a Bonferroni correction. Additionally, Table 1 shows the theoretically derived $E(W)$ values for each system. For the $M/M/4$ system, the $E(W)$ value is exactly the same as the approximated $\langle\overline{W}\rangle$ value. Although the $E(W)$ values for the $M/M/1$ and $M/M/2$ systems are slightly higher, they do lie within the confidence interval of $\langle\overline{W}\rangle$.

Statistics				
	$\langle\overline{W}\rangle$	$E(W)$	Confidence radius ($p = 95\%$)	S^2
$n = 1$	8.976	9.00	$1.17 \cdot 10^{-1}$	$4.40 \cdot 10^{-1}$
$n = 2$	4.2270	4.26	$3.93 \cdot 10^{-2}$	$5.00 \cdot 10^{-2}$
$n = 4$	1.9700	1.97	$1.38 \cdot 10^{-2}$	$6.14 \cdot 10^{-3}$

F-Test			
	$n = 1$	$n = 2$	$n = 4$
$n = 1$	x	$\ll 0.001$	$\ll 0.001$
$n = 2$		x	$\ll 0.001$
$n = 4$			x

Welch's Test			
	$n = 1$	$n = 2$	$n = 4$
$n = 1$	x	$\ll 0.001$	$\ll 0.001$
$n = 2$		x	$\ll 0.001$
$n = 4$			x

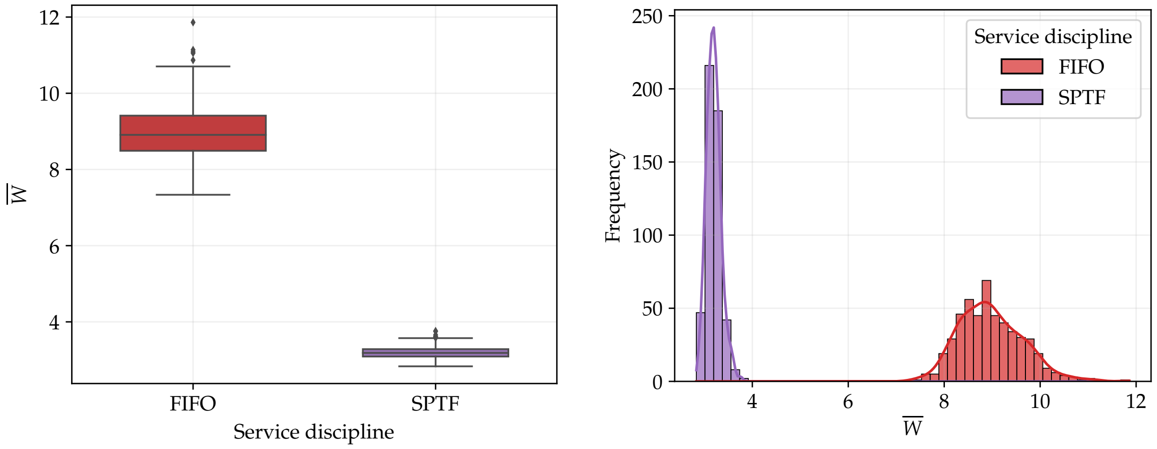
Table 1: *Statistical analysis for $M/M/n$ systems ($\rho = 0.9$) with $n \in \{1, 2, 4\}$. Data is based on 500 repeated runs with 10^5 time steps.*

3.3 Comparison Service Discipline (FIFO vs SPTF)

Next, we will compare the $M/M/1$ system with the $M/M/1$-$SPTF$ system with ρ at 0.9, to show the impact of the two different service disciplines. Figure 5 shows the data distribution of these systems, using both boxplots (fig. 5a) and histograms (fig. 5b). From Figure 5a it can be observed that both systems have outliers, in all case for high values of $\overline{W}$. The $M/M/1$-$SPTF$ system has smaller differences between its minimum and maximum (and to the same extent its first and third quartile) compared to the $M/M/1$ system. Furthermore, the $M/M/1$-$SPTF$ system has a lower median value compared to the $M/M/1$ system.

Similar results can be observed within the histograms of the data distribution as shown in Figure 5b. The $M/M/1$-$SPTF$ has a smaller mean value of $\overline{W}$, with more data points centered around the mean and thus less variation. Similarly to as seen before in Figure 4b, the distributions are slightly skewed to the right, but generally resemble a normal distribution.

The statistics of the data distributions within Figure 5b are shown within Table 2. The sample mean waiting time with added confidence interval ($p = 95\%$) are: $8.976 \pm 1.17 \cdot 10^{-1}$ for $M/M/1$, and, $3.1957 \pm 2.63 \cdot 10^{-2}$ for $M/M/1$-$SPTF$. The sample variances were: $4.40 \cdot 10^{-1}$ for $M/M/1$, and, $2.24 \cdot 10^{-2}$ for $M/M/1$-$SPTF$. The differences within sample mean and sample variance between both service disciplines were shown to be statistically significant using F-tests and Welch-tests ($\alpha = 0.05$). Again, these p-values were adjusted using a Bonferroni correction.



***(a)** Box plot of mean waiting times.*

***(b)** Histogram of mean waiting times.*

***Figure 5:** Data distributions of mean waiting times shown for both the $M/M/1$-$FIFO$ system and $M/M/1$-$SPTF$ system for $\rho = 0.9$. Shown data is collected from 500 runs with 10^5 time steps. The parameter μ was fixed at a value of 1 and λ at 0.9.*

	Statistics		
	$\langle\overline{W}\rangle$	Confidence radius ($p = 95\%$)	S^2
FIFO	8.976	$1.17 \cdot 10^{-1}$	$4.40 \cdot 10^{-1}$
SPTF	3.1957	$2.63 \cdot 10^{-2}$	$2.24 \cdot 10^{-2}$

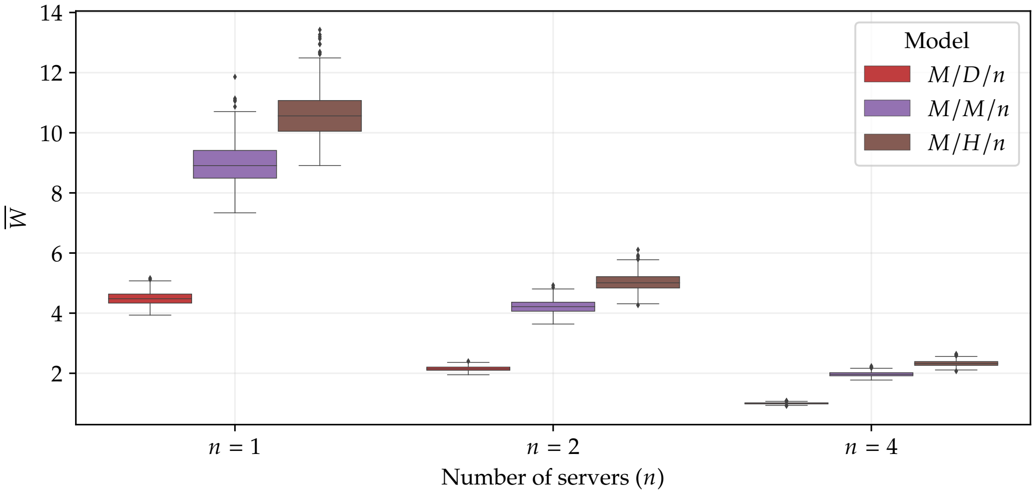
	F-test	*Welch's Test*
FIFO vs SPTF	$\ll 0.001$	$\ll 0.001$

***Table 2:** Statistical analysis for the $M/M/1$-$FIFO$ system and $M/M/1$-$SPTF$ system for $\rho = 0.9$, $\lambda = 0.9$ and $\mu = 1$. Data is based on 500 repeated runs with 10^5 time steps*

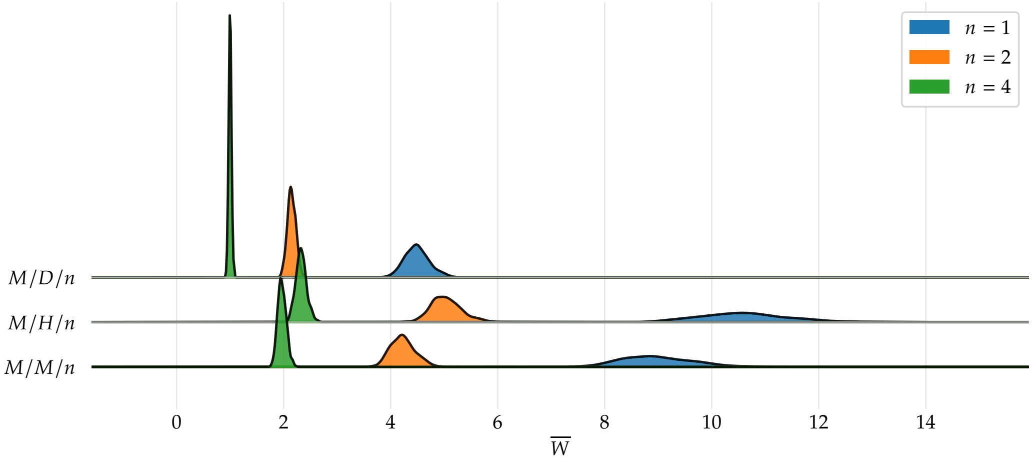
3.4 Varying Service Time Distributions

Finally, we consider non-Markovian service time distributions. We compare $M/M/n$, $M/H/n$, and $M/D/n$ systems with $n \in \{1, 2, 4\}$. The service time distribution for the $M/H/n$ system is weighted as described in Section 2.2.2: we sample service times from one of two exponential distributions with different means with some probability. The sample means of different systems are visualized in Figures 6a and 6b. We first note that the deterministic system has the lowest waiting times for all n: adding randomness increases the waiting times. Furthermore, we see that hyper-exponential distributed service times lead to longer waiting times due to there being a small probability of large service times occurring.

The statistics of the results from Figure 6 are shown in Table 3. As clear from the graphs, the $M/D/n$ system has the lowest mean waiting times, then the $M/M/n$ system, and finally, the $M/H/n$ system has the highest average waiting times for all n. The random systems ($M/M/n$ and $M/H/n$) have higher sample variances compared to the deterministic ($M/D/n$) system, as is expected. The $M/H/n$ system has the highest sample variance of all due to it sampling from an exponential distribution with a higher mean by some probability. We perform pair-wise F-tests and Welch-tests ($\alpha = 0.05$) between every type of system and every n (Table 4). We find significant differences in the sample mean and sample variance of all types of systems for all n with corrected p-values $\ll 0.001$. As seen in previous experiments, the mean waiting time decreases as the number of servers increases.



(a) Box plot of waiting times.



(b) Ridgeline plot of waiting times.

Figure 6: *Data distributions of mean waiting times in $M/M/n$, $M/D/n$, and $M/H/n$ systems with $n \in \{1, 2, 4\}$ for $\rho = 0.9$. Shown data is collected from 500 runs with 10^5 time steps. Arrival rates are scaled to ensure even system load across systems with different number of servers: $\lambda_n = 0.9n$. The service rate μ is fixed at 1 for $M/M/n$ and $M/D/n$ systems, while for $M/H/n$ we have two different distributions with $\mu = 0.8$ and $\mu^* = 5\mu$.*

	Statistics		
	$\langle\overline{W}\rangle$	Confidence radius (p = 95%)	S^2
$M/M/1$	8.976	$1.17 \cdot 10^{-1}$	$4.40 \cdot 10^{-1}$
$M/D/1$	4.4923	$3.90 \cdot 10^{-2}$	$4.93 \cdot 10^{-2}$
$M/H/1$	10.61	$1.41 \cdot 10^{-1}$	$6.42 \cdot 10^{-1}$
$M/M/2$	4.2270	$3.93 \cdot 10^{-2}$	$5.00 \cdot 10^{-2}$
$M/D/2$	2.1505	$1.35 \cdot 10^{-2}$	$5.93 \cdot 10^{-3}$
$M/H/2$	5.0410	$4.90 \cdot 10^{-2}$	$7.77 \cdot 10^{-2}$
$M/M/4$	1.9700	$1.38 \cdot 10^{-2}$	$6.14 \cdot 10^{-3}$
$M/D/4$	0.99954	$4.68 \cdot 10^{-3}$	$7.08 \cdot 10^{-4}$
$M/H/4$	2.3286	$1.73 \cdot 10^{-2}$	$9.66 \cdot 10^{-3}$

Table 3: *Statistical analysis for the $M/M/n$, $M/D/n$, and $M/H/n$ systems with $n \in \{1, 2, 4\}$ for $\rho = 0.9$. Data is based on 500 repeated runs with 10^5 time steps.*

	F-test	*Welch-test*
$M/M/1$ vs $M/D/1$	$\ll 0.001$	$\ll 0.001$
$M/M/1$ vs $M/H/1$	$\ll 0.001$	$\ll 0.001$
$M/D/1$ vs $M/H/1$	$\ll 0.001$	$\ll 0.001$
$M/M/2$ vs $M/D/2$	$\ll 0.001$	$\ll 0.001$
$M/M/2$ vs $M/H/2$	$\ll 0.001$	$\ll 0.001$
$M/D/2$ vs $M/H/2$	$\ll 0.001$	$\ll 0.001$
$M/M/4$ vs $M/D/4$	$\ll 0.001$	$\ll 0.001$
$M/M/4$ vs $M/H/4$	$\ll 0.001$	$\ll 0.001$
$M/D/4$ vs $M/H/4$	$\ll 0.001$	$\ll 0.001$

Table 4: *Corrected p-values for F-test and Welch-test for mean waiting times of varying service time distributions.*

4 Discussion

In this report, we review some important concepts of queueing theory by experimenting with a variety of different queueing systems. First, we investigate both theoretically and experimentally the impact of increasing the number of servers on the mean waiting time for $M/M/n$ systems with the same system load. Second, we compare the first in first out (FIFO) service discipline with the shortest processing time first (SPTF) service discipline in terms of mean waiting time. Finally, we compare the mean waiting times of systems with different service time distributions for both single- and multi-server systems.

We first show analytically that $M/M/2$ systems have lower mean waiting times compared to $M/M/1$ systems. This result is then verified through simulation and statistical analysis, finding that $M/M/2$ systems have significantly smaller waiting times. We extend this result to show that $M/M/4$ systems have significantly lower mean waiting times than $M/M/2$ systems. Simulations were done for different system loads to show that this holds for any system load. We also show that as the system load approaches one, longer simulation times are necessary to reach steady-state and obtain statistically significant results. An intuitive explanation for the decrease in mean waiting time using multiple servers was provided using the Karen argument, where a customer with a relatively large service time blocks up the queue. Within a $M/M/1$ system, this increases the waiting time for all customers behind the Karen, however, with $M/M/n$ systems customers can continue to be served by the other server(s). Although the proportion of these Karens within the multi-server systems is the same as for a single-server system, the possibility of these Karens happening at every server at the same time (and thus blocking up the whole system) within the multi-server system is small.

Next, we compare the $M/M/1$-$FIFO$ system with the $M/M/1$-$SPTF$ system where we always give priority to the smallest job instead of serving the customers in order of arrival. We find that, given the same system load, the $M/M/1$-$SPTF$ system has significantly lower mean waiting times compared to the $M/M/1$-$FIFO$ system. In addition, the variance of the mean waiting times of the $M/M/1$-$SPTF$ system was significantly less. The same Karen argument as mentioned within the previous paragraph can be used to explain the decrease in waiting time using the $SPTF$ service discipline. Although customers with very high service times happen only sporadically, when using the FIFO service discipline, the waiting times for all customers behind such a Karen will increase. Using the $SPTF$ service discipline, the Karen has to wait until all customers with lower service times have been served, hence cannot block up the queue.

Finally, we compare the impact of different service time distributions: Markovian ($M/M/n$), deterministic ($M/D/n$), and hyper-exponential ($M/H/n$). We run simulations for $n \in \{1, 2, 4\}$, and find significant differences both in sample mean and sample variance for all systems and number of servers. Of these systems, the deterministic system has the lowest mean waiting time for all n. This can be explained for $M/D/1$ by the Pollaczek–Khinchine formula (Pollaczek, 1930), which shows that the variance of the service times has a strong influence on mean values. Since the service time of the deterministic system has zero variance, it ends up having the smallest mean waiting times. We further show that this also holds for $M/D/2$ and $M/D/4$ systems. After that, the $M/M/n$ system has the lowest mean waiting times, and the $M/H/n$ has the highest mean waiting times for all n. The $M/H/n$ system has the highest sample variance, followed by the $M/M/n$ system, and as expected, the $M/D/n$ system has the lowest sample variance.

Although this report experiments with a large variety of queueing systems, it is a mere glimpse into the models and theorems that encompass queueing theory. Hence, there is a lot of possibilities for extending this report's work. First, similar experiments can be conducted with higher values t to attain higher statistical significance, as well as experiments with different values of ρ. Second, and related, techniques for reducing the initialization bias can be implemented to faster reach the steady-state of the system and reduce the number of measurements required by truncating the initial warm-up period (Delaney, 1995). Finally, systems with non-Markovian inter-arrival time distributions ($G/M/n$), limited queue length ($G/G/n/N$), and different service disciplines can be explored.

References

Adan, I., & Resing, J. (2015). Department of mathematics and computing science eindhoven university of technology po box 513, 5600 mb eindhoven, the netherlands march 26, 2015.

Bell, N. (2018). *Introduction to statistics*. Tritech Digital Media.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, *8*, 3–62. https://ci.nii.ac.jp/naid/20001561442/en/

Delaney, P. J. (1995). *Control of initialization bias in queueing simulations using queueing approximations.* (tech. rep.). VIRGINIA UNIV CHARLOTTESVILLE SCHOOL OF ENGINEERING and APPLIED SCIENCE.

Erlang, A. K. (1909). The theory of probabilities and telephone conversations. *20*.

Erlang, A. K. (1920). Telephone waiting times. *31*.

Haramoto, H., Matsumoto, M., Nishimura, T., Panneton, F., & L'Ecuyer, P. (2008). Efficient jump ahead for $\mathbb{F}_2$-linear random number generators. *INFORMS Journal on Computing*, *20*(3), 385–390.

Heumann, C., Schomaker, M., & Shalabh. (2017). *Introduction to statistics and data analysis: With exercises, solutions and applications in r*. Springer International Publishing AG.

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.

Little, J. D. C. (1961). A proof for the queuing formula: $L = \lambda W$. *Operations Research*, *9*(3), 383–387. https://doi.org/10.1287/opre.9.3.383

Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *8*(1), 3–30.

Meares, H., & Jones, M. (2020). When a system breaks: Queueing theory model of intensive care bed needs during the covid-19 pandemic. *Medical Journal of Australia*, *212*. https://doi.org/10.5694/mja2.50605

Nelson, R. (2013). *Probability, stochastic processes, and queueing theory: The mathematics of computer performance modeling*. Springer Science & Business Media.

Perlman, Y., & Yechiali, U. (2020). Reducing risk of infection – the covid-19 queueing game. *Safety Science*, *132*, 104987–104987.

Pollaczek, F. (1930). Über eine aufgabe der wahrscheinlichkeitstheorie. i. *Mathematische Zeitschrift*, *32*(1), 64–100.

Ross, S. M. (2013). *Simulation* (Fifth edition.). Academic Press.

Simpy. (2020). Retrieved 2021-11-30, from https://simpy.readthedocs.io/en/latest/index.html

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

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

Zimmerman, S., Rutherford, A. R., van der Waall, A., Norena, M., & Dodek, P. (2021). A queuing model for ventilator capacity management during the covid-19 pandemic. *medRxiv*. https://doi.org/10.1101/2021.03.17.21253488