ANALYSIS OF A TIME-LIMITED POLLING SYSTEM WITH MARKOVIAN ARRIVAL PROCESS AND PHASE TYPE SERVICE

BY

IMED FRIGUI

B. S. (C. E.), University of Arizona, Tucson (1990)
M. S. (C. E.), University of Texas at Austin, Austin (1992)
M. Sc. (I. E.), University of Manitoba, Winnipeg (1994)

A Dissertation
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Industrial Engineering
The University of Manitoba
Winnipeg, Manitoba, Canada

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

COPYRIGHT PERMISSION

ANALYSIS OF A TIME-LIMITED POLLING SYSTEM WITH MARKOVIAN ARRIVAL PROCESS AND PHASE TYPE SERVICE

BY

IMED FRIGUI

A Thesis/Practicum submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

I hereby declare that I am the sole author of this thesis.

I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Manitoba requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

To My Mother and

the memory of my Father

ABSTRACT

Polling systems have been the subject of many studies and are of interest in the analysis of communication systems, operating systems scheduler, traffic intersections, and manufacturing systems. For communication and operating systems, the time-limited service discipline is very important since it allows one to limit the time the server is away from a particular queue. Nevertheless, it has received little attention, whereas, the exhaustive and gated service discipline have been studied extensively. In addition, most of the available results ignore correlation between arrivals.

In this thesis, we have modeled the Fair Share Scheduler as a discrete time polling system. In this polling system, each queue is visited according to the exhaustive time-limited service discipline, customers arrive according to the Markovian arrival process and their service time has a phase type distribution. Both cyclic and table polling are considered. In addition, we consider, separately, the case when all the queues have infinite buffer capacity and when all the queues have finite buffer capacity. Our solution is based on the decomposition approach. Thus, for the infinite buffer capacity case, each queue in the polling system is treated as a MAP/PH/1 with vacation periods and is analyzed using the matrix-analytic approach. On the other hand, for the finite buffer capacity case, each queue is considered as a MAP/PH/1/K with vacation periods, for which the queue length distribution is obtained using the block Gauss-Seidel iterative procedure.

The results of the MAP/PH/1 or the MAP/PH/1/K are then incorporated into an iterative procedure to obtain the mean waiting time for each queue in a polling system. Because of the time-limited service discipline, the vacation and visit period distributions are represented by discrete-time phase distribution in the case of cyclic polling. However, for table polling, since the type of vacation the server takes depends on its position in the polling table, the vacation period looks like the convolution of discrete phase distributions and is represented by a MAP. In order to incorporate the

correlation between the vacation and visit period distributions, the vacation period is obtained as the sum of an independent and a dependent part. The independent part is the convolution of the visit period of the queues visited while the server is on vacation. The dependent part is computed using an approach similar to that of Lee and Sengupta. The convergence of the iterative procedure is proved for the cyclic polling case using stochastic dominance. We have also proved that if we start with a stable system, then the iterative procedure is stable (for cyclic polling). Comparison between the iterative results and the simulation results shows that the iterative procedure provides reasonable results over a wide range of input parameters.

However, the computational time increases as the dimension of the vacation period becomes large. In our case, the dimension of the vacation period distribution depends on 1) the number of queues in the polling system, 2) the time threshold for the queues visited while the server is on vacation, and 3) the number of visits in the case of table polling. In order to reduce the computational time, the dimension of each phase type vacation period distribution is reduced using the moments matching approach. Comparison between the original and reduced MAP shows that the error in the probability mass function and the coefficient of correlation is very small.

Keywords: Polling systems, mean waiting time, exhaustive time-limited service discipline, vacation models, Markovian arrival process, phase type distribution.

Acknowledgements

I thank Manitoba Hydro for its financial support. I am especially grateful to Mr. E.C. (Ted) Cotton for his support during my graduate studies at the University of Manitoba.

I would like to express my sincere thanks to Dr. Attahiru Sule Alfa, supervisor of this thesis, for his valuable advice, encouragement, support, and friendship.

I thank Dr. R. D. Mcleod, Dr. C. M. Laucht for serving as advisors and examinars of this thesis and Dr. M. Rao from the University of Bowling Green for serving as the external examinar.

TABLE OF CONTENTS

CHAPTER PAGE

CHAPTER PAGE

LIST OF FIGURES

LIST OF TABLES

TABLE **PAGE**

NOMENCLATURE & ACRONYMS

Latin Letters

$\bar{b}$	Mean service time
T	Time threshold

Greek Letters

λ	Arrival rate
$\rho_i = \lambda_i \bar{b}_i$	Queue utilization
$\rho = \sum \rho_i$	System utilization

Acronyms

ATM	Asynchronous transfer mode
CPU	Central processing unit
GSPN	Generalized stochastic Petri nets
FDDI	Fiber distributed data interface
FMS	Flexible manufacturing system
FSS	Fair share scheduler
IBP	Interrupted Bernoulli process

TABLE **PAGE**

IPP	Interrupted Poisson process
ISDN	Integrated service digital network
iid	independent and identically distributed
LAN	Local area network
LST	Laplace-Stieltjes transform
MAP	Markovian Arrival process
MHD	Material handling device
MM	Moment matching
ML	Maximum likelihood
MMPP	Markov modulated Poisson process
PGF	Probability generating function
PH	Phase type distribution
PSA	Power series algorithm
TDM	Time division multiplexing
THT	Token holding time
TRT	Token rotation time

CHAPTER 1

INTRODUCTION

1.1 General

Polling models are a natural representation of many problems in the field of engineering and science. A polling model consists of a single server and many queues. Its use is motivated by reducing wasteful resources and improving efficiency. This is achieved by serving many queues, each possibly having different type of customers, which under normal operating conditions do not require a dedicated server. Furthermore, polling reduces networks' complexity and improves their architecture. Historically, polling systems have been used to model manufacturing systems and traffic intersections. In recent years, and due to technological advances in the areas of computer architecture and communication networks, polling models have been used to represent computer-communications and operating systems. Applications of polling models to engineering problems is discussed in Levy and Sidi [119] and for computer networks in Takagi [169]. Later in this Chapter we present several examples related to the modeling of engineering and computer systems. But for now, let us show the importance of polling through a simple computer communication problem.

Consider a network of four workstations which we wish to inter-connect to share information (e.g. emails). It is feasible, albeit wasteful, to have a dedicated communication line between each pair of workstations as shown in Figure 1.1(a). Clearly, when station 1 is communicating with station 2, the communication links $(1-3)$ and $(1-4)$ may be idle. Consequently, to reduce the complexity of the network and use the resources more efficiently the four workstations may be connected in a bus topol-

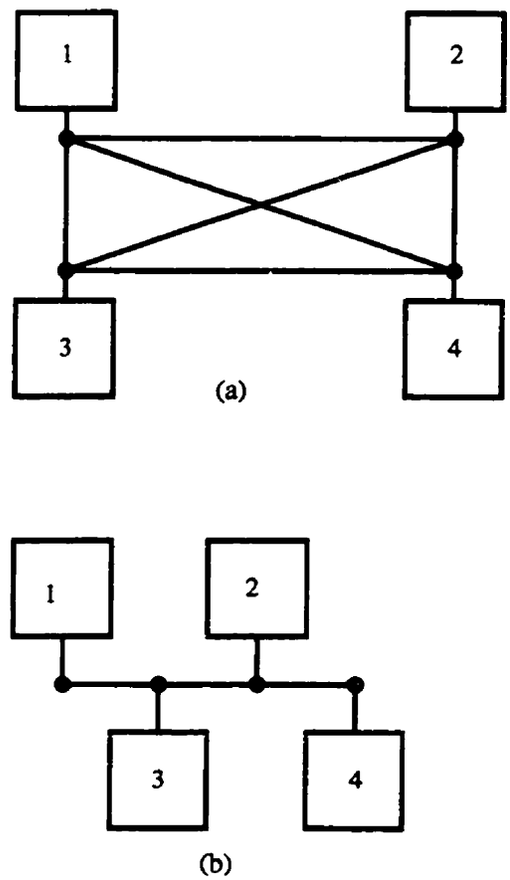

Figure 1.1: 4 Workstations Network

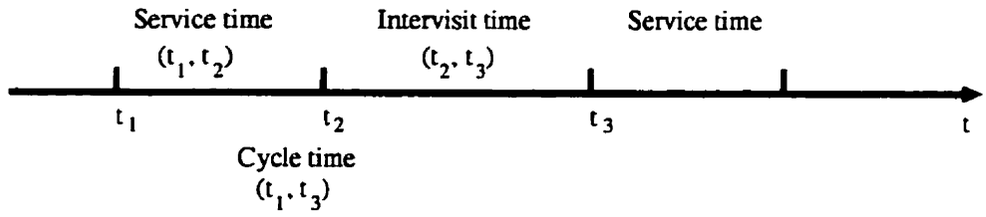

Figure 1.2: Cycle Time, Intervisit Time and Visit Period Relationship

ogy as shown in Figure 1.1(b). This topology raises the issue of contention among the workstations to use the shared communication link. For example, contention occurs when station 1 wants to communicate with station 2 and at the same time station 3 wants to communicate with station 4. Thus a protocol that dictates who can use the communication link and for how long is needed. The new network can be modeled as a polling system with the communication link being the server and each workstation as a queue. The performance of this network is measured in terms of the following:

- The cycle time distribution which is the distribution of the time between successive polls of the same station (queue).
- The intervisit time distribution which is the distribution of the time between the end of a service period and the beginning of the next poll.
- The visit period distribution which is the distribution of the time between polling a station and leaving that station. The relationship between cycle time, intervisit time and visit period are shown in Figure 1.2.
- The queue length distribution at each station.
- The joint queue length distribution at polling instant.
- The waiting time distribution.

In order to obtain one or more of the above performance measures we mathematically model this system as a multi-queue single server system which is also known as

a polling system and is shown in Figure 1.3. Note that the term polling originates in the data link control scheme according to Takagi [167]. Formally, a polling system is defined by:

- The number of queues or stations (e.g. machines, computer terminals, etc.). Note that in this thesis the words station and queue are used interchangeable.
- The input process to each queue, usually represented by a stochastic process like the Poisson process in continuous time models or the Bernoulli process in discrete time models.
- The time it takes to serve a customer which is usually stochastic and known as the service time distribution, for example, the exponential distribution for continuous time models and the geometric distribution for discrete time models.
- The polling order of the queues (e.g. sequential, random, etc.).
- The duration of the visit period for each queue which is determined by the service discipline (e.g. exhaustive, gated, limited, etc.).
- The time span between the end of service at one queue and the beginning of service at the next queue which is known as the switch-over time.

At this juncture, it is worthwhile to differentiate between polling systems and Synchronous Time Division Multiplexing (STDM). In STDM each queue is attended by the server for a fixed length of time whether there are customers to serve or not. In polling if the queue is empty the server does not bother to stay at that queue and moves on to the next queue. Consequently, congestion at each queue in STDM is not affected by congestion at other queues and each queue can be analyzed as a single server queue with deterministic vacation period. On the other hand, congestion at each queue in polling models is affected by other queues. Therefore the analysis must look at the system as a whole. Several analytical methods have been proposed in the literature and they can be exact or approximate. The exact methods are based on the

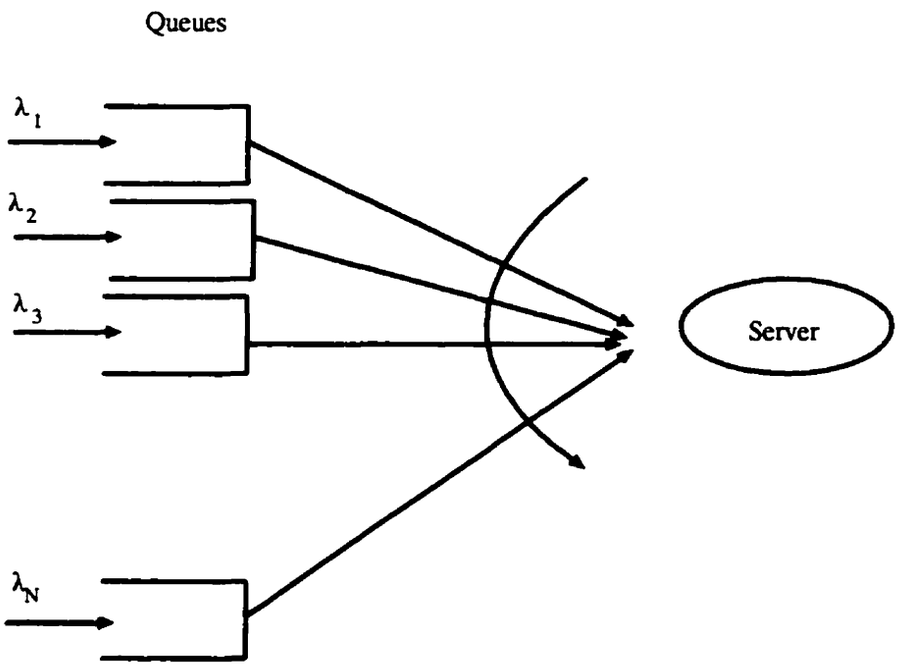

Figure 1.3: Polling System

buffer occupancy, the station time, or the descendant set. The approximate methods are based on the conservation law of Kleinrock [96] or the results of the M/G/1 and GI/M/1 type queues (see Neuts [133, 134]). Chapter 2 elaborates on these issues in more detail. When a mathematical formulation is not possible, a simulation approach is taken. The only problem with the latter approach is that it is time consuming at times and can not be relied upon for optimization.

We are now in a position to discuss some applications of polling systems. However, before we do that, we talk briefly about the thrust of this thesis. The main reason behind this thesis is the modeling of an operating system scheduler. There are several alternatives for process scheduling and they are described in Tanenbaum [175, Chap. 2]. In general, when more than one process is ready to run, the operating system uses the scheduler to decide which is the next process to run. In this thesis we are interested in grouping similar class of process in one queue. The grouping can be done based on the work requirements, priority, etc. A four-class system is shown as a multi-queue system in Figure 1.4(a). In order to make the system fair and equitable, each queue

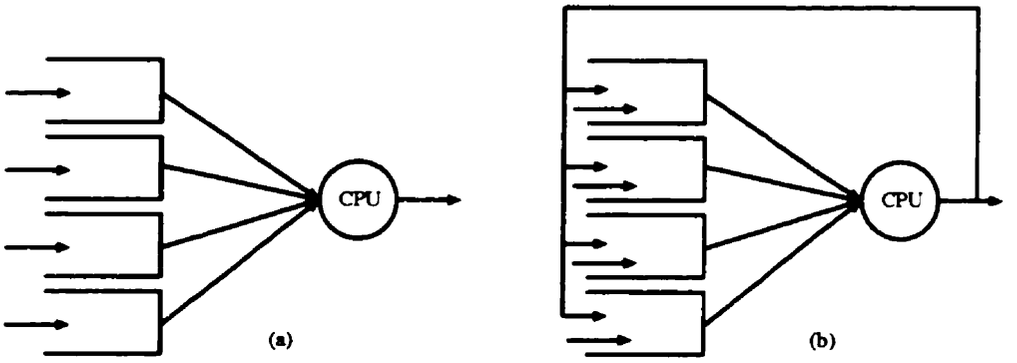

Figure 1.4: A Scheduler with four Classes

has a limited service period after which it relinquishes the central processing unit (CPU). In general, a served process may leave the system, rejoin the same or another queue, or generate another request. This is represented by feedback in Figure 1.4(b) and causes correlation in the input process. Thus, the objective of this work is to analyze a polling system with correlated input process. Contention to use the CPU is resolved using a time limit threshold for each queue. The objectives and contributions of this thesis are presented in more details in Sections 1.6 and 1.7, respectively.

For now, we further elaborate on the importance of polling systems, correlated arrival, and time-limited service discipline by considering examples from the fields of computer communication, operating systems, manufacturing systems, and transportation.

1.2 Communication Systems

The main purpose of communication systems is to facilitate the exchange of information between two entities. The information (e.g. files, email) is put into packets conforming to the network protocols and then sent over the transmission medium. Within a network, users compete to have access to the transmission medium or to the switches. Sections 1.2.1 and 1.2.2 outline, respectively, how a Local Area Network

and an asynchronous transfer mode switch can be modeled as a polling system.

1.2.1 Local Area Network

A Local Area Network (LAN) consists of several terminals connected together via radio, twisted pairs, optical fiber, or coaxial cable. Although each terminal could be a computer that can stand alone, their connectivity is desirable since it increases productivity by: 1) easing communication between members of a group working on related projects, and 2) easing the transfer and sharing common resources. Therefore, the terminals are connected together to form a LAN which results in sharing some resources, for example, data bases, files, computer codes, etc. Typical LANs topologies include the ring, star, and bus. They are shown in Figure 1.5.

In a LAN each terminal generates messages at random and store them in its output buffer. The stored messages wait until the station gains access to the transmission medium. The access to the server is determined according to a protocol known to all the terminals. In addition, in some networks, the right for transmission is passed between stations using a token or a central processor. The token is frequently used with bus and ring topologies. The central processor grants permission to access the transmission medium according to a table. Under this scheme, the central processor may grant the right of access to high priority stations more often. The most common tables are elevator polling which is used to model bus topology and star polling which can model a half-duplex transmission medium. In the elevator polling case, the stations are visited in the following order $1, 2, \ldots, N, N-1, N-2, \ldots, 1, \ldots$, where N is the number of stations. This polling scheme reduces wasted time when the connect time between stations is very large.

It is shown in Altman *et al.* [6] that for the globally gated service discipline the mean waiting time is independent of the station index. In the case of two way traffic, we have one station, say station N, that sends messages to stations 1 through $N-1$ and in return may receive messages from all the stations. This is similar

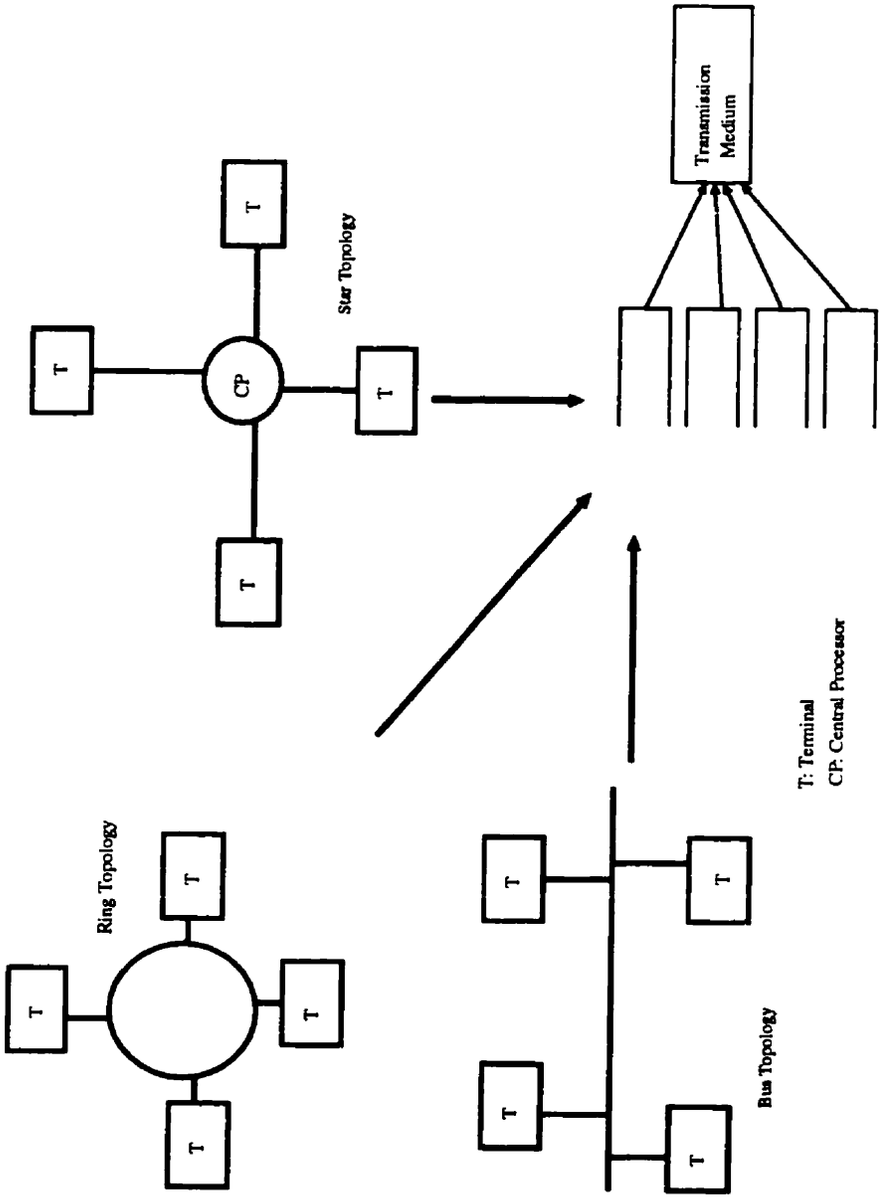

Figure 1.5: Local Area Networks

to the star topology. In this case, the stations are visited in the following order, $1, N, 2, N, \ldots, N, N-1, N, 1, \ldots$. This, for example, can represent a LAN with 4 workstations and a printer. Every time a job is printed the job's owner has to be notified. In order to speed up the notification process, the printer is granted access to the transmission medium after every workstation. Table polling can be used to model networks with general topology. Schwartz [151, Chap. 12] showed how to model the communication protocol for an Airline reservation system as a polling system. Recently, Takagi [169] showed how some of the results of polling models can be used for communication networks (e.g. half-duplex transmission, Newhall loop, token passing protocols, etc.).

1.2.2 Asynchronous Transfer Mode Switch

Currently there is an increased interest in Asynchronous Transfer Mode (ATM) networks. This lead to many ATM switch architectures. One of the many considered architectures is the shared-medium. In a shared medium switch all packets arriving on the input links are forwarded to the output links over a common high-speed medium such as a parallel bus. Each output link is capable of receiving all packets addressed to it. A shared-medium packet switch with N input links and M output links can be modeled as a polling system (e.g. Zaghloul and Perros [185]). A generic shared medium switch is shown in Figure 1.6.

The N input links are attached to the shared bus and contend for access when they have one or more messages (cells) to transmit. The order in which the input queues are served is determined by the bus arbitration or polling scheme. In order to model such a system accurately it is imperative to take into consideration system characteristics such as the buffer capacity, the burstiness of the input traffic, and non-symmetric load conditions. Because the buffer capacity at the input links is finite, cells arriving to a full input queue will be lost. Furthermore, if one of the output queues is full, the flow of messages (cells) will be stopped, and consequently, the

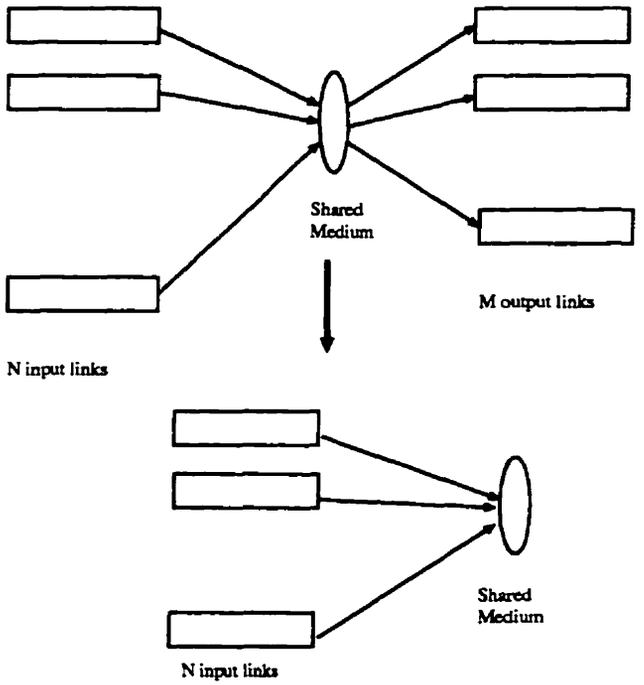

Figure 1.6: ATM Shared-medium Switch

server becomes idle. Because the input buffer capacity is finite and because of the blocking at the output queues closed form solution are difficult if not impossible to obtain. As a result such queueing networks are usually analyzed approximately using the notion of decomposition.

1.3 Operating Systems

In this section we consider the Fair Share Scheduler (FSS) and the X11 client/server model. We will show how both of these models can be viewed as multi-queue single server models.

1.3.1 Fair Share Scheduler

The (FSS) described in Henry [80] is a process scheduling scheme for distributed operating systems. Under the FSS processes having the same work requirements are

grouped together. For example, in a UNIX environment, professors are assigned to one group $G1$, graduate students to a second group $G2$, and undergraduate students to a third group $G3$. Each group is then allocated a percentage of the system resources proportional to its usage and priority. This will ensure that a heavy usage by one group does not clog the system and starve other processes. For example, during the end of a school term, undergraduate students are rushing to finish term projects. To ensure that professors and graduate students receive their share of the system resources, undergraduate students' system resource utilization is limited to what they were initially assigned (see Figure 1.7). However, if one group is inactive then its share of system resources is divided between the active groups in accordance to their system's usage.

The FSS can be modeled as a polling system where each group of users is assigned to a queue. Within each group there is a "think" period after which a message or a request is generated by a user. Thus, messages to each group's queue arrive according to a random process. Since the pool of users is not identical, the service time of each message is, generally, represented by a random process. It is asserted in Tanenbaum [175] that in many time-sharing systems the time is discretized into time slots (quantum) with transitions between states occurring at these time slots' boundaries. Thus, the FSS is better modeled in discrete time. In this model, the percentage of the CPU usage per group can be viewed as the maximum time the sever can spend at the corresponding group's queue. However, if a particular queue is empty, instead of wasting resources, the CPU serves messages from the next group. As with any real system the buffer capacity is finite.

The performance of transaction driven computer system (TDCS) can be found in Groenendijk and Levy [77] and that of a disk drive in Tanenbaum [174].

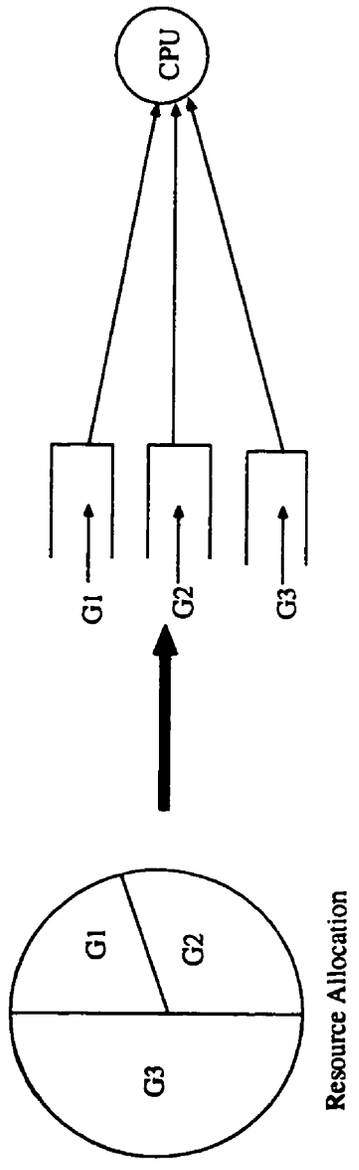

Figure 1.7: Fair Share Scheduler

1.3.2 Client/Server Model

In a client/server environment, many clients share a single server. The server provides service to these clients in a random or sequential fashion. Each client has a "think" period after which a request is generated and sent to the server. The server, if free, will provide services to that request. Otherwise, the request is enqueued until the server becomes available. Most universities and research institutions have a computer environment which is distributed and has a client/server architecture which can be described similarly to Section 1.3.1. The focus of this section is the X11 client/server environment (see Figure 1.8) which is introduced by Scheifler and Gettys [149]. Notice that X11 gives the impression that the role of the client and the server are reversed.

The role of an X11 server is to multiplex requests from clients to the display. The clients are the applications which use the server to display information on the screen. With the help of a terminal, a user can have several windows open at the same time. For example, a user can have a window to read mail, the second to edit text and a third to compile programs. The X11 server, in a round-robin fashion, tends to these applications. The basic resources provided by the X11 server are windows, fonts, mouse cursors, and off-screen images. Clients request creation of resources by providing appropriate parameters. For example, to display text in a window, the client has to provide the drawing color, the window identifiers, the font, and the string of characters. When applications have information to display on the screen, they contend to use the server. For example, consider the case where an email has arrived, the user is editing a file using emacs, and compilation of a program has finished. The X11 server displays, in the appropriate window, in a round-robin fashion, the output of each application.

This system can be modeled as a polling system in which X11 is the server, and each application has a dedicated queue for its requests. Applications generate messages at random and the server displays the results of these requests in a round-robin fashion. According to the X11 protocol a served request may generate a reply

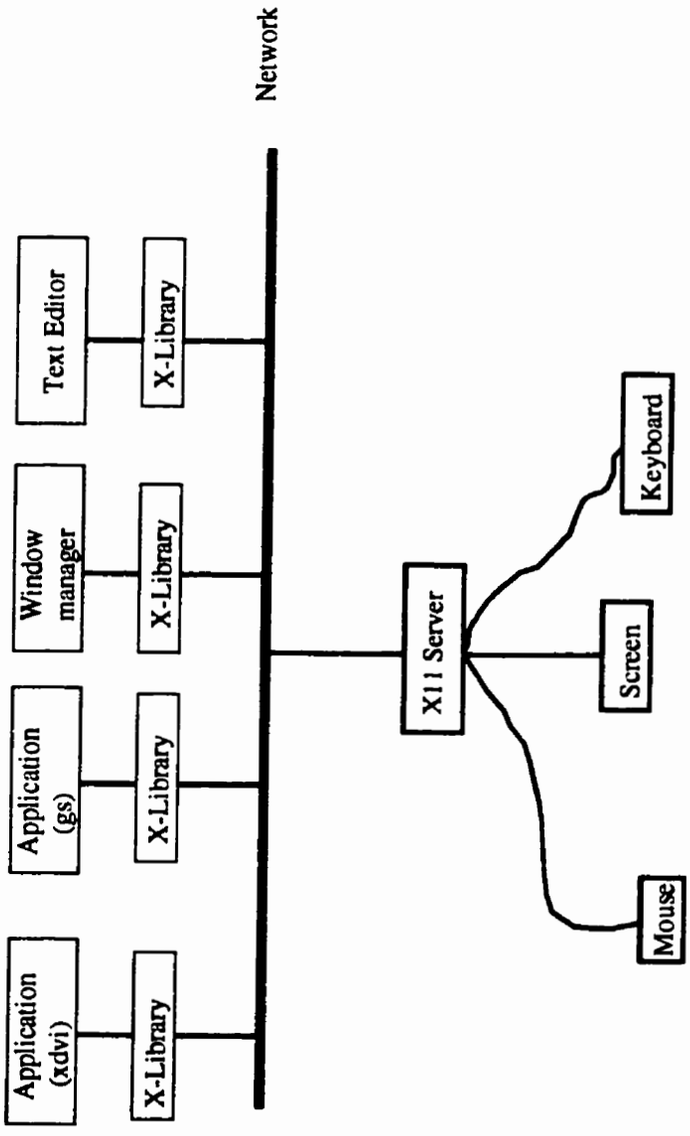

Figure 1.8: X11 Server

which in return may create another request. Thus, it is necessary to consider an arrival process that can capture this correlation in the inter-arrival time between requests. Note also that each client, application, has a finite buffer capacity which may lead to blocking when the buffer is full.

1.4 Manufacturing Systems

This section discusses two classical problems in the area of manufacturing: 1) the machine repair person problem and 2) the material handling device problem.

1.4.1 Machine Repair Person

In a manufacturing environment, several machines are patrolled by a single repair person whose movement between the machines is pre-specified. The machines may request one of two types of service: routine maintenance or repairs. Therefore, customers in this system are of two kinds: low priority (maintenance) and high priority (repairs). Because old machines are more prone to break down, the repair person may visit them more frequently in a given cycle. In addition, the arrival process, machine break down or routine maintenance, has to take into consideration the inherent correlation between the age, the last time a station is served and the next time it will require service. Notice also that the buffer size is equal to one. This is because if a machine breaks down or requires a routine maintenance then it will stay idle until it is visited by the repair person.

The machine repair person can be viewed as a polling system in which each queue, machine, has a buffer size equal to one. The switch-over time is the time it takes the repair person to move from one machine to the next. This problem was analyzed by Mack *et al.* [128] and Mack [127].

1.4.2 Material Handling Device

There are a large number of Flexible Manufacturing Systems (FMSs) configurations. In this section we consider a configuration that can be modeled as a polling system. Specifically, a material handling device moving parts from a set of machines is modeled as a polling system.

Consider a manufacturing environment that consists of several work stations with each work station having many parallel machines (see Figure 1.9). In addition to the central storage area, each work station has a local material handling device (MHD). The role of the local MHD is to move parts from each machine to the central storage area. If the central storage area is full then blocking occurs. Parts are generated by each machine according to a random pattern and then stored in its buffer. The machines have finite buffer capacity which could be equal to one if a machine can work only on a single job at a time and has no self storage area. If that buffer is full then parts are blocked. Therefore, in this configuration blocking may occur at two stages: If the MHD is not available to move parts to the central storage and the buffer is full (input blocking), or if the central storage is full (output blocking). Notice the striking similarity between this and the configuration for the ATM switch presented in Section 1.2.2.

For analytical modeling purposes, we decompose the system and consider only one work station. A work station has several machines and each machine generates parts at random and store them in its buffer. Hence, a single machine can be viewed as a single queue with finite buffer capacity. Because in a manufacturing system the input to a machine is the output of another machine, the arrival process should be one that takes into account correlation. The MHD moves the parts from the machines to the central storage area. If the central storage area is finite then output blocking becomes significant. For this model, due to storage limitations, blocking is a very important performance measure to management (a blocked machine is an idle machine).

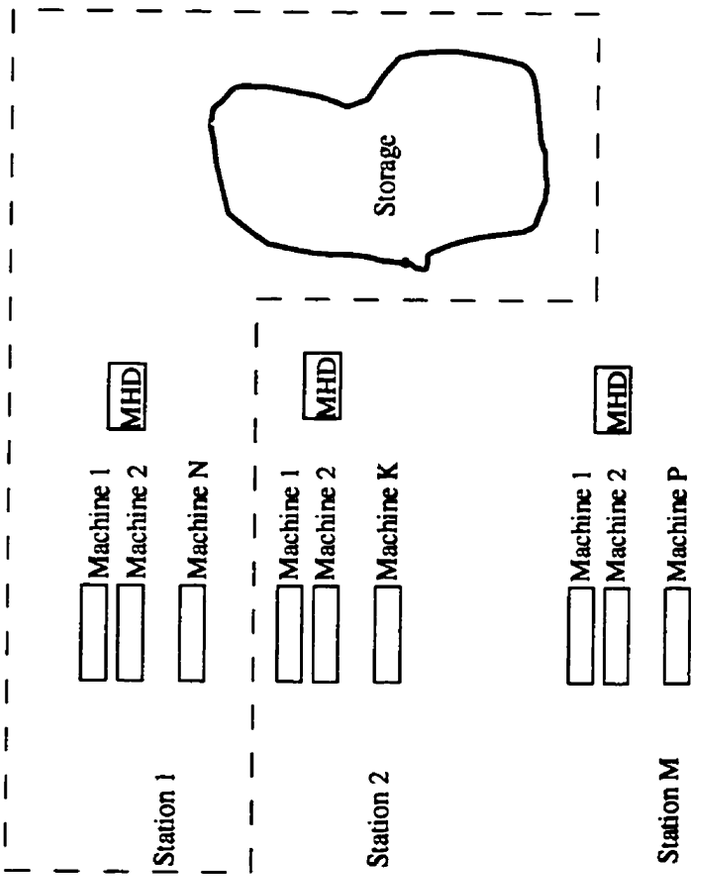

Figure 1.9: Material Handling Device

1.5 Traffic Signal Control

A common sight in our daily life are traffic intersections. Looked at closely, a traffic intersection can be modeled as a multi-queue single server system where the road intersection and the road lanes represent, respectively, the server and the queues. There is a competition between the lanes to use the intersection. A typical road intersection is shown in Figure 1.10. In order to permit an orderly usage of the intersection by the vehicles, we use traffic lights to control access to the intersection in a pre-determined fashion. Each lane has a finite capacity and cars arrive according to a random pattern. Since the input to a traffic intersection is a collection of outputs of upstream traffic lights, the arrival process has some correlation. This correlation is best captured using the Platoon arrival process present in Alfa and Neuts [4] or by using MAP as in Alfa [3]. Notice that traffic intersections in which each lane has a fixed time period resemble STDM models, hence each lane can be analyzed separately. When the traffic light is vehicle-actuated each lane can no longer be analyzed as a single queue and therefore one can model it as a polling system.

Sections 1.2-1.5 presented some applications of polling models. As will be presented in Chapter 2, this diversity in applications has resulted in hundreds of research articles which give rise to the question "Why another thesis on polling systems?".

1.6 Objectives

The motivation behind this work stemmed originally from the model presented in Section 1.3.1. Thus, we use a discrete time model. However, the suggested model can be used for many other applications. It is clear from the above applications that:

- The arrival process has to be one that can capture correlation between inter-arrival times. This can be achieved by adopting the Markovian Arrival process introduced by Neuts [132] and described in Section 3.3.

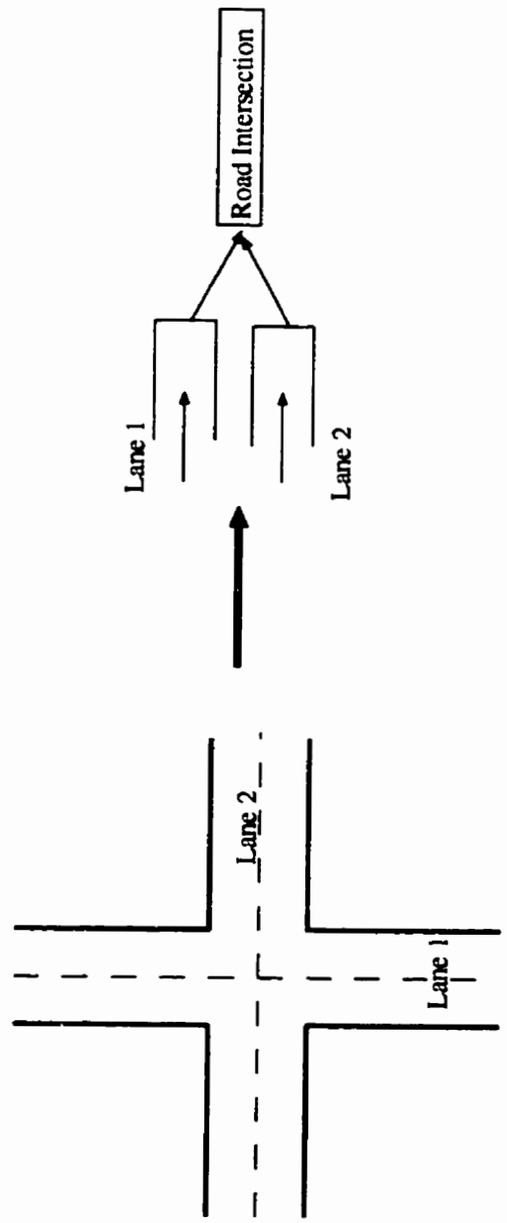

Figure 1.10: Road Intersection

- The service time distribution depends on the type of applications and can be deterministic as in the case of serving (transmitting) an ATM cell to general as in the case of LANs. However, because the phase distribution, presented in 3.2, is well suited for numerical computation [133, page 79] and can be used to represent most service time distributions we use it.

- In order to guarantee fairness and accessibility to the server, and at the same time provide high priority customers with quality service, we use the exhaustive time-limited service discipline. In this discipline, each queue is served for a maximum period T preemptively (the server interrupts an on-going service and will resume where it left off in a future visit). However, if the queue becomes empty before the threshold T, then the server moves on to the next queue.

- The switch-over time is set equal to zero to reduce delays. This can be achieved by using distributed control polling. Thus, the last message to be served in each queue is a signal to the next queue to receive service (see Schwartz [151, page 265].

Therefore, in this thesis we provide an approximate analytic solution for polling systems with either cyclic or table polling order. Under each polling order, we consider two cases: 1) when all the queues have finite capacity and 2) when all the queues have infinite capacity. Customers arrive according to MAP and their service time is of phase type. The switch-over time is equal to zero. Our solution is based on the decomposition approach. Each queue is considered as a single server queue with visit and vacation periods, where the vacation period for each queue is the service period of the other queues. Our focus is the mean waiting time which is considered to be the most important performance measures for computer networks [98, Chap. 3]. Because of the inter-relationship between the visit and vacation period distributions an iterative procedure is used.

As a result of using the decomposition approach, the dimension of the vacation period distribution becomes quite large. Thus, we extend the three moments approach for fitting continuous time phase distributions of Altiok [5] to the discrete phase type distributions.

1.7 Significance and Contributions

Although several researchers have worked on polling systems, few have considered the idea of using vacation models. They mostly used the buffer occupancy, station time, or descendant set method. This may be due to the fact that most researchers considered an input process of the Poisson type. Thus, a mathematical formulation based on the lack-of-memory property can be easily done. Notice that the assumption of a Poisson process is not a bad assumption for homogeneous networks. However, as a result of the multimedia revolution, future networks will offer integrated services such as the superposition of video, voice, and data. Thus, there is a need to use a more versatile arrival process like MAP which is used in this thesis.

Although the service time distribution is of the phase type (previous work used the general distributions), the results obtained here can be applied to a wide range of service time distributions. This is because the phase type distribution can be used to represent most service time distributions and is very well suited for numerical investigations [133, page 79]. Therefore, the models presented in this thesis use the discrete time phase distribution to represent the service time distribution.

In addition, it is known that for asymmetric polling systems heavily loaded queues tend to starve the rest of the queues. This results in very unbalanced mean waiting times. This conflict is resolved in this thesis by using the exhaustive time-limited service discipline. Also, this work would be one of the few that combines table polling with exhaustive time-limited service discipline.

Lastly, even though the model considered in this thesis has zero switch-over time,

its extension to the case of non-zero switch-over time can be easily done by modifying the vacation period distribution as shown in Section 3.7.

1.8 Outline of the Thesis

The remainder of this thesis consists of six chapters. **Chapter two** is dedicated to background and literature review. The chapter starts with the definition of some polling terms, then over 150 articles were reviewed. **Chapter three** introduces the discrete arrival process MAP and the discrete phase type distribution. The analysis of the MAP/PH/1 and the MAP/PH/1/K queue is then followed by the iterative procedure for cyclic polling with the exhaustive time-limited service discipline. In **Chapter four**, we extend the results of Chapter 3 to handle table polling. **Chapter five** presents the state space reduction of MAPs with special structures using the moments matching approach. **Chapter six** concludes this work and outlines future challenges.

CHAPTER 2

BACKGROUND And LITERATURE REVIEW

2.1 Introduction

Polling is a scheduling mechanism for multiple queue and single server systems. The server attends the queues according to one of the polling order policies outlined in Section 2.2.2. Despite the complexity of the model arising from the multiplicity of the queues, the arrival process, the service time distribution, etc, there is a large body of literature on polling systems. The progress in the area of queueing analysis has made it possible to assess the performance of many engineering problems using polling system as a modeling tool. Polling has been used as early as 1950s in the British cotton industry. Mack *et al.* [128] and Mack [127] modeled the patrolling machine repair person problem as a polling system with single buffer at each queue. Later, polling systems were used to study the problem of vehicle-actuated traffic signal by Newell [136], Newell and Osuna [137], and Stidham [162]. The introduction of computer communication protocols have created a wide array of problems. Initially, polling was used for data transfer from terminals on multi-drop lines to a central computer as in Konheim and Meister [105] and in time-shared systems as in Kleinrock [97]. Later, it was used by Bux [30] for token passing local area networks (i.e. token ring and token bus) and by Levy and Kleinrock [100] for broadcasting systems like the ALOHA protocol. In the current studies of ATM networks, Zaghloul and Perros [184] used polling to model a shared medium switch in an ATM network. A major reason for the diversified use of polling is that resource sharing, single server and multi-queues system, is natural in many fields of engineering and sciences.

Due to the large body of literature on polling, first we refer to many survey articles. Later this Chapter focuses on two aspects of polling. First, we review the different solution methods of polling systems and then present some of the most recent literature. The discussion of each article focuses on one aspect that makes the work stands out, for example, the service discipline. However, before discussing the literature let us define some terms associated with polling systems.

2.2 Definitions

Each polling system has two important characteristics, namely the service discipline and the polling order. The service discipline determines the distribution of the time the server spends at a queue and the polling order gives the sequence in which the queues are visited. There is a large variety of service disciplines and polling orders. First, we address the different service disciplines then the different polling orders.

2.2.1 Service Disciplines

Exhaustive Discipline: The queue is served until all present and arriving customers in the current visit period are served. The server leaves the queue when it becomes empty.

Semi-Exhaustive Discipline: The queue is served until the number of customers in the queue is 1 less than the number of customers present at the polling instant.

Gated Discipline: Only customers present at the polling instant are served. Customers arriving in the current visit period are served in the next visit.

Exhaustive K-limited Discipline: At most K customers are served in a visit. The server leaves the queue once the queue becomes empty or K customers are served.

Gated K-limited Discipline: The minimum of K or the number of customers present at the polling instant is served.

Time-Limited Discipline: A polled queue is served for a maximum period T. Sim-

ilar to the K-limited, this can be exhaustive or gated.

Probabilistically-Limited Discipline: The maximum number of customers served at a queue during a server visit is determined by a probability function.

Binomial Discipline: The number of customers to be served during a server visit is binomially distributed with parameters X_i, the number of customers present at queue i at the polling instant, and p_i, $0 < p_i \leq 1$. This is a special case of the probabilistically-limited discipline.

Reservation Discipline: At the end of a visit period the queue makes a reservation for its service requirements for the next visit.

It is worth mentioning here that in a polling system it is not necessary for all the queues to have the same service discipline. However, as shown in [123], the exhaustive service discipline minimizes the unfinished work in the system with no regard to delay limits.

2.2.2 Polling Orders

Cyclic Polling: The queues are visited cyclically.

Table Polling: The queues are visited according to a pre-specified table (e.g. star polling, elevator polling).

Random Polling The queues are visit randomly. At the end of a service period each queue i has probability p_i of being the next queue to seize the server.

Markovian Polling: The next queue to be polled is determined according to an irreducible positive recurrent discrete parameters Markov chain.

Bernoulli Discipline: After service completion of a customer at queue i, the server will start service of the next customer at queue i with probability q_i and will leave the queue with probability $1 - q_i$. However, if the queue becomes empty then the server polls the next queue.

This wide range of service disciplines combined with the different polling orders resulted in many solution methods which are discussed next.

2.3 Analysis

Several methods have been developed to determine various performance measures in polling systems. Initially, the buffer occupancy method was used to analyze polling systems with exhaustive or gated service discipline. Later, the station time method was used to compute the mean waiting time for polling systems with exhaustive or gated service discipline. The quest for an easier approach lead to the use of branching theory for the gated and exhaustive service disciplines. Nevertheless, the mean waiting times can be obtained only by solving a system with N equations, where N is the number of queues. For the limited (time or number) service discipline, the computation of performance measures such as the queue length or the waiting time distributions are very difficult if not impossible. This is attributed mainly to the non-Markovian property of the limited service disciplines.

Aside from the exact analytic methods, approximate methods were developed to obtain performance measures for polling systems due mainly to:

- The requirement to solve $O(N)$ equations to obtain only the mean waiting time for the gated and exhaustive service disciplines.
- The difficulty associated with obtaining performance measures for the limited service discipline.
- The need for delay bounds for analytically intractable models.

The approximate approaches are based on either the extension of the conservation law introduced in [96] or on an iterative approach that uses the M/G/1 type queue with vacations. These approximate approaches may lead to exact results under special cases (symmetric systems). When a mathematically tractable formulation of the network is not possible, the practitioner or researcher is left with simulation which is always an alternative, albeit an expensive one.

In order to discuss the different solution methods for cyclic polling systems, consider a system in which customers arrive according to the Poisson process and their service time is given by a general distribution. The switch-over time, if there is any, is also generally distributed.

2.3.1 Buffer Occupancy Approach

The buffer occupancy method was used by many researchers [42, 43, 52, 78, 105, 144], among many others, for the analysis of cyclic polling systems with or without switch-over time. Later, in a monograph, Takagi [165] presented results for both exhaustive and gated service disciplines for the continuous and discrete time polling models. This approach is based on defining random variable (rv) X_i^j, $1 \leq i,\ j \leq N$, representing the number of customers at queue j when queue i is polled and relies heavily on the Laplace-Stieltjes transform (LST). The relationship between queue i and queue $i+1$ was utilized to obtain expressions for the mean queue length, $E[X_i^j]$. The cross correlations, $E[X_i^j X_i^k]$, are obtained by solving numerically a set of N^3 equations. It is known (see Takagi [165]) that for symmetric system this set can be reduced to N^2 equations. The LST of the waiting time distribution can be obtained using the relationship between the busy period and the queue lengths distribution. The summary of the results for the queue length and the waiting time distributions are available in Takagi [165]. Later, Levy and Kleinrock [117] extended this method to polling systems with zero switch-over periods. Note that the buffer occupancy method was the most widely used method and provided many useful results for cyclic polling systems. However, its application is limited to systems in which the inter-arrival time is exponentially distributed. Also, if one is interested in the whole distribution, say of the queue lengths, then inverting LST is necessary since most of the analysis is performed behind a *Laplacian curtain*. Although, this is not terribly difficult due to the various techniques to invert LSTs (see Duffy [51]), it is an inconvenience.

A variant of the buffer occupancy method was introduced in Swartz [164] for a

discrete-time polling system with the exhaustive service discipline. In this method, each queue at the polling instant is considered as a gambler ruin problem. The initial number of customers corresponds to the gambler' s initial capital, the service time of a customer is the playing fee, and the number of arrivals per time slot is the pay off. Notice that here the time to ruin in the gambler' s ruin problem corresponds to the exhaustive service discipline. The advantage of this method is that it reduces the number of computations required to obtain the mean queue length. However, this approach is limited to polling systems with exhaustive service discipline and slotted service discipline (i.e.,the service time is discretized).

2.3.2 Station Time Approach

The station time, which corresponds to the visit and switch-over time, approach presented in [32, 60, 84], was used for symmetric and asymmetric systems with exhaustive or gated service discipline. In this method, the waiting time distribution is obtained based on the analysis of the station-time distribution. As in the case of buffer occupancy method, the station time method relies heavily on the LST. The key idea of this approach is to define the station time for each queue and then write a recursion formula for the joint queue station times. Once the station time is obtained, the cycle time and inter-visit time are derived. The LST of the waiting time distribution is then obtained based on the distributions of the station time and the inter-visit time. Notice that, like the buffer occupancy method, most of the analysis is done under a *Laplacian curtain* and that the mean waiting times are obtained by solving a set of N^2 equations. Although, these equations are less complicated to solve, in terms of storage and intermediate results, than the buffer occupancy approach, the station time method, as the buffer occupancy method, is limited to polling systems with Poisson input and exhaustive or gated service discipline.

A variant of the station time method was introduced by Sarkar and Zangwill [148] and relies on the solution of N equations to obtain the mean waiting times. Unlike

the standard station time method where the variance of the cycle time is obtained by solving N^2 equations, in [148] the variance of the cycle time is obtained by solving N equations. This can be achieved by relating the cycle times for station i and station $i+1$. However, the resulting N equations are dense and the benefit of reducing the number of equations is off set by using a numerical approach that requires $O(N^3)$ to obtain the mean waiting times.

2.3.3 Descendant Set Approach

The descendant set approach, based on branching theory, was used initially by Avi-Itzhak, Maxwell and Miller [11] for the analysis of alternating queues. Later, it was used by Fuhrmann and Cooper [68] for the stochastic decomposition of the M/G/1 queue and in [23, 41, 104, 143] for the analysis of polling systems. This method is valid only for systems with exhaustive or gated service discipline in which customers arrive according to a Poisson process. Like the buffer occupancy approach, the descendant set method derives the moments of the queue length at the polling instant. This is achieved by considering each customer in a polling system to be either an original (parent) or a non-original (children) customer. An original customer is a customer that arrives to the system during the switch-over time and a non-original customer is a customer that arrives to the system during the service time of another customer (be it original or non-original). Using the generating function, the queue length distribution at polling instant is derived based on the relationship between the number of original and non-original customers. This relationship is obtained based on the service discipline and the Markovian property of the arrival process. While the descendant set method relies on the generating function technique, it is more efficient than the station time and the buffer occupancy methods since the number of computations to obtain the mean waiting time is of $O(N)$. However, similar to the buffer occupancy and station time methods, only the first few moments of the queue length distribution are computable. The full distribution can be obtained only

through inverting the generating function of the queue length distribution.

2.3.4 Approximate Approaches

Several approximate methods are used for the analysis of polling systems. They can be grouped into two methods. The first is based on extending the conservation law introduced by Kleinrock [96] to pseudo-conservation laws and the second is based on the decomposition approach.

The pseudo-conservation law used in [23, 25, 35, 56, 57], and by many others, usually yields a weighted average or an upper bound for the mean waiting time. It is well known that polling systems with switch-over time are not work conserving systems since the server remains idle during switch-over time, although work might be present in the system. Nevertheless, pseudo-conservation laws were derived for polling systems based on the stochastic decomposition results of polling systems by Fuhrmann [67] and the stochastic decomposition results of the M/G/1 queue by Fuhrmann and Cooper [68], Doshi [49, 50], and Scholl and Kleinrock [150] (further references related to the M/G/1 queue and its analysis can be found in [134, 170]). The stochastic decomposition result proves that the total amount of work in a polling system is composed of two independent parts: one is the amount of work in the corresponding system with no switch-over times; and the second is the amount of work at an arbitrary epoch during switch-over period. A survey of conservation law results with application to polling systems can be found in [22]. Similarly, the stochastic decomposition of the M/G/1 queue with vacation states that the total amount of work in the queueing system is composed of two parts: 1) the corresponding amount of work in the M/G/1 queue with no vacation, and 2) the amount of work added to the system by those customers that arrive to during the vacation period. The proof of these results can be found in [49, 68, 150].

The decomposition approach (i.e. decompose the polling system into single server queues with vacation) was used by many researchers, among them [31, 47, 65, 109,

113, 115, 184], to approximate the behavior of polling systems. In this method each queue is treated, separately, as a single server queue with vacation. The analysis is done in two stages. In the first part, which is exact, the performance measures of the single server queue with vacation are derived. The second part of the analysis focuses on obtaining an approximation for the vacation period distribution. When possible, the vacation period distribution is taken as the convolution of the visit periods of the other $N-1$ queues, where N is the number of queues in the system. However, when the vacation period does not lend itself to a simple convolution of the visit periods, an approximation of the vacation period based on a dependent and an independent part is taken. In either case, using an iterative procedure, the decomposition approach converges fairly fast to within an acceptable error. The decomposition approach is being used more frequently for several reasons, among them:

- The arrival process can no longer be assumed to be Poisson. More realistic traffic models have been proposed to characterize bursty traffic, for example MAP was used by Blondia [18, 19] and Blondia and Theimer [20] for B-ISDN. Sriram and Whitt [158] and Heffes and Lucantoni [79] modeled a packetized voice and data traffic using MMPP. The importance of the effect of correlated arrivals on the performance of queueing system is discussed in Patuwo *et al.* [140]; and
- The limited service discipline is emerging as the preferred service discipline. This is reflected by ANSI/IEEE [1] and ANSI [159] standards.

2.3.5 Limited Service Analysis

In general limited (time or number) service disciplines are inherently difficult to analyze and do not lend themselves to an exact analyses. Exact results are known only for few special cases (e.g. symmetric systems, alternating queues). For the special case of fully symmetric system with Poisson input, general service time distribution and 1-limited service discipline one can use the buffer occupancy approach (see Takagi [165]). For the case of alternating queues, a solution is available for systems with

general input parameters via translating the problem into a boundary value problem like a Riemann-Hilbert problem as in Eisenberg [53] or using matrix-analytical approach as in Alfa [3].

Although the limited service discipline is the most important for applications, there is no known method that leads to exact results. Thus, many researchers used approximate methods based on either the pseudo-conservation law or the decomposition approach to obtain some performance measures. The approximate solutions available are model dependent and require substantial computational time. Particularly, the time-limited service discipline is approximated by: 1) exponential timer in Coffman *et al.* [40] and Leung [113], 2) the sum of exponential-phase timers in Leung and Lucantoni [115], 3) probabilistically-limited service in Leung [112], 4) the k-limited service in Fuhrmann and Wang [69] and Frigui, Stone and Alfa [65], and 5) the Bernoulli service in Blanc and van der Mei [17] and Servi [153].

This concludes the review of the solution methods available for polling systems. In brief, with either the exhaustive or gated service discipline it is possible to use 1) station time approach, 2) the buffer occupancy approach or 3) the descendant set method and obtain a set of equations that can be solved numerically for the mean queue lengths. Upper bounds and weighted average of the mean waiting times can be obtained by pseudo-conservation law. Unfortunately, these methods rely on the transform method which leads the analysis away from probabilistic arguments (Neuts [133, page 3]). In addition, these methods require the arrival process to each queue to be Poisson which restricts the arrival processes which can be modeled.

Eventhough the limited (time or number) service discipline is the most important service discipline from an application point of view, very few exact results are available. Most of the work done under this service discipline is done using the pseudo-conservation law or the decomposition method.

2.4 Review Articles

The use of polling models in the field of transportation, manufacturing, and computer communications resulted in a large body of literature. One of the first papers that addressed the use of queueing in computer communications is by Kobayshi and Konheim [102] in which they presented some aspects of applying queueing to computer communication (over 150 citations). Later, Penney and Baghdadi [141, 142] surveyed the application of polling to computer communications and Bux [30] surveyed the applications of polling to local area networks. More recently, Sachs [147] presented a review on the different access protocols for LANs. She presented a thorough review of random access, demand assignment, and adaptive assignment protocols. She included over 150 references. And Kleinrock [99] presented some applications of queueing theory to wide-area networks, packet radio networks and local area networks (140 citation). On the theoretic side, Watson [181] summarized the results for cyclic polling systems with exhaustive, gated, or 1-limited service discipline. Later, Takagi [165, 166] presented most of the analytical results available up to to 1988 for polling systems. In a sequel article Takagi [167] presented an update on polling systems (over 400 articles). The review papers by Takagi focus mainly on the analysis approaches.

2.5 Cyclic Polling

2.5.1 Finite-Buffer Systems

Finite buffer capacity models are a natural representation of real life queueing systems. However, their analysis is difficult. In this section, unless otherwise mentioned, customers arrive according to the Poisson process, service time and switch-over time are generally distributed.

Tran-Gia [177] proposed an algorithmic solution for polling systems with 1-limited service discipline and general renewal input traffic. He developed an iterative procedure to obtain the queue length distribution based on the fast Fourier transform algorithms. In each iteration the conditional cycle time, the queue length distribution, and the group size arrival distribution are computed. The iterative algorithm is stopped once the difference between the mean of the queue lengths for two consecutive iterations is less than a prescribed tolerance. The important aspect of this paper is the use of the general renewal input process. However, the complexity of the computational scheme hinders the use of this analysis approach to other service disciplines like the time-limited and k-limited, $k > 1$, service disciplines.

Eisenberg [54], using the same technique as in [52], derived the LSTs of the waiting time distribution for a polling system in which the server comes to a stop once the system is empty. He considered three stopping rules and two starting rules. This paper is unique since most papers do not address the issue of the server stopping when there are no customers in the system.

Ibe and Trivedi [86] considered the finite-population model in which the service time and the switch-over time are given by an exponential distribution. Their solution is based on the generalized stochastic Petri nets (GSPN). Based on the one-to-one correspondence between the reachability graph of Petri nets and the continuous time Markov chains a set of linear equations for the steady state probabilities of the polling system were obtained. Using successive over-relaxation and the Gauss-Seidel method, Ibe and Trivedi [86] computed the steady state probabilities from which they obtained the mean waiting time using Little' s law. However, a major drawback of GSPN is its storage requirements. This is because GSPN requires generating all the states of the reachability graph which can be very large for a large number of queues. For example, for three-queue polling system with single buffer capacity and population size equal to 10, the number of states in the Markov chain is 7623 and the number of non-zeros in the transition matrix of the Markov chain is about 28000 (See Table IX in [86]).

Another drawback of using GSPN is the population size. Most polling systems have a large, if not infinite, population size, however, in order to generate the reachability graph the population size has to be finite.

Using embedded Markov chains, Ganz and Chlamtac [70] analyzed a polling system similar to that of Ibe and Trivedi [86]. However, in their system time is slotted and each station generates a message in each time slot with probability r. The state space is defined as the total number of customers in the system at each embedded point (beginning of each time slot). This state space allowed them to limit the number of equations to $N.L+1$, where N is the number of queues and L is the buffer size. Solving the $N.L+1$ equations yields the steady state probability vector of the number of customers in the system. The individual queue length distribution was obtained using the notion of "n-indistinguishable balls" and "m-distinguishable urns" where the capacity of each urn is equal to the buffer capacity. Although, the authors presented an accurate and simple way to obtain the mean waiting time for slotted communication systems, the model is very limited in several aspects (e.g. finite population, messages arrival process).

Although [70, 86] presented models for finite population polling systems, care must be exercised in using these models. This is because the number of customers already in the system at any point in time affects the number of potential new customers arriving to the system (i.e. the pool of potential customers gets smaller as the number of customers in the queueing system increases).

Lee [110] analyzed the M/G/1/K queue with vacation periods using the embedded Markov chain approach. His results were used to study the performance of a cyclic polling system with an exhaustive service policy, where each queue has a finite capacity. He also considered the M/G/1/K queue with vacation periods and exhaustive limited service discipline in [111]. The LST of the busy period and cycle time were obtained using an embedded Markov chain. The waiting time distribution, blocking probability, and queue length were obtained by the method of supplementary vari-

ables and sample biasing techniques. Later, Kofman [103] used the decomposition results of the M/G/1/K queue to obtain the blocking probability, throughput and the mean waiting time for a polling system with exhaustive, gated, and limited service disciplines for finite buffer capacity polling systems. Takagi [168] used the results of Lee [110] and Courtois [44] for the M/G/1/K queue to analyze finite buffer capacity polling systems. Jung and Un [90] used the buffer occupancy method to analyze the finite-buffer polling system with the exhaustive service discipline.

A shared medium switch for an ATM network with input and output links was analyzed by Zaghloul and Perros [184, 185] and Hong, Perros, and Yamashita [82]. Both the input and the output links have finite capacity waiting room. Note that in [184] there are N input links and a single output link and in [185, 82] there are N input links and M output links. The switch-over time is equal to zero. Messages from the input links are generated according to the interrupted Bernoulli process (IBP) and are routed over a high-speed medium (parallel bus) to the output links. The service time is deterministic and given by one unit time. However, because the output links have a finite capacity, the service time is adjusted to account for blocking. This is because a blocked customer is, in effect, depriving the next customer in the queue from receiving service. The service time is also adjusted to account for bus contention. The adjusted service time is called effective service time and each queue is then analyzed separately under three service disciplines: Time Division Multiplexing (TDM), cyclic, and random polling. Note that for TDM, only the blocking probability affects the service time since the server visits the queues at specific time periods. Each queue is then analyzed as an embedded Markov chain and the steady state probability vector is obtained using the Gauss-Seidel iterative procedure. Performance measures such as the queue length distribution, system throughput, and the blocking probability were obtained. A similar model with bursty arrival process was analyzed by Jou, Nillson and Lai [89].

Notice that under the TDM service discipline the server may be idle while cus-

tomers are waiting at other queues. This observation is confirmed by the numerical results in [185] (e.g. blocking probability under the cyclic service discipline is better than under TDM). Although IBP is a good approximation for the cells arrival process, MAP is a better representation for the arrival process in B-ISDN as suggested by Blondia [18, 19] and Blondia and Theimer [20].

Recently, Rubin and Wu [145] used a variant of the M/G/1 queue with vacation to study the performance of fiber distributed data interface (FDDI) timed-token rings. Each station in the network is approximated by a single server queue with vacations. Each station is assumed to generate messages according to a Poisson process with a random number of fixed size segments (batch Poisson input). The transmission time of one segment is deterministic and is equal to one time slot. The system is thus divided into time slots of equal size. The transmission time of a segment is given by B_n, where $\{B_n, n \geq 1\}$ forms a sequence of independent and identically distributed (i.i.d.) r.v. The service time has a discrete general distribution given by $b(i) = P(B_n = i)$, $i = 1, \ldots, B_{max}$; $B_{max} < \infty$. Similar to the service time distribution, Rubin and Wu have defined a vacation time distribution given by $v(i) = P(V_n = i)$, $i = 1, \ldots, V_{max}$; $V_{max} < \infty$, and a visit time distribution given by $g(i) = P(G_n = i)$, $i = 1, \ldots, G_{max}$; $G_{max} < \infty$. Each station in the ring is analyzed based on an embedded Markov chain, where the embedded points are the instants of packet departure and token arrival. A set of balance equations is then derived and, based on the boundary probabilities of token arrival and departure, an iterative procedure to compute the limiting state distribution is obtained. The queue length distribution at an arbitrary time is then obtained using the supplementary variables technique. In addition, the packet delay distribution is obtained based on decomposing the delay into two independent distributions: 1) the forward recurrence time distribution representing the instant of arrival and the next embedded instant and 2) the residual packet delay distribution given by the time from the embedded instant until the transmission of the tagged packet. This distribution, residual packet delay, represents the service time of

the packets enqueued ahead of the tagged message. The approximate vacation time distribution of a queue is constructed by convolving the switch-over time distribution and the visit period distribution of the other queues. The vacation period distribution is computed in accordance with the traffic intensity of the other queues. The analysis presented in [145] was used to approximate the behavior of the FDDI timed-token ring. Although the results compare very well with simulation, this cyclic polling model is restricted to networks in which the arrival process can be approximated by a batch Poisson process. Note also that because there is no hard time limit, the visit period can exceed the maximum time allocated for a given queue. This can be a problem in an asymmetric system with long packets (service time skewed toward B_{max}).

2.5.2 Infinite-Buffer Systems

Because of the difficulties in modeling finite buffer systems, several researchers assumed the buffer size to be infinite. This simplifies the analysis somewhat and makes the problem mathematically tractable. Similar to the previous section, unless otherwise mentioned, the input process is Markovian, the service time and the switch-over time, if any, are generally distributed.

Carsten, Newhall and Posner [32] pioneered the station time method and used it for the analysis of scan time in non-symmetric polling systems with exhaustive service discipline. Later, Ferguson and Aminetzah [60] derived the mean waiting time for non-symmetric polling systems using the station time method for the gated service discipline.

In a widely referenced monograph, Takagi [165] considered cyclic polling systems with infinite buffers and exhaustive or gated service disciplines. His solution is based on the buffer occupancy method. He defined the joint marginal generating function F_i of the number of messages at queue i at polling instants. He then related F_i to F_{i+1} and obtained analytical expressions for the first and second moments of

the queue length. For symmetric systems (arrival rate, switch-over time and service time are independent of the queue's number) a closed form solution was obtained for the first and second moments of the queue length. For asymmetric systems (arrival rate, switch-over time, and service time depend on the queue's number) the second moments of the queues' length are obtained by solving numerically a set of $O(N^3)$ equations. Takagi [165] obtained the LST of queue length distribution by defining regeneration points as the points when queue one is polled and all the queues are empty. The LST of the waiting time distribution was obtained from the relationship between the LST of the queue length and busy period distributions. For the limited service policy, Takagi [165] considered a symmetric cyclic polling system and obtained the mean queue length and the mean waiting time using the buffer occupancy approach.

It is shown in Takagi [165] that for the discrete-time model, for the same total utilization, the mean waiting time at queue one, in the case where all utilization is concentrated at queue one, is smaller than the mean waiting time in the symmetric polling system for the exhaustive and gated service disciplines. He showed also that for symmetric cyclic polling systems the exhaustive service discipline has the least mean waiting time and the limited service policy has the largest mean waiting time i.e.

$$E(W)|_{exhaustive} \leq E(W)|_{gated} \leq E(W)|_{limited}.$$

Because of the complexity associated with obtaining the mean waiting time for asymmetric polling systems, Bux and Truong [31] considered each queue in the polling system as a M/G/1 queue with service and vacation periods. It is known that for the M/G/1 queue with vacation periods, the mean waiting time depends on the mean and variance of the vacation period. The mean of the vacation period was obtained from the mean of the cycle time and service period. The variance of the vacation period was obtained by using a heuristic extrapolation from the case of $N = 2$.

Another approximation is by Srinivasan [157] and it is for the 1-limited service discipline. Srinivasan' s approximation is based on the analysis of the cycle time

and the vacation period. Later, Takine and Hasegawa [173] derived the LST of the waiting time distribution for a cyclic polling system with finite source model and 1-limited service discipline. Sidi *et al.* [156] analyzed a polling system in which served customers may leave the system or be routed to another queue. Using the buffer occupancy method, they obtained the queue lengths distribution, the mean waiting time in the queues, and the mean waiting time of customers that follow a specific path in the network. Their analysis is for the gated and exhaustive service disciplines. They have also extended the pseudo-conservation law of Boxma [22] to their polling model.

An alternative solution approach, based on the power-series algorithm (PSA), for infinite buffer polling systems was proposed. This method is based on the power series expansions of the state probabilities and the moments of the queue length distribution as functions of the load in a system with light traffic. It was used in [13, 14, 15, 16] to analyze polling systems with and without switch-over time. Although PSA is an additional tool for the analysis of polling systems it is limited to systems with Poisson input. For the K-limited service discipline, PSA is limited to systems with moderate value of K as shown in [15]. As K becomes large more terms of the power series are needed which results in more memory requirements and large computational time.

Due to the limitation of the Poisson process with single arrivals, several researchers attempted to obtain performance measures for polling system with batch Poisson process and renewal input process. First, Kuehn [106] considered a cyclic polling system with batch Poisson arrivals and non-exhaustive service discipline. He used the concept of conditional cycle times to derive the LST of the delay distribution through the embedded Markov chain approach. Later, Levy and Sidi [120] analyzed a polling system with simultaneous arrivals. They used the buffer occupancy method to obtain the mean waiting time under the exhaustive and gated service disciplines. More recently, for a polling system with gated, exhaustive, globally gated or time-limited service discipline and renewal input processes, Altman and Kofman [7] obtained upper bounds for the cycle time, the total amount of work in a station at

time t, and the amount of work that leaves the system from the polled station. Their analysis is based on characterizing the inputs by bounds on the average arrival rate and burstiness and uses previous results obtained for polling systems with Poisson inputs.

Notice that most papers considered until now use the exhaustive or gated service discipline. The remaining part of this section considers polling systems with a variant of the time-limited service discipline.

Leung [113] obtained the queue length distribution for a polling system with exponentially time-limited service discipline. He used the results of Leung and Eisenberg [112, 114]. Using the discrete Fourier transform, Takagi and Leung [171] analyzed the discrete time single server queueing system with time-limited service. In this model, the arrival process and the service time distribution are defined in terms of two generating functions.

Recently, de Souza *et al.* [47] considered a polling system with exponential service time distribution with infinite (or finite) buffer capacity. The service discipline is of the time limited and can be either preemptive or non-preemptive. In the preemptive case, once the visit period reaches the time limit an on-going service is interrupted and the preempted customer is returned to the line of the waiting customers and its service time is re-sampled (i.e. identical to a customer who received no service). In the non-preemptive case, the server does not interrupt an on-going service to switch to an other queue. de Souza *et al.* [47] presented a solution approach that can be applied to a number of service disciplines (e.g exhaustive time-limited, gated time-limited, etc.). However, they presented only the exhaustive time-limited service discipline in detail. Their analysis is based on studying the embedded Markov chains defined at the sequence of the points of server arrival and departure from each polling station. The joint queue length distributions of these two embedded Markov chains are obtained based on the uniformization or randomization technique. Based on the results of Markov chains with rewards some time average results are obtained. Note that,

although their solution approach can be used to a wide range of service disciplines, it is limited to systems in which service time and inter-arrival time are exponentially distributed. Notice also that their solution approach yields only the joint queue length distributions at server arrival and departure points from which, and based on Markov chains with rewards, they were able to obtain time average measures (e.g average delay). Thus, the difficulty associated with the analysis of limited service discipline can be alleviated by the use of the M/G/1 queue with vacation periods as a basis for an iterative procedure.

Two vacation models for an M/G/1 queue with constant time-limited service or vacation-dependent time-limited service were proposed by Leung and Lucantoni [115] for the performance analysis of stations in a timed-token network. For the time-limited service discipline, a queue is visited for a maximum time period. For the vacation-dependent time-limited service discipline, if the previous cycle time exceeded the queue pre-specified cycle time threshold then that queue receives no service in the current cycle, else the queue is served in the same manner as in the case of a time-limited service discipline. Under both service disciplines a customer service is not interrupted if the queue visit-time limit is reached (i.e. non-preemptive service discipline). In order to analyze these models, the time-limit is approximated by a number of time stages where each stage is exponentially distributed. Thus, the visit-time limit can be characterized by an Erlangian distribution instead of being deterministic.

The time-limited service discipline was modeled as a Markov chain defined at the points of customer departure from which the steady state probability vector is obtained. The computation of the steady state probability vector requires the inversion of the probability generating functions (PGF) of the arrival process, the service time distribution and the vacation period distribution. In order to get around inverting PGFs, one can represent the service time and the vacation period by phase distributions. This leads to a simple recursive approach to compute the steady state

probabilities. For the vacation-dependent time-limited service the Markov chain is defined at the point of customer departure. The target cycle time is also approximated by a number of time stages.

Since both models in [115] are of the M/G/1 paradigm presented in [134], Leung and Lucantoni [115] used the matrix analytic approach to solve for the queue length distribution. Because the block matrices are of infinite dimension, an appropriate truncation point is necessary to use the matrix analytic approach. Since the time-limit is approximated by time stages, the numerical results depend on the number of stages used. The numerical experimentation by the authors suggests using a moderate number of stages (about 16 stages). However, it is not clear whether this number of stages will hold for other service time distributions since the examples presented are for exponential service time distributions. Also, the authors did not present how to obtain the vacation period distribution in the case of polling systems. Although the presented models are good tools for the performance analysis of timed-token networks, they can not be used, as stated by the authors, in a network where the characteristic of traffic under consideration, the frame arrival process may be non-Poisson or even non-renewal. The Markovian arrival process has been shown to be effective in capturing the correlations among frame arrivals of voice and video traffic.

In a somewhat related model, Chiarawongse *et al.* [37] considered the M/G/1 queue with vacations under the time-limited, cycle time-limited, and the cycle time-limited with accumulated lateness service disciplines. Their analysis is based on the matrix-analytic approach presented in [134] and yields the queue length distribution.

The manufacturing automation protocol is based on token bus and token ring network. In this protocol, each station in the network has two timers for controlling visit period length. The first timer controls the token holding time (THT) and the second controls the rotation time (cycle time) (TRT). Yue and Brooks [183] approximated the behavior of this protocol for a symmetric and an asymmetric system. For the symmetric case, all the stations have a THT with no target rotation time, the

mean waiting time was obtained based on k-limited service discipline approximation due to Fuhrmann [66]. Because of the non-preemptive nature of the THT, the visit period is actually longer than the THT. In order to obtain the mean visit period, Yue and Brooks [183] used an excess holding time variable which they derive using renewal theory and the inversion of the LST of the service time distribution. For the asymmetric case, they analyzed a network with nine stations having only THT, and one station with TRT and THT. The mean of the visit period for the TRT station is obtained empirically based on some simulation runs which is then used to computing the mean waiting time. Thus, the models presented in this paper are for specific configurations. Although, the symmetric case can be used for a large number of queues, it is limited in the sense that only Poisson arrival is allowed. The asymmetric approximation is limited to the network given in [183].

Lee and Sengupta [109] considered a polling system with limited service and reservation. For this service policy, each queue makes a reservation for the number of services required for cycle $j+1$ after receiving service in cycle j. However, the minimum number of services must be at least one and at most M. Their solution is based on the concept of a single queue with visit and vacation periods. Their iterative procedure assumes that the vacation period of queue 1 in iteration $(k+1)$ is given by the mixture of the following two terms: $\sum_{i=2}^{N} S_i^{(k)}$ with probability $(1-P)$, where $S_i^{(k)}$ is the service period for queue i in iteration k, $S_i^{(k)}$ are independently identically random variables (sum of independent service periods), and $(N-1)S^{(k)}$ with probability P, where $S^{(k)}$ is a generic service period (sum of dependent service periods) and N is the number of queues. The results obtained consist of the queue length and sojourn-time distributions. This polling system was used to model satellite communication. A similar model was considered by Tran-Gia and Dittmann [178]. They used the decomposition approach along with the results of the M/G/1 queue with vacation to obtain packet transfer time for a cyclic reservation multiple access protocol.

2.6 Priority Based Polling Systems

When using the term priority polling it is important to distinguish between priority at the station level and priority at the customer level. Priority at the station level means that the order in which the server visits the stations is based on the station's priority level. Its application is in the area of duplex transmission and central controllers. Priority at the customer level means that each station in the polling system can have more than one type of customer. Once a station is polled then enqueued customers are served according to their priority level within the station. Priority based polling has applications in the area of integrated services.

Fournier and Rosberg [61] analyzed a polling system with multiple priorities at each queue. They considered several service disciplines and used the stochastic decomposition law for single server queue with vacations to obtain the pseudo-conservation law for the mean waiting times. Similar results were obtained by Shimogawa and Takahashi [155].

Manfield [129] considered a polling system with two way data traffic. In this polling system, priority is given to messages going from the central controller (server) to the queues. The system is analyzed by considering $(N + 1)$ queues, where N queues are dedicated to the incoming messages (messages going from the queues to the server), and the $(N+1)$st queue is dedicated for the outgoing messages (messages going from the server to the queues). The mean delay for the outgoing messages is exact and for the incoming messages is an approximation. For a similar network, Giannakouros and Laloux [73] used the pseudo-conservation law to obtain the mean waiting time under the exhaustive, gated and 1-limited. They also obtained conservation laws for the case of mixtures of the three service disciplines. In a related model, Stavrakakis [160] derived tight bounds for packet delay in an alternating queue where one queue hosts the high priority packets and the other hosts the low priority packets.

Karvelas and Garcia [91] modeled an integrated packet voice/data token-passing

ring as a polling system. In order to limit the cycle time, they considered the 1-limited service discipline. Each station in the network has a single buffer for voice messages and infinite buffer for data messages. For this polling system, voice and data packets are assumed to arrive according to two i.i.d. batch Poisson processes. The service time of a packet is given by a general distribution. Because of delay constraints voice packets have higher priority than data packets at the station level which implies that data packets are transmitted only when the high priority buffer at the station level is empty. By extending the cycle time analysis presented in [106] for a polling system with single priority, Karvelas and Garcia were able to obtain the mean waiting time for the voice and data packets. It is important to notice here that although their results match very well with simulation, it was shown elsewhere (e.g [79, 158]) that voice and data traffic is best characterized by MMPP.

The proposed service discipline of Karvelas and Garcia [91] can cause large delays for data packets when the arrival rate of the voice packets is very high (for a given station). This is because in every visit the server may have to serve the high priority message, in this case voice packet, and leave the low priority message behind which are the data packets for this integrated network. Thus, limiting the cycle time may not be the best alternative to reduce the waiting time in polling systems where high priority messages have high arrival rates.

Pang and Tobagi [139] obtained the throughput for a polling system with heavy traffic with a cycle-dependent mechanism which is employed in IEEE 802.4 token bus and the FDDI token ring standard. This service discipline enhances the performance measures of real-time applications. By deriving bounds on the cycle length, the authors were able to obtain approximate results for the throughput. Later, Hong [83] obtained the mean waiting time for cycle-dependent polling systems with 1-limited service discipline. He used the results of Kuehn [106] and the notion of effective service times.

Gianini and Manfield [72] considered a polling system where each queue in the

system has two priority levels. They considered the case of exhaustive and gated (at the priority level) service disciplines. In these service disciplines, a queue is polled at its low priority level only if there are no high priority messages anywhere in the system. Their method of solution is based on defining a low priority poll busy period, a high priority poll busy period, and the moment generating function of the queue length at polling instants. They derived the first and second moment of the queue length and the waiting time for the high and low priority messages. For the same polling system, Frigui, Stone and Alfa [65] used Bux and Truong [31] approximation of the vacation period and the results of the M/G/1 queue with priority [154, 92] to obtain the mean waiting time for the high and low priority messages under the exhaustive service discipline.

Tsai and Rubin [179] obtained exact results for a polling system with two priority levels with exhaustive or limited service disciplines. Their system is different from that of Gianini and Manfield [72] since they considered the case where each queue has a single buffer high priority queue and an infinite buffer low priority queue. A queue can seize the server at low priority only if all high priority buffers are empty. During a low priority poll with exhaustive service policy, the server continues to transmit messages until both queues are empty. Thus, in a low priority poll, all messages found in the queue and those that arrive (high or low) during the service period are transmitted in the current cycle. If a queue seizes the server at a high priority level, then only the high priority message is transmitted. For the limited service policy, during a high (low) priority poll the server transmits one high (low) priority message. Later, Tsai and Rubin [180] extended their results to the case where each priority has an infinite buffer capacity. The service discipline they considered is such that high priority are served exhaustively and low priority are number limited. The analysis in [180] takes advantage of the symmetry of the polling system and is based on the cycle time analysis.

The model in [72] was later generalized by Poko *et al.* [75]. They considered

a polling system with multiple priorities at each station. However, they used the 1-limited service discipline. By assuming that customers are served in the cycle in which they arrive (clearly, this assumption would be invalid in systems where low priority messages have low arrival rates and high priority messages have high arrival rates). Poko *et al.* were able to obtain the mean waiting time for each priority.

2.7 Non-Cyclic Polling

Because the optimization problem of polling systems is not analytically tractable, many researchers suggested optimizing the performance measures of a queue using alternative polling orders. The next two Sections discuss table and random polling systems, respectively.

2.7.1 Table Polling

Although table polling is periodic, a pre-specified table dictates the rotation of the server among the queues, we consider it to be non-cyclic in the sense that the vacation period of a given queue depends on the position from which the server leaves the queue in the table. In such a system, we have pseudo-cycle time and pseudo-station. A pseudo-cycle time is the time between polls of the same station in the table. Notice that each station may have more than one pseudo-cycle and, in most cases, the pseudo-cycles have different distributions. Furthermore, for each pseudo-cycle we can define a pseudo-station. A pseudo-station is a fictitious station that has the same parameters as a station that appears more than once in the table.

Among the first attempts to solve multi-queue systems with table polling is the work of Eisenberg [52]. He considered a table polling system with exhaustive service discipline. He obtained the (LST) of the inter-visit time and the LST of the waiting time at queue i. He considered four embedded Markov chains: 1) service beginning, 2) service completion, 3) beginning of queue visit, and 4) end of queue visit. His solution

relies on the relationship between the probabilities of the embedded Markov chains mentioned above (for instance, the beginning of a queue visit must coincide with a service beginning). The notion of pseudo-station and pseudo-cycle were used by Baker [12] to obtain the mean waiting time for a table polling systems with exhaustive service discipline and by Choudhury [38] to obtain the mean waiting time for the gated service discipline. Chang and Hwang [33] used the embedded Markov chain and derived a new recursive method to compute the moments of the pseudo-cycle time. The moments of the pseudo-cycle time are then used to obtain the mean waiting time for general polling systems with gated service discipline. Altman, Khamisy and Yechiali [6] derived the mean waiting time for elevator polling systems with a globally gated service discipline. They showed that the mean waiting time is identical for all the queues even for the non-symmetric case. However, due to the difficulty associated with getting performance measures for table polling systems, many researchers used approximate methods to derive bounds for the mean waiting time.

Federgruen and Katalan [59] used the decomposition results of Fuhrmann and Cooper [68] to approximate the queue length and waiting time distributions in general polling systems with exhaustive, gated, or a mix of exhaustive and gated service disciplines. And Boxma *et al.* [25] extended the conservation laws to polling tables with batch input process and deterministic service times. Recently, Frigui and Alfa [63] used the pseudo-station and pseudo-cycle and approximated the time-limited by the K-limited service discipline to obtain the mean waiting time in table polling with time-limited service discipline.

2.7.2 Random Polling

Performance measures for the exhaustive, gated, and limited service disciplines were derived by Kleinrock and Levy [100] for the case of random polling systems using the same analysis as Takagi [165]. For symmetric random polling systems the exhaustive service discipline has the least mean waiting time and the limited service discipline

has the largest mean waiting time i.e.

$$E(W)|_{exhaustive} \leq E(W)|_{gated} \leq E(W)|_{limited}.$$

A comparison between cyclic and random polling by Kleinrock and Levy [100] showed that for the same system parameters and service discipline cyclic polling yields a lower mean waiting time than random polling.

In addition to random polling, four other probabilistic models were analyzed in the literature. Frist, Servi [153] derived the first two moments of the busy period for the M/G/1 queue with Bernoulli schedule. These moments are then used to estimate the mean waiting time for each queue in a polling system. In a later paper, Tedijanto [176] analyzed a polling system with Bernoulli schedule.

The second model, the probabilistic limited service discipline, was analyzed by Leung [112] by defining four embedded Markov chains as in Eisenberg [52]. The queue length distribution is obtained via the discrete Fourier transform. From the mean queue length, Leung [112] obtained the mean waiting time using Little' s law. However, since the solution is based on a numerical approach, the memory and CPU time are exponential functions of the number of queues. Hence, under heavy loads only relatively small systems can be solved.

Thirdly, Levy [116] introduced the so-called binomial-gated service discipline. This service discipline would allow the designer to prioritize the queues by choosing high p_i for high priority queues. Using the buffer occupancy approach, he obtained closed form solution for the mean queue lengths for symmetric systems. However, for asymmetric polling systems the mean queue lengths are obtained by solving numerically a set of N^3 equations. As presented by Levy [116], the binomial-gated service discipline is an effective way to prioritize the queues. This can be achieved by minimizing the waiting costs when the service times and cost per unit of waiting time are identical in all the queues.

Lastly, Lye and Seah [126] proposed a Markovian polling scheme to reduce access delay for a network with a large number of stations. Later, Chung, Un and Jung [39]

Note To Users

The original document received by UMI contained pages with poor print. Pages were filmed as received.

53

This reproduction is the best copy available.

UMI

Among the first researchers to work on conservation laws for polling systems was Everitt [56]. He extended the conservation law introduced in [96] to the exhaustive k-limited service discipline for symmetric polling systems. In a sequel paper, Everitt [57] summarized the pseudo-conservation laws for cyclic service systems with exhaustive, gated, and limited service disciplines. He also derived a new result for the exhaustive limited service policy. Fuhrmann [66] used the decomposition results of the M/G/1 queue to establish an upper bound for the mean waiting time in symmetric cyclic polling systems. Later, Fuhrmann and Wang [69] derived upper bounds for the exhaustive k-limited and gated k-limited service disciplines for asymmetric polling systems.

The pseudo-conservation laws were used by Boxma and Meister [28] to approximate the mean waiting time of non-exhaustive service disciplines (i.e. serve at most one customer) for cyclic polling systems. Chang and Sandhu [35, 36] used the pseudo-conservation laws and the concept of conditional cycle time to approximate the mean waiting time for the k_i-limited service discipline. Boxma and Groenendijk [23] derived pseudo-conservation laws for polling systems. They extended the work of Watson [181] for the case of exhaustive, gated, and 1-limited to polling systems with mixed service disciplines. Later, Levy and Sidi [118] extended their results to polling systems with correlated arrivals. In [24] pseudo-conservation laws for the discrete-time model were obtained. And Boxma and Weststrate [29] obtained pseudo-conservation laws for Markovian polling systems. Lu and Lin [124] used pseudo-conservation law to analyze an FDDI network. de Moraes and Fuhrmann [46] approximated the mean waiting time for a polling system with batch Poisson input via the pseudo-conservation law. For a more general model, Takahashi and Kumar [172] derived pseudo-conservation law for a polling system with priority in which each priority has its own service strategy.

Groenendijk [76] obtained approximate results for cyclic service systems with mixed service strategies (i.e. exhaustive, gated, and 1-limited). His analysis is based

on the pseudo-conservation laws and the exact results of alternating queue systems where one queue is served exhaustively and the other is 1-limited.

2.10 Stability Papers

Stability, monotonicity, and invariant quantities are fundamental issues of polling systems. They were considered by several authors [8, 45, 71, 106, 121, 186]. In this section, we focus on the literature that formally establishes some of these important relationships.

One of the most important results concerns the cycle time. It is shown in the literatures (see, e.g. Kuehn [106]) that the distribution of the cycle time is different for different queues. However, the mean cycle time, C, is identical for all the queues and depends only on the total switch-over time, R, and the polling system utilization ρ. The mean cycle time is given by:

$$C = \frac{R}{1-\rho}.$$

Levy *et al.* [121] used sample path analysis to compare the efficiency of the exhaustive and gated type service disciplines in polling systems based on the amount of unfinished work found in the system at any time. They established that the exhaustive service discipline is the most efficient one in the sense that the amount of unfinished work found in the system by an arriving customer is the smallest. Their studies did not consider the case of asymmetric systems.

Fuhrmann [67] provided decomposition results for polling systems with Poisson input, general service time distribution, constant switch-over time and gated or exhaustive service discipline. He showed that the number of customers in the system is given by two sets. The first is given by the number in a polling system with no switch-over time. The second set consists of those that arrive during the switch-over time and their descendant (i.e. customers that arrive during the service time of those

who arrived during the switch-over time period).

Servi [152] established a relationship for 1) the maximum number of queues that can use a server without the system becoming saturated, 2) the relative number of customers served per cycle in two queues, and 3) the relative lengths of the busy period for two queues. He also studied the effect of initiating low priority jobs, such as maintenance, on the performance of a polling system.

Altman *et al.* [8] considered the stability of a cyclic polling system with general service discipline (e.g. exhaustive, gated etc). In this polling system, customers arrive according to the Poisson process. The service time and the switch-over time are of the general distribution type. Using Foster' s criterion, Altman *et al.* obtained sufficient conditions for the ergodicity and geometric ergodicity of the queue length distribution. They have also shown that the queue lengths, the cycle times, and the inter-visit times are stochastically increasing in: 1) arrival rates, 2) service times, 3) walking times, and 4) number of queues. Lastly, they showed that the mean cycle time, the mean inter-visit time, and the mean station time in the steady state are invariant under general service disciplines and general stationary arrival and service processes. This is a very important result, since in many instances, especially in the case of token rings with target rotation time, the mean of the cycle time is used as a performance measure. Later, Fricker and Jaibi [62] derived the stability condition for periodic (cyclic or table) polling models with a mixture of service disciplines. Each queue can be served according to more than one service discipline in the case of table polling. They showed that

$$\rho + \max_{1 \le j \le N} (\lambda_j / K_j) R < 1$$

is a necessary and sufficient condition for stability (the ratio λ_j / K_j is equal to infinity if the queue is served exhaustively, $K_j = \infty$).

Yaron and Sidi [182] established two bounds for communication networks and showed that the bounds decay exponentially. These bounds are then used to study the performance of a multiplexer with several input traffic streams. Chang [34] es-

tablished stability conditions for queueing networks. He introduced a new traffic characterization, minimum envelop rate.

2.11 Remarks

Although we presented over 150 articles, this literature review is by no means exhaustive. However, one can make two observations. First, we remark that most addressed problems in the literature focus on polling systems with:

1. Poisson input.
2. General switch-over time.
3. General service time distribution.
4. Exhaustive and gated service discipline. Recently, the limited service discipline is gaining more attention.

Although these models can be used to compute performance measures for homogeneous network, they are of little use for integrated services networks. This is attributed to the limitation of the Poisson process. Therefore, future research should focus on using a more versatile arrival process like the Markovian arrival process. Second, and somehow a more difficult problem to answer, is the issue of optimization (i.e. given an arrival process, a service time distribution, and the number of queues how should the queues be visited and for how long in order to minimize, say, the weighted sum of the mean waiting times).

In this thesis, we attempt to answer the first question. The polling system we consider is one with MAP input, phase type service distribution, exhaustive time-limited service discipline and zero switch-over time.

CHAPTER 3

TIME-LIMITED CYCLIC POLLING

3.1 Introduction

We are about to embark on a detailed study of a cyclic polling system. As outlined in Section 1.6, our objective is to develop an iterative procedure to compute the mean waiting time for a cyclic polling system. The cyclic polling system we consider in this Chapter consists of:

- Q queues $1 < Q < \infty$ (all of the queues have either finite or infinite capacity).
- Arrival to queue i, $i = 1, 2, \ldots Q$, occurs according to the MAP with representation $D_{0,i}$ and $D_{1,i}$.
- Service time of customers of queue i, $i = 1, 2, \ldots Q$, is a phase type distribution with representation $(\boldsymbol{\beta}_i, S_i)$.
- Service discipline to queue i, $i = 1, 2, \ldots Q$, is exhaustive time-limited with time limit T_i.
- The switch-over time is equal to zero.

However, before we outline our solution approach, for completeness, in the next two sections we introduce to the reader the arrival process and the service time distribution. In elaborating further about the arrival and service processes, we will not include the suffix for the queue to save on use of notation.

3.2 Phase Distribution

Consider an $(m+1)$ state Markov chain with transition probability matrix given by

$$P = \begin{bmatrix} S & \mathbf{S}^\circ \\ 0 & 1 \end{bmatrix},$$

where, the square matrix S is of order m and $\mathbf{S}^\circ = \mathbf{e} - S\mathbf{e}$ is an $m \times 1$ column vector and $\mathbf{e}$ is a column vector of 1s. The absorption into state $m+1$ from any state is guaranteed if the inverse $(I-S)^{-1}$ exists. Let $(\boldsymbol{\alpha}, \alpha_{m+1})$ be the initial probability vector of the Markov chain. The probability density, on the non-negative integers, of the time until absorption is given by:

$$p_0 = \alpha_{m+1} \quad p_k = \boldsymbol{\alpha} S^{k-1} \mathbf{S}^\circ, \quad k \geq 1.$$

The mean of this distribution is given by:

$$\mu = \boldsymbol{\alpha}(I-S)^{-1}\mathbf{e}.$$

The probability density $\{p_k\}$ is said to be of phase type. The pair $(\boldsymbol{\alpha}, S)$ is the representation of this phase type distribution. In this thesis, each customer takes at least one unit of time for service, therefore, the variable $\alpha_{m+1} = 0$.

The phase type distribution can be used to represent servers in series. For example, consider a service station with two servers in series. In addition, assume that we can serve only one customer at a time. Then, if the sojourn time in the first server is given by a geometric distribution with parameter α and the sojourn time in the second server is given by β, then the service time can be represented by the following phase type distribution:

$$S = \begin{bmatrix} \alpha & 1-\alpha \\ 0 & \beta \end{bmatrix}.$$

The initial probability vector is given by [1 0].

The phase type distribution can also be used to represent servers in parallel. For example, in the above example instead of the servers being in series, they are in parallel now. In addition suppose that the probability that a customer receives service from the first server is 0.5 and from the second server is 0.5. This can be represented by the following phase type distribution:

$$S = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}.$$

The initial probability vector is given by [0.5 0.5].

Details about the phase type distribution can be found in Neuts [133, Chap. 2] for continuous time and in Neuts [131] for discrete time.

3.3 Markovian Arrival Process

In order to discuss the discrete Markovian arrival process, we first consider the Bernoulli arrival process. Let the rate of the Bernoulli process be α. $N(t)$ is the number of arrivals between 0 and t. The process $N(t)$ is then a Markov chain on the state space $\{i, i \geq 0\}$ with transition probability matrix P of the form

$$P = \begin{vmatrix} d_0 & d_1 & \dots & & \\ & d_0 & d_1 & \dots & \\ & & d_0 & d_1 & \dots \\ & & & \ddots & \ddots \end{vmatrix},$$

where, $d_0 = 1 - \alpha$ and $d_1 = \alpha$. After a geometric sojourn time in state i, the process jumps to state $i+1$ with probability α where the transition corresponds to an arrival.

The discrete Markovian arrival process is constructed to allow for non-geometric times between the arrivals. Consider a 2-dimensional Markov chain $\{N(t), J(t)\}$ on

the state space $\{(i,j) : i \geq 0;\ 1 \leq j \leq n\}$ with transition probability P of the form

$$P = \begin{vmatrix} D_0 & D_1 & & & \dots \\ & D_0 & D_1 & & \dots \\ & & D_0 & D_1 & \dots \\ & & & \ddots & \dots \end{vmatrix}$$

where $D_j; j = 0,\ 1$ are sub-stochastic matrices, and $(I - D_0)$ is a non-singular matrix. We, also, denote by D,

$$D = D_0 + D_1,$$

an irreducible stochastic matrix of order n. In this Markov chain, $N(t)$ represents a counting variable and $J(t)$ represents a state phase variable. This Markov chain represents a discrete arrival process. The transition from state (i,j) to state $(i+1,l)$ where $1 \leq j,l \leq n$ correspond to an arrival. Since $(I - D_0)$ is non-singular, then the sojourn time in the set space $\{(i,j) : i \geq 0; 1 \leq j \leq n\}$ is finite which implies that the arrival process does not terminate. The stationary vector $\boldsymbol{\pi}$ of the Markov chain described by D satisfies the equations

$$\boldsymbol{\pi} D = \boldsymbol{\pi} \quad \text{and} \quad \boldsymbol{\pi}\mathbf{e} = 1 \tag{3.1}$$

where $\boldsymbol{\pi} D_1 \mathbf{e}$ is the probability that, in the stationary version of the arrival process, there is an arrival at an arbitrary time point. Correspondingly, $\lambda = \boldsymbol{\pi} D_1 \mathbf{e}$ is the expected number of arrivals per unit time and also is referred to as the fundamental rate of the process.

The Markovian arrival process was introduced by Neuts [132] and was later generalized by Lucantoni [125]. Several well known arrival processes can be represented by MAP. For example,

Discrete Phase Distribution: The phase type renewal process with representation $(\boldsymbol{\beta}, S)$, introduced in Section 3.2, is a MAP with $D_0 = S$ and $D_1 = \mathbf{S}^\circ \boldsymbol{\beta}$, where $\mathbf{S}^\circ = \mathbf{e} - S\mathbf{e}$.

The Markov Modulated Bernoulli Process: MMBP is represented by two $m \times m$ matrices D and Δ. The matrix D is irreducible, stochastic and governs the transition of the underlying Markov Chain. The matrix Δ is a diagonal matrix with elements $0 < p_i < 1$, for $i = 1, \ldots m$. MMBP can be represented by MAP with $D_0 = D(I - \Delta)$ and $D_1 = D\Delta$.

The Interrupted Bernoulli Process: IBP is an arrival process with an active period with a geometric distribution having a parameter α and an idle period with a geometric distribution having a parameter β. Thus the underlying Markov chain of the IBP is given by

$$D = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{bmatrix}.$$

During the active period customers arrive according to the Bernoulli process with parameter p_1. Thus Δ is given by:

$$\Delta = \begin{bmatrix} p_1 & 0 \\ 0 & 0 \end{bmatrix}.$$

IBP can be represented by MAP with $D_0 = D(I - \Delta)$ and $D_1 = D\Delta$.

This ends our high level description of the arrival process and the service time distribution used in this thesis. Later, in Chapter 5 we discuss how to use the moments matching approach to reduce the dimension of MAP with special structures (i.e. convolution of phase type distributions).

3.4 Cyclic Polling System

This Section focuses on the analysis of discrete time cyclic polling systems. In order to make the description of the solution approach simple we consider a polling

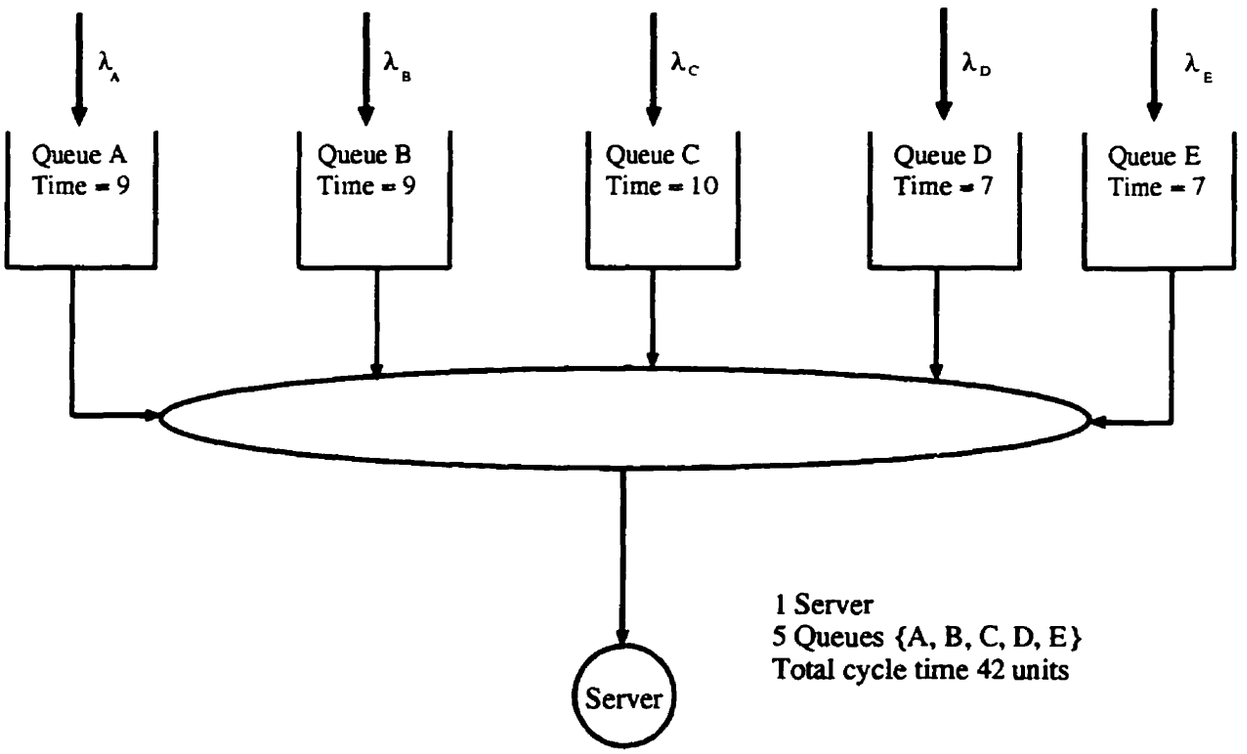

Figure 3.1: Polling System

system with five queues $\{A,\ B,\ C,\ D,\ E\}$. Each queue has an arrival rate λ_α, $\alpha = A,\ B,\ C,\ D,\ E$. Fig. 3.1 represents the design parameters of this polling system. Although, it is possible to define a Markov chain on an appropriate state space for the whole polling system, this is not recommended for obvious reasons (curse of dimensionality). Alfa [3] analyzed an alternating queueing system with a finite buffer. It is shown in [3], for the case of two queues, that the transition matrix becomes quite large. In order to analyze the polling system at hand, we consider each queue separately (decomposition approach) and treat it as a single server queue with vacation as shown in Fig. 3.2. Arrivals to the queueing system occur according to MAP as described in Section 3.3. Service is of phase type distribution as described in Section 3.2 and the switch-over time is equal to zero. We consider both the infinite and finite buffer cases.

Each queue in the polling system can be represented as a MAP/PH/1 queue

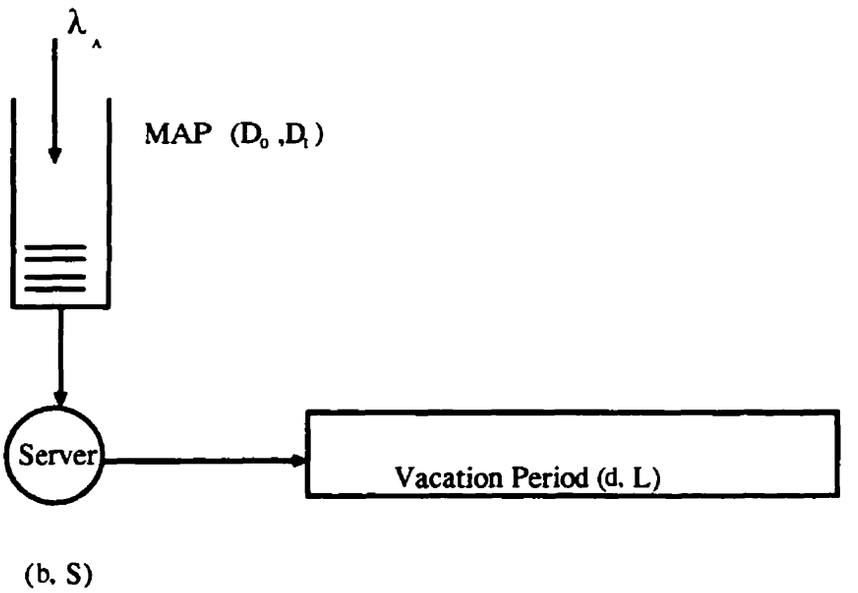

Figure 3.2: Single Server Queue with Vacation

with vacation for the infinite buffer case and as a MAP/PH/1/K queue with vacation for the finite buffer case where K is the buffer size. Each queue is then analyzed as a single server queue with exhaustive time-limited service discipline and vacation periods. For a polling system with Q queues the vacation period is the visit period of the other $(Q-1)$ queues in the polling system. As will be shown later, the visit period for a given queue is a phase type distribution. Let (γ_i, B_i) be the representation of the visit period distribution for queue i ($i = A,\ B,\ C,\ D,\ E$). Because the vacation period has a finite support it can be represented by a phase type distribution (see Neuts [131]). Thus, the independent part of the vacation period distribution for queue A of the polling system defined by stations $\{A,\ B,\ C,\ D,\ E\}$ is given by:

$$Z_A = \begin{bmatrix} B_B & \mathbf{B}_B^{\circ}\gamma_C & 0 & 0 \\ 0 & B_C & \mathbf{B}_C^{\circ}\gamma_D & 0 \\ 0 & 0 & B_D & \mathbf{B}_D^{\circ}\gamma_E \\ 0 & 0 & 0 & B_E \end{bmatrix} \qquad \mathbf{Z}_A^{\circ} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{B}_E^{\circ} \end{bmatrix}$$

$$\boldsymbol{\delta}_A = \begin{bmatrix} \boldsymbol{\gamma}_B & 0 \end{bmatrix}$$

Notice that this phase type distribution has a natural justification. Once the server goes on vacation it starts serving queue B. This is denoted by the visit period to queue B, B_B, which corresponds to the block matrix in position $Z_A(1,1)$. When service at queue B is finished, absorption occurs according to $\mathbf{B}_B^\circ$ and service starts in queue C according to $\boldsymbol{\gamma}_C$. Thus, $\mathbf{B}_B^\circ \boldsymbol{\gamma}_C$ in position $Z_A(1,2)$. The remaining block matrices of Z_A are obtained in the same way. The vector $\mathbf{Z}_A^\circ$ denotes the end of the vacation period. Since the end of the vacation period coincides with the end of the visit period at queue E, we have $\mathbf{B}_E^\circ$ in the last position of the vector $\mathbf{Z}_A^\circ$. The vector $\boldsymbol{\delta}_A$ is the initial probability vector of the vacation period distribution. Since once the server goes on vacation it visits queue B, we have $\boldsymbol{\gamma}_B$ in the first position of the initial probability vector, $\boldsymbol{\delta}_A$, of the vacation period of queue A.

The vacation period distribution as described above ignores the fact that there is some correlation between the visit and vacation period distribution. In fact, the above equation assumes that the vacation period and the visit period are independent. In order to bring in the inherent dependency between the vacation and visit period we use Lee and Sengupta's [109] approach. In their approach, for a reservation cyclic polling system, they assumed that the vacation period is a mixture of two random variables. The first one is the sum of the visit period of $Q-1$ queues with probability P_i for queue i. The second one is $(Q-1)S$ with probability $1-P_i$, where S is a generic random variable. Since the vacation period has a finite support, similarly to the independent part, the dependent part can be represented by a phase type distribution. Let $(\boldsymbol{\psi}_i, Y_i)$ denote this phase type distribution. The vacation period distribution is given by $P_i(\boldsymbol{\delta}_i, Z_i) + (1-P_i)(\boldsymbol{\psi}_i, Y_i)$, where in this notation $P_i(\boldsymbol{\delta}_i, Z_i)$ implies that each element of the initial probability vector $\boldsymbol{\delta}_i$ is multiplied by P_i and each element of the transition matrix Z_i and the absorption vector $\mathbf{Z}_i^\circ$ is also multiplied by P_i. The probability P_i is computed based on the system parameters. In general,

the vacation period of station i, $i = 1, 2, \ldots, Q$, is given by a phase type distribution with dimension r and representation (γ_i, V_i). The dimension of the vacation period distribution is equal to the maximum time the server can be away serving the other $Q-1$ queues, therefore, $r = \sum_{j=1, j\neq i}^{Q} T_j$. The mean of the vacation period is given by $\bar{v}_i = \boldsymbol{\delta}_i(I - V_i)^{-1}\mathbf{e}$.

Notice that similar to the vacation period, the visit period distribution, (γ_i, B_i), depends on the vacation period distribution. However, in this thesis this dependency is not included.

Because of the inter-relationship between the visit and the vacation period distribution we use an iterative approach to solve the exhaustive time-limited polling system. In iteration k we use the results of iteration $k-1$ to obtain the vacation period distribution and solve the single server queue with vacation. Before we go over the iterative procedure we present in the next two sections the analyses of the MAP/PH/1 and the MAP/PH/1/K queues. Note that for ease of notation the station index i is dropped.

3.4.1 MAP/PH/1 Queue with Exhaustive Time-Limited Service and Vacations

Consider a Markov chain described by the state space $\Delta = \{(i, (0, k, l') \cup (j, k, l)), i \geq 0; j = 1, 2, \cdots, T; k = 1, 2, \cdots, n; l' = 1, 2, \cdots, r; l = 1, 2, \cdots, m\}$, where i is the number of customers in the queue during service (vacation); the three tuple $(0, k, l')$ refers to a vacation period with 0 representing vacation state, k representing the phase of arrival and l' the phase of the vacation; the three tuple (j, k, l) refers to the service state with j representing the time clock of service (i.e. how long the service has been going on since the return from vacation), k referring to the phase of arrival and l the phase of service of the customer who is currently in service. The transition matrix of

this Markov chain P is given as

$$P = \begin{bmatrix} B_{00} & B_{01} & 0 & 0 & 0 & \cdots & \cdots \\ B_{10} & A_1 & A_0 & 0 & 0 & \cdots & \cdots \\ 0 & A_2 & A_1 & A_0 & 0 & \cdots & \cdots \\ 0 & 0 & A_2 & A_1 & A_0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

where,

$B_{00} = D_0 \otimes V + D_0 \otimes (\mathbf{V}^{\circ}\boldsymbol{\delta}) \qquad B_{01} = [D_1 \otimes V \quad \mathbf{e}_1' \otimes D_1 \otimes (\mathbf{V}^{\circ}\boldsymbol{\beta})]$

$$B_{10} = \begin{bmatrix} 0 \\ \mathbf{e} \otimes D_0 \otimes (\mathbf{S}^{\circ}\boldsymbol{\delta}) \end{bmatrix}, \qquad A_v = \begin{bmatrix} A_v^3 & A_v^2 & 0 \\ 0 & 0 & I \otimes A_v^0 \\ A_v^1 & \cdots & 0 \end{bmatrix}, \qquad v = 0, 1, 2 \text{ and}$$

$A_2^3 = A_2^2 = 0$, where,

$A_0^0 = D_1 \otimes S, \quad A_0^1 = D_1 \otimes (S\mathbf{e})\boldsymbol{\delta}, \quad A_0^2 = D_1 \otimes (\mathbf{V}^{\circ}\boldsymbol{\beta}^*), \quad A_0^3 = D_1 \otimes V$

$A_1^0 = D_0 \otimes S + D_1 \otimes (\mathbf{S}^{\circ}\boldsymbol{\beta}), \quad A_1^1 = D_0 \otimes (S\mathbf{e})\boldsymbol{\delta} + D_1 \otimes (\mathbf{S}^{\circ}\boldsymbol{\delta}),$

$A_1^2 = D_0 \otimes (\mathbf{V}^{\circ}\boldsymbol{\beta}^*), \quad A_1^3 = D_0 \otimes V,$

$A_2^0 = D_0 \otimes (\mathbf{S}^{\circ}\boldsymbol{\beta}), \quad A_2^1 = D_0 \otimes (\mathbf{S}^{\circ}\boldsymbol{\delta}), \quad \mathbf{V}^{\circ} = \mathbf{e} - V\mathbf{e}, \quad \mathbf{S}^{\circ} = \mathbf{e} - S\mathbf{e},$

and $\boldsymbol{\beta}^* = \boldsymbol{\beta}^*(S + \mathbf{S}^{\circ}\boldsymbol{\beta})$, with $\boldsymbol{\beta}^*\mathbf{e} = 1$.

The symbol $\otimes$ is the Kronecker product sign. $\mathbf{e}_v'$ is the transpose of the column vector $\mathbf{e}_v$, which has 1 in the v^{th} position and 0 elsewhere. The block matrices A_0, A_1 and A_2 are square matrices of dimensions $n(r+Tm)$, the block matrix B_{00} is a square matrix of dimension nr, the block matrix B_{01} is of dimension $nr \times n(r+Tm)$, and the block matrix B_{10} is of dimension $n(r+Tm) \times nr$. Note that the vector $\mathbf{e}_1'$ in B_{01} is of dimension T. $\boldsymbol{\beta}^*$ is used to denote resumption of service after an interruption. Its justification is based on the properties of the phase type distribution and can be found in Neuts [133, page 52].

The detailed analysis of this queueing system could be found in Alfa [2]. Here we quote the major results without their proof. The rate matrix R can be obtained by solving:

$$R = A_0 + RA_1 + R^2A_2. \tag{3.2}$$

The mean number of customers in the system at an arbitrary time, μ_L, and the mean waiting time, W_L, are given, respectively, as

$$\mu_L = \mathbf{x}_1(I - R)^{-2}\mathbf{e}, \quad W_L = \frac{\mu_L}{\lambda}. \tag{3.3}$$

Let v_o be the probability that the server is on vacation, then

$$v_o = \mathbf{x_0 e} \ + \ \mathbf{x}_1[I - R]^{-1}(\mathbf{e}_1 \otimes \mathbf{e}), \tag{3.4}$$

where $\mathbf{e}_i$ is a column vector of zeros and 1 in the i^{th} position.

This queue is stable if $\lambda\bar{b} < T/(T + \bar{v})$. This condition implies that the expected service time of the expected number of arrivals in a cycle consisting of a service period and a vacation period is less than the maximum time allowed per visit. In the remainder of this chapter we assume that this condition holds whenever we are dealing with the infinite buffer case.

3.4.1.1 Duration of a queue visit

In order to obtain the queue visit distribution, we present a simple recursive formula for the computation of the busy period, then we show how to obtain the visit period distribution. Let $p_i(j)$ be the probability that a busy period initiated by j customers lasts i units of time. Note that because we are dealing with discrete time systems the service time of j customers must be at least equal to j, thus $p_i(j) = 0$ for $i < j$. Let $g_{i,j}$ be the probability that the service time of j customers lasts i units of time. Let $d_{i,j}$ be the probability that j customers arrive in i units of time. The following proposition, due originally to Klimko and Neuts [101], is known to be true for Bernoulli arrival

processes and general discrete time service distributions.

Proposition 1: *The probability that a busy period initiated by j customers lasts i units of time is given by:*

$$p_i(j) = g_{i,j}d_{i,0} + \sum_{l=1}^{i-j} g_{i-l,j} \sum_{k=1}^{l} d_{i-l,k}p_l(k) \tag{3.5}$$

Proof: The arguments leading to this proposition are as follows. The first term on the right hand side (RHS) is due to the probability that the service time of j customers lasts i units of time and during that period no new customers join the queue. The second term on the RHS is due to the probability that the service time of j customers last $i-l$ units of time. During the first $i-l$ units of time k new customers join the queue. These k customers initiate a busy period that lasts l units of time. □

Next, we extend this result to the more general case i.e. we consider arrival to be represented by MAP and service by phase type distribution. In the case of the discrete MAP/PH/1 queue let $G^{(j)}(i)$ be a matrix of dimension $m \times m$ with its entries $G^{(j)}_{u,v}(i)$ representing the probability that the service time of j customers last i units of time given that the service of the first customer starts in phase u and that of the jth customer ends in phase v. Letting $S_1 = \mathbf{S}^0\boldsymbol{\beta}$, the matrix $G^{(j)}(i)$ is given by:

$$G^{(1)}(i) = S^{i-1}S_1 \text{ for } i \geq 1 \tag{3.6}$$

$$G^{(i)}(i) = S_1^k \text{ for } i \geq 1 \tag{3.7}$$

$$G^{(j)}(i) = S_1G^{(j-1)}(i-1) + SG^{(j)}(i-1) \text{ for } i \geq j+1,\ j \geq 2. \tag{3.8}$$

Also we define the matrix $B^{(j)}$ of dimension $mn \times mn$ such that its entries $B^{(j)}_{u,v}(i)$ represent the probability that a busy period initiated by j customers lasts i units of time given that the first customer's service and arrival are in phase u, $1 \leq u \leq mn$, and that the service of the last customer and arrival are in phase v, $1 \leq v \leq mn$. In order to extend the result of proposition 1 to the case of MAP arrival and phase

service we let the scalar $g_{i,j}$ be the matrix $G^{(j)}(i)$ and $d_{i,j}$ be

$$d_{i,j} = \begin{pmatrix} i \\ j \end{pmatrix} D_0^{i-j} D_1^j.$$

Then we have

$$B^{(j)}(i) = \left(G^{(j)}(i) \otimes D_0^i\right) + \sum_{l=1}^{i-j} \sum_{k=1}^{l} \begin{pmatrix} i-l \\ k \end{pmatrix} \left(G^{(j)}(i-l) \otimes D_0^{i-l-k} D_1^k\right) B^{(k)}(l)$$

However, since our primary interest is in the probability that the busy period lasts i units of time we define $p_i(j) = \boldsymbol{\mu} B^{(j)}(i)\mathbf{e}$, where $\boldsymbol{\mu}$ is the steady state probability that arrival is in phase i, $i = 1, \ldots, n$, and service is in phase j, $j = 1, \ldots, m$ (i.e. $\boldsymbol{\mu}$ is mn vector).

Proposition 2: *The probability that a busy period initiated by j customers lasts i units of time is given by:*

$$\begin{aligned} p_i(j) &= \boldsymbol{\mu} \left(G^{(j)}(i) \otimes D_0^i\right) \mathbf{e} \\ &+ \sum_{l=1}^{i-j} \sum_{k=1}^{l} \begin{pmatrix} i-l \\ k \end{pmatrix} \boldsymbol{\mu} \left(G^{(j)}(i-l) \otimes D_0^{i-l-k} D_1^k\right) \mathbf{e} p_l(k) \end{aligned} \tag{3.9}$$

Proof: The proof of this proposition follows directly from Proposition 1 by replacing $g_{i,j}$ with $G^{(j)}(i)$ and $d_{i,j}$ with $\begin{pmatrix} i \\ j \end{pmatrix} D_0^{i-j} D_1^j$. □

The mean duration of a busy period initiated by j customers, μ_j, is given by:

$$\mu_j = \sum_{i=j}^{\infty} i p_i(j) \tag{3.10}$$

$$= \sum_{i=j}^{\infty} i \left\{ \boldsymbol{\mu} \left(G^{(j)}(i) \otimes D_0^i\right) \mathbf{e} \right\}$$

$$+ \sum_{i=j}^{\infty} i \left\{ \sum_{l=1}^{i-j} \sum_{k=1}^{l} \begin{pmatrix} i-l \\ k \end{pmatrix} \boldsymbol{\mu} \left(G^j(i-l) \otimes D_0^{i-l-k} D_1^k\right) \mathbf{e} p_l(k) \right\} \tag{3.11}$$

Note that because this is an infinite sum we choose a large number J such that $1 - \sum_{i=j}^{J} p_i(j) < \epsilon$, where ϵ is a very small positive number. Now we are in a position to compute the visit period distribution.

Let θ_i, $(0 \leq i \leq T-1)$, be the probability that the server returns from a vacation to find i customers waiting and $\tilde{\theta}_T = \sum_{j=T}^{\infty} \theta_j$. For our problem

$$\theta_i = \bar{v}\mathbf{x}_{i,0}(\mathbf{e} \otimes I)\mathbf{V}^{\circ}/v_0, \quad 1 \leq i \leq T-1, \text{ and} \tag{3.12}$$

$$\theta_0 = \bar{v}\mathbf{x}_0(\mathbf{e} \otimes I)\mathbf{V}^{\circ}/v_0 \tag{3.13}$$

Therefore, we have to consider three cases:

Zero Customers Waiting

If when the server returns from a vacation it finds no customer waiting, then the duration of the visit to the queue is zero, because the server departs immediately for a vacation. The probability of this occurring is θ_0.

At Least One (but less than T) Customer Waiting

If when the server returns from a vacation it finds k customers waiting $(1 \leq k \leq T-1)$, then the duration of a visit to the queue is a phase type distribution with parameters $(\boldsymbol{\alpha}(k), F(k))$, where $\boldsymbol{\alpha}(k) = [1\ 0\ 0\ \cdots 0]$, and

$$F(k) = \begin{bmatrix} 0 & b_1'(k) & 0 & \cdots & 0 \\ 0 & 0 & b_2'(k) & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & b_{T-1}'(k) \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix},$$

where $b_j'(k) = 1 - \tilde{b}_j(k), \quad j \geq 1$, and

$$\tilde{b}_1(k) = b_1(k) \text{ and} \tag{3.14}$$

$$\tilde{b}_j(k) = b_j(k)(1 - \sum_{i=1}^{j-1} b_i(k))^{-1} \quad 2 \leq j \leq T-1. \tag{3.15}$$

The probability of this occurring is θ_k.

At Least T Customers Waiting

If when the server returns from a vacation it finds at least T customers waiting then the duration of its visit to the queue is exactly T units and can be represented by a phase type distribution with parameters $(\boldsymbol{\alpha}(T), F(T))$, where $\boldsymbol{\alpha}(T) = [1\ 0\ 0\ \cdots 0]$, and

$$F(T) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}.$$

The probability of this occurring is $\tilde{\theta}_T$.

The duration of a visit is thus a phase type distribution with parameters $\gamma_0 = \theta_0$, $\boldsymbol{\gamma} = [1 - \theta_0\ 0\ 0\ \cdots\ 0]$. Since the server visits a queue even though it is empty, the distribution of the visit period must have a probability mass equal to zero at zero, hence $\gamma_0 = 0$ (i.e. the probability of not visiting a queue is zero). Therefore, the vector $\boldsymbol{\gamma}$ must have a one in position 1 and zero every where else i.e. $\gamma_0 = 0$, and $\boldsymbol{\gamma} = [1\ \mathbf{0}]$. The transition matrix, B, and the absorption vector, $\mathbf{B}^\circ$, of the visit period distribution are given, respectively, by:

$$B = \left(\tilde{\theta}_T F(T) + \sum_{k=1}^{T-1} \theta_k F(k)\right) / (1 - \theta_0), \text{ and} \tag{3.16}$$

$$\mathbf{B}^\circ = \left(\tilde{\theta}_T \mathbf{F}^\circ(T) + \sum_{k=1}^{T-1} \theta_k \mathbf{F}^\circ(k)\right) / (1 - \theta_0) \tag{3.17}$$

where $\mathbf{F}^\circ(i) = \mathbf{e} - F(i)\mathbf{e}$.

3.4.2 The MAP/PH/1/K Queue with Exhaustive Time-Limited Service and Vacations

The state space of the Markov chain of this queueing system is the same as that of the MAP/PH/1 queue except that the maximum number of customers in the buffer is K. The transition matrix P describing this Markov chain is given as:

$$P = \begin{bmatrix} B_{00} & B_{01} & & & & \\ B_{10} & A_1 & A_0 & & & \\ & A_2 & A_1 & A_0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_2 & A_1 & A_0 \\ & & & & A_2 & \hat{A} \end{bmatrix}$$

where the matrices B_{00}, B_{01}, B_{10}, A_0, A_1, A_2 are given in Section 3.4.1. The matrix $\hat{A}$ is given by

$$\hat{A} = A_0 + A_1.$$

The steady state probability vector $[\mathbf{x}_0\ \mathbf{x}_1\ \ldots\ \mathbf{x}_{K_1}]$ can be obtained by solving the system of equation $\mathbf{x}P = \mathbf{x}$ and $\mathbf{x}\mathbf{e} = 1$, where K_1 is the number of customers in the system i.e. $K_1 = K + 1$. Because this system of equations is finite and sparse we use the block Gauss-Seidel iterative method. A discussion on the use of iterative algorithms for the solution of Markov chains is available in Stewart [161] or Grassmann [74]. The mean number of customers in the system at arbitrary times and the mean waiting time are given, respectively, by:

$$\mu_L = \sum_{i=1}^{K_1} i\mathbf{x}_i\mathbf{e}, \quad W_L = \frac{\mu_L}{\lambda(1 - P_{K_1})}, \tag{3.18}$$

where $P_{K_1} = \mathbf{x}_{K_1}\mathbf{e}$ and is the blocking probability. The probability that the server is on vacation is given by

$$v_0 = \mathbf{x}_0\mathbf{e} + \sum_{i=1}^{K_1} \mathbf{x}_i(\mathbf{e}_1 \otimes \mathbf{e}). \tag{3.19}$$

Similar to the infinite buffer case, we need to obtain the visit period distribution. This is achieved easier by first obtaining the number served during a visit from which we can compute the busy period distribution.

3.4.2.1 The number served during a busy period

In this section, we discuss the number of customers served during a busy period for the MAP/PH/1/K queue. Again K_1 is the number of customers in the queueing system, $K_1 = K + 1$. The number of customers served during a busy period has a phase type distribution with representation $(\phi(k), L)$ where $\phi(k)$ is the initial probability vector. The vector $\phi(k)$ has 1 in position k and zero everywhere else. The matrix L is given by:

$$L = \begin{bmatrix} d_1 & d_2 & d_3 & \cdots & d_{K_1-1} & 1 - \sum_{i=0}^{K_1-1} d_i \\ d_0 & d_1 & d_2 & \cdots & d_{K_1-2} & 1 - \sum_{i=0}^{K_1-2} d_i \\ 0 & d_0 & d_1 & \cdots & d_{K_1-3} & 1 - \sum_{i=0}^{K_1-3} d_i \\ \vdots & \ddots & \ddots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d_0 & 1 - d_0 \end{bmatrix}.$$

where d_i is the probability of having $i \geq 0$ arrivals during the service time of one customer. Let P_j be the probability that the service time of one customer lasts j time units, and let Q_j^i be the probability of having i arrivals in j time units. P_j, Q_j^i and d_i are given, respectively, by:

$$P_j = \boldsymbol{\beta} S^{j-1} \mathbf{S}^{\circ} \tag{3.20}$$

$$Q_j^i = \binom{j}{i} \boldsymbol{\pi} D_0^{j-i} D_1^i \mathbf{e} \tag{3.21}$$

$$d_i = \sum_{j=1}^{\infty} P_j Q_i^j \tag{3.22}$$

From a computational aspect, because the sum for d_i goes to infinity one would stop when the increment in the total probability is less than an acceptable tolerance,

say $\epsilon < 10^{-8}$. Hence, the probability that v; $v = k, k+1, ...$; customers are served during a busy period initiated by k customers is given by $N_v(k) = \phi(k)L^{v-1}\mathbf{L}^\circ$. The mean number of customers, $h_1(k)$, served during a busy period initiated by k customers is given by the mean of the phase type distribution $(\phi(k), L)$ i.e. $h_1(k) = \phi(k)(I - L)^{-1}\mathbf{L}^\circ$ where $\mathbf{L}^\circ = \mathbf{e} - L\mathbf{e}$.

3.4.2.2 Duration of a queue visit

From the number of customers, j, served during a busy period we can compute, $b_i(k)$, the probability that a busy period initiated by k customers lasts i time units. $b_i(k)$ is given by:

$$b_i(k) = \sum_{j=k}^{i} N_j(k)\beta G^{(j)}(i)\mathbf{e} \tag{3.23}$$

where $G^{(j)}(i)$ is as defined in Section 3.4.2.2.

Let θ_i, $(0 \leq i \leq min(T-1, K))$, be the probability that the server returns from a vacation to find i customers waiting. We have to make a distinction between two cases i.e. $T \leq K$ and $T > K$.

Case 1: $T \leq K$

Let $\tilde{\theta}_T = \sum_{j=T}^{K} \theta_j$. For our problem

$$\theta_i = \tilde{v}\mathbf{x}_{i,0}(\mathbf{e} \otimes I)\mathbf{V}^\circ/v_0, \quad 1 \leq i \leq K, \text{ and} \tag{3.24}$$

$$\theta_0 = \tilde{v}\mathbf{x}_0(\mathbf{e} \otimes I)\mathbf{V}^\circ/v_0. \tag{3.25}$$

Consider each of the followings:-

Zero Customers Waiting

If when the server returns from a vacation it finds no customer waiting, then the duration of the visit to the queue is zero, because the server departs immediately for a vacation. The probability of this occurring is θ_0.

At Least One (but less than T) Customer Waiting

If when the server returns from a vacation it finds k customers waiting, $1 \leq k \leq$

$(T-1)$, then the duration of a visit to the queue is a phase type distribution with parameters $(\boldsymbol{\alpha}(k), F(k))$, where $\boldsymbol{\alpha}(k) = [1\ 0\ 0\ \cdots 0]$, and

$$F(k) = \begin{bmatrix} 0 & b_1'(k) & 0 & \cdots & 0 \\ 0 & 0 & b_2'(k) & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & b_{T-1}'(k) \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix},$$

where $b_j'(k) = 1 - \tilde{b}_j(k), \quad j \geq 1.$

$$\tilde{b}_1(k) = b_1(k), \text{ and} \tag{3.26}$$

$$\tilde{b}_j(k) = b_j(k)(1 - \sum_{i=1}^{j-1} b_i(k))^{-1} \quad 2 \leq j \leq T-1. \tag{3.27}$$

The probability of this occurring is θ_k.

At Least T Customers Waiting

If when the server returns from a vacation it finds at least T customers waiting then the duration of its visit to the queue is exactly T units and can be represented by a phase type distribution with parameters $(\boldsymbol{\alpha}(T), F(T))$, where $\boldsymbol{\alpha}(T) = [1\ 0\ 0\ \cdots 0]$, and

$$F(T) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}.$$

The probability of this occurring is $\tilde{\theta}_T$.

The duration of a visit is thus a phase type distribution with parameters $\gamma_0 = \theta_0$, $\boldsymbol{\gamma} = [1 - \theta_0\ 0\ 0\ \cdots\ 0]$. For the same reasoning as in Section 3.4.1. we set $\gamma_0 = 0$

and $\gamma = [1\ \mathbf{0}]$. The transition matrix, B, and the absorption vector, $\mathbf{B}^\circ$, of the visit period distribution are given, respectively, by:

$$B = \left(\tilde{\theta}_T F(T) + \sum_{k=1}^{T-1} \theta_k F(k)\right) / (1 - \theta_0), \text{ and} \tag{3.28}$$

$$\mathbf{B}^\circ = \left(\tilde{\theta}_T \mathbf{F}^\circ(T) + \sum_{k=1}^{T-1} \theta_k \mathbf{F}^\circ(k)\right) / (1 - \theta_0) \tag{3.29}$$

where $\mathbf{F}^\circ(i) = \mathbf{e} - F(i)\mathbf{e}$.

Case 2: $K < T$

This case is not realistic from a design point of view and hence will not be presented here. It would not make sense to assign less memory than slotted time.

3.5 Iterative Procedure

Before we present the iterative procedure, let us briefly explain how we adopt Lee and Sengupta's [109] idea to deal with the correlation between the visit period and the vacation period. The vacation period is taken to be a mixture of an independent and a dependent random variable. This mixture is assumed to depend on the system parameters. The first part of the vacation period is obtained using the visit period distribution presented in Section 3.4.1.1 for the infinite buffer case and in Section 3.4.2.2 for the finite buffer case. The second part of the vacation period, in our case, depends on the polling system utilization. This is because under medium to heavy load conditions the server, once it leaves a queue, has a higher probability of staying on vacation for the maximum vacation period than under light load conditions. Our experimentation with the algorithm, using different input parameters, showed that it is more efficient to use different vacation periods (dependent parts) for $\rho < 0.65$ and $\rho \geq 0.65$, where ρ is the system utilization.

For the case of $\rho < 0.65$ the dependent part of the vacation period is computed using Algorithm 2 of Lee and Sengupta [109]. This can be achieved by defining a

phase type distribution (ψ_i, Y_i). In our case, this phase type distribution is given by: $\psi = [1\ 0 \ldots\ 0]$ and

$$Y_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & \cdots & \vdots \\ 0 & \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ 0 & \vdots & 0 & p'(Q-1) & 0 & \cdots & \vdots \\ 0 & \vdots & 0 & 0 & p'(Q) & 0 & \cdots \\ 0 & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & p'(\sum_{j=1,j\neq i}^{Q-1} T_j) \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}.$$

where $p'(Q-1) = 1 - p(Q-1)$,
$p'(j) = 1 - p(j)(1 - \sum_{i=Q-1}^{J} p(i))^{-1}$, and $J = \sum_{j=1,j\neq i}^{Q-1} T_j$. The probability $p(j)$ is computed using Lee and Sengupta [109] (Algorithm 2). For our model, $p(j)$ is computed as follow. Let $r_i(k)$ be the probability that a visit period lasts i, $2 \leq i \leq T_k$, units of time for queue k. Then,

$$r_i(k) = \gamma(k) B^{i-1}(k) \mathbf{B}^{\circ}(k).$$

Next, we sort in descending order all of $r_i(k)$, $2 \leq i \leq T_k$ and $k \in (Q-1)$, to get $p(j)$. Note that the distribution (ψ, Y) implies that:

- Once the server goes on vacation it visits all $Q-1$ queues, and
- the remaining part of the vacation period represents a visit period where all $Q-1$ queues are treated as a single queue.

For the case of $\rho \geq 0.65$ the dependent part of the vacation period is assumed to be deterministic. Thus for queue i the length of the vacation period is given by $\sum_{j,j\neq i}^{Q} T_j$ which can be represented by a phase type distribution with 1's in the super-diagonal positions and 0's every where else. This is because for moderate to high

system utilization, once the server goes on vacation from queue i it is more likely to stay away serving each of the other queues for its whole time period.

The iterative procedure to solve the cyclic polling system with infinite buffer is outlined below:

1. For $i = 1$ to Q let the distribution of the visit period, (γ_i, B_i), be given by a phase type distribution of dimension T_i.

$$B_i = \begin{bmatrix} 0 & \rho_i & 0 & \cdots \\ 0 & 0 & \rho_i & \vdots \\ 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \rho_i \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{3.30}$$

   where ρ_i is queue i utilization.

2. For $i = 1$ to Q

   if $0.8 \geq \rho$ set $P_i = \rho^{Q-1} + \rho_i$

   if $0.65 \leq \rho \leq 0.8$ set $P_i = (1 - \rho_i)\rho^{Q-1} + \rho_i$

   if $\rho \leq 0.65$ set $P_i = (1 - \rho_i)\omega^{Q-1} + \rho_i$, where $\omega = \frac{\sum_{j,j \neq i} T_j \rho_i}{T_i}$

   For infinite buffer queues:

   a- Compute the vacation period $V = (1 - P_i) \times Z + P_i \times Y$

   b- Compute the rate matrix R^i (Eq. 3.2)

   c- Compute $[\mathbf{x_0}\ \mathbf{x_1}]$

   d- Compute the average queue length μ_L^i (Eq. 3.3)

   e- Compute the probability that the server is on vacation (Eq. 3.4)

   f- Update the visit period distribution

   end

For finite buffer queues:

if $P_i \geq 1$ set $P_i = 1$

a- Compute the vacation period $V = (1 - P_i) \times Z + P_i \times Y$

b- Compute $[\mathbf{x_0}\ \mathbf{x_1}\ \ldots \mathbf{x}_{K_i}]$ using Gauss-Seidel

c- Compute the average queue length μ_L^i (Eq. 3.18)

d- Compute the probability that the server is on vacation (Eq. 3.19)

e- Update the visit period distribution

end

3. If the average queue length and the mean vacation period did not converge go back to 2, else stop.

Note that we chose to initialize the visit period using Equation 3.30 because it makes convergence faster. This is because when the utilization is low the average visit period will be small and when ρ_i is high the average visit period will be high. This is confirmed by the results of most of our computer runs. Notice that in the case ρ_i is one (queue is unstable) then the average visit period will be equal to T_i.

During the computation of the performance measures of the polling system at each iteration we store only the information pertinent to the current queue i.e. the arrays used to compute the performance measures for queue $i-1$ are reused to compute the performance measures for queue i. This, of course, could be done by creating a subroutine to solve for the vacation model and a main that calls this subroutine for each queue.

3.5.1 Convergence of the Iterative Algorithm

In order to prove that the algorithm adopted for this cyclic polling system with exhaustive time-limited service discipline, MAP input, phase type service distribution, and zero switch over time converges we follow the same steps as Lee and Sengupta [109]. The proof is presented for the infinite buffer case. Similar arguments

could be made for the finite buffer queue case. First, consider two single server systems of the type analyzed in Subsections 3.4.1 and 3.4.2 in which the vacation periods are denoted by $V^{(1)}$ and $V^{(2)}$. Let the corresponding jth service period be denoted by $B_j^{(i)}$ and let the jth queue length when the server leaves for vacation be denoted by $N_j^{(i)}$ $i = 1, 2$ and $j = 1, 2, \ldots$.

Lemma If $V^{(1)} \geq_{st} V^{(2)}$ then $B_j^{(1)} \geq_{st} B_j^{(2)}$ and $N_j^{(1)} \geq_{st} N_j^{(2)}$.

The $\geq_{st}$ is defined in Stoyan [163] as stochastically dominant. The proof of this Lemma for our cyclic polling system is done into two steps. First, we prove monotonicity and then we prove comparability. Let the kernel of the Markov chain $\{N_j^{(i)}, i = 1, 2\}$ be denoted by $Q_i(x, y) = Prob\{N_{j+1}^{(i)} \leq y | N_j^{(i)} = x\}$.

- Monotonicity: For $i = 1, 2$ the following relationship is true. Let $A^{(i)}$ represent the number of arrivals in a vacation period $V^{(i)}$, $C^{(i)}$ the number of arrivals in a service period, and $D^{(i)}$ the number served in a visit period.

$$\begin{aligned} Q_i(x_1, y) &= Prob\{(x_1 + A^{(i)} + C^{(i)} - D^{(i)}, 0)^+ \leq y | N_j = x_1\} \\ &\leq Prob\{(x_2 + A^{(i)} + C^{(i)} - D^{(i)}, 0)^+ \leq y | N_j = x_2\} \\ &= Q_i(x_2, y) \end{aligned}$$

  where $(z, 0)^+ = z$ if $z \geq 0$ and $(z, 0)^+ = 0$ otherwise.

  This relationship is true for any x_1, x_2, y positive integer for the following reason. The number of arrivals during the vacation period is the same for both cases. The number of arrivals during the visit period depends on its length. Since we are dealing with discrete time single arrival queues, in the worst case the number of arrivals during the visit period given $N_j^{(i)} = x_2$ is equal to the number of arrivals during the visit period given that $N_j^{(i)} = x_1$. This implies that both visit periods are equal. Hence, the number served in the visit period given $N_j^{(i)} = x_2$ is the same as that served in the visit period given $N_j^{(i)} = x_1$. Thus the monotonicity proof.

- Comparability: The comparability condition is based on the following arguments. Since $V^{(2)} \leq_{st} V^{(1)}$, then the number of arrivals in $V^{(2)}$ is less than the number of arrivals in $V^{(1)}$. Hence, the visit period associated with $V^{(2)}$ is shorter which results in a smaller number of arrivals during the visit period. Thus, the number of customers served during a visit period under vacation $V^{(2)}$ is at best equal to the number served under vacation period $V^{(1)}$.

3.5.2 Stability of the Iterative Algorithm

In order to show that the proposed algorithm is stable and converges to the solution we first show that the vacation period in iteration k, $V^{(k)}$, $k = 1, 2, \ldots$ represents a stochastically non-decreasing sequence of random variables and second that if the original cyclic server queue is stable then the vacation model remains stable throughout the iterative procedure. The first part of the proof is identical to Lee and Sengupta [109] and therefore, we avoid its repetition here. For the second part, consider a system in which at every visit the server spends T_i time units at the queue and in which one unit of walk time is incurred at every queue. Call this system system H. If system H is stable then our system is stable. A sufficient condition for stability for system H is given in Georgiadis [71] as

$$\rho_i < \frac{T_i}{Q}(1 - \rho) \tag{3.31}$$

where ρ_i is the queue utilization and ρ is the system utilization. From Eq. 3.31 we have

$$Q\rho_i < T_i(1 - \rho) \tag{3.32}$$

which after algebraic manipulation leads to

$$\rho_i(Q + T_i) + T_i \sum_{n=1, n\neq i}^{Q} \rho_n < T_i \tag{3.33}$$

If we set $\rho_n = 0$, $\forall\, n$, except for ρ_i and ρ_j in Eq. 3.33 we get

$$\rho_i \frac{Q + T_i}{T_i} < 1 - \rho_j \tag{3.34}$$

If we set $\rho_j = 0$ in Eq. 3.34, then

$$\rho_i < \frac{T_i}{Q + T_i} \tag{3.35}$$

Note that the average visit period is bounded by that of the exhaustive discipline. Hence,

$$\bar{s}_i \leq \frac{\bar{b}_i}{1 - \rho_i} \tag{3.36}$$

where $\bar{s}_i$ is the average visit period. For the algorithm to converge we must have $\bar{v}_i = \sum_{n=1, n\neq i}^{K} \bar{s}_n$. From Eq. 3.36 we can write

$$\sum_{n=1, n\neq i}^{Q} \bar{s}_n \leq \sum_{n=1, n\neq i}^{Q} \frac{\bar{b}_n}{1 - \rho_n} \tag{3.37}$$

$$\bar{v}_i \leq \sum_{n=1, n\neq i} \frac{T_n}{\rho_i (Q + T_n)} \tag{3.38}$$

The first inequality (Eq. 3.37) is due to Eq. 3.36 and the second inequality (Eq. 3.38) is due to Eq. 3.34 (take the inverse of Eq. 3.34 and then take the sum over n). For the vacation model to be stable we must show that $\rho_i < \frac{T_i}{T_i + \bar{v}_i}$ for $i = 1, 2, ..., Q$ (stability condition for the MAP/PH/1 queue).

From Eq. 3.38 we can write

$$\bar{v}_i < \sum_{n=1, n\neq i}^{Q} \frac{T_n}{\rho_i (T_n + Q)} \tag{3.39}$$

$$\rho_i \bar{v}_i + T_i \rho_i < T_i \rho_i + \sum_{n=1, n\neq i}^{Q} \frac{T_n}{T_n + Q} \tag{3.40}$$

$$\rho_i (\bar{v}_i + T_i) < \sum_{n=1, n\neq i}^{Q} \frac{T_n}{T_n + Q} + T_i \frac{T_i}{T_i + Q} \tag{3.41}$$

The second term in this inequality (Eq. 3.41) is due to Eq. 3.35. The next step in the proof is to show that the right hand side of the inequality (Eq. 3.41) is bounded by

T_i. This can be done by routine algebraic manipulations.

$$\begin{aligned}
\rho_i(\bar{v}_i + T_i) &< \sum_{n=1}^{Q} \frac{T_n}{T_n + Q} + \frac{T_i^2}{T_i + Q} - \frac{T_i}{T_i + Q} \\
&< Q + \frac{T_i(T_i - 1)}{T_i + Q} \\
&< \frac{K(T_i + Q) + T_i(T_i - 1)}{T_i + Q} \\
&< \frac{K^2 + QT_i + T_i^2 + KT_i - KT_i - T_i}{T_i + Q} \\
&< \frac{(Q + T_i)^2 - QT_i - T_i}{T_i + Q} \\
&< Q + T_i - Q - 1 \\
&< T_i
\end{aligned}$$

Hence we have $\rho_i(\bar{v}_i + T_i) < T_i$ which proves our claim. This proves that the average vacation period keeps increasing from one iteration to the next but never exceeds the limiting value. Note that this proves only that the algorithm converges. Although we have proven that the algorithm converges and is stable, we did not prove that it converges to the exact distribution. The stability of the algorithm is confirmed by starting with unstable conditions. The results obtained for the mean queue length and the mean waiting time were equal to infinity. This confirms our proof.

3.6 Numerical Examples

In comparing the approximate approach results with those of the simulation we define the percent error as $\frac{W_{sim} - W_{app}}{W_{sim}}$, where W_{sim} is the simulation mean waiting time and W_{app} is the approximate approach mean waiting time. A negative percent error indicates that the approximate method over-estimates the simulation results. For the simulation we used 5 replications each of which is of 10^5 time units long for $Q \leq 6$ and 10^6 for $Q \geq 7$, where Q is the number of queues, and a warm up period of 10^5 time units, we show the half-width of the 95% confidence interval using Law

and Kelton [108, Chap. 9] Since, there are many system parameters we chose to show some of the results to give a general idea on the performance of the iterative procedure.

3.6.1 Infinite Capacity Model

The first set of examples shows the performance of the iterative method for different system utilization. Tables 3.1-3.4 show the comparison between the simulation and the approximate approach of the mean waiting time for a 4 queues polling system. The service time distribution is identical for all queues and is of the geometric type. Customers arrive according to the geometric distribution and the probability of arrival is given by λ. The maximum time allocated for each queue is given by T_{max}. This

Table 3.1: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.75$

Queue	λ	T_{max}	W_{app}	W_{sim}	% Error
1	0.2	5	13.120	14.987 ± 0.694	12.5%
2	0.1	4	8.922	9.892 ± 0.247	9.8%
3	0.2	6	9.123	10.006 ± 0.390	8.8%
4	0.1	3	14.270	17.119 ± 0.895	16.6%

polling system is asymmetric in terms of the load and the time allocated to each queue. We vary the system utilization between 0.5 and 0.75 while keeping the mean service time constant. In all four examples queue 2 and 4 have the same arrival rate, yet consistently queue 2 has a lower mean waiting time than queue 4. This is due to queue 2 having a higher time limit (4 time units for queue 2 compared to 3 time units for queue 4). In Table 3.4, queue 3 and 4 have the same arrival rate with queue 3 having a much higher time limit (i.e. queue 3 have twice the time allocated to queue 4). The mean waiting time for queue 3 is much lower than that of queue 4. This

Table 3.2: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.6875$

Queue	λ	T_{max}	W_{app}	W_{sim}	% Error
1	0.15	5	7.201	7.052 ± 0.185	−2.1%
2	0.1	4	7.967	7.074 ± 0.812	−12.6%
3	0.2	6	7.449	6.844 ± 0.196	−8.8%
4	0.1	3	11.398	10.854 ± 0.291	−5.0%

Table 3.3: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.625$

Queue	λ	T_{max}	W_{app}	W_{sim}	% Error
1	0.1	5	4.524	3.918 ± 0.0668	−15.5%
2	0.1	4	4.944	5.149 ± 0.0637	4.0%
3	0.2	6	5.457	4.743 ± 0.0831	−15.1%
4	0.1	3	7.258	7.407 ± 0.234	2.0%

is mainly due to the frequent timeouts for queue 4 while queue 3 is served almost exhaustively.

The maximum absolute error, 20.6%, for these examples is encountered with a low system utilization, $\rho = 0.5$, for queue 3 in Table 3.4. All other examples have a maximum absolute percentage error of 15%.

The second set of examples, Tables 3.5-3.7, shows the performance of the iterative procedure when the number of queues is 5, 6 and 8 while keeping the system utilization constant. Again, arrival and service time are represented by geometric distribution. Although, queues {1, 5} in Table 3.5 and 3.6, queues {2, 6} in Table 3.6, and queues {1, 4, 7} and {2, 6, 8} in Table 3.7 have the same parameters (arrival rate, and T_{max}) their simulation mean waiting time is not identical. This is due mainly to

Table 3.4: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.50$

Queue	λ	T_{max}	W_{app}	W_{sim}	% Error
1	0.2	5	2.530	2.430 ± 0.0354	−4.1%
2	0.1	4	2.765	3.055 ± 0.0550	9.5%
3	0.2	6	2.445	2.028 ± 0.0486	−20.6%
4	0.1	3	3.714	4.003 ± 0.0598	7.2%

Table 3.5: 5 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.75$

Queue	λ	T_{max}	W_{app}	W_{sim}	% Error
1	0.15	5	12.610	12.522 ± 0.703	−0.7%
2	0.1	4	11.533	11.832 ± 0.318	2.5%
3	0.15	6	9.622	8.727 ± 0.418	−10.3%
4	0.05	3	11.550	10.808 ± 0.179	−6.9%
5	0.15	5	12.610	12.345 ± 0.749	−2.2%

sampling errors. The performance of the algorithm is comparable in terms of the percentage error with the case of 4 queues. Note that in most cases the iterative approach over-estimates the mean waiting time on the average by about $\pm 10\%$.

In all the examples studied, the iterative algorithm converges in less than 30 iterations, with a run time under 15 minutes on an IBM RS6000/590. Most of the CPU time for each run is taken to solve for the rate matrix R, and the queue length distribution. The rate matrix R is obtained using the algorithm given in Alfa [2]. The stopping criterion for convergence was chosen to be $\epsilon = 10^{-8}$ i.e. the program stops when the difference in the mean queue length and the difference in the mean vacation period for all the queues is less than 10^{-8}. In the examples ran, the number

Table 3.6: 6 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.75$

Queue	λ	T_{max}	W_{app}	W_{sim}	% Error
1	0.15	5	10.235	8.971 ± 0.374	-14.1%
2	0.1	4	14.534	13.494 ± 0.741	-7.7%
3	0.15	6	11.554	9.963 ± 0.268	-16.0%
4	0.05	3	12.496	12.008 ± 0.466	-4.1%
5	0.15	5	10.235	9.098 ± 0.480	-12.5%
6	0.1	4	14.534	13.778 ± 0.648	-5.5%

of iterations required for convergence depends on the number of queues, Q, and the allocated time, T, for each queue. This is because when T is large, the dimension of the phase type distribution of the vacation period becomes large, thus, a large rate matrix to solve for. Similarly, when Q is large we have to solve for Q rate matrices and queue lengths (assuming that the polling system is not symmetric).

3.6.2 Finite Capacity Model

Tables 3.8-3.11 show the comparison between the simulation and the approximate approach of the mean waiting time for a 4 queue polling system. The service time distribution is identical for all queues and is of the geometric type. Customers arrive according to the geometric distribution and the probability of arrival is given by λ. The maximum time allocated for each queue is given by T_{max}, and the buffer size by K.

Table 3.12 shows the performance of the iterative procedure when the system utilization is larger than 1. Notice that queues 1 and 3 have the same buffer sizes, but queue 3 has a larger allocated time, thus it has a lower mean waiting time. This is because the blocking probability for queue 3 is smaller. Similar arguments can be

Table 3.7: 8 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.75$

Queue	λ	T_{max}	W_{app}	W_{sim}	% Error
1	0.06	3	14.093	15.031 ± 0.307	6.2%
2	0.06	4	10.737	9.400 ± 0.130	−14.2%
3	0.12	5	13.700	12.109 ± 0.151	−13.1%
4	0.06	3	14.093	15.114 ± 0.191	6.8%
5	0.12	5	13.700	12.061 ± 0.132	−13.6%
6	0.06	4	10.737	9.441 ± 0.136	−13.7%
7	0.06	3	14.093	15.091 ± 0.236	6.6%
8	0.06	4	10.737	9.406 ± 0.060	−14.2%

made for queues 2 and 4.

In all the examples studied, the iterative algorithm converges in less than 30 iterations, with a run time less than 15 minutes. Most of the CPU time during each run is used up in solving for the queue length distribution. The queue length is obtained using block Gauss-Seidel iterative procedure with a stopping criterion for convergence of 10^{-10}. The block dimension is of size mn, where m is the dimension

Table 3.8: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.75$

Queue	λ	T_{max}	K	W_{app}	W_{sim}	% Error
1	0.2	5	8	9.069	7.828 ± 0.072	−15.9%
2	0.1	4	5	7.371	6.767 ± 0.116	−8.9%
3	0.2	6	8	5.673	6.273 ± 0.100	9.6%
4	0.1	3	5	8.220	9.259 ± 0.130	11.2%

Table 3.9: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.6875$

Queue	λ	T_{max}	K	W_{app}	W_{sim}	% Error
1	0.15	5	8	5.541	5.619 ± 0.063	1.4%
2	0.1	4	5	5.095	5.602 ± 0.099	9.0%
3	0.2	6	8	5.894	5.120 ± 0.040	−15.1%
4	0.1	3	5	7.903	7.526 ± 0.124	−5.0%

Table 3.10: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.625$

Queue	λ	T_{max}	K	W_{app}	W_{sim}	% Error
1	0.1	5	8	3.820	3.784 ± 0.082	−1.0%
2	0.1	4	5	4.020	4.463 ± 0.038	9.9%
3	0.2	6	8	4.338	4.074 ± 0.021	−7.7%
4	0.1	3	5	5.707	5.989 ± 0.061	4.7%

of the service time distribution and n is the dimension of the arrival process. The stopping criterion for convergence for the iterative polling algorithm was chosen to be $\epsilon = 10^{-8}$ i.e. the program stops when the difference in the mean queue length and the difference in the mean vacation period for all the queues is less than 10^{-8}. In the examples run, the number of iterations required for convergence depends on the number of queues, Q, the buffer size of each queue, K, and the allocated time for each queue, T. This is because when T and/or K is large, the dimension of the probability vector becomes large. Similarly, when Q is large we have to solve for Q queue lengths (assuming that the polling system is not symmetric).

Similarly to the infinite buffer capacity case, the maximum percentage error for the finite buffer capacity case is ±15%. Both models yielded reasonable results for

3.7 Variant of the Model

In this section, we describe how to take the model for the polling system without switch-over time and modify it to account for a switch-over time between the queues. The only change necessary is for the vacation period distribution. Let the switch-over time between queue i and queue $i+1$ be of phase type distribution with the following representation $(\boldsymbol{\kappa}_i, U_i)$. Let $\mathbf{U}^\circ$ be the absorption vector for the phase type distribution. The initial probability vector $\boldsymbol{\kappa}$ has 1 in the first position and zero every where else. With this information we construct the independent part of the vacation period given in Section 3.4. For the example given there the vacation period becomes

$$Z_A = \begin{bmatrix} U_B & \mathbf{U}_B^\circ\gamma_B & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & B_B & \mathbf{B}_B^\circ\boldsymbol{\kappa}_C & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & U_C & \mathbf{U}_C^\circ\gamma_C & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B_C & \mathbf{B}_C^\circ\boldsymbol{\kappa}_D & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & U_D & \mathbf{U}_D^\circ\gamma_D & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & B_D & \mathbf{B}_D^\circ\boldsymbol{\kappa}_E & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & U_E & \mathbf{U}_E^\circ\gamma_E & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_E & \mathbf{B}_E^\circ\boldsymbol{\kappa}_A \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & U_A \end{bmatrix}$$

$$Z_A^\circ = \begin{bmatrix} \mathbf{0} \\ \mathbf{U}_A^\circ \end{bmatrix}$$

$$\boldsymbol{\delta}_A = \begin{bmatrix} \boldsymbol{\kappa}_B & 0 & \dots \end{bmatrix}$$

Block $Z_A(1,1)$ denotes the beginning of the switch-over time from queue A to queue B. Block $Z_A(1,2)$ denotes the end of the switch-over time and the beginning

of service in queue B. The remaining of the blocks of the transition matrix Z_A can be explained in the same way. Note that the dependent part of the vacation period would be computed using an approximate approach similar to that of Section 3.5.

CHAPTER 4

TIME-LIMITED TABLE POLLING

4.1 Introduction

In Chapter 3 an iterative procedure for the exhaustive time-limited service discipline for cyclic polling systems was presented. In this Chapter we extend those results to the case of table polling. Consider a multi-queue system with Q queues visited periodically according to a table of size N, $N \geq Q$. This type of polling includes star polling, elevator polling, cyclic polling and custom-made tables. For each queue i; $i = 1, \ldots Q$; of this polling system:

- Customers arrive according to an m-dimensional Markovian arrival process with representation $(D_{0,i}, D_{1,i})$ and fundamental rate λ_i.
- The service time distribution is an n-dimensional phase type with representation $(\boldsymbol{\beta}_i, S_i)$.
- The service discipline for queue i is exhaustive time-limited with time limit T_i. Notice that the time limit is hard preemptive i.e. an on-going service is interrupted at the time limit and in the next visit the server resumes serving the customer where it left off. Notice that the time threshold, T_I, is fixed for a station for all visits in a table.
- Each queue is visited M_i times, $M_i \geq 1$, in a table.
- The switch-over time between queue i and queue $i+1$ is equal to zero.

Similar to the cyclic polling case, the analysis is based on the decomposition method. Each queue is considered separately as MAP/PH/1 queue with vacation in the case of infinite buffer capacity model and as a MAP/PH/1/K queue with vacation in the case of finite buffer capacity model. The analysis of MAP/PH/1 and of MAP/PH/1/K with phase type vacation distribution is presented in Chapter 3. In the following we discuss how to extend those results to the case of multiple vacations.

The visit period has a phase type distribution with dimension T_i and representation (γ_i, B_i), $i = 1, \ldots Q$. The visit period distribution is obtained in the same way as in Section 3.4 for the case of infinite buffer capacity and as in Section 3.5 for the finite buffer case. The necessary changes to account for multiple vacations are presented in Section 4.3. Similarly to the vacation period in Section 3.4, each sub-cycle vacation period (i.e. the time between successive visits to the same queue in the table) can be represented by a phase type distribution with dimension $\sum_{j \in SC} T_j$, where SC is the set of queues visited during the sub-cycle vacation period. Therefore, if queue i, $i = 1, \ldots Q$, is visited M_i, $M_i \geq 1$, times in a table, then queue i has M_i phase type vacations which we denote by $(\boldsymbol{\delta}_k, L_k)$; $k = 1, \ldots M_i$.

From the distribution of each of the sub-cycle vacation period we can construct the vacation period distribution for each queue by noting that the type of sub-cycle vacation the server takes depends on its current position in the table. This correlation between the position in the table and the type of vacation the server takes can be captured using MAP. Thus, we denote by $(V_{0,i}, V_{1,i})$ the vacation period distribution of queue i. In this notation element (u, v) of matrix $V_{0,i}$ denotes transition from state u to state v with the vacation period still going-on and the element (l, k) of the matrix $V_{1,i}$ denotes transition from state l to state k with the vacation period ending from state l and the next vacation period beginning from state k. For clarity and ease of notation from here on the queue index is used only when it is absolutely necessary.

As an example consider a polling system with five queues $\{A,\ B,\ C,\ D,\ E\}$. Furthermore, suppose that we have to:

- Visit queue A 2 times per cycle with $T_A = 4$.
- Visit queue B 1 times per cycle with $T_B = 5$.
- Visit queue C 2 times per cycle with $T_C = 3$.
- Visit queue D 1 times per cycle with $T_D = 6$.
- Visit queue E 1 times per cycle with $T_E = 4$.

A possible polling table is given as A, B, C, A, D, E, C. In this polling table, queues A, and C are served two times in this table with each visit having a maximum length equal to 4 units of time, and 3 units of time, respectively. Queues B, D, and E are visited once for a maximum of 5, 6, and 4, respectively. Let us consider the vacation period of queue A. Since queue A is served two times it has two sub-cycles. The first consists of queues $\{A, B, C\}$ and the second $\{A, D, E, C\}$. Therefore, during the first sub-cycle vacation period the server visits queues $\{B, C\}$ and thus has a dimension equal to $T_B + T_C = 8$. Let this vacation be denoted by $(\boldsymbol{\delta}_1, L_1)$, which is given by:

$$L_1 = \begin{bmatrix} B_B & \mathbf{B}_B^{\circ}\gamma_C \\ 0 & B_C \end{bmatrix}, \quad \mathbf{L}_1^{\circ} = \begin{bmatrix} 0 \\ \mathbf{B}_C^{\circ} \end{bmatrix},$$

$$\boldsymbol{\delta}_1 = \begin{bmatrix} \gamma_B & 0 \end{bmatrix},$$

where, $\mathbf{B}^{\circ} = \mathbf{e} - B\mathbf{e}$. $\mathbf{e}$ is a column vector of 1's. The probabilistic interpretation of the above distribution is as follows. Once the server leaves queue A it goes on vacation. The initial probability vector of the vacation period is given by $\boldsymbol{\delta}_1$. Since after leaving queue A, the server goes to queue B we have the initial probability vector of the visit period of queue B, γ_B, in the first position of $\boldsymbol{\delta}_1$. Once the server is in queue B, it stays there according to the visit period distribution of queue B. This is given by B_B in position $L_1(1,1)$. At the end of the visit period in queue B, the server moves to queue C. Therefore, we have absorption according to $\mathbf{B}_B^{\circ}$ and

beginning of service in queue C according to γ_C which is given in position $L_1(1,2)$. Then, service go on at queue C according to its visit period distribution B_C at the end of which the sub-cycle vacation period is over. Thus, the absorption vector $\mathbf{L}_1^\circ$ has $\mathbf{B}_C^\circ$ in position $\mathbf{L}_1^\circ(2,1)$.

In the second sub-cycle vacation, the server visits queues $\{D,\ E,\ C\}$, thus its dimension equals to $T_D + T_E + T_C = 13$. Let this vacation be denoted by $(\boldsymbol{\delta}_2, L_2)$ and is given by:

$$L_2 = \begin{bmatrix} B_D & \mathbf{B}_D^\circ \gamma_E & 0 \\ 0 & B_E & \mathbf{B}_E^\circ \gamma_C \\ 0 & 0 & B_C \end{bmatrix}, \quad \mathbf{L}_2^\circ = \begin{bmatrix} 0 \\ 0 \\ \mathbf{B}_C^\circ \end{bmatrix},$$

$$\boldsymbol{\delta}_2 = \begin{bmatrix} \gamma_D & 0 \end{bmatrix}.$$

The arguments leading to $(\boldsymbol{\delta}_2, L_2)$ are the same as those for $(\boldsymbol{\delta}_1, L_1)$. A schematic diagram for queue A with two types of vacation is shown in Fig. 4.1. Using $(\boldsymbol{\delta}_1, L_1)$ and $(\boldsymbol{\delta}_2, L_2)$ the vacation period of queue A is given by:

$$V_0 = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix}, \quad V_1 = \begin{bmatrix} 0 & \mathbf{L}_1^\circ \boldsymbol{\delta}_2 \\ \mathbf{L}_2^\circ \boldsymbol{\delta}_1 & 0 \end{bmatrix}.$$

The distribution (V_0, V_1) indicates that the server stays on vacation according to V_0 and that V_1 indicates the end of the vacation period and which type of vacation the server will take next. $V_0(1,1)$ indicates that the server is taking a vacation according to $(\boldsymbol{\delta}_1, L_1)$ with its end denoted by $V_1(1,2)$. Furthermore, $V_1(1,2)$ indicates that the next time the server goes on vacation from queue A its vacation period will be given by $(\boldsymbol{\delta}_2, L_2)$, thus, we have $\mathbf{L}_1^\circ \boldsymbol{\delta}_2$ in $V_1(1,2)$. Similar arguments lead to positions $V_0(2,2)$ and $V_1(2,1)$.

In general, the vacation period of a queue visited M times in a cycle is given by:

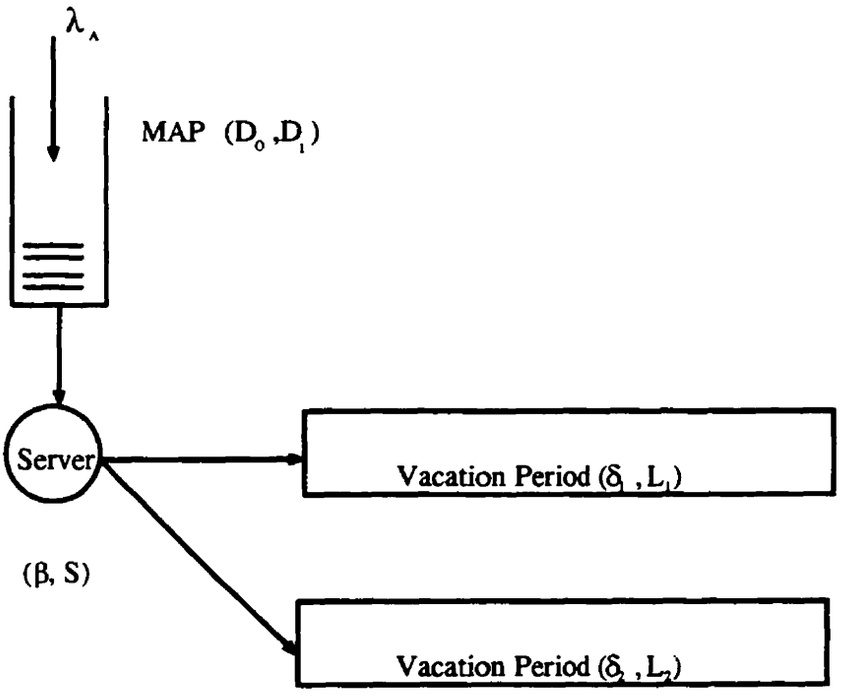

Figure 4.1: Vacation Model

$$V_0 = \begin{bmatrix} L_1 & 0 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & L_{M-1} & 0 \\ 0 & 0 & 0 & 0 & L_M \end{bmatrix},$$

$$V_1 = \begin{bmatrix} 0 & \mathbf{L}_1^{\circ}\boldsymbol{\delta}_2 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{L}_2^{\circ}\boldsymbol{\delta}_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{L}_{M-1}^{\circ}\boldsymbol{\delta}_M \\ \mathbf{L}_M^{\circ}\boldsymbol{\delta}_1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

In addition to the vacation period distribution, we introduce four additional matrices V_2, V_3, V_4, V_5. The matrix V_2 is defined for transition from the end of vacation to the beginning of service at queue i. It is defined for the case when the server, upon return from vacation, resumes service for a customer whose service has been interrupted in the previous queue visit. It is given by:

$$V_2 = \begin{bmatrix} 0 & \mathbf{L}_1^\circ \boldsymbol{\beta}^* & 0 & 0 & 0 \\ 0 & 0 & \mathbf{L}_2^\circ \boldsymbol{\beta}^* & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{L}_{M-1}^\circ \boldsymbol{\beta}^* \\ \mathbf{L}_M^\circ \boldsymbol{\beta}^* & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the probability vector $\boldsymbol{\beta}^*$ is the initial probability vector of resuming an interrupted service.

The matrix V_3 represents transitions from the end of vacation to the beginning of service at queue i. It is for the case when the server, upon return from vacation, starts service of a new customer and is given by:

$$V_3 = \begin{bmatrix} 0 & \mathbf{L}_1^\circ \boldsymbol{\beta} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{L}_2^\circ \boldsymbol{\beta} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{L}_{M-1}^\circ \boldsymbol{\beta} \\ \mathbf{L}_M^\circ \boldsymbol{\beta} & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the probability vector $\boldsymbol{\beta}$ is the initial probability vector of the service time distribution.

The matrix V_4 represents the transition from a visit period to a vacation period. It is for the case when the server finishes serving a customer and goes on vacation

because the queue is empty, it is given by:

$$V_4 = \begin{bmatrix} 0 & \mathbf{S}^\circ\boldsymbol{\delta}_2 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{S}^\circ\boldsymbol{\delta}_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{S}^\circ\boldsymbol{\delta}_M \\ \mathbf{S}^\circ\boldsymbol{\delta}_1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the probability vector $\mathbf{S}^\circ$ denotes the absorption vector of the service time distribution.

Lastly, the matrix V_5 represents transition from a visit period to a vacation period. It denotes the case when the server interrupts an on-going service because of the time-limit and goes on vacation, it is given by:

$$V_5 = \begin{bmatrix} 0 & S\mathbf{e}\boldsymbol{\delta}_2 & 0 & 0 & 0 \\ 0 & 0 & S\mathbf{e}\boldsymbol{\delta}_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & S\mathbf{e}\boldsymbol{\delta}_M \\ S\mathbf{e}\boldsymbol{\delta}_1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the probability vector $S\mathbf{e}$ denotes service interruption.

The matrices V_0, V_1, V_2, V_3, V_4, and V_5 are used to modify the vacation models in Sections 3.4.1 and 3.4.2 as follow:

- The matrix V is replaced with the matrix V_0.
- The absorption vector $\mathbf{V}^\circ$ is replaced with the matrix V_1. In addition, the inner product of $\mathbf{V}^\circ$ with a vector is changed to a Kronecker product between the matrix V_1 and the vector (e.g. $\mathbf{V}^\circ\boldsymbol{\beta}$ becomes $V_1 \otimes \boldsymbol{\beta}$).
- The matrix $\mathbf{V}^\circ\boldsymbol{\beta}^*$ is replaced with the matrix V_2.

- The matrix $\mathbf{V}^{\circ}\boldsymbol{\beta}$ is replaced with V_3.
- The matrix $\mathbf{S}^{\circ}\boldsymbol{\kappa}$ is replaced with the matrix V_4.
- The matrix $(S\mathbf{e})\boldsymbol{\kappa}$ is replaced with the matrix V_5.

This ends our description of the construction of the vacation period distribution for a given queue. In the next section we show how to obtain the visit period distribution.

4.2 Duration of a Queue Visit

The visit period distributions for both the infinite and finite buffer capacity case are computed in the same way as in Subsections 3.4.1.1 and 3.4.2.1, respectively, with the following change:

$$\theta_0 = \bar{v}\mathbf{x}_0(\mathbf{e}\otimes I)V_1\mathbf{e}/v_0, \tag{4.1}$$

$$\theta_i = \bar{v}\mathbf{x}_{i,0}(\mathbf{e}\otimes I)V_1\mathbf{e}/v_0, \quad 1\leq i\leq T-1, \tag{4.2}$$

where the vector $\mathbf{x}_0$ is the probability that the queue is empty and $\mathbf{x}_{i,0}$ is the probability that there are i customers in the queue at the end of the vacation period. $\bar{v}$ is the mean of the vacation period.

4.3 Iterative procedure

The dependence of the visit period distribution on the vacation period distribution and vice versa is clear from the previous two sections. Therefore, we use an iterative procedure to obtain the performance measures of a queue in the polling system. In this procedure, the vacation and visit period distributions in iteration l are used to update the vacation and visit period distributions in iteration $l+1$. Our experimentation with

the algorithm showed that the results are, under certain load combination, very far from the simulation results. This is attributed to the inherent dependency between the vacation and the visit period distributions. In order to improve the accuracy of the iterative procedure we have adopted the method by Lee and Sengupta [109]. In their method the vacation period is considered to be a combination of an independent part and a dependent part. We adopt their algorithm as follows. For each queue and for each sub-cycle vacation period we construct a phase type distribution (ψ, Y). This distribution is computed differently when $\rho < 0.65$ and $\rho \geq 0.65$.

For the case of $\rho < 0.65$ the dependent part of the vacation period is computed using Algorithm 2 of Lee and Sengupta [109]. In our case, the phase type distribution $(\psi, Y)_i$, $i \leq M$, is given by: $\psi = [1\ 0 \ldots\ 0]$ and

$$Y = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & \cdots & \vdots \\ 0 & \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ 0 & \vdots & 0 & p'(L-1) & 0 & \cdots & \vdots \\ 0 & \vdots & 0 & 0 & p'(L) & 0 & \cdots \\ 0 & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & p'(\sum_{j=1}^{L} T_j) \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix},$$

where

$$\begin{aligned} p'(L-1) &= 1 - p(L-1), \\ p'(j) &= 1 - p(j)(1 - \sum_{i=L-1}^{J} p(i))^{-1}, \text{ and } J = \sum_{i \in SC} T_i, \end{aligned}$$

where L is the number of queues visited in the sub-cycle. The probability $p(j)$ is computed using Lee and Sengupta [109] (Algorithm 2). The $p(j)$'s are computed in the same way as in the case of cyclic polling.

For the case of $\rho \geq 0.65$ the dependent part of the vacation period is assumed to be deterministic. Thus for queue i the length of the vacation period is given by $\sum_{j,\, j \in SC} T_j$, where SC is the set of queues in the vacation period of the sub-cycle. This distribution can be represented by a phase type distribution with 1's in the super-diagonal positions and 0's every where else. This is because for moderate to high system utilization, once the server goes on vacation from queue i it is more likely to stay way serving each of the other queues for its whole time period.

The iterative procedure for table polling systems with infinite buffer capacity queues is outlined below:

1. For $i = 1$ to Q initialize the distribution of the visit period, (γ_i, B_i), to

$$B_i = \begin{bmatrix} 0 & \rho_i & 0 & \cdots & 0 \\ 0 & 0 & \rho_i & \cdots & \vdots \\ 0 & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 0 & \rho_i \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix}, \tag{4.3}$$

   and $\gamma_i = [1 \;\; \mathbf{0}]$.

2. For $i = 1$ to Q

   - For $j = 1$ to M_i

     a) Compute (γ_j, L_j)

     b) Compute (ψ_j, Y_j)

     c) Compute P_i according to Algorithm 1.

     d) Let $(\gamma_j, L_j) = (1 - P_i) \times (\gamma_j, L_j) + P_i(\psi_j, Y_j)$

   - Compute $V_0^i,\ V_1^i,\ V_2^i,\ V_3^i,\ V_4^i,\ V_5^i$

   - Compute the block matrices $A_0^i,\ A_1^i,\ A_2^i$

- Compute the rate matrix R^i
- Compute the probability vector $[\mathbf{x}_0\ \mathbf{x}_1]^i$
- Compute μ_L^i and W_L^i.
- Update the visit period distribution.

3. If the average queue length and the mean waiting vacation period did not converge go back to 2, else stop.

Algorithm 1

- If $\rho \leq 0.50$ set $P_i = (1-\rho)\rho^Q$,
- If $0.5 < \rho \leq 0.75$ set $P_i = (1-\rho)\rho^Q + \rho_i$,
- If $0.75 < \rho$ set $P_i = \rho^Q + \rho_i$.

The iterative procedure for table polling systems with finite buffer queues is outlined below:

1. For $i = 1$ to Q initialize the distribution of the visit period, (γ_i, B_i) according to (Eq. 4.3).
2. For $i = 1$ to Q
   - For $j = 1$ to M_i
     a) Compute (γ_j, L_j)
     b) Compute (ψ_j, Y_j)
     c) Compute P_i according to Algorithm 2.
     d) Let $(\gamma_j, L_j) = (1-P_i) \times (\gamma_j, L_j) + P_i(\psi_j, Y_j)$
   - Compute $V_0^i,\ V_1^i,\ V_2^i,\ V_3^i,\ V_4^i,\ V_5^i$
   - Compute the block matrices $A_0^i,\ A_1^i,\ A_2^i$
   - Compute the probability vector $[\mathbf{x}_0\ \mathbf{x}_1\ \cdots\ \mathbf{x}_{K_1}]^i$

- Compute μ_L^i and W_L^i.
- Update the visit period distribution.

3. If the average queue length did not converge go back to 2, else stop.

Notice that for the finite buffer case, we used the block Gauss-Seidel iterative approach to obtain the queue length distribution.

Algorithm 2

- If $\rho \leq 0.50$ set $P_i = (1-\rho)\rho^Q$,
- If $0.5 < \rho \leq 0.75$ set $P_i = (1-\rho)\rho^Q + \rho_i$,
- If $0.75 < \rho < 1.0$ set $P_i = \rho^Q + \rho_i$.
- If $1.0 \leq \rho$ set

$$P_i = \begin{cases} 1-\rho_i & \text{if } \rho - \rho_i < 1 \\ 1 & \text{otherwise} \end{cases}$$

where $\bar{v}_i$ is the mean vacation period for queue i. Notice that for the finite buffer case it is not necessary to have $\rho < 1$. Therefore, if a queue is unstable, its visit period distribution is deterministic and equals to T_{max}.

4.4 Numerical Examples

In comparing the approximate approach results with those of the simulation we define the percent error as $\frac{W_{sim}-W_{app}}{W_{sim}}$, where W_{sim} is the simulation mean waiting time and W_{app} is the iterative approach mean waiting time. A negative percent error indicates that the approximate method over-estimates the simulation results. For the simulation we used 5 replications each of which is of 10^6 time units long and a warm up period of 10^5 time units, we show the half-width of the 95% confidence interval.

Since, there are many system parameters we chose to show some of the results to give a general idea on the performance of the iterative procedure.

4.4.1 Infinite Capacity Model

The first set of examples shows the performance of the iterative method for different system utilization. Tables 4.1-4.7 shows the comparison between the simulation and the approximate approach of the mean waiting time for table polling systems with infinite queue buffer capacity. In all the examples, customers arrive according to the Bernoulli process with an arrival rate given by λ and their service times is geometrically distributed with mean $\bar{b}$. The maximum time period for each queue visit and the system utilization are given by T_{max} and ρ, respectively. The polling order is given in the tables by Pol-Order.

The maximum absolute error, 17%, for these examples is encountered with a high system utilization, $\rho = 0.861$, for queue 2 in Table 4.7. All other examples have a maximum percentage error of $\pm 15\%$.

Table 4.1: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.625$

Poll-Order	1 2 3 2 4 2				
Queue	λ	T_{max}	W_{App}	W_{Sim}	% Error
1	0.1	5	5.842	5.791 ± 0.067	-0.9%
2	0.2	5	3.049	3.288 ± 0.018	7.3%
3	0.1	5	5.842	5.799 ± 0.056	-0.7%
4	0.1	5	5.842	5.751 ± 0.072	-1.6%

Table 4.1 shows the mean waiting time for a table polling system with 4 queues and a utilization of $\rho = 0.625$. This table represent a star polling system with queue 2 being visited after every other queue. This can represent the special case

of queue 2 being a high priority queue which is polled after every low priority poll, queues {1, 3, 4}. It can also represent a half duplex transmission medium between queues {1,3,4} and a central server represented by queue 2 where after every poll the central server transmits its outbound traffic. All four queues have the same maximum time threshold, however, queue 2 has twice the arrival rate of queues {1, 3, 4}. Nevertheless, queue 2 has a lower mean waiting time. This is because queue 2 is visited more frequently. The maximum error in this example is encountered for queue 2 and is about 7%.

Table 4.2 shows the mean waiting time for a polling system with 4 queues and a utilization of $\rho = 0.5$. The polling order is the same as that in Table 4.1. All four queues have the same arrival rate. Although, Queue 2 has a smaller maximum visit period, 3 time units, compared with queues {1, 3, 4}, its mean waiting time is lower than that of queues {1, 3, 4}. Again, this is due to the higher number of visits to queue 2 compared to those of queues {1, 3, 4}. The maximum error in this example is encountered for queue 2 and is about −16%. Notice the difference in the % decrease of the mean waiting time between queues {1, 3, 4} and queue 2. This is because the decrease in load happens at queue 2, i.e. a lighter load at queue 2 results in a greater decrease in the mean waiting times for queues {1, 3, 4} than for queue 2.

Table 4.2: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.5$

Poll-Order	1 2 3 2 4 2				
Queue	λ	T_{max}	W_{App}	W_{Sim}	% Error
1	0.1	5	3.476	2.996 ± 0.019	−16.0%
2	0.1	3	2.189	2.591 ± 0.016	15.5%
3	0.1	5	3.476	3.013 ± 0.005	−15.4%
4	0.1	5	3.476	3.006 ± 0.007	−15.6%

The next two examples exhibit the behavior of a polling system with 4 queues. First we consider the case when each queue is visited twice, given in Table 4.3. Then, we consider the case of increasing the number of visits to queue 3 to 3 times and reduce the number of visits to queue 4 to 1. This is because queue 3 has the largest arrival rate and queue 4 has the smallest arrival rate. In order to counter balance the decrease in the number of visits of queue 4 we increase its maximum visit period to 5 time units which is given in Table 4.4. Although the number of visits to queues {1, 2} did not change their mean waiting times are higher in Table 4.4. This is due to the asymmetry introduced into the system. For instance, for queue 1 in Table 4.3 every time the server leaves the queue it visits queues {2, 3, 4}. However, in Table 4.4 when the server leaves queue 1 it may visit queues {2, 3} or queues {3, 4, 2, 3}. The same argument can be made for queue 2. On the other hand, queue 3 is visited more frequently in Table 4.4 which explains the lower mean waiting time. Lastly, although the maximum visit period of queue 4 has been increased to 5 time units, because queue 4 is visited only once in Table 4.4 its mean waiting time has increased compared to Table 4.3.

Table 4.3: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.65625$

Poll-Order	1 3 2 4 1 3 4 2				
Queue	λ	T_{max}	W_{App}	W_{Sim}	% Error
1	0.15	5	5.145	5.796 ± 0.061	11.2%
2	0.1	4	5.402	6.017 ± 0.040	10.2%
3	0.2	6	4.907	5.614 ± 0.038	12.5%
4	0.075	3	8.397	7.412 ± 0.049	−13.3%

Table 4.5 is for a polling system with 5 queues. Even though queue 1 has a higher arrival rate, because it has a larger T_{max} and it is visited twice during a cycle

Table 4.4: 4 Queues Polling System, $\tilde{b} = 1.25$, $\rho = 0.65625$

Poll-Order:	1 2 3 1 3 4 2 3				
Queue	λ	T_{max}	W_{App}	W_{Sim}	% Error
1	0.15	5	6.449	6.209 ± 0.059	-3.9%
2	0.1	4	5.871	6.370 ± 0.055	7.83%
3	0.2	6	3.313	3.928 ± 0.017	15.7%
4	0.075	5	8.907	8.094 ± 0.046	-10.0%

its mean waiting time is almost half that of queues {2, 3, 4, 5}. Although queues {2, 3, 4, 5} have the same arrival rate and maximum time period their approximated mean waiting time is not the same. This is due to the way the vacation period is built. Notice that queues {2, 4}, and {3, 5}, have the same vacation period pattern, thus, they have the same mean waiting times.

Table 4.5: 5 Queues Polling System, $\tilde{b} = 1.25$, $\rho = 0.75$

Poll-Order	1 2 3 1 4 5				
Queue	λ	T_{max}	W_{App}	W_{Sim}	% Error
1	0.2	5	6.563	7.284 ± 0.055	9.9%
2	0.1	4	13.346	15.557 ± 0.333	14.2%
3	0.1	4	13.376	15.536 ± 0.173	13.9%
4	0.1	4	13.346	15.388 ± 0.154	13.3%
5	0.1	4	13.376	15.586 ± 0.220	14.2%

The next two examples, Table 4.6 and 4.7, show the performance of the iterative

procedure as we increase the number of queues. Notice here that the approximate solution does not give the exact same result for identical queues. This is due to the way the vacation period are built. Notice, again, that queues with similar vacation pattern have similar mean waiting times (e.g. queues $\{3, 7\}$ in Table 4.7).

Table 4.6: 6 Queues Polling System, $\bar{b} = 1.111$, $\rho = 0.778$

Poll-Order	1 2 3 1 4 5 1 6				
Queue	λ	T_{max}	W_{App}	W_{Sim}	% Error
1	0.2	5	5.157	5.512 ± 0.060	6.5%
2	0.1	6	10.289	10.357 ± 0.188	0.7%
3	0.1	6	10.300	10.386 ± 0.143	0.8%
4	0.1	6	10.288	10.346 ± 0.125	0.6%
5	0.1	6	10.300	10.483 ± 0.174	1.8%
6	0.1	6	10.299	10.444 ± 0.110	1.4%

In all the examples studied, the iterative algorithm converges in less than 30 iterations, with a run time under 30 minutes on an IBM RS6000/590. The simulation run on a Sun station lightly loaded were over an hour long. Most of the CPU time for each run for the iterative procedure is taken to solve for the rate matrix R, and the queue length distribution. The rate matrix R of dimension $n(r + mT_{max})$, where n is the dimension of the service time distribution; m is the dimension of the arrival process; r is the dimension of the vacation period distribution, is obtained using the algorithm given in Alfa [2] . The stopping criterion for convergence was chosen to be $\epsilon = 10^{-8}$ i.e. the program stops when the difference in the mean queue length and the difference in the mean vacation period for all the queues between two consecutive iterations is less than 10^{-8}. In the examples run, the number of iterations

inter-arrival time between customers is given by a geometric distribution and the probability of arrival is by λ. The maximum time allocated for each queue and the buffer size are given by T_{max} and K, respectively.

The system parameters for Table 4.8 are the system as those in Table 4.1 with the exception that the buffer capacity is finite and is equal to 8 for all queues. Notice

Table 4.8: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.625$

Poll-Order	1 2 3 2 4 2					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.1	5	8	5.623	5.426 ± 0.030	-3.6%
2	0.2	5	8	3.023	3.064 ± 0.013	1.4%
3	0.1	5	8	5.632	5.342 ± 0.039	-5.4%
4	0.1	5	8	5.637	5.412 ± 0.021	-4.2%

that because of the blocking probabilities the mean waiting times in Table 4.8 are consistently lower than those in Table 4.1. However, in Table 4.9 even though the buffer capacity is smaller, 7 for each queue, the mean waiting times are comparable to those in Table 4.2. This is mainly due to the lower system utilization, $\rho = 0.5$ in Table 4.9 and $\rho = 0.625$ in Table 4.8. The same observation can be made about Tables 4.10 and 4.11.

Table 4.10 and 4.11 show, similarly to the case of infinite capacity case, that changing the polling order by increasing the frequency of visits have a significant impact on the mean waiting time for all the queues in the system, even those which kept the same number of visits.

Table 4.13-4.15 show the mean waiting times for a polling system as we move from a high utilization case (Table 4.13) to overload cases, $\rho > 1.0$, by first increasing the arrival rate for each queue (Table 4.14) and increasing the mean service time

Table 4.9: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.5$

Poll-Order	1 2 3 2 4 2					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.1	5	7	3.403	2.939 ± 0.010	-15.8%
2	0.1	3	7	2.172	2.567 ± 0.008	15.4%
3	0.1	5	7	3.403	2.939 ± 0.010	-15.8%
4	0.1	5	7	3.403	2.948 ± 0.008	-15.8%

(Table 4.15). In both cases the iterative method gives reasonable results.

In all the examples studied, the iterative algorithm converges in less than 30 iterations, with a run time less than 45 minutes. Most of the CPU time during each run is used up in solving for the queue length distribution. The queue length is obtained using block Gauss-Seidel iterative procedure with a stopping criterion for convergence of 10^{-10}. The block dimension is of size mn, where n is the dimension of the service time distribution and m is the dimension of the arrival process. The stopping criterion for convergence for the iterative polling algorithm was chosen to be $\epsilon = 10^{-8}$ i.e. the program stops when the difference in the mean queue length and the difference in the mean vacation period for all the queues is less than 10^{-8}. In the examples run, the number of iterations required for convergence depends on the number of queues, Q, the table size, N, the buffer size, K, and the allocated time, T, and the number of visits, M, for each queue. This is because when T, M, and/or K is large, the dimension of the probability vector becomes large. Similarly, when Q is large we have to solve for Q queue lengths (assuming that the polling system is not symmetric).

Table 4.10: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.65625$

Poll-Order	1 3 2 4 1 3 4 2					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.15	5	8	4.340	5.045 ± 0.023	14.0%
2	0.1	4	8	4.886	5.533 ± 0.032	12.0%
3	0.2	6	8	3.961	4.631 ± 0.015	14.5%
4	0.075	3	8	6.653	6.799 ± 0.032	2.1%

4.5 Conclusions

We have presented an iterative procedure for the analysis of discrete time table polling systems with Markovian arrival process, phase type service time distribution and exhaustive time-limited service discipline (preemptive). The numerical examples run show that the algorithm converges relatively fast and gives reasonable results. However, we did not prove that the algorithm converges. This remains to be done. It is also worth mentioning that as the number of queues, threshold time, or the table size increases, the time required to compute the performance measures increases. Nonetheless, the iterative procedure still remains a better option compared to simulation since it takes less time.

Although, the model presented in this Chapter assumes that the time threshold for every queue is the same for every visit in the table, it possible to extend the model to the case where a queue may have different threshold for each visit. This can be done by first changing the Markov chain describing the MAP/PH/1 queue for the infinite buffer case and MAP/PH/1/K queue for the finite buffer case. The necessary modifications are given in Appendix A.

Table 4.11: 4 Queues Polling System, $\bar{b} = 1.25$, $\rho = 0.65625$

Poll-Order:	1 2 3 1 3 4 2 3					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.15	5	8	6.311	5.393 ± 0.025	-17.0%
2	0.1	4	8	5.338	5.851 ± 0.030	8.8%
3	0.2	6	8	3.824	3.536 ± 0.015	-8.1%
4	0.075	5	8	8.248	7.343 ± 0.034	-12.3%

Table 4.12: 6 Queues Polling System, $\bar{b} = 1.111$, $\rho = 0.778$

Poll-Order	1 2 3 1 4 5 1 6					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.2	5	8	3.842	4.473 ± 0.019	14.1%
2	0.1	6	8	8.780	8.404 ± 0.030	-4.5%
3	0.1	6	8	8.783	8.411 ± 0.036	-4.4%
4	0.1	6	8	8.787	8.435 ± 0.032	-4.2%
5	0.1	6	8	8.794	8.471 ± 0.029	-3.8%
6	0.1	6	8	8.804	8.429 ± 0.031	-4.4%

Table 4.13: 7 Queues Polling System, $\bar{b} = 1.111$, $\rho = 0.861$

Poll-Order	1 2 3 1 4 5 2 1 6 7					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.15	5	8	5.693	6.610 ± 0.086	13.9%
2	0.125	6	8	7.073	7.769 ± 0.023	9.0%
3	0.1	7	8	14.660	13.466 ± 0.059	−8.9%
4	0.1	7	8	14.665	13.446 ± 0.052	−9.1%
5	0.1	7	8	14.682	13.515 ± 0.072	−8.6%
6	0.1	7	8	14.708	13.507 ± 0.043	−8.9%
7	0.1	7	8	14.758	13.558 ± 0.091	−8.9%

Table 4.14: 7 Queues Polling System, $\bar{b} = 1.111$, $\rho = 1.028$

Poll-Order	1 2 3 1 4 5 2 1 6 7					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.175	5	8	11.684	12.809 ± 0.036	8.8%
2	0.125	6	8	16.627	14.627 ± 0.049	−13.7%
3	0.125	7	8	32.104	29.979 ± 0.123	−7.1%
4	0.125	7	8	32.482	29.940 ± 0.134	−8.5%
5	0.125	7	8	32.994	30.005 ± 0.148	−10.0%
6	0.125	7	8	33.718	29.938 ± 0.095	−12.6%
7	0.125	7	8	33.336	30.016 ± 0.122	−11.1%

Table 4.15: 7 Queues Polling System, $\bar{b} = 1.25$, $\rho = 1.0625$

Poll-Order	1 2 3 1 4 5 2 1 6 7					
Queue	λ	T_{max}	K	W_{App}	W_{Sim}	% Error
1	0.2	5	8	18.263	20.760 ± 0.080	12.0%
2	0.15	6	8	21.610	24.011 ± 0.076	10.0%
3	0.1	7	8	46.568	42.055 ± 0.145	−10.7%
4	0.1	7	8	46.633	42.092 ± 0.158	−10.8%
5	0.1	7	8	46.722	42.048 ± 0.114	−11.1%
6	0.1	7	8	46.851	42.151 ± 0.060	−11.2%
7	0.1	7	8	46.731	42.175 ± 0.118	−10.8%

CHAPTER 5

STATE SPACE REDUCTION OF MAP WITH SPECIAL STRUCTURE

5.1 Introduction

In Chapters 3 and 4 we developed the vacation period for a discrete-time polling system with exhaustive time-limited service discipline. The vacation period is a MAP with special structure and looks like a convolution of discrete phase type distributions. Through the numerical examples, we found that the execution time of the code increases as a function of the dimension of the vacation period. In this Chapter, we focus on reducing the dimension of each phase type distribution which will result in a MAP of smaller dimension. This is achieved by using the moment matching approach. Specifically, the first three moments of an n-dimensional discrete phase type distribution are matched to the corresponding moments of a 2-dimensional discrete phase type distribution. Therefore, if the initial MAP (vacation period distribution in Chapter 4) looks like the convolution of l discrete phase type distribution each of dimension n_i; $i = 1, \ldots, l$; then its dimension is $\sum_{i=1}^{l} n_i$. However, once we reduce the dimension of each phase type distribution, using the moment matching approach, the resulting vacation period distribution has a dimension of $2l$ which is significantly smaller. For example, in Section 4.1 queue A has two phase type vacations with dimension 8 and 13. The corresponding MAP has a dimension equal to 21. Once the phase vacations are reduced to 2×2, the corresponding MAP will have a dimension equal to 4. This is significantly smaller. Note here that the significance of this

reduction manifests itself in solving for the rate matrix, R, (Eq. 3.2). Without this reduction, the R of queue A in Section 4.1, is a 25×25 matrix. Whereas after this reduction R is an 8×8 matrix. This reduction will bring about a significant reduction in the computational effort required to obtain performance measures for each queue in a polling system. This Chapter is organized as follows. First, we briefly review some literature related to fitting distribution. Second, we show how to adapt Altiok's method [5] to discrete-time phase type distributions. In the last Section, we discuss some numerical examples.

5.2 Brief Literature Review

In this Section, we review some of the work done in the area of fitting distributions. Four methods for fitting distributions are available: 1) maximum likelihood estimators (MLEs), 2) moment matching (MM), 3) least square estimator, and 4) unbiased estimator. In this review, we focus on the MLEs and MM methods. Particularly, we focus on MM method for two reasons: 1) finding MLEs is not always easy [108, page 370], and 2) in queueing theory, especially with applications in the engineering field, usually the first few moments are sufficient since they provide a good insight into the behavior of the system (Neuts [133, page 42]). In either case the goodness of fit is measured in terms of the error between the actual and the fitted distribution.

Johnson and Taaffe [87] showed that it is possible to match the first k ($k < \infty$) moments of a non-degenerate distribution with support on $[0, \infty)$ with the moments of a mixture of Erlang distributions of common order. Later, in [88], they used the non-linear programming approach to approximate the moments of phase type distribution. Earlier, Altiok [5] approximated a general distribution with known coefficient of variation by a 2-dimensional phase type distribution using the first three moments. On the other hand, Asmussen and Nerman [10] used the maximum likelihood approach to fit phase type distribution. However, their approach may run

into some problems when the number of phases becomes very large. In a recent paper, Asmussen [9] and, independently, Lang and Arthur [107] provided comprehensive reviews of the state of fitting phase type distributions and a comparison between the moment matching approach and the maximum likelihood estimators approaches.

With regard to MAP fitting, only special classes are addressed in the literature. In particular, the Markov modulated Poisson process (MMPP) with 2 states was used to model traffic in integrated services network. Hellstern [130] used a numerical approach based on the maximum likelihood to fit an MMPP with two arrival rates. And Heffes and Lucantoni [79] approximated the superposition of data and voice packets using an MMPP. In other applications, Keogh presented an approach to fit the output of video coders using a birth-death process in [94] and [93] and using a discrete-space continuous-time Markov process in [95]. More recently, Ni *et al.* [138] used a discrete-time Markov modulated deterministic process (MMDP) to model the traffic for an MPEG-2 movie video traffic. The different methods for fitting MMPP are summarized in a survey by Ryden [146]. Elsayed and Perros [55] presented an approach to approximately characterize the superposition of N, $N \geq 2$, arbitrary discrete-time Markov renewal process.

Recently, Diamond and Alfa [48] showed that the autocorrelation sequence of inter-arrival times for MAP of order two is geometric. Based on the value of the autocorrelation and the value of the coefficient of variation, they discussed different fitting approach for 2×2 MAPs. It is also shown in their paper that it is quite difficult to fit general MAPs. This is the only paper we are aware off that deals with MAP fitting.

Due to this difficulty, we limit ourselves here to a special class of MAPs. We focus on MAPs that look like the convolution of discrete-time phase type distributions. Notice that because of the lack of a better term, in the rest of this Chapter, we use the word convolution to imply looks like convolution or MAP obtained by assembling discrete-time phase type distributions.

5.3 Reduction Technique

As stated earlier, our concern here is the state space reduction for MAPs with special structure. Since the MAP we are interested in is obtained by assembling l discrete time phase type distributions, and there are well known results for fitting phase type distribution, our task is quite simple. First, we reduce the dimension of each discrete phase type distribution to a 2-dimensional discrete phase type distribution. Then, we assemble these distributions. One of the straight forward fitting approach is by Altiok [5] using the moments matching approach. Since he dealt with continuous time phase type distribution, in Section 5.3.1, we adopt his method for the discrete time phase type distributions. Section 5.3.2 shows the original and resulting MAP. It also gives the measures which we adopt for testing the performance of the reduction technique.

5.3.1 Phase Distribution Reduction

Let $(\boldsymbol{\beta}, S)$ be the representation of a discrete time phase type distribution of dimension n. Given the first three factorial moments of $(\boldsymbol{\beta}, S)$; m_1, m_2, and m_3; we seek a discrete phase type distribution of dimension 2 and representation $(\boldsymbol{\alpha}, T)$ such that the first three factorial moments of $(\boldsymbol{\beta}, S)$ and $(\boldsymbol{\alpha}, T)$ are identical. The factorial moments for a discrete phase type distribution with representation $(\boldsymbol{\beta}, S)$ are given in Neuts [133, Chap. 2] as $m_k = P^{(k)}(1) = k!\boldsymbol{\beta}S^{k-1}(I-S)^{-k}\mathbf{e}$. Thus, given m_1, m_2, and m_3 we need to obtain a 2-dimensional distribution with representation $(\boldsymbol{\alpha}, T)$ such that

$$\begin{cases} m_1 &= \boldsymbol{\alpha}(I-T)^{-1}\mathbf{e} \\ m_2 &= 2\boldsymbol{\alpha}T(I-T)^{-2}\mathbf{e} \\ m_3 &= 6\boldsymbol{\alpha}T^2(I-T)^{-3}\mathbf{e}. \end{cases}$$

Notice that, in general, a 2-dimensional discrete phase type distribution is defined

in terms of six variables and is given by:

$$\boldsymbol{\alpha} = [a+b \;\; 1-a-b], \;\; T = \begin{bmatrix} 1-\mu_1-\mu_3 & \mu_1 \\ \mu_4 & 1-\mu_2-\mu_4 \end{bmatrix}, \;\; \mathbf{T}^\circ = \begin{bmatrix} \mu_3 \\ \mu_2 \end{bmatrix},$$

where $0 \leq a+b \leq 1$, $0 < \mu_1+\mu_3 < 1$, $0 < \mu_2+\mu_4 < 1$, and $0 \leq \mu_1,\ \mu_2,\ \mu_3,\ \mu_4,\ a,\ b \leq 1$. However, the relationship between the variables ($\mu_1,\ \mu_2,\ \mu_3,\ \mu_4,\ a,\ b$) is nonlinear and an attempt to include higher moments will lead to a set of equations that is very difficult if not impossible to solve. Therefore, we limit the set of feasible phase type distributions to those that look like the generalized negative binomial. This is achieved by setting $\mu_3 = \mu_4 = b = 0$. The numerical examples will show that this is a good simplification. Thus, $(\boldsymbol{\alpha}, T)$ is given by : $\boldsymbol{\alpha} = [a \; 1-a]$, $T = \begin{bmatrix} 1-\mu_1 & \mu_1 \\ 0 & 1-\mu_2 \end{bmatrix}$, and the absorption vector $\mathbf{T}^\circ = \begin{bmatrix} 0 \\ \mu_2 \end{bmatrix}$, where $0 \leq a \leq 1$, $0 < \mu_1 < 1$ and $0 < \mu_2 < 1$. Therefore, we have to solve the following 3 nonlinear equations for the unknowns a, μ_1, and μ_2.

$$\begin{cases} m_1 &= \frac{a\mu_2+\mu_1}{\mu_1\mu_2} \\ m_2 &= 2a(1-\mu_1)\left(\frac{1}{\mu_1^2}+\frac{1}{\mu_1\mu_2}+\frac{1}{\mu_2^2}\right)+2\frac{1-a-\mu_2+a(\mu_1+\mu_2)}{\mu_2^2} \\ m_3 &= 6a(1-\mu_1)^2\left(\frac{1}{\mu_1^3}+\frac{1}{\mu_1^2\mu_2}+\frac{1}{\mu_1\mu_2^2}+\frac{1}{\mu_2^3}\right)+6\frac{a\mu_1(2-\mu_1-\mu_2)+(1-a)(1-\mu_2)^2}{\mu_2^3}. \end{cases}$$

Let D be the root of the polynomial:

$$K_1 Z^2 + K_2 Z + K_3 = 0, \tag{5.1}$$

where

$$\begin{cases} K_1 &= 2m_1m_3 - 3m_2^2 \\ K_2 &= 12m_1^2 + 6m_1m_2 - 12m_1 - 12m_2 - 2m_3 \\ K_3 &= -12m_1^2 + 12m_1 + 6m_2. \end{cases}$$

Then a, μ_1, and μ_2 are given, respectively, by:

$$\begin{cases} a &= \frac{-2(B+C))}{D(2m_3m_1-3m_2^2)A} \\ \mu_1 &= \frac{m_1D-1}{A} \\ \mu_2 &= D, \end{cases}$$

where

$$\begin{cases} A &= m_2D-2m_1+2m_1D \\ B &= D(2m_1^2m_3-6m_2^2m_1+12m_1^4+6m_1^3m_2-12m_1^3-12m_1^2m_2) \\ C &= -2m_1m_3+3m_2^2-12m_1^4+12m_1^3+6m_1m_2). \end{cases}$$

It is worth mentioning here that there are two roots for equation 5.1. Using either root results in matching the first three moments. Since the computational time is very small, one can compute both probability mass functions and use the one with the smaller error. Now that we have the reduced phase type distribution, next we address how to study its effect on the assembled MAP.

5.3.2 MAP Reduction

Suppose we have a MAP with representation (V_0, V_1) which is the convolution of l phase type distribution with representation $(\boldsymbol{\beta}_i, S_i)$ each with dimension n_i, $i = 1, \ldots, l$. For each $(\boldsymbol{\beta}_i, S_i)$ there exists a 2-dimensional phase type distribution with representation $(\boldsymbol{\alpha}_i, T_i)$. Therefore, if the original MAP, (V_0, V_1) is given by:

$$V_0 = \begin{bmatrix} S_1 & 0 & 0 & \mathbf{0} & 0 \\ 0 & S_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & S_{l-1} & 0 \\ 0 & 0 & 0 & 0 & S_l \end{bmatrix}, V_1 = \begin{bmatrix} 0 & \mathbf{S}_1^{\circ}\boldsymbol{\beta}_2 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{S}_2^{\circ}\boldsymbol{\beta}_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{S}_{l-1}^{\circ}\boldsymbol{\beta}_l \\ \mathbf{S}_l^{\circ}\boldsymbol{\beta}_1 & 0 & 0 & 0 & 0 \end{bmatrix};$$

then the reduced MAP, (V_0', V_1'), is given by:

$$V_0' = \begin{bmatrix} T_1 & 0 & 0 & 0 & 0 \\ 0 & T_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & T_{l-1} & 0 \\ 0 & 0 & 0 & 0 & T_l \end{bmatrix}, V_1' = \begin{bmatrix} 0 & \mathbf{T}_1^{\circ}\boldsymbol{\alpha}_2 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{T}_2^{\circ}\boldsymbol{\alpha}_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{T}_{l-1}^{\circ}\boldsymbol{\alpha}_l \\ \mathbf{T}_l^{\circ}\boldsymbol{\alpha}_1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The effect of reducing the dimension of MAP is considered for two time origins: 1) at an arbitrary point and 2) at an event starting point. This is achieved by letting $\boldsymbol{\pi}$ be the steady state probability vector and $\boldsymbol{\pi}_{ev}$ be the probability vector at an event (see Neuts [135]) of (V_0, V_1), and $\boldsymbol{\pi}'$ be the steady state probability vector and $\boldsymbol{\pi}'_{ev}$ be the probability vector at an event of (V_0', V_1'). $\boldsymbol{\pi}$, $\boldsymbol{\pi}_{ev}$, $\boldsymbol{\pi}'$, and $\boldsymbol{\pi}'_{ev}$ are given, respectively, by

$$\boldsymbol{\pi} = \boldsymbol{\pi}(V_0 + V_1) \text{ and } \boldsymbol{\pi}\mathbf{e} = 1 \tag{5.2}$$

$$\boldsymbol{\pi}_{ev} = \frac{1}{\lambda}\boldsymbol{\pi}V_1 \tag{5.3}$$

$$\boldsymbol{\pi}' = \boldsymbol{\pi}'(V_0' + V_1') \text{ and } \boldsymbol{\pi}'\mathbf{e} = 1 \tag{5.4}$$

$$\boldsymbol{\pi}'_{ev} = \frac{1}{\lambda'}\boldsymbol{\pi}'V_1', \tag{5.5}$$

where λ and λ' are the fundamental rates of (V_0, V_1) and (V_0', V_1') and given, respectively, by:

$$\lambda = \boldsymbol{\pi}V_1\mathbf{e} \tag{5.6}$$

$$\lambda' = \boldsymbol{\pi}'V_1'\mathbf{e}. \tag{5.7}$$

Notice that, although equations 5.2 (5.3) and 5.4 (5.5) are identical, the vector $\boldsymbol{\pi}'$ $(\boldsymbol{\pi}'_{ev})$ has a significantly smaller dimension compared to $\boldsymbol{\pi}$ $(\boldsymbol{\pi}_{ev})$, $2l$ versus $\sum_{i=1}^{l} n_i$. Therefore, the probability distribution (steady state) for the original and reduced MAP are given, respectively, by:

$$P(k) = \boldsymbol{\pi}V_0^{k-1}V_1\mathbf{e} \tag{5.8}$$

$$P'(k) = \boldsymbol{\pi}' V_0'^{k-1} V_1' \mathbf{e}, \quad k \geq 1. \tag{5.9}$$

And the probability distribution (given that an event occurred) for the original and reduced MAP are given, respectively, by:

$$Q(k) = \boldsymbol{\pi}_{ev} V_0^{k-1} V_1 \mathbf{e} \tag{5.10}$$

$$Q'(k) = \boldsymbol{\pi}'_{ev} V_0'^{k-1} V_1' \mathbf{e}, \quad k \geq 1. \tag{5.11}$$

Equations 5.8-5.11 allow us to compare the probability mass functions. Another important characteristic of MAP is that it can capture correlation. The correlation of a MAP is given in [19], for our case, the coefficient of correlation for the original and the reduced MAP are given, respectively, by:

$$c_{coor} = \frac{\boldsymbol{\pi} V_1^2 \mathbf{e} - \lambda^2}{\boldsymbol{\pi} V_1 - \lambda^2} \tag{5.12}$$

$$c'_{coor} = \frac{\boldsymbol{\pi}' V_1'^2 \mathbf{e} - \lambda^2}{\boldsymbol{\pi}' V_1' - \lambda'^2} \tag{5.13}$$

In the next Section, some examples are presented. The goodness of fit is measured in terms of the following errors:

- $Error_1 = \sqrt{\sum_k (P(k) - P(k)')^2}$.
- $Error_2 = \sqrt{\sum_k (Q(k) - Q(k)')^2}$.
- $Error_3 = |c_{coo} - c'_{coo}|$.

$Error_1$ ($Error_2$) measures the difference between the probability mass function of the original and the reduced MAP under steady state (given an event occurred). $Error_3$ measures the difference between the coefficient of correlation between the original and the reduced MAP.

5.4 Numerical Examples

In this Section, we discuss three examples. In each example, we give the original phase type distribution followed by its reduced form. The original and the reduced

MAP have the same form as given in Section 5.3.2. The results presented here compare the performance of the fitting algorithm if one consistently chooses one root, for example, the smaller of the two roots.

5.4.1 Example 1

In this example we use two phase type distributions to obtain a MAP. The first phase type distributions is given by:

$$S_1 = \begin{bmatrix} 0.3 & 0.3 & 0.4 & 0.0 \\ 0.0 & 0.3 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.6 \end{bmatrix} \mathbf{S}_1^\circ = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.3 \\ 0.4 \end{bmatrix} \boldsymbol{\beta}_1 = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \end{bmatrix},$$

and its reduced form is given by:

$$T_1 = \begin{bmatrix} 0.5700 & 0.4300 \\ 0.0000 & 0.6056 \end{bmatrix} \mathbf{T}_1^\circ = \begin{bmatrix} 0.0000 \\ 0.3944 \end{bmatrix} \boldsymbol{\alpha}_1 = \begin{bmatrix} 0.6593 & 0.3407 \end{bmatrix}.$$

The second phase type distribution is given by:

$$S_2 = \begin{bmatrix} 0.2 & 0.4 & 0.4 & 0.0 \\ 0.0 & 0.2 & 0.7 & 0.1 \\ 0.0 & 0.0 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.4 & 0.2 \end{bmatrix} \mathbf{S}_2^\circ = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.3 \\ 0.4 \end{bmatrix} \boldsymbol{\beta}_2 = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \end{bmatrix},$$

and its reduced form is given by:

$$T_2 = \begin{bmatrix} 0.4065 & 0.5935 \\ 0.0000 & 0.6796 \end{bmatrix} \mathbf{T}_2^\circ = \begin{bmatrix} 0.0000 \\ 0.3204 \end{bmatrix} \boldsymbol{\alpha}_2 = \begin{bmatrix} 0.5961 & 0.4039 \end{bmatrix}.$$

Therefore, the original MAP has a dimension of 8. The reduced MAP has a dimension of 4. The probability mass function of the original and reduced MAP are shown in

Figure 5.1. The coefficient of correlation of the original MAP is -0.1377 and that of the reduced MAP is -0.1484. The summary of the three errors is given in Table 5.1.

Table 5.1: Example 1-Errors

$Error_1$	$Error_2$	$Error_3$
0.0028	0.0170	0.0107

5.4.2 Example 2

In the second example, we use three phase type distributions. The first and second phase type distribution are the same as those given in Example 1. The third phase type distribution is given by:

$$S_3 = \begin{bmatrix} 0.1 & 0.5 & 0.4 & 0.0 \\ 0.0 & 0.1 & 0.8 & 0.1 \\ 0.0 & 0.0 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.6 \end{bmatrix}, \quad \mathbf{S}_3^0 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.3 \\ 0.4 \end{bmatrix}, \quad \boldsymbol{\beta}_3 = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \end{bmatrix},$$

and its reduced form is given by:

$$T_3 = \begin{bmatrix} 0.4503 & 0.5497 \\ 0.0000 & 0.6333 \end{bmatrix}, \quad \mathbf{T}_3^0 = \begin{bmatrix} 0.0000 \\ 0.3667 \end{bmatrix}, \quad \boldsymbol{\alpha}_3 = \begin{bmatrix} 0.6360 & 0.3640 \end{bmatrix}$$

Thus, the original MAP has a dimension of 12 and the reduced MAP has a dimension of 6. The probability mass function of the original and reduced MAP are given in Figure 5.2. The coefficient of correlation of the original MAP is -0.1442 and that of the reduced MAP is -0.1543. The summary of the three errors is given in Table 5.2.

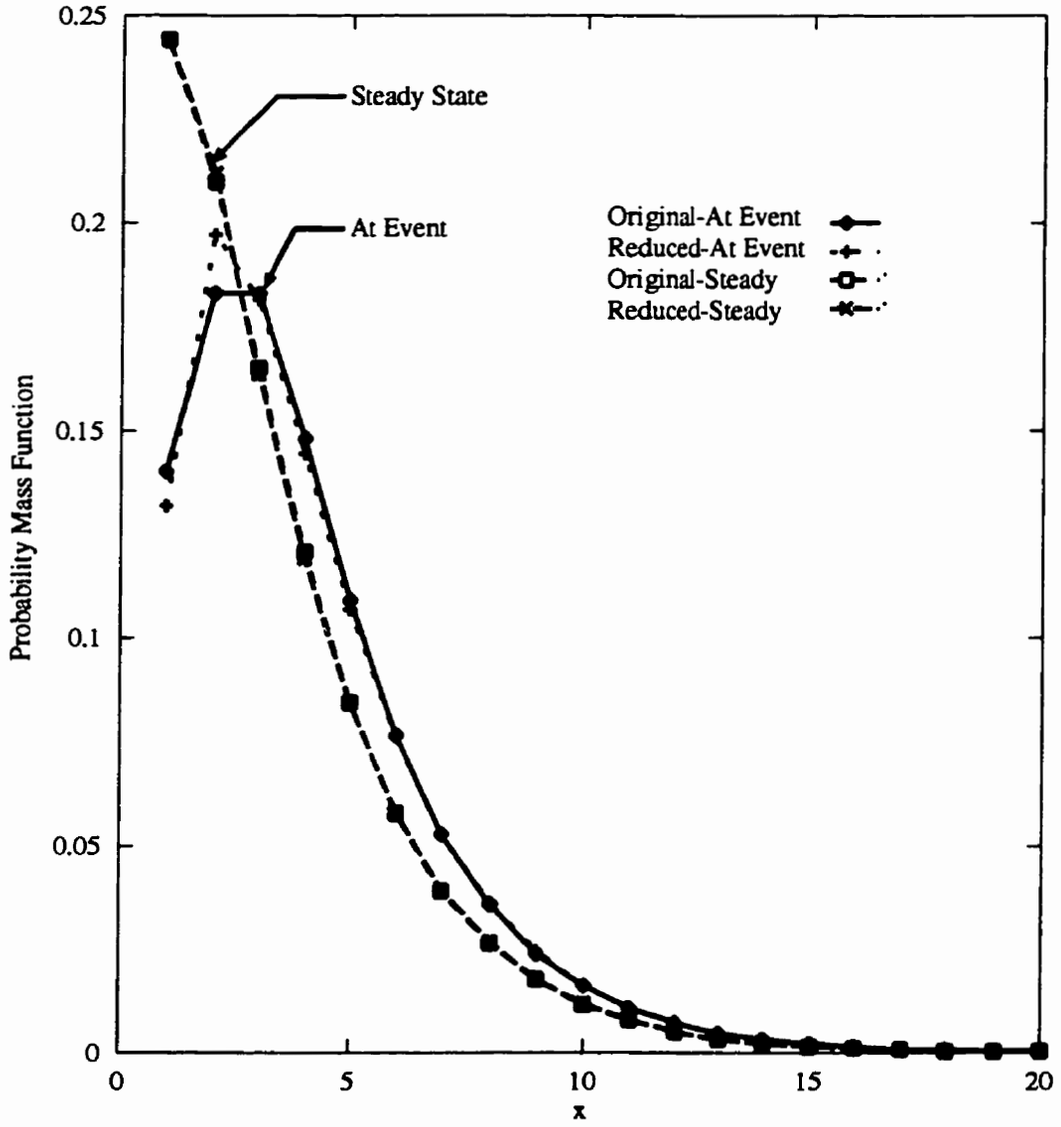

Figure 5.1: MAP with 2 Phase Distributions, Original Dim.=8, Reduced Dim.=4

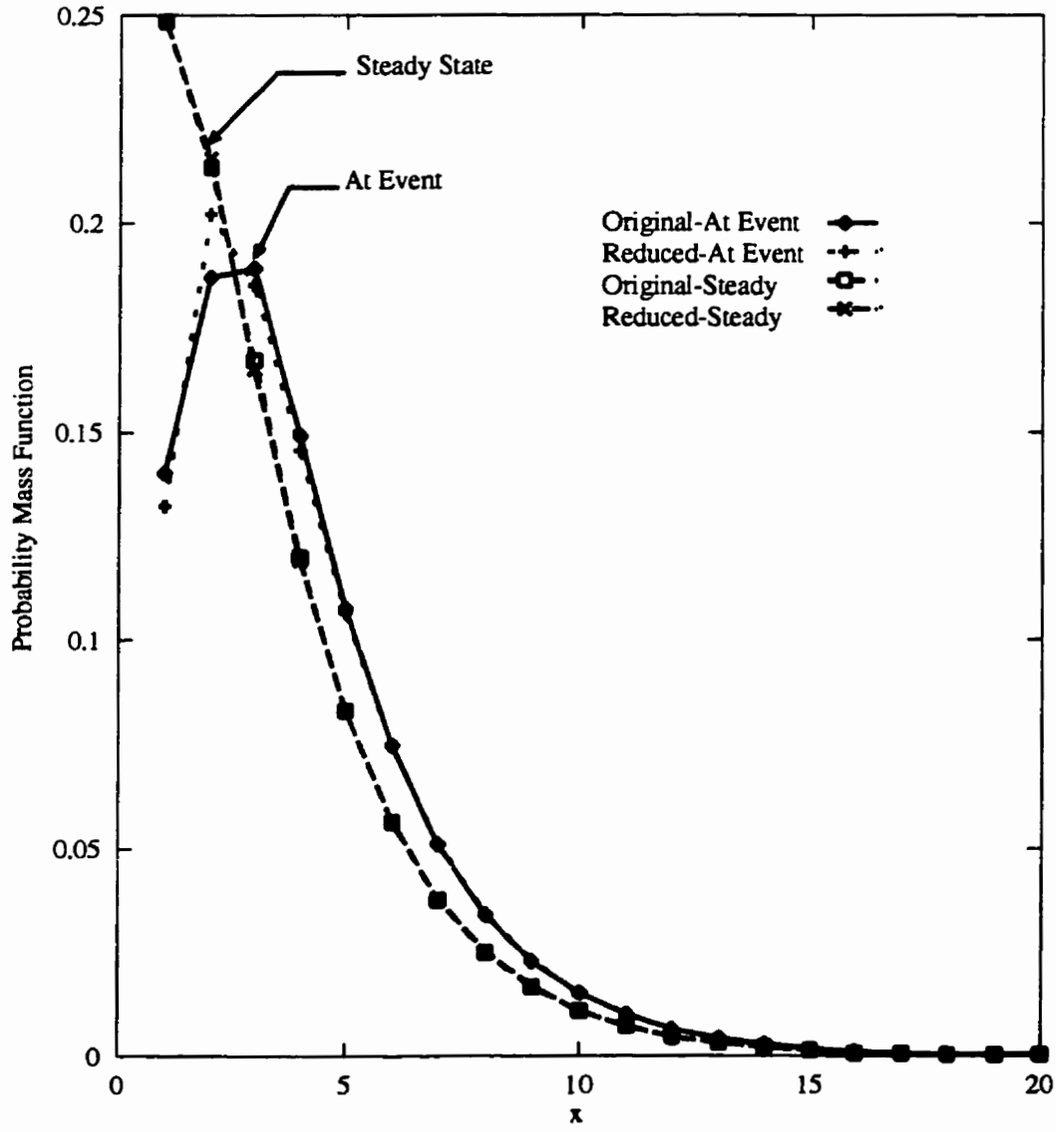

Figure 5.2: MAP with 3 Phase Distributions, Original Dim.=12, Reduced Dim.=6

Table 5.2: Example 2-Errors

$Error_1$	$Error_2$	$Error_3$
0.0028	0.0177	0.0101

5.4.3 Example 3

In this example we use three phase type distributions. The first is the same as the one given in Example 1. The second is given by:

$$S_2 = \begin{bmatrix} 0.2 & 0.4 & 0.4 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.7 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.6 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.4 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.3 & 0.4 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.4 \end{bmatrix}, \quad \mathbf{S}_2^\circ = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.3 \\ 0.4 \\ 0.2 \\ 0.4 \end{bmatrix},$$

$$\boldsymbol{\beta}_2 = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 & 0.0 & 0.0 \end{bmatrix},$$

and its reduced form is given by:

$$T_2 = \begin{bmatrix} 0.4065 & 0.5935 \\ 0.0000 & 0.6796 \end{bmatrix}, \quad \mathbf{T}_2^\circ = \begin{bmatrix} 0.0000 \\ 0.3204 \end{bmatrix}, \quad \boldsymbol{\alpha}_2 = \begin{bmatrix} 0.5961 & 0.4039 \end{bmatrix}.$$

The third phase type distribution is given by:

$$S_3 = \begin{bmatrix} 0.1 & 0.5 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.1 & 0.8 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.6 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.5 & 0.1 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 0.4 \end{bmatrix}, \quad \mathbf{S}_3^\circ = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.2 \\ 0.3 \\ 0.2 \end{bmatrix}.$$

$$\boldsymbol{\beta}_3 = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix},$$

and its reduced form is given by:

$$T_3 = \begin{bmatrix} 0.2221 & 0.7779 \\ 0.0000 & 0.6738 \end{bmatrix}, \quad \mathbf{T}_3^\circ = \begin{bmatrix} 0.0000 \\ 0.3262 \end{bmatrix}, \quad \boldsymbol{\alpha}_3 = \begin{bmatrix} 0.6557 & 0.3443 \end{bmatrix}.$$

Thus, the original MAP has a dimension of 18 and the reduced MAP has a dimension of 6. The probability mass function of the original and reduced MAP are given in Figure 5.3. The coefficient of correlation of the original MAP is -0.1434 and that of the reduced MAP is -0.1629. The summary of the three errors is given in Table 5.3.

Table 5.3: Example 3-Errors

$Error_1$	$Error_2$	$Error_3$
0.0049	0.0290	0.0195

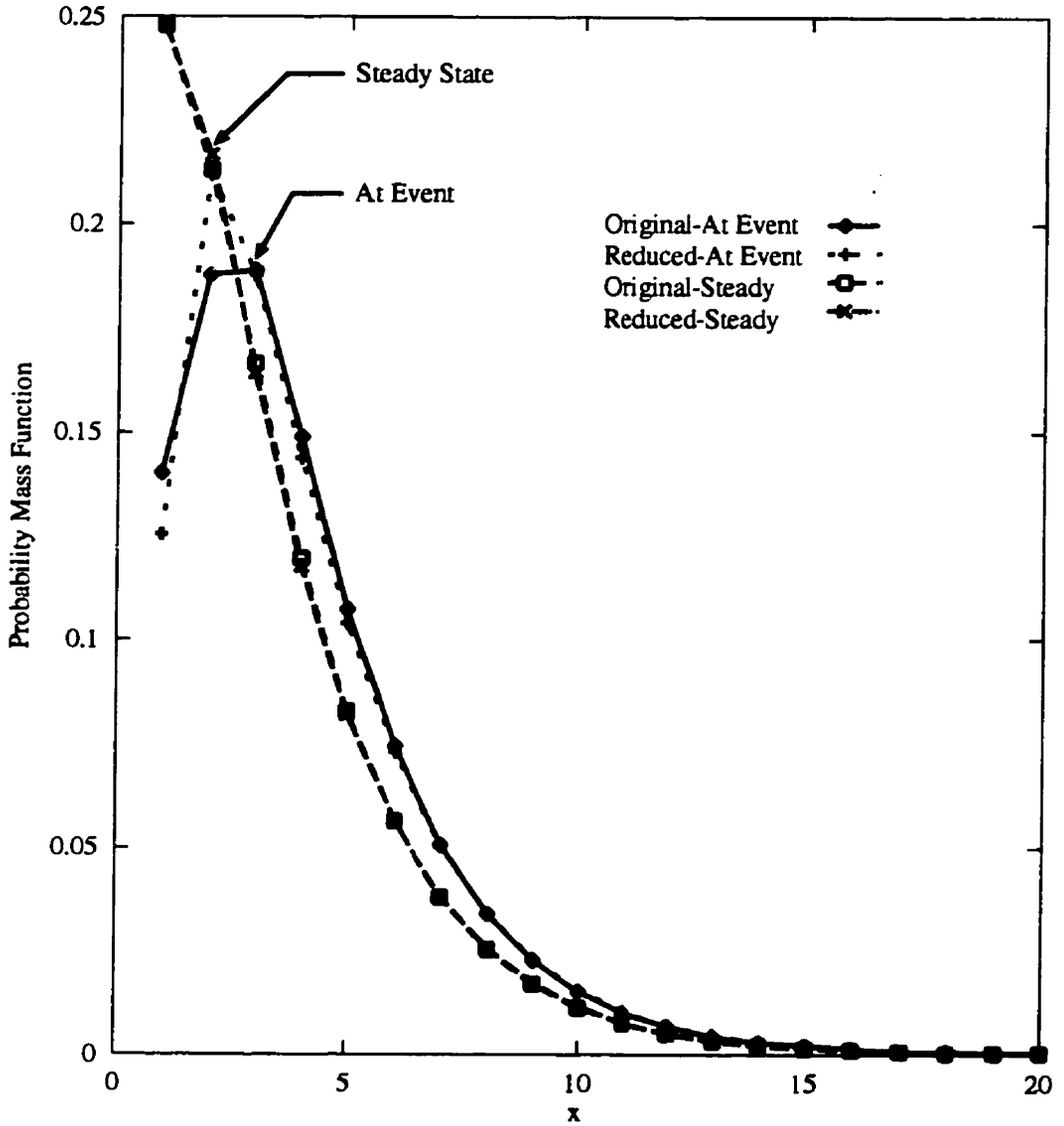

Figure 5.3: MAP with 3 Phase Distributions, Original Dim.=18, Reduced Dim.=6

5.4.4 Discussion

For all three examples, we notice that in the steady state the probability mass function of the reduced and original MAP are almost identical. However, the probability mass function given that an event happens exhibit a large difference especially for $x = 1, 2, 3$. However, for $x \geq 4$ the original and reduced MAP are almost identical. Notice also that for all three examples, the reduced MAP has a higher coefficient of correlation than the original MAP. The difference is about 10%.

Lastly, we should mention that since this procedure is quite simple and fast, instead of using the smaller of the two roots of equation Eq. 5.1, it is better to compute two distributions (one corresponding to the smaller root and the second to the larger root). Then, choose the distribution that yields the least error.

In conclusion, in this Chapter we showed how to adopt Altiok [5] method for discrete-time phase type distributions. The results are then used to reduce the dimension of MAPs that look like a convolution of phase type distributions. The method is quite simple and easy to implement.

CHAPTER 6

SUMMARY, CONCLUSIONS, & FUTURE WORK

6.1 Summary

The objectives of this thesis were two. First, to develop an iterative procedure to compute the mean waiting time for a discrete time cyclic/table polling system where all the queues have either infinite or finite buffer capacity. In this polling system, customers arrive according to the Markovian arrival process and their service time can be represented by a phase type distribution. In addition, each queue is visited according to the exhaustive time-limited service discipline. The iterative procedure was then tested using different network configurations. This objective was accomplished by the following steps:

1) For cyclic polling systems, each queue in the polling system was modeled as a single server queue with exhaustive time-limited service discipline and vacation periods. For the infinite buffer capacity case, we used the matrix analytic approach to compute performance measures for each queue. The rate matrix R is obtained using the algorithm given in Alfa [2]. For the finite buffer capacity case, we used the block Gauss-Seidel iterative method procedure to obtain the queue length distribution. Each block has dimension mn, where m is the dimension of the service time distribution and n is the dimension of the arrival process.

The single server queue with vacation period models were then incorporated into an iterative procedure to obtain the queue length distribution and the mean waiting time for each queue in the polling system. Due to the time-limited service discipline, we were able to represent the visit and vacation periods, for each queue, by phase type distributions. Notice that in this iterative procedure, the vacation period distribution of a queue is given by the visit period of all the queues visited while the server is away. In addition, the correlation between the visit period and the vacation period was captured using an approach similar to that of Lee and Sengupta [109].

2) For table polling systems, we extended the results of the single server queue with exhaustive time-limited service discipline and phase type vacation periods to include MAP type vacation periods. The use of MAP is justified by the correlation between the position of the server visit to a queue in the table and the type of vacation the server will take.

   The stopping criteria for the iterative procedure in the case of cyclic or table polling is the smaller of the difference between the mean waiting time and the mean vacation period in two subsequent iterations, for instance, in the examples ran in Chapter 3 and 4 the tolerance was set to $\epsilon = 10^{-8}$.

3) The results obtained by the iterative procedure were then compared to those obtained by simulation. The effect of the system utilization, the number of queues, the time slot threshold for each queue, and in the case of table polling the sequence of queue visits on the performance of the iterative procedure were studied. For the finite buffer capacity model, the effect of over load, and the buffer capacity were also studied.

The second objective was to reduce the dimension of MAPs with special structures. The MAP we were concerned with is obtained by assembling discrete phase type distributions and it represented the vacation period distribution for a queue visit

depends on the number of queues, Q, and the allocated time, T, for each queue. This is because when the time thresholds, T, of the queues visited while the server is away are large, the dimension of the phase distribution of the vacation period becomes large, thus, a large rate matrix to solve for. Similarly, when Q is large we have to solve for Q rate matrices and queue lengths (assuming that the polling system is not symmetric). In addition when M, the number of visits for a queue in the table, is large the dimension of the vacation period becomes large which increases the dimension of the rate matrix and the probability vector.

For the finite capacity model, the iterative algorithm converges in less than 30 iterations with a run time less than 30 minutes on an IBM RS6000/590. Most of the CPU time during each run is used up in solving for the queue length distribution. In the examples ran, the number of iterations required for convergence depends on the number of queues, Q, the buffer size of each queue, K, and the allocated time for each queue, T. This is because when T and/or K is large, the dimension of the probability vector becomes large. Similarly, when Q is large we have to solve for Q queue lengths (assuming that the polling system is not symmetric). In addition, when M, the number of visits for a queue in the table, is large the dimension of the vacation period becomes large which increases the dimension the probability vector.

Both iterative models (infinite and finite) yielded an error in the mean waiting time of about 20%. In addition, the results were reasonable for the mean waiting time under different load and time allocation. The proposed iterative procedure can be used to solve both symmetric and asymmetric systems in terms of load and time allocation.

6.2.2 MAP Reduction

The study of the effect of reducing the dimension of MAP yielded the following conclusions:

- The difference between the original probability mass function and the reduced probability mass function is not noticeable for the steady state.
- The difference between the original probability mass function and the reduced probability mass function given that an event has occured is worst when $x \leq 4$. For $x > 4$ the two probability mass functions are almost identical. Thus, the fitting comparison should be based on the probability mass function given an that event has occured rather than the probability mass function at steady state.
- The reduced MAP has a slightly higher coefficient of correlation for the examples shown in Chapter 5.

6.2.3 Limitations of this study

For the discrete time polling system with time-limited service discipline, this study is based on an iterative approach and it has the following limitations:

- The model is for discrete time, and its extension to continuous time is not easy since we loose the advantage of representing the visit and vacation period by phase type distributions.
- The convergence criteria for the iterative procedure is set to be the smaller of the difference between the mean waiting times and the difference between the mean vacation periods for two consecutive iterations.
- Each queue has only one time threshold.
- Only a single arrival process is considered.
- The service discipline is the same for all the queues. However, each queue may have its own time threshold.
- The results are intended for the steady state region.

For the state space reduction, we considered only MAPs with special structures. The results are good only for MAPs that look like a "convolution" of discrete phase distributions. Their extension to general MAPs seems to be quite difficult as discussed in Diamond and Alfa [48].

6.3 Recommendations for Further Research

Because of the limitations stated in Section 6.2.3, future work should attempt to:

- Extend the analysis to include multiple time thresholds for queue's visited more than once in the case of table polling. Although, the single server queue with vacation period model used in Chapter 4 assumes that the time threshold for every queue visit is the same in the case of table polling, it is possible to extend the model to the case where a queue may have different time thresholds for each server visit.

- Extend the analysis to allow batch arrival process B-MAP. This would give us the flexibility to model a system where customers arrive in batches (packets), however, the server (switch) can serve (transmit) only one customer (cell) at a time. The task of obtaining the waiting time distribution is a challenging one. For a related model see Frigui, Alfa and Xu [64].

- Allow each queue to have its own service discipline. However, the flexibility to set a time threshold for every queue may offset any need for this task.

- Extend this analysis to consider convergence based on the whole waiting time distribution.

- For cyclic polling systems, we proved that the iterative procedure converges. However, for table polling we did not prove that the algorithm converges.

- The extension of the independent part of the vacation period to include switch-over time was presented in Section 3.7. However, the extension of the dependent part of the vacation period was not presented. That remains to be done.

In addition to the above directions for future research, the general question of interest is the optimization of the system. As stated in Section 2.11, given a set of queues, a single server, an arrival process, and a set of service disciplines. What is the best polling order policy to optimize a performance measure, say, the weighted sum of the mean waiting times.

REFERENCES

[1] ANSI/IEEE Standard 802.4, *Token-Passing Bus Method.* IEEE Press, New York, 1986.

[2] A.S. Alfa, " A discrete single-server MAP/PH/1 queue with vacations and exhaustive time-limited service," *Oper. Res. Letters*, vol. 18 pp. 31–40, 1995.

[3] A.S. Alfa, "modeling traffic queues at a signalised intersection with vehicle-actuated control and Markovian arrival processes," *Computers Math Applic.*, vol. 30 pp. 105–119, 1995.

[4] A.S. Alfa and M.F. Neuts, "Modeling vehicle traffic using the discrete time Markovian arrival process," *Transportation Science*, vol. 29 pp. 109–117, 1995.

[5] T. Altiok, "On the phase-type approximations of general distributions," *IIE Trans.*, vol. 17 pp. 110–116, 1985.

[6] E. Altman, A. Khamisy, and U. Yechiali, "On elevator polling with globally gated regime," *QUESTA*, vol. 11 pp. 85–90, 1992.

[7] E. Altman and D. Kofman, "Bounds for performance measures of token rings," *IEEE/ACM Trans. on Networking*, vol. 4 pp. 292–299, 1996.

[8] E. Altman, P. Konstantopoulos, and Z. Liu, "Stability, monotonicity and invariant quantities in general polling systems," *QUESTA*, vol. 11 pp. 35–57, 1992.

[9] S. Asmussen, "Phase-type distributions and related point processes: Fitting and recent advances," In S. Chakravarthy and A.S. Alfa, editors, *First International Conference on Matrix-Analytic Methods in Stochastic Models*, pp. 137–149. Marcel Dekker Inc., 1996.

[10] S. Asmussen and O. Nerman, "Fitting phase-type distribution via EM algorithm," In K. Vest Nilsen, editor, *Anvendt Statistik*, pp. 335–346, Copenhagen, 1991.

[11] B. Avi-Itzhak, W.L. Maxwell, and L.W. Miller, "Queueing with alternating priorities," *Oper. Res.*, vol. 13 pp. 306–318, 1965.

[12] J.E. Baker and I. Rubin, "Polling with general-service order," *IEEE Trans. on Commun.*, vol. 35 pp. 283–288, 1987.

[13] J.P.C. Blanc, "A numerical approach to cyclic-service queueing models," *QUESTA*, vol. 6 pp. 173–188, 1990.

[14] J.P.C. Blanc, "The power-series algorithm applied to polling systems," *Commun. Statis-Stoch. Mod.*, vol. 7 pp. 527–545, 1991.

[15] J.P.C. Blanc, "An algorithmic solution of polling models with limited service disciplines," *IEEE Trans. on Commun.*, vol. 40 pp. 1152–1155, 1992.

[16] J.P.C. Blanc, "Performance evaluation of polling systems by means of power-series algorithm," *Ann. Oper. Res.*, vol. 35 pp. 155–186, 1992.

[17] J.P.C. Blanc and R.D. Van der Mei, "Optimization of polling systems with Bernoulli schedules," *Perf. Eval.*, vol. 22 pp. 139–158, 1995.

[18] C. Blondia, "A discrete-time Markovian arrival process," Technical report, RACE Document, PRLB_123_0015_CC_CD, 1989.

[19] C. Blondia, "A discrete-time batch Markovian arrival process as B-ISDN traffic model," *Belgian J. of Oper. Res., Statis. and Comput. Scien.*, vol. 32 pp. 3–23, 1994.

[20] C. Blondia and T. Theimer, "A discrete-time model for ATM traffic," Technical report, RACE Document, PRLB_123_0018_CC_CD / UST_123_0022_CC_CD, October 1989.

[21] S.C. Borst, O.J. Boxma, and H. Levy, "The use of service limits for efficient operation of multistation single-medium communications systems," *IEEE/ACM Trans. on Networking*, vol. 3 pp. 602–612, 1995.

[22] O.J. Boxma, "Workloads and waiting times in single-server systems with multiple customer classes," *QUESTA*, vol. 5 pp. 185–214, 1989.

[23] O.J. Boxma and W.P. Groenendijk, "Pseudo-conservation laws in cyclic-service systems," *J. of Appl. Probab.*, vol. 7 pp. 299–308, 1987.

[24] O.J. Boxma and W.P. Groenendijk, "Waiting times in discrete-time cyclic-service systems," *IEEE Trans. on Commun.*, vol. 36 pp. 164–170, 1988.

[25] O.J. Boxma, W.P. Groenendijk, and J.A. Weststrate, "A pseudoconservation law for service systems with polling table," *IEEE Trans. on Commun.*, vol. 38 pp. 1865–1870, 1990.

[26] O.J. Boxma, H. Levy, and J.A. Weststrate, "Efficient visit frequencies for polling tables: minimization of waiting cost," *QUESTA*, vol. 9 pp. 133–162, 1991.

[27] O.J. Boxma, H. Levy, and J.A. Weststrate, "Efficient visit orders for polling systems," *Perf. Eval.*, vol. 18 pp. 103–123, 1993.

[28] O.J. Boxma and B.W. Meister, "Waiting-time approximations for cyclic-service systems with switchover times," *Perf. Eval.*, vol. 7 pp. 299–308, 1987.

[29] O.J. Boxma and J.A. Weststrate, "Waiting times in polling systems with Markovian server routing," In G. Stiege and J.S. Lie, editors, *Messung, Modellierung und Bewertung von Rechensystemen und Netzen*, pp. 89–104, Berlin, 1989. Springer.

[30] W. Bux, "Local-area subnetwork: A performance comparison," *IEEE Trans. on Commun.*, vol. 29 pp. 1465–1473, 1981.

[31] W. Bux and H. L. Truong, "Mean-delay approximation for cyclic service queueing systems," *Perf. Eval.*, vol. 3 pp. 187–196, 1983.

[32] R.T. Carsten, E.E. Newhall, and M.J.M. Posner, "A simplified analysis of scan times in an asymmetrical newhall loop with exhaustive service," *IEEE Trans. on Commun.*, vol. 25 pp. 951–957, 1977.

[33] C.J. Chang and L.C. Hwang, "New recursive method for the mean waiting time in a polling network with gated general order service," *IEICE Trans. on Commun.*, vol. E77-B pp. 985–990, 1994.

[34] C.S. Chang, "Stability, queue length and delay, of deterministic and stochastic queueing networks," *IEEE Trans. Automat. Contr.*, vol. 39 pp. 913–931, 1994.

[35] K. Chang and D. Sandhu, "Mean waiting time approximations in cyclic-service systems with exhaustive limited service policy," In *INFOCOM-91*, pp. 1168–1177, 1991.

[36] K.C. Chang and D. Sandhu, "Mean waiting time approximations in cyclic-service systems with exhaustive limited service policy," *Perf. Eval.*, vol. 15 pp. 21–40, 1992.

[37] J. Chiarawongse, M.M. Srinivasan, and T.J. Teorey, "The M/G/1 queueing system with vacations and timer-controlled service," *IEEE Trans. on Commun.*, vol. 42 pp. 1846–1855, 1994.

[38] G.L. Choudhury, "Polling with a general service order table: Gated service," *IEEE Trans. on Commun.*, vol. 38 pp. 268–276, 1990.

[39] H. Chung, C.K. Un, and W.Y. Jung, "Performance analysis of Markovian polling systems with single buffers," *Perf. Eval.*, vol. 19 pp. 303–315, 1994.

[40] E.G. Coffman et al., "Two queues with alternating service periods," In P.J. Courtois and G. Latouche, editors, *Performance'87*, pp. 227–239, Amsterdam, North-Holland, 1988.

[41] E.G. Coffman and A. Stoylar, "Continuous polling on graphs," *Prob. Eng. and Info. Sc.*, vol. 7 pp. 209–226, 1993.

[42] R.B. Cooper, "Queues served in cyclic order: Waiting times," *Bell Syst. Tech. J.*, vol. 49 pp. 399–413, 1970.

[43] R.B. Cooper and G. Murray, "Queues served in cyclic order," *Bell Syst. Tech. J.*, vol. 49 pp. 675–689, 1969.

[44] P.J. Courtois, "The M/G/1 finite capacity queue with delays," *IEEE Trans. on Commun.*, vol. 28 pp. 165–172, 1980.

[45] C.F. Daganzo, "Some properties of polling systems," *QUESTA*, vol. 6 pp. 137–154, 1990.

[46] L.F. de Moraes S. Fuhrmann, "Mean delay approximations for polling systems with batch Poisson input," *Perf. Eval.*, vol. 12 pp. 147–156, 1991.

[47] E. de Souza e Silva et al., "Polling systems with server timeouts and their application to token passing networks," *IEEE/ACM Trans. on Networking*, vol. 3 pp. 560–575, 1995.

[48] J. Diamond and A.S. Alfa, "On two dimensional approximations for MAPs," Technical report, Dept. Mechanical and Industrial Eng., Univ. of Manitoba, 1996.

[49] B. T. Doshi, "Queueing systems with vacations - a survey," *QUESTA*, vol. 1 pp. 29–66, 1986.

[50] B.T. Doshi, "A note on stochastic decomposition in a GI/G/1 queue with vacations or set-up times," *J. of Appl. Probab.*, vol. 22 pp. 419–428, 1985.

[51] D.G. Duffy, "On the numerical inversion of Laplace transforms: Comparison of three new methods on characteristic problems from applications," *ACM Tran. on Math. Soft.*, vol. 9 pp. 333–359, 1993.

[52] M. Eisenberg, "Queues with periodic service and changeover times," *Oper. Res.*, vol. 20 pp. 440–451, 1972.

[53] M. Eisenberg, "Two queues with alternating service," *Siam J. Appl. Math.*, vol. 36 pp. 287–303, 1979.

[54] M. Eisenberg, "The polling system with stopping server," *QUESTA*, vol. 18 pp. 387–431, 1994.

[55] K.M. Elsaved and H.G. Perros, "The superposition of discrete-time Markov renewal processes with an application to statistical multiplexing of bursty traffic sources," preprint, 1995.

[56] D. Everitt, "A conservation-type law for the token ring with limited service," *British Telecom. Tech. J.*, vol. 2 pp. 51–61, 1986.

[57] D. Everitt, "A note on the pseudoconservation laws for cyclic service systems with limited service disciplines," *IEEE Trans. on Commun.*, vol. 37 pp. 781–783, 1989.

[58] O. Fabian and H. Levy, "Polling system optimization through dynamic routing policies," In *INFOCOM-93*, pp. 1250–1258, 1993.

[59] A. Federgruen and Z. Katalan, "Approximating queue size and waiting time distributions in general polling systems," *QUESTA*, vol. 18 pp. 353–386, 1994.

[60] M.J. Ferguson and Y.J. Aminetzah, "Exact results for nonsymmetric token ring systems," *IEEE Trans. on Commun.*, vol. 33 pp. 223–231, 1985.

[61] L. Fournier and Z. Rosberg, "Expected waiting time times in polling systems under priority disciplines," *QUESTA*, vol. 9 pp. 419–440, 1991.

[62] C. Fricker and M.R. Jaibi, "Monotonicity and stability of periodic models," *QUESTA*, vol. 15 pp. 211–238, 1995.

[63] I. Frigui and A.S. Alfa, "Approximate Method for Polling Systems with Time-Limited-Based Polling Tables," In *IEEE WESCANEX'95 Conference*, pp. 398–402, 1995.

[64] I. Frigui, A.S. Alfa, and X. Xu, "Algorithms for computing waiting time distributions under different queue disciplines for the D-BMAP/PH/1," Naval Research Logistic, 2nd revision, 1996.

[65] I. Frigui, R. Stone, and A. S. Alfa, "Message Delay for Priority-Based Automatic Meter Reading Network," *Computer Commun.*, vol. forthcoming, 1996.

[66] S.W. Fuhrmann, "Symmetric queues served in cyclic order," *Oper. Res. Letters*, vol. 4 pp. 139–144, 1985.

[67] S.W. Fuhrmann, "A decomposition result for a class of polling models," *QUESTA*, vol. 11 pp. 109–120, 1992.

[68] S.W. Fuhrmann and R. Cooper, "Stochastic decomposition in the M/G/1 queue with generalized vacations," *Oper. Res.*, vol. 33 pp. 1117–1129, 1985.

[69] S.W. Fuhrmann and Y. T. Wang, "Analysis of cyclic service systems with limited service: Bounds and approximations," *Perf. Eval.*, vol. 9 pp. 35–54, 1988.

[70] A. Ganz and I. Chlamtac, "A linear solution to queueing analysis of synchronous finite buffer networks," *IEEE Trans. on Commun.*, vol. 38 pp. 10–12, 1990.

[71] L. Georgiadis and W. Szpankowski, "Stability of token passing rings," *QUESTA*, vol. 11, 1992.

[72] J. Gianini and D.R. Manfield, "An analysis of symmetric polling systems with two priority classes," *Perf. Eval.*, vol. 8 pp. 93–115, 1988.

[73] N.P. Giannakouros and A. Laloux, "Waiting-time approximation for service systems with star polling sequence and mixed service strategies," *IEEE Trans. on Commun.*, vol. 39 pp. 1041–1045, 1991.

[74] W.K. Grassmann, "Computational methods in probability theory," In P.D. Heyman and M.J. Sobel, editors, *Stochastic Models*, pp. 199–253, Amsterdam, North-Holland, 1990. Elsevier Science Publishers B. V.

[75] B. Grela-M'Poko, M.M. Ali, and J.F. Hayes, "Approximate analysis of asymmetric single-service prioritized token passing systems," *IEEE Trans. on Commun.*, vol. 39 pp. 1037–1040, 1991.

[76] W.P. Groenendijk, "Waiting-time approximations for cyclic-service systems with mixed service strategies," In M. Bonati, editor, *Teletraffic Science for New Cost-Effective Systems, Networks and Services, ITC-12*, pp. 1434–1441, North-Holland, 1989. Elsevier Science Publishers B.V.

[77] W.P. Groenendijk and H. Levy, "Performance analysis of transaction driven computer systems via queueing analysis of polling models," *IEEE Trans. on Comput.*, vol. 41 pp. 455–466, 1992.

[78] O. Hashida, "Analysis of Multiqueues," *Rev. Elec. Commun. Lab.*, vol. 20 pp. 189–199, 1972.

[79] H. Heffes and D.M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Select. Areas in Commun.*, vol. 4 pp. 856–868, 1986.

[80] G.J. Henry, "The fair share scheduler," *AT&T Bell Lab. Tech. J.*, vol. 63 pp. 1845–1857, 1984.

[81] M. Hofri and K.W. Ross, "On the Optimal control of two queue with server set-up times and its analysis," *IEEE Trans. on Commun.*, vol. 35 pp. 283–288, 1987.

[82] S. Hong, H.G. Perros, and H. Yamashita, "A discrete-time queueing model of the shared buffer ATM switch with bursty arrival," *Telecommunication Systems*, vol. 2 pp. 1–20, 1993.

[83] S.H. Hong, "Approximate analysis of timer-controlled priority scheme in the single-service token-passing systems," *IEEE/ACM Trans. on Networking*, vol. 2 pp. 206–215, 1994.

[84] P. Humblet, "Source coding for communication concentrators," Technical report, Electron. Syst. Lab. MIT,Cambridge, MA, ESL-R-798, 1978.

[85] L.C. Hwang and C.J. Chang, "Optimal design of a finite-buffer polling network with mixed mixed service discipline and general service order sequence," *IEE Proc.-Commun.*, vol. 142 pp. 1–6, 1995.

[86] O. Ibe and K. Trivedi, "Stochastic Petri net models of polling systems," *IEEE J. Select. Areas in Commun.*, vol. 8 pp. 6–12, 1990.

[87] M.A. Johnson and M.R. Taaffe, "Matching moments to phase distributions: Mixtures of Erlang distributions of common order," *Commun. Statist.-Stoch. Mod.*, vol. 5 pp. 711–743, 1989.

[88] M.A. Johnson and M.R. Taaffe, "Matching moments to phase distributions: nonlinear programming approaches," *Commun. Statist.-Stoch. Mod.*, vol. 6 pp. 258–281, 1989.

[89] F. Jou, A. Nilsson, and F. Lai, "The upper bounds for delay and cell loss probability of bursty ATM traffic in a finite capacity polling system," In *Second Int. Workshop on Queueing Networks with Blocking*, RTP, North Caroline, 1992.

[90] W.Y. Jung and C.K. Un, "Analysis of a finite-buffer polling system with exhaustive service based on virtual buffering," *IEEE Trans. on Commun.*, vol. 42 pp. 3144–3149, 1994.

[91] D. Karvelas and L. Leon-Garcia, "Performance of integrated packet voice/data token rings," *IEEE J. Select. Areas in Commun.*, vol. 6 pp. 823–832, 1986.

[92] O. Kella and U. Yechiali, "Priorities in M/G/1 queue with server vacations," *Naval Research Logistics, Quart.*, vol. 35 pp. 23–34, 1988.

[93] D.B. Keogh, "Birth-death processes suitables for modelling the output of video coders," *ATR*, vol. 25, 1991.

[94] D.B. Keogh, "Birth-death processes suitables for modelling the output of video coders-Part II," *ATR*, vol. 28, 1994.

[95] D.B. Keogh, "Markov model of the short and long term behaviour of variable bit rate video sources," *ATR*, vol. 29, 1995.

[96] L. Kleinrock, "A conservation law for a wide class of queueing disciplines," *Naval Logistics Research Quart.*, vol. 12 pp. 181–192, 1965.

[97] L. Kleinrock, "Scheduling, queueing and delays in time-shared systems and computer networks," In N. Abramson and F.F. Kuo, editors, *Computer-Communication Networks*, pp. 95–141, Englewood Cliffs, NJ, 1973. Prentice-Hall.

[98] L. Kleinrock, *Queueing Systems.*, volume 2 John Wiley and Sons, New York, N.Y, 1976.

[99] L. Kleinrock, "Performance evaluation of distributed computer-communication systems," In O.J. Boxma and R. Syski, editors, *Queueing Theory and Its Applications - Liber Amicorum for J.W. Cohen*, pp. 1–57. Elsevier Science Publisher B.V. North-Holland, 1988.

[100] L. Kleinrock and H. Levy, "The analysis of random polling systems," *Oper. Res.*, vol. 36 pp. 716–732, 1988.

[101] E.M. Klimko and M.F. Neuts, "The single server queue in discrete time - Numerical analysis II," *Naval Research Logistics, Quarterly*, vol. 20 pp. 305–319, 1973.

[102] H. Kobayashi and A.G. Konheim, "Queueing models for computer communications systems analysis," *IEEE Trans. on Commun.*, vol. 25 pp. 2–29, 1977.

[103] D. Kofman, "Blocking probability, throughput and waiting time in finite capacity polling systems," *QUESTA*, vol. 14 pp. 385–411, 1993.

[104] A.G. Konheim, H. Levy, and M.M. Srinivasan, "Descendant set: An efficient approach for the analysis of polling systems," *IEEE Trans. on Commun.*, vol. 42 pp. 1245–1252, 1994.

[105] A.G. Konheim and B. Meister, "Waiting lines and times in a system with polling," *J. Assoc. Comput. Mach.*, vol. 21 pp. 470–490, 1974.

[106] P.J. Kuehn, "Multiqueue systems with nonexhaustive cyclic service," *Bell Syst. Tech. J.*, vol. 59 pp. 671–699, 1979.

[107] A. Lang and J.L. Arthur, "Parameter approximation for phase-type distributions," In S. Chakravarthy and A.S. Alfa, editors, *First International Conference on Matrix-Analytic Methods in Stochastic Models*, pp. 151–206. Marcel Dekker Inc., 1996.

[108] A.M. Law and W.D. Kelton, *Simulation Modeling & Analysis.* McGraw-Hill, Inc., New York, NY, second edition, 1991.

[109] D.S. Lee and B. Sengupta, "An approximation analysis of cyclic server queue with limited service and reservation with application to satellite communications," *QUESTA*, vol. 11 pp. 153–178, 1992.

[110] T.T. Lee, "M/G/1/N queue with vacation time and exhaustive service discipline," *32*, pp. 774–784, 1984.

[111] T.T. Lee, "M/G/1/N queue with vacation time and limited discipline," *Perf. Eval.*, vol. 9 pp. 181–190, 1988.

[112] K.K. Leung, "Cyclic-service systems with probabilistically-limited service," *IEEE J. Select. Areas in Commun.*, vol. 9 pp. 185–193, 1991.

[113] K.K. Leung, "Cyclic-service systems with nonpeemptive time-limited service," *IEEE Trans. on Commun.*, vol. 42 pp. 2521–2524, 1994.

[114] K.K. Leung and M. Eisenberg, "A single server queue with vacations and gated time-limited service," *IEEE Trans. on Commun.*, vol. 38 pp. 1454–1462, 1990.

[115] K.K. Leung and D.M. Lucantoni, "Two vacation models for token-ring networks where service is controlled by timers," *Perf. Eval.*, vol. 20 pp. 165–184, 1994.

[116] H. Levy, "Binomial-gated service: A method for effective operation and optimization of polling systems," *IEEE Trans. on Commun.*, vol. 39 pp. 1341–1350, 1991.

[117] H. Levy and L. Kleinrock, "Polling systems with zero switch-over periods. A general method for analyzing the expected delay," *Perf. Eval.*, vol. 13 pp. 97-107, 1991.

[118] H. Levy and M. Sidi, "Correlated arrivals in polling systems," In *IEEE INFOCOM'89*, pp. 907–913, Ottawa, Canada, 1989.

[119] H. Levy and M. Sidi, "Polling systems: Applications, modeling, and optimization," *IEEE Trans. on Commun.*, vol. 38 pp. 1750–1759, 1990.

[120] H. Levy and M. Sidi, "Polling systems with simultaneous arrivals," *IEEE Trans. on Commun.*, vol. 39 pp. 823–827, 1991.

[121] H. Levy, M. Sidi, and O.J. Boxma, "Dominance relations in polling systems," *QUESTA*, vol. 6 pp. 155–172, 1990.

[122] Z. Liu and P. Nain, "Optimal scheduling in some multi-queue singel server systems," *IEEE Trans. Auto. Contr.*, vol. 37 pp. 247–252, 1992.

[123] Z. Liu, P. Nain, and D. Towsley, "On optimal polling policies," *QUESTA*, vol. 11 pp. 59–83, 1992.

[124] C.C. Lu and K.Y. Lin, "Delay time analysis of FDDI protocol," In *INFOCOM-91*, pp. 1440–1445, 1991.

[125] D.M. Lucantoni, "New results on the single server queue with a batch Markovian arrival process," *Commun. Statist.-Stoch. Mod.*, vol. 7 pp. 1–46, 1991.

[126] K.M. Lye and K.G. Seah, "Random polling scheme with priority," *Electronics Letters*, vol. 28 pp. 1290–1291, 1992.

[127] C. Mack, "The efficiency of N machines unidirectionally patrolled by one operative when walking time is constant and repair times are variable," *J. of the Royal Statistical Society, Series B*, vol. 19 pp. 173–178, 1957.

[128] C. Mack, T. Murphy, and N.L. Webb, "The efficiency of N machines unidirectionally patrolled by one operative when walking time and repair times are constants," *J. of the Royal Statistical Society, Series B*, vol. 19 pp. 166–172, 1957.

[129] M. Manfield, "Analysis of a priority polling system for two-way traffic," *IEEE Trans. on Commun.*, vol. 33 pp. 1001–1006, 1985.

[130] K.S. Meir-Hellstern, "A fitting algorithm for Markov-modulated Poisson process having two arrival rates," *EJOR*, vol. 29 pp. 370–377, 1987.

[131] M. F. Neuts, "Probability distribution of phase type," In H. Florin, editor, *Liber Amicorum Prof. Emeritus*, pp. 173–206, Belgium, 1975. University of Louvain.

[132] M.F. Neuts, "A versatile Markovian point process," *J. of Appl. Probab.*, vol. 16 pp. 764–779, 1979.

[133] M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models - An Algorithmic approach.* The Johns Hopkins University Press, Baltimore, MD, 1981.

[134] M.F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and their Applications.* Marcel Dekker, 1989.

[135] M.F. Neuts, "Modelling with the Markovian arrival process," Technical report, Dept. Systems and Industrial Eng., Univ. of Arizona, 1992.

[136] G.F. Newell, "Properties of vehicle-actuated signals: I. One-way streets," *Transportation Science*, vol. 3 pp. 30–52, 1969.

[137] G.F. Newell and E.E. Osuna, "Properties of vehicle-actuated signals: II. Two-way streets," *Transportation Science*, vol. 3 pp. 99–125, 1969.

[138] J. Ni, T. Ynag, and D.H.K. Tsang, "Source modelling, queueing analysis, and bandwidth allocation for VBR MPEG-2 video traffic in ATM networks," *IEE Proc.-Commun.*, vol. 143, 1996.

[139] J.W.M. Pang and F.A. Tobagi, "Throughput analysis of a timer controlled token passing protocol under heavy load," *IEEE Trans. on Commun.*, vol. 37 pp. 694–702, 1989.

[140] B.E. Patuwo, R.L. Disney, and D.C. McNickle, "The effect of correlated arrivals on queues," *IIE Trans.*, vol. 25 pp. 105–110, 1993.

[141] B.K. Penney and A.A. Baghdadi, "Survey of computer communications loop networks:Part 1," *Comput. Commun.*, vol. 2 pp. 165–180, 1979.

[142] B.K. Penney and A.A. Baghdadi, "Survey of computer communications loop networks:Part 2," *Comput. Commun.*, vol. 2 pp. 224–241, 1979.

[143] J.A.C. Resing, "Polling systems and multi-type branching processes," *QUESTA*, vol. 13 pp. 409–426, 1993.

[144] I. Rubin and L.F. de Moraes, "Message delay analysis for polling and token multiple-access schemes for local communications networks," *IEEE J. Select. Areas in Commun*, vol. 1 pp. 935–947, 1983.

[145] I. Rubin and J.C.H. Wu, "Analysis of an M/G/1/N queue with vacations and its iterative application to FDDI time-token rings," *IEEE/ACM Trans. on Networking*, vol. 3 pp. 842–856, 1995.

[146] T. Ryden, "Parameter estimation for Markov modulated Poisson process," preprint, 1992.

[147] S.R. Sachs, "Alternative local area network access protocols," *IEEE Commun. Magazine*, vol. 26 pp. 25–45, 1988.

[148] D. Sarkar and W.I. Zangwill, "Expected waiting time for nonsymmetric cyclic queueing systems-exact results and applications," *Management Science*, pp. 1463–1474, 1987.

[149] R.W. Scheifler and J. Gettys, "The X window system," *ACM Trans. on Graph.*, vol. 5 pp. 79–109, 1986.

[150] M. Scholl and L. Kleinrock, "On the M/G/1 queue with rest periods and certain service-independent queueing disciplines," *Oper. Res.*, vol. 31 pp. 705–719, 1983.

[151] M. Schwartz, *Computer-Communication Network Design and Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1977.

[152] L.D. Servi, "Capacity estimation of cyclic queues," *IEEE Trans. on Commun.*, vol. 33 pp. 279–282, 1985.

[153] L.D. Servi, "Average delay approximation of M/G/1 cyclic service queues with Bernoulli schedules," *IEEE J. Select. Areas in Commun.*, vol. 4 pp. 813–822, 1986.

[154] J.G. Shanthikumar, "Analysis of priority queues with server control," *Opsearch*, vol. 21 pp. 183–192, 1984.

[155] S. Shimogawa and Y. Takahashi, "A note on the pseudo-conservation law for a multi-queue with local priority," *QUESTA*, vol. 11 pp. 145–151, 1992.

[156] M. Sidi, H. Levy, and S.W. Fuhrmann, "A queueing network with a single cyclically roving server," *QUESTA*, vol. 11 pp. 121–144, 1992.

[157] M.M. Srinivasan, "An approximation for mean waiting times in cyclic server systems with nonexhaustive service," *Perf. Eval.*, vol. 9 pp. 17–33, 1988.

[158] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Select. Areas in Commun.*, vol. 4 pp. 833–846, 1986.

[159] ANSI Standard, *FDDI Token Ring-Media Access Control.* ANSI X3.139-1987, 1986.

[160] I. Stavrakakis, "Delay bounds on a queueing system with consistent Priorities," *IEEE Trans. on Commun.*, vol. 42 pp. 615–624, 1994.

[161] W.J. Stewart, *Introduction to Numerical Solutions of Markov Chains.* Princeton University Press, Princeton, NJ, 1994.

[162] S. Stidham, "Regenerative processes in the theory of queues, with applications to the alternating-priority queue," *Adv. in Appl. Probab.*, vol. 4 pp. 542–577, 1972.

[163] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models.* John Wiley, New York, NY, 1983.

[164] G.B. Swartz, "Polling in a loop system," *J. Assoc. Comput. Mach.*, vol. 27 pp. 42–59, 1980.

[165] H. Takagi, *Analysis of Polling Systems.* M.I.T. Press, Cambridge, MA, 1986.

[166] H. Takagi, "Queueing analysis of polling models," *ACM Comp. Surv.*, vol. 20 pp. 5–28, 1988.

[167] H. Takagi, "Queueing analysis of polling models: An update," In H. Takagi, editor, *Stochastic Analysis of Computer and Communication Systems*, pp. 267–318, Amsterdam, North-Holland, 1990. Elsevier Science Publishers B. V.

[168] H. Takagi, "Analysis of finite capacity polling systems," *J. Appl. Probab.*, vol. 23 pp. 373–387, 1991.

[169] H. Takagi, "Applications of polling models to computer networks," *Computer Networks and ISDN Systems*, vol. 22 pp. 193–211, 1991.

[170] H. Takagi, *Queueing Analysis: A Foundation of Performance Analysis, Volume 1 Vacation and Priority Systems.* Elsevier Science Publishers B. V., Amsterdam, North-Holland, 1991.

[171] H. Takagi and K.K. Leung, "Analysis of a discrete-time queueing system with time-limited service," *QUESTA*, vol. 18 pp. 183–197, 1994.

[172] Y. Takahashi and B.K. Kumar, "Pseudo-conservation law for a priority polling system with mixed service strategies," *Perf. Eval.*, vol. 23 pp. 107–120, 1995.

[173] T. Takine and T. Hasegawa, "A cyclic-server finite source model with round-robin scheduling," *QUESTA*, vol. 11 pp. 91–108, 1992.

[174] A.S. Tanenbaum, *Operating Systems, Design and Implementation.* Prentice-Hall, Englewood Cliffs, NJ, 1987.

[175] A.S. Tanenbaum, *Modern Operating Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1992.

[176] T.E. Tedijanto, "Exact analysis for the cyclic-service queue with a Bernoulli schedule," *Perf. Eval.*, vol. 11 pp. 107–115, 1990.

[177] P. Tran-Gia, "Analysis of polling systems with general input process and finite capacity," *IEEE Trans. on Commun.*, vol. 40 pp. 1078–1092, 1992.

[178] P. Tran-Gia and R. Dittmann, "A discrete-time analysis of the cyclic reservation multiple access protocol," *Perf. Eval.*, vol. 16 pp. 185–200, 1992.

[179] Z. Tsai and I. Rubin, "Analysis for token ring networks operating under message priorities and delay limits," In *IEEE INFOCOM'89*, pp. 322–331, Ottawa, Canada, 1989.

[180] Z. Tsai and I. Rubin, "Mean delay analysis for a message priority-based polling scheme," *QUESTA*, vol. 11 pp. 223–240, 1992.

[181] K.S. Watson, "Performance evaluation of cyclic service strategies -A survey," In E. Gelenbe, editor, *Performance'84*, pp. 521–533, New York: North-Holland, 1984.

[182] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Trans. Networking*, vol. 1 pp. 372–385, 1993.

[183] O.C. Yue and C.A. Brooks, "Performance of time token scheme in MAP," *IEEE Trans. on Commun.*, vol. 38 pp. 1006–1012, 1990.

[184] A. Zaghloul and H. Perros, "Approximate analysis of a discrete-time polling system with bursty arrival," In *IFIP Workshop on Modelling and Performance Evaluation of ATM Technology*, Maritinique, 1993.

[185] A.O. Zaghloul and H. G. Perros, "Approximate analysis of a shared-medium ATM switch under bursty arrivals and nonuniform destinations," *Perf. Eval.*, vol. 21 pp. 111–129, 1994.

[186] V.S. Zhdanov and E.A. Saksonov, "Conditions of existence of steady-state modes in cyclic queueing systems," *Autom. Remote Control*, vol. 40 pp. 176–184, 1979.

APPENDICES

APPENDIX A

EXTENSION TO VARIABLE TIME LIMIT

A.1 Introduction

Consider a single server queue with Markovian arrival process (MAP) of dimension m and representation (D_0, D_1), phase type (PH) service distribution of dimension n and representation $(\boldsymbol{\beta}, S)$, and N phase type vacation distribution of dimension r_u and representation $(\boldsymbol{\delta}_u, L_u)$, $u = 1, \ldots N$. The service period is exhaustive time-limited (preemptive). In addition, prior to the vacation period of type u the visit period has a time limit T_v, $v = 1, \ldots, M$ and $M \leq N$. Let Q and $\bar{Q}$ be two irreducible Markov chains of dimension $q = max(N, M)$ where $Q_{i,j}$ denotes the probability that at the end of visit period of type i the server takes a vacation of type j. And $\bar{Q}_{i,j}$ denotes the probability that at the end of vacation period i the visit period will be of type j. For example, let $N = M = 3$ then the transition matrices Q and $\bar{Q}$ are given, respectively, by:

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}, \quad \bar{Q} = \begin{bmatrix} \bar{q}_{11} & \bar{q}_{12} & \bar{q}_{13} \\ \bar{q}_{21} & \bar{q}_{22} & \bar{q}_{23} \\ \bar{q}_{31} & \bar{q}_{32} & \bar{q}_{33} \end{bmatrix}.$$

A typical cycle is given in Table A.1. Notice that we consider the case when the server comes from vacation and finds the queue empty (the server goes on another vacation) to be an end of a visit period. Therefore, the transition from the end of a vacation of type i and beginning of a vacation of type j because the queue is empty will be denoted by q_{ij}.

Table A.1: Visit and Vacation Cycle for $N = M = 3$

Visit	Vacation	Visit	Vacation	Visit	Vacation
$\tilde{q}_{11}T_1$	$q_{11}(\boldsymbol{\delta}_1, L_1)$	$\tilde{q}_{12}T_2$	$q_{22}(\boldsymbol{\delta}_2, L_2)$	$\tilde{q}_{23}T_3$	$q_{33}(\boldsymbol{\delta}_3, L_3)$

The state space of the Markov chain of this queueing system is given by:

$$\Delta = \{(i,(0,k,l'_u,u) \cup (j_v,k,l,v)) \text{ where } \begin{cases} i \geq 0; \\ j_v = 1,2,\cdots,T_v; k = 1,2,\cdots,n; \\ l'_u = 1,2,\cdots,r_u; l = 1,2,\cdots,m, \end{cases} \text{ where } i$$

is the number of customers in the queue during service (vacation); the four tuple

$$(0,k,l'_u,u) \text{ refers to } \begin{cases} \text{vacation period represented by 0;} \\ k \text{ representing the phase of arrival;} \\ l'_u \text{ representing the phase of vacation type } u; \\ u \text{ vacation type } u = 1,\ldots N. \end{cases}$$ The four tuple

$$(j_v,k,l,v) \text{ refers to the service state with } \begin{cases} j_v \text{ time clock of service } 1 \leq j_v \leq T_v; \\ k \text{ representing the phase of arrival;} \\ l \text{ representing the phase of service;} \\ v \text{ visit period type } v = 1,\ldots M. \end{cases}$$ The

transition matrix of this Markov chain P is given as

$$P = \begin{vmatrix} B_{00} & B_{01} & 0 & 0 & 0 & \cdots & \cdots \\ B_{10} & A_1 & A_0 & 0 & 0 & \cdots & \cdots \\ 0 & A_2 & A_1 & A_0 & 0 & \cdots & \cdots \\ 0 & 0 & A_2 & A_1 & A_0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{vmatrix},$$

where,

For example, let $N = M = 3$. The block matrices are then given by:

$$B_{00} = \begin{bmatrix} D_0 \otimes (L_1 + q_{11}\mathbf{L}_1^{\circ}\boldsymbol{\delta}_1) & D_0 \otimes q_{12}\mathbf{L}_1^{\circ}\boldsymbol{\delta}_2 & D_0 \otimes q_{13}\mathbf{L}_1^{\circ}\boldsymbol{\delta}_3 \\ D_0 \otimes q_{21}\mathbf{L}_2^{\circ}\boldsymbol{\delta}_1 & D_0 \otimes (L_2 + q_{22}\mathbf{L}_2^{\circ}\boldsymbol{\delta}_2) & D_0 \otimes q_{23}\mathbf{L}_2^{\circ}\boldsymbol{\delta}_3 \\ D_0 \otimes q_{31}\mathbf{L}_3^{\circ}\boldsymbol{\delta}_1 & D_0 \otimes q_{32}\mathbf{L}_3^{\circ}\boldsymbol{\delta}_1 & D_0 \otimes (L_3 + q_{33}\mathbf{L}_3^{\circ}\boldsymbol{\delta}_3 \end{bmatrix}.$$

$$B_{01} = \begin{bmatrix} D_1 \otimes L_1 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{11}\mathbf{L}_1^{\circ}\boldsymbol{\beta}) & 0 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{12}\mathbf{L}_1^{\circ}\boldsymbol{\beta}) & 0 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{13}\mathbf{L}_1^{\circ}\boldsymbol{\beta}) \\ 0 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{21}\mathbf{L}_2^{\circ}\boldsymbol{\beta}) & D_1 \otimes L_2 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{22}\mathbf{L}_2^{\circ}\boldsymbol{\beta}) & 0 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{23}\mathbf{L}_2^{\circ}\boldsymbol{\beta}) \\ 0 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{31}\mathbf{L}_3^{\circ}\boldsymbol{\beta}) & 0 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{32}\mathbf{L}_3^{\circ}\boldsymbol{\beta}) & D_1 \otimes L_3 & \boldsymbol{e}_1' \otimes D_1 \otimes (q_{33}\mathbf{L}_3^{\circ}\boldsymbol{\beta}) \end{bmatrix}.$$

$$B_{10} = \begin{bmatrix} 0 & 0 & 0 \\ \boldsymbol{e} \otimes D_0 \otimes (q_{11}\mathbf{S}^{\circ}\boldsymbol{\delta}_1) & \boldsymbol{e} \otimes D_0 \otimes (q_{12}\mathbf{S}^{\circ}\boldsymbol{\delta}_2) & \boldsymbol{e} \otimes D_0 \otimes (q_{13}\mathbf{S}^{\circ}\boldsymbol{\delta}_3) \\ \boldsymbol{e} \otimes D_0 \otimes (q_{21}\mathbf{S}^{\circ}\boldsymbol{\delta}_1) & \boldsymbol{e} \otimes D_0 \otimes (q_{22}\mathbf{S}^{\circ}\boldsymbol{\delta}_2) & \boldsymbol{e} \otimes D_0 \otimes (q_{23}\mathbf{S}^{\circ}\boldsymbol{\delta}_3) \\ \boldsymbol{e} \otimes D_0 \otimes (q_{31}\mathbf{S}^{\circ}\boldsymbol{\delta}_1) & \boldsymbol{e} \otimes D_0 \otimes (q_{32}\mathbf{S}^{\circ}\boldsymbol{\delta}_2) & \boldsymbol{e} \otimes D_0 \otimes (q_{33}\mathbf{S}^{\circ}\boldsymbol{\delta}_3) \end{bmatrix}.$$

$$A_v = \begin{bmatrix} {}^1A_v^3 & q_{11}\,{}^1A_v^2 & & q_{12}\,{}^1A_v^2 & & q_{13}\,{}^1A_v^2 & \\ 0 & 0 & I \otimes ({}^1A_v^0) & & & & \\ q_{11}\,{}^1A_v^1 & & q_{12}\,{}^2A_v^1 & & q_{13}\,{}^3A_v^1 & & \\ & q_{21}\,{}^2A_v^2 & {}^2A_v^3 & q_{22}\,{}^2A_v^2 & & q_{23}\,{}^2A_v^2 & \\ 0 & 0 & & & I \otimes ({}^2A_v^0) & & \\ q_{21}\,{}^1A_v^1 & & q_{22}\,{}^2A_v^1 & & q_{23}\,{}^3A_v^1 & & \\ & q_{31}\,{}^3A_v^2 & {}^2A_v^3 & q_{32}\,{}^3A_v^2 & & q_{33}\,{}^3A_v^2 & \\ 0 & 0 & & & & & I \otimes ({}^3A_v^0) \\ q_{31}\,{}^1A_v^1 & & q_{32}\,{}^2A_v^1 & & q_{33}\,{}^3A_v^1 & & \end{bmatrix}.$$

$v = 0, 1, 2$ and ${}^iA_2^3 = {}^iA_2^2 = 0$, where,

$$\begin{cases} {}^iA_0^0 = D_1 \otimes S, \\ {}^iA_0^1 = D_1 \otimes (S\boldsymbol{e})\boldsymbol{\delta}_i, \\ {}^iA_0^2 = D_1 \otimes (\mathbf{L}_i^{\circ}\boldsymbol{\beta}^*), \\ {}^iA_0^3 = D_1 \otimes L_i \end{cases} \quad \begin{cases} {}^iA_1^0 = D_0 \otimes S + D_1 \otimes (\mathbf{S}^{\circ}\boldsymbol{\beta}), \\ {}^iA_1^1 = D_0 \otimes (S\boldsymbol{e})\boldsymbol{\delta}_i + D_1 \otimes (\mathbf{S}^{\circ}\boldsymbol{\delta}_i), \\ {}^iA_1^2 = D_0 \otimes (\mathbf{L}_i^{\circ}\boldsymbol{\beta}^*), \\ {}^iA_1^3 = D_0 \otimes L_i, \end{cases}$$

$$\begin{cases} {}^iA_2^0 = D_0 \otimes (\mathbf{S}^{\circ}\boldsymbol{\beta}), \\ {}^iA_2^1 = D_0 \otimes (\mathbf{S}^{\circ}\boldsymbol{\delta}_i), \end{cases} \quad \mathbf{L}_i^{\circ} = \boldsymbol{e} - L_i\boldsymbol{e},\ \mathbf{S}^{\circ} = \boldsymbol{e} - S\boldsymbol{e},\ i = 1, 2, 3.$$

and $\boldsymbol{\beta}^* = \boldsymbol{\beta}^*(S + \mathbf{S}^{\circ}\boldsymbol{\beta})$, with $\boldsymbol{\beta}^*\boldsymbol{e} = 1$,

where $\mathbf{S}^{\circ} = \boldsymbol{e} - S\boldsymbol{e}$, and $\mathbf{L}^{\circ} = \boldsymbol{e} - L\boldsymbol{e}$.