University of Windsor

Scholarship at UWindsor

Major Papers

Theses, Dissertations, and Major Papers

2018

QUEUES WITH SERVER UTILIZATION OF ONE

Robert Aidoo
University of Windsor, aidoor@uwindsor.ca

Queues With Server Utilization Of One

by

Robert Aidoo

A Major Research Paper
Submitted to the Faculty of Graduate Studies
through the Department of Mathematics and Statistics
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2018

Queues With Server Utilization Of One

by

Robert Aidoo

APPROVED BY:

A. Hussein
Department of Mathematics and Statistics

M. Hlynka, Advisor
Department of Mathematics and Statistics

January 11, 2018

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify, to the best of my knowledge, that my thesis does not infringe upon anyone's copyright or violate any proprietary rights and that any ideas, techniques, quotations, or any other materials from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted materials that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission of the copyright owner(s) to include such materials in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by committee and the Graduate Studies Office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

In most queueing systems of type $GI/G/1$, the stability condition requires that the server utilization be strictly less than 1. The standard exception is a $D/D/1$ system in which stability still holds for server utilization equal to 1. This paper presents other cases when server utilization can equal 1, and discusses their characteristics.

Acknowledgements

I would like to express my deepest gratitude to Dr. Myron Hlynka for his supervision and encouragement. In fact, his unflinching support has been helpful during my studies. Whenever I run into trouble spots and other challenges, Dr. Hlynka was always open and available to assist. His enthusiasm and outstanding achievement has been a source of motivation to study.

Also, I would like to express my thanks to my committee. In particular, I would like to acknowledge Dr. Abdulkadir Hussein for his support and mentorship. I really appreciate the time you took from your busy schedule to read and evaluate my work.

This major paper would not have been a success without some important people in my life. For this reason, I would like to express my profound gratitude to my family for unfailing support and continuous encouragement. I am also grateful for friends who have been supportive along the way.

My sincere thanks also goes to all the faculty, staff and students in the department for your help and friendship, which has been a contributing factor to my progress. Ms. Dina Labelle has been helpful, and I appreciate everything you have done for me.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1. Scope and Motivation of Research

In 1908, a mathematician by name Agner Krarup Erlang was the first to propose a method for solving a queueing theory problem for telephone call exchanges by using the Poisson process [1]. Currently, queueing theory has widespread applications to numerous systems that provide service.

Queues are ubiquitous and are commonly seen in places such as fast-food restaurants, retail shops, hospitals, airports, motor vehicle traffic congestion, electoral polls, etc. Primarily, queues are designed to manage and control customers with the goal of improving productivity, by decreasing waiting times and increasing the number of customers being served [2]. Queueing theory has many applications in everyday life. An article by Lawrence Wein, a professor at the School of Business at Stanford mentioned the use of queueing theory to study the effect of bioterrorism in USA, as well as suggesting ways of reducing the waiting times of patients receiving medication, which would subsequently reduce the number of deaths in case of any such attacks [3]. With increasing application of queues to model situations, managers are looking for efficient ways of meeting the expectations of customers by reducing the waiting times in queues. Others have suggested strategies for keeping customers happy while they wait in queues. Richard Larson, the head of the Center for Engineering Systems Fundamentals at the Massachusetts Institute of Technology has said that queueing experience can improve by eliminating waiting times [4].

In this major paper, we studied queues with server utilization of one by using Lindley's recursion formula. Several examples of queueing systems with server utilization of one will be considered, in particular, the $M/M/1$, $M/D/1$, $D/M/1$, and $D/D/1$. Furthermore, we illustrated queueing systems graphically in order to deepen our understanding of such queues. More importantly, we addressed the issue of scheduling customers such that no one waits in the queue by considering special cases of a modified $D/D/1$ system.

CHAPTER 2

Markovian Queueing Systems

2.1. *M/M/1* Queueing Systems

The following information is standard in queueing texts, such as [5]. The $M/M/1$ queueing system is a classic example of a queueing system. Server utilization is simply the percentage of time during which the server is busy processing jobs. In the $M/M/1$ queueing system, M denotes Markov (memoryless, i.e. exponential distribution). Typically, the first letter represents the inter-arrivals (λ, rate per unit time), the second letter represents the service (μ, rate per unit time), and the third represents the number of servers. An $M/M/1$ system is a birth and death process, which assumes the states of the system are 0, 1, 2..., as shown in Figure 2.1 below:

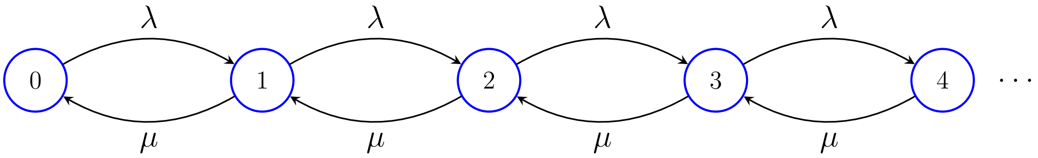

Figure 2.1. General Birth and Death Process

The birth and death process of the $M/M/1$ queueing system has a rate matrix, Q of form:

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \dots \\ 0 & \mu & -(\lambda+\mu) & \lambda & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{1}$$

From the rate matrix, Q, one can determine the limiting probabilities by the fact that the sum of all limiting probabilities is one, i.e $\vec{O} = \vec{\pi}Q$. The limiting probabilities satisfy:

$$\begin{aligned}
\pi_1 &= \left(\tfrac{\lambda}{\mu}\right)\pi_0 \\
\pi_2 &= \left(\tfrac{\lambda}{\mu}\right)^2\pi_0 \\
\pi_3 &= \left(\tfrac{\lambda}{\mu}\right)^3\pi_0 \\
&\vdots \\
\pi_n &= \left(\tfrac{\lambda}{\mu}\right)^n\pi_0, \quad \text{for} \quad n \geqslant 0. \\
&\vdots
\end{aligned} \tag{2}$$

Also,

$$1 = \textstyle\sum_{n=0}^{\infty}\pi_n = \sum_{n=0}^{\infty}\left(\tfrac{\lambda}{\mu}\right)^n\pi_0 \tag{3}$$

From (2) and (3), we obtain $\pi_0 = \left\{\sum_{n=0}^{\infty}\left(\tfrac{\lambda}{\mu}\right)^n\right\}^{-1} = 1-\rho, \quad \text{where} \quad \rho = \tfrac{\lambda}{\mu}$. It follows that for all $n \geqslant 0$,

$$\pi_n = \tfrac{\lambda^n}{\mu^n}\left(1-\tfrac{\lambda}{\mu}\right) = \rho^n(1-\rho), \quad \text{for} \quad n \geqslant 0 \tag{4}$$

This series (3) converges as long as $\tfrac{\lambda}{\mu} < 1 \quad \text{or} \quad \lambda < \mu$. Generally, the service utilization for $M/M/1$ and $M/G/1$ is defined as $\rho = \tfrac{\lambda}{\mu}$ or $\rho = \lambda E(S)$, where $E(S)$ is the expected service time. Clearly, equation (4) above is a geometric distribution of the form $\Pr(X = x) = (1-p)^x p$, with $p = 1 - \tfrac{\lambda}{\mu}$, and expected length of queue and expected wait are as follows:

$$E(L) = \tfrac{1-(1-\frac{\lambda}{\mu})}{1-\frac{\lambda}{\mu}} = \tfrac{\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}} = \tfrac{\rho}{1-\rho} \quad \text{and} \quad E(W) = \tfrac{1}{\mu-\lambda} \tag{5}$$

We quote from [5] (Gross et al.), “When $\rho = 1$, unless arrivals and service are deterministic and perfectly scheduled, no steady state exists, since randomness will prevent the queue from ever emptying out and allowing the servers to catch up, thus causing the queue to grow without bound.” We demonstrated in this paper that more precision is needed when considering the $\rho = 1$ case.

The following properties and results (see [6]) are useful for queues with server utilization of one:

PROPERTY 2.1. For an $M/M/1$ queueing system with $\rho = \frac{\lambda}{\mu} = 1$, then $E_{10} = \infty$, i.e. the expected number of steps to move from state 1 to state 0 is infinity, where states represent the number of customers in the system [6].

PROOF. This can be viewed as a one-sided random walk, it follows that;

$$P(\text{move up 1 state} \mid \text{state} > 0) = \tfrac{\lambda}{\lambda+\mu} = P(\text{move down 1 state} \mid \text{state is} > 0)$$
$$\text{Let } X = \text{the number of steps to move from 1 to 0 for the first time.}$$
$$\text{When } \rho = 1,\ \tfrac{\lambda}{\lambda+\mu} = \tfrac{1}{2}, \text{ so}$$

$$\begin{gathered} E(X) = \tfrac{1}{2}(1) + \tfrac{1}{2}(E(X) + E(X)) \\ \implies 0 = \tfrac{1}{2}, \text{unless } E(X) = \infty. \\ \therefore E(X) = \infty \end{gathered}$$

□

PROPERTY 2.2. For an $M/M/1$ queueing system with $\rho = \frac{\lambda}{\mu} = 1$, then $P_{00} = 1$. In other words, the probability of return to zero (no customer) is one if the system begins in position zero [6].

PROOF. This is a one-sided random walk;

$$\begin{gathered} \text{Note that the system will always move from 0 to 1 eventually if } \lambda > 0. \\ \text{Note that } \quad \tfrac{\lambda}{\lambda+\mu} = P(\text{move down 1 state} \mid \text{state is} > 0) = \tfrac{1}{2} \\ \text{Let } p = \text{probability of eventual return to zero} = P_{00} \\ p = \tfrac{1}{2} + \tfrac{1}{2}p^2, \\ \implies p^2 - 2p + 1 = 0 \\ \implies (p-1)(p-1) = 0 \\ \therefore p = 1 \end{gathered}$$

□

An $M/M/1$ queueing system could be simulated to obtain the figure below, such that the inter-arrival times and service times are exponentially distributed. For $\rho = 1$, the $M/M/1$ queueing system will have a limiting probability 0 for every state. In this queueing system, we make the assumption that customers are served on a first-come, first-served basis, as usually seen in practice. It is important to note that the $M/M/1$ queueing system with $\frac{\lambda}{\mu} < 1$ has limiting probabilities which are geometrically distributed, as shown in equations (2), (3) and (4). From the sample path in Figure 2.2, we could determine the maximum number of customers or time length of the queue for any busy period. The expected waiting time can be found by applying Little's law, $E(L) = \lambda E(W)$.

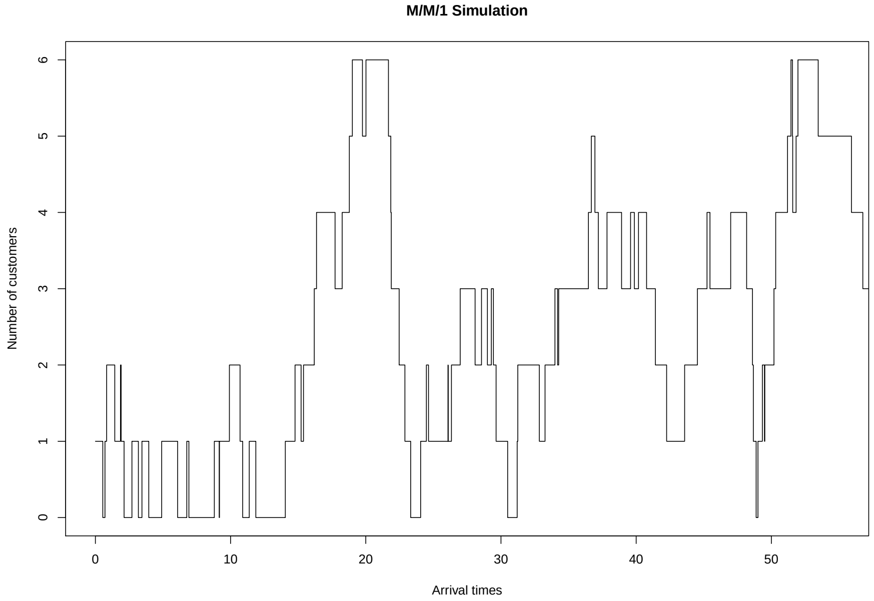

FIGURE 2.2. $M/M/1$ simulation

2.2. *D/D/1* Queueing Systems

Another queueing system with server utilization of one worth investigating is the $D/D/1$ queueing model. Unlike the $M/M/1$ system where inter-arrival times and service times are exponentially distributed, in the case of $D/D/1$ both the inter-arrival times and service times are deterministic (fixed). As an example of how this simulation behaves, we consider replicates of ones for both inter-arrival times and service times of 50 digits each. Here inter-arrival times, $T = \{1, 1, 1, ..., 1\}$ and service times are $S = \{1, 1, 1, ..., 1\}$. Clearly, in this example the waiting time of each customer is zero. Careful examination of Figure 2.3

reveals an instantaneous spike at zero. Such a system would occur in an assembly line in which the inter-arrival time matches the service time exactly.

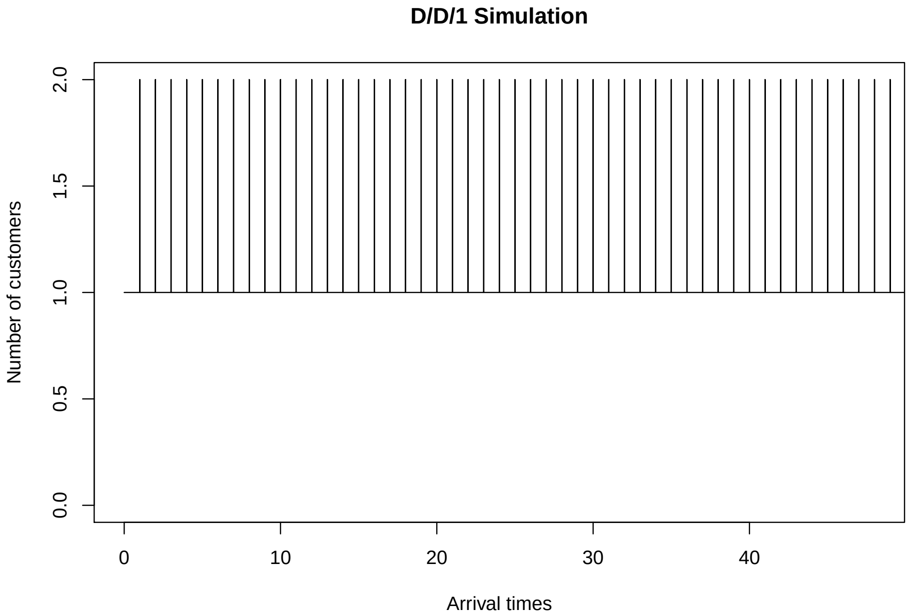

FIGURE 2.3. $D/D/1$ simulation: replicates of inter-arrival and service times

We next introduce a new class of queueing systems, which are not of the traditional $D/D/1$ type with $\rho = 1$, having identical constant inter-arrival and service times. We refer to the new class as "**modified**" $D/D/1$ systems.

A new novel example of a modified $D/D/1$ queueing system worth considering consists of sequence of numbers for both inter-arrival times and service times. In this case the inter-arrival times, $T = \{0, 1, 2, 3, ..., 49\}$ and service times, $S = \{1, 2, 3, ..., 50\}$. This queueing system has a server utilization of one, and also the waiting times between successive customers is zero. Moreover, the inter-arrival times between successive customers increase as time increases, and this explains the increasing gaps in Figure 2.4. We could repeat the values in T and S indefinitely to get a system as $t \to \infty$.

Another new fancy example of a modified $D/D/1$ system is obtained by setting the inter-arrival times, $T = \{0, 1, 2, 3, ..., 49\}$ and then reversing the order of the service times, $S = \{50, 49, 48, ..., 1\}$. Just as in the two previous examples, both inter-arrival times and service in this system are deterministic. As shown

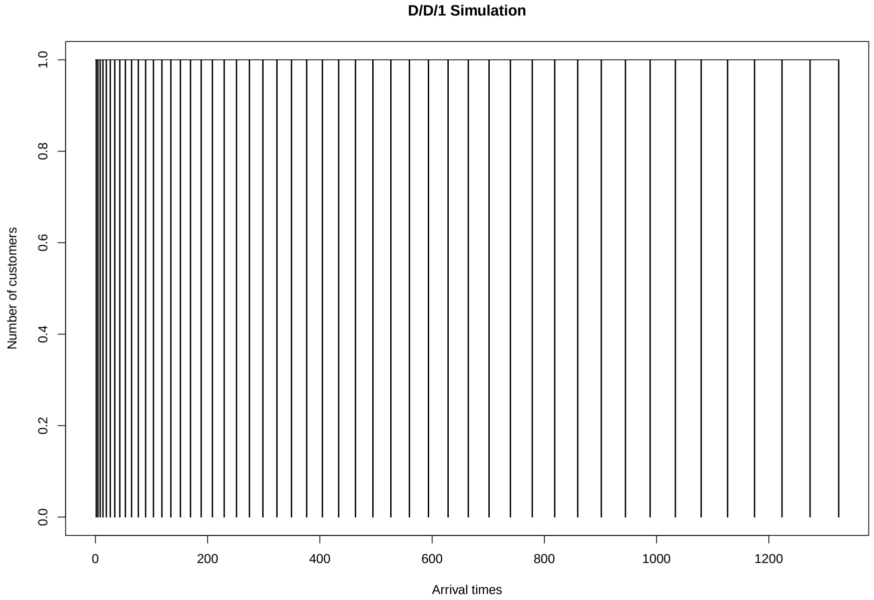

FIGURE 2.4. $D/D/1$ simulation: sequences of inter-arrival and service times

in Figure 2.5, the rate at which customers arrive increases while the service rate decreases. The right half of the graph shows an decreased rate of arrivals and an increase rate of service. This is an example of how "bad" the system, can be, in an extreme case.

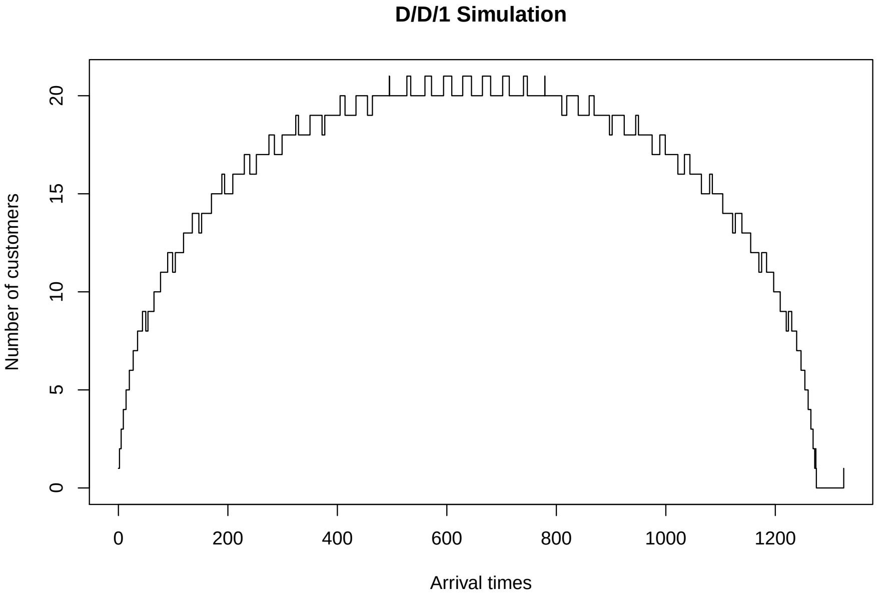

FIGURE 2.5. $D/D/1$ simulation: reversed sequences of times

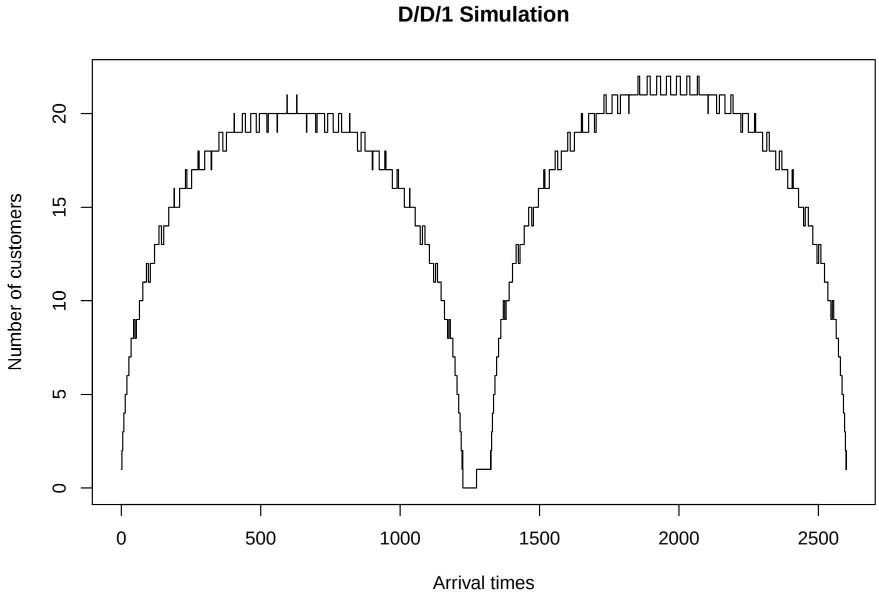

FIGURE 2.6. $D/D/1$ simulation: double reversed sequences of times

The above process can be repeated. For example, by repeating the sequences twice, we produce Figure 2.6 above.

Still another new example results from interchanging the values for inter-arrival times and services, and then repeating the procedure as shown in Figure 2.7. In this case, $T = \{51, 50, 49, ..., 1\}$ and $S = \{1, 2, 3, ..., 50\}$.

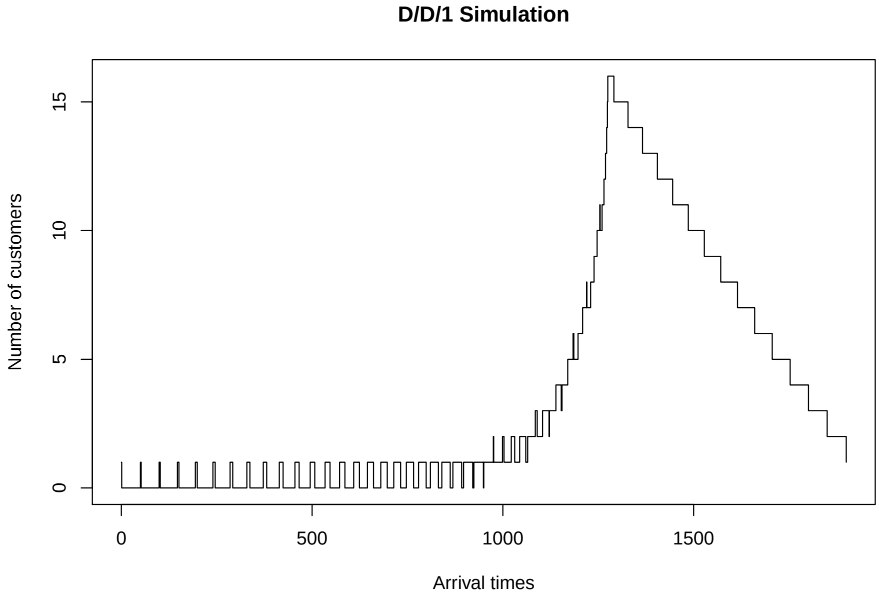

FIGURE 2.7. Modified $D/D/1$ simulation: reversed sequences of times

As shown in Figure 2.7 above, when the first customer enters the queue, it takes 50 time units for the next customer to arrive in queue. So the server utilization is not equal to 1 initially. Further to that, the next customer takes 49 time units to arrive in the queue, and the waiting times decrease as time increases. Also, we realized that the inter-arrival times did not match service times perfectly, and this explains the congestion between 1000 to 1500 time units. The tail end shows a sudden drop since all arrivals are used up, and only services remain. Server utilization is not 1 since the server is empty for much of the left hand side.

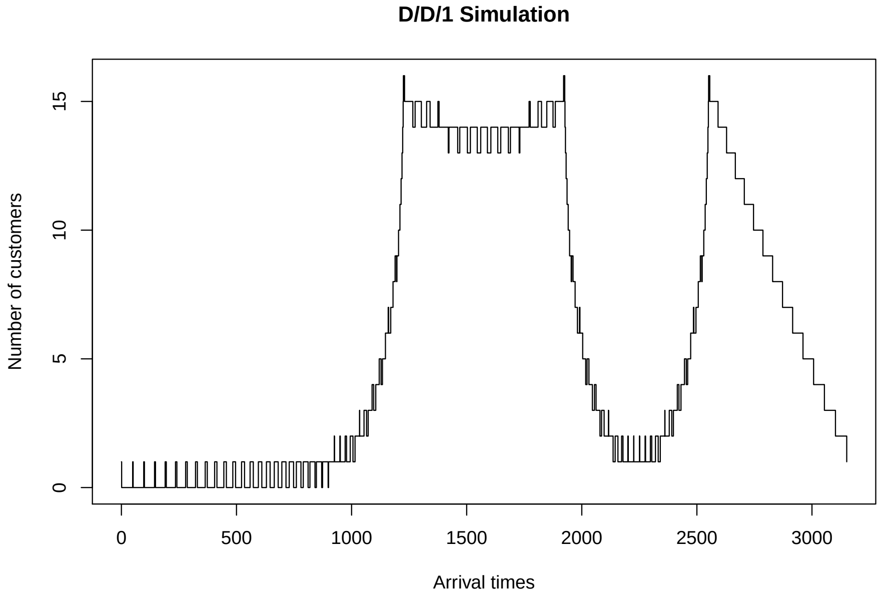

FIGURE 2.8. Modified $D/D/1$ simulation: double reversed sequences of times

In like manner, Figure 2.8 behaves the same way as Figure 2.7 except that the process is repeated twice. The waiting times between successive customers decrease over time, but after 1000 time units the arrival rates increases. Around 1500 time units, the waiting times decrease. However, since there are many customers in the system, most of them are served after 2000 time units. The remaining half of the diagram, around 2200 time units is the same as Figure 2.7. If we repeated the pattern indefinitely, the server utilization would equal 1.

2.3. *M/D/1* Queueing Systems

For the $M/D/1$ queueing system, the inter-arrival times are exponentially distributed and service times are deterministic. In other words, the inter-arrival times are determined by a Poisson process while the service times are non-random. The $M/D/1$ queueing system is not stable for $\rho = 1$, and we show with probability one the system returns to zero and the expected return time is infinity, as in the $M/M/1$ case. From Figure 2.9, we could easily observe a pattern in the service, which is quite predictive.

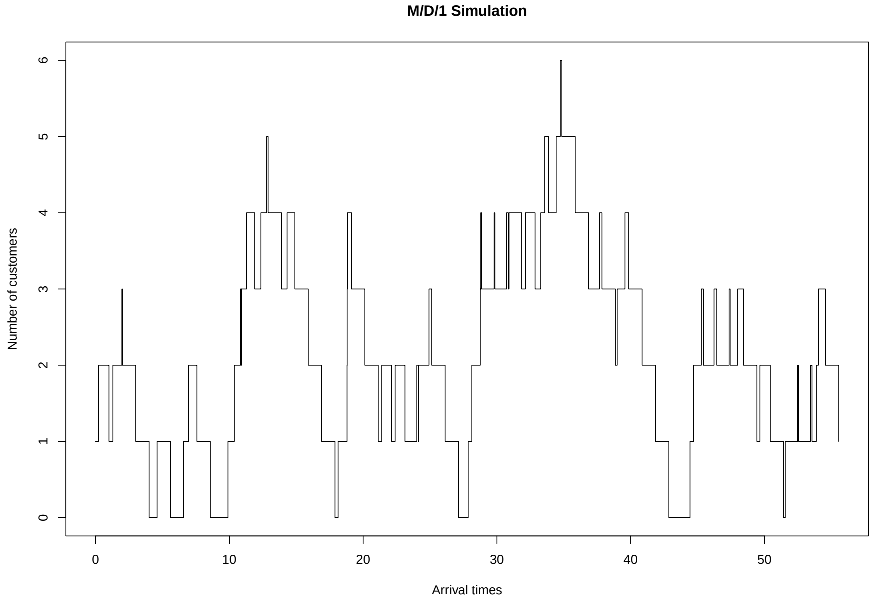

FIGURE 2.9. $M/D/1$ simulation

For the $M/D/1$ queueing system, the following two new theorems are proposed:

THEOREM 2.3. *With* $\rho = \frac{\lambda}{\mu} = 1$*,* $E_{10} = \infty$ *for* $M/D/1$ *system, where* E_{10} *is the expected number of steps to move from 1 to 0.*

PROOF. Without loss of generality, assume $\lambda = 1$, $\mu = 1$. The probability of no arrival before the first service time is $\frac{1^0 e^{-1}}{0!} = e^{-1}$.
Otherwise, there are $n > 0$ arrivals during the first service time with probability $\frac{1^n e^{-1}}{n!}$.

If there are n arrivals followed by one service completion, then the number of customers remaining is $1 + n - 1 = n$ and the number of steps used is $n + 1$.
So,

$$\begin{gathered}
E_{10} = 1 * \mathrm{P}(\text{no arrivals}) + \textstyle\sum_{n=1}^{\infty} \mathrm{P}(n \text{ arrivals})(n + 1 + E_{n0}) \\
\implies E_{10} = e^{-1} + \textstyle\sum_{n=1}^{\infty} \frac{1^n e^{-1}}{n!}\big(\ n + 1 + nE_{10}\big) \\
\implies E_{10} = e^{-1} + \textstyle\sum_{n=1}^{\infty} \frac{1^n e^{-1}}{n!} n + \sum_{n=1}^{\infty} \frac{1^n e^{-1}}{n!} + E_{10} \sum_{n=1}^{\infty} \frac{1^n e^{-1}}{n!} n \\
\implies E_{10} = e^{-1} + 1 + \big(1 - e^{-1}\big) + E_{10}\big(1\big) \\
\implies E_{10} = 2 + E_{10}
\end{gathered}$$

since,

$$\begin{gathered}
\textstyle\sum_{n=1}^{\infty} \frac{1^n e^{-1}}{n!} n = \sum_{n=0}^{\infty} \frac{1^n e^{-1}}{n!} n = E(X) = 1 \\
\text{where } X \text{ is Poisson}(1) \\
\text{Thus} \quad 0 = 2 \quad \text{or} \quad E_{10} = \infty. \\
\text{Hence} \quad E_{10} = \infty.
\end{gathered}$$

□

THEOREM 2.4. *With $\rho = \frac{\lambda}{\mu} = 1$, $P_{10} = 1$ for the $M/D/1$ system, where P_{10} is the probability of moving from 1 to 0 customer eventually.*

PROOF. Without loss of generality, take $\lambda = 1$, $\mu = 1$.

$$\begin{gathered}
P_{10} = e^{-1} + \textstyle\sum_{n=1}^{\infty} \frac{1^n e^{-1}}{n!} P_{0+n,0} \\
\text{But} \quad P_{n0} = P_{10}^n \\
\text{so,} \quad P_{10} = e^{-1} + \textstyle\sum_{n=1}^{\infty} \frac{1^n e^{-1}}{n!} P_{10}^n \\
\implies P_{10} = \textstyle\sum_{n=0}^{\infty} \frac{1^n e^{-1}}{n!} P_{10}^n \\
\implies P_{10} = e^{-1} e^{P_{10}} \\
\implies P_{10} = e^{P_{10} - 1}
\end{gathered}$$

See graph of $x = e^{x-1}$, Figure 2.10 below.

so $P_{10} = 1$ is the unique solution.

Hence the result.

□

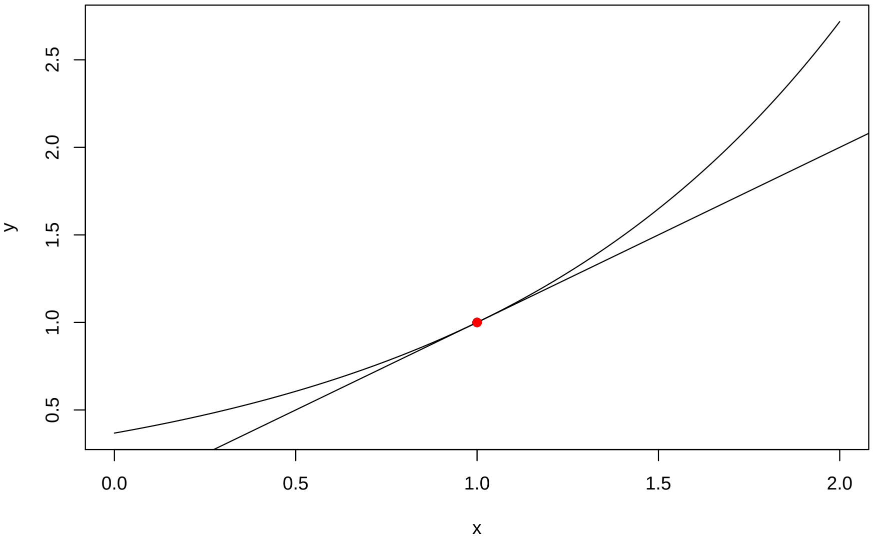

FIGURE 2.10. Graph of $x = e^{x-1}$

2.4. *D/M/1* Queueing Systems

Similar to the $M/D/1$ is the $D/M/1$ queueing system which has fixed or deterministic inter-arrival times and exponentially distributed service times. Note the pattern in which customers arrive is deterministic as shown in Figure 2.11.

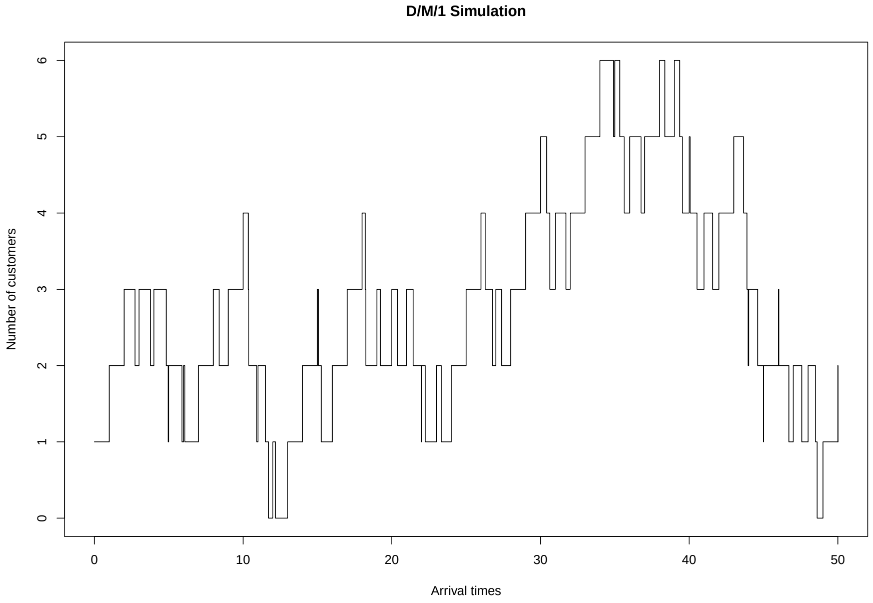

FIGURE 2.11. $D/M/1$ simulation

CHAPTER 3

Lindley's Recursion

3.1. Application of Lindley's Recursion

In this section, Lindley's recursion is implemented on a simple queueing system with a single server, which is made up of arrival times and service times. Lindley's recursion is one of the important relations in queueing theory [7], which is given by the following equation:

$$W_{n+1} = \max\{0, B_n - A_n + W_n\} \tag{6}$$

where:
A_n is the inter-arrival time between the n-th customer and $(n+1)$-st customer;
B_n is the service time of the n-th customer in queue;
W_n is the waiting time of the n-th customer (not including service time);
W_{n+1} is the waiting time of the $(n+1)$-st customer.

This will allow us to get graphical output of the type already seen, plus other cases, requiring only a list of inter-arrival times and service times. Although this graphical output is fundamental to queueing systems, we could not find R code for this so we developed our own.

By way of implementation, the following example shall be considered, of inter-arrival times and services times of eight customers.

Table 3.1. Inter-arrival and service times of a single-server queueing system

Customers	Inter-arrival time	Service time
1	0	1
2	4	1
3	5	9
4	2	6
5	5	3
6	5	8
7	9	7
8	9	3

Table 3.1 above contains the first seventeen decimal places of π, with the odd and even positions representing the inter-arrival times and the service times respectively. It is generally believe that the digits of π are random, so using them would mean that arrival rates and service rates would have the same value. Thus ρ should be 1.

As shown in the table, the first customer has no inter-arrival time and thus the queue has one customer at time zero. Complete the table by including arrival times, time service begins, waiting times, time service ends, and time customer spends in systems as summarized in Table 3.2 below.

Table 3.2. Complete table for a single-server queueing system

Customers	Inter-arrival time	Arrival time	Service time	Beginning service time	Waiting time	Time service ends	System time
1	0	0	1	0	0	1	1
2	4	4	1	4	0	5	1
3	5	9	9	9	0	18	9
4	2	11	6	18	7	24	13
5	5	16	3	24	8	27	11
6	5	21	8	27	6	35	14
7	9	30	7	35	5	42	12
8	9	39	3	42	3	45	6
	$\sum = 39$		$\sum = 37$		$\sum = 29$		$\sum = 67$

The table is constructed as follows. The inter-arrival times and service times are known. $W_1 = 0$ is konwn. From Lindley's result, we find all waiting times. The arrival times are found from cumulative sums of inter-arrival times.

$$\text{Beginning Service Time} = \text{Arrival Time} + \text{Waiting Time};$$
$$\text{Time Service Ends} = \text{Beginning Service Time} + \text{Service Time};$$
$$\text{System Time} = \text{Time Service Ends} - \text{Arrival Time}.$$

In order to construct a graph, we need to merge the Arrival Times with Time Service Ends.

TABLE 3.3. Arrival and completion times of a single-server queueing system

Merging arrival & Completion times	Label
0	+1
1	–1
4	+1
5	–1
9	+1
11	+1
16	+1
18	–1
21	+1
24	–1
27	–1
30	+1
35	-1
39	+1
42	–1
45	–1

We merge the arrival times and service completion times of customers, and then label these arrival times and service completion times with +1 and –1 respectively. This is summarized in Table 3.3, the cumulative values of the label column represent the number of customers in the queueing system. Its graphical representation appears in Figure 3.1 below. The program should be written such that an arrival occurs before service. Code in R for this appears in Appendix A.

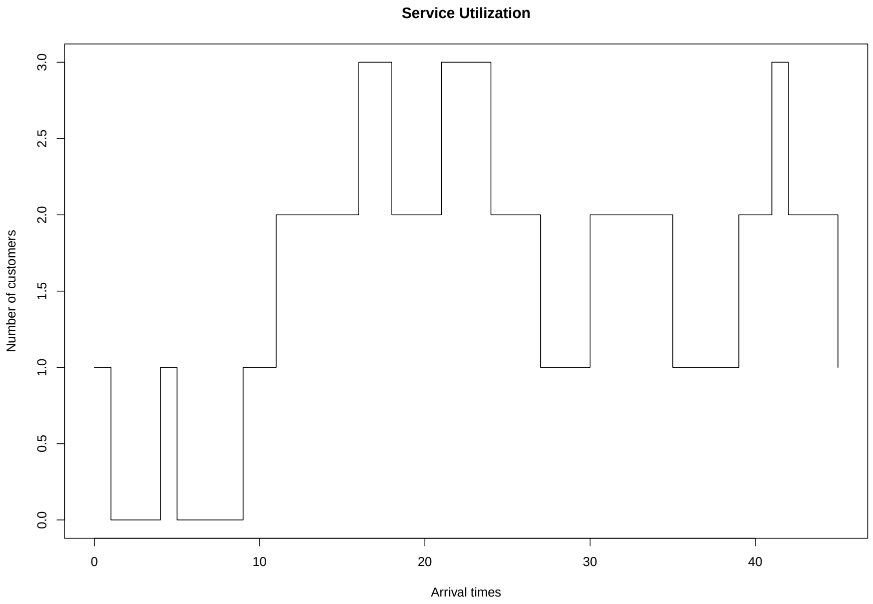

Figure 3.1. Service utilization: 17 digits of π

Similarly, the procedure could be repeated for 300, 500 and 1000 decimal places of π, which is represented graphically in Figures 3.2, 3.3 and 3.4. Not much can be deduced from these plots, and so the next chapter will focus primarily on some important cases of modified deterministic processes.

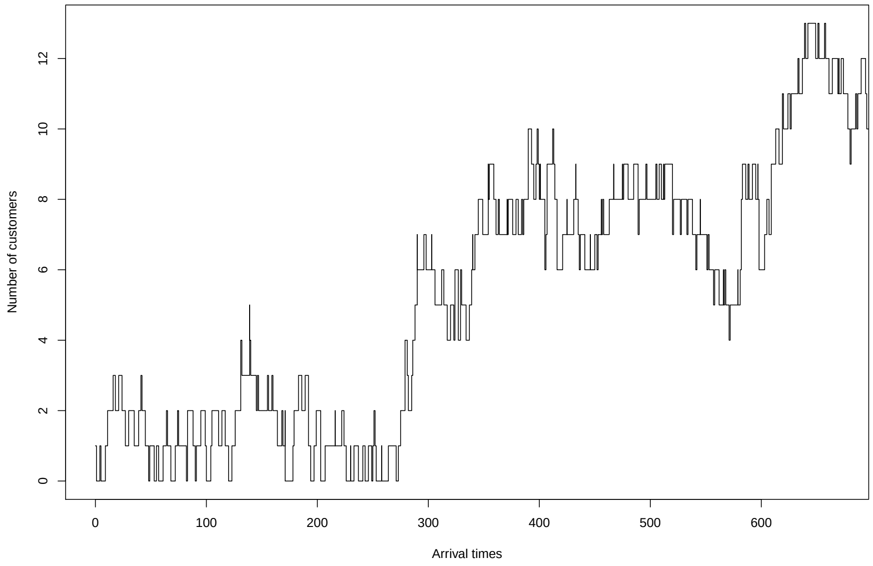

FIGURE 3.2. Service utilisation: 300 digits of π

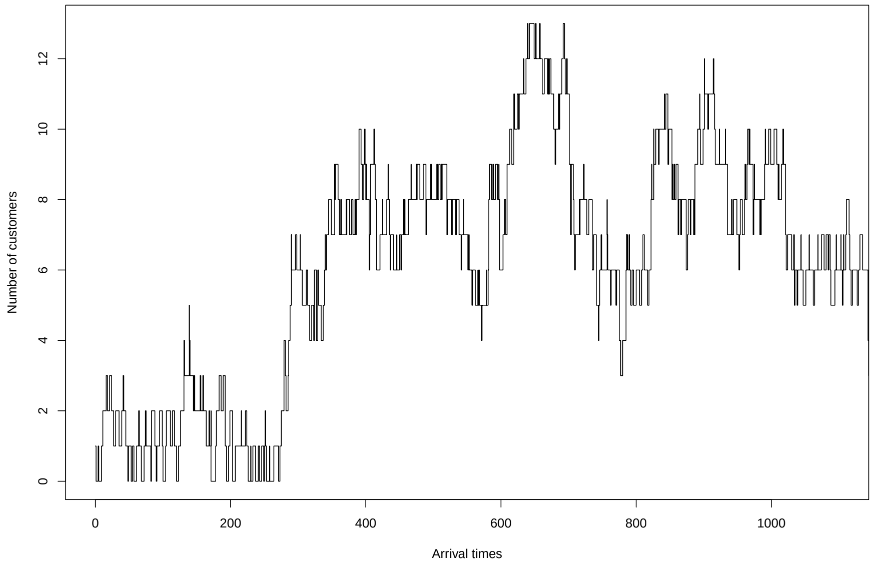

FIGURE 3.3. Service utilization: 500 digits of π

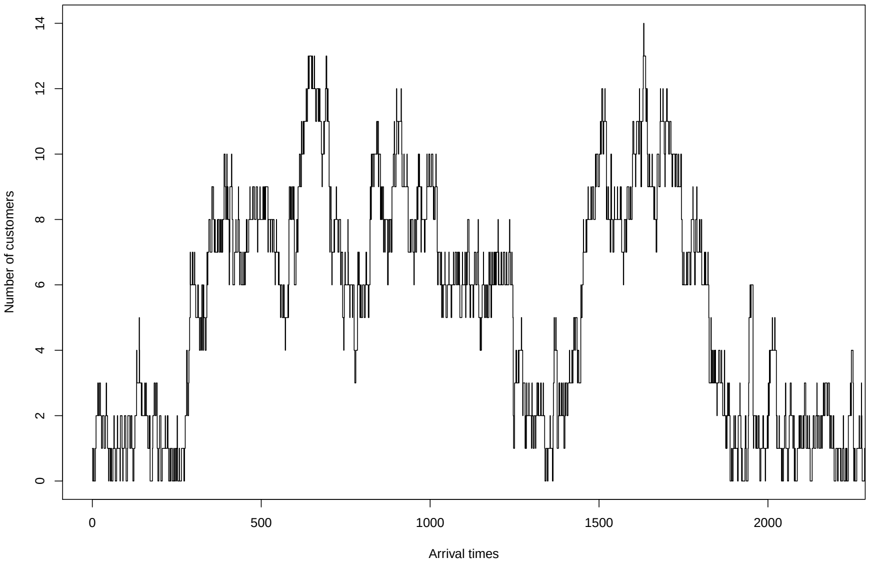

FIGURE 3.4. Service utilization: 1000 digits of π

CHAPTER 4

Modified Deterministic Processes: Special Cases

4.1. Grouped Inter-arrival and Service Times

In this chapter several examples of deterministic processes are studied, in particular grouped and sampled inter-arrival and service times. Since these cases are deterministic, one goal is to schedule the process such that the waiting times will be zero. In fact, when queues are properly scheduled it may address some of the challenges that may arise in queues, which may include zipping, jockeying, balking, reneging, etc. Throughout this chapter, we assume $\rho = 1$

Example 4.1. Consider the inter-arrival and service times of customers arriving in groups of (1, 2, 3).

In this example, the inter-arrival times are fixed but the service times are allowed to vary in the sense that we can reorder (1, 2, 3). Based on this example, the expected queue length, $E(L)$ can be determined by an examination of the first graph in Figure 4.1 by comparing the proportion of time that the system is at levels 1 and 2. Using $E(L) = E(L_q) + E(L_s)$ with $E(L_s) = 1$, we can find $E(L_q)$.

From (7), the expected length wait can be determined by using Little's Law in the following equation:

$$E(L_q) = \lambda E(W_q) \tag{7}$$

(a) Case 1: Inter-arrival times= 1, 2, 3 and Service times= 1, 2, 3 and 1, 3, 2:

This can be summarized in the table and diagram below. As explained in the previous chapter, the first has no inter-arrival time as expected:

TABLE 4.1. Inter-arrival and service times of a single-server queueing system

Customers	Inter-arrival time	Service time (1, 2, 3)	Service time (1, 3, 2)
1	0	1	1
2	2	2	3
3	3	3	2
4	1	1	1
5	2	2	3
6	3	3	2
7	1	1	1
8	2	2	3
⋮	⋮	⋮	⋮

Lindley's recursion relation is applied to the above inter-arrival and service times, which produce the following diagrams in Figure 4.1. With these diagrams one can compute the queue length for each scenario by applying equations (7) and (8). Clearly, the service times (1, 3, 2) has a shorter queue length as compared to the service times (1, 2, 3).

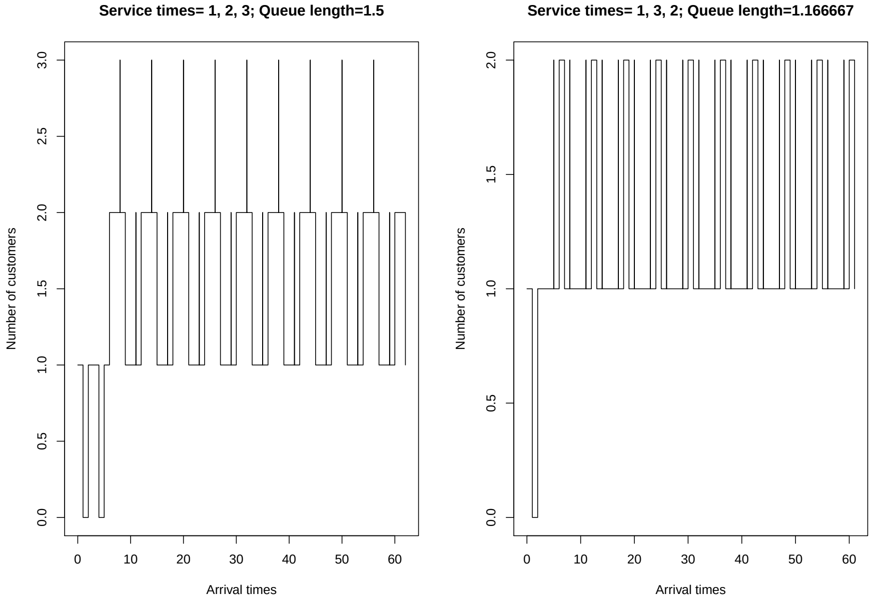

FIGURE 4.1. Case 1

The procedure as described in case 1, can be repeated by changing the order of service times, and this is clearly shown in Figure 4.2.

(b) Case 2: Inter-arrival times= 1, 2, 3 and Service times= 2, 1, 3 and 2, 3, 1:

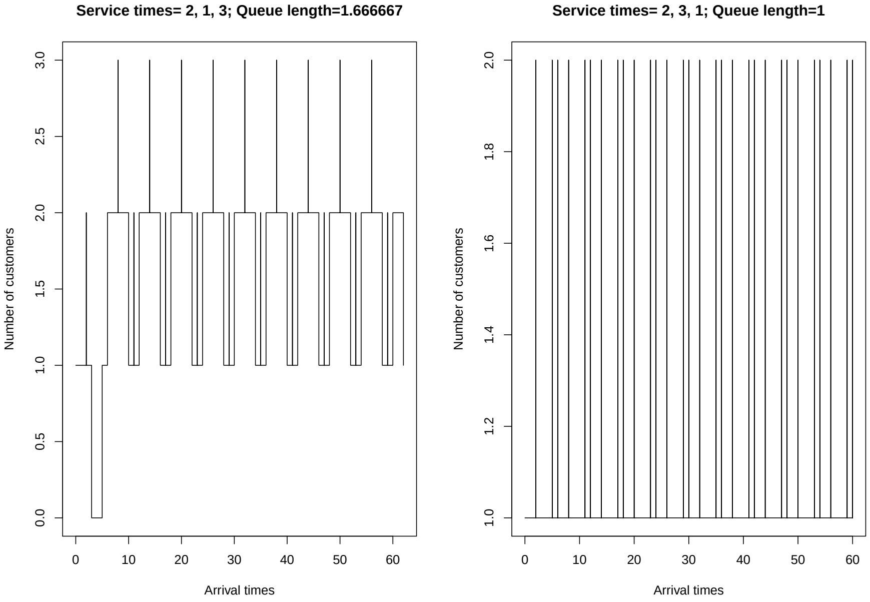

FIGURE 4.2. Case 2

(c) Case 3: Inter-arrival times= 1, 2, 3 and Service times= 3, 1, 2 and 3, 2, 1:

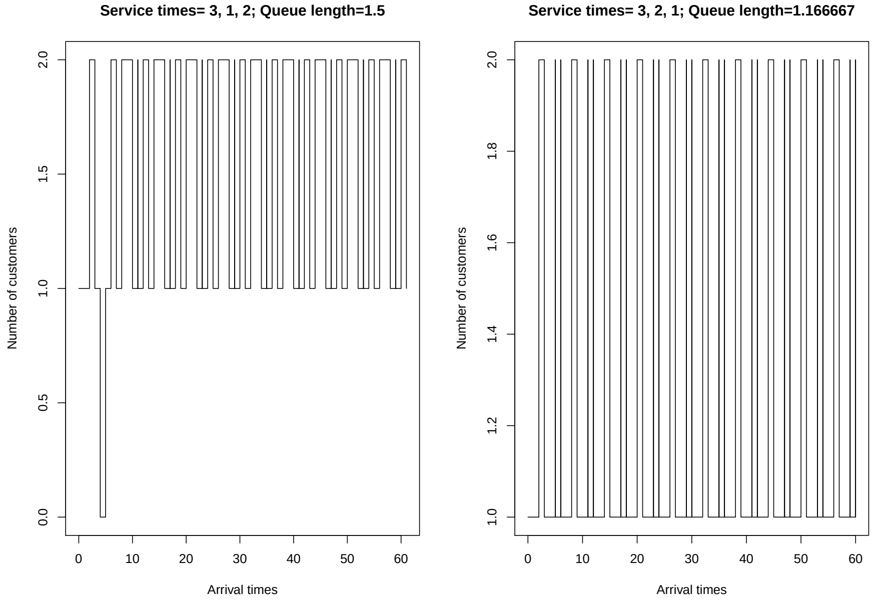

FIGURE 4.3. Case 3

Figure 4.3 illustrates the two remaining orderings of the service times.

Note that in Case 2, with service times 2, 3, 1, Figure 4.2 shows that the queue length is always 1 and no customer ever waits. This is the ideal situation and proper scheduling should always result in an ideal situation. Medical doctors, take note!

In this chapter, we showed that examples of cases other than standard $D/D/1$ systems can have $\rho = 1$ and still be stable. However, our system lost "independence" and "identically distributed" properties.

CHAPTER 5

Further Deterministic Processes

5.1. Inter-arrival and Sampled Service Times

Apart from the grouped inter-arrival and services times discussed in section 4, we can also consider cases, which are identically distributed but are not independent.

By way of illustration, the following queues with inter-arrival and sampled service times can be considered.

(a) Case 1: Inter-arrival times= 1, 2, 3 and sampled service times= 1, 2, 3:

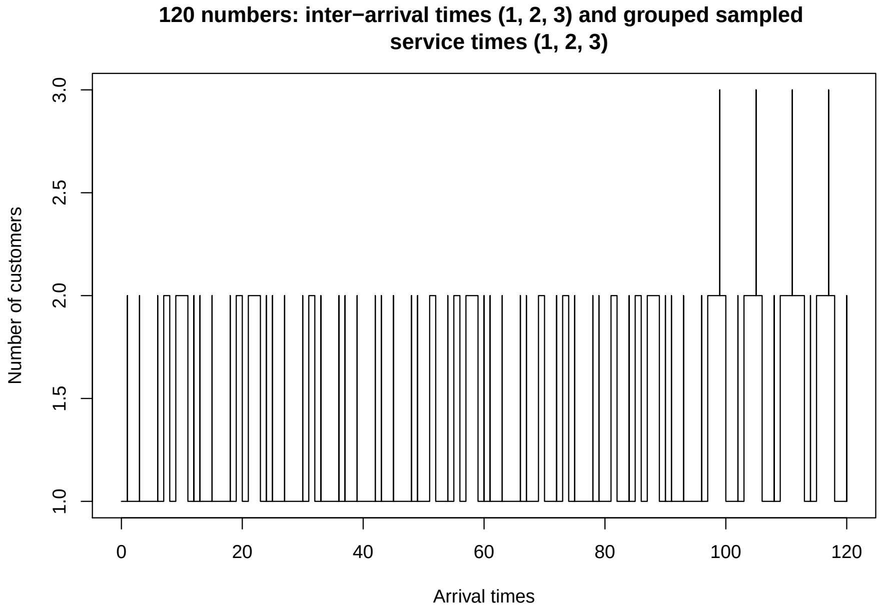

Figure 5.1. Case 1

The printout below shows the first 20 positions of inter-arrival vector and service vector:

```
> Inter-arrival Times
[1] 0 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1
```

```
> Service Times
[1] 1 2 3 2 3 1 1 2 3 2 3 1 1 2 3 2 1 3 1 2
```

Repeating Case 1 two more times produce the following:

(b) Case 2: Inter-arrival times= 1, 2, 3 and Service times= 1, 2, 3:

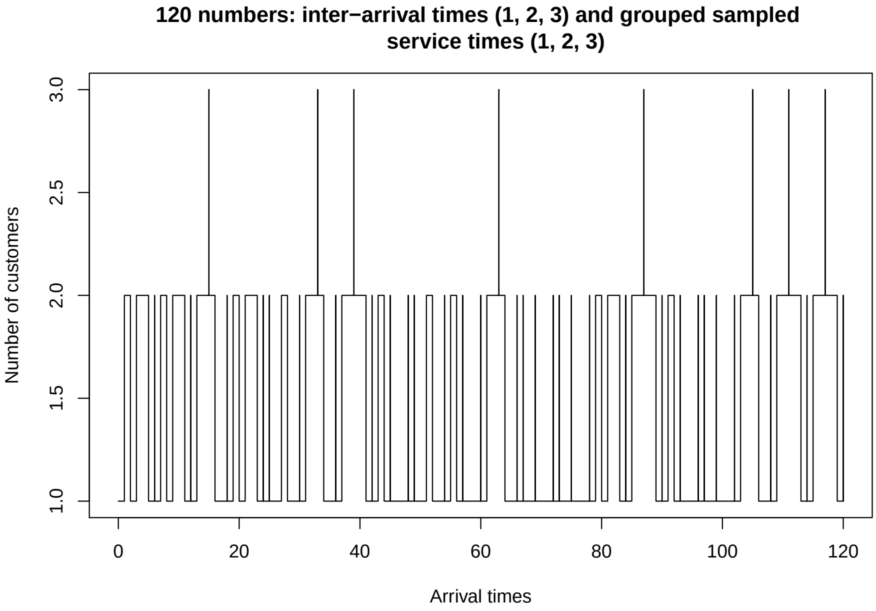

Figure 5.2. Case 2

(c) Case 3: Inter-arrival times= 1, 2, 3 and Service times= 1, 2, 3:

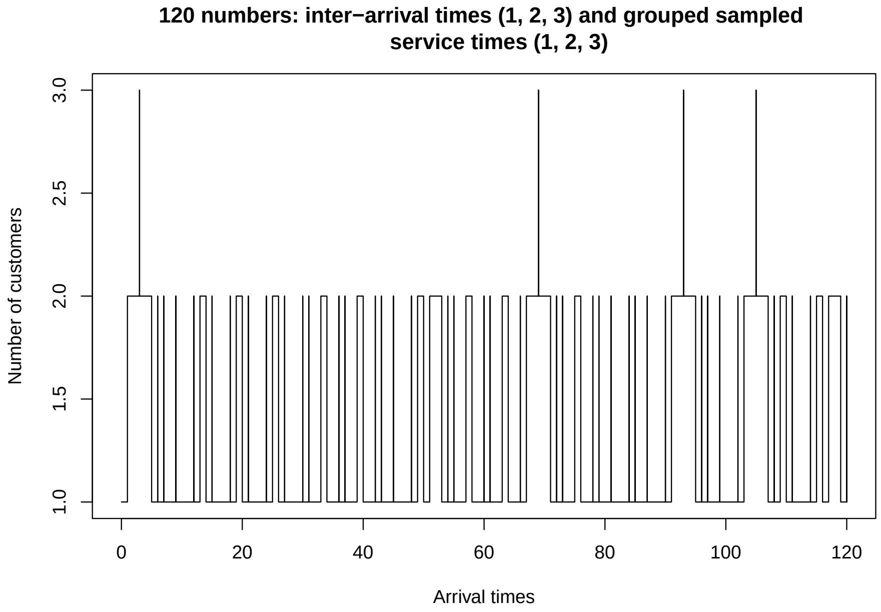

FIGURE 5.3. Case 3

PROPERTY 5.1. For a sequence of inter-arrival times (1, 2, 3) and a sampled sequence of service times (without replacement) of (1, 2, 3) the identical queue length should reappear at time $t_0 + 6, t_0 + 12, t_0 + 18, ...$ for some t_0.

REMARK 5.2. Since we sample without replacement, after $1+2+3 = 6$ services, we should be in an original position. Look at queue length versus time in the following:

Table 5.1. Queue length versus time

Customer #	Time	Queue length
1	0	1
2	1	2
3	3	3
4	6	2
5	7	1
6	9	$\textcircled{2}$
7	12	1
8	13	2
9	15	1
10	18	2
11	19	1
12	21	$\textcircled{2}$
13	24	1
14	25	2
15	27	3
16	30	2
17	31	1
18	33	$\textcircled{2}$
19	36	1
20	37	2
21	39	1
22	42	2
23	43	1
24	45	$\textcircled{2}$
⋮	⋮	⋮

As shown in Table 5.1, after every six steps the queue length is 2, which means $t_0 = 6$.

Note 5.3. In this chapter, we presented queues with identically distributed service times, which were not independent. This type of system has not been considered elsewhere in the queueing literature.

CHAPTER 6

Results and Findings

In this project, queues with server utilizations of one were discussed. Generally, queueing systems of the form $GI/G/1$ are stable if server utilization is strictly less than 1. However, there is an exception with $D/D/1$ in which stability is maintained for server utilization equal to one. In this paper, $M/M/1$, $D/D/1$, $M/D/1$, and $D/M/1$ systems were discussed. Other queueing disciplines such as periodic grouped inter-arrival times and sampled service times were examined.

In order to study the queueing behaviours, use was made of the Lindley's recursion relation for single-server queueing models. Although the Lindley's recursion is fundamental, the R code was not available on the internet. Consequently, we developed R codes for executing any single-server queue.

Most of the initial trial examples were done using the digits of π by assigning the odd places as the inter-arrival times and even places as the service times. The motivation for using π is simply because we believe that in the long run will have server utilization of one. Specifically, the mean inter-arrival times and mean service times are both 4.5 since the digits of π are uniform between 0 to 9, and the average is 4.5.

By way of adding to the literature, two new theorems (2.3 and 2.4) for $M/D/1$ queueing systems were discussed. Moreover, results from the grouped sampled of inter-arrival and service times indicate that it is possible to schedule customers such that no one need wait in queue.

CHAPTER 7

Future Work

Our future works will focus primarily on using the level-crossing approach to determine the distribution for a single-server queueing system by simulating the virtual waits. For every state-space level and sample path, the law of conservation states that the total upcrossing rate is equal to the total downcrossing rate, in the long run [8]. Figure 7.1 below took considerable effort but took advantage of the earlier graphs for the number of customers.

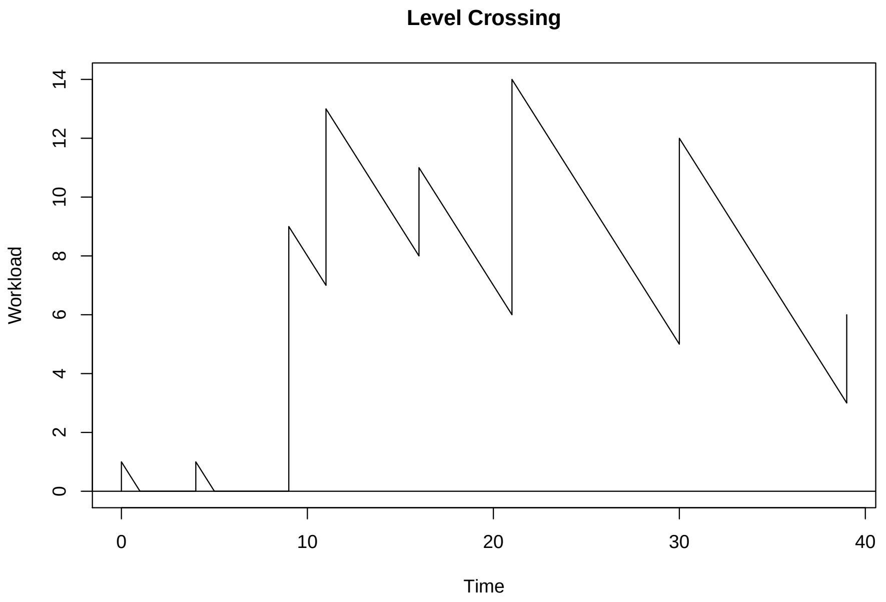

Figure 7.1. Sample path of the workload Level

Also, we will try to prove the following for $D/M/1$ with server utilisation of one: the expected number of steps to move from state 1 to 0 is ∞, and the probability to move from state 1 to 0 is 1.

Appendix A

Codes and Programs

This section shows the R programming codes and commands used in the study:

```
### THE FIRST 17 DECIMAL PLACES (DIGITS OF PI) ###

### ODD NUMBERS OF INTERARRIVALS ###

t=c(3,4,5,2,5,5,9,9,2)

t1=t[-1]

t1

### CUMULATIVE INTERARRIVAL TIMES ###

### INTERARRIVAL TIMES ###

t2=c(0,t1)

t2

ct=cumsum(t2)

ct


### EVEN NUMBERS OF SERVICE TIMES ###

s=c(1,1,9,6,3,8,7,3)


### INTERARRIVAL TIMES OF (N+1)TH AND (N)TH ###

ia=1:8

ia[1]=0

for (i in 1:8){ia[i+1]=ct[i+1]-ct[i]}

ia


### WAITING TIMES (LINDLEY EQUATION) ###

w=1:8

w[1]=0

for (i in 1:8){w[i+1]=max(0,s[i]-ia[i+1]+w[i])}

w



### SERVICE COMPLETION TIMES ###
```

```
sc=s[1:8]+ct[1:8]+w[1:8]
sc
sc[1]=1
sc
ct

### TIME CUSTOMER SPENDS IN SYSTEM ###
tsc=c()
for (i in 1:8){tcs[i]=sc[i]-ct[i]}
tcs

### MERGING SERVICE COMPLETION AND CUMULATIVE INTERARRIVAL TIMES ###
me=c(sc,ct)
me
sort(me)
ct
rep(1,8)
mct=t(matrix(c(ct, rep(1,9)),9,2))
mct
msc=t(matrix(c(sc,rep(-1,8)),8,2))
msc
merge=t(matrix(c(mct[1,],msc[1,],mct[2,],msc[2,]),17,2))
merge
merge2=t(merge)

r=merge2[order(merge2[,1],-merge2[,2]),] ## 1 to -1 is decreasing
r
plot(r[,1],cumsum(r[,2]))
plot(r[,1],cumsum(r[,2]),"s", main="Service Utilization", xlab="Arrival times",
ylab="Number of customers",xlim = c(0,42))

### LEVEL CROSSING ###
t2_new=t2[-9]
t2_new
ct_new=cumsum(t2_new)  #actual arrival times
w_new=c(0); for (i in 2:8){w_new[i]=max(w_new[i-1]+s[i-1]-t2_new[i],0)}
#Wait times (Lindley)
vir=w_new+s    #virtual service times
vir
for (i in 8:2){ if (w[i]==0) {vir=append(vir,0,i-1); w_new=append(w_new,0,i-1);
s=append(s,0,i-1); t2_new=append(t2_new,0,i-1);
ct_new=append(ct_new,ct_new[i-1]+s[i-1],i-1)}}
n=length(ct_new)
m1=matrix(c(w_new,vir),n,2)
y=as.vector(t(m1)) #y values to be plotted
m2=matrix(c(ct_new,ct_new),n,2)
x=as.vector(t(m2))   #x values to be plotted
plot(x,y,"l",xlab="Time",ylab="Workload")
abline(0,0)



#### MODIFIED D/D/1 SYSTEM ####
### INTER-ARRIVAL TIMES:(1, 2, 3) AND SAMPLED SERVICE TIMES:(1, 2, 3) ###
aa1=rep(c(1,2,3),length.out=60)
aa1
t33b=c(0,aa1)
t34b=t33b[-1]
t34b

### CUMULATIVE INTERARRIVAL TIMES ###
### INTERARRIVAL TIMES ###
t35b=c(0,t34b)
t35b
ct33b=cumsum(t35b)
ct33b

### SERVICE TIMES ###
bb1=c()
for (i in 1:20) {bb1=c(bb1,sample(c(1,2,3),3,replace = FALSE, prob = NULL))}
bb1
s33b=bb1
s33b

### INTERARRIVAL TIMES OF (N+1)TH AND (N)TH ###
ia33b=1:60
ia33b[1]=0
for (i in 1:60){ia33b[i+1]=ct33b[i+1]-ct33b[i]}
ia33b
```

```
### WAITING TIMES (LINDLEY EQUATION) ###
w33b=1:60
w33b[1]=0
for (i in 1:60){w33b[i+1]=max(0,s33b[i]-ia33b[i+1]+w33b[i])}
w33b

### SERVICE COMPLETION TIMES ###
sc33b=s33b[1:60]+ct33b[1:60]+w33b[1:60]
sc33b
sc33b
ct33b

### TIME CUSTOMER SPENDS IN SYSTEM ###
tcs33b=c()
for (i in 1:60){tcs33b[i]=sc33b[i]-ct33b[i]}
tcs33b

### MERGING SERVICE COMPLETION AND CUMULATIVE INTERARRIVAL TIMES ###
me33b=c(sc33b,ct33b)
me33b
sort(me33b)
ct33b
mct33b=t(matrix(c(ct33b, rep(1,61)),61,2))
mct33b
msc33b=t(matrix(c(sc33b,rep(-1,60)),60,2))
msc33b
merge33b=t(matrix(c(mct33b[1,],msc33b[1,],mct33b[2,],msc33b[2,]),121,2))
merge33b
merge34b=t(merge33b)

r33b=merge34b[order(merge34b[,1],-merge34b[,2]),]
r33b
plot(r33b[,1],cumsum(r33b[,2]))
plot(r33b[,1],cumsum(r33b[,2]),"s",main="120 numbers: inter-arrival times (1, 2, 3) and
grouped sampled service times (1, 2, 3)", xlab="Arrival times", ylab="Number of customers")
```

Bibliography

[1] V. Sundarapandian, *Probability, statistics and queuing theory.* Phi Learning New Delhi, 2009.

[2] (2017, June 21) Queueing theory. [Online]. Available: http://www.investopedia.com/terms/q/queuing-theory.asp

[3] E. H. Kaplan, D. L. Craft, and L. M. Wein, "Analyzing bioterror response logistics: the case of smallpox," *Mathematical Biosciences*, vol. 185, no. 1, pp. 33–72, 2003.

[4] (2017, June). [Online]. Available: http://www.cnn.com/2008/TECH/science/11/20/queuing.psychology/

[5] D. Gross, J. Shortle, J. Thompson, and C. Harris, "Fundamentals of queueing theory (fourth edition)." John Wiley, 2008, pp. 9–10.

[6] M. Hlynka, "Lecture notes in stochastic processes," September 2016.

[7] D. V. Lindley, "The theory of queues with a single server," vol. 48, no. 2, pp. 277–289, 1952.

[8] P. H. Brill, "A brief outline of the level crossing method in stochastic models," *CORS Bulletin*, vol. 34, no. 4, pp. 9–21, 2000.

Vita Auctoris

Robert Aidoo was born in Techiman, the Brong Ahafo region of Ghana in 1980. He completed the Kwame Nkrumah University of Science and Technology in 2006 with First Class honours in Mathematics. In 2012, he pursued graduate studies in Applied Mathematics at the University of Western Ontario. He is currently a masters candidate in the Statistics program at the University of Windsor, and is expected to graduate in January 2018.