1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

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

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

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

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

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

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

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

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses
on Microfiche Service

Services des thèses canadiennes
sur microfiches

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moullées ou dactylographier

Full Name of Author — Nom complet de l'auteur

GLENFORD EZRA MAPP

Date of Birth — Date de naissance

11th February 1961

Country of Birth — Lieu de naissance

TRINIDAD AND TOBAGO

Permanent Address — Residence fixe

3, BEDESSIE TRACE,
WATTS STREET, ST AUGUSTINE
TRINIDAD, WEST INDIES

Title of Thesis — Titre de la thèse

"THE ANALYSIS OF PRIORITY POLLING SYSTEMS
WITH COMPLEX CYCLES"

University — Université

CARLETON UNIVERSITY

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

MASTER OF ENGINEERING

Year this degree conferred — Année d'obtention de ce grade

1985

Name of Supervisor — Nom du directeur de thèse

DR DAVID MANFIELD

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication, ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

12th Sept 1985

Signature

Glenford Mapp

THE ANALYSIS OF PRIORITY POLLING SYSTEMS
WITH COMPLEX CYCLES

by

Glenford E. Mapp, B.Sc.

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

Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
Ottawa, Ontario

© August 1985

The undersigned recommend to the Faculty of Graduate
Studies and Research, acceptance of the thesis

**"THE ANALYSIS OF PRIORITY POLLING SYSTEMS
WITH COMPLEX CYCLES"**

by Glenford E. Mapp, submitted in partial fulfillment of
the requirements of the degree of Master of Engineering.

Thesis Supervisor

Chairman,
Department of Systems and
Computer Engineering

Carleton University

September 1985

ABSTRACT

Polling mechanisms are commonly used to arbitrate access by distributed stations to a shared communications medium. Stations may be polled either explicitly by a central controller or implicitly by token passing. When a station is polled and has a message to send, it seizes the medium and transmits its message after which the polling cycle continues. Polling systems are found in switching systems with distribution control, computer communications systems and the cyclic scheduling algorithms of real-time computers.

A usual requirement of such systems is to provide priority access to certain classes of messages. This can be achieved via a complex polling cycle in which the number of visits to a particular station in the overall cycle is proportional to the priority of that station in the system.

This thesis presents the performance modelling and analysis of priority polling systems with complex cycles. The analysis is based on a multiqueue $M[x]/G/1$ model with non-exhaustive service and generally-distributed service and queue switchover times. The concept of priority polling subcycles is first introduced and expressions are developed for the first and second moments of the subcycle times. Using an imbedded Markov chain analysis,

simultaneous equations for the generating functions of the state probabilities of the queue at scan instants are obtained. With these results, expressions for characteristic function of the waiting time distribution and mean delay are derived in terms of the subcycle moments. Algorithms to compute the subcycle moments for use in the mean delay formula are developed. Some numerical results are given and validated against computer simulations.

In general, it was found that the accuracy of this approximate technique decreased with increased variance of the service time distribution, increased traffic, increased number of queues, subcycles and priority levels in the complex cycle. However, results within the normal operating range of 30-50% utilization of the communications medium have always been found to be within 10% of the exact value and are suitable for practical engineering purposes.

ACKNOWLEDGEMENTS

I would like to express my thanks to all those who have contributed to the research and preparation of this thesis.

I wish to express my deepest appreciation for the guidance and help of my thesis supervisor, Dr. David Manfield. His experience and insight were invaluable to the development and completion of this thesis. It was a great honour for me to work with him.

I would like to thank my graduate colleagues, too numerous to mention, for their encouragement and support. Many thanks to Diane Dodds for the typing and presentation of this work. I would also like to thank Professor A.U.H. Sheikh, my program advisor, and Professor A.R. Kaye for their help.

Finally, I wish to thank the Association of Universities and Colleges of Canada and the Administration of the Canadian Commonwealth Scholarships and Fellowship Plan, without whom my participation in this program would not have been possible.

TABLE OF CONTENTS

ABSTRACT	111
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF SYMBOLS	xi
 CHAPTER 1 -- INTRODUCTION	 1
1.1 Motivation	1
1.2 The General Queueing Model	3
1.3 Thesis Objective	7
1.4 Thesis Outline	7
 CHAPTER 2 -- A SURVEY OF PREVIOUS WORK	 9
2.1 Review of Literature	9
2.2 Review of Kuehn's Work [18]	11
2.2.1 Conditional Cycle Times	11
2.2.2 Markov Chain Analysis of the State Distribution At Scan Instants	16
2.2.3 State Distribution at Departure Instants	17
2.2.4 Delay Analysis	18
2.2.5 Evaluation of Kuehn's Method	19
2.3 Review of Krym's Thesis [19]	20
2.3.1 Network Model	20
2.3.2 Mathematical Model	22
2.3.3 The Mean Square Cycle Time	23
2.3.4 Mean Access Delay	24
2.3.5 Application of Krym's Technique to a General Queueing Model	26
2.3.6 The Mean Square Cycle Time For The General Queueing Model	26
2.3.7 Evaluation of Krym's Method	27

to the next queue in the sequence.

Queues may be served either under an exhaustive or non-exhaustive discipline. For exhaustive service, the server arrives at an arbitrary queue, and serves this queue until there are no customers left in the queue. With non-exhaustive service, the server arrives at queue i and serves up to l_i customers before moving on to the next queue. Non-exhaustive service with $l_i=1$ for all queues is referred to as ordinary non-exhaustive service. This thesis investigates ordinary non-exhaustive service, because of its widespread use in practical polling systems.

The waiting time or delay of a customer in queue i is defined as the time the customer spends in the queue waiting to be served. This is denoted by T_{W_i} with distribution function (d.f.) $W_i(\cdot)$ and mean w_i .

Customers in queue i require a random service time T_{H_i} , with d.f., $H_i(\cdot)$, and mean $h_i = ET_{H_i}$. When the server is finished serving a particular queue, it moves to the next queue in a finite switchover time known as the walktime. The walktime from queue x to queue y is given by $T_{U_{xy}}$ with d.f. $U_{xy}(\cdot)$ and mean u_{xy} . The total walktime in a cycle is represented by C_0 with mean c_0 .

Customers arrive according to a general independent (GI) arrival process with distribution function $A_1(t) = P\{T_{A_1} \leq t\}$ where T_{A_1} denotes the random variable of the interarrival time between customers in queue 1. The

4.3	Analysis of Complex Cyclic Service With Batch Input	66
4.3.1	State Distribution of an Arbitrary Queue at Scan Instants	66
4.3.2	State Distribution at Departure Instants During an Arbitrary Subcycle	69
4.3.3	Delay Analysis	69
4.4	Results	70
4.4.1	General Formula	71
CHAPTER 5 -- PRESENTATION OF ALGORITHMS TO COMPUTE THE MEAN DELAY		72
5.1	Introduction	72
5.2	System Characteristics	72
5.3	Subcycle Times	73
5.3.1	Introduction	73
5.3.2	Information Gathering Routines	78
5.3.3	The Coefficient Matrix	83
5.3.4	Walktime Contributions	87
5.3.5	The Right Hand Column Vector	88
5.3.6	The Linear Equations	89
5.4	The Conditional Mean Subcycle Times	89
5.5	The Second Moment of the Subcycle Time	95
CHAPTER 6 -- RESULTS AND DISCUSSION		98
6.1	Results	98
6.2	Discussion	102
6.3	Scope for Future Work	107
CHAPTER 7 -- CONCLUSIONS		109
REFERENCES		110

LIST OF FIGURES

1.2.1	General Multiqueue Polling Model	4
2.2.1	General Cycle Time T_{Cj}	13
2.3.1	Frame Format	21
3.2.1	The Elementary Priority Cycle	33
3.2.2	General Subcycle C_{1k}	35
3.4.1	General Subcycle C_{1k} Conditioned on the State of Queue 1 During Subcycle p	41
3.4.2	The Elementary Priority Cycle Conditioned on the State of Queue 1 During the C_{11} Subcycle	43
3.4.3	General Subcycle C_{1k} Conditioned on the State of Queue s During Subcycle t	47
3.5.1	The State Distribution at the $(k-1)^{th}$ and k^{th} Scan Instants of Queue 1 for Complex Cyclic Service	50
4.3.1	The State Distribution at the $(k-1)^{th}$ and k^{th} Scan Instants of Queue 1 for Complex Cyclic Service With Batch Input	67
5.3.1	The 2^3 or Harmonic Priority Cycle	74
5.3.2	Subcycle C_{21} of the Harmonic Priority Cycle	77
5.3.3	Simultaneous Linear Equations for the Mean Subcycle Times of the HPQ of the Harmonic Priority Cycle	90
5.4.1	The Harmonic Priority Cycle Conditioned on at Least One Customer in Queue 1 During the First Subcycle	92
5.4.2	Simultaneous Linear Equations for the Conditional Mean Subcycle Times of the HPQ of the Harmonic Priority Cycle Conditioned On At Least One Customer Being in Queue 1 During the C_{11} Subcycle	96

6.1.1	Mean Delay Versus Total Utilization for the Elementary Priority Cycle With Uniform Loads and Deterministic Service Time Distribution	99
6.1.2	Mean Delay Versus Total Utilization for the Elementary Priority Cycle with Uniform Loads and Exponential Service Time Distribution	100
6.1.3	Mean Delay Versus Total Utilization for the Harmonic Priority Cycle With Uniform Load and Deterministic Service Time Distribution	101
6.1.4	Mean Delay Versus Total Utilization for the Harmonic Priority Cycle With Uniforms Loads and Exponential Service Time Distribution	103
6.1.5	Mean Delay Versus Total Utilization for the Elementary Priority Cycle With Mean Batch Size 4, Uniform Loads and Deterministic Service Time Distribution For Both Constant and Geometric Batch Size Distributions	104
6.1.6	Mean Delay Versus Total Utilization for the Harmonic Priority Cycle With Proportional Loads and Deterministic Service Time Distribution	105

LIST OF SYMBOLS

Greek Symbols

- α_{1k} - the probability of the server finding at least one customer in queue 1 on its arrival to queue 1 during subcycle C_{1k} .
- $\alpha_{1k,1p}'$ - the probability of the server finding at least one customer in queue 1 during subcycle C_{1k} given that the server finds no one in queue 1 during subcycle C_{1p} .
- $\alpha_{1k,1p}''$ - the probability of the server finding at least one customer in queue 1 during subcycle C_{1k} given that the server finds at least one customer in queue 1 during subcycle C_{1p} .
- $\alpha_{1k,st}'$ - the probability of the server finding at least one customer in queue 1 during subcycle C_{1k} given that the server finds no customer in queue s during subcycle C_{st} .
- $\alpha_{1k,st}''$ - the probability of the server finding at least one customer in queue 1 during subcycle C_{1k} given that the server finds at least one customer during subcycle C_{st} .
- λ_1 - the arrival rate of customers to queue 1.
- λ_{B1} - the arrival rate of batches to queue 1.
- ρ_1 - the utilization of queue 1.
- ρ_0 - the total utilization of the system.

General Symbols

- A_{1k} - $\begin{cases} 1 & \text{for } n_{1k} \geq 1 \\ 0 & \text{for } n_{1k} = 0 \end{cases}$
- C - the overall cycle.
- C_0 - the total walktime in the overall cycle.
- C_{1k} - subcycle k of queue 1.

C_{1k0}	- the total walktime during subcycle C_{1k} .
d	- the total number of queue positions in the complex cycle.
$d.f.$	- distribution function.
EPC	- The Elementary Priority Cycle.
EK_1	- mean batch size.
EK_i^2	- second moment of the batch size distribution.
g	- the total number of queues in the overall cycle.
$g(C_{1k})$	- the set of queues polled during subcycle C_{1k} .
$G_k(x)$	- the generating function of $p_k(n)$.
$G_k^*(x)$	- the generating function of $p_k^*(n)$.
HPQ	- the Highest Priority Queue.
K_1	- random batch size
l_1	- the maximum number of customers in queue 1 that can be served at any one time for limiting non-exhaustive service.
m_i	- the number of times queue i is polled in the overall cycle.
N_{1k}	- the number of customers the server finds in queue i on its arrival during subcycle C_{1k} .
$p_k(n)$	- the probability distribution of the number of customers the server finds in queue i on its arrival during the C_{1k} subcycle.
$p_k^*(n)$	- the number of customers in queue i as seen by a departing customer leaving queue i during the C_{1k} subcycle.
q_{1k}	- the probability distribution of K_1 .
RV	- Random Variable.
T_{A1}	- RV of the interarrival time between customers arriving to queue i .

T_{B1}	- RV of the interarrival time between batches arriving to queue 1.
T_{H1}	- RV of the service time of a customer in queue 1.
T_{Uxy}	- RV of the walktime between queue x and queue y.
T_Z	- RV of the time associated with parameter Z of the general polling system.
$V_j(C_{ik})$	- the set of subcycles of queue j that begin during subcycle C_{ik} .
$\text{Var}T_Z$	- Variance of T_Z .
$Z(\cdot)$	- the distribution function of T_Z .
$\phi Z(\cdot)$	- the transform function of $Z(\cdot)$.
$z, \bar{z}$ or ET_Z	- the expected value of T_Z .
$z(2)$	- the mean square or second moment of T_Z .
2^3 cycle	- the cycle denoted by (1,2,1,3,1,2,1,4).
Harmonic Priority Cycle	- see 2^3 cycle.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Polling is a method that is commonly used to regulate access to a shared resource. A typical example of a polling system is a communication bus that is shared among several stations. Access to the bus is controlled via a bus arbiter which polls each station in turn. If the polled station has messages to send it seizes the bus and transmits one or more messages after which the polling cycle continues.

Implementation of polling systems are found in data communications, switching systems with distributed control, and certain I/O mechanisms of real time computers. The performance of such systems is of great interest in the fields of traffic and systems engineering, where resources, sometimes costing millions of dollars, are allocated in the design and maintenance of large distributed systems. Methods of optimizing their performance, therefore, have received considerable attention from both the academic and industrial communities. This usually coincided with major technological developments in the telecommunications and computer industries.

In the late Sixties, extensive performance analysis was done in connection with the central control scheduling algorithm of the No.1 Electronic Switching System (#1 ESS) which was modelled as a cyclic polling system.

The early Seventies saw the evolution of computer networks from a single main frame with local terminals to large networks with numerous processing stations. These systems were frequently analyzed using multiqueue polling models. More recently, such models have been applied in the performance evaluation of Local Area Networks (LANs), in particular the Token Ring and Token Bus LANs where stations are implicitly polled as a token is passed to each station.

An example of a token passing network comprises a number of geographically distributed stations connected via a communications medium to form a ring. Access to the medium is controlled by means of a token which circulates around the ring. A token is a unique signalling sequence recognized by all stations. When a station has data to send and detects the token going by, it takes control of the medium by modifying the token and thus preventing access by other stations. It subsequently transmits its data onto the ring. All stations 'listen' to the transmission and the terminal to which it is addressed copies the messages. The data is removed from the ring by the sending station which then generates a new token and passes it to the next station in the sequence. All other stations are therefore polled before the token returns to

a given station. This is an effective way of ensuring fair and equal access to the shared communications resource.

However, the efficient operation of many systems require that certain stations have priority access to the communications medium. An example of such systems is a network on which a central facility (e.g., a file server or database) is heavily used by all other stations. To guarantee a reasonable response time, the central facility must have priority access to the medium or else it quickly becomes the system bottleneck.

Provision for priority access is specified in a number of standards. For example, in the IEEE 802.4 and 802.5 LAN standards, the MAC sub-layer specifications provides support for eight (8) levels of priority access.

1.2 The General Queuing Model

The multiqueue model for a general polling system is depicted in Figure 1.2.1. The polling mechanism is modelled as a single server visiting one queue after another in a cyclic schedule. The total number of queues in the system is denoted by g .

Messages arriving in the queue to be transmitted are called customers. When the server arrives at a particular queue, it checks the queue for customers waiting to be served. These epochs are referred to as scan instants. If the server finds no customers in the queue (i.e., the station has no messages to transmit), it immediately moves

Figure 1.21 General Multiqueue Polling Model

to the next queue in the sequence.

Queues may be served either under an exhaustive or non-exhaustive discipline. For exhaustive service, the server arrives at an arbitrary queue, and serves this queue until there are no customers left in the queue. With non-exhaustive service, the server arrives at queue i and serves up to l_i customers before moving on to the next queue. Non-exhaustive service with $l_i=1$ for all queues is referred to as ordinary non-exhaustive service. This thesis investigates ordinary non-exhaustive service, because of its widespread use in practical polling systems.

The waiting time or delay of a customer in queue i is defined as the time the customer spends in the queue waiting to be served. This is denoted by T_{W_i} with distribution function (d.f.) $W_i(\cdot)$ and mean w .

Customers in queue i require a random service time T_{H_i} , with d.f., $H_i(\cdot)$, and mean $h_i = ET_{H_i}$. When the server is finished serving a particular queue, it moves to the next queue in a finite switchover time known as the walktime. The walktime from queue x to queue y is given by $T_{U_{xy}}$ with d.f. $U_{xy}(\cdot)$ and mean u_{xy} . The total walktime in a cycle is represented by C_0 with mean c_0 .

Customers arrive according to a general independent (GI) arrival process with distribution function $A_1(t) = P\{T_{A_1} \leq t\}$ where T_{A_1} denotes the random variable of the interarrival time between customers in queue i . The

mean interarrival time is given by a_1 and the arrival rate, $\lambda_1 = 1/a_1$. In this thesis, we examine only the Markovian arrival process, given by $A_1(t) = 1 - e^{-\lambda_1 t}$.

For batch arrivals, the arrival process is defined by both the random interarrival time, T_{B1} of the batches and the random batch size K_1 with probability distribution $q_{1k} = P\{K_1 = k\}$. The arrival rate of the batches is given by λ_{B1} and the arrival rate of the customers, $\lambda_1 = EK_1 \lambda_{B1}$.

The general polling cycle is represented by $(q_1, q_2, \dots, q_d)$ where q_1 is the queue in the first position, q_d is the queue in the last position and d is the length of the polling cycle. When each queue is polled just once in each cycle, this is called simple cyclic service and can be represented by the cycle $(1, 2, 3, \dots, g)$ where d , the length of the cycle, is equal to g , the number of queues in the system.

In contrast, when queues are visited at different frequencies during the polling cycle, this is referred to as complex cyclic service. For example, in the cycle $(1, 2, 1, 3, 1, 2, 1, 4)$; queue 1 is polled four times during the cycle, queue 2 is visited twice, while queues 3 and 4 are polled only once. This thesis analyses systems using complex cyclic service.

Two basic approaches are utilized in the analysis of multiqueue polling models. Firstly, solutions may be developed from a purely analytical approach based on multi-dimensional generating functions. This approach is at best difficult and solutions are often not available,

especially for non-exhaustively served systems which is our concern in this thesis.

The second and more useful approach is the use of approximate analyses based on imbedded Markov chains at scan epochs. These techniques usually provide solutions which are quite suitable for engineering purposes and require far less time for calculation. This is the approach we use in our analysis. A survey of the literature indicates that while a lot of attention has been given to multiqueue systems using simple cyclic service [1-19], relatively little work has been done on systems using complex cyclic service [20].

1.3 Thesis Objective

The objective of this thesis is to present the techniques for the performance modelling and analysis of polling systems with complex cyclic service. The analysis is based on a multiqueue model using Poisson batch arrivals, ordinary non-exhaustive service, with general service and walktime distributions. The goal is to obtain approximate, but reasonably accurate, expressions for the queue lengths and waiting time distributions as well as the mean delay in any queue. In addition, algorithms will be developed to quickly compute numerical results for the mean delay.

1.4 Thesis Outline

In Chapter 2, previous work on the modelling and analysis of cyclic polling systems is surveyed. The works

of Kuehn [18] and Krym [19] are described in detail as they provide a base for our analysis.

Chapter 3 presents the analysis of complex cyclic service. This is achieved by breaking up the complex cycle into subcycles, and finding the probability state distribution at scan instants taking into account the dependence between successive subcycles. Expressions for the queue length and waiting time distribution, as well as the mean delay, are obtained for an arbitrary queue. Based on results for simple priority cycles, a general formula for the mean waiting time in an arbitrary queue during an arbitrarily complex cycle is deduced.

In Chapter 4, systems with batch input are investigated by extending the approach developed in Chapter 3.

Chapter 5 describes algorithms for the numerical evaluation of the mean delay expressions given in the previous chapters. These algorithms are therefore constructed to compute the first and second moments of the subcycle times and conditional mean subcycle times as these are among the components of the mean delay formula.

In Chapter 6, some numerical results are generated and validated by computer simulations. The accuracy of the approximate analysis is discussed and factors which affect it are identified.

Finally, the conclusions of this thesis are presented in Chapter 7.

CHAPTER 2

A SURVEY OF PREVIOUS WORK

2.1 Review of Literature

The initial thrust in the analysis of multiqueue systems came from the field of Operations Research [1]. Earliest work in polling systems involved two-queue models using exhaustive service and zero walktime [2,3]. This analysis was later extended to non-zero walktimes [4,5]. Cooper and Murray [6,7] examined a system with an arbitrary number of queues, exhaustive service and zero walktimes. This model was useful in the performance analysis of the #1 ESS and time sharing computer systems.

The common goal of these studies was to obtain exact expressions for the queue length and waiting time distribution. These solutions were intractable and thus approximate techniques that were accurate and relatively easy to compute were sought.

In the early Seventies, considerable attention was devoted to the performance analysis of distributed loop and ring networks that were constructed during this period [9-14]. For example, Kaye [9] analyzed a loop system using fixed-length messages, with each station having a buffer of one message length. Konheim and Meister [14] examined a discrete-time model using non-zero overhead and exhaustive service.

Later, the analysis of token ring LANs using exhaustive service consisted of approximate techniques that incorporated exact formulae developed in earlier studies [15,16]. The results were generally quite favourable.

As previously noted, non-exhaustive service proved more difficult to analyze. An important assumption [8] was therefore made in early attempts to model non-exhaustively served systems. It was assumed that the probability distribution of the number of customers the server meets on arrival to each queue is independent of the distribution in other queues. This assumption of independence simplified the analysis by ignoring the correlation between queues, thus eliminating the need for joint probability state distributions of all the queues in the system.

Using this assumption, Hashida and Ohara [17] obtained the characteristic function of the waiting time distribution for a finite-buffer system using ordinary non-exhaustive service.

A significant contribution to the analysis of multiqueue systems using ordinary non-exhaustive service was made by Kuehn [18]. He introduced the concept of conditional cycles in which the cycle times were dependent on the state of queue i as it is polled during the cycle. Using an imbedded Markov chain analysis, he obtained expressions for the generating functions of the state

characteristic function of the waiting time distribution, and the mean delay.

In his Master's thesis, Krym [19] examined a local area network with constant overhead and fixed length messages. Using a different approach, Krym obtained Kuehn's result for the mean delay in any queue for ordinary non-exhaustive service. Both Krym's and Kuehn's techniques are discussed in detail later in the chapter.

All the systems investigated in the material reviewed above, employed simple cyclic service. However, D. Manfield [20] examined a particular priority polling system using two-way traffic between a central controller and a set of peripheral controllers. Priority was given to the outgoing queue which was served exhaustively, while the incoming queues were served using ordinary non-exhaustive service. Favourable results for the mean delay in all queues were obtained.

2.2 Review of Kuehn's Work [18]

2.2.1 Conditional Cycle Times

Kuehn investigated a multiqueue $M^{[X]}/G/1$ model with batch input, general service and switchover times, operating under simple cyclic service. He first defined π_j as the probability of the server finding at least one customer in queue j on its arrival to queue j . For simple cyclic service, π_j also corresponds to the average number of customers served in queue j during a cycle. The

intuitive result is given by [21]:

$$a_j = \lambda_j c \quad (2.1)$$

This result indicates that the probability of finding at least one customer in queue j is the product of the arrival rate, λ_j , and the average time interval since the queue was polled at the previous scan instant. The state of queue j at a given scan instant therefore is dependent on the cycle time of the previous cycle. Kuehn therefore examined the general cycle time T_{Cj} as shown in Figure 2.2.1.

T_{Cj} is dependent on the state of queue j at the beginning of the cycle. If at least one person is in queue j , a service time H_j occurs. This additional time increases the probability of customers arriving in other queues during the same cycle, thus further increasing the cycle time. So the cycle time is conditioned on the state of queue j at the scan instant ~~beginning~~ the cycle. Let T_{Cj}' be the cycle time if no one is in queue j and T_{Cj}'' , if at least one customer is found in the queue at that scan instant. The corresponding cycles are denoted by C_j' and C_j'' with distributions $C_j'(\cdot)$ and $C_j''(\cdot)$ and transforms $\phi C_j'(s)$ and $\phi C_j''(s)$.

Taking expectations:

$$c(2) = c_0(2) + 2c_0 \sum_{i=1}^g A_i H_i + \sum_{i=1}^g [A_i H_i]^2 + \sum_{i=1}^g \sum_{j \neq i}^g [A_i H_i] \cdot [A_j H_j] \quad (2.48)$$

Since the walktime is independent of the service time for systems we will be looking at,

$$c_0 \sum_{i=1}^g A_i H_i = c_0 \sum_{i=1}^g a_i h_i \quad (2.49)$$

$$[A_i H_i]^2 = A_i^2 \cdot H_i^2 = a_i h_i (2) \quad (2.50)$$

$$[A_i H_i][A_j H_j] = A_i A_j \cdot H_i H_j \quad (2.51)$$

$$A_i A_j = a_{j1} \cdot a_j \quad H_i H_j = h_i \cdot h_j \quad (2.52)$$

$$c(2) = c_0(2) + 2c_0 \sum_{i=1}^g a_i h_i + \sum_{i=1}^g a_i h_i (2) + \sum_{i=1}^g \sum_{j \neq i}^g a_{j1} \cdot a_j \cdot h_i \cdot h_j \quad (2.53)$$

This is the general mean square cycle time obtained from Krym's technique.

2.3.7 Evaluation of Krym's Method

Krym's technique is also approximate as it does not take into account interactions between successive cycles. The advantage of Krym's result, however, is that the mean

The transforms $\phi C_j'(s)$ and $\phi C_j''(s)$ may be written as:

$$\begin{aligned}\phi C_j'(s) &= \prod_{i=1}^g U_{i,i+1}(s) \prod_{i=j}^g (\alpha_{ji}' \phi H_i(s) + [1 - \alpha_{ji}']) \\ \phi C_j''(s) &= \prod_{i=1}^g U_{i,i+1}(s) \prod_{i=j}^g (\alpha_{ji}'' \phi H_i(s) + [1 - \alpha_{ji}'']) \\ &\quad \phi H_j(s) \quad (2.2)\end{aligned}$$

where $\phi U_{i,i+1}(s)$ is the transform of the distribution of walktime from queue i to queue $i+1$.

$\phi H_i(s)$ is the transform of the service time distribution of a customer in queue i .

α_{ji}' is the probability of finding at least one customer in queue i given that no customers are in queue j .

α_{ji}'' is the probability of finding at least one customer in queue i given that there is at least one customer in queue j .

We can write $\phi C(s)$, the transform of T_{Cj} , as the weighted sum of $\phi C_j'(s)$ and $\phi C_j''(s)$ as follows:

$$\phi C(s) = (1 - \alpha_j) \phi C_j'(s) + \alpha_j \phi C_j''(s) \quad (2.3)$$

Similarly,

$$c = (1 - \alpha_j) c_j' + \alpha_j c_j'' \quad (2.4)$$

where c_j' and c_j'' are conditional mean cycle times with

$$c_j' = \frac{c_0}{1 - \rho_0 + \rho_j} \quad (2.5)$$

$$\text{where } \rho_0 = \frac{g}{\sum_{i=1}^g \lambda_i h_i}$$

and is the total utilization of the system.

$$c_j'' = \frac{c_0 + h_j}{1 - \rho_0 + \rho_j} \quad (2.6)$$

In addition, the second moment of the cycle time is given by:

$$c^{(2)} = [1 - \alpha_j][\text{Var}T_{Cj}' + c_j'^2] + \alpha_j [\text{Var}T_{Cj}'' + c_j''^2] \quad (2.7)$$

where

$$\text{Var}T_{Cj}' = \sum_{i=1}^g \text{Var}T_{U_i} + \sum_{i \neq j}^g (\alpha_{ji}' \cdot h_i^{(2)} - \alpha_{ji}'^2 \cdot h_i^2) \quad (2.8)$$

and

$$\begin{aligned} \text{Var}T_{Cj}'' &= \sum_{i=1}^g \text{Var}T_{U_i} + \sum_{i \neq j}^g (\alpha_{ji}'' h_i^{(2)} - \alpha_{ji}''^2 \cdot h_i^2) \\ &\quad + \text{Var}T_{H_j} \end{aligned} \quad (2.9)$$

The variance of the cycle time, $\text{Var}T_C$ is denoted by

$$\text{Var}T_C = [1 - \alpha_j][\text{Var}T_{Cj}' + c_j'^2] + \alpha_j [\text{Var}T_{Cj}'' + c_j''^2] - c^2 \quad (2.10)$$

Thus we can represent the general properties of T_{Cj} in terms of the properties of T_{Cj}' and T_{Cj}'' .

2.2.2 Markov Chain Analysis of the State Distribution At Scan Instants

Let N be the number of customers in the queue at a given scan instant with probability distribution; $p(n) = P\{N=n\}$. Using conditional cycles, we can obtain the state of the queue at the present scan instant, $p(n)$, given that we know the state of the same queue at the previous scan instant, $p(m)$. The transition probability $p(mn)$ is given by:

$$p(mn) = \begin{cases} \int_{t=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} dC'(t) & ; \text{ for } m = 0 \\ \int_{t=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^{n-m+1}}{(n-m+1)!} dC''(t); & \text{ for } m > 0 \end{cases} \quad (2.11)$$

where the subscript j for the queue is suppressed.

This allows us to perform the following Markov chain analysis:

$$p(n) = p(0) \cdot p(0n) + \sum_{m=1}^{n+1} p(m) p(mn) \quad (2.12)$$

$$G(x) = \sum_{n=0}^{\infty} p(n) x^n \quad (2.13)$$

$$G(x) = \frac{p(0) x \phi C'(z) - \phi C''(z)}{x - \phi C''(z)} \quad (2.14)$$

where $z = \lambda[1-x]$

Using l'Hospital's rule in the limit as $x \rightarrow 1$, the following result is obtained:

$$p(0) = \frac{1 - \lambda c^n}{1 - \lambda(c^n - c^1)} \quad (2.15)$$

2.2.3 State Distribution at Departure Instants

We now examine the state distribution as seen by a departing customer as he leaves the system. The corresponding generating function $G^*(x)$ is used to obtain the characteristic function of the waiting time distribution $\phi W(s)$.

Let N^* be the number of customers left behind by a departing customer with distribution $p^*(n) = P\{N^* = n\}$. Let $p(m)$ be the probability of the server finding m customers in the queue at its scan instant. Since departure instants occur when a customer in the queue is served, we must find the probability that the server finds m customers given that the queue is not empty. This is given by:

$$\frac{p(m)}{1 - p(0)} \quad (2.16)$$

The probability that the departing customer sees n customers given that the server finds m customers at the previous scan instant is the probability that $n - m + 1$ arrivals occur during the service time of the departing customer.

Thus,

$$p^*(n) = \sum_{m=1}^{n+1} \frac{p(m)}{1 - p(0)} \cdot \int_{t=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^{n-m+1}}{(n-m+1)!} dH(t) \quad (2.17)$$

$$G^*(x) = \sum_{n=0}^{\infty} p^*(n) x^n \quad (2.18)$$

After some algebraic manipulation

$$G^*(x) = \frac{1}{1-p(0)} \left[\sum_{m=1}^{\infty} p(m) x^{m-1} \phi H(z) \right] \quad (2.19)$$

Thus,

$$G^*(x) = \frac{1}{1-p(0)} \left[\frac{G(x) - p(0)}{x} \right] \cdot \phi H(z) \quad (2.20)$$

2.2.4 Delay Analysis (FIFO Service)

To complete the analysis, we need an expression for $G^*(x)$ in terms of the transform of the waiting time distribution, $\phi W(\cdot)$ so we can equate the two expressions of $G^*(x)$ and thus obtain an expression for $\phi W(\cdot)$. This is done by noting that the number of customers as seen by the departing customer is equal to the number of arrivals during his total time in the system, T_s . This is referred to as the sojourn time and is given by $T_s = T_w + T_H$ where T_w is the waiting time and T_H is the service time, with pdfs $W(\cdot)$ and $H(\cdot)$ respectively. Hence

$$p^*(n) = \int_{t=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} d(W(t) \otimes H(t)) \quad (2.21)$$

where $\otimes$ is the convolution operator.

$$G^*(x) = \sum_{n=0}^{\infty} p^*(n) x^n = \phi W(z) \phi H(z) \quad (2.22)$$

Equating the two expressions for $G^*(x)$, we get

$$\phi W(z) = \frac{1}{1-p(0)} \cdot \left[\frac{G(x) - p(0)}{x} \right] \quad (2.23)$$

Substituting $s=z\lambda[1-x]$ and rearranging,

$$\phi W(s) = \frac{1-\lambda c^n}{c'} \cdot \left[\frac{1-\phi C'(s)}{s-\lambda[1-\phi C^n(s)]} \right] \quad (2.24)$$

$$W = \left. \frac{-d\phi W(s)}{ds} \right|_{s \rightarrow 0} \quad (2.25)$$

Using l'Hospital's Rule, the result is given by:

$$W = \frac{c'(2)}{2c'} + \frac{\lambda c^n(2)}{2(1-\lambda c^n)} \quad (2.26)$$

Kuehn subsequently examined the same model using batch input. This analysis is reviewed in Chapter 4.

2.2.5 Evaluation of Kuehn's Method

Though the Markov chain analysis itself is exact, the approach is approximate as it only partially takes into account the interaction of successive cycles. This is because the state of queue j during the current cycle is dependent on the states of all the queues in the previous cycle not just queue j .

Results obtained from Kuehn's method therefore, generally underestimates the values from simulation runs.

The accuracy decreases with an increasing number of queues and increasing service time variances. It increases, however, with increasing overhead and low traffic.

2.3 Review of Krym's Thesis [19]

2.3.1 Network Model

Krym analyzed an integrated voice/data fibre optic local area network.

Stations on the network are connected via a fibre optic cable which is joined to form a loop or ring. Messages are transported by means of a frame which circulates around the loop. The frame is divided into various fields as shown in Figure 2.3.1. The token is a 1 bit field that may be set to one of two states; busy (1) and free (0). When a station has a message to transmit and detects that the token is free, it changes the token to busy and places its message in the data field with the source and destination addresses contained in the data header.

Since the token is marked busy, other stations may not deposit data in the data field. A station copies the message if the destination address matches its own. The frame therefore returns to the sending station which may either:

- (1) Set the token to free, thus allowing other stations to use the token.

Figure 2.3.1 Frame Format

or

- (2) Keep the token as busy and transmit its next queued message.

The first option corresponds to ordinary non-exhaustive service which is considered in this analysis.

The frame takes a fixed time interval called the frame time, to circulate around the loop, whether it is free or busy. This is denoted by t_f . Thus a station detecting the Sync field going by will, after t_f seconds, recognize the Sync field going by again. This synchronizes all the stations on the loop and the network is said to be synchronous.

2.3.2 Mathematical Model

The analytical model consists of a multiqueue polling system using simple cyclic service and ordinary non-exhaustive service discipline. Because the network is synchronized, the total walktime C_0 and the service time H_1 are both deterministic and equal to the frame time t_f .

Thus we may write the cycle time C as follows:

$$C = t_f + \sum_{i=1}^g A_i t_f \quad (2.27)$$

where g is the number of stations in the system

$$A_i \ (i=1 \dots g) = \begin{cases} 1, & \text{for } n_i \geq 1 \\ 0, & \text{for } n_i = 0 \end{cases} \quad (2.28)$$

n_i is the number of messages in queue i .

The mean cycle time, c , is given by:

$$c = t_f + \sum_{i=1}^g \bar{A}_i t_f \quad (2.29)$$

where $\bar{A}_i = 1.P\{n_i \geq 1\} + 0.P\{n_i = 0\} = \alpha_i$

α_i is the probability that station i has at least one message to transmit (as in Kuehn's analysis),

thus $\alpha_i = \lambda_i c$.

Thus the mean cycle time c is given by:

$$c = \frac{t_f}{1 - t_f \sum_{i=1}^g \lambda_i} \quad (2.30)$$

2.3.3 The Mean Square Cycle Time

This is obtained directly:

$$\begin{aligned} [C]^2 &= \left[t_f + \sum_{i=1}^g A_i t_f \right]^2 \\ &= \left[\sum_{i=1}^g A_i t_f \right]^2 + 2 t_f \sum_{i=1}^g A_i t_f + t_f^2 \end{aligned} \quad (2.31)$$

$$\left[\sum_{i=1}^g A_i t_f \right]^2 = t_f^2 \sum_{i=1}^g A_i^2 + t_f^2 \sum_{i=1}^g \sum_{\substack{j=1 \\ j \neq i}}^g A_i A_j \quad (2.32)$$

Taking expectations,

$$\begin{aligned} c(2) &= t_f^2 \sum_{i=1}^g \overline{A_i^2} + t_f^2 \sum_{i=1}^g \sum_{\substack{j=1 \\ j \neq i}}^g \overline{A_i A_j} \\ &\quad + 2 t_f^2 \sum_{i=1}^g \overline{A_i} + t_f^2 \end{aligned} \quad (2.33)$$

$$\text{but } \overline{A_1^2} = \overline{A_1} = \alpha_1 \quad (2.34)$$

$$\begin{aligned} \overline{A_1 A_j} = & 1.1 \cdot P\{A_1=1, A_j=1\} + 1.0 P\{A_1=1, A_j=0\} \\ & + 0.1 P\{A_1=0, A_j=1\} + 0.0 P\{A_1=0, A_j=0\}. \end{aligned} \quad (2.35)$$

Using conditional probability,

$$\overline{A_1 A_j} = P\{A_1=1, A_j=1\} = P\{A_1=1|A_j=1\} \cdot P\{A_j=1\}. \quad (2.36)$$

The term $P\{A_1=1|A_j=1\}$ is the probability that queue 1 has at least one message to transmit given that a preceding station j transmits a message during the same cycle. This is exactly equal to α_{j1}'' used in Kuehn analysis.

Using our intuitive result,

$$\alpha_{j1}'' = \lambda_j c_j'' \quad (2.37)$$

where c_j'' is the mean cycle time given that at least one message is in queue j .

$$\begin{aligned} \overline{A_1 A_j} = & \alpha_{j1}'' \cdot \alpha_j \\ c^{(2)} = & t_f^2 \sum_{i=1}^G \alpha_i + t_f^2 \sum_{i=1}^G \sum_{\substack{j=1 \\ j \neq i}}^G \alpha_{j1}'' \alpha_j \\ & + 2 t_f^2 \sum_{i=1}^G \alpha_i + t_f^2 \end{aligned} \quad (2.38)$$

2.3.4 Mean Access Delay

The access delay, d_a , is defined as the time interval that begins when a message arrives at a transmit queue and ends as the message is about to be transmitted. This is

equivalent to the waiting time in the general queueing model.

The access delay is composed of two components:

- (a) The residual cycle-time - the time taken for a free token to arrive at the queue, denoted by r .

$$\text{From renewal theory, } \bar{r} = \frac{c^{(2)}}{2c} \quad (2.39)$$

- (b) Once the free token arrives at the queue, messages ahead of the test message are transmitted once every cycle. The cycle time under those conditions is given by T_C'' , the conditional cycle time given that a message has been transmitted.

On the average, an arriving message sees $\bar{n}$ messages in the queue waiting to be transmitted. So the mean delay, $\bar{d}_a$, may be written as:

$$\bar{d}_a = \bar{r} + \bar{n}c'' \quad (2.40)$$

where c'' is the mean conditional cycle time.

From Little's law

$$\bar{n} = \lambda \bar{d}_a \quad (2.41)$$

$$\text{Hence } \bar{d}_a = \frac{\bar{r}}{(1 - \lambda c'')} \quad (2.42)$$

Thus,

$$\bar{d}_a = \frac{c^{(2)}}{2c(1 - \lambda c'')} \quad (2.43)$$

This result can be shown to be exactly equal to Kuehn's solution by expressing the mean residual time, r as:

$$r = \frac{(1-\alpha)c'}{c} \cdot \frac{c'(2)}{2c'} + \frac{\alpha c''}{c} \cdot \frac{c''(2)}{2c''} \quad (2.44)$$

where $\frac{c'(2)}{2c'}$ and $\frac{c''(2)}{2c''}$ are the conditional mean residual time based on whether a C' or C'' cycle is in progress when the message arrives at the queue.

2.3.5 Application of Krym's Technique to a General Queueing Model

Here we extend Krym's results to a general queueing model with general service time distribution H_1 with mean h_1 and a total general walktime C_0 with mean c_0 , so;

$$\text{The mean cycle time, } c = c_0 + \sum_{i=1}^g \overline{A_i H_i} \quad (2.45)$$

$\overline{A_i H_i}$ is the expected value of two independent random variables equal to $\alpha_i h_i$

where $\alpha_i = \lambda_i c$

$$c = \frac{c_0}{1-\rho_0} \quad \text{which is the general mean cycle time formula} \quad (2.46)$$

2.3.6 The Mean Square Cycle Time For The General Queueing Model

$$[C]^2 = \left[c_0 + \sum_{i=1}^g \overline{A_i H_i} \right]^2 \quad (2.47)$$

Taking expectations:

$$c(2) = c_0(2) + 2c_0 \sum_{i=1}^g A_i H_i + \sum_{i=1}^g [A_i H_i]^2 + \sum_{i=1}^g \sum_{j \neq i}^g [A_i H_i] \cdot [A_j H_j] \quad (2.48)$$

Since the walktime is independent of the service time for systems we will be looking at,

$$c_0 \sum_{i=1}^g A_i H_i = c_0 \sum_{i=1}^g a_i h_i \quad (2.49)$$

$$[A_i H_i]^2 = A_i^2 \cdot H_i^2 = a_i h_i (2) \quad (2.50)$$

$$[A_i H_i][A_j H_j] = A_i A_j \cdot H_i H_j \quad (2.51)$$

$$A_i A_j = a_{j1} \cdot a_j \quad H_i H_j = h_i \cdot h_j \quad (2.52)$$

$$c(2) = c_0(2) + 2c_0 \sum_{i=1}^g a_i h_i + \sum_{i=1}^g a_i h_i (2) + \sum_{i=1}^g \sum_{j \neq i}^g a_{j1} \cdot a_j \cdot h_i \cdot h_j \quad (2.53)$$

This is the general mean square cycle time obtained from Krym's technique.

2.3.7 Evaluation of Krym's Method

Krym's technique is also approximate as it does not take into account interactions between successive cycles. The advantage of Krym's result, however, is that the mean

delay formula is in terms of the mean square cycle time, $c^{(2)}$, and the mean cycle time, c , which can be computed from the general equations shown in the previous subsections.

In contrast, Kuehn's method requires the evaluation of conditional mean square cycle times, $c'^{(2)}$ and $c''^{(2)}$, using conditional variances, $\text{VAR } T_C'$ and $\text{VAR } T_C''$, and mean conditional cycle times, c' and c'' . In general therefore, Kuehn's result is more difficult to evaluate numerically. This is significant for complex cyclic service, e.g., for complex cycles divided into n subcycles, 2^n conditional subcycle moments must be calculated for Kuehn's method compared to n subcycle moments for Krym's. Therefore, in subsequent chapters, we will make use of Krym's method to obtain the moments of subcycle times.

CHAPTER 3

ANALYSIS OF COMPLEX CYCLIC SERVICE

3.1 Introduction

In many communications systems, priority service is frequently required for certain classes of users. With polling systems, this can be provided by a complex polling cycle in which higher priority queues are visited more often during the overall polling cycle than lower priority queues. Thus systems with different levels of priority may be polled using complex cycles in which the number of visits to a given queue during the cycle is proportional to its relative priority in the system.

An example of a basic priority cycle (Example 1) is given by (1,2,1,3) where queue 1, the high priority queue, is polled twice per cycle while queues 2 and 3 are low priority queues and are visited only once every cycle. This complex cycle is used to illustrate concepts introduced in this chapter and is referred to as the Elementary Priority Cycle (EPC).

An example of a complex cycle having three levels of priority (Example 2) is denoted by (1,2,1,3,1,2,1,4). In this cycle, queue 1 has the highest priority as it is polled four times in the cycle, queue 2 has the second highest priority level and is visited twice per cycle;

queues 3 and 4 are of lowest priority and are polled only once during the cycle. This cycle and the EPC are members of a general set of complex cycles referred to as the 2^n cycles because the number of times each queue is polled per cycle can be expressed in powers of 2 and there are n priority levels in the cycle. Thus the cycle (1, 2, 1, 3, 1, 2, 1, 4) is called the 2^3 of Harmonic Priority Cycle.

Complex cycles may be subdivided into shorter polling sequences called subcycles. A subcycle is defined as the polling schedule between two successive scan instants of a given queue. For example, the EPC (1,2,1,3) may be broken up into subcycles (1,2) and (1,3) based on the scan instants of queue 1.

Polling cycles with various levels of priority may be subdivided into sets of subcycles based on the scan instants of an arbitrary queue. The 2^3 cycle can be grouped into the following sets $\{(1,2),(1,3),(1,2),(1,4)\}$ and $\{(2,1,3,1),(2,1,4,1)\}$. A given set of subcycles derived from the scan instants of queue 1 are referred to as subcycles relative to queue 1. Thus the set $\{(1,2),(1,3),(1,2),(1,4)\}$ is referred to as subcycles relative to queue 1. These subcycles are expressed as C_{ik} where i is the queue on which the subcycles are based and k is the position of the subcycle within the cycle. For the 2^3 cycle, therefore, subcycles relative to queue 1 are given by C_{11} , C_{12} , C_{13} , C_{14} respectively. For queue 1 polled once every cycle, $C_{1k} = C$, the overall cycle.

The state of the queue at scan instants is denoted by N_{1k} and represents the number of customers that the server finds in queue 1 on its arrival at the start of subcycle C_{1k} . For queues polled once per cycle, N_{1k} becomes N_1 . The total walktime during the subcycle C_{1k} is given by C_{1k0} with mean c_{1k0} .

This chapter begins by examining cycle and subcycle times in a general polling cycle for which formulas for the means and second moments are obtained. Conditional mean subcycle times are also investigated. Expressions are derived for the generating function of the state probabilities at scan and departure instants based on an imbedded Markov chain analysis. Techniques are also developed to obtain solutions for the waiting time distribution and the mean delay. For an arbitrarily complex cycle, a general formula for the mean delay in any queue is proposed.

3.2 Mean Cycle and Subcycle Times

In this section, we examine the mean cycle and subcycle times. The cycle time of a complex cycle is given by T_C with distribution function $C(\cdot)$ and mean c while the subcycle times are denoted by $T_{C_{1k}}$ with (d.f.), $C_{1k}(\cdot)$ and mean c_{1k} . To obtain expected values of cycle and subcycle times, we use the basic result as indicated in the previous chapter.

Recall that the probability of finding at least one

person in queue 1 at a given scan instant is the product of its arrival rate, λ_1 , and the expected value of the time interval from the previous scan instant.

This result is applied to the EPC, as shown in Figure 3.2.1. The probability of finding at least one customer in queue 1 during the C_{11} subcycle,

$$P\{n_{11} \geq 1\} = \lambda_1 c_{12} \quad (3.1)$$

while the probability of finding at least one customer in queue 1 during the C_{12} subcycle,

$$P\{n_{12} \geq 1\} = \lambda_1 c_{11} \quad (3.2)$$

However, the probability of finding at least one customer in queue 2 is given by

$$P\{n_2 \geq 1\} = \lambda_2 c \quad (3.3)$$

$$\text{Similarly, } P\{n_3 \geq 1\} = \lambda_3 c \quad (3.4)$$

The probability of finding at least one customer in queue 1 during subcycle C_{1k} is denoted by a_{1k} .

Hence

$$a_{1k} = \lambda_1 c_1(k-1) \quad (3.5)$$

where $c_1(k-1)$ is the expected value of the previous subcycle time. Thus $(k-1)$ is equal to $(k-1) \bmod m_1$ where m_1 is the number of times queue 1 is visited during the cycle.

Figure 3.2.1 The Elementary Priority Cycle

3.2.1 Mean Cycle Time

Consider an arbitrarily complex cycle in which queue 1 is visited m_1 times during the overall cycle, and the corresponding subcycles relative to queue 1 are given by $C_{11}, \dots, C_{1m_1}$ where

$$\sum_{k=1}^{m_1} C_{1k} = C, \text{ the cycle time} \quad (3.6)$$

The mean cycle time, c , may be expressed as follows:

$$c = c_0 + \sum_{i=1}^G \sum_{k=1}^{m_i} a_{ik} h_i \quad (3.7)$$

But $a_{1k} = \lambda_1 C_{1(k-1)}$ and $\sum_{k=1}^{m_1} C_{1(k-1)} = c$,
the mean cycle time.

$$\text{Hence, } c = c_0 + \rho_0 c \quad (3.8)$$

$$\rho = \frac{c_0}{1 - \rho_0} \quad \text{This is the general formula for the mean cycle time.} \quad (3.9)$$

3.2.2 Mean Subcycle Time

In this subsection, we determine the mean subcycle time of an arbitrary subcycle C_{1k} . Queue j is an arbitrary queue that is polled during C_{1k} . Subcycles of queue j that begin during the C_{1k} subcycle are denoted by C_{j1} as shown in Figure 3.2.2.

Figure 3.2.2 General Subcycle C_{ik}

The probability of finding at least one customer in queue j at the start of a C_{j1} subcycle, a_{j1} , is given by $\lambda_j c_{j(1-1)}$. The set of subcycles of j that begin during C_{ik} is given by $V_j(C_{ik})$ and the set of queues polled during C_{ik} is denoted by $g(C_{ik})$.

The mean subcycle time, c_{ik} , may be written as:

$$c_{ik} = c_{ik0} + \sum_{j \in g(C_{ik})} \rho_j \sum_{l \in V_j(C_{ik})} c_{j(1-1)} \quad (3.10)$$

To illustrate, we apply this result to the EPC (Figure 3.2.1). For easy representation, since queue 1 is the only high priority queue, we suppress the subscript i during the analysis. i.e.,

$$c_{11} = c_1 = c_{10} + (\lambda_1 c_2) h_1 + (\lambda_2 c) h_2 \quad (3.11)$$

$$\text{But } c = c_{11} + c_{12} = c_1 + c_2$$

$$\text{Thus } c_1 = \frac{c_{10} + c_2 (\rho_1 + \rho_2)}{(1 - \rho_2)} \quad (3.12)$$

Similarly,

$$c_2 = c_{20} + (\lambda_1 c_1) h_1 + (\lambda_3 c) h_3 \quad (3.13)$$

$$c_2 = \frac{c_{20} + c_1 (\rho_3 + \rho_1)}{(1 - \rho_3)} \quad (3.14)$$

Thus we have two simultaneous linear equations with unknowns c_1 and c_2 . In general therefore, to obtain the mean subcycle times of an arbitrary queue polled m_1 times during the complex cycle, we must solve m_1 simultaneous

linear equations. Except for small values of m_1 , algebraic techniques will be needed to quickly compute numerical results. Chapter 5 develops algorithms to calculate the mean subcycle times.

3.3 The Mean Square Cycle and Subcycle Times

In this section, we extend Krym's technique to obtain the mean square cycle and subcycle times for complex cyclic service.

3.3.1 Mean Square Cycle Times

The cycle time C may be expressed as

$$C = C_0 + \sum_{i=1}^G \sum_{k=1}^{m_1} A_{ik} H_i \quad (3.15)$$

where A_{ik} is given by
$$\begin{cases} 1 & \text{for } n_{ik} \geq 1 \\ 0 & \text{for } n_{ik} = 0 \end{cases}$$

Squaring and taking expectations

$$\begin{aligned} c^{(2)} = c_0^{(2)} &+ 2c_0 \sum_{i=1}^G \sum_{k=1}^{m_1} \overline{A_{ik}} \overline{H_i} + \sum_{i=1}^G \sum_{k=1}^{m_1} \overline{A_{ik}^2} \overline{H_i^2} \\ &+ \sum_{i=1}^G \sum_{k=1}^{m_1} \sum_{p \neq k}^{m_i} \overline{A_{ip} \cdot A_{ik}} \overline{H_i H_i} \\ &+ \sum_{i=1}^G \sum_{k=1}^{m_1} \sum_{s \neq 1}^G \sum_{t=1}^{m_s} \overline{A_{st} \cdot A_{ik}} \overline{H_i H_s} \end{aligned} \quad (3.16)$$

$\overline{A_{ik}^2} = \overline{A_{ik}} = a_{ik}$, the probability of finding at least one customer in queue i during the k th subcycle.

Similarly,

$$\overline{A_{ip} A_{ik}} = P(A_{ip}=1, A_{ik}=1) = P(A_{ik}=1|A_{ip}=1)P(A_{ip}=1) \quad (3.17)$$

$$P(A_{ik}=1|A_{ip}=1) = \alpha_{ik, ip} \quad (3.18)$$

This is the probability of finding at least one customer in queue i during subcycle k given that at least one customer is in the same queue during subcycle p .

$$\alpha_{ik, ip} = \lambda_1 c_{i(k-1), ip} \quad (3.19)$$

where $c_{i(k-1), ip}$ is the conditional mean subcycle time $c_{i(k-1)}$ conditioned on ip - the server always finding at least one customer in queue i during the p th subcycle.

$$\overline{A_{st} A_{ik}} = P(A_{st}=1, A_{ik}=1) = P(A_{ik}=1|A_{st}=1)P(A_{st}=1) \quad (3.20)$$

$P(A_{ik}=1|A_{st}=1) = \alpha_{ik, st}$. This is the probability that at least one customer is in queue i during subcycle k given that at least one customer is in queue s during subcycle t .

$$\alpha_{ik, st} = \lambda_1 c_{i(k-1), st} \quad (3.21)$$

Finally, all service time distributions are independent of each other. Thus $c^{(2)}$ may be written as follows:

$$c^{(2)} = c_0^{(2)} + 2c_0 \sum_{i=1}^g \sum_{k=1}^{m_1} \alpha_{ik} h_i$$

$$\begin{aligned}
& + \sum_{i=1}^g \sum_{k=1}^{m_i} \alpha_{ik} h_i^{(2)} + \sum_{i=1}^g \sum_{k=1}^{m_i} \sum_{p \neq k}^{m_i} \alpha_{ik,ip} \alpha_{ip} h_i^2 \\
& + \sum_{i=1}^g \sum_{k=1}^{m_i} \sum_{s \neq i}^g \sum_{t=1}^{m_s} \alpha_{ik,st} \alpha_{st} h_i h_s \quad (3.22)
\end{aligned}$$

3.3.2 Mean Square Subcycle Times

Using the same approach in the previous subsection, the subcycle time C_{ik} may be expressed as follows:

$$C_{ik} = C_{ik0} + \sum_{j \in g(C_{ik})} \sum_{l \in V_j(C_{ik})} A_{jl} H_j \quad (3.23)$$

$$\text{where } A_{jl} = \begin{cases} 1 & \text{for } n_{jl} \geq 1 \\ 0 & \text{for } n_{jl} = 0 \end{cases}$$

Squaring and taking expectations, the result is given by:

$$\begin{aligned}
c_{ik}^{(2)} &= c_{ik0}^{(2)} + 2c_{ik0} \sum_{j \in g(C_{ik})} \sum_{l \in V_j(C_{ik})} \alpha_{jl} h_j \\
&+ \sum_{j \in g(C_{ik})} \sum_{l \in V_j(C_{ik})} \alpha_{jl} h_j^{(2)} \\
&+ \sum_{j \in g(C_{ik})} \sum_{l \in V_j(C_{ik})} \sum_{p \neq l} \alpha_{jl,jp} \alpha_{jp} h_j^2 \\
&+ \sum_{j \in g(C_{ik})} \sum_{l \in V_j(C_{ik})} \sum_{s \neq j} \sum_{t \in V_s(C_{ik})} \alpha_{jl,st} \alpha_{st} h_s h_j \quad (3.24)
\end{aligned}$$

3.4 The Conditional Mean Subcycle Times

To obtain the mean square cycle and subcycle times, it is first necessary to calculate the conditional mean subcycle times in Section 3.3. Firstly, we examine conditional mean subcycle times of queue i based on the state of queue i during an arbitrary subcycle p . These conditional mean subcycle times are also important in the Markov chain analysis in Section 3.5.

This approach is then extended to obtain the conditional mean cycle times of queue i based on the state of queue s during subcycle t .

3.4.1 Conditional Mean Subcycle Times of Queue i Based On The State of Queue i During Subcycle p

Consider a general complex cycle, in which queue i is visited m_i times per cycle. We condition the system on the state of queue i during subcycle p . There are two conditional states of interest. The first, denoted by ip' , corresponds to the server finding no customers in queue i during subcycle p . The second, denoted by ip'' , is the server finding at least one customer in queue i during subcycle p . Queue j is an arbitrary queue in subcycle C_{ik} . Subcycles of queue j that begin during the subcycle C_{ik} are denoted by C_{jl} . The system is depicted in Figure 3.4.1. The set of subcycles of queue j that begin during the C_{ik} subcycle is given by $V_j(C_{ik})$ and the set of queues polled in C_{ik} is denoted by $g(C_{ik})$. For the condition ip' , the probability that the server finds at least one

System Conditioned On The State Of
Queue s At This Scan Instant

Figure 3.4.3 General Subcycle C_{ik} Conditioned on the
State of Queue s During Subcycle t

Figure 3.4.2 The Elementary Priority Cycle Conditioned on the State of Queue 1 During the C_{11} Subcycle

Figure 3.4.2 The Elementary Priority Cycle Conditioned on the State of Queue 1 During the C_{11} Subcycle

The conditional subcycles based on no one being in queue 1 during the first subcycle are expressed as C_{11}' and C_{21}' with means c_{11}' and c_{21}' respectively.

Using expressions developed above,

$$c_{11}' = c_{10} + \sum_{j=2} \rho_j \sum_{l=1} c_{j(l-1),11}' \quad (3.30)$$

Since queue 2 is polled once in the cycle,

$$c_{j(l-1),11}' = c_{11}' + c_{21}' \quad (3.31)$$

$$c_{11}' = c_{10} + \rho_2 (c_{11}' + c_{21}')$$

$$c_{11}' = \frac{c_{10} + c_{21}'(\rho_2)}{(1-\rho_2)} \quad (3.32)$$

For the second subcycle, we must take into account the probability of finding at least one person in queue 1 during the second subcycle.

$$c_{21}' = c_{20} + \sum_{j=1,3} \rho_j \sum_{\substack{l=2 \text{ for } j=1 \\ l=1 \text{ for } j=3}} c_{j(l-1),11}' \quad (3.33)$$

For queue 1

$$c_{11,11}' = c_{11}' \quad (3.34)$$

For queue 3

$$c_{31,11}' = c_{11}' + c_{12}' \quad (3.35)$$

$$c_{21}' = \frac{c_{20} + c_{11}'(\rho_3 + \rho_1)}{(1-\rho_3)} \quad (3.36)$$

So the conditional mean cycle times are expressed as two simultaneous linear equations, similar to but not the same as, those for the unconditional mean subcycle times Section 3.2.

Now consider the system conditioned on at least one customer being present in queue 1 during the first subcycle. The conditional subcycles are C_{11}'' and C_{21}'' with conditional mean subcycle times c_{11}'' and c_{21}'' respectively.

Using the same approach,

$$c_{11}'' = c_{10} + b_1 + \rho_2 (c_{11}'' + c_{21}'') \quad (3.37)$$

$$c_{11}'' = \frac{c_{10} + b_1 + c_{21}'' (\rho_2)}{(1-\rho_2)} \quad (3.38)$$

$$\text{and } c_{21}'' = \frac{c_{20} + c_{11}'' (\rho_1 + \rho_3)}{(1-\rho_3)} \quad (3.39)$$

So again, in order to obtain numerical values for c_{11}'' and c_{21}'' two simultaneous linear equations must be solved.

In general, therefore, the conditional mean subcycle times of an arbitrary queue i polled m_i times during the cycle and conditioned on the state of queue i during subcycle p will require that m_i simultaneous equations be solved, to obtain numerical values. Chapter 5 describes techniques to solve these equations.

3.4.2 Conditional Mean Subcycle Times of Queue 1 Based on the State of Queue s During Subcycle t

In this subsection, we extend this analysis to conditioning of the system based on the state of an arbitrary queue s during an arbitrary subcycle t. The states are represented by st' and st'' respectively. The conditional subcycle s of queue 1 are denoted by $C_{1k,st'}$ and $C_{1k,st''}$. Queue j is an arbitrary queue polled during subcycle C_{1k} and subcycles of queue j initiated during the C_{1k} subcycle are denoted by C_{j1} . The set of subcycles of j that began during the C_{1k} subcycle is given by $V_j(C_{1k})$ and the set of queue in the C_{1k} subcycle is given by $g(C_{1k})$. The system is shown in Figure 3.4.3. The analysis is similar to the last subsection, but here we must take into account when $C_{j1} = C_{st}$. For the st' condition, queue s will be ignored during subcycle t while for the st'' condition as service time T_{Hs} occurs at queue s during subcycle t. Therefore the conditional mean subcycle times are given by:

For the st' condition where $C_{j1} \neq C_{st}$

$$C_{1k,st'} = C_{1k0} + \sum_{j \in g(C_{1k})} \rho_j \sum_{1 \in V_j(C_{1k})} C_{j(1-1),st'} \quad (3.40)$$

If $C_{j1} = C_{st}$

$$C_{1k,st'} = C_{1k0} + \sum_{j \in g(C_{1k})} \rho_j \sum_{1 \in V_j(C_{1k})} C_{j(1-1),st'} \quad \text{1st when } j=s \quad (3.41)$$

System Conditioned On The State Of
Queue s At This Scan Instant

Figure 3.4.3 General Subcycle C_{ik} Conditioned on the
State of Queue s During Subcycle t

Similarly, for st^* conditioning

$$C_{j1} \neq C_{st}$$

$$C_{ik,st^*} = C_{ik0} + \sum_{j \in g(C_{ik})} \rho_j \sum_{l \in V_j(C_{ik})} C_{j(1-1),st^*} \quad (3.42)$$

$$C_{j1} = C_{st}$$

$$C_{ik,st^*} = C_{ik0} + \sum_{j \in g(C_{ik})} \rho_j \sum_{l \in V_j(C_{ik})} C_{j(1-1),st^*+h_s} \quad \text{let when } j=s \quad (3.43)$$

Again, it is found that for queue i , polled m_i times per cycle, the conditional mean cycle times will be expressed in terms of m_i simultaneous linear equations. The algorithms for calculating these conditional subcycle times are discussed in Chapter 5.

3.5 Analysis of Complex Cyclic Service

In this section, we analyse complex cyclic service. Expressions for the generating functions at scan and departure instants of the queue during an arbitrary subcycle k , the characteristic function of the waiting time distribution, and the mean delay are derived.

3.5.1 State Distribution of An Arbitrary Queue At Scan Instants

In this subsection, the state distribution of an arbitrary queue i at the k th scan instant is obtained

using an imbedded Markov chain and the state of the queue at the $(k-1)$ th scan instant (Figure 3.5.1), where

$$(k-1) = (k-1) \bmod m_i \quad \text{and}$$

$P_{ik-1}(m)$ = the probability of the server finding m customers in queue i on its arrival to that queue during the $(k-1)$ th subcycle.

$G_{ik-1}(x)$ = the corresponding generating function.

$P_{ik}(n)$ = the probability of the server finding n customers in queue i on its arrival to the queue during the k th subcycle.

$G_{ik}(x)$ = the corresponding generating function.

For easy representation, the subscript i is suppressed in the following analysis, i.e.,

$$P_{ik-1}(m) \rightarrow P_{k-1}(m); \quad G_{ik-1}(x) \rightarrow G_{k-1}(x)$$

$$P_{ik}(n) \rightarrow P_k(n); \quad G_{ik}(x) \rightarrow G_k(x)$$

The probabilities $P_{k-1}(m)$ and $P_k(n)$ will be represented by $p(m)$ and $p(n)$ except when distinguishing between specific values of m and n .

The following Markov chain analysis is exact, but as stated in Section 2.2.5, the use of the Markov chain is approximate because it only partially takes into account the interaction between successive subcycles. The transition probability $p(mn)$ is given by:

Figure 3.5.1 The State Distribution at the $(k-1)^{\text{th}}$ and k^{th} Scan Instants of Queue i for Complex Cyclic Service

$$p(mn) = \begin{cases} \int_{t=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} dC_{(k-1)(k-1)}'(t) & \text{for } m = 0 \\ \int_{t=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^{n-m+1}}{(n-m+1)!} dC_{(k-1)(k-1)}''(t) & \text{for } m > 0 \end{cases} \quad (3.44)$$

where $C_{(k-1)(k-1)}'$ and $C_{(k-1)(k-1)}''$ are the conditional subcycles of queue 1 based on the state of queue 1 during the $(k-1)$ th subcycle.

$$p(n) = p_{k-1}(0) p(0n) + \sum_{m=1}^{n+1} p(m) p(mn) \quad (3.45)$$

$$G_k(x) = \sum_{n=0}^{\infty} p(n) x^n \quad (3.46)$$

Using algebraic manipulation,

$$G_k(x) = p_{k-1}(0) \phi C_{(k-1)(k-1)}'(z) + \sum_{m=1}^{\infty} p(m) x^{m-1} \phi C_{(k-1)(k-1)}''(z) \quad (3.47)$$

But

$$\sum_{m=1}^{\infty} p(m) x^{m-1} = \frac{1}{x} [G_{k-1}(x) - p_{k-1}(0)] \quad (3.48)$$

Thus in general,

$$G_k(x) = p_{k-1}(0) \phi C_{(k-1)(k-1)}'(z) + \frac{1}{x} [G_{k-1}(x) - p_{k-1}(0)] \phi C_{(k-1)(k-1)}''(z) \quad (3.49)$$

where $k-1 = (k-1) \bmod m_1$

For $m_1 = 1$

$$G(x) = p(0) \phi C'(z) + \frac{1}{x} [G(x) - p(0)] \phi C''(z)$$

this is Kuehn's result [18] (3.50)

For $m_1 = 2$

$$G_1(x) = p_2(0) \phi C_{22}'(z) + \frac{1}{x} [G_2(x) - p_2(0)] \phi C_{22}''(z) \quad (3.51)$$

$$G_2(x) = p_1(0) \phi C_{11}'(z) + \frac{1}{x} [G_1(x) - p_1(0)] \phi C_{11}''(z) \quad (3.52)$$

It can be readily seen, therefore, that for queue 1 polled m_1 times in the cycle, the generating functions of the state distribution ; $G_1(x) \dots, G_{m_1}(x)$ are expressed as m_1 simultaneous equations.

3.5.2 The State Distribution at Departure Instants During An Arbitrary Subcycle

In this subsection, we examine, the state distribution as seen by a departing customer leaving the queue during an arbitrary subcycle k . We wish to obtain the generating function $G_k^*(x)$, as this is needed to obtain the expression for the waiting time distribution function.

Let $p_k(m)$ be the probability that server finds m customers in queue 1 on its arrival to the queue, at the start of the k th subcycle. The probability of the

server finding at least one person in the queue during the k th subcycle $= 1 - p_k(0)$, and the probability of finding exactly m customers given that the queue is not empty is

$$\frac{p_k(m)}{1 - p_k(0)} \quad (3.53)$$

Let $p_k^*(n)$ be the distribution as seen by a departing customer leaving during the k th subcycle.

$$p_k^*(n) = \sum_{m=1}^{n+1} \frac{p_k(m)}{1 - p_k(0)} \cdot \int_{t=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^{n-m+1}}{(n-m+1)!} dH(t) \quad (3.54)$$

$$G_k^*(x) = \sum_{n=0}^{\infty} p_k^*(n) x^n \quad (3.55)$$

This yields after some rearranging

$$G_k^*(x) = \frac{1}{1 - p_k(0)} \left[\frac{[G_k(x) - p_k(0)]}{x} \right] \cdot \phi H(z) \quad (3.56)$$

3.5.3 The General Expression for the Distribution at Departure Instants

The distribution seen by a departing customer during an arbitrary subcycle k , $p_k^*(n)$, will generally be different for different subcycles. However, to obtain $\phi W(s)$ we need $G^*(x)$ which is the unconditioned distribution as seen by a departing customer. This is independent of the subcycle during which he leaves. We must therefore work out the probability of an arbitrary customer leaving

D

the queue during an arbitrary subcycle k , given by, p_{ck} , under all conditions of the system. Hence,

$$G^*(x) = \sum_{k=1}^{m_1} p_{ck} G_{ik}^*(x) \quad (3.56)$$

We now demonstrate an approximate approach of obtaining p_{ck} . Firstly, we define a busy period with respect to queue i as the time interval that begins when the server finds at least one person in the queue given that there was no one in the queue at the previous scan instant, and ends when the server again finds no customers in the queue. The subcycles that occur during this period are referred to as busy subcycles.

Since a customer must be served during a busy subcycle these subcycles will generally be larger than subcycles in which the customer finds no one in the queue. Thus there is a greater probability of a customer arriving during a busy period than an idle one. This new customer will generate another busy subcycle thus extending the busy period.

For a given arrival rate λ_i , there is a greater probability of customer arriving during long busy periods than short ones. Thus most customers are likely to arrive when a long busy period is in progress. These customers will leave the system during successive subcycles and therefore an equal amount will leave the queue during different subcycles. Under these conditions, $G^*(x)$ can be represented by an equally weighted sum of the generating functions $G_{i1}^*(x) \dots G_{im}^*(x)$, i.e.,

$$G^*(x) = \frac{1}{m_1} \sum_{k=1}^{m_1} G_{ik}^*(x) \quad (3.57)$$

Clearly, this equation is more accurate for higher values of λ_1 , the arrival rate to queue 1, as under these conditions long busy periods over a large number of busy subcycles will occur. At lower values of λ_1 , we have busy periods that, on the average, will be only a few busy subcycles long and hence an equal number of customers may not leave during the different subcycles. The equation therefore, may be less accurate under these conditions. We shall assume that this equation is accurate throughout the range of conditions we will examine*.

3.5.4 Delay Analysis (FIFO Service)

The distribution as seen by a departing customer considering his sojourn time in the system:

$$p_n^* = \int_0^\infty e^{-\lambda t} \frac{(\lambda t)^n}{n!} d(W(t) \odot H(t)) \quad (3.58)$$

$$\text{with generating function } G^*(x) = \phi W(z) \phi H(z) \quad (3.59)$$

This distribution is independent of the subcycle in progress when the departing customer leaves the system.

Equating the two expressions for $G^*(x)$

$$\phi W(z) = \frac{1}{m_1} \sum_{k=1}^{m_1} \frac{1}{1-p_k(0)} \cdot \left[\frac{G_k(x) - p_k(0)}{x} \right] \quad (3.60)$$

* In Section 6.1, we see that in fact, this assumption does not significantly affect accuracy of low loads.

The mean waiting time w for queue 1 is given by:

$$w = - \lim_{s \rightarrow 0} \frac{d\phi W(s)}{ds} \quad \text{where } s = z - [1-x] \quad (3.61)$$

3.6 Results

For $m_1=1$, i.e., the queue is polled only once per cycle, we have Krym's and Kuehn's result [18,19]

$$w = \frac{c(2)}{2(1-c^2)} \cdot \frac{1}{c} \quad (3.62)$$

To obtain results for $m_1=2$, we examine the high priority queue of the EPC.

$$G_1(x) = p_2(0) \cdot \phi C_{22}'(z) + \frac{1}{x} [G_2(x) - p_2(0)] \phi C_{22}''(z) \quad (3.63)$$

$$G_2(x) = p_1(0) \cdot \phi C_{11}'(z) + \frac{1}{x} [G_1(x) - p_1(0)] \phi C_{11}''(z) \quad (3.64)$$

Substituting and rearranging,

$$G_1(x) = \frac{x p_2(0) [x \phi C_{22}'(z) - \phi C_{22}''(z)] + p_1(0) \phi C_{22}''(z) [x \phi C_{11}'(z) - \phi C_{11}''(z)]}{[x^2 - \phi C_{11}''(z) \phi C_{22}''(z)]} \quad (3.65)$$

$$G_2(x) = \frac{x p_1(0) [x \phi C_{11}'(z) - \phi C_{11}''(z)] + p_2(0) \phi C_{11}''(z) [x \phi C_{22}'(z) - \phi C_{22}''(z)]}{[x^2 - \phi C_{11}''(z) \phi C_{22}''(z)]} \quad (3.66)$$

$$\phi W(z) = \frac{1}{2} \left[\frac{1}{p_1(0)} \left[\frac{G_1(x) - p_1(0)}{x} \right] + \frac{1}{1 - p_2(0)} \left[\frac{G_2(x) - p_2(0)}{x} \right] \right] \quad (3.67)$$

$$\text{Let } \phi W_1(z) = \frac{1}{1-p_1(0)} \left[\frac{G_1(x) - p_1(0)}{x} \right] \quad (3.68)$$

$$\phi W_2(z) = \frac{1}{1-p_2(0)} \left[\frac{G_2(x) - p_2(0)}{x} \right] \quad (3.69)$$

$$w_1 = \frac{d\phi W_1(s)}{ds} \Big|_{s \rightarrow 0} \quad w_2 = \frac{d\phi W_2(s)}{ds} \Big|_{s \rightarrow 0} \quad (3.70)$$

$$w = \frac{1}{2} [w_1 + w_2] \quad (3.71)$$

$$\phi W_1(z) = \frac{p_2(0) [x \phi C_{22}'(z) - \phi C_{22}''(z)] + p_2(0) [x \phi C_{22}''(z) \phi C_{11}'(z) - x]}{[1-p_1(0)] [x^2 - \phi C_{11}''(z) \phi C_{22}''(z)]} \quad (3.72)$$

$$\phi W_2(z) = \frac{p_1(0) [x \phi C_{11}'(z) - \phi C_{11}''(z)] + p_2(0) [x \phi C_{11}''(z) \phi C_{22}'(z) - x]}{[1-p_2(0)] [x^2 - \phi C_{11}''(z) \phi C_{22}''(z)]} \quad (3.73)$$

Substituting $s=z=\lambda[1-x]$

Differentiating and using l'Hospital's Rule,

$$w_1 = \frac{-d\phi W_1(s)}{ds} \Big|_{s \rightarrow 0} = \left[\frac{c_1^{(2)} + c_2^{(2)}}{2[2-\lambda[c_{11}'' + c_{22}'']]} \right] \cdot \frac{1}{c_2} \quad (3.74)$$

$$w_2 = \left[\frac{c_1^{(2)} + c_2^{(2)}}{2[2-\lambda[c_{11}'' + c_{22}'']]} \right] \cdot \frac{1}{c_1} \quad (3.75)$$

For $n_1=2$

$$w = \left[\frac{c_1^{(2)} + c_2^{(2)}}{4[2-\lambda[c_{11}'' + c_{22}'']]} \right] \cdot \left[\frac{1}{c_1} + \frac{1}{c_2} \right] \quad (3.76)$$

Similarly, we can also show that for $n_1=3$

$$w = \left[\frac{c_1^{(2)} + c_2^{(2)} + c_3^{(3)}}{6[\lambda^{-1}(c_{11}'' + c_{22}'' + c_{33}'')] } \right] \left[\frac{1}{c_1} + \frac{1}{c_2} + \frac{1}{c_3} \right] \quad (3.77)$$

3.6.1 General Formula

Based on these results, a general formula for the mean delay in an arbitrary queue i which is polled m_i times in an arbitrarily complex cycle may be deduced. It is given by:

$$w_i = \frac{\sum_{k=1}^{m_i} c_k^{(2)}}{2m_i \left[m_i - 1 \left[\sum_{k=1}^{m_i} c_{kk}'' \right] \right]} \cdot \left[\sum_{k=1}^{m_i} \frac{1}{c_k} \right] \quad (3.78)$$

where $c_k^{(2)}$, c_{kk}'' and c_k can be calculated from expressions developed in this chapter.

CHAPTER 4

ANALYSIS OF COMPLEX CYCLIC SERVICE WITH BATCH INPUT

4.1 Introduction

Under certain conditions, some data communication switching systems are modelled using queueing models with batch input. For example, in packet switching a maximum packet size is imposed on many networks. Alternatively, some networks use fixed-length packets throughout the system. In both cases, messages longer than the imposed constraint are broken up into several packets which are transmitted individually across the network. This process is therefore represented as batch input to the queueing model.

This chapter examines cyclic queueing systems using complex cyclic service, with batch input. We first summarize Kuehn's results for systems using simple cyclic service. Expressions for the generating function of the state probabilities, the characteristic function of waiting time distribution and the mean delay, are derived. For an arbitrary complex priority cycle a general formula for the mean delay is deduced.

4.2 Simple Cyclic Service With Batch Input

This section reviews Kuehn's work on simple cyclic service with batch input and is used as a starting point in the analysis of complex cyclic service:

4.2.1 The Batch Arrival Process

Packets (which we shall also refer to as customers) arrive at a particular queue in batches of size K with distribution

$$q(k) = P\{K=k\} \quad (4.1)$$

The probability generating function is given by

$$Q(x) = \sum_{k=0}^{\infty} q(k) x^k \quad (4.2)$$

The arrival of batches in time instant $(0, t)$ is represented by the Poisson distributed denoted by

$$P\{N_B(t)=n\} = \frac{(Bt)^n}{n!} e^{-Bt} \quad (4.3)$$

with generating function $G_B(x) = e^{-Bt(1-x)}$ (4.4)

Let $N_A(t)$ be the total number of customers arriving in the queue between time interval $(0, t)$. We define

$$G_A(x) = \sum_{k=0}^{\infty} P\{N_A(t)=k\} x^k = e^{-Bt[1-Q(x)]} \quad (4.5)$$

4.2.2 State Distribution at Scan Instants

Using an embedded Markov chain analysis at the scan instant to the queue

$$p(n) = p(0) p(0n) + \sum_{m=1}^{n+1} p(m) p(mn) \quad (4.6)$$

where $p(n)$ is the distribution of the customers in the queue for the current scan instant; $p(m)$ the distribution for the previous scan instant. $p(mn)$ is given by:

$$p(mn) = \begin{cases} \int_{t=0^-}^{\infty} \sum_{v=0}^{\infty} P\{N_B(t)=v\} \cdot P\{N_A(t)=n|N_B(t)=v\} \cdot dC'(t); & \text{for } m=0 \\ \int_{t=0^-}^{\infty} \sum_{v=0}^{\infty} P\{N_B(t)=v\} \cdot P\{N_A(t)=n-m+1|N_B(t)=v\} \cdot dC''(t); & \text{for } m>0 \end{cases} \quad (4.7)$$

$$G(x) = \sum_{n=0}^{\infty} p(n)x^n \quad (4.8)$$

Incorporating results from the previous section, with techniques introduced in Chapter 2, $G(x)$ may be represented as:

$$G(x) = p(0) \frac{[x\phi C'(z) - \phi C''(z)]}{[x - \phi C''(z)]} \quad \text{where } z = \lambda_B[1-Q(x)] \quad (4.9)$$

This is identical to the results for M/G/1 case in Chapter 2, but the argument z is different to take into account the batch arrivals.

4.2.3 State Distribution At Departure Instants

Given the distribution of the queue at the scan instants, denoted by $p(m)$, the distribution as seen by a departing customer, $p^*(n)$ is expressed as

$$p^*(n) = \sum_{m=1}^{n+1} \frac{p(m)}{1-p(0)} \int_0^\infty P\{N_B(t)=v\} P\{N_A(t)=n-m+1\} N_B(t)=v \cdot dH(t) \quad (4.10)$$

$$G^*(x) = \sum_{n=0}^{\infty} p^*(n) x^n \quad (4.11)$$

Substituting $s=n-m+1$ and rearranging

$$G^*(x) = \frac{1}{1-p(0)} \cdot \left[\frac{G(x)-p(0)}{x} \right] \cdot \phi H(z) \quad (4.12)$$

Again the result is identical to the result for the M/G/1 case but with $z = \phi[1-Q(x)]$.

4.2.4 Delay Analysis (FIFO Service)

In order to obtain the characteristic function of the waiting time distribution, $\phi W(s)$, an alternative way of expression $G^*(x)$, which includes $\phi W(\cdot)$ must be found. As in the M/G/1 case, $G^*(x)$ is also the generating function of the probability distribution of the number of arriving customers during the sojourn time of an arbitrary customer (known as test customer) in the system. For the batch case, therefore, N^* , the number of customers in the queue as seen by the departing test customer is given by:

$$N^* = N_1^* + N_2^* \quad (4.13)$$

where N_1^* = the no. of customers, in the same batch as the test customer that are yet to be served when the test customer leaves the system.

N_2^* = the no. of customers that have arrived in subsequent batches.

Let $r_n = P\{N_1^* = n\}$ be the probability distribution of n customers in the same batch as the test customers but left behind as the test customer departs.

The probability that the test customer arrived in a particular batch of size k , denoted by $q^*(k)$ should be proportioned to the batch size k , as well as the relative occurrence of such sizes. Thus

$$q^*(k) = \frac{kq(k)}{EK} \quad (4.14)$$

the test customer may, with probability $1/k$, be in any position in the batch. To leave n customers behind however, he must be in the $(k-n)$ th position of the batch.

Thus

$P\{\text{test customer in the } (k-n)\text{th position of a batch}$

$$\text{size}\} = \frac{q(k)}{EK} \quad (4.15)$$

To obtain r_n , we sum for batch sizes from $k=n+1$ to ∞ ;

$$r_n = \sum_{k=n+1}^{\infty} \frac{q(k)}{EK} \quad (4.16)$$

$$R(x) = \sum_{n=0}^{\infty} r(n)x^n \quad (4.17)$$

Rearranging, the result is given by:

$$R(x) = \left[\frac{1-Q(x)}{1-x} \right] \cdot \frac{1}{EK} \quad (4.18)$$

$R(x)$, therefore is the generating function of the probability distribution of the number of customers in the same batch as the test customers that are still in the queue as the test customer departs. Thus

$$r_b = \lim_{x \rightarrow 1} \left[\frac{dR(x)}{dx} \right]_{x=1} \quad \text{is the expected value of } N_1^*.$$

Using l'Hospital's rule,

$$r_b = \frac{1}{2} \left[\frac{EK^2}{EK} - 1 \right] \quad (4.19)$$

(It can also be shown that $R(x)$ is the generating function of the probability distribution of the number of customers in the same batch ahead of the test customer and r_b is the expected value).

Having obtained $R(x)$ we derive $G^*(x)$ as follows

$$p^*(n) = \int_{t=0^-}^{\infty} \left[\sum_{\nu=0}^n P\{N_1^* = \nu\} \sum_{\nu=0}^{\infty} P\{N_2^*(t) = n-\nu | N_B(t) = \nu\} \cdot P\{N_B(t) = \nu\} \right] \cdot d(W(t) \otimes H(t)) \quad (4.20)$$

$$G^*(x) = \sum_{n=0}^{\infty} p^*(n) x^n \quad (4.21)$$

This simplifies to

$$G^*(x) = R(x) \int_{t=0}^{\infty} e^{-\lambda_B t [1-Q(x)]} \cdot d(W(t) \otimes H(t)) \quad (4.22)$$

$$G^*(x) = R(x) \cdot \phi W(z) \phi H(z) \\ \text{where } z = \lambda_B [1-Q(x)] \quad (4.23)$$

We now equate our two (2) alternative expressions for $G^*(x)$ and obtain

$$\phi W(z) = \frac{1}{R(x)} \cdot \left[\frac{1}{1-p(0)} \right] \cdot \left[\frac{G(x)-p(0)}{x} \right] \quad (4.24)$$

By substitution, the characteristic function $\phi W(s)$ is

$$\phi W(s) = \frac{1-\lambda c^w}{c'} \cdot \left[\frac{1-\phi C'(s)}{\phi C^w(s)-x} \right] \cdot \left[\frac{1-x}{s} \right] \quad (4.25)$$

The mean delay is found by differentiating $\phi W(s)$, and using l'Hospital's rule. The result is given by:

$$w = \frac{c'(2)}{2c'} + \frac{\lambda c^w(2)}{2(1-\lambda c^w)} + \frac{c^w}{2(1-\lambda c^w)} \cdot \left[\frac{EK^2}{EK} - 1 \right] \quad (4.26)$$

For the analysis of arbitrarily complex cycles, it is necessary to modify the expression of the mean delay as follows

(a) Use Krym's result for the M/G/1 waiting time.

$$(b) \text{ Substitute } r_b = \frac{1}{2} \cdot \left[\frac{EK^2}{EK} - 1 \right]$$

(c) $1/\lambda$ = the mean interarrival time of the customers
to the queue = a

So equation (4.26) may be written as

$$W = \frac{c(2)}{2c(1-\lambda c^m)} + r_b \cdot a \left[\frac{1}{1-\lambda c^m} - 1 \right] \quad (4.27)$$

4.3 Analysis of Complex Cyclic Service With Batch Input

In this section we examine complex cyclic service with batch input. Building on the approach of the previous section, and techniques demonstrated in Chapter 3, the generating functions at the scan and departure instants of queue 1 during an arbitrary subcycle k , the characteristic function of the waiting time distribution, as well as the mean delay, are obtained.

4.3.1 State Distribution of An Arbitrary Queue At Scan Instants

Here, we obtain the state distribution of an arbitrary queue 1 at the k th scan instant using an imbedded Markov chain. Consider the system as shown in Figure 4.3.1, where $(k-1) = (k-1) \bmod m_i$ and

$P_{1k-1}(m)$ = the probability of the server finding m customers in queue 1 on its arrival to that queue during the $(k-1)$ th scan instant.

$G_{1k-1}(x)$ = the corresponding generating function.

$p_{1k}(n)$ = the probability of the server finding n customers in queue 1 on its arrival during the k th scan instant.

Figure 4.3.1 The State Distribution at the $(k-1)^{th}$ and k^{th} Scan Instants of Queue i for Complex Cyclic Service With Batch Input

$G_{1k}(n)$ = the corresponding generating function.

As in the previous chapter;

$$p_{1k-1}(m) \rightarrow p_{k-1}(m); \quad G_{1k-1}(x) \rightarrow G_{k-1}(x)$$

$$p_{1k}(n) \rightarrow p_k(n); \quad G_{1k}(x) \rightarrow G_k(x)$$

$p_{k-1}(m) \rightarrow p(m)$ and $p_k(n) \rightarrow p(n)$ except when distinguishing between specific values of m and n .

The transition probability, $p(mn)$ is given by:

$$p(mn) = \begin{cases} \int_{t=0}^{\infty} \sum_{v=0}^{\infty} p\{N_B(t)=v\} \cdot p\{N_A(t)=n|N_B(t)=v\} \cdot dC_{(k-1)(k-1)}'(t); & \text{for } m = 0 \\ \int_{t=0}^{\infty} \sum_{v=0}^{\infty} p\{N_B(t)=v\} p\{N_A(t)=n-m+1|N_B(t)=v\} dC_{(k-1)(k-1)}''(t); & \text{for } m > 0 \end{cases} \quad (4.28)$$

$$p(n) = p_{k-1}(0) p(0n) + \sum_{m=1}^{n+1} p(m) p(mn) \quad (4.29)$$

$$G_k(x) = \sum_{n=0}^{\infty} p(n) x^n \quad (4.30)$$

Using some techniques from Kuehn's analysis, we obtain the general result:

$$G_k(x) = p_{k-1}(0) \phi_{C_{(k-1)(k-1)}}'(z) + \frac{1}{x} [G_{k-1}(x) - p_{k-1}(0)] \phi_{C_{(k-1)(k-1)}}''(z) \\ \text{where } z = \lambda_B[1-Q(x)] \quad (4.31)$$

So again for queue 1 polled m_1 times per cycle, the generating functions of the state distribution at scan instants, $G_{11} \dots G_{1m_1}$ is expressed in terms of m_1 simultaneous expressions.

4.3.2 State Distribution at Departure Instants During An Arbitrary Subcycle

Using the same analytical approach developed in Section 3.5, the generating function of the state distribution of queue 1 as seen by a departing customer during an arbitrary subcycle k is given by:

$$G_k^*(x) = \frac{1}{1-p_k(0)} \left[\frac{G_k(x) - p_k(0)}{x} \right] \phi H(z) \quad (4.32)$$

Again, we obtain the unconditioned generating function, $G^*(x)$ as seen by a departing customer leaving the system by equally weighting the generating functions $G_1^*(x) \dots G_{m_1}^*(x)$, i.e.,

$$G^*(x) = \frac{1}{m_1} \sum_{k=1}^{m_1} \frac{1}{1-p_k(0)} \left[\frac{G_k(x) - p(0)}{x} \right] \cdot \phi H(z) \quad (4.33)$$

4.3.3 Delay Analysis (FIFO Service)

From Section 4.2

$$G^*(x) = R(x) \phi W(z) \phi H(z) \quad (4.34)$$

$$\text{where } z = \lambda_B [1 - Q(x)], \quad R(x) = \frac{1}{EK} \left[\frac{1 - Q(x)}{1-x} \right]$$

Equating the two expressions for $G_k(x)$

$$\phi W(z) = \frac{1}{m_1} \sum_{k=1}^{m_1} \frac{1}{1-p_k(0)} \left[\frac{G_k(x) - p_k(0)}{x} \right] \cdot \frac{1}{R(x)} \quad (4.35)$$

Substituting $s = z = \lambda_B(1-Q(x))$ and differentiating the mean delay is given by:

$$W = - \frac{d\phi W(s)}{ds} \quad s \rightarrow 0 \quad (4.36)$$

4.4 Results

For $m_1 = 1$

$$W = \frac{c(2)}{2c[1-\lambda c^2]} + r_b a \left[\frac{1}{1-\lambda c^2} - 1 \right] \quad (4.37)$$

For $m_1 = 2$

$$W = \left[\frac{c_1(2) + c_2(2)}{4[2-\lambda(c_{11}'' + c_{22}'')]} \right] \left[\frac{1}{c_1} + \frac{1}{c_2} \right] + r_b a \left[\frac{c}{2[2-\lambda(c_{11}'' + c_{22}'')]} \left[\frac{1}{c_1} + \frac{1}{c_2} \right] - 1 \right] \quad (4.38)$$

For $m_1 = 3$

$$W = \left[\frac{c_1(2) + c_2(2) + c_3(2)}{6[3-\lambda(c_{11}'' + c_{22}'' + c_{33}'')]} \right] \left[\frac{1}{c_1} + \frac{1}{c_2} + \frac{1}{c_3} \right] + r_b a \left[\frac{c}{3[3-\lambda(c_{11}'' + c_{22}'' + c_{33}'')]} \left[\frac{1}{c_1} + \frac{1}{c_2} + \frac{1}{c_3} \right] - 1 \right] \quad (4.39)$$

4.4.1 General Formula

Based on these results, a general formula for the mean waiting time in queue i which is polled m_i times per cycle is deduced. It is expressed as follows:

$$w_i = \frac{\sum_{k=1}^{m_i} c_k}{2m_i \left[m_i - 1 \left[\sum_{k=1}^{m_i} c_{kk} \right] \right]} \cdot \left[\sum_{k=1}^{m_i} \frac{1}{c_k} \right] \quad (2)$$

$$+ r_b a \left[\frac{c}{m_i - 1 \left[\sum_{k=1}^{m_i} c_{kk} \right]} \left[\sum_{k=1}^{m_i} \frac{1}{c_k} \right] - 1 \right] \quad (4.40)$$

CHAPTER 5

PRESENTATION OF ALGORITHMS TO COMPUTE THE MEAN DELAY

5.1 Introduction

This chapter describes algorithms to compute the mean delay based on the general formulas derived in Chapters 3 and 4. To this end, techniques are developed to calculate the first and second moments of the subcycle times, as well as for the conditional mean subcycle times using expressions from Chapter 3 where the conditional and unconditional mean subcycle times were represented by simultaneous linear equations.

Starting with the basic elements of the system, algorithms are constructed to generate these equations and solve them by the Gauss-Seidel iteration technique. With these results, the second moment of the subcycles are then calculated and subsequently the mean delay in every queue is obtained.

5.2 System Characteristics

Parameters required to compute the components of the mean delay formula include:

- g - the total number of queues in the system

A_1

- the arrival rate of customers to queue 1. For batch arrivals, B_1 , the batch arrival rate as well as the first and second moments of the batch size EK_1 and EK_1^2 , are required.

$E_1, h_1^{(2)}$

- the first and second moments of the service time T_{H1} .

$u_{xy}, u_{xy}^{(2)}$

- the first and second moments of the walktime.

$(q_1, \dots, q_d)$

- the complex cycle being employed.

5.3 Subcycle Times

5.3.1 Introduction

To highlight some problems that arise in trying to obtain the mean subcycle times, we consider the 2^3 cycle as shown in Figure 5.3.1.

From Chapter 3, equation (3.10), recall the general expression for the mean subcycle time is given by:

$$C_{1k} = C_{1k0} + \sum_{j \in g(C_{1k})} \frac{p_j}{0} \sum_{l \in V_j(C_{1k})} C_{j(l-1)} \quad (5.1)$$

where j is an arbitrary queue polled during the C_{1k} subcycle and subcycles of j that begin during C_{1k} are denoted by C_{jl} . $g(C_{1k})$ is the set of queues polled during the C_{1k} subcycle and $V_j(C_{1k})$ is the set of subcycles of queue j that begin during subcycle C_{1k} . $C_{j(l-1)}$ is the

Figure 5.3.1 The 2^3 or Harmonic Priority Cycle

mean subcycle time of the previous subcycle $c_{j(1-1)}$.

Using this expression we may write down the mean subcycle times of queue 1 in the 2^3 cycle as:

$$c_{11} = c_{110} + p_1 c_{14} + p_2 c_{22} \quad (5.2)$$

$$c_{12} = c_{120} + p_1 c_{11} + p_3 c_{33} \quad (5.3)$$

$$c_{13} = c_{130} + p_1 c_{12} + p_2 c_{21} \quad (5.4)$$

$$c_{14} = c_{140} + p_1 c_{13} + p_4 c_{44} \quad (5.5)$$

These four equations contain six unknowns, namely: c_{11} , c_{12} , c_{13} , c_{14} , c_{21} and c_{22} where c_{21} and c_{22} are the mean subcycle times of queue 2.

In general, to solve the simultaneous equations for the mean subcycle times, it is necessary to express the mean subcycle times of each queue in terms of the mean subcycle time of a reference queue. This reference queue is chose to be the highest priority queue (HPQ), as it is polled most frequently in the cycle. Hence, the subcycle times of the HPQ are shorter than the subcycle times of other queues and this allows us to express their mean subcycle times in terms of the mean subcycle times of the HPQ.

To express the mean subcycle times of a given queue in terms of the mean subcycle times of the HPQ, we first break up the complex cycle into subcycles of the given queue and the HPQ queue. To illustrate the technique we will express the mean subcycle c_{21} in terms of the mean

subcycle times of queue 1 for the 2³ cycle. See Figure 5.3.2.

The C_{21} subcycle begins during the C_{11} subcycle and ends during the C_{13} subcycle. It also includes C_{12} . We therefore try to express the parts of the C_{11} and C_{13} subcycles that contribute to the mean subcycle time of C_{21} . For C_{11} , this is equal to the mean subcycle time c_{11} less the average time the server spends in the queues and corresponding walktimes before the C_{21} subcycle begins. By inspection, this is expressed as:

$$c_{11} - \rho_1 c_{14} - u_{12} \quad (5.7)$$

where $\rho_1 c_{14}$ is average time the server spends in queue 1 during the C_{11} subcycle and u_{12} is the mean walktime between queues 1 and 2.

Conversely, the part of C_{13} that contributes to c_{21} is the average time the server spends in the queues and walktimes before the C_{21} ends. This is denoted by:

$$\rho_1 c_{12} + u_{12} \quad (5.8)$$

where $\rho_1 c_{12}$ is the average time the server spends in queue 1 during the C_{13} subcycle. So the mean subcycle time, C_{21} , is expressed as

$$c_{21} = c_{11} - \rho_1 c_{14} - u_{12} + c_{12} + \rho_1 c_{12} + u_{12} \quad (5.9)$$

Using the same approach,

Figure 5.3.2 Subcycle C_{21} of the Harmonic Priority Cycle

$$c_{22} = c_{13} - .1c_{12} - u_{12} + c_{14} + .1c_{14} + u_{12} \quad (5.10)$$

Thus, in general, to obtain the expression for the mean subcycle time c_{j1} in terms of the subcycles of the HPQ, denoted by C_{HPQk} , we first examine the complex cycles to see in which subcycles of the HPQ, the subcycle C_{j1} , starts and terminates. Let us represent these subcycles as C_{HPQb} and C_{HPQe} respectively.

Subcycles of the HPQ that are included in C_{j1} are given by C_{HPQ1} . Thus to obtain c_{j1} we express the components of C_{HPQb} and C_{HPQe} that contribute to the mean subcycle time of the c_{j1} subcycle and add the result to the mean subcycle times of the C_{HPQ1} subcycles.

For priority cycles, like the 2^3 cycle, we may write down the components of c_{21} and c_{22} by inspection. This becomes increasingly difficult as the cycle becomes more complex. It is important, therefore, to automate this process in order to quickly compute the mean delay. To do this we must first extract the necessary structural information from the input data in Section 5.2.

5.3.2 Information Gathering Routines

This subsection shows how the structural information for the complex cycle is obtained to calculate the mean subcycle times.

Step 1 -- The first routine checks the complex cycle to determine how many times each queue is polled during the cycle. This is represented by the array X where $X(i)$ is

the number of times queue 1 is polled. Thus for the 2^3 cycle

$$X = (4, 2, 1, 1)$$

Step 2 -- The second algorithm takes the array X and works out the priority level of all the queues based on the number of times each queue is polled during the cycle. This is expressed by the matrix B of dimensions (4×2) , where the first column contains the number of times the queue is polled and the second column contains the queue index. Thus $B(1,2)$ represents the highest priority queue, HPQ, and $B(1,1)$ is the number of times the queue is polled. Again for the 2^3 cycle

$$B = \begin{bmatrix} 4 & 1 \\ 2 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}$$

Step 3 -- Using $B(1,1)$ and $B(1,2)$ we divide the complex cycle into subcycles of the HPQ. This is given by another matrix E , where $E(z,b)$ is the queue in position b of subcycle z of the HPQ. Thus we respect to the 2^3 cycle:

		Positions	
		1	2
$E =$	C_{11}	1	2
	C_{12}	1	3
	C_{13}	1	2
	C_{14}	1	4

In a similar manner, all the average walktimes in the overall cycle is grouped together based on the subcycles of the HPQ. This is represented by the matrix SK where $SK(z,b)$ is the mean walktime between queues $E(z,b)$ and $E(z,b+1)$ in subcycle z of the HPQ, or if b is the last position in the subcycle z , then $SK(z,b)$ represents the walktime from $E(z,b)$ to the HPQ in the $z+1$ th subcycle. For the 2^3 cycle SK is given by:

$$SK = \begin{array}{c|cc} & \text{Positions} & & \\ & 1 & 2 & \\ \hline C_{11} & u_{12} & u_{21} & \\ C_{12} & u_{13} & u_{31} & \\ C_{13} & u_{12} & u_{21} & \\ C_{14} & u_{14} & u_{41} & \end{array}$$

The array CYC is the expected value of the total mean walktime in subcycle a of the HPQ. Thus for the 2^3 cycle:

$$CYC = \begin{array}{c|c} C_{11} & [u_{12} + u_{21}] \\ C_{12} & [u_{13} + u_{31}] \\ C_{13} & [u_{12} + u_{21}] \\ C_{14} & [u_{14} + u_{41}] \end{array}$$

We also construct another matrix using the E matrix called HAV, where $HAV(i,z)$ indicates the position of queue i in subcycle z . With respect to the 2^3 cycle:

		Subcycles			
Queues		C ₁₁	C ₁₂	C ₁₃	C ₁₄
HAV =	1	1	1	1	1
	2	2	0	2	0
	3	0	2	0	0
	4	0	0	0	2

Note that since queue 1 is the highest priority queue then its row elements are all 1's.

Step 4 -- We now proceed to build another matrix H. This matrix indicates how many times each queue is polled in each subcycle of the HPQ. So for the 2³ cycle

		Subcycles			
Queues		C ₁₁	C ₁₂	C ₁₃	C ₁₄
H =	1	1	1	1	1
	2	1	0	1	0
	3	0	1	0	0
	4	0	0	0	1

By definition, the row corresponding to the HPQ must all be 1's.

Step 5 -- Using E and H a 3-dimensional matrix L is constructed to indicate the relative positions of each queue in all the subcycles by determining which queues are polled before and after the queue in a given subcycle. Let us denote an arbitrary subcycle of the HPQ by z. Using H, all possible pair combinations of queues, denoted by (x,y) are examined to determine if queues x and y are

both polled during subcycle z . If not, the element $L(x,y,z)$ is set to zero. However, if both queues are polled during the subcycle, the matrix E is then checked to detect which queue is polled ahead of the other. If queue x is polled ahead of queue y , then $L(x,y,z) = 1$ and $L(y,x,z) = -1$. Conversely, if queue y is polled if queue y is polled before queue x , then $L(y,x,z) = 1$ and $L(x,y,z) = -1$.

Thus, the matrix L is a coded representation of the positions of the queues in all the subcycles of the HPQ. For a given subcycle, all queues ahead of a certain queue will have corresponding elements of value -1 , while all queues behind this queue in the subcycle will have corresponding elements of $(+)1$.

Consider the second subcycle C_{12} of the 2^3 cycle. The relevant elements of matrix L are given by

Queues	Queues			
	1	2	3	4
1	0	0	1	0
2	0	0	0	0
3	-1	0	0	0
4	0	0	0	0

Looking across the first row, queue 1 sees that queue 3 is the only other queue in the subcycle and also that it is ahead of queue 3, because the element involved has a value of 1. Queue 3 also deduces the same information by examining its corresponding elements. For queues 2 and 4

2 2
OF / DE

which are not polled during this subcycle, all the relevant elements are zero.

5.3.3 The Coefficient Matrix

From the previous subsection we now have enough information to assemble the coefficient matrix of the linear equations for the mean subcycle times of the HPQ. For a given subcycle of the HPQ, the coefficients of the linear equation are obtained by considering the contribution of each queue to the mean subcycle time and summing the result over all the contributions of the queues in the subcycle.

The contribution of each queue to the mean subcycle time is given by the average time the server spends in the queue during the subcycle. Thus for a subcycle of HPQ, C_{HPQk} , the contribution of an arbitrary queue j is given by:

$$\rho_j \sum_{1 \in V_j(C_{HPQk})} c_{j(1-1)} \quad (5.11)$$

From our previous section, $c_{j(1-1)}$ can be expressed in terms of the mean subcycle times of the HPQ. We can represent this expression for $c_{j(1-1)}$ in terms of a coefficient matrix of the mean subcycle times of the HPQ such that when we multiply the corresponding elements by ρ_j we obtain the contribution of queue j to the mean subcycle time of C_{HPQk} .

To illustrate this concept we again use the 2^3 subcycle. For queue 1, which is the HPQ, the contribution of queue 1 to subcycle C_{1k} is given by $\rho_1 c_{1(k-1)}$. Since queue 1 is the HPQ, the contribution of queue 1 will already be expressed as mean subcycle times of the HPQ.

We can therefore represent the coefficient arrangement for queue 1 as a two-dimensional structure shown.

$$\begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{cccc}
 C_{11} & C_{12} & C_{13} & C_{14} \\
 \begin{bmatrix}
 C_{11} & 0 & 0 & 0 & 1 \\
 C_{12} & 1 & 0 & 0 & 0 \\
 C_{13} & 0 & 1 & 0 & 0 \\
 C_{14} & 0 & 0 & 1 & 0
 \end{bmatrix}
 \end{array}$$

The contribution of queue 1 to the mean subcycle time of the C_{11} subcycle is given by $\rho_1 c_{14}$. In general, this particular pattern will be similar for the HPQ of any complex polling cycle.

The relationships for all the queues can therefore be represented by three-dimensional matrix T of size $(g, \#_{HPQ}, \#_{HPQ})$ where the $\#_{HPQ}$, from our previous notation, is the number of times the highest priority queue is polled in the cycle.

We now look at the coefficients of queue 2 which is of lower priority than queue 1. To obtain the coefficients for queue 2 during subcycle C_{13} , we use the same basic approach outlined in Section 5.3.1. The procedure is as follows:

Step 1 -- The subcycles of the HPQ in which the C_{21} subcycle starts and terminates are found using the H matrix. The result is stored temporarily in the vector V, where the V(1) contains the subcycle of the HPQ on which C_{21} starts and V(2), the subcycle of the HPQ in which C_{21} ends. Thus for C_{21} , $V(1) = 1$ and $V(2) = 3$.

Step 2 -- The corresponding elements of T for queue 2 in subcycle C_{13} that represent the subcycles of the HPQ that are included in the C_{21} subcycle, i.e. C_{12} ; as well as the subcycle in which C_{21} begins, i.e. C_{11} , are set to one. Thus $T(2,3,1)$ and $T(2,3,2)$ are set to one.

Step 3 -- Using the L matrix we find out all the queues polled before queue 2, in the C_{11} subcycle (in this case queue 1). To obtain the average time the server spends in queue 1 during the C_{11} subcycle, we consult the T matrix for queue 1 during the C_{11} subcycle and multiply the corresponding elements $T(1,1,1)$ through $T(1,1,4)$ by ρ_1 . This gives us the contribution of queue 1 during the C_{11} subcycle. The result for each element multiplication is subtracted from the corresponding elements for queue 2 during the C_{13} subcycle, i.e. $T(2,3,1)$ through $T(2,3,4)$ as the contribution of queue 1 to the C_{11} subcycle is not part of C_{21} .

Step 4 -- From the L matrix, we now work out the queues ahead of queue 2 in the C_{13} subcycle (again this is queue 1). Thus we take elements $T(1,3,1)$ through $T(1,3,4)$ and multiply them by ρ_1 , the result for each element is added

to the corresponding elements of $T(2,3,1)$ through $T(2,3,4)$.

Thus the elements $T(2,3,1)$ through $T(2,3,4)$ now contain the coefficients of the equation for c_{21} in terms of the mean subcycle times of the HPQ (queue 1).

Using the same technique on the C_{22} subcycle, the coefficients for queue 2 are given by:

$$\begin{array}{c} C_{11} \quad C_{12} \quad C_{13} \quad C_{14} \\ \begin{array}{c} C_{11} \\ C_{12} \\ C_{13} \\ C_{14} \end{array} \begin{bmatrix} 0 & -\rho_1 & 1 & 1+\rho_1 \\ 0 & 0 & 0 & 0 \\ 1 & (1+\rho_1) & 0 & -\rho_1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

For queues which are polled one per cycle, we represent the corresponding mean cycle time is represented in terms of the sum of the mean subcycle times of the HPQ. Thus for queue 3 in the 2^3 subcycle we have

$$\begin{array}{c} C_{11} \quad C_{12} \quad C_{13} \quad C_{14} \\ \begin{array}{c} C_{11} \\ C_{12} \\ C_{13} \\ C_{14} \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

The same approach is used for queue 4.

We now multiply the elements of queue 1 in the T matrix by ρ_1 and sum over all the queues to get an $m_{HPQ} \times m_{HPQ}$ matrix. This matrix is then subtracted from the identity matrix to yield the coefficient matrix.

5.3.4 Walktime Contributions

From Section 5.3.2, we recall the equation (5.9) for c_{21}

$$c_{21} = c_{11} = 11 c_{14} - u_{12} + 11 c_{12} + u_{12} \quad (5.11)$$

where u_{12} is the mean walktime between queues 1 and 2.

These walktime contributions must be placed on the right hand side of the simultaneous equations as they are constants. We, therefore, build a matrix WLKT where the net walktime contribution of queue x to the mean subcycle time of subcycle z of the HPQ, is given by $c_x \cdot \text{WLKT}(x,z)$. To illustrate how the value of $\text{WLKT}(x,z)$ is obtained we again use the contribution of queue 2 during the C_{13} subcycle as an example. The procedure is outlined as follows:

Step 1 -- Using the L matrix, the queues in positions ahead of queue 2 in the C_{11} and C_{13} subcycles are found.

Step 2 -- We now use the HAV matrix from Section 5.3.2 to determine the position of the queues within the respective subcycles. Recall that $\text{HAV}(i,z)$ indicates the position of queue i in subcycle z of the HPQ queue. The mean walktime from queue i to the next queue in subcycle z is given by the element $\text{SK}(z, \text{HAV}(i,z))$ in the SK matrix, also introduced in Section 5.3.2. Thus for queue 1 in the C_{11} subcycle, $\text{HAV}(1,1)=1$ i.e., (since queue 1 is the HPQ it will always occur in the first position). We therefore access the $\text{SK}(1,1)$ which contains the mean walktime to

queue 2, i.e. u_{12} . This value, $SK(1,1)$, is subtracted from $WLKT(2,3)$, which is the running sum of the walktime contribution of queue 2 during C_{13} , because it is not included in the C_{21} subcycle.

Step 3 -- For queue 1 in the C_{13} subcycle, we access $HAV(1,3)$ and subsequently $SK(3,1)$ to obtain the mean walktime from queue 1 to queue 2 during the C_{13} subcycle. We now add this to the running sum contained in $WLKT(2,3)$ as this included in the C_{21} subcycle.

Step 4 -- We now consider the walktime contributions of queues ahead of queue 2 in C_{11} and C_{13} respectively. For queue 1 in the C_{11} subcycle, the walktime components $WLKT(1,1)$ is multiplied by ρ_1 . The result is the net walktime contribution of queue 1 to the mean subcycle time, C_{11} . Since this is not included in the C_{21} subcycle, it is subtracted from $WLKT(2,3)$.

Step 5 -- For queue 1 in subcycle C_{13} we multiply the element $WLKT(1,3)$ by ρ_1 and add the result to $WLKT(2,3)$.

The sum of all the contributions will be zero for queue 2 in subcycle C_{13} . This is due to the particular polling sequence of the 2^3 cycle. Thus in general, the value of $WLKT(x,z)$ is dependent on the relative positions of the queues during the different subcycles.

5.3.5 The Right Hand Column Vector

Having obtained the $WLKT$ matrix, we now proceed to determine the right hand column vector of the simultaneous

linear equations. This is contained in the array RH, where $RH(z)$ is the constant for the linear equation for the mean subcycle time of subcycle z .

Elements of RH will have components of the CYC array and the WLKT matrix. Firstly, we obtain all the net walktime contributions of all the queues in a given subcycle. Thus for each subcycle z , we multiply each element $WLKT(x,z)$ by p_x and sum over all the queues in subcycle z . We now add the result to $CYC(z)$ to obtain $RH(z)$.

5.3.6 The Linear Equations

The simultaneous linear equations are represented by the co-efficient matrix BC. The right hand column vector represented by the RH array and the array SUBCYC containing the values of the mean subcycle time, which are the unknown variables. For the 2^3 cycle, $RH=CYC$ because all the walktimes contributions sum to zero. So the simultaneous equations is given by Figure 5.3.3.

The Gauss-Siedel routine is now used to solve for the mean subcycle times.

5.4 The Conditional Mean Subcycle Times

In this section, we discuss the algorithms required to obtain the conditional mean subcycle times of the HPQ, conditioned on the server finding at least one customer in the arbitrary queue i during the subcycle C_{1k} . From these

$$\begin{array}{c}
 \text{BC} \\
 \left[\begin{array}{cccc}
 1 & \rho_1 \rho_2 & -\rho_2 & -[\rho_1 + \rho_2 + \rho_1 \rho_2] \\
 -[\rho_1 + \rho_3] & [1 - \rho_3] & -\rho_3 & -\rho_3 \\
 -\rho_2 & -[\rho_1 + \rho_2 + \rho_2 \rho_1] & 1 & \rho_1 \rho_2 \\
 -\rho_4 & -\rho_4 & -[\rho_1 + \rho_4] & [1 - \rho_4]
 \end{array} \right]
 \end{array}
 \begin{array}{c}
 \text{SUBCYC} \\
 \left[\begin{array}{c}
 c_{11} \\
 c_{12} \\
 c_{13} \\
 c_{14}
 \end{array} \right]
 \end{array}
 =
 \begin{array}{c}
 \text{RH} \\
 \left[\begin{array}{c}
 c_{110} \\
 c_{120} \\
 c_{130} \\
 c_{140}
 \end{array} \right]
 \end{array}$$

Figure 5.3.3 Simultaneous Linear Equations for the Mean Subcycle Times of the HPQ of the Harmonic Priority Cycle

results, we obtain the corresponding conditional mean subcycle times of all other queues in the system as these values are needed to calculate the second moments of the subcycle times. The basic approach is the same as that of the previous section but additional routines must be set up to take into account the conditioning of the system.

To illustrate these concepts, we consider the Harmonic Priority cycle conditioned on at least one customer being in queue 1 during the C_{11} subcycle as shown in Figure 5.4.1. Since queue 1 is also the HPQ, we wish to obtain the conditional mean subcycle times of queue 1 based on the condition of queue 1 in subcycle p. From equations (3.28) and (3.29), the general expression is given by

for $k \neq p$

$$c_{kp}'' = c_{1k0} + \sum_{j \in G(C_{1k})} \rho_j \sum_{l \in V_j(C_{1k})} c_{j(1-1),1p}'' \quad (5.12)$$

for $k=p$

$$c_{pp}'' = c_{1p0} + \sum_{j \in G(C_{1k})} \rho_j \sum_{l \in V_j(C_{1k})} c_{j(1-1),1p}'' + h_1 \quad (5.13)$$

In addition, we must express the conditional mean subcycle times of the other queues in terms of the HPQ, queue 1. Thus we may represent the conditional mean subcycle times by the following equations.

Figure 5.4.1 The Harmonic Priority Cycle Conditioned on at Least One Customer in Queue 1 During the First Subcycle

$$c_{11}'' = c_{110} + h_1 + \mu_2 (c_{31}'' - \mu_1 c_{21}'' + c_{41}'' + h_1) \quad (5.14)$$

$$c_{21}'' = c_{120} + \mu_1 c_{11}'' + \mu_3 (c_{11}'' + c_{21}'' + c_{31}'' + c_{41}'') \quad (5.15)$$

$$c_{31}'' = c_{130} + \mu_1 c_{21}'' + \mu_2 (c_{11}'' - h_1 + \mu_1 c_{21}'' + c_{21}'') \quad (5.16)$$

$$c_{41}'' = c_{140} + \mu_1 c_{31}'' + \mu_4 (c_{11}'' + c_{21}'' + c_{31}'' + c_{41}'') \quad (5.17)$$

From equation (5.14), the average time the server spends in queue 1 during the first subcycle is given by h_1 , the average service time, and is independent of the conditional mean subcycle times. We must therefore modify the T matrix to reflect this condition of independence. This is done by setting the corresponding elements for queue 1 during subcycle C_{11} to zero. Thus the elements of queue 1 are given by

	C_{11}	C_{12}	C_{13}	C_{14}
C_{11}	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	0	0	0
C_{12}		1	0	0
C_{13}		0	1	0
C_{14}		0	0	1

Using the same procedure as in the previous section, we now proceed to build the conditional coefficient matrix.

The conditioning of the system also introduces the numerical constants h_1 and $\rho_2 h_1$. These values must appear on the right hand side of the conditional mean subcycle time equations. To this end, we set up another matrix called the CUT matrix.

The contribution of queue 1 to the conditional mean subcycle time of subcycle z is determined as follows:

- (a) For the system conditioned on a given queue 1 in subcycle z of the HPQ the corresponding CUT element, $CUT(1,z)$ is given by h_1 . This represents the total contribution of queue 1 to the conditional mean subcycle time.
- (b) For the system not conditioned on a given queue 1 in subcycle z , the contribution of queue 1 to the conditional mean subcycle time of z is given by $\rho_1 * CUT(1,z)$.

To demonstrate, we shall obtain the corresponding CUT element for queue 2 during the C_{13} subcycle, $CUT(2,3)$, with the system conditioned on at least one person being in queue 1 during the C_{11} subcycle. The procedure is as follows:

Step 1 -- Using the L matrix, queues ahead of queue 2 in the subcycle C_{11} and C_{13} (in both cases, queue 1) are determined.

Step 2 -- Since the system is conditioned on queue 1 during the C_{11} subcycle, we therefore subtract $CUT(1,1)$ from the element $CUT(2,3)$ as this is the total contribution of queue 1 to the C_{11} subcycle and is not

part of the C_{21} subcycle.

Step 3 -- For queue 1 in the C_{31} subcycle, we multiply the CUT elements by ρ_1 and add the result to $CUT(2,3)$ as this contributes to the C_{21} subcycle.

To obtain the right hand column vector for the conditional mean subcycle times we multiply the elements of queue 1 in the CUT matrix, except the element on which the system is conditioned, by ρ_1 and sum the result over all the queue in each subcycle. This is then added to the corresponding RH array to yield the right hand column vector of the conditional simultaneous linear equations. Thus for the 2^3 cycle, conditioned at at least one customer being in queue 1 during the C_{11} subcycle the simultaneous linear equations are shown in Figure 5.4.2.

The Gauss-Siedel method is now used to solve for the conditional mean subcycle times.

5.5 The Second Moment of the Subcycle Time

From equation (3.24), the general expression for the second moment of the subcycle time is given by:

$$\begin{aligned}
 c_{ik}^{(2)} = & c_{ik0}^{(2)} + 2c_{ik0} \sum_{j \in G(C_{ik})} \sum_{l \in V_j(C_{ik})} a_{jl} h_j \\
 & + \sum_{j \in G(C_{ik})} \sum_{l \in V_j(C_{ik})} a_{jl} h_j^{(2)} \\
 & + \sum_{j \in G(C_{ik})} \sum_{l \in V_j(C_{ik})} \sum_{p \neq l} a_{jl,lp} a_{lp} h_j^2
 \end{aligned}$$

$$\begin{bmatrix}
 1 & -\rho_1 & -\rho_2 & -\rho_3 \\
 -[c_1 + c_3] & [1 - c_3] & -c_3 & -\rho_3 \\
 -\rho_2 & -[c_1 + c_2 + \rho_1 c_2] & 1 & 0 \\
 -\rho_4 & -\rho_4 & -[\rho_1 + \rho_4] & [1 - \rho_4]
 \end{bmatrix}
 \begin{bmatrix}
 c_{11}'' \\
 c_{21}'' \\
 c_{31}'' \\
 c_{41}''
 \end{bmatrix}
 =
 \begin{bmatrix}
 c_{110} + h_1 + \rho_2 h_1 \\
 c_{120} \\
 c_{130} - \rho_2 h_1 \\
 c_{140}
 \end{bmatrix}$$

Figure 5.4.2 Simultaneous Linear Equations for the Conditional Mean Subcycle Times of the HPQ of the Harmonic Priority Cycle Conditioned On At Least One Customer Being in Queue 1 During the C_{11} Subcycle

$$+ \sum_{j \in G(C_{1k})} \sum_{l \in V_j(C_{1k})} \sum_{s \neq j} \sum_{t \in V_s(C_{1k})} a_{jl, st}'' = a_{st} h_s h_j \quad (5.14)$$

where $a_{jl} = \lambda_j c_{j(1-1)}$

$$a_{jl, ip}'' = \lambda_j (c_{j(1-1)}, ip)''$$

$$a_{jl, st}'' = \lambda_j c_{j(1-1), st}''$$

The mean subcycle time $c_{j(1-1)}$ can be calculated using algorithms from Section 5.3, while $c_{j(1-1), ip}''$ and $c_{j(1-1), st}''$, the conditional mean subcycle times are calculated using techniques from the previous section.

The remaining parameters are obtained directly from the input data. Thus both the conditional and unconditional terms are summed to yield the second moment of the subcycle time.

At this point, therefore, we have demonstrated how to obtain all the numerical values which are needed to calculate the mean delay in any queue.

Figure 6.1.3

Figure C.1.1

Figure 6.1.2

Figure 6.1.3

are shown in Figure 6.1.4.

Results for batch arrivals are given for the Elementary Priority Cycle, EPC, with deterministic service time distribution and a mean batch size of 4 in Figure 6.1.5. Both geometric and constant batch size distributions are shown.

Our final graph, Figure 6.1.6, depicts the Harmonic Priority Cycle with deterministic service times but under asymmetric loading. The loads are proportional to the amount of visits each queue receives in the complex cycle. Thus queue 1 has half the total load, queue 2 has one quarter, while queues 3 and 4 carry one eighth of the total traffic.

6.2 Discussion

Generally, the numerical results from our approximate analysis underestimate those from computer simulations. This is because while the solution for the mean subcycle times are exact, the technique generally underestimates the variance of the subcycle times. Thus, the accuracy decreases as the variance of the subcycle time increases.

This observation was also made by Keuhn [18] in his analysis of simple cyclic service. Since the variance of the subcycle times is directly related to the service time distribution, the accuracy will decrease as we go from deterministic to exponential distributions, as seen in Figures 6.1.1, and 6.1.2 or Figures 6.1.3 and 6.1.4. It

Figure 6.1.4

Figure 6.1.5

Figure 6.1.6

was also found that the results for the higher priority queue was generally better than lower priority queues. This also concurs with results obtained earlier [20].

It was also observed that for all systems, the accuracy decreased as the load increased. This indicates that the subcycle variance is also dependent on the arrival process. Generally, the variance of the subcycle will increase as we move from low to heavy utilization of the system. The relationship, however, is dependent on other factors including load distribution, and the complex cycle being used.

From Figures 6.1.1 to 6.1.4, it can be clearly seen that the accuracy decreases as we go from the EPC to the Harmonic Priority cycle. This is due to an increase in the number of queues and more importantly an increase in the number of subcycles. As the number of subcycles increases the correlation between the different subcycle, will increase. This is especially true for complex cycles with various levels of priority like the 2^3 cycle. Thus, as the number of subcycles and priority levels increase, the accuracy of our result will decrease, as these interactions are not taken into account in our analysis.

For batch arrivals, Figure 6.1.5, good results were obtained for both the geometric and constant batch sizes. Thus the accuracy of our expression for the mean delay for batch arrivals is independent of the type of batch size distribution being employed. This is because in both the constant and geometric distributions that were used, the

waiting time component due to the batch arrival process, is much greater than the M/G/1 component. Thus any inaccuracy due to the M/G/1 component of the waiting time is minimized. Hence, the batch results were very favourable. Kuehn [18] also obtained similar findings.

The load distribution is an important factor in determining the performance of a given polling system. For example, a system using complex cycles in which the highest priority queue is heavily loaded cannot guarantee that messages in the highest priority queue will have the lowest delay. This stands in contrast to standard priority systems, in which highest priority queues can monopolize the network completely, shutting out all queues of lower priority, and thus giving the highest priority queue, the lowest delay.

We now compare complex cycles using asymmetrical load Figure 6.1.6 to symmetrical load Figure 6.1.3. For the asymmetrical systems, loads are proportional to the number of times each queue is polled in the cycle. The mean delay in the high priority queues were lower for the symmetrical load but the mean delays in the lowest priority queues were higher. Thus, both the complex cycle and the load distribution must be jointly considered when evaluating the performance of a polling system.

6.3 Scope for Further Work

Extensions for the work in this thesis are:

- Correlations between subcycles to get better estimates of the subcycle variance.
- Limiting non-exhaustive service where more than one customer could be served at each visit to the queue.
- Exhaustive service.
- Combinations of exhaustive and non-exhaustive service discipline.

CHAPTER 7

CONCLUSIONS

In this thesis, the approximate analysis of priority polling systems using complex cycles has been presented. This was accomplished by introducing the concept of subcycle and deriving expressions for the first and second moments of the subcycle times.

General formulas for the mean delay have been demonstrated. The expressions can be applied to any complex cycle although numerical calculation may be difficult for extremely complex or irregular cycles.

It was found that the accuracy of the approximate analysis decreased with increased traffic, increased variance of the service time distributions, increased number of queues, and greater number of subcycles and priority levels in the complex cycle. However, results within the normal operating range of 30-50% utilization of the communications medium have been found to be within 10% of the exact value and thus are suitable for practical engineering purposes.

REFERENCES

- [1] C. Mack, T. Murphy, and N.L. Webb, "The Efficiency of N Machines Unidirectionally Patrolled By One Operative When Walking and Repair Time are Constants", J. Royal Stat. Soc., Ser.B, 19, pp.166-172 (1957).
- [2] L. Takacs, "Two Queues Attended By A Single Server", Operations Research, Vol.16, No.3, pp.639-650, 1968.
- [3] B. Avi-Itzhak, W.L. Maxwell and L.W. Miller, "Queues With Alternating Priorities", J. Oper. Res. Soc. America, Vol.13, No.2, pp.306-318, 1965.
- [4] J.S. Sykes, "Simplified Analysis of An Alternating Priority Queueing Model With SetUp Time", Operations Res., Vol.18, pp.399-413, 1970.
- [5] M.A. Eisenberg, "Two Queues With Change Over Times", Operations Res., Vol.19, pp.386-401, 1971.
- [6] R.B. Cooper and G. Murray, "Queues Served In Cyclic Order", BSTJ, Vol.48, 675-689, March 1969.
- [7] R.B. Cooper, "Queues Served In Cyclic Order: Waiting Times", BSTJ, Vol.49, pp.399-413, March 1970.
- [8] M.A. Leibowitz, "An Approximate Method for Treating A Class of Multiqueue Problems", IBM Journal, pp.204-209, July 1961.
- [9] A.R. Kaye, "Analysis of A Distributed Control Loop for Data Transmission", Proc. Symp. Computer Communications Networks and Teletraffic, Polytech. Inst. Brooklyn, Brooklyn, N.Y., April 4-6, 1972.
- [10] J.F. Hayes and D.N. Sherman, "Traffic Analysis of a Ring Switched Data Transmission System", BSTJ, Vol.50, pp.2947-2978, November 1971.
- [11] J.D. Spragins, "Analysis of Loop Transmission Systems", Proc. ACM/IEEE 2nd Symp. on Problems in the Optimization of Data Comm. Systems, Palo Alto, pp.175-182, October 1971.
- [12] M.L.T. Yuen, B.A. Black, E.E. Newhall and A.N. Venetsanopoulos, "Traffic Flow In A Distributed Loop Switching System", Proc. Symp. Comp. Comm., Networks and Teletraffic, (New York Polytechnic Press, 1972).

- [13] J.D. Spragins, "Loops Used for Data Collection", Proc. Symp. Computer Comm. Networks and Teletraffic, (New York Polytechnic Press, 1972).
- [14] Konheim and Meister, "Waiting Lines and Times In A System With Polling", J. ACM, Vol.21, No.3, pp.470-490, July 1974.
- [15] W. Bux, "Local Area Subnetworks: A Performance Comparison", IEEE Trans. on Comm., COM-29, No.10, pp.1465-1473, October 1981.
- [16] W. Bux and H.L. Truong, "Token Ring Performance: Mean Delay Approximation", Proceedings of the 10th ITC (Montreal) Canada, North Holland Publ. Company, Amsterdam, pp.3.1-3/1-3.1-3/6, 1983.
- [17] O. Hashida and K. Ohara, "Line Accomodation Capacity On A Communications Control Unit", Rev. Elect. Comm. Lab., Vol.20, No.3-4, pp.231-239, March-April 1972.
- [18] P.J. Kuehn, "Multiqueue Systems With Non-Exhaustive Cyclic Service", BSTJ, Vol.58, pp.671-699, March 1979.
- [19] M. Krym, "Performance Analysis of A Local Area Network With Application to Fibre Optic Intergrated Voice/Data", Master's Thesis, Carleton University, August 1984.
- [20] D. Manfield, "Analysis of a Priority Polling System for Two-Way Traffic", To appear in IEEE Trans. on Communications.
- [21] O. Hashida, "Analysis of Multiqueue", Rev. Elec. Comm. Labs., 20, pp.189-199, March-April 1972.

END

1 5 0 4 8 6

FIN