National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

UNIVERSITY OF ALBERTA

Analysis of an M/M/S Queue with Vacations

BY

Anshuman Tyagi

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

DEPARTMENT OF COMPUTING SCIENCE

Edmonton, Alberta
Spring 1994

ISBN 0-612-11395-7

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Anshuman Tyagi

TITLE OF THESIS: Analysis of an M/M/S Queue with Vacations

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: 1994

(Signed) . . Anshuman Tyagi . . . . . .

Anshuman Tyagi
Tarni Street
Sardhana, Meerut
UP, India-250342.

Date: Nov. 5th. '93 . .

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Analysis of an M/M/S Queue with Vacations** submitted by Anshuman Tyagi in partial fulfillment of the requirements for the degree of Master of Science.

J. Harms (Co-Supervisor)

for A. Kamal (Co-Supervisor)

D. Kelker (External)

U. M. Maydell (Examiner)

L. Stewart (Chair)

Date: Nov. 4th, '93

Abstract

In this thesis, a multiple server queue, in which each server takes a vacation after serving one customer is studied. The arrival process is Poisson and service time is exponential. The duration of a vacation is a random variable with known distribution. Two types of distributions are considered: exponential and a phase distribution of order 2.

This kind of queue has not been considered in the literature before. This queueing model can be used for the analysis of different kinds of communication networks, such as multislotted networks, multiple token rings and multiple server polling systems.

We apply two techniques to do the steady state analysis of this model:

1. Balance Equation.

2. Matrix Geometric Method.

Using the first technique we are able to derive the mean queue length when the number of servers (S) in the system is less than 4. But the second technique, which is algorithmic, gives us the mean queue length and mean waiting time of the customers for any value of S.

Acknowledgements

I am indebted to my supervisors, Dr. Janelle Harms and Dr. Ahmed Kamal for their support, encouragement and guidance.

I am also thankful to Dr. Ahmed Kamal from whom I learn the Queueing Theory concepts which have been of great help in doing this thesis.

Finally, I thank the members of my examining committee, Dr. D. Kelker, Dr. U. M. Maydell and Dr. L. Stewart for their valuable time and suggestions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Queueing Systems with Vacations, are those in which the server(s) take "time off" or a vacation. In contrast to most queueing systems, the server is not always alert and waiting for customers to arrive and a server may leave when there are still customers to be served. A number of phenomena, which occur in real life, can be classified as a queueing system in which the server takes a vacation. A good example, that may be observed by a person almost daily, happens at a traffic light crossing where the color of traffic light indicates whether the server is on vacation or not. For the drivers (customers), the green light for their lane means that the server is present, while the red light means it is on vacation. In the latter case, the drivers must wait for the green light (server) to return. All the results which are important in the analysis of this system are generally relevant to other vacation systems. The main points of interest are the traffic accumulation and the average waiting time of the traffic at the lights, which in queueing theory are termed as queue length

distribution and mean waiting time. By this example, it is clear why the two performance measures require attention and should be studied during the analysis of Queueing Systems with Vacations.

In this thesis, we study these two performance measures for a queuing system with multiple servers that take vacations.

1.1 Applications of Vacation Models

Queueing Systems in which the server is on vacation for a certain duration of time, arise in many computer, communication and other systems. The reason for the server vacation may be due to lack of work, server failure or due to some other task being assigned. An example of the latter case occurs in the traffic light crossing analogy. In which, when the server is on vacation for drivers in the red light lane, it is serving the drivers in the green light lane.

We will discuss in detail various problems which can be reduced to queueing systems with vacation. The emphasis will be on problems related to computer systems and communication.

1. Computer Maintenance and Testing: Processors in computer and communication systems are required to do considerable testing and maintenance besides doing other primary functions (for example: processing telephone calls, receiving and transmitting data). Testing and maintenance is required for proper functioning and to increase the reliability of the system. The period during which the testing and maintenance

is done can be viewed as a vacation of the system, thus these problems can be analyzed as vacation models[5].

2. CPU Scheduling: The scheduling of different tasks can be analyzed by using a vacation model. The time during which a task does I/O and waits for the CPU can be considered as a CPU(server) vacation. These cases can be modelled as queues with multiple servers or a single server depending on the number of CPUs. These problems are also referred to as cyclic queues with different tasks switching between I/O and CPU bursts, until a task is over, when it is replaced by another task.

3. Polling Systems: A polling system is a system of multiple queues, accessed by a server(s) in a cyclic order, with a switch over time from queue i to queue $i+1$. These systems are also referred to as cyclic queues. Depending on the number of servers present in the system, they are classified as single or multiple server polling systems. They arise in many communication systems in which the different stations are served in a cyclic order. Token Ring and Token Bus protocols are good examples. In these systems the token(server) moves from one station to another in cyclic order and the station with the token is given the privilege of transmitting the message. After the message reaches the destination station the token moves to the next station. The vacation for a particular queue, in this case, is the time when the token is at other queues. Many researchers have analyzed these queues by using the results of single server queues with vacation(for example [4]). To analyze multiserver polling systems, which arise for example in mul-

tiple token ring systems, we can similiarily use the results of multiple server queues with vacation.

4. Time Division Multiple Access: The study of multiple access from a set of N data sources, to a single packet-switched data communication channel, can be done using queueing models with vacation times. Time Division Multiple Access (TDMA)[16] is a good example. In TDMA, each data source generates fixed size packets to be transmitted on a FCFS basis. The system assigns to each data source a periodic sequence of time slots on the channel (packet transmission time being equal to one slot). The channel slots are usually switched to users in a cyclic order. This system can be modelled by single server queues, where one slot is assigned to each source and thus vacation time is fixed at N−1 slots[25].

5. Priority Queues: In a two priority non pre emptive queueing system, we consider the time period when the server is serving the low priority customers, to be the server vacation from the viewpoint of high priority customers. Using this concept, priority queues can be modelled as single server vacation models[20]. An N priority queue can be analyzed by a similiar model. In this case, for class i customers, the server's vacation is the time between successive visits to that class.

6. Machine Breakdown and Maintenance in Production Systems: In a production system the machines may break down at random, thus causing a "rest period" or vacation. Thus, maintenance has to be scheduled so

as to minimize the random breakdowns. The maintenance and breakdown can again be regarded as a vacation[5]. This problem can be modelled as a single-server queue with vacation.

7. Related Models: Various other situations where the server is not always available to serve its primary customers are closely related to vacation models as mentioned in[5]. One example is queueing systems that require set-up time. During the set-up time the server is not available for service and this can be viewed as being on vacation.

1.2 Background

In this section we will discuss the parameters which describe a Queueing System. These parameters are known or assumed before the analysis of the system is done. In the analysis, different results which are important measures of system performance are derived (for example, queue length distribution). The shorthand notation that is used to specify a queueing system is also described.

To specify a queueing system, in which a server(s) takes vacations, it is required that the stochastic processes, which describe the arrival stream, the service facility and the vacation be identified.

The arrival process is described in terms of interarrival time by a probability distribution, denoted by $A(t)$ where,

$$A(t) = P[\text{time between arrivals} \leq t].$$

$P[X]$ denotes the probability of event X. In this thesis, the arrival process is assumed to be Poisson, which means that $A(t) = 1 - e^{-\lambda t}$, where λ is the mean arrival rate.

Two important aspects to describe the service process are the service time distribution and the order of service. The service process is described in terms of service time and is denoted by $B(x)$ where,

$$B(x) = P[\text{service time} \leq x].$$

When the service time is exponential, $B(x) = 1 - e^{-\mu x}$, where μ is the mean service rate and $\frac{1}{\mu}$ is the mean service time. The order in which the customers are served is also important in describing the queueing discipline. First Come First Serve(FCFS), Last Come First Serve(LCFS) and Random Order of Service(ROS) are the standard queueing disciplines. The thesis assumes exponential service time and the order FCFS.

There are three important aspects to describe the vacation process: distribution of vacation time and specifications of when the vacation starts and ends. The vacation process is described in terms of vacation time (length of time the server is on vacation) and is denoted by $V(x)$.

$$V(x) = P[\text{vacation time} \leq x]$$

If exponential, the vacation process can be described in the same way as the service process, that is, $V(x) = 1 - e^{-\theta x}$, where θ is the mean vacation rate and $\bar{v} = \frac{1}{\theta}$ is the mean vacation time of the server. In this thesis, we consider two distributions: exponential and phase.

When the vacation process follows a Phase Distribution it is described by (ν, T) and by the order of the phase distribution. The phase distribution

is a generalization of Erlang's Method of Stages[15]. A phase distribution of order m has $m+1$ stages(phases). The $m+1$ phase is the absorbing stage, and the rest of the stages are transient states. The vector ν gives the initial probability of finding the process in one of the m phases. The vacation is over when the vacation process comes into the absorbing stage. The transition rate among the transient stages is given by the $m * m$ matrix, T. The mean vacation time is given by $\bar{v} = -\nu T^{-1}\vec{e}$, where $\vec{e}$ is a column vector of size $m+1$ with all elements equal to 1. The phase distribution will be described in more detail in Chapter 4.

The time at which the server takes vacation is an important parameter to characterize vacation models. In some of the models, vacation starts only when the queue is empty. This kind of service is referred to as *exhaustive*, since all waiting customers are served. In other models, vacation starts at a random time (for example, due to server breakdown), which is independent of the state of the queue. This can be referred to as *non-exhaustive*. Finally, vacation may start after a server has served k customers and this is called *k-limited* service. Though in queueing theory the convention of exhaustive, non-exhaustive and 1-limited are attributes of service but here we have used for the vacation also since it gives time when the server goes for vacation.

To characterize a vacation, the behaviour of a server on arrival to an empty queue after vacation completion is also important. In some cases, the server waits at the queue for the customer arrival. While in other cases, the server will take another vacation and will continue doing so as long as the queue is empty when the server returns. These cases are referred to as Single(V_S) and Multiple(V_M) Vacations, respectively. In this thesis we look

at a 1-limited multiple vacation model.

To specify a queueing system, a standard shorthand notation is used, as in [Kleinrock, Vol I, Pg.viii]. For example, a single server queue in which the arrival rate is Poisson(M), service time is generalized(G) and the server takes multiple vactions, is denoted by $M/G/1/V_M$.

1.3 Problem Definition and Motivation

In this thesis, a multiple server queue, in which each server takes vacation after serving one customer is studied. During their vacation time the servers can do other assigned work, like serving other queues in the case of Polling Systems. The arrival process is Poisson and service time is exponential. The duration of a vacation is a random variable with known distribution. Two models are considered. In the first, vacation follows an exponential distribution. In the second, the vacation follows a phase distribution of order 2. In both cases, it is a multiple vacation model. The shorthand notation to specify these queues is $M/M/S/V_M$ with 1-limited service. We derive the mean and the second moment of queue length and mean waiting time for this model.

Exponential distributions for service and vacation time, and the Poisson process for arrivals is assumed because the memoryless property of the exponential distribution and the related Poisson process makes the analysis easier and also, they are accurate in some applications. By assuming a phase distribution for the vacation time in our second model we are taking a more general distribution which can be solved easily by an algorithmic method

using the techniques discussed in [23].

Multiple server vacation systems have not been studied much even though there are many applications where the results for these kind of queues can be used. Single server queues with vacations have been used to study single server cyclic queues[4], similiarly multiple server queues with vacations can be used to study multiple server cyclic queues. Cyclic queues arise in multiprocessor system[22], multiple token ring and multislotted networks[11]. In these applications, generally, only one message or task is serviced before the server moves to the next queue. Therefore, this motivates the study of M/M/S Vacation queues where one customer is serviced per visit by a server.

The study of a second model in which the vacation period is phase distributed is done because the approximate analysis of multiple server cyclic queues can be done better. The vacation of the servers of the cyclic queues can be modeled more accurately by using a phase distribution.

To summarize, the main motivating factor of studying M/M/S/V_M queues has been its applications in communication networks (Multiple Token Ring, Multiple Server Polling), and multiprocessor systems. All these systems can be treated as multiple server cyclic queues but analyzing these queues directly is very complex which is clear from the techniques used in the analysis of single server cyclic queues[28].

1.4 Thesis Overview

The thesis organization is as follows. In this chapter various applications of vacation models were discussed. The problem to be analyzed and the

motivation for doing it was also presented. The next chapter is devoted to describing the M/G/1 and M/M/S type Vacation queues. The basic queueing techniques which have been used by different researchers to analyze these queues are also explained.

In Chapter 3, the analysis of our model for exponential vacation time is done. The properties of z-transforms are used along with Balance Equations to study the model. The method proves to be incapable of handling cases where the number of servers is four or more.

In Chapter 4, a Matrix Geometric Method has been used to analyze both our models. In the first model, the vacation is exponential and in the second it follows a phase distribution of order 2. Instead of obtaining explicit results for the steady state probability and mean waiting time, an algorithm must be followed to get the results using this technique.

The analysis of the results is presented in Chapter 5. Chapter 6 summarizes the research conclusions and provides suggestions for future work.

Chapter 2

A Survey of Analytical Models for Queues with Vacations

There are numerous applications which can be analyzed using vacation models. Depending on these applications, different kinds of vacation models have been studied. In this chapter we will discuss some of these vacation models. M/G/1/V_M and M/M/S/V_M models are discussed in detail. Models with multiple vacations are discussed because these type of vacations are generally applicable in communication and multiprocessor systems and secondly, these models have been extensively studied. The different techniques that have been used to analyze these queues are outlined. The results of the analysis, for example, mean queue length and mean waiting time, are presented.

The purpose of this chapter is to present the techniques that have been used in the study of Vacation models and to summarize the results. This chapter will help in understanding the techniques used to study our models

in later chapters.

2.1 Single Server Vacation Models

Single Server Vacation models have been studied for different arrival, service and vacation characterstics (for example, Poisson or exponential(M) and general(G)). In these models, the server can take vacation at one of the following time instances: when the queue is depleted of messages(exhaustive), or the server has served k-customers(k-limited) or at some random time(non-exhaustive).

In the first part of this section, the various techniques used in the analysis of M/G/1/V_M models are presented along with the important results. In the second part, we list other single server vacation models and the techniques used for their analysis.

2.1.1 M/G/1/V_M Queues

In this section we will discuss M/G/1/V_M model with exhaustive service. The z-transform of queue length distribution($Q(z)$) and Laplace Stieltjes Transform(LST) of waiting time distribution($W^*(s)$) of the model are derived by following the different methods used by various researchers. $Q(z)$ and $W^*(s)$ are defined as follows:

$$Q(z) = \sum_{i=0}^{\infty} P[\text{Number of customers at the queue=i}]z^i$$

$$W^*(s) = \int_0^{\infty} e^{-st} w(t)dt$$

where $w(t)$ is the probability density function(pdf) of the waiting time.

Different techniques that have been used to analyze this model[5] are:

1. Embedded Markov-Chain Approach.
2. Decomposition Method.
3. Level Crossing Argument.

Embedded Markov-Chain Approach

Levy and Yechiali[18], Scholl and Kleinrock[25], Cooper[4] and Heyman[10] have all analyzed these queues using the Embedded Markov Chain Approach. This technique is very prevalant in queueing theory and has been used in the study of M/G/1 queues with no vacations. The fundamental idea behind this method is to simplify the description of states from the 2-dimensional description [N(t),$X_0(t)$], where N(t) is the number of customers in the system and $X_0(t)$ is the time for which the current customer has received service, to a one dimensional description N(t). The specification of $X_0(t)$ is required for a generalized service distribution because it does not have the memory-less characteristic like the exponential service distribution [Kleinrock Vol I, Pg.66]. Using only N(t) to describe the system implies that the time expended on service for the customer in service should be implicit. Thus, the system is examined at departure instances, where $X_0(t)$ is 0.

To analyze an M/G/1 queue with vacations, the system is studied at vacation termination and service completion instances[17, 18]. The states of the Markov chain with transitions occurring at these instances are defined

as $\{(i,j)$: $i = 0,1$; $j = 0,1,\ldots\}$, where, when $i = 0$, j is the number of customers at vacation termination and, when $i = 1$, j denotes the number of customers immediately after a service completion. If t_n is the nth transition epoch and i_n and j_n are the values of i and j at the nth transition, then we can write the following transition laws:

$$\begin{aligned}(i_{n+1}, j_{n+1}) &= (1, j_n + \epsilon - 1), && \text{if } j_n \geq 1 \\ &= (0, N^*), && \text{if } (i_n, j_n) = (1,0)\end{aligned}$$

where ϵ is the number of arrivals during a service time and N^* is the number of customers present at the end of a vacation period.

The first transition law relates to service completion and the second relates to the vacation termination. The first law says that if, at the nth transition, there are $j_n(\geq 1)$ customers and the server is present ($i_n = 1$) at the queue, then, at the $(n+1)$th transition, the number of customers will be equal to the number of customers present immediately after the nth transition minus 1 (the customer that leaves at the $(n+1)$ transition) plus the number of customers that arrive during service time of the customer (ϵ), whose service completion will occur at the $(n+1)$th transition epoch. The second law says that if on the nth transition there are no customers left in the queue then on the $(n+1)$th transition which occurs at vacation termination, the number of customers will be equal to the number of arrivals during the vacation period of the server (N^*).

The steady state transition, $\pi_{i,j} = Lim_{n\to\infty} P(i_n = i, j_n = j)$ can then be obtained from the transition laws. The z-transform of the steady state probability, $\pi_i(z) = \sum_{j=0}^{\infty} \pi_{ij} z^j$ can then be obtained.

The number of customers seen by an arbitrary service completion of a customer, is simply the random variable j conditioned on the event $i = 1$. This is equal to $Q(z) = \frac{\pi_1(z)}{\pi_i(1)}$. These are given by Eqs. 32 and 33 in [18], respectively. Thus,

$$\begin{aligned} Q(z) &= \frac{H(z)(V(z)-1)}{z-H(z)} \cdot \frac{(1-\lambda\bar{x})}{\lambda\bar{v}} \\ &= \frac{H(z)(1-\rho)(1-z)}{H(z)-z} \cdot \frac{1-V(z)}{(1-z)\lambda\bar{v}} \end{aligned}$$

where $\rho = \lambda\bar{x}$, $H(z)$ and $V(z)$ are the z-transforms of the distribution functions of the number of customer arrivals during the service time of a customer and server vacation, respectively.

The first factor on the right hand side is the z-transform of the number of customers at a service completion epoch in the standard M/G/1 queue and the second term is the z-transform of the number of arrivals during a forward recurrence time of a vacation. The forward recurrence time of a vacation is the time remaining in any random vacation at an arbitrary time instant.

In the FIFO queueing discipline, the customers left behind by a departing customer are precisely those which arrive during the departing customer's sojourn time (waiting time + service time), thus

$$\begin{aligned} Q(z) &= \int_0^\infty \sum_{j=0}^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} z^j s(t) dt \\ &= S^*(\lambda - \lambda z) \end{aligned}$$

where $s(t)$ is the pdf of sojourn time and $S^*(\lambda - \lambda z)$ is the LST of the number of arrivals in the sojourn time.

Assuming $(\lambda - \lambda z) = s$, the value of $S^*(s)$ can be derived from the

following relation:

$$S^*(s) = Q(1 - s/\lambda)$$

The sojourn time is the sum of waiting time and service time, hence,

$$S^*(s) = B^*(s)W^*(s) \tag{2.1}$$

Thus we can obtain the value of $W^*(s)$ from the following relation:

$$W^*(s) = \frac{Q(1 - s/\lambda)}{B^*(s)}$$

Decomposition Method

The expressions of $Q(z)$ and $W^*(s)$, derived in the previous section can be written as the product of some previously known queueing results. In this section, how these results are derived using a decomposition form are presented.

The Embedded Markov Chain approach showed that the number of customers present in the system at a random point in time at equilibrium is distributed as the sum of the following two independent random variables[7]:

1. The number of Poisson arrivals during a time interval which is distributed as the equilibrium forward recurrence time(residual life) of a vacation.

2. The number of customers present at a random point in time at equilibrium in the corresponding standard M/G/1 queueing system.

That is,

$$Q(z) = Q_{M/G/1}(z)V_0(z)$$

where $Q_{M/G/1}(z)$ represents the z-transform of the distribution function of the number of customers in a regular (no vacation) M/G/1 queue(Kleinrock Vol-I, Eq. 5.86):

$$Q_{M/G/1}(z) = H(z)\frac{(1-\rho)(1-z)}{H(z)-z}$$

and $V_0(z)$ represents the z-transform of the distribution function of the number of arrivals during a time interval distributed as the forward recurrence time of a vacation(Kleinrock Vol-I, Eq. 5.11)

$$V_0(z) = \frac{1-V(z)}{\bar{v}(\lambda-\lambda z)}$$

An intuitive explanation for this result is given by Furhmann[7]. He defines "primary customers" to be the customers which arrive while the server is on vacation and "secondary customers" to be those which arrive while the server is busy. He defines a "Virtual 1-busy period" to be the time interval from when the server begins serving a primary customer until the next point when the primary customer and all the secondary customers that arrived during its sojourn time have departed. Each of these virtual 1-busy periods follow the same stochastic laws as does a 1-busy period in the standard M/G/1 and each one is independent. A 1-busy period, as defined for the standard M/G/1 queue, is the duration for which the server is busy. The period starts with an arrival of a customer to an idle queue and terminates when there are no customers present[15].

Using the properties of the Poisson arrival he shows that the number of primary customers that a random(tagged) customer leaves behind is $V_0(z)$. The total number of customers left behind by the tagged customer is the sum of the number of primary customers and the number of secondary customers. Thus

$$Q(z) = V_0(z)Q_{M/G/1}(z)$$

Since the probability generating function of a virtual 1-busy period follows the same stochastic laws as a 1-busy period in the standard M/G/1 queue, the number of secondary customers that the tagged customer leaves behind will have the same distribution as does the number of customers left behind by a random departure in a standard M/G/1 queue, that is $Q_{M/G/1}(z)$.

This result is valid for any non-preemptive queueing discipline that selects customers in a manner that is independent of the service time because the distribution of the number of customers in the system is the same in all these queues.

Furhamnn also showed that for the FIFO queueing discipline,

$$S^*(s) = V_0^*(s)S_{M/G/1}^*(s)$$

where $S_{M/G/1}^*(s)$ is LST of the pdf of sojourn time in a standard M/G/1 queue and $V_0^*(s)$ is the LST of the pdf of forward recurrence time of vacation.

For a FIFO queueing discipline, the relation between $Q(z)$ and $S^*(s)$ was derived in a previous section. Using this relation the above decomposition property can be proved.

$$S^*(s) \;=\; Q(1 - s/\lambda)$$

$$= V_0^*(s)Q_{M/G/1}(1-s/\lambda)$$
$$= V_0^*(s)S_{M/G/1}^*(s)$$

Since $V_0(z) = V_0^*(\lambda - \lambda z)$ from the Eq. 5.46 in [15]. Using Eq. 2.1, we can obtain the following relation:

$$W^*(s) = V_0^*(s)W_{M/G/1}^*(s)$$

Level Crossing Approach

A level crossing technique has been used to study related vacation models. Brill and Posner in [3] showed that for a stable M/G/1 type queue, the rate $D(x)$ at which work decreases at level $x > 0$ is equal to the pdf of waiting time, $w(x)$, and should equal the rate at which work jumps from below x to above x. The work in the system at time t is defined as the time required to empty the system of all customers present at time t. It is often referred to as unfinished work[15].

A similiar argument is used by Shantikumar[26] for the vacation model. He treats the vacation period to be additional work having the same distribution as the vacation process. Then, using the result of Brill and Posner, that $w(x)$ in a stable M/G/1 queue is equal to the rate at which work exceeds level x. The level crossing result for the M/G/1/V_M model gives

$$I^1(x) = w(x) = \lambda \int_0^x w(y)(1-B(x-y))dy$$
$$+ w(0)(1-V(x)),\ (x>0)$$

where $w(0) = \frac{1-\rho}{\bar{v}}$, since vacation is treated as additional work. Using the Laplace transform, we get the same result for the LST of $w(x)$ as obtained

in the previous section:

$$W^*(s) = \frac{1 - V^*(s)}{s\bar{v}} \frac{s(1-\rho)}{s - \lambda + \lambda B^*(s)} = V_0^*(s) W_{M/G/1}^*(s)$$

2.1.2 Related Models

In this section we will briefly discuss other single server vacation models and the queueing techniques used in their analysis.

A single server, single vacation model has been studied by Levy and Yechiali using the Embedded Markov Approach[18]. They derive the LST of waiting time and the z-transform of the number of customers in the queue. Itzhak and Naor in [2] have analyzed an M/G/1 model in which the server goes for vacations under different circumstances. An M/G/1 vacation model with finite waiting time is studied by Lee[17], using an embedded Markov chain to determine the queue length. The blocking probability and waiting time distribution of the system are also studied using supplementary variables and a sample biasing technique. The study of an $M/G/1$ queueing system with vacation and non-exhaustive service has been done by Furhmann and Cooper[8] and by Ali and Neuts[1]. Their analysis revealed a 3-way decomposition of queue length distribution which is in contrast to the 2-way decomposition for exhaustive service discipline, discussed in Section 2.1.1. A probabilistic argument of this 3-way decomposition is given by Furhmann and Cooper[8].

The study of queues having non-Markovian arrival with server vacation have not been so fruitful owing to the complexity of the problem. There are very few results about the queue length distribution. Gelenbe

and Iasnogarodsk[9] and Keilson and Servi[13] have studied the G/G/1/V_M model and have derived waiting time[5].

2.2 Multiple Server Vacation Models

In this section, M/M/S queues, in which the servers take exponentially distributed vacation, after all the customers in the queue are served (exhaustive) or at some random time (non-exhaustive) are discussed. The 1-limited case has not been studied in the literature and will be presented in this thesis. The techniques used to analyze these queues and the results are presented. In the first part we consider the M/M/S/V_M model with exhaustive service and in the second part, other types of multiple server models are presented.

2.2.1 M/M/S/V_M Queues

An M/M/S queue with server vacations and exhaustive service has been analyzed by Levy and Yechiali[19], Kao and Kumar[12]. In both of these papers the servers are considered to be identical with service rate μ and the arrival is Poisson with rate λ. When the queue is depleted of messages, the servers go for an exponentially distributed vacation, with mean vacation time $= \frac{1}{\theta}$. Two techniques are used to analyze this model:

1. Balance Equation Method.

2. Matrix Geometric Approach.

Balance Equation Method

The steady state analysis using balance equations has been done for many Markovian queues, (for example, M/M/1, M/M/S, etc. [15]). Levy and Yechiali, in [19], have used a balance equation method to derive the distribution of the number of busy servers and the mean number of customers in the system(L).

The process has been formulated as a continous time Markov Chain with a state space $\{(j,i);\ j$=0,1,...$S;\ i \geq j\}$, where j denotes the number of busy servers and i the number of customers in the system. The steady state probabilities for these states is defined by $p_{j,i}$. Balance equations were written for all the states. The number of variables in the equations exceed the number of equations thus a technique, used by Mitrani and Ivi-Itzhak[21] and by Yechiali[29] to analyze other models, is employed. The approach is to define a partial generating function for the number of busy servers j, $G_j(z) = \sum_{i=j}^{\infty} p_{j,i} z^i$ and to exploit its properties. The set of simultaneous equations are obtained from the balance equations and can be written in matrix form as:

$$A(z)\vec{g(z)} = \vec{b(z)}$$

where $A(z)$ is the coefficient matrix of size $(S+1)^*(S+1)$ and $\vec{g(z)}$, $\vec{b(z)}$ are column vectors. $\vec{g(z)}$ is $[G_0(z), G_1(z), \ldots, G_S(z)]^t$ and $\vec{b(z)}$ consists of the right hand side constants.

Using Cramer's rule they obtained the following relation:

$$|A(z)|G_j(z) = |A_j(z)|, \qquad j = 0, 1, \ldots, S \tag{2.2}$$

where $|A|$ is the determinant of A. The matrix $A_j(z)$ is obtained by replacing the j th column of $A(z)$ by $b(\vec{z})$. For $G_j(z)$ to be positive, every root of $|A(z)|$ should also be a root of $|A_j(z)|$. Using this argument they derive $S-1$ relations in terms of $p_{j,\cdot}$'s, (probability that j servers are busy), by putting the $S-1$ roots (z_j's) of $|A(z)|$ into $|A_j(z)|$ and equating it to zero.

In this work they are able to find explicit formulas for the unknown probabilities $p_{j,\bullet}$'s as given below:

$$p_{j,\bullet} = \sum_{i=0}^{j-1} f_i(z_j) D_{i,j-1}(z_j) p_{i,\bullet}/h_j(z_j) \quad j = 1,2,\ldots,S-1$$

where

$$\begin{aligned} D_{i,j}(z) &= \Pi_{k=i}^{j} \frac{(S-k)\theta}{f_k(z)+h_k(z)} \\ f_k(z) &= \lambda z(1-z) - k\mu(1-z) \\ h_k(z) &= (S-k)\theta z \end{aligned}$$

This set of recursive equations, together with $\sum_{j=1}^{S} j p_{j,\cdot} = \frac{\lambda}{\mu}$, is used to find values of $p_{1,\bullet}, p_{2,\bullet}, \ldots p_{S,\bullet}$ as a function of $p_{0,\bullet}$. Then $p_{0,\cdot}$ is obtained from the normalizing equation $\sum_{j=0}^{S} p_{j,\bullet} = 1$.

The distribution of the number of busy servers, that is $G_j(z)$, is obtained from Eq. 2.2 by substituting the values of $p_{j,\cdot}$'s in $|A_j(z)|$.

The mean number of customers in the system(L) is given by

$$\begin{aligned} L &= \sum_{j=0}^{S} G_j'(1) \\ &= \frac{\rho}{1-\rho} + \frac{1}{1-\rho} \sum_{j=0}^{S-1} (1-\frac{j}{S})(\frac{\lambda - j\mu}{\theta} + j) p_{j,\cdot} \end{aligned}$$

where $\rho = \frac{\lambda}{\mu S}$.

Matrix Geometric Approach

An M/M/S/V_M model exactly like the one discussed in the previous section has been studied by Kao and Kumar [12]. They use a matrix-geometric approach for modelling the system. They derive the stationary, joint probability distribution of queue length and the number of busy servers, the waiting time distribution and the length of busy period.

The state variables are identical to those used by Levy and Yechiali [19] except they are listed in reverse order (i,j), in order to get the structure of a Quasi-Birth-Death(QBD) process in a form identical to that of Neuts [23]. Recall that i denotes the number of customers in the system and j denotes the number of busy servers. Using the elementary argument of birth-death processes they obtain the infinitesimal generator $\tilde{Q}$ for the Continuous Time Markov Chain(CTMC). $\tilde{Q}$ is an infinite matrix which gives the rate of transition among different states. The stationary probability of $\tilde{Q}$ is denoted by $\vec{\mathbf{x}} = (\vec{x_0}, \vec{x_1}, \ldots)$, where $\vec{x_k}$ is a vector of size min(S,k)+1 and each of its terms gives the stationary joint probability of k customers present at the queue and $0, 1, \ldots, k$ servers serving the queue respectively(equivalent to $p_{j,k}$ of the previous model).

The normal procedure to solve for the joint probability vector $\vec{\mathbf{x}}$ is to use $\vec{\mathbf{x}}\tilde{Q} = 0$ equations and the normalizing equation $\sum_{i=0}^{\infty} \vec{x_i}\vec{e_i} = 1$, where $\vec{e_i}$ is a column vector of size min(S,i)+1 with all elements equal to 1. The solution using the above procedure seems quite impossible due to the infinite number of unknown variables. But the stationary joint probability $\vec{\mathbf{x}}$ is "modified matrix-geometric", that is, it is of the form $\vec{x_k} = \vec{x_S}R^{k-S}$, $k \geq S$, where R

is an $(S+1)\times(S+1)$ matrix. Therefore it is only required to obtain the values of $(\vec{x_0}, \vec{x_1}, \ldots, \vec{x_S})$ by using the normal procedure and the rest can be obtained by using the "matrix-geometric" property.

The stationary queue length(L) is given by

$$\begin{aligned} L &= \sum_{i=0}^{\infty} i\vec{x_i}\vec{e_i} \\ &= \sum_{i=0}^{S-1} i\vec{x_i}\vec{e_i} + \vec{x_S}\sum_{i=0}^{\infty}(i+S)R^i\vec{e_S}. \end{aligned}$$

For this model the waiting time distribution and the busy period are also derived. Algorithms must be followed to get these results.

2.2.2 Related Models

There are few other M/M/S vacation models that have been studied. In this section we briefly discuss the technique used and the results obtained for each of them.

Levy and Yechiali[19], have studied the M/M/S/V_S queue using the same model as they used for the V_M model discussed in the Section 2.2.1. They outline methods to obtain probabilities to derive mean queue length and the stationary probabilities of finding k busy servers.

A steady state M/M/S queueing system where each server is subjected to a random breakdown of exponentially distributed duration has been studied by Mitrany and Avi-Itzhak[21] and Neuts and Lucantani[24]. The broken down server is brought back to an operative state by a repair process that starts immediately after the occurrence of the breakdown. This model can be

viewed as a preemptive priority system with two classes of customers where there are S high priority customers.

Mitrany and Avi-Itzhak have used the balance equation method similiar to the one used by Levy and Yechiali[19] to obtain the generating function of the queue size. For $S \leq 2$, they derive the explicit form but for large S a numerical method is suggested.

Neuts and Lucantoni used the Matrix Geometric Method to solve this model and they obtained an algorithm to solve the waiting time distribution and steady state probability. The state variables are the same as those used in [12], the matrix geometric model described in the previous section.

2.3 Summary

In this chapter we discussed the various vacation models studied by different researchers. The single server case has been studied in more detail as is clear from the survey presented in this chapter. The analysis of multiple server models is quite complicated due to the extra variables required to describe a state. Hence, few multiple server models have been analyzed. The techniques used in the analysis of single server and multiple server queues are also very different. While in the single server case, the techniques give explicit formulas for waiting and queue length distribution, in the multiple server case, we are not able to obtain formulae for mean waiting time by the balance equation method, even for the case in which all the processes, that is, arrival, service and vacation, are exponential. The Matrix Geometric Approach does give the distribution for waiting time and queue length. But

in this technique an algorithm must be followed to get the queue length and waiting time distribution.

In spite of the complexity of the multiserver queues, they need to be studied more as their results can be used in the study of cyclic queues with multiple servers, which arise in many communication and multiprocessors systems.

In the thesis, M/M/S/V_M, a 1-limited vacation model is considered. The 1-limited case has been studied for single server case by Ali and Neuts[1] but has not been studied for multiple servers. To analyze this multi server vacation model we have used the Balance Equation Method and Matrix Geometric Approach discussed in this chapter.

Chapter 3

Balance Equation Method

The balance equation technique can be used to study those models in which the different processes, that is arrival, departure and vacation are exponential or Poisson. Multiple server vacation models with exhaustive service have been studied using this technique as well[19, 21]. In this chapter, an M/M/S/V_M queueing system with 1-limited service is studied.

3.1 Mathematical Model

The system can be formulated as a Continuous Time Markov Chain (CTMC) with states defined as

$$\{(i,j) : i \geq j; j = 0, 1, \ldots, S\}$$

where i is the number of customers and j is the number of busy servers in the system. $i \geq j$ because the number of servers should be less than or equal to the number of customers in a V_M type model.

Let $p_{i,j}$ = P[i customers and j servers are present at the queue] and λ, μ, θ denote the arrival, departure and vacation rates, respectively. In equilibrium, that is when the system is in steady state, the rate of leaving a state is equal to the rate of entering the state. On the basis of this fundamental rule we can obtain the following Balance Equations for our model:

$$\lambda p_{0,0} = \mu p_{1,1} \tag{3.1}$$

$$(\lambda + S\theta)p_{i,0} = \lambda p_{i-1,0} + \mu p_{i+1,1}; \qquad i = 1, 2, \ldots \tag{3.2}$$

$$(\lambda + j\mu)p_{j,j} = (S - j + 1)\theta p_{j,j-1} + (j + 1)\mu p_{j+1,j+1}$$
$$j = 1, 2, \ldots, S - 1 \tag{3.3}$$

$$(S\mu + \lambda)p_{S,S} = \theta p_{S,S-1} \tag{3.4}$$

$$[\lambda + j\mu + (S - j)\theta]\, p_{i,j} = (S - j + 1)\theta p_{i,j-1} + \lambda p_{i-1,j} + (j + 1)\mu p_{i+1,j+1}$$
$$j = 1, 2, \ldots, S - 1;\ i > j \tag{3.5}$$

$$(\lambda + S\mu)p_{i,S} = \theta p_{i,S-1} + \lambda p_{i-1,S};\ i > S \tag{3.6}$$

We will explain how we obtained Eq. 3.3 and the rest can similarly be understood. The left hand side of this equation gives the rate of leaving state (j, j) which can happen by an arrival of a customer or by a departure of any one of the j servers on service completion. The right hand side gives the rate of entering state (j, j), which can happen by an arrival of a server to state $(j, j-1)$ or by the departure of any one of $j+1$ servers on service completion. Equating these two rates gives Eq. 3.3.

We define $p_{\bullet,j} = \sum_{i=j}^{\infty} p_{i,j}$, for $j = 0, 1, \ldots, S$, which is the probability that j servers are busy. Using this definition, the summation of Eq. 3.2, for

$i = 1, 2, \ldots$ and Eq. 3.1 yields:

$$S\theta p_{0,0} = S\theta p_{\bullet,0} - \mu p_{\bullet,1} \tag{3.7}$$

Summing Eq. 3.5 for $i = j + 1, \ldots$ and Eq. 3.3 we get

$$\begin{aligned}(S-j)\theta p_{j,j} + (S-j)\theta p_{\bullet,j} - (j+1)\mu p_{\bullet,j+1} \\ = [S-(j-1)]\theta p_{j-1,j-1} - [S-(j-1)]\theta p_{\bullet,j-1} + j\mu p_{\bullet,j}\end{aligned}$$

On the basis of the above equation we can inductively obtain the following set of equations:

$$\begin{aligned}(S-j)\theta p_{j,j} = (S-j)\theta p_{\bullet,j} - (j+1)\mu p_{\bullet,j+1} \\ j = 1, 2, \ldots, S-2\end{aligned} \tag{3.8}$$

Summing Eq. 3.6 for $i = S + 1, \ldots$ and Eq. 3.4 yields

$$\theta p_{S-1,S-1} = \theta p_{\bullet,S-1} - S\mu p_{\bullet,S} \tag{3.9}$$

which is similar to Eq. 3.8.

To solve for $p_{\bullet,j}$ we have $(S + 2)$ equations and they are: Eqs. 3.1, 3.7 3.8 and 3.9 along with the normalization equation:

$$\sum_{j=0}^{S} p_{\bullet,j} = 1 \tag{3.10}$$

Unfortunately, these $(S+2)$ equations have $2S+1$ unknown variables. These are the $p_{j,j}$'s and the $p_{\bullet,j}$'s. Since the number of equations is less than the number of unknowns, we have to follow a different approach to find the values of the unknowns.

3.2 Generating Function

The approach is to define a partial generating function for the number of busy servers and to exploit its properties. A similar method has been used in [19, 21, 29].

We define the generating function of $p_{i,j}$ as $G_j(z) = \sum_{i=j}^{\infty} p_{i,j} z^i$ where $(j = 0, 1, 2, \ldots S; |z| \leq 1)$. Using generating functions we will try to represent the balance equations in terms of $G_j(z)$'s and $p_{.,j}$'s. The advantage of this is that we will obtain more equations with fewer unknown variables and secondly, we can exploit the generating function properties to obtain more equations.

After multiplying Eq. 3.2 by z^i, summing over all i and adding Eq. 3.1 gives us

$$[\lambda(1-z) + S\theta]G_0(z) = S\theta p_{0,0} + \frac{\mu}{z} G_1(z)$$

Replacing $S\theta p_{0,0}$ by Eq. 3.7 yields

$$[\lambda(1-z) + S\theta]zG_0(z) - \mu G_1(z) = S\theta z p_{\bullet,0} - \mu z p_{\bullet,1} \qquad (3.11)$$

Multiplying Eq. 3.5 by z^i, summing over all i and adding Eq. 3.3 multiplied by z^j gives

$$[\lambda(1-z) + j\mu + (S-j)\theta]G_j(z) - (S-j+1)\theta G_{j-1}(z)$$
$$-\frac{(j+1)\mu}{z} G_{j+1}(z) = (S-j)\theta p_{j,j} z^j - (S-j+1)\theta p_{j-1,j-1} z^{j-1}$$

Now, substituting for $p_{j,j}$ and $p_{j-1,j-1}$ the values obtained from Eq. 3.8, gives us

$$[\lambda(1-z) + j\mu + (S-j)\theta]zG_j(z) - (S-j+1)\theta z G_{j-1}(z)$$

$$-(j+1)\mu G_{j+1}(z) = [(S-j)\theta p_{\bullet,j} - (j+1)\mu p_{\bullet,j+1}]\, z^{j+1}$$
$$-[(S-j+1)\theta p_{\bullet,j-1} - j\mu p_{\bullet,j}]\, z^{j};\ j = 1,\ldots,S-1 \qquad (3.12)$$

Multiplying Eq. 3.6 by z^i, summing over all i and adding Eq. 3.4 multiplied by z^S yields

$$(\lambda + S\mu)G_S(z) = \theta p_{S-1,S-1} z^{S-1} + \lambda z G_{S-1}(z)$$

Replacing $p_{S-1,S-1}$ by Eq. 3.9

$$[\lambda(1-z) + S\mu]\, zG_S(z) - \theta G_{S-1}(z)z = -\{\theta p_{\bullet,S-1} - S\mu p_{\bullet,S}\} z^S \qquad (3.13)$$

From Eqs. 3.11, 3.12 and 3.13 we can calculate the values of all $G_j(z)$'s if we know the values of $p_{\bullet,j}$'s. This implies that we should have $(S+1)$ equations in the $p_{\bullet,j}$ unknowns. After summing Eq. 3.12 (for all j=1 to $S-1$), Eq. 3.11, Eq. 3.13 and substituting $z = 1$, we get

$$\sum_{j=1}^{S} jG_j(1) = \frac{\lambda}{\mu} = \sum_{j=1}^{S} jp_{\bullet,j} \qquad (3.14)$$

since $G_j(1) = \sum_{i=j}^{\infty} p_{i,j}$.

Including the normalizing equation Eq. 3.10 and the above relation we have 2 equations in $p_{\bullet,j}$'s.

3.3 Busy Servers Probability

In this section we derive the probability of finding j busy servers, that is $p_{\bullet,j}$, and show how the generating function is used to derive the values of $p_{\bullet,j}$. The values are obtained for $S \leq 3$. After derivation the $p_{\bullet,j}$'s are used to find values of the queue length distribution and the mean queue length.

3.3.1 $S = 1$ and $S = 2$ Cases

For the $S = 1$ case, we can solve for $p_{\bullet,0}$ and $p_{\bullet,1}$, the only two unknowns, from Eqs. 3.10 and 3.14. The values of the probability of 0 and 1 server present at the queue are repectively:

$$\begin{aligned} p_{\bullet,0} &= 1 - \frac{\lambda}{\mu} \\ p_{\bullet,1} &= \frac{\lambda}{\mu} \end{aligned}$$

To solve for the $S = 2$ case we have Eqs. 3.1, 3.7, 3.9, 3.10 and 3.14 which have 5 unknowns: $p_{\bullet,0}$, $p_{\bullet,1}$, $p_{\bullet,2}$, $p_{0,0}$ and $p_{1,1}$. Thus we can use these 5 equations and solve for $p_{\bullet,0}$, $p_{\bullet,1}$ and $p_{\bullet,2}$. The values we obtain are:

$$\begin{aligned} p_{\bullet,0} &= \frac{2\theta(\theta+\mu)(2\mu-\lambda) - \lambda^2\theta}{2\theta(2\mu+\lambda)(\theta+\mu)} \\ p_{\bullet,1} &= \frac{2\mu\lambda\theta - \lambda^2\theta + 2\mu^2\lambda}{\mu(2\mu+\lambda)(\theta+\mu)} \\ p_{\bullet,2} &= \frac{2\lambda^2\theta + \mu\lambda^2}{2\mu(2\mu+\lambda)(\theta+\mu)} \end{aligned}$$

3.3.2 $S > 2$ Case

To solve for $S > 2$, there are $(2S + 1)$ unknowns and we already have $S + 3$ relations: Eqs. 3.14, 3.10, 3.1, 3.7, 3.8 and 3.9. We require $(S-2)$ more equations in $p_{\bullet,j}$'s to find these unknowns. We use generating function properties to derive these relations (for example, $G_j(z) > 0$ for $0 \leq z \leq 1$).

As mentioned in Chapter 2, models in [19, 21, 29], require $S - 1$ relations in $p_{\bullet,j}$s along with Eqs. 3.14 and 3.10. For our model we require $S - 2$ relations, since the equations we are using to solve for the unknowns are

different. We are using Eqs. 3.1, 3.7 and 3.8 in addition to Eqs. 3.14 and 3.10 which are used by other authors.

Let $f_k(z) = \lambda z(1-z) + k\mu z$ and $h_k(z) = (S-k)\theta z$. Eq. 3.11, 3.12 and 3.13 can be written in the form of a matrix as $A(z)\vec{g(z)} = \vec{b(z)}$ where

$$A(z) = \begin{bmatrix} f_0(z)+h_0(z) & -\mu & & \\ -h_0(z) & f_1(z)+h_1(z) & & \\ \vdots & -h_1(z) & \ddots & -S\mu \\ 0 & & & f_S(z)+h_S(z) \end{bmatrix} \quad (3.15)$$

$$\vec{g(z)} = [G_0(z), G_1(z), G_2(z), \cdots, G_S(z)]^t \quad (3.16)$$

$$\vec{b(z)} = [b_0(z), b_1(z), b_2(z), \cdots, b_S(z)]^t \quad (3.17)$$

where (3.18)

$$b_0(z) = S\theta z p_{\bullet,0} - \mu z p_{\cdot,1} \quad (3.19)$$

$$b_k(z) = \{(S-k)\theta p_{\bullet,k} - (k+1)\mu p_{\bullet,k+1}\}z^{k+1} - \{(S-k+1)\theta p_{\bullet,k-1} - k\mu p_{\bullet,k}\}z^k;\ k = 1, 2, \cdots, S-1 \quad (3.20)$$

$$b_S(z) = -\{\theta p_{\bullet,S-1} - S\mu p_{\bullet,S}\}z^S \quad (3.21)$$

Relation $A(z)\vec{g(z)} = \vec{b(z)}$ represents a set of $(S+1)$ relations.

Using Cramer's Rule

$$|A(z)|G_j(z) = |A_j(z)| \quad (3.22)$$

where $|A_j(z)|$ is obtained by replacing the jth column of $|A(z)|$ by $\vec{b(z)}$, we should be able to determine all $G_j(z)$s, if we know all the unknowns in $|A_j(z)|$, that is, the $p_{\cdot,j}$'s. Since for $0 \le z \le 1$, $G_j(z) > 0$, every z which is a root of $|A(z)|$ should also be a root of $|A_j(z)|$. For each root we have $(S+1)$

equations but all these equations are dependent, that is, each of them can be obtained by multiplying an appropriate constant to any other equation, since $\frac{A_i(z)}{A_j(z)} = \frac{G_i(z)}{G_j(z)}$. Thus, what we require is that we should have at least $S-2$ roots (z_k) of $|A(z)|$ between 0 and 1 so that we can have $(S-2)$ equations by putting $|A_j(z_k)| = 0$.

In our model, when S is even, $|A(z)|$ has $\frac{S}{2}-1$ roots between 0 and 1 and $\frac{S-1}{2}$ roots when S is odd (for Proof see Appendix A). When $S = 3$ we have a root of $A(z)$ between 0 and 1, the value of which depends on λ, μ and θ. Using the $|A_j(z)| = 0$ relationship, we can find an equation in $p_{\bullet,j}$'s. Thus, we have the required $(S-2)$ more relations to get the values of $p_{\bullet,0}$, $p_{\bullet,1}$, $p_{\bullet,2}$ and $p_{\bullet,3}$.

Unfortunately, for values of $S \geq 4$, the number of roots is not sufficient $(< (S-2))$ to derive the values of the $p_{\bullet,j}$'s.

The reason for not getting the required number of roots may be the large number of multiple roots which exist at $z = 0$.

3.4 Distribution of the Number in the Queue and Mean Queue Length

The z-transform of the distribution of the number of customers in the queue is given by

$$G(z) = \sum_{j=0}^{S} G_j(z) \tag{3.23}$$

The values of $G_j(z)$'s (distribution of the number of busy servers) can be obtained from Eq. 3.22, for $S \leq 3$.

The mean queue length can be derived by differentiating Eq. 3.23 and substituting $z = 1$.

$$L = G'(1) = \sum_{j=0}^{S} G'_j(1)$$

All the above results can be derived for $S \leq 3$ but the expressions are large and are not presented here.

3.5 Summary

In this chapter by using the Balance Equation method we derived mean queue length for $S = 1, 2, 3$ cases. For higher values of S the method fails to give results due to enough relations. In the next chapter we use a Matrix Geometric Method to obtain mean queue length and mean waiting time for any value of S.

Chapter 4

Matrix Geometric Approach

The approach followed in the previous chapter proved incapable of handling the $S \geq 4$ cases of our model. Hence, a different technique, namely the Matrix Geometric, is used to derive the joint probability of queue length and the number of servers and the mean waiting time. A review of the Matrix Geometric Method is given in Appendix C. This technique can be used to algorithmically obtain results for any value of S. We consider two types of vacation distributions in this chapter: exponential and phase distribution. It is easier to solve for the former case but the latter is more generally applicable.

Part 1: Exponential Case

4.1 Model Description

The model can be considered as a Continous Time Markov Chain(CTMC), with the number of customers and the servers present at a queue describing

a state of the system. The state space for the model is given by (i,j), similiar to that described in the previous chapter, where i denotes the number of customers and j denotes the number of servers at the queue. These variables can take on the following values $0 \leq i \leq \infty$ and $0 \leq j \leq S$. The value of j must be $\leq i$, since this is a multiple vacation model. That is, if there are i customers present at a queue then there can be 0,1,2, ... min(i,S) servers at the queue.

The rate of transition from one state to another can be written in a matrix form referred to as an infinitesimal generator.

We can define this infinitesimal generator $\tilde{Q}$ as follows:

$$\tilde{Q} = \begin{bmatrix} A_1^0 & A_2^0 & & & & & & & \\ A_0^1 & A_1^1 & A_2^1 & & & & & & \\ & A_0^2 & A_1^2 & A_2^2 & & & & & \\ & & A_0^3 & A_1^3 & A_2^3 & & & & \\ & & & & \ddots & & & & \\ & & & & A_0^{S-1} & A_1^{S-1} & A_2^{S-1} & & \\ & & & & & A_0^S & A_1^S & A_2 & \\ & & & & & & A_0 & A_1 & A_2 \\ & & & & & & & & \ddots \end{bmatrix}$$

where the elements (A's) of $\tilde{Q}$ are matrices. $\tilde{Q}$ is an infinitesimal generator and hence $A_0^i\vec{e} + A_1^i\vec{e} + A_2^i\vec{e} = 0$ and $A_0\vec{e} + A_1\vec{e} + A_2\vec{e} = 0$ are satisfied. The $\vec{e}$ is a column vectors with all elements equal to 1, its size depends on the matrices with which it is multiplied.

The row and column position of each submatrix in $\tilde{Q}$ indicate the number

of customers present at the queue before and after the transition, respectively. For example, A_0^i's row number is i and column is $i-1$, thus there are i customers before and $i-1$ customers after the transition (row and column numbers start from 0). In a similar way, the row and column of the elements within the submatrices corresponds to the number of servers at the queue. Since the number of servers cannot exceed the number of customers at a queue, the number of rows and columns possible in a submatrix is limited by the number of customers corresponding to that submatrix or by the number of servers. For example, A_0^i can have 0,1,...,i servers; while the A matrices without any superscript can have 0,1,...,S servers(here the dimension is limited by the number of servers, which cannot exceed S). Thus the A_0^i, A_1^i and A_2^i submatrices are of varying dimensions since their size is limited by the customers; while the matrices A_0, A_1 and A_2 are of the same dimensions, that is, $(S+1)\times(S+1)$, since their size is limited by the number of servers.

The submatrices with subscript 0 denote the rate at which a server leaves the queue and hence a customer as well, submatrices with subscript 1 denote the rate at which a customer arrives at the queue and submatrices with subscript 2 denote the rate at which the number of customers remain the same though there may be a change in the number of servers at the queue after the transition.

Let λ, μ, θ represent arrival, service and vacation rates for Poisson arrival and exponential service and vacation processes. On the basis of our model we can obtain the values of the submatrices. The matrices A_1^0 and A_2^0 are $-\lambda$ and $\begin{bmatrix} \lambda & 0 \end{bmatrix}$ respectively.

The matrix A_0^i is an $(i+1) \times i$ matrix for i= 1 to S and its contents are

$$A_0^i = \begin{bmatrix} 0 & & & & \\ \mu & 0 & & & \\ & 2\mu & 0 & & \\ & & & \ddots & 0 \\ & & & & i\mu \end{bmatrix}$$

It gives the rate of departure of a customer and therefore also the server.

The matrix A_2^i is of size $(i+1) \times (i+2)$, for i=1 to $S-1$, and its contents are

$$A_2^i = \begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \ddots & \\ & & & \lambda \quad 0 \end{bmatrix}$$

It gives the rate of arrival of a customer at the queue.

We can write the contents of A_1^i, where i=1 to S as

$$A_1^i = \begin{bmatrix} -(\lambda + S\theta) & S\theta & & & \\ & -(\lambda + \mu + (S-1)\theta) & (S-1)\theta & & \\ & & \ddots & \ddots & \\ & & & \ddots & (S-i+1)\theta \\ & & & & -(\lambda + i\mu) \end{bmatrix}$$

It has dimension $(i+1) \times (i+1)$. The off-diagonal elements give the rate of server arrival to the queue and the diagonal elements give the rate at

which the state remains the same and are obtained using the relationship $A_0^i\vec{e} + A_1^i\vec{e} + A_2^i\vec{e} = 0$.

When the number of customers at a queue exceeds S, the matrices A_0^i, A_1^i and A_2^i corresponds to A_0, A_1 and A_2, respectively. Their dimensions are $(S+1) \times (S+1)$.

The matrix $A_2 = \lambda I$, A_0 is

$$A_0 = \begin{bmatrix} 0 & & & & & \\ \mu & 0 & & & & \\ & 2\mu & 0 & & & \\ & & & \ddots & 0 & \\ & & & & S\mu & 0 \end{bmatrix}$$

The matrix A_1 is as follows

$$A_1 = \begin{bmatrix} -(\lambda + S\theta) & S\theta & & & & \\ & -(\lambda + \mu + (S-1)\theta) & (S-1)\theta & & & \\ & & & \ddots & \ddots & \\ & & & & \ddots & 0 \\ & & & & & -(\lambda + S\mu) \end{bmatrix}$$

4.1.1 Stability Condition

The infinitesimal generator $\tilde{Q}$ has been defined completely. Next we derive the condition for the system to reach steady state. To do this, we define the

matrix $A = A_0 + A_1 + A_2$. Thus A can be written as

$$A = \begin{bmatrix} -S\theta & S\theta & & & \\ \mu & -(\mu+(S-1)\theta) & (S-1)\theta & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \theta \\ & & & S\mu & -S\mu \end{bmatrix}$$

Matrix A is stochastic. The stationary probability vector $\Pi = (\pi_0, \pi_1, \ldots, \pi_S)$ of A can be obtained using the $\Pi A = \mathbf{0}$ relationship along with the normalizing equation $\sum_{i=0}^{S} \pi_i = 1$. On solving these equations we get $\pi_i = \binom{S}{i} \left(\frac{\theta}{\mu}\right)^i \frac{\mu^S}{(\mu+\theta)^S}$. Using the stability condition, that is, $\Pi A_2 \vec{e} < \Pi A_0 \vec{e}$ (refer to Appendix C) we get the following condition for the equilibrium of the queue and also the condition for $\tilde{Q}$ to be positive recurrent. Positive recurrent means that the sum of probabilities of coming back to the same state after any number of transitions (that is 1,2,...,∞) is 1. Due to the positive recurrence, the matrix geometric property can be applied (refer to Appendix C) for solving the system. The stability condition is

$$\mu \sum_{i=0}^{S} i \binom{S}{i} \left(\frac{\theta}{\mu}\right)^i > \lambda \left(1 + \frac{\theta}{\mu}\right)^S$$

This simplifies to,

$$\frac{S}{\lambda} > \frac{1}{\theta} + \frac{1}{\mu} \tag{4.1}$$

The above relation implies that the mean customer interarrival time should be greater than the mean time between server availability for the system to reach steady state.

4.1.2 Steady State Probabilities

We now solve the steady state joint probabilities for $\tilde{Q}$ by using the stochastic properties of infinitesimal generator $\tilde{Q}$ and the "matrix geometric" property which the steady state probabilities satisfy.

Let $\vec{\mathbf{x}} = [\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{S-1}, \vec{x}_S, \vec{x}_{S+1}, \ldots]$ be the steady state probability vector of the CTMC with generator $\tilde{Q}$. $\vec{x}_k = (x_{k,0}, x_{k,1}, \ldots, x_{k,min(S,k)})$, where $x_{k,i}$ gives the probability of k customers and i servers being present at the queue. $\vec{\mathbf{x}}$ satisfies the "modified matrix geometric" property, that is, $\vec{x}_k = \vec{x}_S R^{k-S}$, $k \geq S$. R is an $(S+1) \times (S+1)$ matrix and is the minimal non-negative solution of the quadratic equation(refer to Appendix C)

$$R^2 A_0 + R A_1 + A_2 = 0 \tag{4.2}$$

The value of R can be obtained from the above quadratic equation and the following relation:

$$R A_0 \vec{e} = A_2 \vec{e} \tag{4.3}$$

The above equation implies that the rate of transition from a state where there are $\mathbf{i}$ customers, to a state with $\mathbf{i+1}$ matches the transition rate from $\mathbf{i}$ to $\mathbf{i-1}$.

Using the simultaneous equations obtained from

$[\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{S-1}, \vec{x}_S, \vec{x}_{S+1}, \ldots]\tilde{Q} = 0$

$$\vec{x}_0 A_1^0 + \vec{x}_1 A_0^1 = \mathbf{0} \tag{4.4}$$

$$\vec{x}_{r-1} A_2^{r-1} + \vec{x}_r A_1^r + \vec{x}_{r+1} A_0^{r+1} = \mathbf{0} \text{ for } 1 \leq r \leq S-1 \tag{4.5}$$

$$\vec{x}_{S-1} A_2^{S-1} + \vec{x}_S (A_1^S + RA_0) = \mathbf{0} \tag{4.6}$$

and the normalizing equation

$$\vec{x}_0 + \vec{x}_1 \vec{e} + \ldots + \vec{x}_{S-1}\vec{e} + \vec{x}_S (I-R)^{-1}\vec{e} = 1 \tag{4.7}$$

we can solve for $[\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{S-1}, \vec{x}_S, \vec{x}_{S+1}, \ldots]$.

These steady state joint probabilities are then used to find the mean and the second moment of queue length and finally to derive the mean waiting time.

4.2 The $S = 1$ Case

For the case of 1 server, the value of R can be obtained from Eqs. 4.2 and 4.3 explicitly in term of λ, μ and θ. The values of the steady state probabilities are then obtained using the method described in the last section.

When S=1,

$A_0 = \begin{bmatrix} 0 & 0 \\ \mu & 0 \end{bmatrix}$, $A_1 = \begin{bmatrix} -(\lambda+\theta) & \theta \\ 0 & -(\mu+\lambda) \end{bmatrix}$ and $A_2 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$.

Using equations 4.3 and 4.2 we get,

$$R = \begin{bmatrix} (\mu+\lambda)\lambda\mu^{-1}\theta^{-1} & \lambda\mu^{-1} \\ \lambda^2\mu^{-1}\theta^{-1} & \lambda\mu^{-1} \end{bmatrix} \tag{4.8}$$

The equations corresponding to 4.4, 4.6 and 4.7 are

$$\begin{aligned} \vec{x}_0 A_1^0 + \vec{x}_1 A_0^1 &= 0 \\ \vec{x}_0 A_2^0 + \vec{x}_1 (A_1^1 + RA_0) &= \mathbf{0} \\ \vec{x}_0 + \vec{x}_1 (I - R)^{-1} \vec{e} &= 1 \end{aligned}$$

where $\vec{x}_0 = [x_{0,0}]$, $\vec{x}_1 = [x_{1,0}, x_{1,1}]$ and $\vec{x}_i = [x_{i,0}, x_{i,1}]$ for $i = 2, \ldots$. Using these equations we obtain

$$\begin{aligned} \vec{x}_0 &= \left[\frac{\mu\theta - \lambda\mu - \lambda\theta}{\mu\theta}\right] \\ \vec{x}_1 &= \left[\frac{(\mu+\lambda)(\mu\theta - \lambda\mu - \lambda\theta)\lambda}{\theta^2\mu^2}, \frac{(\mu\theta - \lambda\mu - \lambda\theta)\lambda}{\mu^2\theta}\right] \end{aligned}$$

We can write $p_{.,0}$ and $p_{.,1}$ that is, the probability of finding no server and a server at the queue, respectively, as

$$\begin{aligned} p_{\bullet,0} &= x_{0,0} + x_{1,0} + x_{2,0} + x_{3,0} + \ldots \\ p_{\bullet,1} &= x_{1,1} + x_{2,1} + x_{3,1} + \ldots \end{aligned}$$

Thus

$$\begin{aligned} [p_{\bullet,0}, p_{\bullet,1}] &= [x_{0,0}, 0] + \vec{x}_1 + \vec{x}_2 R + \vec{x}_3 R^2 + \ldots \\ &= [x_{0,0}, 0] + \vec{x}_1 (I - R)^{-1} \\ &= \left[\frac{\mu - \lambda}{\mu}, \frac{\lambda}{\mu}\right] \\ &= [1 - \rho, \rho] \end{aligned}$$

where ρ is the utilization.

The above result is the same as that obtained by using the Balance Equation Method in Chapter 3.

4.3 The $S \geq 1$ Case

It is difficult to obtain R for $S \geq 1$ using the same method as applied to $S = 1$ in the previous section since there are a large number of unknowns and also some of the relations are quadratic (obtained from Eq. 4.2).

As specified in Chapter 2, in [12] and [24] the matrix geometric method is used to study different types of queueing systems. In the model discussed in [12] the matrices are all upper triangular, which makes it simpler to derive R by using a simple recursive algorithm. For our model we use an approximation method to find the value of R. To find $\vec{\mathbf{x}}$ the idea in [24] is used. For their model, all matrices are of the same dimensions and are square. Hence modifications are required for our model due to the non-squareness and varying dimensions of the matrices (A_0^i, A_1^i and A_2^i) $\vec{\mathbf{x}}$ of our model.

4.3.1 Solution for R

We know that

$$R^2 A_0 + R A_1 + A_2 = 0$$

$$\Rightarrow R = -A_2 A_1^{-1} - R^2 A_0 A_1^{-1}$$

Taking the initial value of $R = \mathbf{0}$ we can iteratively solve for R and can check the accuracy of this approximation by using Equation 4.3.

The value of R will converge since $-A_1^{-1}$ and $(A_2 + R^2 A_0)$ are positive. Hence, in each iteration, the value of R will increase monotonically.

4.3.2 Solution of $\vec{x}$

To solve for $\vec{x}$, we represent each $\vec{x}_k$, where $k \leq S$ in terms of $\vec{x}_S$ and then obtain the value of $\vec{x}_S$. We can write $\vec{x}_S = \vec{x}_S I$.

Using Equation 4.6 we can write

$$\vec{x}_{S-1} = -\vec{x}_S(A_1^S + RA_0)\Delta_{S-1}(\lambda^{-1})$$

where $\Delta_{S-1}(\lambda^{-1})$ is of dimension $(S+1) \times S$ and all its elements are 0 except, where the indices are equal, the value of that element is λ^{-1}.

From Equation 4.5 we get

$$\begin{aligned} \vec{x}_{S-r} &= -(\vec{x}_{S-r+1}A_1^{S-r+1} + \vec{x}_{S-r+2}A_0^{S-r+2})\Delta_{S-r}(\lambda^{-1}) \\ &\quad \text{for r = 2 to S} \end{aligned}$$

where $\Delta_{S-r}(\lambda^{-1})$ is of dimension $(S-r+2) \times (S-r+1)$.

To represent $\vec{x}_k$ for $k \leq S$ in terms of $\vec{x}_S$, we assume

$$\vec{x}_{S-r} = \vec{x}_S C_{S-r} \qquad \text{r = 0 to } S \tag{4.9}$$

The value that C_i will take at different i is as follows:

$$\begin{aligned} C_S &= I \\ C_{S-1} &= -(A_1^S + RA_0)\Delta_{S-1}(\lambda^{-1}) \\ C_{S-r} &= -(C_{S-r+1}A_1^{S-r+1} + C_{S-r+2}A_0^{S-r+2})\Delta_{S-r}(\lambda^{-1}) \\ &\quad \text{for r = 2 to S} \end{aligned} \tag{4.10}$$

The dimension of C_{S-r} is $(S+1) \times (S-r+1)$. From the above set of equations we can recursively solve for $C_i (i = 0, 1, \ldots, S-1)$.

Using Eq. 4.9, we can represent $\vec{x}_i(i = 0, 1, \ldots, S-1)$ in terms of $\vec{x}_S$. Now using Equation 4.7 and the equations obtained from $\vec{\mathbf{x}}\tilde{Q} = \mathbf{0}$, that is, Eqs. 4.4- 4.6, we can solve for $\vec{x}_S$.

$$\vec{x}_S[\sum_{r=0}^{S-1} C_r\vec{e} + (I-R)^{-1}\vec{e}] = 1 \tag{4.11}$$

$$\vec{x}_S[C_{r-1}A_2^{r-1} + C_rA_1^r + A_0^{r+1}] = \mathbf{0} \qquad \text{for r=1 to } S-1 \tag{4.12}$$

$$\vec{x}_S[C_0A_1^0 + C_1A_0^1] = 0 \tag{4.13}$$

For each value of r we will get only one equation, thus using the $S+1$ equations we can find the $S+1$ unknowns of $\vec{x}_S$.

From $\vec{x}_S$, we can find the values of $\vec{x}_i$, for $i = 0, 1, \ldots S$, by using Eq. 4.9 and $\vec{x}_k$, for $k > S$, by using the relation $\vec{x}_k = \vec{x}_S R^{k-S}$. In this way, we can obtain the steady state joint probability vector $\vec{\mathbf{x}}$, for any value of S.

4.4 The Mean and Second Moment of the Number of Customers

The mean and second moment of the number of customers at the queue can be obtained exactly as in [12, 23, 24].

$$\begin{aligned} E[L] &= \sum_{i=1}^{S} i\vec{x}_i\vec{e} + \sum_{i=S+1}^{\infty} i\vec{x}_i\vec{e} \\ &= \sum_{i=1}^{S} i\vec{x}_i\vec{e} + \vec{x}_S R[((I-R)^{-1})^2 + S(I-R)^{-1}]\vec{e} \qquad (4.14) \\ E[L^2] &= \sum_{i=1}^{S} i^2\vec{x}_i\vec{e} + \sum_{i=S+1}^{\infty} i^2\vec{x}_i\vec{e} \end{aligned}$$

$$= \sum_{i=1}^{S} i^2 \vec{x}_i \vec{e} - \vec{x}_S[(S^2 - 2S + 1)(I - R)^{-1} + (2S - 3)((I - R)^{-1})^2 + 2((I - R)^{-1})^3 - S^2 I]\vec{e} \quad (4.15)$$

4.5 Waiting Time Analysis

To derive the waiting time distribution the waiting time of an arriving (tagged) customer is considered. States corresponding to the number of customers present in the queue {0,1,2,...} and an absorbing state { *} form the state space of the CTMC. Thus the state space of the CTMC is $\{*\} \cup \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \mathbf{S-1}, \mathbf{S}, \mathbf{S+1}, \ldots\}$. On entering the absorbing state denoted by $*$, a tagged customer starts receiving service. This happens at the arrival of a server from vacation when the customer is at the head of the line.

The transition rate matrix Q_1 for this CTMC is as follows

$$Q_1 = \begin{array}{c} \\ * \\ 0 \\ 1 \\ 2 \\ \vdots \\ S-1 \\ S \\ S+1 \\ S+2 \\ \\ \end{array} \begin{bmatrix} & & & & & & & & & & \\ 0 & 0 & \ldots & & & & & & & & \\ c_0 & D_0 & & & & & & & & & \\ c_1 & B_1 & D_1 & & & & & & & & \\ c_2 & & B_2 & D_2 & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & D_{S-1} & & & & & \\ c_S & & & & & B_S & D & & & & \\ & & & & & & A_0 & D & & & \\ & & & & & & & A_0 & D & & \\ & & & & & & & & & \ddots & \end{bmatrix}$$

Each element of the state space except $*$ represents $min(i, S) + 1$ state pairs, (i, j), corresponding to $(i, 0), (i, 1) \ldots (i, min(i, S))$, where (i, j) represents i customers and j servers at the queue.

c_i is a column vector of size $i + 1$, whose last element has value $(S - i)\theta$. This value gives the rate at which the tagged customer enters the absorbing state. The value is 0 for all states except where the number of customers present equals the number of servers serving the queue. B_i gives the rate at which the customers ahead of the tagged customer leave the queue and hence is identical to A_0^i. D_i is equal to $A_1^i + \lambda I - diag\{0, \ldots, (S - i)\theta\}$. It gives the rate at which the number of servers increases and the rate at which the customers ahead of a tagged customer remain same.

D has the same significance as D_i and is identical to $A_1 + \lambda I$. In matrix Q_1, λ, the customer arrival rate is not required since we are considering a FCFS queue and hence customers that arrive after the tagged customer do not have any impact on the analysis of waiting time. The transition rate matrix Q_1 is infinitesimal generator. Hence $c_i + B_i\vec{e} + D_i\vec{e} = 0$ and $A_0\vec{e} + D\vec{e} = 0$.

To derive the mean waiting time, we apply a method used in [12] and [24] with modifications as required by our model. The basic intent is to find the time it takes for the tagged customer to reach the absorbing state $(*)$. At steady state, the tagged customer on its arrival will see the system in state (i, j) with probability $x_{i,j}$. The tagged customer will not receive service immediately on arrival as it must wait for the customers which are ahead of it to receive service (handled by B's and A_0's). If all customers ahead of it

have received or are receiving service it must wait for a server to arrive from its vacation (handled by c_i's). Thus the tagged customer receives service only when the number of servers in the queue becomes equal to the number of customers present and then a server arrives at the queue after vacation.

Let $\mathbf{y(t)} = (\mathbf{y_*(t)}, \mathbf{y_0(t)}, \mathbf{y_1(t)}, \ldots)$, where $\mathbf{y_i(t)} = \{y_{ij}(t)\}$ is of size min(i, S) + 1 and corresponds to the probabilities of 0,1,... min(i,S) servers and i customers present at time t. $\mathbf{y_*(t)}$ is the probability that the tagged customer is in the absorbing state at time t. In our case, at time 0, $\mathbf{y(0)} = \{0, \vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots\}$, where $\vec{x}_i$s are the steady state probabilities obtained earlier. Let $w(t)$ denote the pdf of waiting time. Then $w(t) = y_*(t)$.

The tagged customer sees the system in state (i, j) with probability $y_{ij}(0)$ for $i \geq S$, the LST of the first passage time to a state (S, j') in S is given by the j'th element of the row vector $\Psi(s)$.

$$\Psi(s) = \sum_{i=S}^{\infty} \mathbf{y_i(0)}[(sI - D)^{-1} A_0]^{i-S} \tag{4.16}$$

Let $\phi_j(i, s)$ be the LST of the absorption time to state $*$ given that the process starts from state (i, j), for $0 \leq i \leq S$, $0 \leq j \leq i$. Let $\Phi(i, s)$ denote the column vector of dimension $(i + 1)$ containing $\phi_j(i, s)$. On the basis of Q_1 we can write the following relations:

$$\Phi(0, s) = (sI - D_0)^{-1} c_0 \tag{4.17}$$

$$\Phi(i + 1, s) = (sI - D_{i+1})^{-1} B_{i+1} \Phi(i, s) + (sI - D_{i+1})^{-1} c_{i+1} \tag{4.18}$$

$$0 \leq i \leq S - 1$$

The LST for the waiting time distribution is given by

$$W^*(s) = \sum_{i=0}^{S-1} \mathbf{y_i(0)} \Phi(i, s) + \Psi(s) \Phi(S, s) \tag{4.19}$$

Mean Waiting Time

The mean waiting time can be obtained from $W^*(s)$:

$$E[W] = -\sum_{i=0}^{S-1} y_i(0)\Phi'(i,0) - \Psi'(0)\vec{e} - \Psi(0)\Phi'(S,0) \qquad (4.20)$$

The first term gives the mean time to reach an absorbing state by the tagged customer if the system is in a state $\leq (S-1)$ on its arrival; the second and third terms give the time to reach the absorbing state if the system is in state $\geq S$ on the arrival of a tagged customer.

To solve for the mean waiting time we must calculate the value of each term in Equation 4.20. Differentiating and substituting $s = 0$ in Eq. 4.17 will give

$$\begin{aligned} \Phi'(0,0) &= -(-D_0)^{-1} I (-D_0)^{-1} c_0 \\ &= -(S\theta)^{-1} \end{aligned} \qquad (4.21)$$

Similarily, differentiating $\Phi(i+1,s)$ and substituting $s = 0$ in Eq. 4.18 and using the relation $B_i\vec{e} + D_i\vec{e} + c_i = 0$ gives

$$\Phi'(i+1,0) = D_{i+1}^{-1}[\vec{e} - B_{i+1}\Phi'(i,0)] \qquad (4.22)$$

Thus we can find $\Phi'(i,0)$ recursively.

The value of $\Psi(0) = \sum_{i=S}^{\infty} \vec{x}_i(0) U^{i-S}$, where $U = (-D)^{-1}A_0$, is obtained by substituting $s = 0$ in Eq. 4.16. The value of $\Psi(0)\vec{e} = 1 - \sum_{i=0}^{S-1} \vec{x}_i\vec{e}$, since $U\vec{e} = \vec{e}$ due to the relation $A_0\vec{e} + D\vec{e} = 0$. The value of $\Psi(0)\vec{e}$ can also be used, as mentioned in [12] to obtain an approximate value of $\Psi(0)$ by finite summation.

To obtain $\Psi'(0)$ we have to differentiate Eq. 4.16 and substitute $s = 0$

$$\begin{aligned}\Psi'(s) &= \sum_{k=0}^{\infty} y_{k+S}(0) \sum_{i=0}^{k-1} [(sI - D)^{-1} A_0]^i \frac{d}{ds}[(sI - D)^{-1} A_0] \\ &\quad [(sI - D)^{-1} A_0]^{k-i-1} \\ \Psi'(0) &= -\sum_{k=1}^{\infty} y_{k+S}(0) \sum_{i=0}^{k-1} U^i (-D)^{-1} U^{k-i}\end{aligned}$$

where $U = (-D)^{-1} A_0$. Using the $U\vec{e} = \vec{e}$ relationship we obtain

$$-\Psi'(0)\vec{e} = \sum_{k=1}^{\infty} \mathbf{y_{k+S}(0)} \sum_{i=0}^{k-1} U^i (-D)^{-1} \vec{e} \tag{4.23}$$

To obtain the value of $-\Psi'(0)\vec{e}$ from Eq. 4.23 we modify the method used in [12] and [24]. We define a stochastic matrix U^0 of order S by deleting the last row and column of our U matrix. We can obtain the values of vector $\vec{\mathbf{u^0}}$ by using the relations which it should satisfy $\vec{\mathbf{u^0}} U^0 = \vec{\mathbf{u^0}}$ and $\vec{\mathbf{u^0}}\vec{e} = 1$. A square matrix U_2 can be constructed in the following way

$$\begin{aligned}(U_2)_{kk'} &= u^0_{k'}, \qquad \text{for } 0 \le k \le S,\ 0 \le k' \le S-1 \\ &= 0, \qquad \text{for } k' = S.\end{aligned}$$

The following relation is satisfied owing to the property $UU_2 = U_2U = U_2$

$$\sum_{r=0}^{k-1} U^r (I - U + U_2) = I - U^r + kU_2 \tag{4.24}$$

Using this relation and the fact that $(I - U + U_2)^{-1}$ exists(see Appendix B for Proof), Eq. 4.23 can be simplified to

$$\begin{aligned}-\Psi'(0)\vec{e} &= \{\sum_{k=1}^{\infty} \mathbf{y_{k+S}(0)} - \sum_{k=1}^{\infty} \mathbf{y_{k+S}(0)} U^k \\ &\quad + \sum_{k=1}^{\infty} k\mathbf{y_{k+S}(0)} U_2\}(I - U + U_2)^{-1}(-D)^{-1}\vec{e}\end{aligned} \tag{4.25}$$

The value of $-\Psi'(0)\vec{e}$ can be calculated by substituting the following values:

$$\begin{aligned}
\sum_{k=1}^{\infty} \mathbf{y_{k+s}(0)} &= \vec{x}_S((I-R)^{-1}-I) \\
\sum_{k=1}^{\infty} \mathbf{y_{k+s}(0)}U^k &= \Psi(0)-\vec{x}_S \\
\sum_{k=1}^{\infty} k\mathbf{y_{k+s}(0)} &= \vec{x}_S((I-R)^{-2}R)
\end{aligned}$$

The second relation is obtained from Eq. 4.16 by putting $s = 0$. This gives us all the values required to solve Eq. 4.20.

4.6 Summary

Using the matrix geometric approach, we have developed a means of finding the steady state joint probability of the number of customers and servers at a queue and the mean waiting time for exponential vacation model. In the second part we derive these two performance measures for vacation having phase distribution.

Part 2: Phase Distribution Case

In this part we assume the probability distribution of vacation to be of phase type. The phase distribution is a generalization of Erlang's method of stages [15] and is well-suited for numerical computation[23]. Since it is more general but still numerically solvable, it is preferred to the exponential

distribution. The advantage of using this distribution is more accurate modelling of practical and complex distributions which usually arise in networks. Secondly, a large number of distributions are special types of phase distributions, for example an n-stage Erlangian distribution can be treated as an n order phase distribution, similiarily a 1-stage exponential distribution can be treated as a phase distribution of order 1, by properly choosing the values of parameters required to describe a phase distribution. Thus from the results of phase distribution we can derive results for other distributions as well.

4.7 Phase Distribution

A phase distribution of order m is described by an $(m+1)$ state Markov process, with infinitesimal generator [23].

$$Q = \begin{bmatrix} \mathbf{T} & T^0 \\ \mathbf{0} & 0 \end{bmatrix}$$

where the $m \times m$ matrix $\mathbf{T}$ satisfies $T_{ii} < 0$, for $i \leq m$ and $T_{ij} \geq 0$, for $i \neq j$. The elements of $\mathbf{T}$, T_{ij}, give the rate of transition from phase i to phase j. The column vector T^0 gives the rate of entering the absorption phase from the different phases. Also $\mathbf{T}\vec{e} + T^0 = 0$ since Q is an infinitesimal generator. The initial probability vector of Q is given by $(\nu\ ,\ \nu_{m+1})$ with $\nu\vec{e} + \nu_{m+1} = 1$. All the states $1, \ldots, m$ are transient, so that the absorption into state $m+1$ from any initial state is certain. In our model, this implies that the vacation time is finite. The mean vacation time is given by $\bar{v} = -\nu\mathbf{T}^{-1}\vec{e}$.

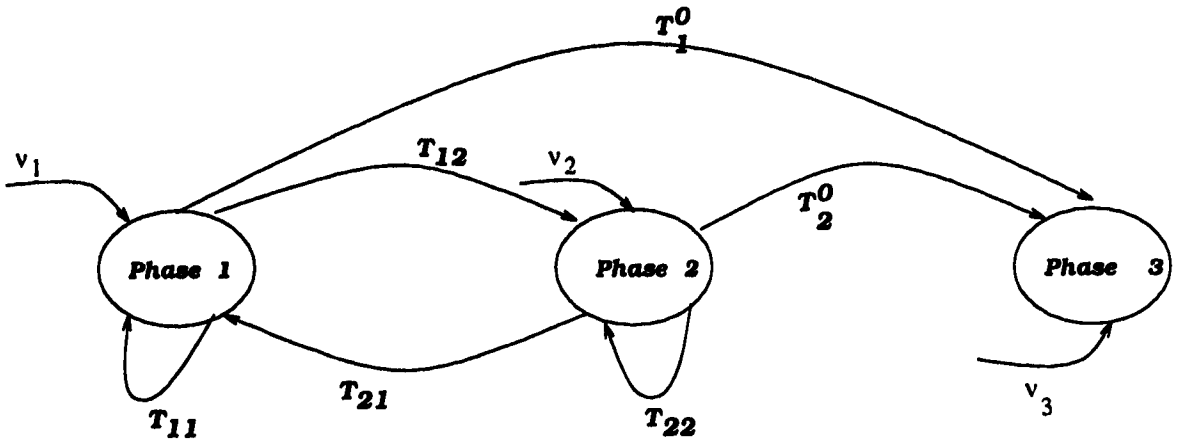

Figure 4.1: Phase Distribution of order 2

A phase distribution of order 2 is shown in Fig. 4.1 and describes the vacations of our model. A server on vacation can move between phase 1 or 2. Once it enters phase 3 (the absorbing phase), it returns to the queue and, if no unserved customer is present, it will immediately take another vacation. When a server leaves the queue (either upon service completion or if it arrives at an empty queue) it will enter either phase 1 or 2 with probability ν_1 or ν_2 respectively. In our case the value of $\nu_3 = 0$, since a server must take a vacation when it leaves a queue.

4.8 Model Description

The model can be formulated as a continous time Markov chain. The possible states are defined by $(i, j; k, l)$ where i denotes the number of customers, j denotes the number of servers at the queue, and k and l denote the number of servers in phase 1 and phase 2 of the vacation distribution respectively. These

variables can take on the following values: $0 \leq i \leq \infty$ and $0 \leq j,k,l \leq S$, such that $j+k+l = S$. Being a multiple vacation model, the $i \geq j$ condition holds. Thus the valid states corresponding to $i \leq S$ are

$$\begin{aligned}(j;k,l) \;=\; & \{(0;0,S),(0;1,S-1),(0;2,S-2),\ldots,(0;S,0);\\ & (1;0,S-1),(1;1,S-2),\ldots(1;S-1,0);\\ & (2;0,S-2),(2;1,S-3),\ldots(2;S-2,0);\\ & \vdots\\ & (i;0,S-i),(i;1,S-i-1),\ldots,(i;S-i,0)\}.\end{aligned}$$

For $i > S$, the valid states are the same as for the case of $i = S$.

As in Part 1 we can define the infinitesimal generator $\tilde{Q}$ as follows:

$$\tilde{Q} = \begin{bmatrix} A_1^0 & A_2^0 & & & & & & & \\ A_0^1 & A_1^1 & A_2^1 & & & & & & \\ & A_0^2 & A_1^2 & A_2^2 & & & & & \\ & & A_0^3 & A_1^3 & A_2^3 & & & & \\ & & & & \ddots & & & & \\ & & & & A_0^{S-1} & A_1^{S-1} & A_2^{S-1} & & \\ & & & & & A_0^S & A_1^S & A_2 & \\ & & & & & & A_0 & A_1 & A_2 \\ & & & & & & & & \ddots \end{bmatrix}$$

The matrix $\tilde{Q}$ is stochastic and hence $A_0^i\vec{e} + A_1^i\vec{e} + A_2^i\vec{e} = 0$ and $A_0\vec{e} + A_1\vec{e} + A_2\vec{e} = 0$ are satisfied.

The meaning of A_i submatrices is the same as in the previous part. The superscript of the A_i's gives the number of customers in the queue before the

transition. The entries within the submatrices however, are matrices (B_i's and E's), which are defined below. We define I_j as the identity matrix of dimension $(S-j+1) \times (S-j+1)$. The matrix $0I_j$ is a square matrix whose elements are all zeros of the same dimension as I_j. We define $0I'_j$, as a $(S-j+1)\times(S-j)$ matrix whose elements are all zero. The B_i and E matrices are used for the ease of representing the A submatrices and are referred to as subsubmatrices. The superscript of these subsubmatrices give the number of servers at the queue before transition. The positions of elements within the subsubmatrices correspond to the number of servers in phase 1 and 2 of vacation. For example, if there are j servers serving the queue then the rest of the $S-j$ servers should be in one of the two phases. For $S=3$, the structure of the different matrices has been described at the end of the section to make the explanation more clear.

B_0^j denotes the rate of entering one of the phases of vacation after service completion by a server. Thus, the transition occurs from $(i,j;k,l) \rightarrow (i-1, j-1; k', l')$ for $j = 1, \ldots S$.

$$B_0^j = \begin{bmatrix} j\mu\nu_2 & j\mu\nu_1 & & \\ & \ddots & \ddots & \\ & & j\mu\nu_2 & j\mu\nu_1 \end{bmatrix}$$

The dimensions of the matrix B_0^j are $(S-j+1) \times (S-j+2)$.

B_1^j denotes the rate of transition of servers among the two phases and the rate at which the number of servers and customers remains the same at the

queue, where $(i,j;k,l) \rightarrow (i,j;k',l')$ for $j = 0 \ldots S$. Its contents are

$$B_1^j = \begin{bmatrix} (S-j)T_{22} & (S-j)T_{21} & & \\ T_{12} & T_{11}+(S-j-1)T_{22} & \ddots & \\ & \ddots & \ddots & T_{21} \\ & & (S-j)T_{12} & (S-j)T_{11} \end{bmatrix}$$

$$- \quad (\lambda + j\mu)I_j$$

B_2^j denotes the rate of vacation termination of a server, thus $(i,j;k,l) \rightarrow (i,j+1;k',l')$, $j = 0,\ldots,S-1$.

$$B_2^j = \begin{bmatrix} (S-j)T_2^0 & & & \\ T_1^0 & (S-j-1)T_2^0 & & \\ & \ddots & \ddots & \\ & & (S-j-1)T_1^0 & T_2^0 \\ & & & (S-j)T_1^0 \end{bmatrix}$$

B_3^j denotes the rate of customer arrival, hence $(i,j;k,l) \rightarrow (i+1,j;k,l)$ for $j = 0,\ldots S$.

$$B_3^j = \begin{bmatrix} \lambda & & & \\ & \lambda & & \\ & & \ddots & \\ & & & \lambda \end{bmatrix}$$

The dimensions of B_3^j are $(S-j+1) \times (S-j+1)$.

E^i denotes the rate of transition when there are servers on vacation but there are no more customers to be served at the que. , thus $(i,j;k,l) \rightarrow (i,j,k',l')$ for $i = j = 0,\ldots,S$. In this case, a server, after completing its vacation, checks the queue, finds it empty and then goes for another vacation instantaneously.

$$E^i = \begin{bmatrix} (S-i)T_2^0\nu_2 & (S-i)T_2^0\nu_1 & & & & \\ T_1^0\nu_2 & (S-i-1)T_2^0\nu_2 + T_1^0\nu_1 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & T_2^0\nu_1 & \\ & & & (S-i)T_1^0\nu_2 & (S-i)T_1^0\nu_1 & \end{bmatrix}$$

Next, we will define the submatrices of $\tilde{Q}$. $A_1^0 = B_1^0 + E^0$ and $A_2^0 = \begin{bmatrix} B_3^0 & 0I_0' \end{bmatrix}$.

A_0^i gives the rate at which the server leaves and hence a customer leaves after service completion. The value of A_0^i, where $i = 1,\ldots S$ is as follows:

$$A_0^i = \begin{bmatrix} 0I_0 & & & \\ B_0^1 & 0I_1 & & \\ & B_0^2 & \ddots & \\ & & \ddots & 0I_{i-1} \\ & & & B_0^i \end{bmatrix}$$

The dimensions of the matrix A_0^i are $\frac{(i+1)(2S+2-i)}{2} \times \frac{i(2S+3-i)}{2}$.

The definition of A_1^i is as follows. The off-diagonal matrices of A_1^i give the rate of server arrival to the queue and the diagonal matrices give the rate

at which the number of customers and the servers at the queue remains the same though the number of servers in the two phases may change and are obtained using the relationship $A_0^i\vec{e} + A_1^i\vec{e} + A_2^i\vec{e} = 0$. The value of A_1^i, where $i = 1, \ldots S$ is as follows:

$$A_1^i = \begin{bmatrix} B_1^0 & B_2^0 & & & \\ & B_1^1 & B_2^1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & B_2^{i-1} \\ & & & & B_1^i + E^i \end{bmatrix}$$

The dimensions of the matrix A_1^i are $\frac{(i+1)(2S+2-i)}{2} \times \frac{(i+1)(2S+2-i)}{2}$.

The matrix A_2^i gives the rate of arrival of a customer. The value of A_2^i, where $i = 1, \ldots S-1$ is as follows:

$$A_2^i = \begin{bmatrix} B_3^0 & & & \\ & B_3^1 & & \\ & & \ddots & \\ & & & B_3^i \quad 0I_i \end{bmatrix}$$

The dimensions of the matrix A_2^i are $\frac{(i+1)(2S+2-i)}{2} \times \frac{(i+2)(2S+1-i)}{2}$.

When the number of customers at a queue exceeds S, the matrices A_0^i, A_1^i and A_2^i correspond to A_0, A_1 and A_2 and are defined as follows:

$$A_0 = \begin{bmatrix} 0I_0 & & & \\ B_0^1 & 0I_1 & & \\ & B_0^2 & 0I_2 & \\ & & \ddots & \ddots \\ & & & B_0^S \quad 0I_S \end{bmatrix}$$

$$A_1 = \begin{bmatrix} B_1^0 & B_2^0 & & & & \\ & B_1^1 & B_2^1 & & & \\ & & \ddots & \ddots & & \\ & & & B_1^{S-1} & B_2^{S-1} & \\ & & & & B_1^S & \end{bmatrix}$$

$$A_2 = \begin{bmatrix} B_3^0 & & & \\ & B_3^1 & & \\ & & \ddots & \\ & & & B_3^S \end{bmatrix}$$

The dimensions of A_0, A_1 and A_2 are $\frac{(S+1)(S+2)}{2} \times \frac{(S+1)(S+2)}{2}$.

An Example

For $S = 3$ the contents of $\tilde{Q}$ are

$$\tilde{Q} = \begin{bmatrix} A_1^0 & A_2^0 & & & & \\ A_0^1 & A_1^1 & A_2^1 & & & \\ & A_0^2 & A_1^2 & A_2^2 & & \\ & & A_0^3 & A_1^3 & A_2 & \\ & & & A_0 & A_1 & A_2 \\ & & & & & \ddots \end{bmatrix}$$

We define the submatrices A_0^1, A_1^1 and A_2^1 to explain the contents of $\tilde{Q}$ in a clear way. From the above explanation we know that

$$A_0^1 = \begin{bmatrix} 0I_0 \\ B_0^1 \end{bmatrix}$$

This can be expanded to the following form:

$$A_0^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mu\nu_2 & \mu\nu_1 & 0 & 0 \\ 0 & \mu\nu_2 & \mu\nu_1 & 0 \\ 0 & 0 & \mu\nu_2 & \mu\nu_1 \end{bmatrix}$$

The rows of A_0^1 corresponds to states $(j;k,l)$ =(0;0,3), (0;1,2), (0;2,1), (0;3,0), (1;0,2), (1;1,1) and (1;2,0). The columns correspond to (0;0,3), (0;1,2), (0;2,1) and (0;3,0). The terms $\mu\nu_1$ and $\mu\nu_2$ gives the rate of departure of the server from a queue and then of entering phase 1 or 2 of the vacation respectively.

The submatrix A_1^1 is

$$A_1^1 = \begin{bmatrix} B_1^0 & B_2^0 \\ 0I_1 & B_1^1 + E^1 \end{bmatrix}$$

This is equivalent to:

$$\begin{bmatrix} 3T_{22} & 3T_{21} & 0 & 0 & 3T_2^0 & 0 & 0 \\ T_{12} & T_{11}+2T_{22} & 2T_{21} & 0 & T_1^0 & 2T_2^0 & 0 \\ 0 & 2T_{12} & 2T_{11}+T_{22} & T_{21} & 0 & 2T_1^0 & T_2^0 \\ 0 & 0 & 3T_{12} & 3T_{11} & 0 & 0 & 3T_1^0 \\ 0 & 0 & 0 & 0 & 2T_{22}+2T_2^0\nu_2 & 2T_{21}+2T_2^0\nu_1 & 0 \\ 0 & 0 & 0 & 0 & T_{12}+T_1^0\nu_2 & T_{11}+T_{22}+T_2^0\nu_2+T_1^0\nu_1 & T_{21}+T_2^0\nu_1 \\ 0 & 0 & 0 & 0 & 0 & 2T_{12}+2T_1^0\nu_2 & 2T_{11}+T_1^0\nu_1 \end{bmatrix}$$

$$-\begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda+\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda+\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda+\mu \end{bmatrix}$$

The rows and columns of A_1^1 correspond to (0;0,3), (0;1,2), (0;2,1), (0;3,0), (1;0,2), (1;1,1) and (1;2,0) states.

The submatrix A_2^1 is defined as

$$A_2^1 = \begin{bmatrix} B_3^0 & & \\ & B_3^1 & 0I_1' \end{bmatrix}$$

This is equivalent to

$$\begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda & 0 & 0 \end{bmatrix}$$

The rows of A_2^1 correspond to (0;0,3), (0;1,2), (0;2,1), (0;3,0), (1;0,2), (1;1,1) and (1;2,0). The columns correspond to (0;0,3), (0;1,2), (0;2,1), (0;3,0), (1;0,2), (1;1,1), (1;2,0), (2;0,1), and (2;1,0).

Similarly all other submatrices of $\hat{Q}$ can be expanded. From the above matrices it is easy to see that the relation $A_0^1\vec{e} + A_1^1\vec{e} + A_2^1\vec{e} = 0$ is satisfied.

4.8.1 Stability Condition

As the infinitesimal generator $\hat{Q}$ has been defined completely, we now derive the condition for the system to reach steady state. To do this we define matrix $A = A_0 + A_1 + A_2$. A can be written as

$$A = \begin{bmatrix} B_1^0 + B_3^0 & B_2^0 & & & \\ B_0^1 & B_1^1 + B_3^1 & B_2^1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & B_2^{S_1} \\ & & & B_0^S & B_1^S + B_3^S \end{bmatrix}$$

The stationary probability vector $\mathbf{\Pi} = (\vec{\pi}_0, \vec{\pi}_1, \ldots \vec{\pi}_S)$ of A can be obtained using the $\mathbf{\Pi}A = 0$ relationship and the normalizing relationship, $\sum_j \vec{\pi}_j \vec{e}_j = 1$, where the vector $\vec{e}_j$ is a column vector of 1's of size $(S - j + 1)$.

Using the stability condition, that is $\mathbf{\Pi}A_2\vec{e} < \mathbf{\Pi}A_0\vec{e}$ (see Appendix C), we can verify that for the equilibrium and also for $\hat{Q}$ to be positive recurrent, the following equation should be satisfied.

$$\bar{v} + \frac{1}{\mu} < \frac{S}{\lambda} \tag{4.26}$$

The above relation means that the sum of mean vacation time and service time divided by S (equivalent to mean server availability) should be less than the mean interarrival time for the system to achieve steady state.

4.8.2 Steady State Probabilities

To derive the steady state joint probabilities of $\tilde{Q}$, we first calculate the rate matrix R. The method used in the previous model to calculate the value of R can be used here as well.

Let $\vec{\mathbf{x}} = [\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{S-1}, \vec{x}_S, \vec{x}_{S+1}, \ldots]$ be the steady state probability vector. $\vec{x}_i = (\vec{x}_{i,0}, \vec{x}_{i,1}, \ldots, \vec{x}_{i,min(S,i)})$, where $\vec{x}_{i,j} = (x_{(i,j;0,S-j)}, x_{(i,j;1,S-j-1)}, \cdots, x_{(i,j;S-j,0)})$. $x_{(i,j;k,l)}$ denotes the probability of i customers, j servers present at the queue and k and l servers present in phase 1 and 2 of the vacation distribution respectively. The value of $\vec{x}_k$, where $k \geq S$ can be obtained by the relation $\vec{x}_k = \vec{x}_S R^{k-S}$.

The boundary probabilities, that is $(\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_S)$ can be obtained from Eq. 4.12 and 4.13. In this case each value of r will give $S - r + 1$ equations. Using the normalizing equation, that is, Eq. 4.11 we will have the required number of equations to obtain the value of $\vec{x}_S$.

4.8.3 Mean and the second moment of the number of customers

The mean and the second moment of the number of customers can be be obtained from Equations 4.14 and 4.15, respectively.

4.9 Waiting Time Analysis

The method of solving the mean waiting time is the same as that used in the previous model, except that the infinitesimal rate matrix for this CTMC is

different.

We define the transition rate matrix for this CTMC as Q_2. Its contents are as follows:

$$
Q_2 = \begin{array}{c|c}
 & \begin{array}{ccccccccc} * & 0 & 1 & 2 & \cdots & S-1 & S & S+1 & S+2 \end{array} \\
\hline
\begin{array}{c} * \\ 0 \\ 1 \\ 2 \\ \vdots \\ S-1 \\ S \\ S+1 \\ S+2 \\ \\ \end{array} &
\begin{bmatrix}
0 & 0 & & & & & & & \\
g_0 & D_0 & & & & & & & \\
g_1 & F_1 & D_1 & & & & & & \\
g_2 & & F_2 & D_2 & & & & & \\
\vdots & & & \ddots & \ddots & & & & \\
g_{S-1} & & & & F_{S-1} & D_{S-1} & & & \\
g_S & & & & & F_S & D & & \\
 & & & & & & A_0 & D & \\
 & & & & & & & A_0 & D \\
 & & & & & & & & \ddots
\end{bmatrix}
\end{array}
$$

g_i is a column vector of size $\frac{(i+1)(2S+2-i)}{2}$ and is given by

$$
g_i = \begin{bmatrix}
0_{S+1} \\
0_S \\
\vdots \\
0_{S-i+1} \\
(S-i)T_2^0 \\
T_1^0 + (S-i-1)T_2^0 \\
\vdots \\
(S-i)T_1^0
\end{bmatrix}
$$

where 0_i is a column vector of size i with all elements equal to 0. g_i gives the rate at which the tagged customer enters the absorbing state. This only happens when the number of customers ahead of the tagged customer are equal to the number of servers at the queue and then a server arrives after completing its vacation.

The matrix $D_i = A_1^{i'} - \text{diag}(0_{S+1};0_S;\ldots 0_{i+1};(S-i)T_2^0, T_1+(S-i-1)T_2^0,\ldots ,(S-i)T_1^0)$, where 0_k is a sequence of k 0s. It gives the rate at which the number of servers increases and the rate at which the customers ahead of tagged customer remain same. The matrix $A_1^{i'}$ is the same as A_1^i but without the λ's and E^i. It gives the rate at which the customers ahead of the tagged customer leave the queue. The reason for not requiring λ is explained in the previous model. E^i is also not included because the analysis is done by using a tagged customer hence, at the server arrival instant, when the number of customers are equal to the number of servers, it will find the tagged customer and will not take another vacation instantaneously.

The matrix $F_i = A_0^i$ gives the rate at which customers depart from the queue. When the number of customers ahead of a tagged customer exceed S then D_i corrresponds to D, which has the same meaning as D_i. $D = A_1 + \lambda I$, here I is an identity matrix of dimensions $\frac{(S+1)(S+2)}{2} \times \frac{(S+1)(S+2)}{2}$.

A technique and argument similar to that used in the previous model will give us the value of mean waiting time.

4.10 Summary

In this part we have analyzed a model in which the vacation time of the servers follows a more generalized distribution compared to the exponential model which was discussed in Part 1. We are able to calculate the value of steady state joint probability and the mean waiting time using algorithms for this model.

4.11 Conclusion

In this chapter we have done the analysis of our model using the Matrix Geometric Method. In the first part, the vacation time is assumed to follow an exponential distribution and in the second we assumed a phase distribution. The method of calculating the steady state probabilities, mean and second moment of queue length and the mean waiting time are presented.

In the next chapter we present and analyse the results obtained from following these algorithms.

Chapter 5

Numerical Results and Their Analysis

In this chapter we present the results obtained for our model using the techniques described in Chapter 3 and 4. In the first section we discuss the effect of different parameters λ, μ and θ on the mean queue length; in the second section we discuss the effect on the waiting time. In the third section we study the effect of the different parameters at constant load on both performance measures. In the fourth section the results of the case in which vacation follows phase distribution are presented and analyzed.

5.1 Mean Queue Length

In Chapter 3, using the Balance Equation Method we derived the queue length for $S < 4$ for exponential vacation time.

S	λ	μ	θ	Queue Length (Balance)	Queue Length (Matrix Geometric)
2	1	1	1.2	8.909090913	8.908472
2	.5	.5	.55	16.41269842	16.408384
2	.3	.3	.35	10.41025641	10.407523
3	.5	.25	1	5.299186077	5.298858
3	.5	.25	.6	14.66924628	14.666427
3	1	1	1	2.2630068	2.262961
3	2	1	2.5	12.33365678	12.332067

Table 5.1: Comparison of Mean Queue Length obtained from Balance Equation and Matrix Geometric Methods

In Table 5.1, we give the mean queue length obtained from the Balance Equation and the Matrix Geometric Method for different values of λ, μ and θ. This table provides a check of the correctness of the implementation of the two methods. As is shown in the table, the results by the Balance Equation and Matrix Geometric Methods for $S = 2$ and $S = 3$ are the same.

As derived in Chapter 3 (see Appendix A) and Chapter 4, the stability condition for our model is

$$\frac{1}{\lambda} > \frac{1}{S} * (\frac{1}{\mu} + \frac{1}{\theta})$$

It implies that the mean customer interarrival time should be greater than the mean server availability. This is similiar to an M/M/1 queue where the server is always present to serve the queue and so the stability condition is $\frac{1}{\lambda} > \frac{1}{\mu}$ and we define the load as $\rho = \frac{\lambda}{\mu}(< 1)$. Using the same definition we define the load per server for our model as $\rho = \frac{\lambda}{S} * (\frac{1}{\mu} + \frac{1}{\theta})$.

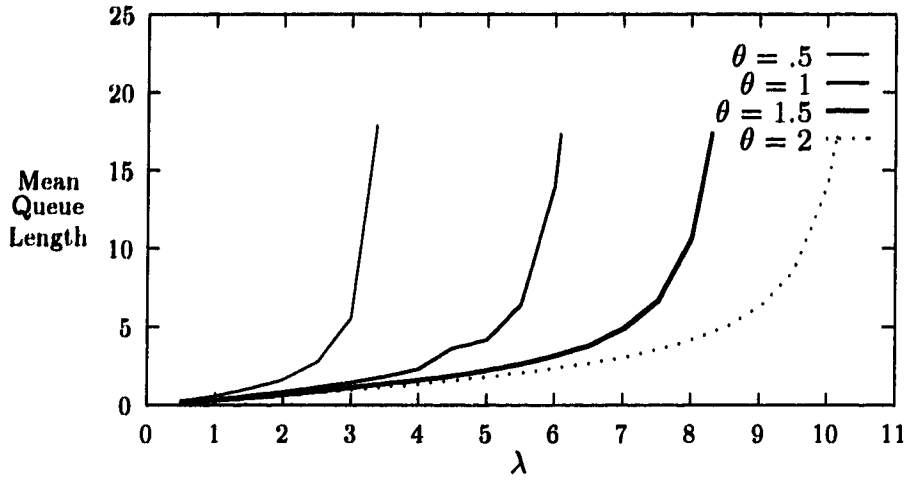

Figure 5.2: Mean Queue Length vs λ ($S = 8$, $\mu = 4$)

5.1.1 Effect of λ on Queue Length

Fig. 5.2 gives the change in queue length as λ is changed. As expected, an increase in arrival rate of customers causes an increase in queue length. When the average load per server, that is ρ, approaches 1, the queue length increases in an unbounded fashion since at this high load the servers are unable to cope with the arrival rate of customers. The value of λ when the queue length starts increasing in an unbounded fashion is larger for higher values of θ. This can be understood mathematically from the definition of the load per server. The analytical explanation is as follows. For higher values of θ, that is for low mean vacation time, the arrival rate of the system can be higher since the mean time between server availabilty is small. We observe the same effect for different values of μ.

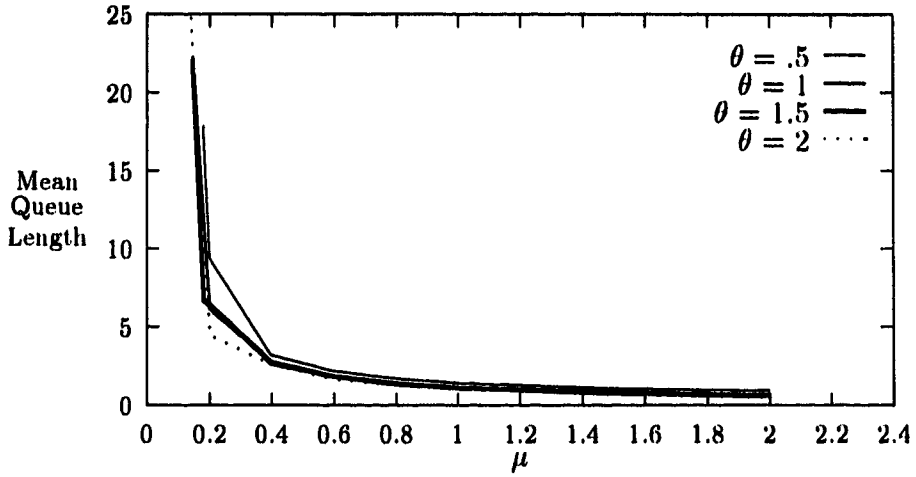

Figure 5.3: Mean Queue Length vs μ ($S = 8$, $\lambda = 1$)

5.1.2 Effect of μ on Queue Length

When the service rate, μ, increases, the mean service time decreases and hence, the queue length decreases. With the increase in μ the customers are served faster, and therefore, leave the queue faster. This is shown in Figs. 5.3 and 5.4. When the arrival rate is low (Fig. 5.3), the number of customers at the queue beyond the point of high load are fewer than in the high arrival rate case (Fig. 5.4). Hence, an increase in service rate and vacation rate does not have much effect on the queue length and the lines for different values of θ are therefore close together. At high arrival rate, the number of customers are greater and the probability that servers are busy is more. Thus increasing the vacation or service rate has an appreciable effect which is clear from Fig. 5.4. Table 5.2 shows that at low load, the servers are free

compared to the high load case and hence the increase in service or vacation rate should obviously have more effect at high load.

Number of Busy Servers	Probability($\lambda = 1$) ($\mu = 2, \theta = 1.5, \rho = .15$)	Probability($\lambda = 8$) ($\mu = 3.5, \theta = 1.5, \rho = .95$)
0	.604733	.070634
1	.305785	.216804
2	.075870	.299828
3	.012129	.241099
4	.001368	.122407
5	.000110	.040009
6	.000005	.008202
7	.000000	.000964
8	.000000	.000051

Table 5.2: Probability of Busy Servers for S=8, at $\rho = .15$ and $\rho = .95$

5.1.3 Effect of θ on Queue Length

The effect of θ on queue length is similar to that of μ. This is clear from Figs. 5.5 and 5.6. The reason for this behaviour is similar to that given in the previous section, that is, when θ is higher, vacation times are shorter making the server availability greater.

5.2 Mean Waiting Time

In this section we study the effect of different parameters on mean waiting time for exponential vacation times.

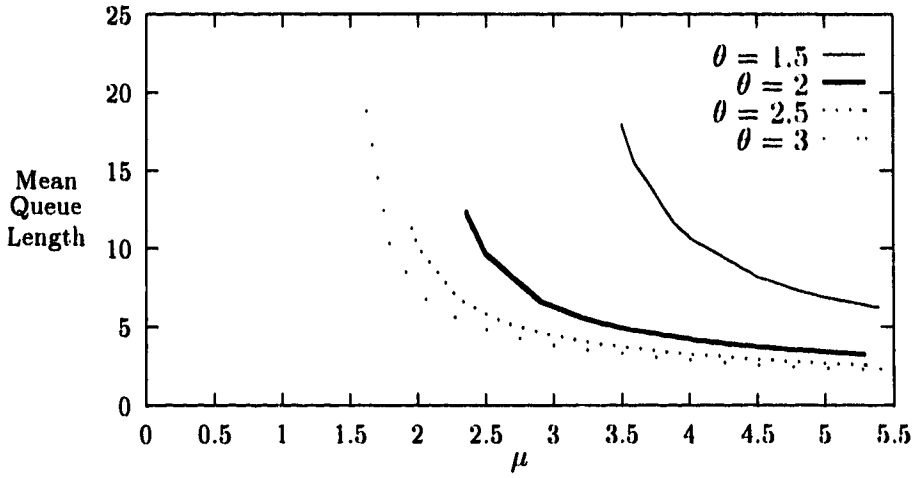

Figure 5.4: Mean Queue Length vs μ ($S = 8$, $\lambda = 8$)

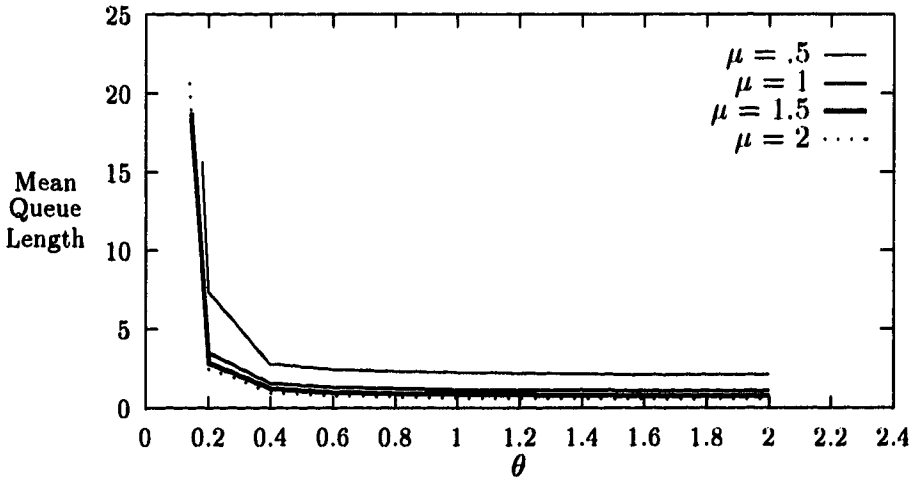

Figure 5.5: Mean Queue Length vs θ ($S = 8$, $\lambda = 1$)

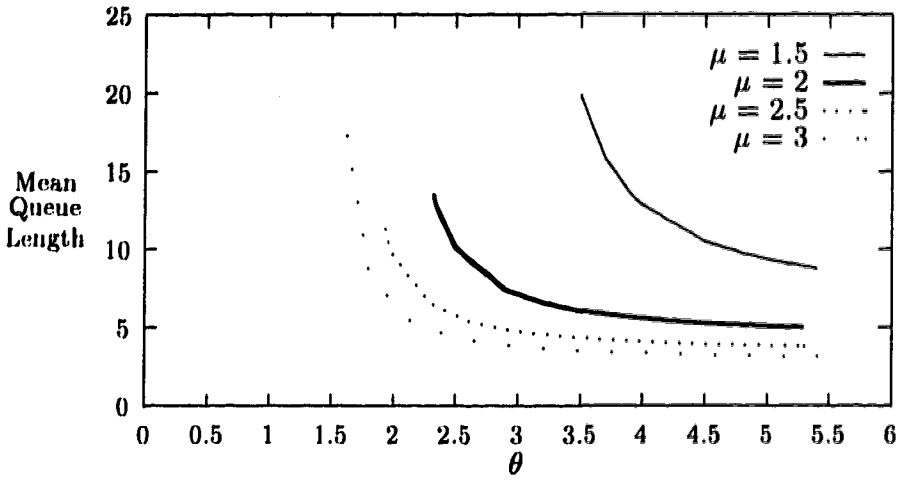

Figure 5.6: Mean Queue Length vs θ ($S = 8$, $\lambda = 8$)

5.2.1 Effect of λ on Waiting Time

In Fig. 5.7, we see that as λ increases the value of mean waiting time also increases. As $\rho \rightarrow 1$, the waiting time increases in an unbounded fashion. Higher λ means the rate of customer arrival is greater and hence an arriving customer sees on average more customers in the queue (see Fig. 5.2) and thus must wait for more time before receiving service. As shown in the figure, for higher values of θ we can accommodate a higher arrival rate without an increase in waiting time. The same behaviour occurs for higher values of μ which is not shown here. The reason is that for higher θ or μ, the mean server availability increases and thus reduces the time for which the customers must wait.

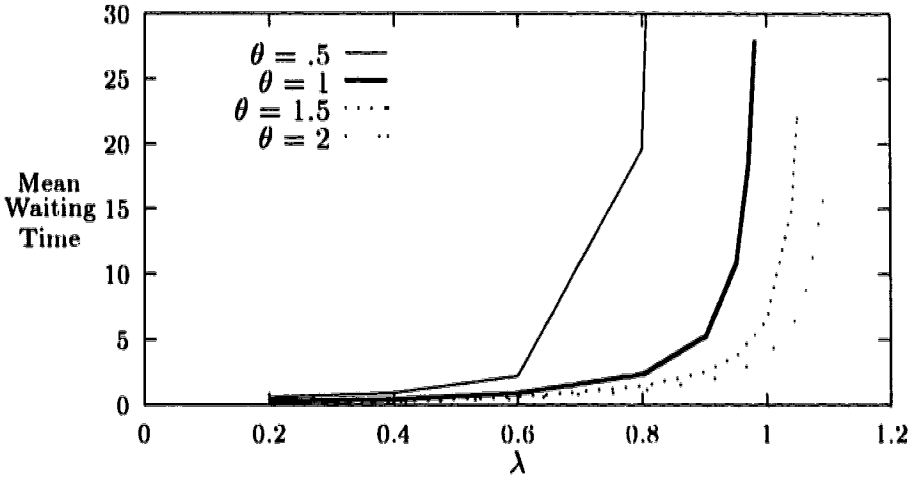

Figure 5.7: Mean Waiting Time vs λ ($S = 5$, $\mu = .25$)

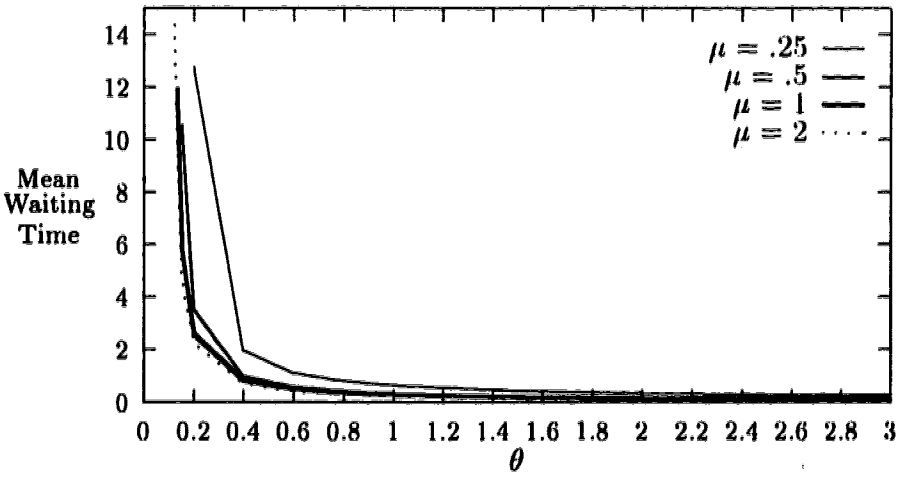

Figure 5.8: Mean Waiting Time vs θ ($S = 5$, $\lambda = .5$)

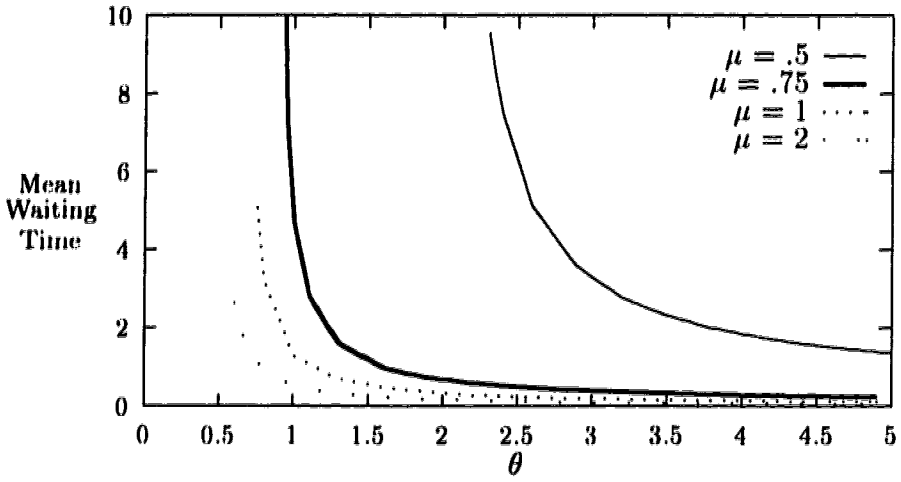

Figure 5.9: Mean Waiting Time vs θ ($S = 5$, $\lambda = 2$)

5.2.2 Effect of θ on Mean Waiting Time

When the vacation rate (θ) increases the server is available more often for service, hence, the waiting time decreases. As shown in Figs. 5.8 and 5.9 the effect of θ on waiting time is similar to its effect on queue length (shown in Figs. 5.5 and 5.6). The reason is that when we keep λ, μ and S the same the customers in the queue are served faster due to increased server availability. The effect of increasing μ has not been included here but is similar to θ.

5.3 Analysis at Constant Load

In this section we study the effect of increasing the number of servers on the mean queue length and mean waiting time when the load is kept constant.

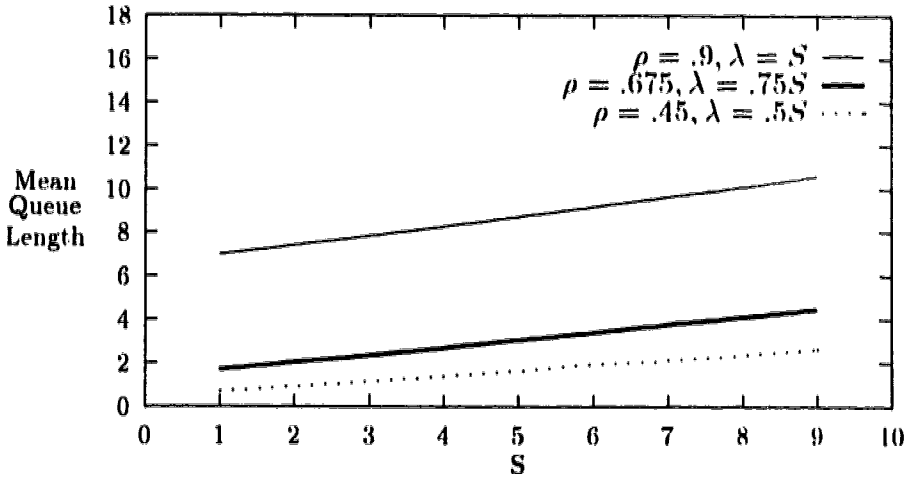

Figure 5.10: Mean Queue Length vs S ($\mu = 2$, $\theta = 2.5$)

5.3.1 Effect of S on Queue Length

In Fig. 5.10 we plot queue length against the number of servers at different loads. To keep the load constant for one particular curve we change the value of λ with S. The figure shows that, at constant load, when we increase S and λ, there is a slight increase in queue length. This is due to the increase in the arrival rate of customers which increases the number of customers. Though the number of servers are increased proportionally, the servers still must take vacations and do not serve the queue all the time. Therefore the arrival rate has more effect than the increase of servers on queue length.

In Figs. 5.11 and 5.12 we plot the queue length against S and keep the load the same by changing the vacation rate and service rate, respectively. We see that the increase in service time causes the queue length to rise for

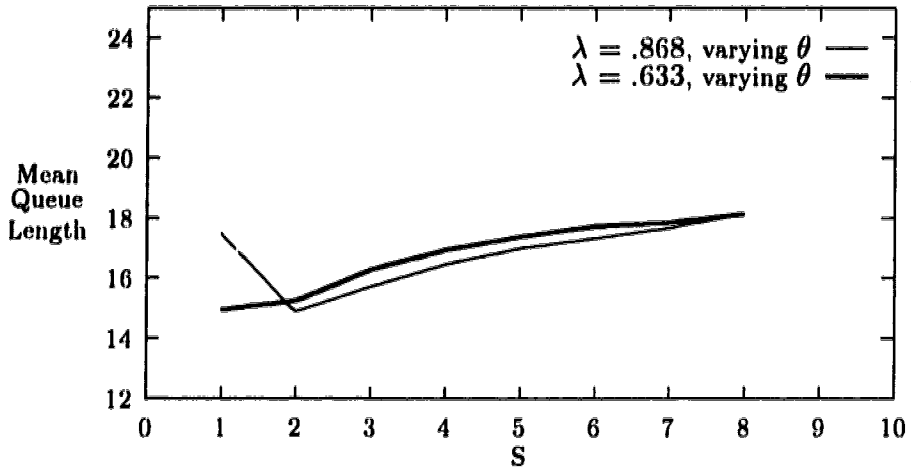

Figure 5.11: Mean Queue Length vs S, varying θ ($\rho = .95$, $\mu = 1$)

the same traffic load at a much faster rate than the increase in vacation time. This is because the queue length depends more on how fast the customers are serviced than on how fast the server comes to the queue. The customers currently being served are included in this queue length, hence, it is important to remove these from the system to reduce queue length.

As shown in Figs. 5.11 and 5.12 the mean queue length decreases and then increases again. This occurs for cases where the initial value (in this case, at $S = 1$) of θ (or μ) is very high compared to μ (or θ). In Fig. 5.11 the value of θ at $S = 1$ when $\lambda = .868$ is 10 and when $\lambda = .633$, $\theta = 2$ at $S = 1$. Thus the ratio of θ and μ is 10 and 2 respectively at $S = 1$. Choosing a high value of θ results in higher λ (.868 in this case), which a single server is not able to handle. This is reflected by the larger mean queue length at $S = 1$

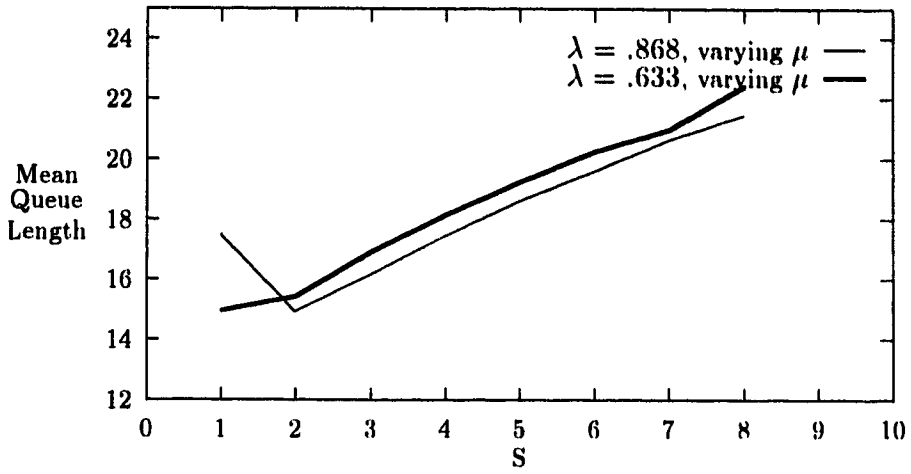

Figure 5.12: Mean Queue Length vs S, varying μ ($\rho = .95$, $\theta = 1$)

when λ is high. On increasing the number of servers to 2, the value of queue length decreases. It increases again because of the increase in vacation time.

The similiar argument applies to Fig. 5.12 to explain the decrease and then increase in mean queue length.

5.3.2 Effect of S on Waiting Time

In Fig. 5.13 we plot the mean waiting time against S where λ is varied to keep the load constant at .9 and .5. We find that with the increase in the number of servers the mean waiting time reduces considerably. This change is due to the increase in the number of servers which are able to serve the queue better even though there is a slight increase in queue length due to increased arrival rate (Fig. 5.10). The waiting time also becomes relatively

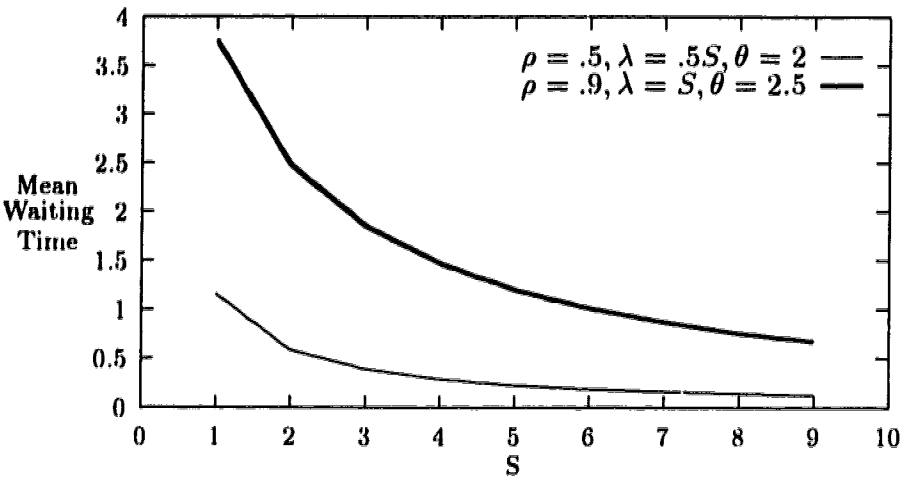

Figure 5.13: Mean Waiting Time vs S ($\mu = 2$)

constant after certain values of S. Therefore S has effect earlier on but as we increase S further there is no decrease in the waiting time.

In Figs. 5.14 and 5.15 we plot the waiting time against S for loads of .9 and .5 respectively. We notice that with an increase in service time the mean waiting time decreases and with an increase in vacation time the mean waiting time increases. The waiting time does not include the service time of the customer and hence the effect of increasing μ is not as great as the effect of increasing θ. Due to the increase in the number of servers, the waiting time decreases initially and then becomes constant. The decrease is due to the increase in number of servers. When we vary θ, we see that there is a sharp rise, this initial rise is because there is a large change in the value of θ from $S = 1$ to $S = 2$ case ($\theta = 20$ when $S = 1$,$\theta = 2.8571$ when $S = 2$ and $\theta = 1.5385$ when $S = 3$).

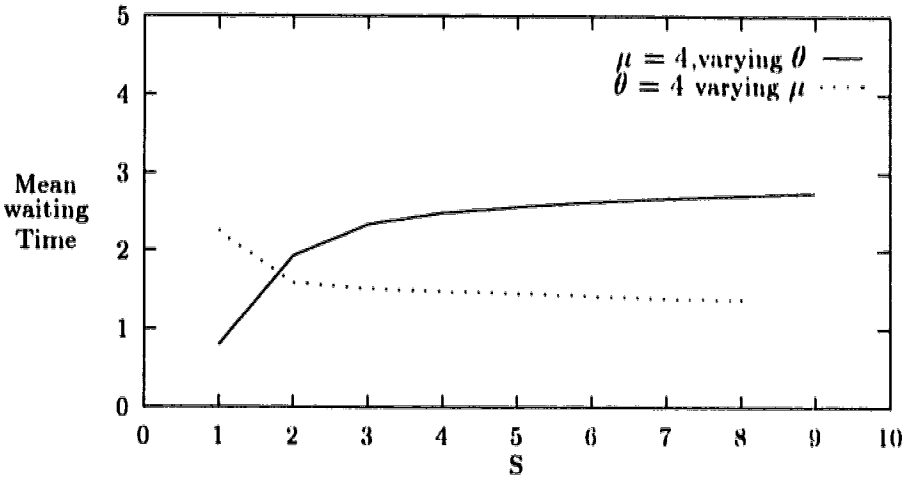

Figure 5.14: Mean Waiting Time vs S ($\rho = .9$, $\lambda = 3$)

The vacation time has more effect on the waiting time than μ has this is clear from the Figs 5.14 and 5.15. This is because the customer has to wait for the server to arrive from vacation for its waiting time to terminate. Hence the waiting depends more on how fast the server arrives at the queue from vacation.

5.4 Phase Distribution Results

In this section we present results of the case in which the vacation time follows a phase distribution. The effect of λ, μ and $\bar{v}$ are similar to the case when vacation follows an exponential distribution. The stability condition for the case of phase distribution is the same as that for exponential vacation time and hence the definition of load per server remains the same.

The steady state vector **x** satisfies

$$\vec{x_k} = \vec{x_S} R^{k-S} \qquad \text{for } k \geq S \tag{C.9}$$

This relation is referred to as Modified Matrix Geometric. The initial $S+1$ components of the steady state probability vector **x**, $(\vec{x_0},\vec{x_1},\ldots,\vec{x_S})$ can be derived by solving the $S+1$ linear equations obtained from $\mathbf{x}\tilde{Q}$ and the normalizing equation:

$$\sum_{i=0}^{S-1} \vec{x_i} + \vec{x_S}(I-R)^{-1} = 1 \tag{C.10}$$

The other $\vec{x_i}$'s can be found by using the relation C.9.

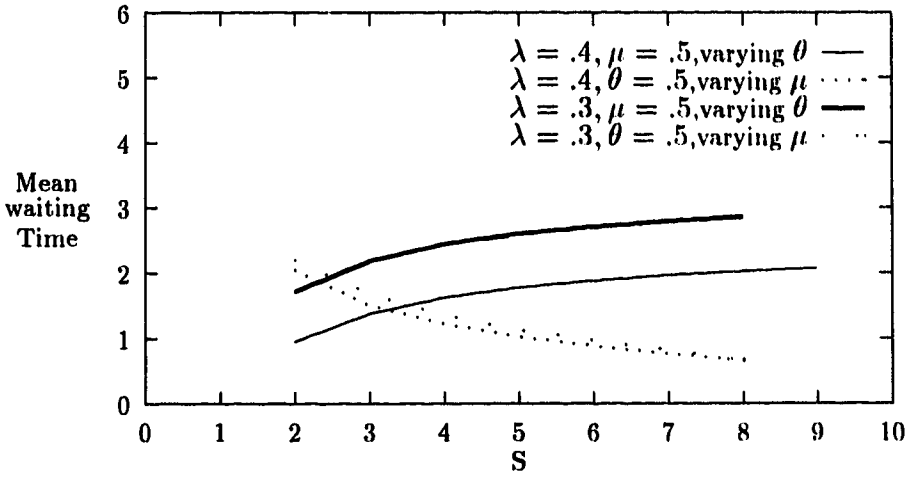

Figure 5.15: Mean Waiting Time vs S ($\rho = .5$)

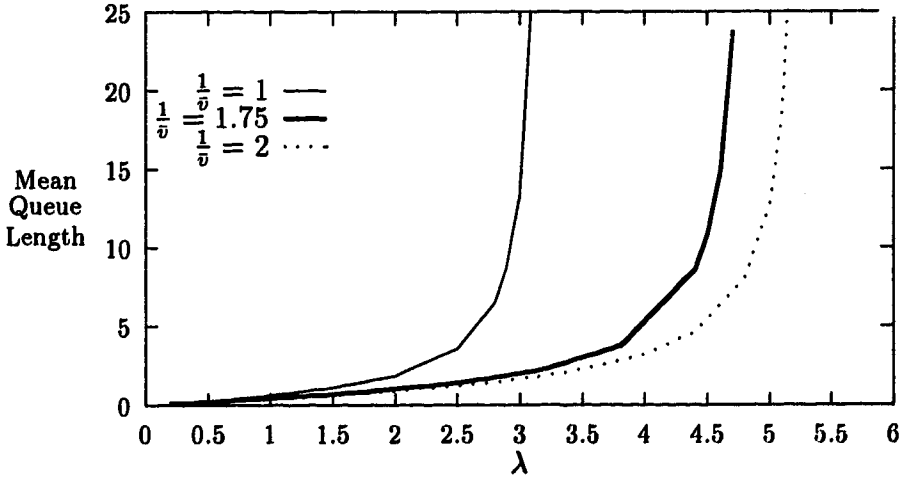

Figure 5.16: Mean Queue Length vs λ ($S = 4$, $\mu = 4$, Phase Distribution)

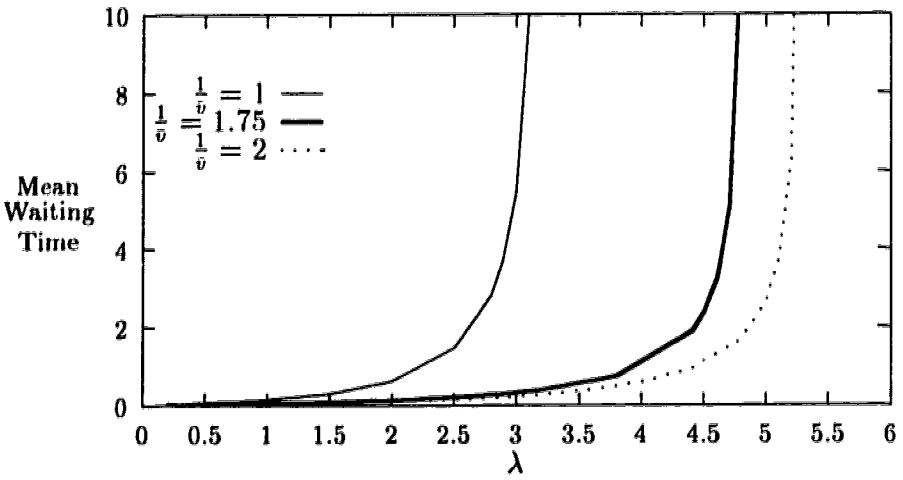

Figure 5.17: Mean Waiting Time vs λ ($S = 4$, $\mu = 4$, Phase Distribution)

As shown in Fig. 5.16 the queue length increases as λ increases in the same way as for the exponential case in Fig. 5.2. Also, in Fig. 5.17 the waiting time increases as expected, similar to Fig. 5.7.

In Table 5.3 we list the queue length and mean waiting time for the same mean vacation time but different phase distribution parameters. Two cases are considered. In the first $T_{12} = .2$ and $T_{21} = .1$ and in the second $T_{12} = .1$ and $T_{21} = .2$. The values of T_{11}, T_{22} are changed to produce the mean vacation times ($\bar{v}$) of the table for each case. We assume values of ν_1 and ν_2 to be .5 in all the cases.

For mean queue length, the results are similar given the same mean vacation time. However, for waiting time, even with the same mean vacation time, the results differ (particularly at higher loads). This shows that higher

moments, which can be obtained from $\vec{v}^n = -\nu T^{-n}\vec{e}$, have some effect on waiting time and not so much on queue length.

$\frac{1}{\nu}$	$T_{12} = .2, T_{21} = .1$		$T_{12} = .1, T_{21} = .2$	
	Queue Length	Waiting Time	Queue Length	Waiting Time
.08	6.322659	43.824955	6.342331	25.317085
.1	3.194856	11.936825	3.199655	9.367373
.16	1.844811	3.335114	1.846804	3.325833
.2	1.572331	2.232178	1.572999	2.076606
.44	1.216201	.801337	1.215032	.793002
.53	1.174503	.675197	1.179302	.683038

Table 5.3: Comparison of Mean Queue Length and Mean Waiting Time at the same first but different higher moments of vacation time($\lambda = .2, \mu = .2$)

5.5 Conclusions

In this chapter we have presented the numerical results obtained for our model. The reasons for getting these results are also analyzed. Here are some of the observations which we feel are pertinent to our model.

In the 1-limited service queues, the servers have to take a vacation after serving one customer. Due to this reason we notice that on increasing the number of servers and the arrival rate to keep the load the same, we get a slight increase in the queue length. We were expecting that in such a situation the queue size should reduce since more servers should be able to handle the system in a better way at the same load.

The waiting time is influenced more by θ than by μ since the customer

waiting for service must wait for a new server to arrive after vacation completion, and the servers at the queue for vacation after serving their respective customers. The queue length is influenced more by μ than the waiting time, since, the queue length includes customers which are being serviced.

Chapter 6

Summary and Future Work

6.1 Summary

The steady state analysis of the M/M/S/V_M queue with 1-limited service is done in this thesis. Two types of distributions are assumed for the vacation time: exponential and phase distribution of order 2. This queue has not been studied before and has applications in communication networks.

Two techniques are used to study this model. From the Balance Equation method we are able to derive explicit equations for mean queue length when $S < 4$, and the vacation time is exponentially distributed. The other technique, Matrix Geometric, gives us the mean and second moment of queue length and the mean waiting time for both exponential as well as phase distributed vacation times. These performance measures are found algorithmically. As shown in Chapter 5, the balance equation method is used to verify the implementation of the matrix geometric method.

From the study, we make the following important conclusions about this type of queue:

1. The mean queue length is dependent more on service rate than on vacation rate.

2. For mean waiting time, the vacation rate is more important than service rate.

3. By increasing the number of servers and the arrival rate to keep the load the same, we get a slight increase in the queue length. But in the same situation the waiting time decreases.

4. As expected, increasing λ results in increased mean queue length and mean waiting time; while increasing μ or θ results in decrease.

5. When vacation time is phase distributed the results are similiar to that of the exponential case. However, there are some differences in mean waiting time for the same mean vacation time when higher moments differ. The queue length does not vary much for the same mean vacation time.

6.2 Future Work

There are two possible directions in which the future work can be conducted. One may look into improving the efficiency of the algorithms and also into using these queues to model communication network applications.

6.2.1 Improve Algorithm Efficiency

The algorithms used in the Matrix Geometric technique to calculate the steady state probability, mean queue length and mean waiting time may be made more efficient if the structural properties of the matrices can be exploited.

In [12], the authors were able to exploit the structure of their matrices which were upper diagonal matrices, since the task of finding the inverse was simple. Our matrices are not so simple. We would like to have some simple algorithms to find the value of R instead of using an approximate method (see Chapter 4) but the tridiagonal nature of our matrices make the task difficult. This needs to be investigated further.

This is particularly true in the phase distribution case, as is clear from the representation of A_0's, A_1's and A_2's that have elements which are matrices. We would like to find a way of solving the different performance measures without expanding these matrices and thus deal with smaller dimension matrices.

6.2.2 Applications of these Queues

Single server cyclic queues have been studied using M/G/1 vacation queues. Similarly we can do the study of multiple server cyclic queues with 1-limited service by using the M/M/S/V_M 1-limited queue discussed in this thesis. These cyclic queues arise in many communication networks: multiple token ring, multislotted networks, mutliple server polling systems (see Chapter 1). We now give a brief description of how cyclic queues can be modelled as a

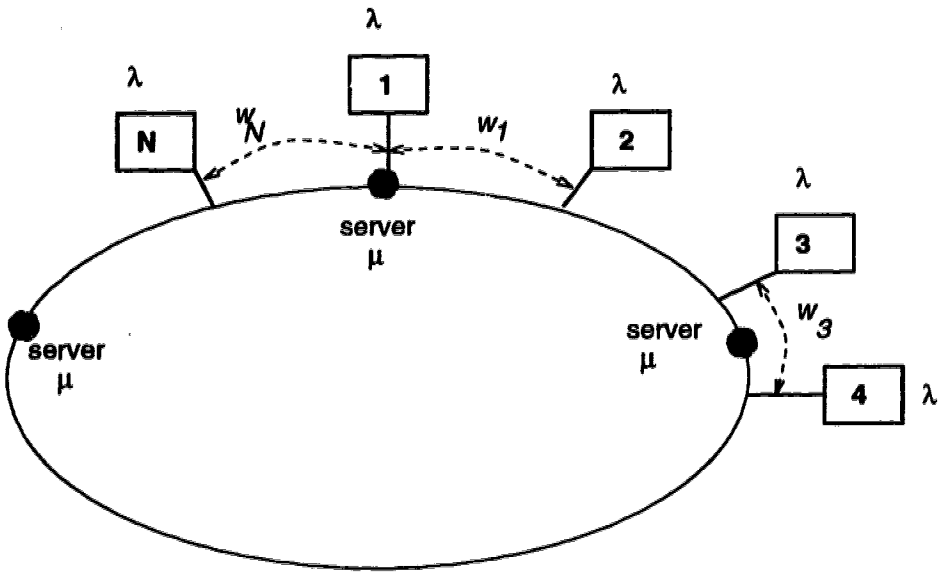

Figure 6.18: Cyclic Queue with N Stations and 3 Servers

vacation queue. In cyclic queues there are N queues which are served in cyclic order by different servers (Fig. 6.18). The time it takes for the server to move from queue i to queue $i+1$ is the switch over time. To analyze these cyclic queues as a single vacation queue we look at the performance measures at a tagged queue. If we can find the time it takes for a server to move to other queues and serve them and finally return to the tagged queue then we can find the mean waiting time and queue length at the tagged queue. But this is not a simple problem, particularly in the case of multiple servers. We consider the time between visits of the server to the tagged queue as the vacation time of the server. This includes the time it serves other queues and the walking time of the server. Here we state a simple and approximate algorithm of finding the vacation time for a symmetric system.

In a symmetric system, all queues and servers are identical and, thus service time, arrival time and walk time are the same for each queue. We model each of the these queues using an M/M/S/V_M queue

Let $s(x)$ be the probability density function (pdf) of time spent at a queue. Hence $s(x)$ = the probability that there is a customer waiting$*b(x)$, where $b(x)$ is the pdf of service time. The vacation time can be obtained as:

$$v = w_1 + s_2 + w_2 + s_3 + \ldots + w_N$$

where w_i is the walking time from queue i to queue $i+1$ and s_i is the time spent at queue i. Assuming w_i's and s_i's to be independent and identical we can write the mean vacation time $\bar{v}$ as

$$\bar{v} = N * \bar{W} + (N-1) * \sum_{i=0}^{S} (p_{.,i} - p_{i,i}) * \bar{b} \qquad (6.1)$$

where $\bar{W}$ is the mean walking time and $\bar{b}$ is service time. $\bar{v}$ is the sum of two terms, the first term gives the total walking time and the second gives the total time spent at the other $N-1$ queues.

We can solve for $\bar{v}$ and use it to get modified values of $p_{.,i}$ and $p_{i,i}$ to use in the next iteration. We must iterate until the value of $\bar{v}$ converges.

The above approximation assumes an exponential distribution and, hence, we have just used the first moments that is, the means, to find vacation rate, θ. (Eq. 6.1). In the phase distribution we will use the higher moments to define the parameters of the phase distributed vacation and thus we can model the vacation time more accurately.

6.3 Conclusions

We have found a method of finding the steady state joint probability of queue length and busy servers, the mean and the second moment of queue length and mean waiting time for the M/M/S/V_M 1-limited model. From these results we have found that for applications which can be modelled as 1-limited service models, the service rate should be kept high if it is required to have smaller queue lengths. But if the emphasis is on having a smaller waiting time then the vacation rate should be high.

The vacation time can be modelled more effectively by phase distribution since there are more parameters describing it. These parameters do make a difference in performance particularly in the waiting time case. Thus it is our belief that by assuming phase distribution for the vacation time, cyclic queues can be analyzed in a better way.

Bibliography

[1] Q. M. E. Ali and M. F. Neuts. A service system with two stages of waiting and feedback of customers. *Journal of Applied Probability*, 21:414, 1984.

[2] B. Avi-Itzhak and P. Naor. Some queueing problems with the service station subject to breakdown. *Operations Research*, 11:303–320, 1963.

[3] P. H. Brill and M. J. M. Posner. Level crossings in point processes applied to queues: Single-server case. *Operations Research*, 25:662 674, 1977.

[4] R. B. Cooper. Queues served in cyclic order: Waiting times. *Bell System Technical Journal*, 49:399–413, 1970.

[5] B. T. Doshi. Queueing systems with vacations - a survey. *Queueing Systems*, 1:29–66, 1986.

[6] R.V. Evans. Geometric distribution in some two-dimensional queueing systems. *Operations Research*, 15:830–846, 1967.

[7] S. W. Fuhrmann. A note on the M/G/1 queue with server vacations. *Operations Research*, 32:1368–1373, 1984.

[8] S. W. Fuhrmann and R. B. Cooper. Stochastic decomposition in the M/G/1 queue with generalized vacations. *Operations Research*, 33:1117–1129, 1985.

[9] E. Gelenbe and R. Iasnogorodski. A queue with server of walking type (autonomous service). *Ann. Inst. Henry Poincare*, 16:63, 1980.

[10] D. P. Heyman. The t-policy for the M/G/1 queue. *Management Science*, 23:775–778, 1977.

[11] A. E. Kamal and V. C. Hamacher. Approximate analysis of non-exhaustive multiserver polling system with application to local area networks. *Computer Network and ISDN Systems*, 17:15–27, 1989.

[12] E. P. C. Kao and K. S. Narayanan. Analyses of an M/M/N queue with servers' vacations. *European Journal of Operational Research*, 54:256–266, 1991.

[13] J. Keilson and L. Servi. Journal of applied probability. *Oscillating random walk models for GI/G/1 vacation systems with Bernoulli schedules*, 23:790–802, 1986.

[14] J. Kemeny and J.L. Snell. *Finite Markov Chains.* Van Nostrand, Princeton, N.J., 1960.

[15] L. Kleinrock. *Queueing Systems, Vol. I: Theory.* Wiley Interscience, New York, 1975.

[16] L. Kleinrock. *Queueing Systems, Vol. II: Computer Applications.* Wiley Interscience, New York, 1976.

[17] T. T. Lee. M/g/1/n queue with vacation time and exhaustive service discipline. *Operations Research*, 32:774–784, 1984.

[18] Y. Levy and U. Yechiali. Utilization of idle time in an m/G/1 queueing system. *Management Science*, 22:202–211, 1975.

[19] Y. Levy and U. Yechiali. An M/M/S queue with servers' vacations. *INFOR*, 14:153–163, 1976.

[20] L. W. Miller. Alternating priorities in multi-class queues, Ph.D. dissertation. Ithaca, New York, 1964. Cornell University.

[21] I. L. Mitrany and B. Avi-Itzhak. A many-server queue with service interruptions. *Operations Research*, 16:628–638, 1967.

[22] R. Morris and Y. Wang. Some results for multi-queue sytems with multiple cyclic servers. In *Performance of Computer-Communication Systems*, Zurich, 1984.

[23] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* John Hopkins, 1981.

[24] M. F. Neuts and D. M. Lucantoni. A Markovian queue with N servers subjected to breakdowns and repairs. *Management Science*, 25:849–861, 1979.

[25] M. Scholl and L. Kleinrock. On the M/G/1 queue with rest periods and certain service-independent queueing disciplines. *Operations Research*, 31:705–719, 1983.

[26] J. Shanthikumar. Analysis of priority queues with server control. *OPSEARCH*, 27:183–192, 1984.

[27] M. Spivak. *Calculus*. Publish or Perish, Inc. CA, 1982.

[28] H. Takagi. *Analysis of Polling Systems*. MIT Press, 1985.

[29] U. Yechiali. A queueing-type birth-and-death process defined on a continuous-time Markov chain. *Operations Research*, 21:604–609, 1973.

Appendix A

The Roots of $|A(z)|$

The fundamental mathematical theorm used for finding the roots is as follows:

> If f is continous on [a,b] and $f(a) < 0 < f(b)$, then there is some x in [a,b] such that $f(x) = 0$[27].

The proof uses the technique of [21] but is more complex since the method must be divided into even and odd cases.

Let $M_k(z) = f_k(z) + h_k(z) = z\{\lambda(1-z) + k\mu + (S-k)\theta\}$, where $f_k(z) = \lambda z(1-z) + k\mu z$ and $h_k(z) = (S-k)\theta z$.

Let

$$\begin{aligned} Q_0(z) &= 1 \\ Q_1(z) &= \frac{1}{z} M_s(z) \\ Q_2(z) &= \frac{1}{z} \begin{vmatrix} M_{S-1}(z) & -S\mu \\ -h_{S-1}(z) & M_S(z) \end{vmatrix} \end{aligned}$$

$$\vdots$$

$$Q_S(z) = \frac{1}{z^*}\begin{vmatrix} M_1(z) & -2\mu & & \\ -h_1(z) & \ddots & & \\ & & \ddots & -S\mu \\ & & -\theta z & M_S(z) \end{vmatrix}$$

$$(1-z)Q_{S+1}(z) = \frac{1}{z^*}\begin{vmatrix} M_0(z) & -\mu & & \\ -h_0(z) & \ddots & & \\ & & \ddots & -S\mu \\ & & -\theta z & M_S(z) \end{vmatrix}$$

where the value of $(*)$ is the number of roots of $|A(z)|$ at $z = 0$ and will be determined later. We can write $|A(z)| = (1-z)z^* Q_{S+1}(z)$.

We can write $Q_i(z)$ in the form of polynomials as follows:

$$Q_0(z) = 1 \tag{A.1}$$

$$Q_{2k-1}(z) = \frac{1}{z^k}[M_{S-2k+2}(z)z^{k-1}Q_{2k-2}(z) - (S-2k+3)(2k-2)\theta\mu z^k Q_{2k-3}(z)] \tag{A.2}$$

$$Q_{2k}(z) = \frac{1}{z^k}[M_{S-2k+1}(z)z^k Q_{2k-1}(z) - (S-2k+2)(2k-1)\theta\mu z^k Q_{2k-2}(z)]$$
$$\text{where } k = 1, 2, \ldots \lfloor \tfrac{S}{2} \rfloor \tag{A.3}$$

$$(1-z)Q_{S+1}(z) = \frac{1}{z^{\frac{S}{2}+1}}[M_0(z)z^{\frac{S}{2}}Q_S(z) - S\theta\mu z^{\frac{S}{2}+1}Q_{S-1}(z)] \text{ where } S \text{ is even} \tag{A.4}$$

$$Q_S(z) = \frac{1}{z^{\frac{S+1}{2}}}[M_1(z)z^{\frac{S-1}{2}}Q_{S-1}(z) - 2(S-1)\theta\mu z^{\frac{S+1}{2}}Q_{S-2}(z)]$$
$$\text{where } S \text{ is odd} \tag{A.5}$$

$$(1-z)Q_{S+1}(z) = \frac{1}{z^{\frac{S+1}{2}}}[M_0(z)z^{\frac{S+1}{2}}Q_S(z) - S\theta\mu z^{\frac{S+1}{2}}Q_{S-1}(z)] \text{ where } S \text{ is odd} \tag{A.6}$$

A.1 Proof of Polynomials

For the first few values of k we can show that the equations are correct by direct calculation. Now using induction we can prove the rest.

Let $A_i(z)$ be the determinant of the matrix formed by taking the i rows and i columns from the bottom right corner of $A(z)$.

We assume that for $i = 2k-1$ and $i = 2k$ the equations are correct, that is,

$$\begin{aligned} Q_{2k-1}(z) &= \frac{1}{z^k} A_{2k-1}(z) \\ &= \frac{1}{z^k}[M_{S-2k+2}(z) z^{k-1} Q_{2k-2}(z) - \\ &\quad (S-2k+3)(2k-2)\theta\mu z^k Q_{2k-3}(z)] \qquad \text{(A.7)} \\ Q_{2k}(z) &= \frac{1}{z^k} A_{2k}(z) \\ &= \frac{1}{z^k}[M_{S-2k+1}(z) z^k Q_{2k-1}(z) - \\ &\quad (S-2k+2)(2k-1)\theta\mu z^k Q_{2k-2}(z)] \qquad \text{(A.8)} \end{aligned}$$

We can write the value of determinant $A_{2(k+1)-1}$ from the corresponding matrix as follows

$$A_{2(k+1)-1}(z) = M_{S-2k}(z) A_{2k}(z) - (S-2k+1) h_{S-2k}(z) A_{2k-1}(z) \qquad \text{(A.9)}$$

Using A.7 and A.8 we get the following:

$$A_{2(k+1)-1}(z) = z^{k+1}[N_{S-2k}(z) Q_{2k}(z) - (S-2k+1) 2k\theta\mu Q_{2k-1}(z)] \qquad \text{(A.10)}$$

In the above equation, $N_{S-2k}(z) = z^{-1} M_{S-2k}(z)$. Dividing the above equation by z^{k+1} we get:

$$Q_{2(k+1)-1}(z) = \frac{1}{z^{k+1}} A_{2(k+1)-1}(z)$$

$$= \frac{1}{z^{k+1}}[M_{S-2k}(z)z^k Q_{2k}(z) - (S-2k+1)2k\theta\mu z^{k+1}Q_{2k-1}(z)] \qquad (A.11)$$

as required.

Similarily we can prove the $i = 2(k+1)$ case.

A.2 Properties of $Q_i(z)$

From the $Q_i(z)$ relationships we can obtain the following properties:

1. $Q_i(z)$ and $Q_{i+1}(z)$, i=1,...,S do not have any joint roots in $(0,\infty)$. If we assume that this statement is false then using A.2 and A.3, the equation A.1 can be proved false.

2. $Q_{i-1}(z_0)$ is opposite in sign to $Q_{i+1}(z_0)$ if $Q_i(z_0) = 0$. If $Q_S(z_0) = 0$ then $Q_{S+1}(z_0)$ and $Q_{S-1}(z_0)$ are opposite in sign when $z_0 < 1$ and similiarly if $z_0 > 1$. This follows directly from the relationships.

3. $Q_i(1) > 0$, $i = 1, 2, \ldots, S$.
   $Q_{S+1}(1) = \frac{1}{(1-z)}|A(z)|_{z=1} = -S!(\mu+\theta)^{S-1}[S\mu\theta - \lambda(\mu+\theta)] < 0$.
   Remark: For the system to reach steady state $[S\mu\theta - \lambda(\mu+\theta)] > 0$. This is also verified using the Matrix Geometric Method (Chapter 4).

4. Sign$[Q_0(0)]$ = +ve
   Sign$[Q_{2k-1}(0)]$=$(-1)^{k-1}$ and Sign$[Q_{2k}(0)]$=$(-1)^k$, $k = 1, 2, \ldots, \lfloor\frac{S}{2}\rfloor$.
   This property can be proved by induction. Using the equation A.4 we can verify that Sign$[Q_{S+1}(0)]$=Sign$[Q_S(0)]$ if S is even and using A.6 we can show that Sign$[Q_{S+1}(0)]$=-Sign$[Q_S(0)]$ if S is odd.

5. Sign$[Q_{S+1}(\infty)] = (-1)^S$ since the largest term is $\frac{(-z^2)^{S+1}}{z^S(1-z)}$.

A.3 Location of Roots of $|A(z)|$

$Q_1(z)$ is of degree 1 and should have 1 root only. Since $Q_1(0) > 0$, $Q_1(1) > 0$ and $Q_1(\infty) < 0$, using properties (4), (3) and (5), it implies that the root is between 1 and ∞. Let the root be $z_{1,1}$.

$Q_2(z)$ has degree 3 and should have 3 roots. $Q_2(0) < 0$, $Q_2(1) > 0$, $Q_2(z_{1,1}) < 0$, $Q_2(\infty) > 0$, using properties (4), (3), (2) and (5). Thus each of the intervals, that is (0,1), (1,$z_{1,1}$) and ($z_{1,1}$,∞), has a root. Let the roots be $z_{2,1} < z_{2,2} < z_{2,3}$.

$Q_3(z)$ has degree 4 and should have 4 roots. $Q_3(0) < 0$, $Q_3(z_{2,1}) < 0$, $Q_3(1) > 0$, $Q_3(z_{2,2}) < 0$, $Q_3(z_{2,3}) > 0$, $Q_3(\infty) < 0$. Except for the first interval, the rest of the intervals have a root each. Thus there is one root between 0 and 1 and 3 roots between 1 and ∞. Let the roots be $z_{3,1} < z_{3,2} < z_{3,3} < z_{3,4}$.

$Q_4(z)$ has degree 6 and should have 6 roots. $Q_4(0) > 0$, $Q_4(z_{3,1}) < 0$, $Q_4(1) > 0$, $Q_4(z_{3,2}) < 0$, $Q_4(z_{3,3}) > 0$, $Q_4(z_{3,4}) < 0$, $Q_4(\infty) > 0$. This shows that there is a root in each of the intervals. Thus there are 2 roots between 0 and 1 and 4 roots between 1 and ∞.

Let us assume that the above pattern is true for $Q_{2k-1}(z)$ and $Q_{2k}(z)$ which implies that $Q_{2k-1}(z)$ is of degree $2(2k-1) - k = 3k - 2$, has $k - 1$ roots between 0 and 1 and the rest $2k - 1$ roots between 1 and ∞. $Q_{2k}(z)$ is of degree $4k - k = 3k$ and has k roots between 0 and 1 and the rest $2k$ roots between 1 and ∞.

From $Q_{2k}(z)$ we know that there are $(k+1)$ intervals in 0 and 1, i.e. $(0,z_{2k,1})$, $(z_{2k,1}, z_{2k,2})$,..., $(z_{2k,k},1)$. To show that there are k roots of $Q_{2(k+1)-1}(z)$ we show that there is no root in the first interval.

From (4) Sign$[Q_{2(k+1)-1}(0)]$=-Sign$[Q_{2k-1}(0)]$. Since $z_{2k,1} < z_{2k-1,1}$ from the induction assumption i.e. there is a root in each interval for $Q_{2k}(z)$, this implies that Sign$[Q_{2k-1}(z_{2k,1})]$ = Sign$[Q_{2k-1}(0)]$, from (2) Sign$[Q_{2(k+1)-1}(z_{2k,1})]$ = -Sign$[Q_{2k-1}(z_{2k,1})]$. Hence Sign$[Q_{2(k+1)-1}(0)]$ = Sign$[Q_{2(2k+1)-1}(z_{2k,1})]$ this implies that there is no root of $Q_{2(k+1)-1}(z)$ in the first interval. The k roots between 0 and 1 are there in the k other intervals. From $Q_{2k}(z)$ we know that there are $(2k+1)$ intervals between 1 and ∞ i.e. $(1, z_{2k,k+1})$, $(z_{2k,k+1}, z_{2k,k+2})$, ..., $(z_{2k,3k}, \infty)$. The $2(k+1)-1$ roots which should exist between 1 and ∞ are one in each interval.

In a similiar way we can prove that $Q_{2(k+1)}(z)$ has $(k+1)$ roots between 0 and 1 and $(2k+2)$ roots between 1 and ∞.

A.3.1 The S+1 case: S Even

From induction we know that $Q_S(z)$ has $\frac{S}{2}$ roots between 0 and 1 and S roots between 1 and ∞. We now show that there are no roots in the interval $(0, z_{S,1})$ and $(z_{S,\frac{S}{2}}, 1)$.

From (4) we know that Sign$[Q_{S+1}(0)]$ = -Sign$[Q_{S-1}(0)]$, from (2) we know that Sign$[Q_{S+1}(z_{S,1})]$ = -Sign$[Q_{S-1}(z_{S,1})]$, from the induction assumption we know that Sign$[Q_{S-1}(z_{S,1})]$ = Sign$[Q_{S-1}(0)]$ since $z_{S,1} < z_{S-1,1}$. All these relations imply that Sign$[Q_{S+1}(0)]$ = Sign$[Q_{S+1}(z_{S,1}]$, thus there is no root in this interval.

From (3) we know that Sign$[Q_{S+1}(1)]$ = -Sign$[Q_{S-1}(1)]$ and from (2) we know that Sign$[Q_{S+1}(z_{S,\frac{S}{2}})]$ = -Sign$[Q_{S-1}(z_{S,\frac{S}{2}})]$, Sign$[Q_{S-1}(z_{S,\frac{S}{2}})]$ = Sign $[Q_{S-1}(1)]$ since $z_{S,\frac{S}{2}} > z_{S-1,\frac{S}{2}-1}$. All this implies that Sign$[Q_{S+1}(1)]$ = Sign$[Q_{S+1}(z_{S,\frac{S}{2}})]$ and hence there is no root in the interval $(z_{S,\frac{S}{2}}, 1)$. Thus of the $\frac{3S}{2}$ roots of $Q_{S+1}(z)$, $\frac{S}{2}-1$ roots are between 0 and 1 and $S+1$ roots are in each of the $S+1$ intervals in the 1 and ∞ range.

We are able to locate all $2(S+1)$ roots of $|A(z)|$. It has $\frac{S}{2}+1$ roots at $z=0$, $\frac{S}{2}-1$ roots between 0 and 1, 1 root at $z=1$ and the remaining $S+1$ between 1 and ∞.

A.3.2 S+1 Case: S Odd

From induction we know that $Q_S(z)$ has $\frac{S+1}{2}-1$ roots between 0 and 1 and S roots between 1 and ∞. We now show that there is no root in the interval $(z_{S,\frac{S-1}{2}}, 1)$.

We will first show that the first interval i.e. $(0, z_{S,1})$ has a root. Sign$[Q_{S+1}(0)]$ = -Sign$[Q_{S-1}(0)]$ from property 4. Sign$[Q_{S-1}(z_{S,1})]$ = -Sign$[Q_{S-1}(0)]$ because $z_{S-1,1} < z_{S,1}$, from induction we know that there is no root of $Q_S(z)$ between 0 and $z_{S-1,1}$ since S is odd. Sign$[Q_{S+1}(z_{S,1})]$ = -Sign$[Q_{S-1}(z_{S,1})]$, property 2. All these relations imply that Sign$[Q_{S+1}(0)]$ = -Sign$[Q_{S+1}(z_{S,1})]$. Hence there is a root between interval $(0, z_{S,1})$.

There is no root in the interval $(z_{S,\frac{S-1}{2}}, 1)$. This can be proved in a similiar way as for S even. Thus of the $\frac{3S+1}{2}$ roots of $Q_{S+1}(z)$, $\frac{S-1}{2}$ roots are between 0 and 1, and the remaining $S+1$ roots are in each of the $S+1$ intervals between 1 and ∞.

$|A(z)|$ has $\frac{S+1}{2}$ roots at $z = 0$, $\frac{S-1}{2}$ roots between 0 and 1, 1 root at $z = 1$ and $S + 1$ roots between 1 and ∞. Thus we are able to locate all the roots of $|A(z)|$.

Appendix B

Proof That $(I - U + U_2)^{-1}$ Exists

For $(I - (U - U_2))^{-1}$ to exist $(U - U_2)^n$ as $n \to \infty$ should equal 0 (Theorm 1.11.1 in [14]). $(U - U_2)^n = U^n - U_2$ due to $UU_2 = U_2$ and $U_2U_2 = U_2$. Now we have to prove that $U^n = U_2$ as $n \to \infty$.

If we consider only U^0 part of U matrix then from Theorm 4.1.3, 4.1.4 and 4.1.6 in [14]

$$(U^0)^n_{n\to\infty} = U_2' = \begin{bmatrix} u_1 & u_2 & \dots & u_S \\ \vdots & & & \vdots \\ u_1 & u_2 & \dots & u_S \end{bmatrix}$$

Now if we consider U as a whole then due to the structure of the matrix

$$U^n_{n\to\infty} = \begin{bmatrix} u_1 & u_2 & \dots & u_S & 0 \\ u_1 & u_2 & \dots & u_S & 0 \\ \vdots & & & \vdots & 0 \\ u_1 & u_2 & \dots & u_S & 0 \end{bmatrix}$$

The zeros in the last column of the above matrix are due to zeros in the last column of U. Due to 1 in the last row of U we get the same elements in the last row as in the other S rows. The intermediate results of the last row does not have any effect due to zero in the last column of U.

Appendix C

Review of Matrix Geometric Method

In this appendix we present the properties of the matrix geometric solutions and related definitions used in the matrix geometric approach.

The Continuous Time Markov Chain (CTMC) of GI/M/1 type called quasi-birth and death processes have a transition rate matrix $\tilde{Q}$ as follows:

$$\tilde{Q} = \begin{bmatrix} B_0 & A_2 & & & & \\ B_1 & A_1 & A_2 & & & \\ & A_0 & A_1 & A_2 & & \\ & & A_0 & A_1 & A_2 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}$$

where B's and A's are matrices.

Since the matrix $\tilde{Q}$ is an infinitesimal generator, the following relation is

satisfied:

$$B_0\vec{e} + A_2\vec{e} = B_1\vec{e} + A_1\vec{e} + A_2\vec{e} = (A_0 + A_1 + A_2)\vec{e} = \vec{0} \qquad \text{(C.1)}$$

where $\vec{0}$ is a column vector with all its elements equal to zero.

These CTMC of GI/M/1 type satisfy the following relation:

$$\vec{x_k} = \vec{x_0}R^k, \qquad \text{for } k \geq 0 \qquad \text{(C.2)}$$

where R is the nonnegative solution to a matrix-quadratic equation[6], see Eq. C.3. The steady state probability vector $\mathbf{x}$=[$\vec{x_0}$,$\vec{x_1}$,$\vec{x_2}$,...] of $\tilde{Q}$, is referred to as a matrix-geometric vector and the above relation C.2 is referred to as matrix geometric. Since $\mathbf{x}$ is a probability vector hence $\sum_{i=0}^{\infty} \vec{x_i}\vec{e} = 1$.

We now present the Theorem 3.1.1 of [23]. This theorem has been used extensively in the analysis of our model.

Theorm 3.1.1. The process $\tilde{Q}$ is positive recurrent if and only if the minimal nonnegative solution R to the matrix quadratic equation

$$R^2A_0 + RA_1 + A_2 = 0 \qquad \text{(C.3)}$$

has all its eigenvalues inside the unit disk, that is $sp(R) < 1$ and the finite system of equations

$$\vec{x_0}(B_0 + RB_1) = 0 \qquad \text{(C.4)}$$

$$\vec{x_0}(I - R)^{-1}\vec{e} = 1 \qquad \text{(C.5)}$$

has a unique positive solution $\vec{x_0}$.

If matrix $A = A_0 + A_1 + A_2$ is irreducible, then $sp(R) < 1$ if and only if

$$\pi A_2 \vec{e} < \pi A_0 \vec{e} \qquad \text{(C.6)}$$

where π is the stationary probability vector of A. The above inequality is also referred to as the stability condition of $\dot{Q}$.

The stationary probability vector $\mathbf{x}=[\vec{x_0},\vec{x_1},\ldots]$ of $\dot{Q}$ is given by

$$\vec{x_i} = \vec{x_0} R^i \qquad i \geq 0 \qquad \text{(C.7)}$$

The equalities

$$RA_0\vec{e} - A_2\vec{e} = RB_1\vec{e} - B_0\vec{e} = \vec{0} \qquad \text{(C.8)}$$

hold.

The transition rate matrix $\tilde{Q}$ in certain complicated CTMCs is of type

$$\tilde{Q} = \begin{bmatrix}
A_1^0 & A_2^0 & & & & & & & \\
A_0^1 & A_1^1 & A_2^1 & & & & & & \\
 & A_0^2 & A_1^2 & A_2^2 & & & & & \\
 & & A_0^3 & A_1^3 & A_2^3 & & & & \\
 & & & & \ddots & & & & \\
 & & & & A_0^{S-1} & A_1^{S-1} & A_2^{S-1} & & \\
 & & & & & A_0^S & A_1^S & A_2 & \\
 & & & & & & A_0 & A_1 & A_2 \\
 & & & & & & & & \ddots
\end{bmatrix}$$

In this case the matrices in the first $S+1$ rows are all unique and after that the matrices in all the rows are same as in the previous CTMC.