Performance Analysis of Priority Queueing Systems in an ATM Environment

by

Jason Pieloor, B. E. (Hons.)

Department of Electrical and Electronic Engineering

Submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

University of Tasmania (October 1996)

School of Electrical Engineering
& Computer Science

Statement of Originality

This thesis contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person, except when due reference is made in the text.

Jason Pieloor

Authority of Access

This thesis may be made available for loan and limited copying in accordance with the Copyright Act 1968.

Jason Pieloor

Abstract

Current and future digital telecommunication networks are facing ever increasing bandwidth and service demands. One scheme aimed at meeting these demands is the broadband integrated services digital network (B-ISDN). The B-ISDN is based on the asynchronous transfer mode (ATM) which provides flexible and dynamic transport and routing functions. One of the main challenges for designers and managers of these networks is to provide a guaranteed quality of service (QoS) for each connection, while still achieving a high network utilisation overall.

To provide a guaranteed QoS, the network must have a mechanism for deciding whether it can support a requested quality of service for a new connection, whilst still maintaining the QoS of existing connections. This decision process is called connection admission control (CAC). Mechanisms for implementing CAC must be acceptably accurate, while executing in as the shortest time as possible. Most CAC mechanisms are based on the application of queueing theory to the network — the accuracy of which is largely dependent on the models of the network traffic used, and the solution method chosen for the queue analysis.

B-ISDN connections can be generally classified as either loss sensitive or delay sensitive. Unfortunately, the requirements for transporting both these types of connections within the same network appear to be at odds with each other. Small internal buffers in ATM switching nodes result in small transmission delays but potentially high loss rates, while the use of large buffer sizes favours small loss rates with long transmission delays. To accommodate both types of connections, a dual buffer approach can be used within the network switches, wherein one buffer receives priority access to the output line over the other. Delay sensitive traffic can then be served ahead of loss sensitive traffic, and a large buffer space can be used to accommodate low loss requirements. The difficulty with the dual buffer approach for the purposes of CAC, is that analysis of the loss queue is complicated due to service interruptions caused by the delay traffic. Fortunately, a relationship between single buffer and dual buffer analyses exists, allowing some of the more important results for the loss queue to be obtained using single buffer analysis only.

This thesis considers the modelling of traffic both at the edges of the network, and at intermediate stages within the network. Several models are proposed, with a particular concern that the bursty nature of actual network traffics be adequately captured. In order to apply these descriptions of the network traffic to connection admission, the population analysis of infinite buffer queueing problems is carried out using the proposed models. Queueing delays are then obtained directly from the queue population results. Although the traffic models are not particularly complicated, closed form solutions for the average and variance of the queue population are obtained only for one type of bursty traffic model. For the other traffic models, exact numerical solutions are discussed, and some simple approximations examined. To overcome limitations in these solutions, a new approximation technique is proposed, which achieves extremely high accuracy for a modest computational cost. In addition to these infinite buffer results, consideration is also given to obtaining the loss probabilities of the finite buffer problem.

The developed queueing theory is lastly applied to a dual buffer example problem to highlight the role of correlations between arrival processes, and to the modelling of queue outputs for the purpose of describing networks of switching elements.

Acknowledgements

I would firstly like to extend my heartfelt thanks to my supervisors, Doctor David J. H. Lewis and Professor D. Thong Nguyen, for all their help and encouragement through the course of my studies. Doctor Lewis has provided me with friendship, valuable guidance, and the freedom to choose the direction of my research, for all of which I am deeply grateful. Professor Nguyen has kept an ever watchful eye on the needs of the postgraduate students in this Department, encouraging us to better achievements, and helping us to do so.

I would like to thank all of the staff of the Department of Electrical and Electronic Engineering at the University of Tasmania, which became my 'home away from home' for many years. In particular I would like to express my thanks to Glenn Mayhew and Russell Twining, who have been friends as well as providing me with much assistance. Many thanks go also to my fellow postgraduate students, who made the study environment cooperative and pleasant. I'd particularly like to thank Marc Stoksik, Andrew Innes, Andrew Bainbridge-Smith, and John McCulloch, who provided me with help and guidance, and with whom I have spent many enjoyable hours.

A very special thanks go to my family and dear friends, for keeping me from going completely bananas when the going got tough. I'd especially like to thank my wife Jewel Ashley for all her support, love, and encouragement. This thesis would not have been accomplished without these fine people, and to all of them I am indebted. It has been a long time coming, but I am proud of the result.

Lastly, I'd like to thank Telstra Research Laboratories, who provided financial assistance and other resources for my studies through their *Postgraduate Research Fellowship Scheme.*

Contents

Glossary

Acronyms

AAL	ATM adaptation layer
ADPCM	Adaptive differential pulse code modulation
ATM	Asynchronous transfer mode
B-ISDN	Broadband integrated services digital network
CAC	Connection admission control
CBR	Constant bit rate
CCITT	International Telegraph and Telephone Consultative Committee
CDV	Cell delay variation
CLP	Cell loss priority
D-BMAP	Discrete-time batch Markov arrival process
FIFO	First in, first out
GFC	Generic flow control
HEC	Header error control
HOL	Head of line
IBP	Interrupted Bernoulli process
IN	Interconnection network
IPC	Input port controller
ISDN	integrated services digital network
kbps	Kilobits (10^3 bits) per second
LAN	Local area network
Mbps	Megabits (10^6 bits) per second
MMPP	Markov modulated Poisson process
msec	Milliseconds (10^{-3} seconds)
μsec	Microseconds (10^{-6} seconds)
N-ISDN	narrowband integrated services digital network
NNI	Network-Node Interface
OPC	Output port controller
OSI	Open systems interconnection

PABX	Private automatic branch exchange
PCM	Pulse code modulation
PDU	Protocol data unit
PRM	Protocol reference model
PSTN	Public switched telephone network
PT	Payload type
QoS	Quality of service
SPP	Switched Poisson process
STM	Synchronous transfer mode
TDM	Time division multiplexing
UNI	User-Network Interface
UPC	Usage parameter control
VBR	Variable bit rate
VC	Virtual channel
VCC	Virtual channel connection
VCI	Virtual channel identifier
VLSI	Very large scale integration
VP	Virtual path
VPC	Virtual path connection
VPI	Virtual path identifier

Mathematical Notation

General mathematical notation:

$\approx$	Approximately equal to
$\rightarrow$	Approaches, or 'tends towards'
$\lceil x \rceil$	Largest integer less than x
$\|x\|$	Absolute value of x
$\Pr(X)$	Probability of event X
$\mathrm{E}[X]$	Expected value of random variable X
$\mathrm{Var}[X]$	Variance of random variable X
$\otimes$	Kronecker product

Notation used throughout this thesis to identify specific quantities:

$\mathbf{I}$	Identity matrix
$\mathbf{A}$	Transition probability matrix

$\mathbf{P}(z)$	Matrix of arrival probability generating functions
$\mathbf{b}$	Empty system probability vector
$\mathbf{e}$	Unit column vector (all elements equal to 1)
$\boldsymbol{\mu}$	Invariant probability vector of matrix $\mathbf{A}$
$\mathbf{X}(z)$	Vector of queue population probability generating functions
$\mathbf{h}_n(z)$	Left eigenvector of $\mathbf{AP}(z)$ corresponding to its nth eigenvalue
$\mathbf{g}_n(z)$	Right eigenvector of $\mathbf{AP}(z)$ corresponding to its nth eigenvalue
$\mathbf{u}(z)$	Left Perron–Frobenius eigenvector of $\mathbf{AP}(z)$
$\mathbf{u}(z)$	Right Perron–Frobenius eigenvector of $\mathbf{AP}(z)$
$\omega_n(z)$	nth eigenvalue of $\mathbf{AP}(z)$
$\delta(z)$	Perron–Frobenius eigenvalue of $\mathbf{AP}(z)$
λ	Average number of arrivals occurring in one time slot
M_2	Second moment of number of arrivals occuring in one time slot
M_3	Third moment of number of arrivals occuring in one time slot
γ	Autocorrelation parameter
η_r	rth moment of the active period of a source
L_q	Average queue population
$\text{Var}[L_q]$	Variance of the queue population
D_q	Average queueing delay
$\text{Var}[D_q]$	Variance of the queueing delay
$R(m)$	Autocorrelation coefficient at a lag of m

Note that a specific source is usually identified by a subscript i term (eg. λ_i). When a subscript already exists for the quantity, the source is identified by a leading i in a comma separated pair (eg. $\eta_{i,r}$). Derivatives are indicated by the primes of the variable (eg. $\delta'(z)$ and $\delta''(z)$ for the first and second derivatives).

Other Notation

geom-geom IBP	An interrupted Bernoulli process (IBP) with geometrically distributed active and inactive periods
phase-geom Binary	A form of IBP with phase-type distributed active periods, and geometric inactive periods, and that generates a single arrival in every time slot for which the process is active (a peak rate of 1)
geom-geom Binary	A type of phase-geom Binary process having geometrically distributed active periods

Preface

Current and future digital telecommunication networks are facing ever increasing bandwidth and service demands, requiring new design approaches. The broadband integrated services digital network (B-ISDN) is a scalable solution to the demand problem, based on the asynchronous transfer mode (ATM). ATM is a packet switching technology, which provides flexible and dynamic bandwidth management capabilities for the B-ISDN by sharing available transmission bandwidth amongst all connections. As a result, one of the main challenges facing the B-ISDN will be providing a guaranteed quality of service (QoS) for individual connections, while still achieving high network utilisation overall.

To provide a guaranteed QoS, the network must have a mechanism for deciding whether it can support a requested quality of service for a new connection, whilst still maintaining the quality of existing connections — a process called connection admission control (CAC). Mechanisms for implementing CAC must be acceptably accurate, while executing in as the shortest possible time. Due to buffering within the ATM switches, most CAC mechanisms are based on the application of queueing theory to the network — the accuracy of which is largely dependent on the accuracy of the models of network traffic sources, and the solution method chosen.

The focus of this thesis is the discrete-time performance analysis of single and dual buffer queueing systems suitable for application in an ATM environment. This research has been motivated by a desire to both expand the current knowledge of queueing theory as applied to these problems, and to investigate practical solution methods and approximations. Implementation of the results within a larger CAC framework has not however been attempted in this work.

Thesis Organisation

This thesis comprises 8 chapters and 6 appendices. Following the introductory material of Chapter 1, Chapters 2 to 5 develop theoretical and numerical results for the population average and variance of some infinite buffer queueing problems based on an IBP traffic model. Losses in finite buffers and the relationship between queueing delays and queue populations are considered in Chapter 6, while Chapter 7 investigates dual buffer systems and queue output processes. Chapter 8 concludes the thesis, and suggests some areas for future research.

Chapter 1 provides an introduction to the development of the B-ISDN. An overview of the structure of the B-ISDN and ATM is provided, and some of the issues relating to traffic management and connection admission control are discussed. ATM traffic modelling is considered, and the models used in the thesis are briefly described. The chapter concludes with a list of the contributions made by the author in this work.

In Chapter 2, a general analytical solution for the average and variance of the population of an infinite buffer discrete-time G/D/1 queue fed by a number of batch Markov arrival processes is developed. The solution is based on a probability generating function approach, and forms the background theory for the application to specific queueing problems in Chapters 3, 4, and 5. Consideration is also given to the analysis of multiple buffer queueing systems on the basis of the single buffer analysis.

Chapter 3 applies the results of Chapter 2 to the analysis of queues subject to arrivals from a heterogeneous mix of two state Markov sources. Since the exact solution technique is shown to be quite limited in the number of sources it can accommodate, several approximate solution methods are investigated for accuracy. A new approximation technique is then proposed, and shown to be extremely accurate.

The queueing behaviour of a related Markov traffic model is then investigated in Chapter 4. This problem has a well known closed form solution for its average queue population, which is reproduced. This solution method is then combined with the analysis of Chapter 2 to finally provide a previously unknown, closed form solution for the queue population variance.

In Chapter 5, the queueing problem of Chapter 3 is re-examined, but with the additional complication of cyclically interrupted service. Although the theoretical development is straightforward, numerical difficulties are encountered that prevent solutions being obtained for even fairly simple systems. An adaptive solution method is proposed which allows accurate estimates to still be obtained when the exact solution method fails. Due to the large computational burden of the exact solution, approximation techniques are investigated, with the new method of Chapter 3 shown to provide the highest accuracy.

Chapter 6 begins with an investigation of the loss probabilities of finite buffer queueing problems. An improvement to the commonly accepted relationship between the loss and tail distribution of the infinite buffer traffic is demonstrated, and the accuracy of an approximation result from the literature is confirmed. The relationship between queueing delays and queue populations are also considered in this chapter.

Chapter 7 applies the average and variance solutions developed in Chapters 3 and 5 to dual buffer queueing systems, identifying the problem of correlation between the high and low priority arrival streams. The problem of modelling the output of a queueing system is also investigated, with a proposed parameter matching method tested for a simple queueing system. The parameter matching method also provides exact solutions for the first moments of the busy periods of the output of the queueing system investigated in Chapter 4.

Finally, concluding remarks on the results of this thesis are presented in Chapter 8 along with a discussion of future topics and research directions.

The six appendices provide supplemental information for the main text of the thesis. Appendices A and C relate to the derivation of the various queueing theory results, while Appendix B provides results on an investigation into the average delays seen by individual traffic sources in a shared queueing system. Appendix D provides a short proof for a result that ties the definition of the autocorrelation parameter used in the thesis to an observable physical quantity. Some practical details on the implementation of iterative queue analysis programs is provided in Appendix E, while Appendix F closes with a collection of miscellaneous mathematical relations.

Publications

This research has resulted in the following publications:

1. J. Pieloor and D. J. H. Lewis. Variance of a discrete-time G/D/1 queue fed by two-state on-off sources. *Electronics Letters*, 32(1):19-20, Jan. 1996.

2. J. Pieloor and D. J. H. Lewis. Queueing behaviour of on-off binary sources. In *Proceedings Australian Telecommunication Networks and Applications Conference*, pages 443–447, Dec. 1995.

3. J. Pieloor. Analysis of an ATM output buffer for a switch carrying both CBR

and VBR traffic. In *Proceedings Australian Telecommunication Networks and Applications Conference*, pages 875–880, Dec. 1994.

4. D. Habibi, D. J. H. Lewis, D. T. Nguyen, and J. Pieloor. Analysis of an access node multiplexer in a system serving CBR and VBR traffic. *Computer Communications*, 16(12):776–780, Dec. 1993.

5. J. Pieloor and D. McLaren. Accurate source models and a high performance simulator for broadband ISDN analysis. In *Proceedings Australian Broadband Switching and Services Symposium*, pages 523–529, July 1992.

6. D. Habibi, D. J. H. Lewis, D. T. Nguyen, and J. Pieloor. Performance of a multiplexer in a B-ISDN network with STM and ATM traffic. In *Proceedings Australian Broadband Switching and Services Symposium*, pages 691–698, July 1992.

In addition, three interim progress reports were submitted to Telstra Research Laboratories in March 1992, January 1993, and December 1994 as part of the *Postgraduate Research Fellowship Scheme* which the Laboratories sponsored, and of which I was a recipient.

Chapter 1

Introduction

Traditionally, network providers have built networks that cater exclusively for specific services. For example, voice communication has been primarily the domain of the public switched telephone network or PSTN, while computer communications have been achieved using packet switched networks based on X.25 or similar protocols. Due to the specific nature of these networks, neither is very efficient at providing services for the other traffic type. This use of service specific transport mechanisms has been an efficient solution for both customers and network providers while the number of different services required has remained small. However, this situation is changing.

Over the last two decades, the demand for new communication services has increased significantly, particularly with recent advances in computer and image processing technologies. It is readily apparent that the provision of separate networks (and perhaps separate connections to customer premises) for each service is becoming inefficient due to the costs involved in maintaining the multiple networks, and from a lack of flexibility in being able to adapt to shifts in demand or the possible provision of future service types.

In 1984, the International Telegraph and Telephone Consultative Committee (CCITT)[1] began to publish recommendations on the standardisation of an *Integrated Services Digital Network* or ISDN — a digital based network that would be capable of providing various communication services in one integrated network. By moving error recovery, flow control, and similar high level functions to the edges of the network, the use of the ISDN for switching and transmission could be made transparent to network customers.

[1] In 1993, the CCITT was replaced by the Telecommunication Standardisation Sector (ITU-T) of the International Telecommunication Union (ITU). We will continue to use the name CCITT in this document however.

The original ISDN proposal, now referred to as narrowband ISDN (N-ISDN), provides multiple 64 kbps digital circuit switched connections that can be used either for voice telephony or computer communications. The circuit switched approach means that connections are effectively isolated from each other, with low transmission delays, and simplified connection overheads. The 64 kbps capacity of the basic circuit is due to the bandwidth requirements of voice telephony.

With the increasing transmission bandwidth requirements of new services, such as video conferencing and high speed intercomputer communication, the rather limited range of capacities available to N-ISDN services were soon seen by the telecommunication industry as inadequate. In recognition of this, the CCITT began in 1988 to concentrate on standards for a high bandwidth version of the ISDN called broadband ISDN (B-ISDN). Unlike the circuit switched approach of the narrowband implementation, B-ISDN is based on a packet switching methodology referred to as *Asynchronous Transfer Mode* or ATM.

In section 1.1, we will look at the various components of the B-ISDN, and in particular look at the role of the ATM in the implementation of the network. In section 1.2, issues relating to the management of the network, in terms of admitting and policing user connections are considered. The modelling of ATM traffic for the purposes of implementing network management strategies is then considered in section 1.3, and finally section 1.4 concludes the chapter by summarising the aims and contributions of this thesis.

1.1 Overview of the Broadband ISDN

Figure 1.1 illustrates an example broadband ISDN, where customers or users gain access to the ATM based network via a standard interface called the *User–Network Interface* or UNI. The interface between intermediate switching nodes within the network is referred to as the *Network–Node Interface* or NNI.

In the following we will discuss the various components of the B-ISDN, starting with a closer look at the ATM itself.

1.1.1 The Asynchronous Transfer Mode

There are two basic methods for implementing a digital network — using circuit switching, or using packet switching. In the former, a dedicated (virtual) circuit is provided for each connection that is logically separate from every other circuit. That is, the

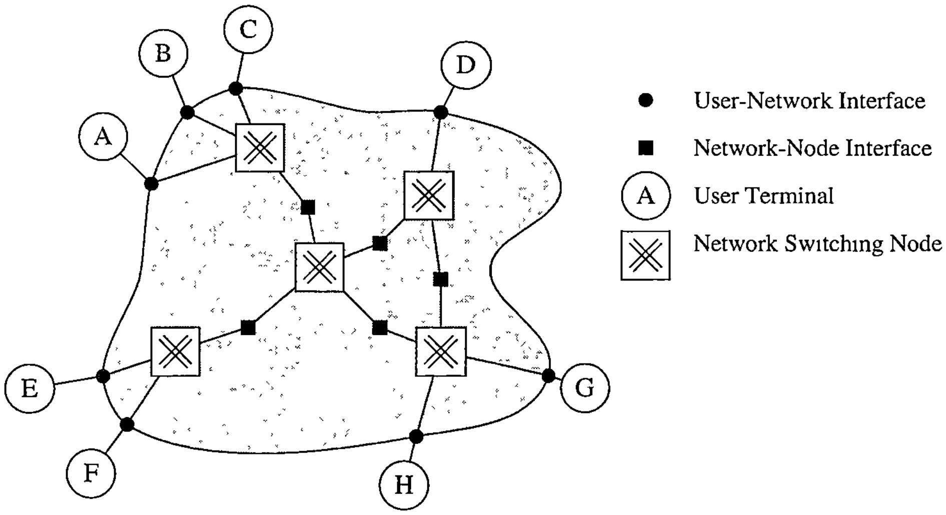

Figure 1.1: *Example of an ATM based B-ISDN*

behaviour of one connection will not affect behaviour of another. In a packet switched network however, the bandwidth is shared amongst all connections in such a manner that the behaviour of each connection affects that of the others.

The main advantage of the circuit switching approach is that, once a connection has been made, information transmitted across the network incurs only a very small and usually constant delay. Its main disadvantage is that the bandwidth that the network can provide to a connection is limited to the capacities of the circuits that the network supports. In addition, the dedication of a circuit to a single connection that may only use the whole circuit capacity sporadically means that the network is being underutilised.

In contrast, a packet switched network is capable of providing a very wide range of capacities, allowing the requirements of each individual connection to be closely matched. As a result, more connections can potentially be simultaneously supported by such an arrangement. The disadvantages of this approach are that transmission delays in such an environment tend to be highly variable, and the shared nature of the available bandwidth requires extra network controls to ensure that all connections do not suffer from the misbehaviour of one.

The CCITT chose a packet switching implementation for B-ISDN in order to provide 'dynamic bandwidth allocation on demand, with a fine degree of granularity' [59] as opposed to the circuit switched nature of the N-ISDN. To enable fast and efficient implementations of packet switches, and to minimise transmission delays, a small fixed packet size of 53 bytes (424 bits) was chosen, and distinguished by the name *cell*. Within each cell, 5 bytes were allocated to the header, providing identification and routing information on a per cell basis, leaving 48 bytes for user information. In addition, error detection and recovery within the network were restricted wholly to the cell headers, with similar functions for user information being moved to the network edges.

The term 'transfer mode' is used by the CCITT to indicate a specific method of transmitting and switching information in a network [42]. The term 'asynchronous' in the ATM label indicates that cells belonging to a particular connection may occur at irregular intervals on the connection medium. This is in contrast to the *synchronous transfer mode* or STM based approach used in digital circuit switched networks, where information belonging to a particular circuit occurs only at specific positions within a predefined constant period referred to as a 'frame'.

Virtual Channels and Virtual Paths

ATM is a connection oriented protocol, requiring a path or route to be set up through the B-ISDN before information can be sent from one user to another. This transmission path remains unchanged for the duration of the connection, and is created or assigned by signalling between the user and the network. A one way connection path between two users is described as a *virtual path connection* or VPC. A single VPC can support multiple *virtual channel connections* or VCCs, each which follow the same network path but belong to logically separate entities (such as different phone conversations between the same two customer sites). The ATM protocol guarantees that under normal (i.e. fault-free) conditions, cell sequence integrity within a VCC will be guaranteed — that is, cells belonging to a specific VCC will leave the network in the same order that they enter it[2].

Cells belonging to a particular VCC and VPC are identified by a virtual channel identifier (VCI) and virtual path identifier (VPI) in the ATM header. Since a VPC will generally pass through a number of ATM switching nodes (as indicated by Table 1.1) it is not possible to guarantee the availability of a unique VPI and VCI for every network path. Instead, a VPC is considered to be made up of a number of *links*, at the end points of which the path and channel identifiers are remapped to allow for unique identification of the connection along the next link.

ATM Cell Header Structure

Figure 1.2 shows the structure of the 5 byte ATM cell header at the UNI and NNI, where the following fields are identified [9]:

[2] Although this does not guarantee that all the cells entering the network will be successfully transported.

Connection Type	Number of nodes
Local	$1-4$
Toll	$5-7$
International	$8-10$

Table 1.1: *Number of ATM switching nodes in typical end-to-end B-ISDN connections. These figures are from Table 1 of ITU draft recommendation E.72x [56].*

GFC: The 4 bit *Generic Flow Control* field has local significance only (applies to the UNI and not the NNI) and can be used to provide standardised local functions (such as flow control) on the customer site. The value encoded in the GFC is not carried end-to-end, and is overwritten by the ATM switches.

VPI/VCI: The Virtual Path and Virtual Channel identifiers are nominally 8 and 16 bits respectively for the UNI, and 12 and 16 bits respectively for the NNI. The actual number of routing bits in these subfields however is negotiated between the user and the network.

PT: The *Payload Type* field consists of 3 bits that identify the contents or payload of the ATM cell as being either user or network information.

CLP: The *Cell Loss Priority* is indicated by this one bit field, and allows the user or the network to optionally indicate the explicit loss priority of the cell. The use of loss priorities will be discussed in more detail in section 1.2.

HEC: The *Header Error Control* field is used by the physical layer for detection and correction of bit errors in the cell header. The HEC can also be used for cell delineation or synchronisation by observing the number of bit errors at particular points in the transmission stream.

When a cell is carrying user information (as distinct from network administration information), the 48 bytes of the payload are transported transparently by the ATM network. Although errors in the cell headers are (usually) identified by the HEC and subsequently dealt with, separate errors in the cell payload will not be identified. As a result, if error handling is desired, it must be performed by end-to-end protocols.

1.1.2 B-ISDN Protocols

In modern communications systems, a layered approach is used for the organisation of all communication functions. The functions of the layers, and the relations of the layers with respect to each other are described in a *Protocol Reference Model* (PRM). The B-ISDN protocol reference model is illustrated in Figure 1.3 as described in ITU-T

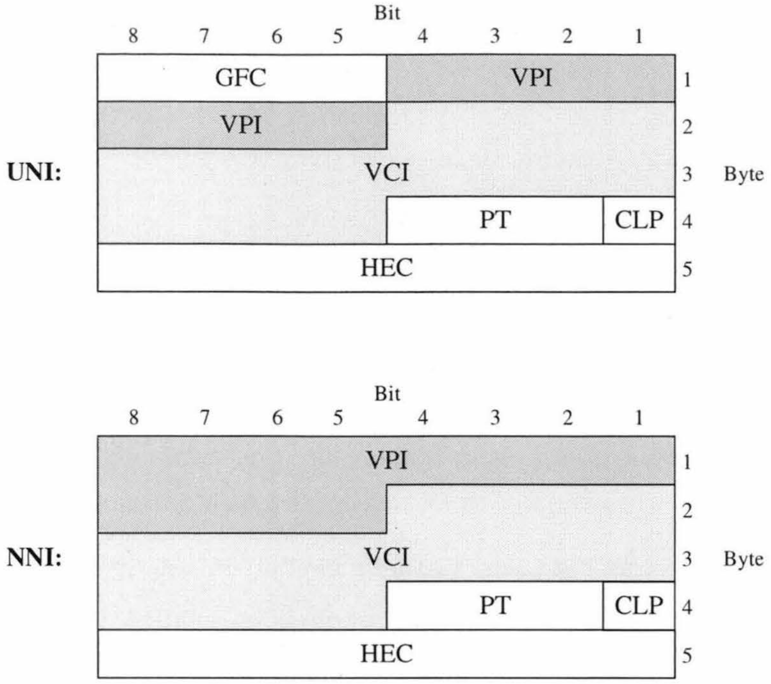

Figure 1.2: *ATM cell header structure at the User Network and Network Node interfaces of the B-ISDN.*

Recommendation I.321 [61] and [9]. The B-ISDN PRM is similar to the Open Systems Interconnection (OSI) PRM defined for CCITT applications in [65], but contains several planes in contrast to the single one of the OSI model.

The three planes of the B-ISDN PRM are identified as:

User Plane: Provides for the transfer of user application information. It contains physical, ATM, and ATM adaptation layer functions required to implement the different user B-ISDN services, as well as application specific higher layer protocols.

Control Plane: Deals with call establishment and release, and other connection control functions necessary to provide switched network services. It shares the physical and ATM layers with the User plane, as indicated by Figure 1.3, in addition to its own ATM adaptation and higher layer protocols.

Management Plane: Provides overall management functions, and the ability to exchange information between the Control and User planes. For convenience, the Management plane is divided into sections — Layer management and Plane management. The Layer management section performs layer specific management functions, while the Plane management section performs management and coordination functions related to the complete system.

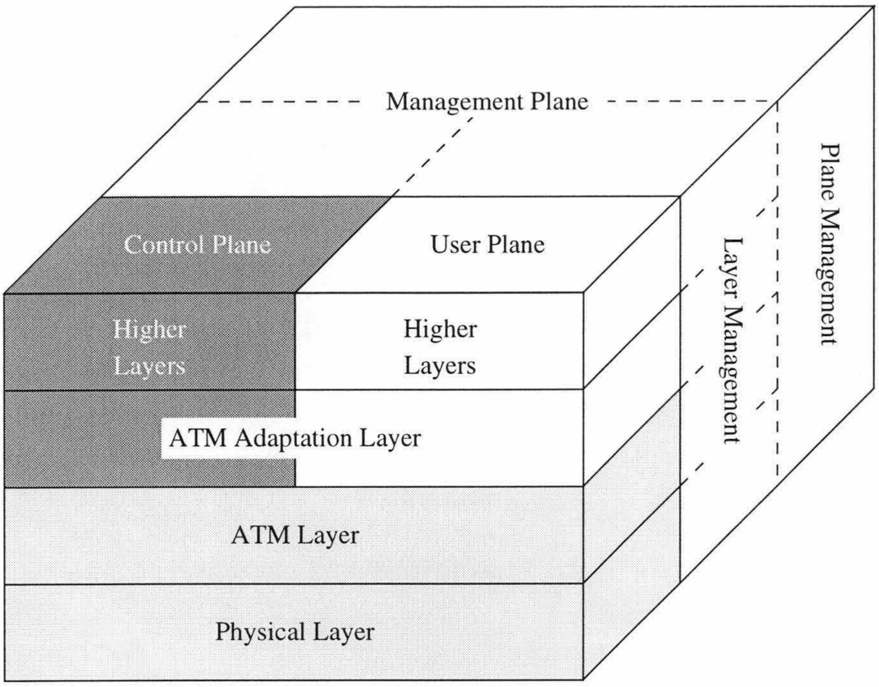

Figure 1.3: *B-ISDN Protocol Reference Model.*

The physical layer is responsible for all physical transport functions, such as the conversion of the ATM cell bit patterns to electrical or optical signals suitable for transmission on the physical medium, synchronisation of transmitters and receivers to match the transmission medium characteristics, and generation and confirmation of the HEC sequence.

The ATM layer provides the multiplexing and switching functions of the B-ISDN protocol, and is independent of the physical transmission medium. In addition, this layer is responsible for the addition and removal of the cell headers, and the mapping and translation of the cell VPI and VCI fields. We will look at some of the switching functions of the ATM layer in the context of switch fabrics in section 1.1.4.

The ATM adaptation layer (AAL) provides a plane-specific interface to the underlying ATM and physical layers. In particular, the AAL can provide functionality for the handling of transmission errors, flow and timing control, and the segmentation and reassembly of higher layer protocol data units into and out of ATM cell payloads. In order to minimise the number of AAL protocols, the CCITT identified four different service classes [62], as shown in Table 1.2. These service classes differ in their timing relations, bit rates, and connection modes.

Corresponding to these four service classes, there are four basic AAL protocol types, defined as follows [63]:

	Class A	Class B	Class C	Class D
Timing between source and destination	Required		Not Required	
Bit rate	Constant	Variable		
Connection mode	Connection Oriented			Connectionless

Table 1.2: *AAL Service classes according to ITU-T Recommendation I.362 [62].*

AAL Type 1: Corresponding to class A, this protocol provides for connection oriented, constant bit rate (CBR) services requiring a constrained timing relationship between the source and destination. N-ISDN support, and other real time or circuit emulation services will use this protocol.

AAL Type 2: Corresponding to class B, this AAL protocol provides for connection oriented services having timing constraints between the source and destination, and a variable bit rate (VBR). An example of a service requiring this protocol is VBR video.

AAL Type 3/4: Corresponding to both classes C and D, this AAL protocol was originally specified as two separate types. This class provides for data transmission services using either connection oriented or connectionless methods.

AAL Type 5: Corresponding to class C only, this AAL protocol type is a simpler form of the type 3/4 protocol that provides better bandwidth efficiencies at the cost of some error handling functions.

AAL service classes A and B can be regarded as being 'delay sensitive' due to their requirements for a constrained timing relation between source and destination. Provision of guaranteed transmission delays is one of the main weaknesses of the shared bandwidth nature of ATM. Similarly classes C and D can be regarded as 'loss sensitive', since these classes are primarily intended for data communications, where it is usually more important that a cell be delivered successfully by the network (i.e. not lost) than be delivered quickly. This categorisation of the four AAL service classes into delay sensitive or loss sensitive services will be discussed again in section 1.2.

1.1.3 The User Network Interface

The UNI is responsible for defining such factors as the interface bit rate, structure, and ATM layer behaviour. With reference to the B-ISDN protocol reference model, the UNI consists of ATM and physical layer functions only within the User-plane, while AAL and higher layer protocol functions are additionally included within the Control-plane.

Interface Structures and Bit Rates

Three physical cell transport mechanisms were foreseen by the CCITT [59] for the B-ISDN UNI:

- Purely ATM based interface — ATM cells can be transmitted back to back, with the entire transmission bandwidth available to be used for the transport of ATM and higher layer information.
- Non-ATM based interface — Some high speed transmission protocols, such as SONET (Synchronous Optical Network) and SDH (Synchronous Digital Hierarchy) require a portion of the transmission bandwidth to handle physical maintenance, synchronisation and framing functions. ATM cells can be carried transparently within the payload of these protocols.
- Framed ATM based interface — To simplify interoperability between the purely ATM and non-ATM based interfaces, physical layer cells (generated by the physical layer) are inserted into the ATM layer cell stream periodically. The inserted cells are typically used to provide physical layer *Operations, Administration, and Maintenance* (OAM) functions.

Purely ATM based interfaces are foreseen for smaller networks (such as private customer networks) where physical layer OAM functions are unlikely to be required [9]. For large or public networks, SDH or framed ATM based interfaces are required. Due to the fact that some of the available link bandwidth in these latter two interfaces is dedicated to functions other than the transport of ATM layer cells, the term *interface transfer rate* is used to distinguish the effective transfer rate of the interface from the physical transmission rate of the link. Table 1.3 lists the four bit rates currently supported by the UNI as specified in [9] and [60]. Other interfaces may be added in future, but for the moment, the 155.52 Mbps interface is seen as the most common UNI for outgoing (user to network) information.

Physical Rate	**Interface Rate**	**Example Physical Protocol**
44.736 Mbps	40.704 Mbps	DS3
100 Mbps	100 Mbps	100 Mbps Multimode Fiber
155.52 Mbps	149.76 Mbps	SONET STS-3
622.08 Mbps	599.04 Mbps	SDH STM-4

Table 1.3: *Four basic UNI bit rates and example interface protocols.*

Note that the use of framed, or non-ATM interfaces at either the UNI or the NNI is generally ignored in the analysis of network switching and multiplexing elements.

Instead, a purely ATM interface is assumed, which introduces some (hopefully small) degree of error into the analytical results. This assumption, and the effect that it has on the perceived network performance is an area requiring further attention.

1.1.4 Switching and Multiplexing

Each intermediate node in the ATM network contains one or more switches which perform the routing functions for ATM cells traversing the network, which as discussed earlier, may also involve VPI and VCI translation. The three main components of an ATM switch are the input port controllers, interconnection network, and output port controllers, as illustrated in Figure 1.4. In addition to these switching components, a central processor (not shown in Figure 1.4) is required to provide ATM layer functions such as connection establishment and release, and to monitor the overall operation of the switch.

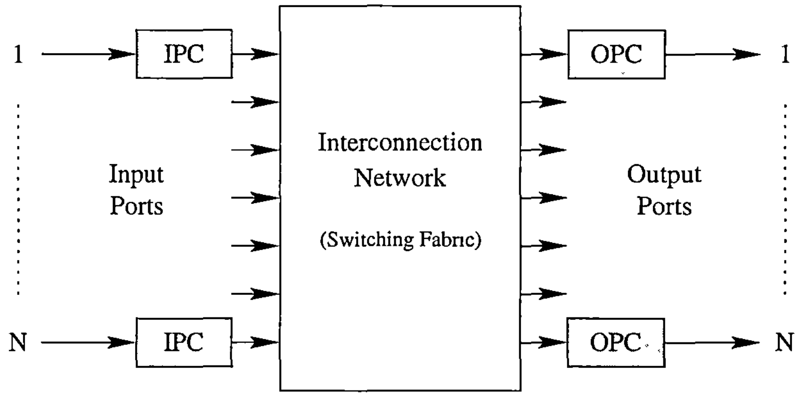

Figure 1.4: *Basic ATM switch configuration, showing the relationship between the input and output port controllers (labelled as IPC and OPC respectively) and the interconnection network of the switch fabric. In most cases, the switch will have an equal number of input and output ports, which all operate at the same line speed.*

Access nodes (on the edge of the network) may also provide multiplexing and demultiplexing functions in addition to switching operations. Multiplexers provide a means for combining cells from a number of low speed input lines into a single cell stream suitable for transmission on one higher speed output line, while demultiplexers perform the corresponding inverse operation. Multiplexers and demultiplexers may also be used to support trunk lines within the network for carrying very large quantities of data over large distances (such as between two cities).

Switches and Switch Fabrics

ATM cells are routed from the input port controllers (IPCs) to the output port controllers (OPCs) through the switch interconnection network (IN). The IN provides a self-routing facility which directs a cell from each input port to an output port using the contents of the cell header. Contention may occur however if two or more cells are directed to the same output port (output blocking), or attempt to use a common link within the fabric of the interconnection network (internal blocking). In either case, only one cell can be transmitted — the others must be either discarded, or preferably stored for subsequent transmission.

There are three basic methods to provide storage (or buffering) for blocked cells within the ATM switch [105]:

Input Queueing: Buffers are provided at each IPC, so that a cell that cannot successfully be routed to an output port (because of either internal or output blocking) waits at the input. Subsequently arriving cells are queued in arrival order. Although simple to implement, the main disadvantage of this method is that while the cell at the head of the queue waits to be transmitted, it delays other cells in the same input queue which might otherwise be successfully routed to other available outputs. This head of line (HOL) blocking effect limits the maximum throughput of the input queueing approach.

Output Queueing: Buffers are provided at each OPC so that multiple cells routed to the same output may be accommodated. The order of arrival of the cells at the output buffer (the order in which the cells are queued) is generally assumed to be random, although this will depend to a large degree on the implementation of the IN. Internal blocking can be avoided by increasing the speed of the interconnection network [5].

Shared Queueing: A single shared buffer is provided into which all cells from the IPCs are stored. A sophisticated control algorithm keeps track of the locations of the cells in this buffer, and determines which cells are to be sent to the output ports. Although cells routed to the same output port are stored in the same logical (or virtual) queue, their physical memory locations may be dispersed throughout the shared buffer. Internal blocking is not immediately obvious in this environment, but can occur in the implementation of the control algorithm. This buffering arrangement is also called central queueing, since the shared buffer is usually central to the switching element implementation.

Combinations of these three types of queueing can also be used. In any implementation the buffers must queue cells in a logical first in, first out (FIFO) order to ensure that cell sequence integrity within a VCC is preserved, as required for ATM. Of the three methods, shared and output queueing appear more favoured by industry (see Table 1.4) despite the simpler implementation of input queueing. The primary reason for this is the poor throughput and higher queueing delays of the latter approach. As an example of the throughput limitation, a simple input queueing switch, using an internally non-blocking interconnection network can only achieve a maximum throughput of 0.586 erlang [69], compared to a maximum of 1.0 for shared and output queueing. Although methods for increasing the throughput and decreasing delays by reducing the HOL blocking effect are available [105], the extra complexity of their implementation can reduce the comparative advantages of input queueing.

Manufacturer	Name	Size	Speed	Buffering
IBM Corporation	PRIZMA	16 × 16	400 Mbps	Shared
Integrated Telecom Technology	WAC-188-A	8 × 8	155 Mbps	Shared
MAZ Hamburg, GmbH	SE DAS 340	4 × 4	155 Mbps	Output
MMC Networks Inc.	ATMS2000	32 × 32	155 Mbps	Shared
Scorpio Communications	ATM JC	2 × 2	640 Mbps	Shared

Table 1.4: *Size, speed, and buffering method details for 5 commercially available ATM switches as of June 1996. The size describes the number of input and output ports of the device, while the speed indicates the maximum line speed per port. The development and functional implementation of IBM Corporation's PRIZMA Switch-on-a-chip is presented in [21].*

Although the both the output and shared buffer implementations provide logical output queues, the shared buffer approach has the advantage that the total storage space required to achieve a desired cell loss probability can be significantly reduced over that required for output buffering. Output buffering has a slight advantage over shared queueing in terms of the complexity of the interconnection network and control logic, but overall design issues tend to favour the use of a shared buffer [42].

A comprehensive survey of the many different interconnection network topologies and buffering methods available is given by Ahmadi and Denzel in [5].

Construction of Larger Switches

Table 1.4 lists some basic details on a selection of commercially available ATM switches, in particular showing that these switching elements have only a small number of input and output ports. The size of these switches is basically restricted by the limitations of silicon based VLSI implementation. Switches with large numbers of inputs and outputs must therefore be constructed by interconnecting smaller switching elements in a suitable fashion. There are numerous methods and topologies for implementing this interconnection, and a serious study of the field is beyond the scope of this introduction — the interested reader is directed to [100] or [5].

Networks of Queues

Within each switching element, cells may experience delays due to queueing. With multiple switching elements making up a single network node, and multiple nodes required to construct a single network path (see Table 1.1) cells belonging to a particular VPC will pass through a sequence of queues.

The entire B-ISDN can in fact be envisaged as an interconnected network of queues. Figure 1.5 illustrates the basic idea for a simple interconnection of four 2 × 2 switches that might be used to construct a larger 4 × 4 switch. Cells belonging to one VPC are delayed in network queues by other cells due to contention for the output link, and may be lost entirely if a full buffer is encountered (no more space to store blocked cells). Factors such as queueing delay and cell loss are important descriptors of the performance of the network. We will look at this issue in the context of network management in the following section.

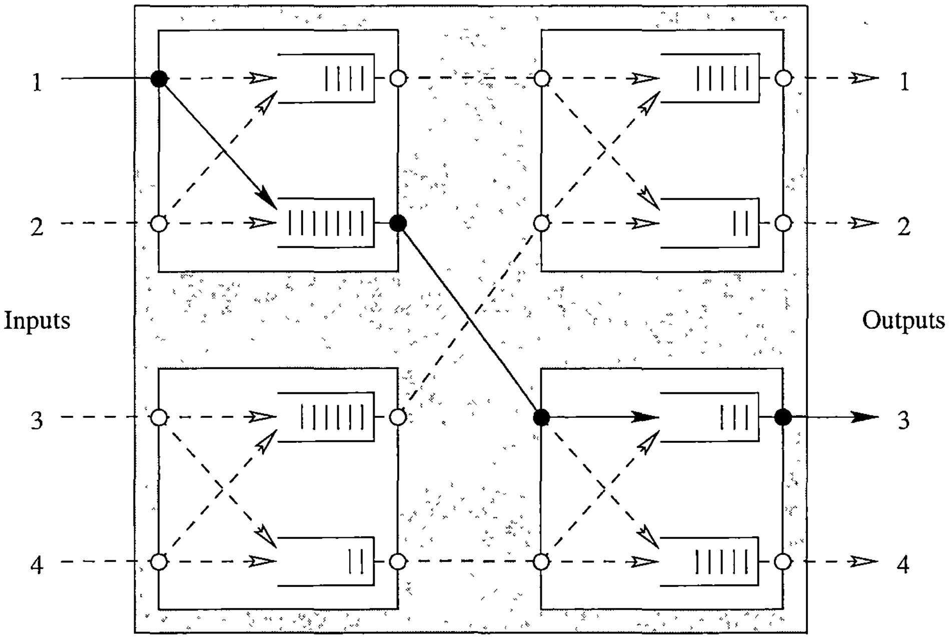

Figure 1.5: *Illustration of how a simply interconnected network of four* 2 × 2 *switches effectively behaves as a network of queues. A single VPC path from input port* 1 *to output port* 3 *that passes through two queues is highlighted.*

1.2 Traffic Management Issues

A customer or user will generally require the network to provide a certain *quality of service* (QoS) for each of its connections. That is, the performance of the network, as seen by cells on the user VPC or VCC must be within certain bounds. The parameters that are used to quantify the connection QoS will vary with the type of information being transmitted, but may include cell transfer delay, cell delay variation, and cell loss ratio [64].

In order to provide QoS guarantees, traffic[3] flows into and within the ATM network are controlled by a network administration or management, which has two primary objectives:

- To ensure that the network meets the QoS requirements of established connections
- To maximise the utilisation of network resources

The CCITT has defined a set of generic traffic control functions to aid in meeting these objectives [64]:

Network Resource Management: Provisioning may be used to allocate network resources for the purpose of separating traffic flows on the basis of quality of service requirements or connection type. Capacity can be reserved along certain network paths in order to accommodate peak hour usages or allow for rerouting of established connections in the event of a node or link failure [93].

Connection Admission Control: CAC is a set of actions taken by the network during the establishment of a connection that determine whether the network can accept the new connection and still meet the QoS requirements of both the existing and new connections. Routing functions are also part of the CAC actions, which allows alternative routes to be investigated when establishing a new VPC.

Usage Parameter Control: Each connection is admitted to the network on the basis of a set of traffic parameters that describe the potential behaviour of cells generated on the VPC or VCC. If traffic on the connection violates the negotiated traffic parameters (either maliciously or unintentionally) the QoS of other connections within the network may be adversely affected. Usage parameter control (UPC) allows the network to monitor and optionally react to changes in the actual

[3] The term 'traffic' is a generic one, used to indicate the behaviour of information flows within the network.

traffic parameters. The UPC function is usually located within the user–network interface.

Traffic Shaping: Traffic shaping is a mechanism that alters the traffic characteristics of a stream of cells on a VCC or VPC to achieve a desired modification of the traffic characteristics. Examples are peak cell rate reduction, burst length limiting, and cell spacing. The UPC may perform traffic shaping on user traffic that violates its negotiated traffic contract.

Priority Control: As mentioned in section 1.1.1, the ATM specification provides for two loss priority levels within each cell header. Priority control allows the network to selectively discard low priority cells, provided that it can still meet the QoS objectives of the connections involved.

We will consider Connection Admission Control in section 1.2.2, but first we address the issue of priority control within the B-ISDN. While CAC allows the network to ensure that it can meet the QoS requirements of its users, priority control provides a means whereby the utilisation of the network resources can be maximised. If all connections are treated similarly, without priority distinction, then CAC and the allocation of network resources must be based on the most stringent of the QoS requirements, thus limiting the connections that the network can support [115].

1.2.1 Priority Control in the B-ISDN

The CCITT has identified two loss priorities for ATM. The loss priority of cells belonging to a particular VPC or VCC may either be set by the user (to separate information into essential cells and ordinary cells, such as might be used in layered video coding [110]) or by the network (to indicate cells violating the agreed traffic parameters at the UPC).

Provision of Loss Priorities

Loss priorities are often referred to as space priorities [72], since selective discard of cells by the network does not violate the cell sequence integrity of a VCC or VPC. Although low priority cells that violate their traffic contract may be discarded by the UPC at the network edges, the main benefit of the use of loss priorities is seen within the switching elements of the network. An ATM switch can implement loss priority distinction within switch buffers using one of two general methods [35, 68]:

Partial Buffer Sharing: Cells of both priority levels are accepted to the buffer while the total number of queued cells is below some threshold value T. When the total number of cells exceeds T, only high priority cells are accepted. The value of T may be fixed or be adaptively controlled.

Modifications of this basic arrangement are possible. One method is to introduce hysteresis into the acceptance and rejection mechanism by providing one threshold above which low priority cells are rejected, and a second threshold below which they are accepted again. Another alternative is to count the number of low priority cells in the buffer, and use this for comparison with the threshold rather than using the total number of queued cells [18].

Pushout: Cells of both priorities are accepted to the buffer until it is full. A high loss priority cell arriving to the full buffer may push out a low priority cell that is already queued in the buffer. High priority cells are therefore not lost from the buffer until there are no low priority cells present. Low priority cells arriving to a full buffer are lost. Provision for a minimum number of low priority cells to be permitted in a full buffer can be used to provide control over the relative low priority loss rate.

An alternative proposed by Suri et al. in [127] allows cells from either loss priority to push out cells of the other priority if there are more than a certain number of those cells in the buffer. The relative numbers of cells allowed in the buffer are adjusted to provide the best loss performance for the low priority cells without violating the QoS of the high priority cells.

Numerous studies have been performed comparing these two methods [16,72,127], and of the two the pushout mechanism generally provides the better solution. This method provides smaller loss probabilities for both priorities, since no cells are discarded until the buffer actually becomes full, and in addition the pushout approach can ensure that no high priority cell is discarded while a low priority cell is queued. The disadvantage of the pushout method is that it requires a more complex control algorithm in order to maintain cell sequence integrity when cells are actually pushed out [72].

In addition to these cell based discard strategies, there has been a growing emphasis on the use of *packet discard strategies* [53,54,113] to further improve network performance for data communications. Many local area and packet switched networks for computer communication use large packet sizes to reduce transmission overheads. In order to transport these large packets across the B-ISDN or through a private ATM network, each packet is segmented into a number of ATM cells. If even a single ATM cell from this packet is lost in transmission due to buffer overflow, the entire packet is lost, even though the network may deliver the remaining cells successfully. In the event of a cell loss, it is to the advantage of the network if it can recognise other cells belonging to the same packet, and discard them as well before they enter the network queues. This action requires the network to recognise the start and end of each packet, requiring a degree of AAL functionality to be provided within each switching node. Although this goes against the ATM philosophy to some extent, the advantages that packet discard strategies can provide to packet based data transmission appear considerable.

Provision of Delay Priorities

As we identified in section 1.1.3, the four AAL service classes recommended by the CCITT can be categorised as either loss sensitive or delay sensitive. Although two loss priorities are specified for ATM, there is no current recommendation for the explicit provision of delay priorities.

One of the arguments often used against providing delay priorities in the B-ISDN is that queueing delays will be small compared to the propagation delays of the network [35, 115]. When network utilisations are low this may indeed be the case, but as the network utilisation increases, the average number of cells stored in the switch buffers and the queueing delays associated with these, will also increase. The use of small buffer capacities (such as in the SE DAS 340 switch of Table 1.4, which provides only 10 cell positions per output buffer) was at one stage seen as necessary to limit queueing delays, but this approach also severely limits the potential utilisation of the network in terms of meeting user loss requirements.

The simplest method to provide delay priorities within ATM switching elements is to allow high priority cells to preferentially be transmitted ahead of low priority cells sharing the same switch buffer. This *preferential service* can be achieved by providing logically separate queues for each priority class, and then implementing a service mechanism to select cells from each of the queues for transmission on the output link (see for example [10, 17, 55, 90, 129]). The simplest service mechanism is one that provides head of line service to the queues in order of priority [35, 134]. That is, the next cell to be transmitted is chosen from the highest priority buffer having queued cells. Thus the lowest priority queue will not receive service until all the other delay priority queues are empty. It is interesting to note that of the five switching elements listing in Table 1.4, three of them provide facilities for implementing delay priorities.

CCITT Draft recommendation E.73x [57] suggests that HOL queueing can be used to provide delay priorities, with the priority of a cell being indicated by the VPI and VCI values in the cell header. The delay priority of cells belonging to a particular VPC or VCC must be the same, unlike for the loss priority case, since otherwise cell sequence integrity could be violated. Different VCCs belonging to the same VPC could have different priorities however, since ATM only guarantees cell sequence integrity on virtual channels. Loss priorities could additionally be implemented using either partial buffer sharing, or pushout within each logical priority queue.

A Loss and Delay Priority Scheme

Where traffic priorities are considered in this thesis (Chapters 2, 5, 6, and 7) two priorities are assumed for the purpose of separating traffic into delay sensitive and loss sensitive categories. That is, we assume that traffic that is not sensitive to delay must be sensitive to loss. This two priority system can be implemented using HOL service priority and a dual buffer arrangement within the ATM switches. Output queueing will be assumed throughout. Figure 1.6 shows a simple ATM switch arrangement using two buffers per output port. The dual buffer arrangement may be realised either by providing physically separate high and low priority buffers, or by logically partitioning the one physical space.

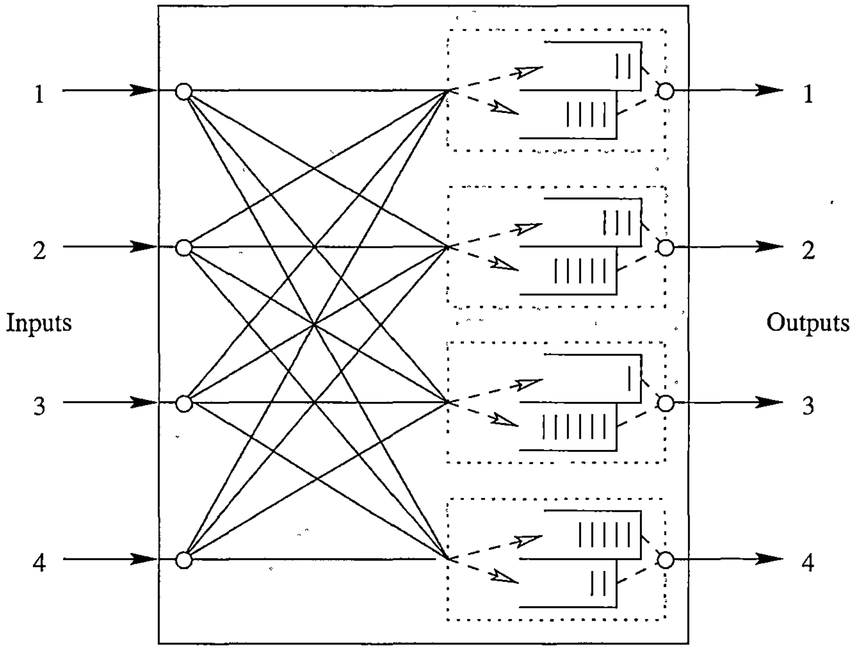

Figure 1.6: *A simple ATM switch using dual buffers at each output port to support separation of network traffic into either delay sensitive of loss sensitive categories. The upper of each buffer pair holds delay sensitive traffic, and recives HOL service priority.*

In order for the delay sensitive traffic to experience small queueing delays, the utilisation of this traffic type is kept fairly low, with correspondingly small queue occupancy levels. High utilisations for the loss sensitive traffic can be achieved at the same time however (leading to high network usage overall) provided that the buffer space allocated to the low priority buffers in the dual buffer implementation is sufficiently large. This approach therefore allows the network to meet the required QoS of both the delay sensitive traffic and loss sensitive traffic simultaneously whilst achieving high network utilisation — something that cannot be achieved using a single buffer implementation.

Hullett et al. [54] discuss basically this arrangement in a design for an ATM switching element they call the Dataswitch. A peak rate allocation scheme[4] is used for the high priority queue, while statistical multiplexing of data traffic is used with an early packet discard mechanism in the low priority buffer.

1.2.2 Connection Admission Control

One of the most important functions of network management is connection admission control, which aims to prevent congestion within the network by determining when new connections should be accepted by the network. Congestion occurs when the network is unable to meet the QoS requirements of its established connections [64].

In order to determine whether a new connection can be supported by the network without causing congestion, the CAC mechanism requires that the user describe the expected or predicted behaviour of the new connection through a *traffic descriptor*, and additionally the required quality of service through a set of QoS parameters. A traffic descriptor is made up of parameters that describe properties of the information flow on the proposed connection. The traffic parameters must be unambiguous and understandable to the user, and importantly must be able to be monitored by the UPC function of the network management [9]. Three possible candidates are:

Peak Cell Rate: This is the peak cell generation rate of the traffic, defined as the inverse of the minimum time between two cells. Equipment wishing to transmit information across the B-ISDN will be required to observe this limit on the generation of cells.

Average Cell Rate: The average cell rate is similarly defined as the number of cells generated by the connection divided by the duration of the connection. Since this quantity will not be known until the connection terminates, a suitable upper bound is generally used instead.

Burst Duration: Many types of traffic are bursty in nature. That is, they generate cells at or near their peak rate for periods of time but are silent (or transmitting at a significantly reduced rate) in between these 'bursts'. An example of this is the transmission of full motion video, which results in periodic bursts of data corresponding to the coding of each video frame. The burst duration traffic parameter is used to describe some measurable statistic (such as the mean or maximum length) of this behaviour.

[4] We discuss peak rate and statistical allocation in section 1 2.2.

Since the CAC operation is performed before the network can accept a new connection, it must be able to be implemented very quickly. This is especially true when the CAC mechanism needs to consider multiple network paths (such as when attempting to establish a new VPC). The basic operation of the CAC is to assume that the connection has already been accepted by the network, and then to analyse the performance of the network, as seen by each connection, to check that the QoS requirements are being met. If the QoS requirements of any connection (existing or new) are violated, the new connection is rejected (or a new route is investigated). The number of elements that must be investigated is potentially very high, which constitutes the main difficulty in implementing the CAC.

The simplest CAC method that a network can use is *peak rate allocation*. A new connection is only accepted if the sum of the peak rates of the new and existing connections at every point of the proposed network path are less than the capacity of the respective links or switching elements. Rather than explicitly calculating the perceived performance for each connection in this situation, this method relies on the network having been suitably dimensioned[5] so that the peak rate allocation method will guarantee the most stringent quality of service that the network supports.

Although the peak rate allocation method is extremely fast, it leads to poor network utilisation for bursty sources, since the average cell rate of these traffics is usually considerably less than their peak cell rate. Fortunately for these types of sources, it is often possible to allow more connections than suggested by peak rate allocation without violating the negotiated quality of service requirements. The method relies on the probability that there are more sources transmitting at the peak rate that can be accommodated by the link in any time interval being small enough to enable the network to still meet its performance objectives — a technique referred to as *statistical multiplexing*. The extra number of connections that can be accommodated by statistical multiplexing (called the statistical multiplexing gain) and hence higher utilisation of the network, is dependent on both the traffic parameters, and their relation to the network transmission capacities.

The use of statistical multiplexing to improve network utilisation means that the CAC mechanism must evaluate the performance of the network with the new connection present in order to judge whether QoS requirements would be satisfied by the new network configuration — something which can be avoided in the simple peak rate

[5] Network dimensioning is the process whereby the number, size, capacity and speed of the various network elements are chosen The same process that is used by the CAC to determine whether connections can be accepted, is generally used to dimension the network by considering the expected network usage, and then adjusting the parameters of the network in order to satisfy the projected QoS requirements.

allocation case. An alternative CAC method exists though, which is similar to peak rate allocation, but is based instead on the notion of *equivalent bandwidth* [38,86,91,145]. This method uses an estimate for the effective or equivalent bandwidth of each type of connection traffic and decides connection acceptance or rejection by requiring the sum of the equivalent bandwidths at each link or network node to be less than the available capacity.

The key to the success of this approach is in knowing the equivalent bandwidth quantity for each connection type before it is needed for actual CAC within a 'live' network. The equivalent bandwidth value is obtained from analysis of simple network arrangements having similar parameters to the actual network. From the number of traffic sources (usually identical) that this simple network can support and still satisfy the QoS criteria, a value for the equivalent bandwidth of the traffic is obtained. Safety margins can then be added to decrease the chance that this CAC implementation results in the wrong decision being made (acceptance of a connection which causes QoS degradation). Since these equivalent bandwidth calculations are performed 'off-line' they can be made as accurate or as complicated as desired, since long calculation times are not as important.

Measuring Network Performance

In both the peak rate allocation and equivalent bandwidth CAC methods, actual performance evaluation of the network is avoided, although it is required for the more general CAC approach. However, in dimensioning the network to use peak rate allocation, or in obtaining the equivalent bandwidths of the various traffic types, some degree of network performance analysis is still required, although not under the same time constraints.

The purpose of such analysis is to investigate how the network performs from the point of view of the quality of service requirements of the carried network traffic. That is, the analysis must determine quantities such as the average cell transfer delay and its variation, and the average cell loss probability on a per connection and per priority basis [9,64]. All three of these quantities are primarily governed by the behaviour and dimensioning of the buffers within the network[6]. Thus, the performance analysis of the network primarily consists of the queueing analysis of the various elements making up the network.

The main focus of this thesis is on the performance analysis of various queueing problems involving both single and dual buffer systems. The results are of importance to the implementation of connection admission control mechanisms, and hence to the manage-

[6] Although there will be a component of the delay that is determined by propogation delays on the chosen network path, this is not effected by the acceptance or rejection of new calls by the CAC

ment of the entire B-ISDN. In order to make accurate predictions of queue behaviour however, suitable models for the queue arrival processes are required, so we will look at the modelling of ATM traffic in the following section.

1.3 Modelling ATM Traffic

The construction of analytic or procedural models of telecommunications traffic, and the subsequent study of their queueing performance, belongs to the area of teletraffic analysis. There are three basic steps involved in this process — traffic characterisation, traffic modelling, and queueing analysis. While the greater part of this thesis is concerned with the last of these, we discuss the first two steps in the following.

1.3.1 Traffic Characterisation

Traffic generation in the ATM network can be characterised using a multilayered structure [29, 32] as illustrated in Figure 1.7. Each layer of this structure (call, burst, and cell) occurs over a different time scale, giving a multiresolution view of an ATM traffic stream. The call level describes the behaviour and incidence of new connections and has intervals and holding times measured over the time scale of seconds to minutes. The burst level indicates periods of high activity within the call and has a time scale of milliseconds. The cell level, which describes the actual pattern of ATM cell generation has a time scale in the order of microseconds (at 155.52 Mbps an ATM cell can be transmitted every 2.73 μsec).

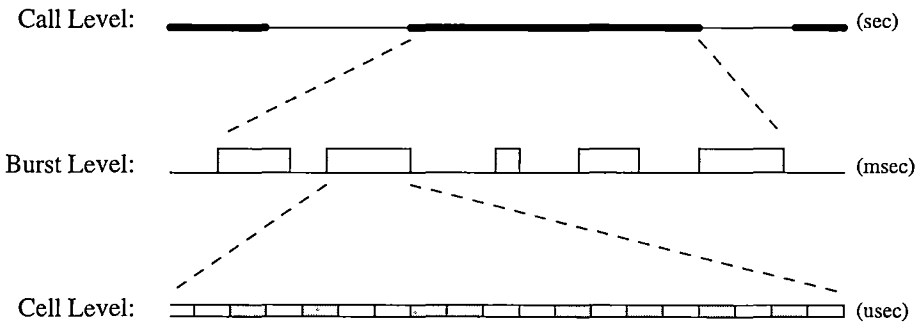

Figure 1.7: *Three time scales used to characterise the behaviour of ATM traffic.*

There are three basic locations for which traffic models can be characterised — at the source, at the network edge, and within the network. The traffic behaviour at the latter two locations is a direct consequence of the traffic generated at the first location, so we begin with identifying this process.

Traffic at the Source

The traffic source in this context is the originating point of ATM cells, which may be a single user terminal on a private ATM network, or the bridging point for a non-ATM network (e.g. N-ISDN or Ethernet LAN). Three generic types of traffic are frequently identified in the context of the B-ISDN (see for example [136]):

Voice Traffic: This is traffic originating from the equivalent of the analog telephone, and may be coded using 64 kbps PCM or 32 kbps ADPCM techniques. Direct PCM techniques result in a constant bit rate connection by continuous sampling of the voice source. To make use of statistical multiplexing, silence detection [58] can be used to prevent transmission of cells which contain little or no useful information, resulting in a bursty traffic process. Voice traffic is robust with respect to cell losses, but is sensitive to variations in transmission delay. Buffering at the receiving end can be used to improve this, but introduces an additional transmission delay to the connection.

Video Traffic: Video traffic is expected to originate from a range of services, from low quality videophone and video conferencing through to standard entertainment and high definition television. The video signal is composed of periodically generated 'frames' of information which describe the changing content of the displayed image, and as a result, the bursts of activity from video sources are also periodic. Because high compression rates are used to reduce the required bandwidth of the video traffic, it is moderately sensitive to cell losses. As with voice traffic, buffering is used to absorb some of variations in network transmission delays.

Data Traffic: Data traffic is a generic label for those communications requiring end-to-end lossless transmission. ATM does not provide this sort of transfer capability, so it is implemented by higher layer functions at the network edges (at the source and destination). As a consequence, a single cell loss can incur a heavy penalty since an entire high level protocol data unit (PDU), representing tens to hundreds of ATM cells, must be retransmitted.

Data traffic is inherently bursty in nature due both to the higher layer protocols that guarantee lossless transmission, and to the intermittent transmission requirements of most data applications. This factor, along with a high tolerance to long transmission delays, makes this traffic a prime candidate for statistical multiplexing.

An important consideration for network performance is the behaviour of these traffics at the cell level. For voice traffic, the generation of cells within a burst (or continuously if silence detection is not used) is governed by the coding process and the size of the ATM cell payload, resulting in constant time intervals between cells[7]. For video and data traffic, the cell generation process in not subject to the same coding restraints, but the traffic parameters negotiated with the network during connection establishment must still be observed, or cells may be discarded by the network UPC. Thus for data traffics, cells are generated at the agreed peak rate within a burst, even though a faster rate may be physically possible — a process referred to as *peak rate limiting.*

The same approach can also be used with video traffic, resulting in bursts of cells at the peak rate followed by a silent period until the next frame begins. In many cases however, the size of the frame of video data is known before transmission begins. Since the time between frames is fixed, it is then possible to adjust the cell rate of the video source on a per frame basis, so that the transmission of each frame is only just completed before the next one is ready. In this situation, the video source never becomes 'silent' but instead acts as a constant bit rate source with a periodically varying rate.

Traffic at the UNI

From the discussion above, the cell generation process within each burst (or over durations measured on a burst level time scale) will exhibit a fixed time interval between cells due to peak rate limiting. This will be the case at the source of the ATM traffic, but may not necessarily be the case at the UNI.

A single UNI may be shared by a number of ATM terminals or devices, or may be part of a private ATM network. In the first of these arrangements, traffic generated by each terminal (or source) is multiplexed with traffic from the other terminals before being passed to the UNI. In the same manner as within network switching elements, the multiplexer resolves the problem of possible simultaneous cell arrivals by buffering the excess cells. When more than one traffic source is being used, queueing in this buffer introduces variation in arrival times of the cells at the UNI. This effect is termed *cell delay variation* (CDV) and is a factor that must be accommodated by the UPC when determining compliance of a connection with its negotiated traffic parameters.

This CDV effect can become more pronounced when a private ATM network provides the access to the B-ISDN, since in addition to delays caused by multiplexing at the UNI destination, cells from each traffic source are queued within the switching fabric of the private network. Thus, although peak rate limiting is used at the traffic sources, the behaviour of the cell generation process at the UNI will exhibit variations in the intervals between cells.

[7] Assuming a full 48 byte payload, cells are generated approximately every 6 msec when using 64 kbps PCM.

Traffic within the Network

As the number of switching elements that cells pass through increases, the effect of CDV on the connection accumulates, making the characterisation of the traffic increasingly more difficult. Thus although we can characterise the behaviour of a particular type of traffic in a certain manner at its source, the behaviour of the traffic within the network is more difficult to describe.

Figure 1.8 shows the two primary operations that need to be considered in a packet switched network — the *merging* of those traffic streams taking the same network path, and the subsequent *splitting* of the merged stream at the next routing or switching point [107, 124]. In practise these two interrelated operations are difficult to describe, and only a few attempts have been made on their analysis (see [20,87,102] and references therein).

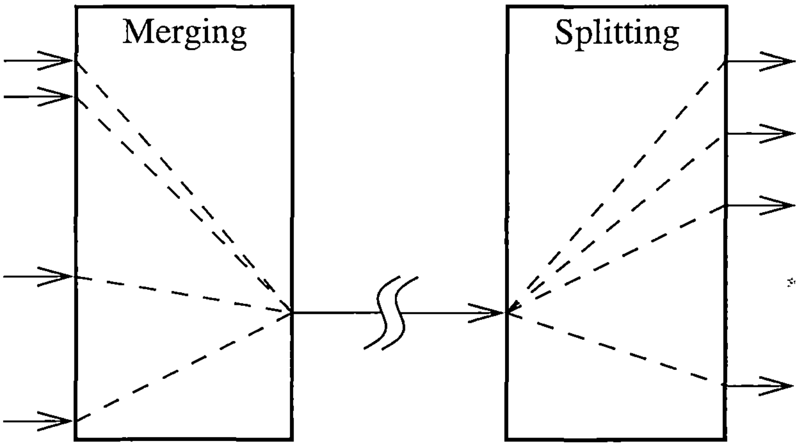

Figure 1.8: *Merging and splitting in network switching elements.*

An important consideration in regard to merging and splitting in the network is that traffic entering the switch buffers is no longer usefully identified on the basis of its originating sources. As an example, an output port of a particular ATM switch having 4 input ports may be carrying traffic from 100 different voice connections, yet the cell arrival process to the relevant output buffer actually only consists of 4 'sources' — the input ports of the switch. In order to describe this arrival process in terms of the original 100 sources would require identification of both the cumulative CDV effects for each traffic, and the correlations between arrivals from each source arising from multiplexing[8].

[8] Cconsider that if we identify whıch sources the currently arrıving 4 cells belong to, we have also effectively identified that the other 96 sources cannot be currently generating cells.

Traffic processes within the network will therefore be difficult to quantify purely on the basis of the characteristics of the sources that contribute to each network node, although some parameters such as the average transmission rate, will be easily identifiable.

1.3.2 Modelling Approaches

There are two basic ways in which the arrival processes within a network can be described. The first considers the times between events (such as an arrival to or a departure from a queue), while the second considers the events that occur at specific times. For example, we may either measure the intervals between arrivals of cells to a specific switch buffer, or we may count the number of cells arriving to the queue in a suitable time period. Both approaches have a certain validity, but for a particular type of problem and analytic method, one traffic modelling approach will be preferable to the other.

In this thesis we will use the event based (or event counting) approach to describe the behaviour of the network traffics, and consequently adopt a sympathetic analytic method in Chapters 2 to 5. Although interval based models are often able to describe a generic traffic source more accurately or more efficiently, the superposition of a number of these sources is considerably more difficult to express than with the alternative. Since we are particularly interested in queueing analysis within the heterogeneous traffic environment of a network switch, the use of models based on event counting methods is the better choice.

One of the convenient aspects of ATM based networks for teletraffic analysis is that the length of an ATM cell is constant, which means that its service time (or the time required to transmit the cell on a fixed speed output line) is deterministic. For modelling purposes, the cell service period is a natural choice for the time interval in which to count events such as arrivals and departures. This time unit will be referred to as a *slot*, so that time is said to be slotted or divided into equal length intervals.

We will assume that the input lines and output lines of all ATM switching elements are transmitting cells at the same speed. During one slot time, at most one queued ATM cell can be transmitted on the output line, and similarly at most one cell can arrive on each of the input lines. Thus each input line can be modelled using a binary process, which takes the value 1 when an arrival occurs in the current slot, and 0 otherwise.

Note that multiplexers and demultiplexers, which involve either low speed inputs or low speed outputs, can be described similarly by restricting the slots in which arrivals or service can occur. For example, a multiplexer may be described by forcing each arrival on an input line to occur separated at some multiple of T time slots, where T is the speed-up factor of the multiplexer (see for example [49]). Similarly, a demultiplexer can be described by only permitting service to occur once every T slots. We will look at this periodic service problem in more detail in Chapter 5.

Constructing the Model

Since we are using event counting as our basic approach, the traffic model must attempt to describe how the 0's and 1's, representing no arrivals and one arrival respectively, are generated over time by the observed traffic. This is usually done by assuming the traffic forms a stochastic process, which is then approximated by a suitable model. The traffic model parameters are adjusted in order to match the statistics of the model to the statistics of the original traffic[9]. With this achieved, analytical techniques can be used to obtain queueing performance results for the model. If the model is a good one, these performance results will closely match those that the real traffic would obtain in the same situation. Simulation is often used in this regard to provide confirmation or otherwise of the model accuracy.

The simplest stochastic event model is the *Bernoulli* arrival process, which generates an arrival in any slot with fixed probability p, and no arrival with probability $(1-p)$. The parameter p represents the average number of cells or arrivals generated by the model in each time slot. Thus, if we denote the equivalent statistic for the actual traffic by λ, then the modelling process simply requires $p = \lambda$. Because of the binary nature of the traffic process, the identification of this average arrival probability means that the marginal or steady state distribution of the traffic and the model are immediately identical.

The Bernoulli traffic model cannot capture the dynamics of the actual traffic arrival process in time (bursts and silences for example) however, because it treats each time slot independently of every other. In order to match the autocorrelation behaviour of the traffic, more sophisticated models are required. Two such models that are used in the queueing analysis of this thesis are presented in the following.

1.3.3 An Interrupted Bernoulli Model of ATM traffic

An *interrupted Bernoulli process* or IBP, is a process that alternates between generating arrivals according to a standard Bernoulli process, and generating no arrivals at all.

[9] An alternatıve that can be used when the queueıng performance of the actual traffic ıs known, ıs to adȷust the parameters of the model ın order to match thıs performance in the same queueing situation (see the discussion on output modelling ın Chapter 7 for example).

This model thus attempts to describe the burst level nature of the actual traffic process, while assuming that the cell arrivals within each burst are randomly distributed according to a Bernoulli process. In [142], Xiong et al. show that this assumption leads to an upper bound on the queueing behaviour of the actual peak rate limited model. However, given the cumulative effect of CDV on the peak rate limited model, the Bernoulli assumption should provide a very good approximation for the actual traffic process during the burst period.

If the IBP model is used to describe the behaviour of a single traffic source at the edge of the network the Bernoulli average arrival rate parameter can be directly equated to the negotiated peak cell rate of the connection. Within the network however, the IBP will be used to describe the arrival process from each input port, with the Bernoulli average arrival rate then determined by the merging and splitting probabilities at each node for the traffics that contribute to the modelled arrival process.

When the IBP is generating cells, it is said to be in an active state, while the alternate behaviour is referred to as the inactive or silent state. In its most general form, the IBP model can provide phase-type distributions[10] for the durations of the active and inactive states, and allow arbitrary correlations between successive durations. However, this level of complexity not only requires a large number of parameters to describe, but also results in intractable queueing problems. The most common form of IBP is one having independent geometric distributions for both the active and inactive periods, requiring only 3 parameters in total to describe the model. We will examine this IBP model in more detail in Chapter 3.

An interesting form of IBP is one having a phase-type distribution for the active periods, a geometric distribution for the silent periods, and importantly an average arrival probability in the active state of 1 (a cell is generated in every time slot for which the IBP is active). Although such a high peak arrival rate is very unlikely for traffics after splitting (switching), it is more likely for the merged cell stream at the output of the switch buffer [124]. The advantage of this traffic model is that closed form solutions for the average and variance of the buffer population exist. We present the derivation of the variance solution in Chapter 4.

Model Nomenclature

For convenience of identification, we will refer to the IBP model with geometrically distributed active and inactive periods by the name *geometric-geometric IBP*. A similar notation is adopted for the second model above, referring to it as a *phase-geometric*

[10] The phase-type distnbution is a general distribution in discrete-time — see Chapter 2 of [95] for an elementary discussion.

Binary process. Thus the hyphenated term in the name of the process denotes the distribution type for the duration of the active and inactive states, in that order, while the 'IBP' and 'Binary' terms indicate that the Bernoulli arrival process during the active state has a generation rate that may be less than 1, or is always equal to 1, respectively.

For ease of use, the hyphenated term for active and inactive distributions will usually be abbreviated in this thesis to *geom-geom* for the geometric-geometric term, and *phase-geom* for the phase-geometric term.

1.3.4 A Cyclic Arrival Model

In an ATM network providing peak rate limited connections to a homogenous mix of CBR traffic sources, transmission delays will be fairly constant, except when a new connection is accepted, or an existing connection is released. When either of these events occur, the arrangement and ordering of the traffic along the effected network paths will be altered, resulting in a different delays at each network queue, and hence a different overall transmission delay. If the lengths of the calls are long (in the order of minutes) then it is likely that the pattern of traffic arrivals at each queue will not change very frequently, and the network may be able to achieve a steady state condition between changes. This will also be the case for a limited mix of CBR traffic sources with different periods, provided that the least common multiple of the traffic periods (which defines the periodicity of the arrival pattern within the network) is small.

If bursty sources are also permitted to use the ATM network, the cell arrival patterns will change with the start and end of every burst. Since the burst time scale is several orders of magnitude smaller than the call level, any periodicity in the arrival patterns will be destroyed. However, if the CBR traffic is given HOL priority over the bursty traffic in an ATM network using dual buffers at each output port, the behaviour of the bursty traffic will be unable to effect the CBR traffic, and the possibility of steady state cell arrival patterns from these sources is again possible.

Thus it appears that in addition to a model of the general bursty arrival process (the IBP discussed above) it would be convenient to have a periodic model as well for the case where CBR periodic sources are present as the high priority delay sensitive traffic in a dual buffer ATM implementation. In order to distinguish this periodically varying cell pattern model from a simple periodic model that generates a single cell every period, we refer to it as a *cyclic* process or model. The cyclic process is described in more detail in Chapter 5.

1.4 Contributions of this Thesis

The focus of this thesis is on the discrete-time performance analysis of single and dual buffer queueing systems in an ATM environment. Two aims were established to provide direction to the work:

1. Development of exact analytical solutions to queueing problems involving heterogeneous mixtures of sources.
2. Investigation of the accuracy of faster approximate solutions to the same problems.

In pursuing these aims, this thesis makes a number of contributions to the body of literature in this field that can be summarised as:

- The development of some basic probability generating function based queueing theory in Chapter 2, providing explicit solutions for the average and variance of an infinite buffer, single server queue fed by a generic Markov modulated batch arrival process. Although this theoretical development is not new of itself, it provides a framework for the chapters that follow that has not been equivalently expressed in the literature.
- Presentation of exact numerical solution implementations for the average and variance of an infinite buffer queue fed by two-state IBP sources in Chapter 3, and by two-state IBP sources and a single cyclic source in Chapter 5. In particular, the difficulties in obtaining exact solutions for the systems of linear equations that must be solved in order to realise the numeric solutions was investigated. In the cyclic source problem, an adaptive solution method was proposed for finding approximate results when exact ones are not achievable.
- The proposal and investigation of a new and extremely accurate approximation technique in Chapter 3 that is faster than the probability generating function based solution method when 4 or more sources are considered, and is capable of providing solutions to problems that would otherwise be intractable using current techniques. The approximation method was also shown to provide similar accuracy for the cyclic source problem of Chapter 5, and may provide a means for solving even more complex problems.
- The development in Chapter 4 of a closed form solution for the population variance of a queue fed by phase-geometric distributed binary sources. This solution adds to the 1990 result for the average queue population of this problem by Neuts [98].

- Investigation of a new loss approximation for single buffer queues in Chapter 6 that improves on the commonly accepted result, where loss probabilities are given directly by the queue population's tail probability distribution.

In addition to these main contributions, several other interesting results are:

- The identification in Chapter 2 of a relationship between dual and single buffer systems that directly leads to a simpler method for obtaining the solution of interrupted service queues when the interruption process is known. As an example, closed form solutions are presented for the population average and variance of a queue with random arrivals and phase-geometric service interruptions.
- Development of the closed form solution for the variance of the queueing delay for a queue with random service interruptions and random service in Chapter 6. Even when the population average and variance for an interrupted service queue are known, obtaining the variance of the delay is not straightforward.
- The empirical observation in Appendix B that the best average queueing delay that any independent traffic class can achieve in a single buffer queueing system is given by the average queue population. Some theoretical support for this claim is obtained when the traffic class is a periodic process.

This thesis does not attempt to determine empirically whether the models presented and analysed here are accurate descriptions of real world traffic, since the emphasis of the work is on the queueing results that have been obtained using these models. Although the models are flexible enough to exhibit various levels of short and long term autocorrelation, they may not adequately capture very long term autocorrelations that fall outside the settling times of the queueing systems that make up the network (such as autocorrelations arising from call connections and disconnections).

However, since these models are based on the mechanics of the splitting and merging operations that occur within ATM networks, they are expected to exhibit a fairly high level of accuracy in practice, at least on time scales commensurate with the settling times of the queueing systems being studied. Consequently, the author believes that this thesis forms a solid base for further research in the areas of both queueing theory and performance analysis, and for subsequent investigation into the actual accuracy of the traffic models presented.

Chapter 2

General Theory for Queue Population Analysis

In this chapter we develop a general analytic solution for the average and variance of the population of an infinite buffer discrete-time G/D/1 queue fed by a number of batch Markov arrival process, based on a probability generating function approach. The results will be used in Chapters 3, 4, and 5 where more specific arrival processes are considered.

In section 2.1 we review some of the solution methods used in the literature before moving on to develop the basic probability generating function theory for the queue population in section 2.3. In sections 2.4 through to 2.7, this basic theory is then developed into a general solution framework for the average and variance. In addition to this single buffer analysis, consideration is given to multiple buffer queueing systems in section 2.9. In particular we find that population analysis of dual buffer (priority) based queueing systems can for the most part be achieved using only single buffer theory. A simple example is provided to show how this is done.

2.1 A Quick Review of Solution Methods

Solution methods for queueing problems fall into two broad areas — exact methods and approximate methods. Approximate methods are usually considerably faster to implement, although with (sometimes greatly) reduced accuracy. In some instances, approximate methods must be used, since exact methods cannot be applied due to the constraints of computer time, memory, or more often than not, numeric precision. In any case, the question of the accuracy of the actual traffic model must contribute to the decision as to which approach is best.

2.1.1 Exact Methods

There are primarily two classes of techniques available for solving for the moments and/or distribution of the population of a queueing system subject to arrivals from Markov modulated sources. The first class is based on a probability generating approach, and is discussed in detail in this chapter. Its use hinges on being able to obtain a vector describing the probability that the system is empty (or that the queue is empty immediately before service) called the empty system probability vector. A common approach to generating this vector uses the fact that the probability generating function of the queue population must be analytic within the unit circle, requiring the poles of the relevant equation to correspond to its zeros. The poles can only be found numerically, and require that the eigenvalues of the transition probability generating matrix of each source be known in algebraic form — which restricts the applicability of this technique to small transition matrices [79,121,126]. An additional complication is that the pole locations are in general complex, resulting in decreased solution accuracy and stability with increasing numbers of sources, or source dimension.

To overcome the limitations of this approach, Neuts presented a geometric matrix analytic approach [97] to finding the empty system vector. Various methods [94,111,112, 116] have been proposed for efficient computation of the relevant matrix components required for evaluation of the vector, but although these methods are very stable, they require considerable amounts of CPU time, usually more so than the previous method of pole and zero evaluation[1].

The second class of queue analysis techniques, which requires the queue to have a finite buffer space, is based on describing the entire queueing system (including the queue population) as a Markov chain. The queueing system is expressed in matrix form, and its invariant probability vector (describing the steady state distribution of the queue population) is obtained either by direct matrix inverse, or by iteration until convergence to some satisfactory limit is achieved. The iterative approach has some significant advantages, and is discussed in more detail in Appendix E.

[1] In order to obtain the empty system probability vector usıng the matrix-geometrıc method, the solution of m linear equations in m unknowns ıs requıred, just as for the pole–zero method, where m is the number of states ın the overall arrival process. However, whereas the pole–zero method requires m numerıcal solutıons of eıgenvalue and vector problems to obtaın the lınear system parameters, the matrix geometric method requıres large numbers of multiplications and addıtıons wıth $m \times m$ matrıces [94]. For larger m, this extra computational requirement means that the matrix geometric method can be consıderably slower than its pole–zero counterpart Support for this can also be found in [89]

One alternative method that provides much the same result, but can be applied to infinite buffer analysis, is based on the matrix geometric approach above. Once the initial empty system vector is obtained, the vector describing the distribution of the queue population can be found recursively (see [112,116] for a suitable method). Again however, this method is very CPU intensive, and will probably not provide any particular benefit over a direct or iterative solution for small numbers of sources.

The first class of techniques has the advantage that, for a combined arrival process having a small state space, the moments of the queue population can be quickly found. The primary advantage of the second class of approach is that it is always stable, can provide a limited amount of information on the transient behaviour of the system, and perhaps more usefully, can accurately describe the finite buffer performance. Both classes however suffer from the problem that the state space increases exponentially with the number of sources — the so called 'curse of dimensionality'. For example, the total state space of an arrival process formed from the superposition of N non-identical two-state processes will be 2^N. For small N (say less than 8) this will be quite manageable, but it is desirable to be able to analyse problems having up to 16 or 32 sources, since these are the sizes of switches currently being manufactured for the ATM network (see Table 1.4 in Chapter 1). Clearly neither the direct nor the iterative methods will be practical for these types of problems, and approximate methods are required.

In this thesis we will concentrate on the probability generating function approach for exact analytic results, using the method of matching the zeros and poles, to determine the empty system probability vector. For confirmation of the accuracy and behaviour of the analytic approach, and to obtain exact results for finite buffer problems, we will use the iterative numeric method.

2.1.2 Approximate Methods

There are three classes of approximation techniques commonly used in queueing theory — fluid flow (or diffusion) methods, geometric tail approximation methods, and model approximation methods. We will not discuss the first technique in any detail here, but direct the interested reader to [28] and [99], and in particular to [7] for a discussion more directly applicable to the on-off source problem.

Geometric tail approximations derive from the principle that the probability that the steady state population is greater than some value t is well described by a geometric distribution for large t for a wide class of queueing problems. In particular [22] shows that any queueing system having a probability generating function that can be expressed as a rational function has this property, while [96] proves its existence for any queueing system if the underlying Markov chain is finite, aperiodic, and irreducible. In particular Sohraby reports in [120, 121] that the decay factor for the geometric tail is given by the solution to the equation $z - \delta(z) = 0$ where $\delta(z)$ is the Perron–Frobenius eigenvalue of the arrival process. Application of this result to queueing problems with many sources will be discussed in Chapters 3. We will also consider the geometric tail approximation in Chapter 6 in the context of finite buffer queueing problems.

Model approximation methods involve describing the superposition of the arrival processes by some more tractable model. This often introduces a second level of approximation, since the individual sources are usually already modelled by some simplified process. Some well known examples are the approximation of the superposition of on-off sources by Markov modulated Poisson processes or MMPPs (see [44, 143] and the references therein), the superposition of IBPs by renewal processes [85, 92, 101, 140], and the superposition of a range of autocorrelated arrival processes by Gaussian approximations [2, 3]. A more recent approach is to model large numbers of sources using Fractal models [25, 26]. Although all of these methods allow the queueing system to be solved more easily, there is often a good deal of computation required in generating the parameters of the simplified model from the original source models, particularly as there are often many ways in which those parameters can be assigned [44, 143].

Other approximation methods exploit various properties of the particular queueing system being modelled, and are hard to classify in a general form. We will encounter this in Chapters 3 and 5, where one method (the kth order approximation) is discussed in more detail.

2.2 Basic Assumptions

The approach used in this and subsequent chapters for the analysis of queueing problems assumes that time is divided equally into discrete units (called slots) equal in duration to the time required to transmit a cell on the outgoing link of the observed buffer (the cell service time). The buffer is considered only to hold those cells waiting for service, and not the cell (if any) currently receiving service. In queueing theory terms, the number of cells in the buffer is therefore the *queue population*, rather than the more common (and perhaps more realistic) *system population*. We have chosen this less common alternative because, with fixed service times, variations in transmission delays are caused only by the variable times that cells spend waiting for service.

The buffer is assumed to receive service at the slot boundaries, with new arrivals being admitted to the buffer during the time slot. Queueing delays are measured in terms of the number of whole time slots that a cell spends in waiting. Thus an arrival to an empty buffer is considered to have a zero queueing delay, although service may not actually begin for some fraction of a time slot. For deterministics service system (such as ATM) this extra delay will be relatively constant for every arrival and hence we will not be concerned with this issue.

2.3 Queueing Analysis for Markov Modulated Sources

In this section we will construct a general solution framework to obtain the average and variance of the queue population of the discrete time infinite buffer G/D/1 queue, which receives arrivals from N independent batch Markov arrival processes (D-BMAPs). Each source is described by a number of states, for which the arrival process from each source is dependent only its state in the current time slot, and on nothing else. The transitions between states is governed by a discrete time Markov process, with the assumption that the sources change state at the beginning of a time slot, and then generate the relevant arrivals for that time slot according to their new state.

Let $\mathbf{A}_i$ be an $m_i \times m_i$ stochastic matrix that describes the state to state transitions of source i, where m_i is the *minimum* number of Markov states required to describe the behaviour of source i. Let $\mathbf{P}_i(z)$ describe a diagonal matrix of probability generating functions describing the arrival process of source i in each state. The combined matrix $\mathbf{A}_i\mathbf{P}_i(z)$ is referred to as the transition probability generating matrix, and describes both the state to state transitions of source i and the arrivals generated by that source. In addition, define the invariant probability vector of source i by $\boldsymbol{\mu}_i$ where $\boldsymbol{\mu}_i\mathbf{A}_i = \boldsymbol{\mu}_i$ so that $\boldsymbol{\mu}_i$ describes the steady state probabilities for the source being in each of its m_i states.

If m_i is the minimum number of states required to describe source i, then $\mathbf{A}_i$ will be an irreducible stochastic matrix, and hence will have an inverse. Consequently, using the similarity transform of standard linear algebra [36] it is possible to write

$$\mathbf{A}_i\mathbf{P}_i(z) = \mathbf{G}_i(z)\boldsymbol{\Omega}_i(z)\mathbf{H}_i(z) \tag{2.1}$$

where $\boldsymbol{\Omega}_i(z)$ is a diagonal matrix of the eigenvalues of $\mathbf{A}_i\mathbf{P}_i(z)$ arranged so that the eigenvalue having a value of 1 at $z = 1$ is in the top left corner. The matrix $\mathbf{G}_i(z)$ is formed from the right-hand (column) eigenvectors of $\mathbf{A}_i\mathbf{P}_i(z)$ corresponding to the eigenvalues in $\boldsymbol{\Omega}_i(z)$. Similarly, matrix $\mathbf{H}_i(z)$ is formed from the left-hand (row) eigenvectors of $\mathbf{A}_i\mathbf{P}_i(z)$, and is also given by $\mathbf{H}_i(z) = \mathbf{G}_i(z)^{-1}$.

The overall arrival process to the queue can also be described by a D-BMAP of the kind described above for each source. Denote the $m \times m$ state to state transition matrix of this combined process by $\mathbf{A}$ with probability generating function matrix $\mathbf{P}(z)$. Since the N sources are independent, we have

$$\begin{aligned}\mathbf{AP}(z) &= \mathbf{A}_1\mathbf{P}_1(z) \otimes \mathbf{A}_2\mathbf{P}_2(z) \otimes \cdots \otimes \mathbf{A}_N\mathbf{P}_N(z) \\ &= \bigotimes_{i=1}^{N} \mathbf{A}_i\mathbf{P}_i(z) \end{aligned} \tag{2.2}$$

where $\otimes$ denotes the Kronecker product [34], and

$$m = \prod_{i=1}^{N} m_i \tag{2.3}$$

We can therefore also write

$$\mathbf{AP}(z) = \mathbf{G}(z)\mathbf{\Omega}(z)\mathbf{H}(z) \tag{2.4}$$

where

$$\mathbf{\Omega}(z) = \bigotimes_{i=1}^{N} \mathbf{\Omega}_i(z) \tag{2.5}$$

$$\mathbf{G}(z) = \bigotimes_{i=1}^{N} \mathbf{G}_i(z) \tag{2.6}$$

$$\mathbf{H}(z) = \bigotimes_{i=1}^{N} \mathbf{H}_i(z) \tag{2.7}$$

and where, as for the individual sources, $\mathbf{H}(z) = \mathbf{G}(z)^{-1}$ also.

2.3.1 The Queue Equation

Let $\mathbf{X}(z)$ describe a row vector of probability generating functions for the stationary distribution of the queue population seen *immediately after* service has occurred, conditioned on the states of $\mathbf{A}$. That is, the jth element of $\mathbf{X}(z)$ describes the probability generating function for the steady state distribution of the queue population, given that the last state of the overall arrival process described by $\mathbf{A}$ was j. Then, in a similar manner as demonstrated for marginally distributed arrivals in Appendix A, it is possible to construct the relation

$$\mathbf{X}(z)\left(z\mathbf{I} - \mathbf{AP}(z)\right) = (z-1)\,\mathbf{b} \tag{2.8}$$

where $\mathbf{b}$ is a row vector describing the stationary probabilities that the queue is empty immediately prior to service (called the 'empty system' probability vector) and $\mathbf{I}$ is the identity matrix. This is the queue equation.

The unconditional behaviour of the queue (i.e. irrespective of the state of the arrival process) is given by $\mathbf{X}(z)\mathbf{e}$ where $\mathbf{e}$ represents an appropriately sized[2] column vector with all components equal to 1. In addition, when $z = 1$ the queue equation becomes

$$\mathbf{X}(1) = \mathbf{X}(1)\mathbf{A} \tag{2.9}$$

which shows that $\mathbf{X}(1) = \boldsymbol{\mu}$, the stationary probability vector of $\mathbf{A}$.

2.3.2 Alternative Form of the Queue Equation

An alternative form of the queue equation exists that is based on equation (2.8), but uses the form of $\mathbf{AP}(z)$ given by equation (2.4) to provide a more explicit representation of $\mathbf{X}(z)$. This result is based on that presented in [79].

On rearranging equation (2.8) we get

$$\mathbf{X}(z) = \left(1 - z^{-1}\right) \mathbf{b} \left(\mathbf{I} - z^{-1}\mathbf{AP}(z)\right)^{-1} \tag{2.10}$$

so that, using the result

$$\sum_{i=0}^{\infty} \mathbf{M}^{i} = (\mathbf{I} - \mathbf{M})^{-1} \tag{2.11}$$

for some matrix $\mathbf{M}$ (see Theorem F.3 in Appendix F) gives

$$\begin{aligned} \mathbf{X}(z) &= \left(1 - z^{-1}\right) \mathbf{b} \sum_{n=0}^{\infty} z^{-n} \left(\mathbf{AP}(z)\right)^{n} \\ &= \left(1 - z^{-1}\right) \mathbf{b} \sum_{i=0}^{\infty} z^{-n} \mathbf{G}(z) \boldsymbol{\Omega}(z)^{n} \mathbf{H}(z) \end{aligned} \tag{2.12}$$

Now, by spectral decomposition, we can write

$$\mathbf{G}(z)\boldsymbol{\Omega}(z)\mathbf{H}(z) = \sum_{j=0}^{m-1} \omega_j(z) \mathbf{g}_j(z) \mathbf{h}_j(z) \tag{2.13}$$

where $\omega_j(z)$ is the jth eigenvalue of the $m \times m$ matrix $\mathbf{AP}(z)$, and $\mathbf{h}_j(z)$ and $\mathbf{g}_j(z)$ are the corresponding left and right eigenvectors, obtained from the relevant row of $\mathbf{H}(z)$ and column of $\mathbf{G}(z)$. Thus equation (2.8) becomes

$$\mathbf{X}(z) = \left(1 - z^{-1}\right) \mathbf{b} \sum_{j=0}^{m-1} \left(\sum_{n=0}^{\infty} z^{-n} \omega_j(z)^{n} \right) \mathbf{g}_j(z) \mathbf{h}_j(z) \tag{2.14}$$

which, making use of the infinite sum of a geometric series, finally yields

$$\mathbf{X}(z) = (z - 1)\, \mathbf{b} \sum_{j=0}^{m-1} \frac{\mathbf{g}_j(z)\mathbf{h}_j(z)}{z - \omega_j(z)} \tag{2.15}$$

[2] Unless explicitly required, the sizes of the identity matrix $\mathbf{I}$ and the unit column vector $\mathbf{e}$ will not be specified any further. It can be assumed that they agree dimensionally with those components of the expressions to which they belong

which is the alternative form of the queue equation.

We will note here that the convention adopted throughout this thesis is for the eigenvalue taking the value of 1 at $z = 1$ is identified by $j = 0$, so that $\omega_0(1) = 1$. Given that $\mathbf{A}$ is irreducible, there will exactly one eigenvalue having this property, so this convention is not ambiguous.

2.3.3 Solving for the 'Empty System' Probability vector

In order to obtain $\mathbf{X}(z)$, the 'empty system' probability vector $\mathbf{b}$ must be found. We will use here the fact that, for a stable queue, $\mathbf{X}(z)$ is analytic within the unit circle ($|z| \leq 1$), and hence the poles of $\mathbf{X}(z)$ must also correspond to its zeros. From the form of the alternate queue equation (2.15), the poles of $\mathbf{X}(z)$ are given by the solution of

$$z - \omega_j(z) = 0 \tag{2.16}$$

for each j. Also, since $\mathbf{X}(z)$ must be real for real z, the poles must either be real or occur in complex conjugate pairs. In general, the poles are complex and can only be found using numeric search algorithms (such as the Newton–Raphson method [109]) since the analytic expression of (2.16) is, in most practical cases, transcendental in z.

If it is known apriori that the poles will be real (such as for the cases in [83, 146] and for the IBP analysis of Chapter 3), the simpler, although not neccesarily faster, but more stable bisection algorithm can be used to solve for the pole locations instead of the Newton–Raphson method.

Since $\mathbf{X}(z)$ is required to be analytic in the unit circle, we must also have zeros at each of the pole locations. Denote the pole that solves equation (2.16) by z_j^*. Then, for each pole $z_j^* \neq 1$, a sufficient condition to keep $\mathbf{X}(z)$ analytic is that $\mathbf{b}\mathbf{g}_j(z_j^*) = 0$. For the special case where $z_0^* = 1$ (which occurs for $j = 0$ by our stated convention), the fact that $\mathbf{X}(z)$ is analytic is not sufficient to establish a boundary condition. However, from the first derivative of equation (2.8) evaluated at $z = 1$, we obtain, on postmultiplication by e that

$$\mathbf{b}\mathsf{e} = 1 - \lambda \tag{2.17}$$

where λ is the steady state average number of arrivals to the queue in each time slot. Since $\mathbf{g}_j(z)$ can only equal a scalar multiple of the column sum vector e when $\omega_j(z) = 1$, this relation can only occur as a boundary condition for the pole at $z = 1$. Thus a linear problem consisting of m equations in m unknowns if formed, which can be solved using standard linear algebra techniques.

In practice, solving for the vector $\mathbf{b}$ involves finding each of the m poles of the queue equation, and then constructing and solving a linear system of m equations in m unknowns. It is the solution of this linear system that usually presents the most difficulties, particularly since m increases geometrically with the number of sources.

2.4 Solution for $\mathbf{X}'(1)$ based on the Alternative Queueing Equation

The first derivative of $\mathbf{X}(z)$, evaluated at $z = 1$ gives a vector describing the average queue population conditioned on the states of $\mathbf{A}$. From equation (2.15)

$$\begin{aligned}\mathbf{X}'(z) &= \mathbf{b}\sum_{j=0}^{m-1}\frac{\mathbf{g}_j(z)\mathbf{h}_j(z)}{z-\omega_j(z)} + (z-1)\,\mathbf{b}\sum_{j=0}^{m-1}\frac{\mathbf{g}_j'(z)\mathbf{h}_j(z)+\mathbf{g}_j(z)\mathbf{h}_j'(z)}{z-\omega_j(z)} \\ &\quad -(z-1)\,\mathbf{b}\sum_{j=0}^{m-1}\frac{\left(1-\omega_j'(z)\right)\mathbf{g}_j(z)\mathbf{h}_j(z)}{\left(z-\omega_j(z)\right)^2} \qquad (2.18)\end{aligned}$$

This relation cannot be evaluated directly at $z = 1$ because $\omega_0(1) = 1$, resulting in indeterminate quantities. Instead, the limit as z approaches 1 is taken for $\mathbf{X}'(1)$, giving

$$\begin{aligned}\mathbf{X}'(1) &= \mathbf{b}\sum_{j=1}^{m-1}\frac{\mathbf{g}_j(1)\mathbf{h}_j(1)}{1-\omega_j(1)} + \lim_{z\to 1}\frac{z-1}{1-\omega_0(z)}\left(\mathbf{b}\mathbf{g}_0'(z)\mathbf{h}_0(z)+\mathbf{b}\mathbf{g}_0(z)\mathbf{h}_0'(z)\right) \\ &\quad +\lim_{z\to 1}\frac{1-\omega_0(z)+(z-1)\,\omega_0'(z)}{\left(z-\omega_0(z)\right)^2}\mathbf{b}\mathbf{g}_0(z)\mathbf{h}_0(z) \qquad (2.19)\end{aligned}$$

where the expression is somewhat simplified by taking into account that $\omega_j(1) \neq 1$ if $j \neq 0$. For convenience, we will write

$$\mathbf{f}_1(z) = \frac{z-1}{1-\omega_0(z)}\left(\mathbf{g}_0'(z)\mathbf{h}_0(z)+\mathbf{g}_0(z)\mathbf{h}_0'(z)\right) \qquad (2.20)$$

and

$$\mathbf{f}_2(z) = \frac{1-\omega_0(z)+(z-1)\,\omega_0'(z)}{\left(z-\omega_0(z)\right)^2}\mathbf{g}_0(z)\mathbf{h}_0(z) \qquad (2.21)$$

so that

$$\mathbf{X}'(1) = \mathbf{b}\sum_{j=1}^{m-1}\frac{\mathbf{g}_j(1)\mathbf{h}_j(1)}{1-\omega_j(1)} + \lim_{z\to 1}\mathbf{b}\mathbf{f}_1(z) + \lim_{z\to 1}\mathbf{b}\mathbf{f}_2(z) \qquad (2.22)$$

To evaluate the first limit, write z as $z = 1-\epsilon$ for some very small ϵ. The first order Taylor series of $\omega_0(1-\epsilon)$ around $\epsilon = 0$ is

$$\omega_0(1-\epsilon) = 1-\epsilon\omega_0'(1)+\epsilon^2 R_2 \qquad (2.23)$$

for some remainder function R_2, where we have made use of $\omega_0(1) = 1$. Then

$$\lim_{z\to 1}\mathbf{f}_1(z) = \lim_{\epsilon\to 0}\frac{-\epsilon\left(\mathbf{g}_0'(1-\epsilon)\mathbf{h}_0(1-\epsilon)+\mathbf{g}_0(1-\epsilon)\mathbf{h}_0'(1-\epsilon)\right)}{1-\epsilon-\omega_0(1-\epsilon)}$$

$$
\begin{aligned}
&= \lim_{\epsilon\to 0} \frac{\mathbf{g}_0'(1-\epsilon)\mathbf{h}_0(1-\epsilon)+\mathbf{g}_0(1-\epsilon)\mathbf{h}_0'(1-\epsilon)}{1-\omega_0'(1)+\epsilon R_2} \\
&= \frac{1}{1-\omega_0'(1)}\left(\mathbf{g}_0'(1)\mathbf{h}_0(1)+\mathbf{g}_0(1)\mathbf{h}_0'(1)\right) \qquad (2.24)
\end{aligned}
$$

For the second limit problem a similar approach is used. The second order Taylor series expansion of $\omega_0(1-\epsilon)$ around $\epsilon = 0$ is

$$
\omega_0(1-\epsilon) = 1-\epsilon\omega_0'(1)+\frac{\epsilon^2}{2}\omega_0''(1)-\epsilon^3 R_3 \qquad (2.25)
$$

for some remainder function R_3. Similarly, the first order Taylor series expansion of $\delta_0'(1-\epsilon)$ around $\epsilon = 0$ is

$$
\omega_0'(1-\epsilon) = \omega_0'(1)-\epsilon\omega_0''(1)+\epsilon^2 R_2' \qquad (2.26)
$$

for some remainder function R_2'. Hence

$$
\begin{aligned}
\lim_{z\to 1}\mathbf{f}_2(z) &= \lim_{\epsilon\to 0}\frac{1-\omega_0(1-\epsilon)-\epsilon\omega_0'(1-\epsilon)}{(1-\epsilon-\omega_0(1-\epsilon))^2}\mathbf{g}_0(1-\epsilon)\mathbf{h}_0(1-\epsilon) \\
&= \lim_{\epsilon\to 0}\frac{\omega_0''(1)+2\epsilon\left(R_3-R_2'\right)}{2\left(1-\omega_0'(1)+\frac{\epsilon}{2}\omega_0''(1)-\epsilon^2 R_3\right)^2}\mathbf{g}_0(1-\epsilon)\mathbf{h}_0(1-\epsilon) \\
&= \frac{\omega_0''(1)}{2\left(1-\omega_0'(1)\right)^2}\mathbf{g}_0(1)\mathbf{h}_0(1) \qquad (2.27)
\end{aligned}
$$

giving

$$
\begin{aligned}
\mathbf{X}'(1) &= \mathbf{b}\sum_{j=1}^{m-1}\frac{\mathbf{g}_j(1)\mathbf{h}_j(1)}{1-\omega_j(1)}+\frac{1}{1-\omega_0'(1)}\left(\mathbf{b}\mathbf{g}_0'(1)\mathbf{h}_0(1)+\mathbf{b}\mathbf{g}_0(1)\mathbf{h}_0'(1)\right) \\
&\quad + \frac{\omega_0''(1)}{2\left(1-\omega_0'(1)\right)^2}\mathbf{b}\mathbf{g}_0(1)\mathbf{h}_0(1) \qquad (2.28)
\end{aligned}
$$

This result will be used later in the solution for the variance of the queue population.

2.5 Solution for the Average Queue Population

Let $\delta(z)$ denote the Perron–Frobenius eigenvalue for the matrix $\mathbf{AP}(z)$. From section C.3 of Appendix C we know that $\delta(1) = 1$, which corresponds only to $\omega_0(1)$ for the convention we have adopted, giving $\delta(z) = \omega_0(z)$. The left and right Perron–Frobenius eigenvectors of $\mathbf{AP}(z)$ corresponding to $\delta(z)$ are denoted by $\mathbf{u}(z)$ and $\mathbf{v}(z)$ respectively, and are formed under the constraints that $\mathbf{u}(z)\mathbf{v}(z) = 1$ and $\mathbf{u}(z)\mathbf{e} = 1$, with the easily proven results that

$$
\mathbf{u}(1) = \boldsymbol{\mu} \qquad (2.29)
$$

and

$$
\mathbf{v}(1) = \mathbf{e} \qquad (2.30)
$$

The relationship between the Perron–Frobenius eigenvectors and the general eigenvectors $\mathbf{g}_0(z)$ and $\mathbf{h}_0(z)$ is described in Appendix C.

Postmultiplying the queue equation (2.8) by $\mathbf{v}(z)$ gives

$$(z-\delta(z))\,\mathbf{X}(z)\mathbf{v}(z) = (z-1)\,\mathbf{b}\mathbf{v}(z) \tag{2.31}$$

The first derivative of equation (2.31) at $z=1$ yields

$$\delta'(1) = \lambda \tag{2.32}$$

while the second gives

$$2\,(1-\lambda)\left(\mathbf{X}'(1)\mathbf{e}+\boldsymbol{\mu}\mathbf{v}'(1)\right) - \delta''(1) = 2\mathbf{b}\mathbf{v}'(1) \tag{2.33}$$

making use of $\mathbf{X}(1)=\boldsymbol{\mu}$ and $\delta'(1)=\lambda$. The term $\mathbf{X}'(1)\mathbf{e}$ represents the average queue population, independent of the state of the arrival process, and is denoted by L_q. Hence

$$L_q = \frac{\mathbf{b}\mathbf{v}'(1)}{1-\lambda} + \frac{\delta''(1)}{2\,(1-\lambda)} - \boldsymbol{\mu}\mathbf{v}'(1) \tag{2.34}$$

However, from the first derivative of the constraint $\mathbf{u}(z)\mathbf{v}(z)=1$ we can show that

$$\mathbf{u}'(1)\mathbf{e} + \boldsymbol{\mu}\mathbf{v}'(1) = 0 \tag{2.35}$$

and from $\mathbf{u}(z)\mathbf{e}=1$ we have $\mathbf{u}'(1)\mathbf{e}=0$, so that $\boldsymbol{\mu}\mathbf{v}'(1)=0$ also. Thus

$$L_q = \frac{\mathbf{b}\mathbf{v}'(1)}{1-\lambda} + \frac{\delta''(1)}{2\,(1-\lambda)} \tag{2.36}$$

is the general solution for the average queue population of the observed infinite buffer queue.

Note that the same result would have been obtained simply by postmultiplying equation (2.28) by $\mathbf{e}$. The use of the Perron–Frobenius approach is deliberate however, since it provides the simplest method for obtaining the solution for the variance of the queue population.

2.6 Solution for the Variance of the Queue Population

The variance of the queue population, independent of the state of the arrival process is denoted by $\mathrm{Var}\,[L_q]$ and is given by

$$\mathrm{Var}\,[L_q] = \mathbf{X}''(1)\mathbf{e} + L_q - L_q^2 \tag{2.37}$$

where $\mathbf{X}''(1)\mathbf{e}$ is obtained from the third derivative of equation (2.31) evaluated at $z = 1$, giving on inspection

$$\operatorname{Var}[L_q] = \frac{\mathbf{bv}''(1)}{1-\lambda} - 2\mathbf{X}'(1)\mathbf{v}'(1) - \boldsymbol{\mu}\mathbf{v}''(1) + \frac{\delta'''(1)}{3(1-\lambda)} + \left(1 + \frac{\delta''(1)}{1-\lambda}\right) L_q - L_q^2 \quad (2.38)$$

Using the results of section C.3 we have

$$2\mathbf{X}'(1)\mathbf{v}'(1) + \boldsymbol{\mu}\mathbf{v}''(1) = \frac{2}{s(1)}\mathbf{X}'(1)\mathbf{g}_0'(1) - 2\frac{s'(1)}{s(1)}L_q + \boldsymbol{\mu}\mathbf{v}''(1) \qquad (2.39)$$

where $s(z)$ is a scalar function relating the general eigenvector solutions and the Perron–Frobenius eigenvectors.

From equation (2.28)

$$\begin{aligned}\mathbf{X}'(1)\mathbf{g}_0'(1) \quad = \quad & \mathbf{b}\sum_{j=1}^{m-1} \frac{\mathbf{g}_j(1)\mathbf{h}_j(1)\mathbf{g}_0'(1)}{1-\omega_j(1)} \\ & + \frac{1}{1-\lambda}\left(\mathbf{b}\mathbf{g}_0'(1)\mathbf{h}_0(1)\mathbf{g}_0'(1) + \mathbf{b}\mathbf{g}_0(1)\mathbf{h}_0'(1)\mathbf{g}_0'(1)\right) \\ & + \frac{\omega_0''(1)}{2(1-\lambda)^2}\mathbf{b}\mathbf{g}_0(1)\mathbf{h}_0(1)\mathbf{g}_0'(1) \qquad (2.40)\end{aligned}$$

where, using the results in Appendix C, we have

$$\mathbf{b}\mathbf{g}_0'(1)\mathbf{h}_0(1)\mathbf{g}_0'(1) = \frac{s'(1)^2}{s(1)}(1-\lambda) + s'(1)\mathbf{b}\mathbf{v}'(1) \qquad (2.41)$$

and from the second derivative of $\mathbf{h}_0(z)\mathbf{g}_0(z) = 1$, evaluated at $z = 1$ we obtain

$$\begin{aligned}2\mathbf{h}_0'(1)\mathbf{g}_0'(1) \quad &= \quad -\mathbf{h}_0''(1)\mathbf{g}_0(1) - \mathbf{h}_0(1)\mathbf{g}_0''(1) \\ &= \quad -\mathbf{u}(1)\mathbf{v}''(1) - \frac{2s'(1)^2}{s(1)^2} \qquad (2.42)\end{aligned}$$

hence

$$\mathbf{b}\mathbf{g}_0(1)\mathbf{h}_0'(1)\mathbf{g}_0'(1) = -\frac{s(1)(1-\lambda)\mathbf{u}(1)\mathbf{v}''(1)}{2} - \frac{s'(1)^2(1-\lambda)}{s(1)} \qquad (2.43)$$

and also

$$\begin{aligned}\mathbf{b}\mathbf{g}_0(1)\mathbf{h}_0(1)\mathbf{g}_0'(1) \quad &= \quad \mathbf{b}\mathbf{v}(1)\mathbf{u}(1)\left(s'(1)\mathbf{v}(1) + s(1)\mathbf{v}'(1)\right) \\ &= \quad (1-\lambda)\, s'(1) \qquad (2.44)\end{aligned}$$

thus

$$\begin{aligned}\mathbf{X}'(1)\mathbf{g}_0'(1) \quad = \quad & \mathbf{b}\sum_{j=1}^{m-1} \frac{\mathbf{g}_j(1)\mathbf{h}_j(1)\mathbf{g}_0'(1)}{1-\omega_j(1)} + s'(1)\left(\frac{\mathbf{bv}'(1)}{1-\lambda} + \frac{\delta''(1)}{2(1-\lambda)}\right) \\ & - \frac{s(1)}{2}\mathbf{u}(1)\mathbf{v}''(1) \qquad (2.45)\end{aligned}$$

and hence

$$2\mathbf{X}'(1)\mathbf{v}'(1) + \mu\mathbf{v}''(1) = \frac{2\mathbf{b}}{s(1)} \sum_{j=1}^{m-1} \frac{\mathbf{g}_j(1)\mathbf{h}_j(1)\mathbf{g}_0'(1)}{1 - \omega_j(1)} \tag{2.46}$$

Consider the derivative of $\mathbf{g}_0'(1)$ which will be given by

$$\mathbf{g}_0'(1) = \sum_{i=1}^{N} \left(\left(\bigotimes_{n=1}^{i-1} \mathbf{g}_{n,0}(1) \right) \otimes \mathbf{g}_{i,0}'(1) \otimes \left(\bigotimes_{n=i+1}^{N} \mathbf{g}_{n,0}(1) \right) \right) \tag{2.47}$$

where $\mathbf{g}_{n,0}(1)$ indicates the eigenvector of source n corresponding to the zeroth eigenvalue of that source, and $\mathbf{g}_{i,0}'(1)$ indicates the first derivative of this eigenvector, evaluated at $z = 1$. Hence

$$\mathbf{h}_j(1)\mathbf{g}_0'(1) = \sum_{i=1}^{N} \left(\left(\bigotimes_{n=1}^{i-1} \mathbf{h}_{n,r_{n,j}}(1)\mathbf{g}_{n,0}(1) \right) \otimes \mathbf{h}_{i,r_{i,j}} \mathbf{g}_{i,0}'(1) \otimes \left(\bigotimes_{n=i+1}^{N} \mathbf{h}_{n,r_{n,j}} \mathbf{g}_{n,0}(1) \right) \right) \tag{2.48}$$

where $r_{n,j}$ is a function that describes which eigenvalue for the nth source is indicated when the overall eigenvalue is j. From the basic properties of eigenvectors however, $\mathbf{h}_{n,r_{n,j}}(1)\mathbf{g}_{n,0}(1)$ will be 0 unless $r_{n,j} = 0$. This means that the entire summation will only be non-zero when the overall eigenvalue j is formed from the eigenvalues of one source only. That is

$$2\mathbf{X}'(1)\mathbf{v}'(1) + \mu\mathbf{v}''(1) = \frac{2\mathbf{b}}{s(1)} \sum_{i=1}^{N} \sum_{j=1}^{m_i-1} \frac{\mathbf{g}_{(i,j)}(1)\mathbf{h}_{i,j}(1)\mathbf{g}_{i,0}'(1)}{1 - \omega_{i,j}(1)} \tag{2.49}$$

where $\mathbf{g}_{(i,j)}(1)$ is given by

$$\mathbf{g}_{(i,j)}'(1) = \left(\bigotimes_{n=1}^{i-1} \mathbf{g}_{n,0}(1) \right) \otimes \mathbf{g}_{i,j}(1) \otimes \left(\bigotimes_{n=i+1}^{N} \mathbf{g}_{n,0}(1) \right) \tag{2.50}$$

which is the general right-hand eigenvector corresponding to the zeroth eigenvalue of each source, except for source i, for which the eigenvector corresponding to the jth eigenvalue is used.

Putting this together in the variance expression finally yields

$$\begin{aligned} \mathrm{Var}\left[L_q\right] &= \frac{\mathbf{b}\mathbf{v}''(1)}{1-\lambda} - \frac{2\mathbf{b}}{s(1)} \sum_{i=1}^{N} \sum_{j=1}^{m_i-1} \frac{\mathbf{g}_{(i,j)}(1)\mathbf{h}_{i,j}(1)\mathbf{g}_{i,0}'(1)}{1 - \omega_{i,j}(1)} \\ &\quad + \frac{\delta'''(1)}{3\left(1-\lambda\right)} + \left(1 + \frac{\delta''(1)}{1-\lambda}\right) L_q - L_q^2 \end{aligned} \tag{2.51}$$

which is the general solution for the variance of the queue population of the observed infinite buffer queue.

Alternatively, using $\mathbf{h}_j(1)\mathbf{e} = 0$ if $j \neq 0$ (which can be proved in the same manner that was used to show $\delta(1) = 1$ in section C.3) we can also write

$$2\mathbf{X}'(1)\mathbf{v}'(1) + \mu\mathbf{v}''(1) = 2\mathbf{b} \sum_{j=1}^{m-1} \frac{\mathbf{g}_j(1)\mathbf{h}_j(1)}{1 - \omega_j(1)} \mathbf{v}'(1) \tag{2.52}$$

and by the same reasoning as for the previous case,

$$2\mathbf{X}'(1)\mathbf{v}'(1) + \mu\mathbf{v}''(1) = 2\mathbf{b}\sum_{i=1}^{N}\sum_{j=1}^{m_i-1}\frac{\mathbf{g}_{(i,j)}(1)\mathbf{h}_{i,j}(1)}{1-\omega_{i,j}(1)}\mathbf{v}_i'(1) \tag{2.53}$$

so that

$$\begin{aligned}\mathrm{Var}\left[L_q\right] &= \frac{\mathbf{b}\mathbf{v}''(1)}{1-\lambda} - 2\mathbf{b}\sum_{i=1}^{N}\sum_{j=1}^{m_i-1}\frac{\mathbf{g}_{(i,j)}(1)\mathbf{h}_{i,j}(1)}{1-\omega_{i,j}(1)}\mathbf{v}_i'(1) \\ &+ \frac{\delta'''(1)}{3\left(1-\lambda\right)} + \left(1+\frac{\delta''(1)}{1-\lambda}\right)L_q - L_q^2 \end{aligned} \tag{2.54}$$

2.7 Mixing Autocorrelated and Random Sources

In some instances, the arrival process to the queue can be described by the superposition of N batch Markov arrival processes, as discussed in detail above, *and* some number of point processes having no autocorrelation (purely random or marginal processes). The superposition of these latter components can be described by a single probability generating function $p(z)$ which is independent of the Markov sources. Using the same notation as previously, the queue equation (2.8) then becomes

$$\mathbf{X}(z)\left(z\mathbf{I} - p(z)\mathbf{A}\mathbf{P}(z)\right) = (z-1)\,\mathbf{b} \tag{2.55}$$

with the alternate queue equation given by

$$\mathbf{X}(z) = (z-1)\,\mathbf{b}\sum_{j=0}^{m-1}\frac{\mathbf{g}_j(z)\mathbf{h}_j(z)}{z - p(z)\omega_j(z)} \tag{2.56}$$

The m poles of $\mathbf{X}(z)$ are therefore given by the solutions of

$$z - p(z)\omega_j(z) = 0 \tag{2.57}$$

Following the same approach for the derivations of the average and variance of the queue population as before yields

$$L_q = \frac{\mathbf{b}\mathbf{v}'(1)}{1-\lambda} + \frac{p''(1) + 2p'(1)\delta'(1) + \delta''(1)}{2\left(1-\lambda\right)} \tag{2.58}$$

and

$$\begin{aligned}\mathrm{Var}\left[L_q\right] &= \frac{\mathbf{b}\mathbf{v}''(1)}{1-\lambda} - \frac{2\mathbf{b}}{s(1)}\sum_{i=1}^{N}\sum_{j=1}^{m_i-1}\frac{\mathbf{g}_{(i,j)}(1)\mathbf{h}_{i,j}(1)\mathbf{g}_{i,0}'(1)}{1-\omega_{i,j}(1)} \\ &+ \frac{p'''(1) + 3p''(1)\delta'(1) + 3p'(1)\delta''(1) + \delta'''(1)}{3\left(1-\lambda\right)} \\ &+ \left(1 + \frac{p''(1) + 2p'(1)\delta'(1) + \delta''(1)}{1-\lambda}\right)L_q - L_q^2 \end{aligned} \tag{2.59}$$

where $\lambda = p'(1) + \delta'(1)$ is the combined average arrival rate to the queue.

These results can also be written in terms of the first three moments of the arrival process described by $p(z)$ as

$$L_q = \frac{\mathbf{b}\mathbf{v}'(1)}{1-\lambda} + \frac{\delta''(1)}{2\,(1-\lambda)} + \frac{p_2 - (1-2\lambda+2p_1)\,p_1}{2\,(1-\lambda)} \tag{2.60}$$

and

$$\begin{aligned} \mathrm{Var}\,[L_q] \;=\;& \frac{\mathbf{b}\mathbf{v}''(1)}{1-\lambda} - \frac{2\mathbf{b}}{s(1)} \sum_{i=1}^{N} \sum_{j=1}^{m_i-1} \frac{\mathbf{g}_{(i,j)}(1)\mathbf{h}_{i,j}(1)\mathbf{g}'_{i,0}(1)}{1-\omega_{i,j}(1)} \\ &+ \frac{\delta'''(1) + 3p_1\delta''(1)}{3\,(1-\lambda)} + \left(1 + \frac{\delta''(1)}{1-\lambda}\right) L_q - L_q^2 \\ &+ \frac{p_3 - 3\,(1+p_1-\lambda)\,p_2 + (2-3\lambda+3p_1)\,p_1}{3\,(1-\lambda)} \\ &+ \left(\frac{p_2 - (1-2\lambda+2p_1)\,p_1}{1-\lambda}\right) L_q \end{aligned} \tag{2.61}$$

where p_r denotes the rth moment of the marginal arrival process.

2.8 Other Methods for finding L_q and $\mathrm{Var}\,[L_q]$

One alternative to obtaining L_q and $\mathrm{Var}\,[L_q]$ using the approach covered in this chapter is to analyse the queueing system using Neuts' matrix geometric method [97]. As discussed at the start of this chapter, this method is computationally intensive, and will generally require longer execution times to provide the desired results. This method does provide the additional benefit of generating the exact probability distribution of the queue population however, and so may be more useful in some circumstances that might require this.

The moments of the queue population can also be found using the 'non-vanishing' roots of $\mathbf{X}(z)$ — those roots lying outside of the unit circle, that are not cancelled by zeros. In [83] Li and Sheng use just this approach to evaluate the moments of the queue population, and show that (with considerable more difficulty) the technique can also be used to obtain the entire distribution of the queue population. Since these roots are normally very difficult to obtain in the z domain, the authors transform the solution into the z^{-1} domain, reflecting the non-vanishing roots into the unit circle.

Although this approach appeared initially to provide a faster method for calculating the average and variance of the queue population, closer inspection shows that the algorithmic complexity of the root substitutions alone is at least equal to, if not greater than the complexity of the method presented in this chapter. Taking into account the additional difficulty of finding the generally complex roots in the first place, it would seem that the best choice for calculating the moments of the queue population is the method presented here.

2.9 Multiple Buffer Queueing Systems

In the preceding sections of this chapter we have developed a basic theory for the average and variance of the queue population for a discrete time system consisting of a single infinite capacity buffer and uninterrupted service. As we have discussed in section 1.2.1 however, we are also interested in dual buffer arrangements that can provide service priority to one of these buffers. Fortunately there is a relationship between these two queueing systems which we shall explore in this section. We start with the following general theorem.

Theorem 2.1 *Consider a single server queueing system providing deterministic service to B infinite capacity FIFO buffers. The distribution of the queue population across all B buffers is independent of B and of the service order and service priorities, provided that*

1. *The same set of arrival processes is considered.*
2. *Service selection between buffers is non pre-emptive.*
3. *The server never enters the idle state if there is an arrival waiting in* any *buffer.*

Proof. *It is easy to see that this system is work conserving if conditions 2 and 3 are met*[3]*, since queued arrivals can only leave the system by completing service, and service must continue if queued arrivals are present. In addition, selection for service between the queued arrivals must be independent of their service times because every arrival has the same service requirement. These two properties mean that the queueing system satisfies Kleinrock's conditions for the system or queue population to be independent of the service order — see pages 113 and 114 of [71]. Then since the use of multiple buffers only changes the effective service order if condition 1 is met, the proof is complete.* ■

Note that in this thesis we assume time is equally divided into slots, with one slot being equal in length to a single cell service period, and that services occur only on

[3] Condition 2 could be dropped if a pre-emptive resume strategy was used that did not violate the work conserving requirement. That is, if arrivals that had their service interrupted were able to recommence service without incurring any additional service requirements.

slot boundaries. This discrete-time assumption effectively means that all deterministic service priority systems must be non pre-emptive at the cell level, since service of a previously selected cell will always be completed by the time the currently selected cell begins service.

2.9.1 Dual Buffer Queueing Systems

We are interested in a dual buffer queueing system, where the buffers each have infinite capacity. Arrivals to this system have high or low priority, with the high priority arrivals queued in one buffer (the high priority buffer) and the low priority arrivals in the other. The service mechanism is such that the low priority buffer receives service only if the high priority buffer is empty at the start of a time slot. This means that the high priority buffer always receives service if it has queued arrivals. We will refer to this arrangement as a dual buffer system with *exhaustive service priority.* The term 'exhaustive' indicates that service of the high priority queue occurs until it is exhausted (empty).

The fact that the service of the high priority buffer is exhaustive means that the low priority buffer might go long periods without service. An alternative is to use a non-exhaustive approach, where some maximum number k of sequential services of the high priority buffer is allowed (called limited-k service). After this maximum, the low priority buffer receives a single service (limited-1) before service returns to the high priority buffer. As with the exhaustive method, if the high priority buffer is empty, service remains available for the low priority buffer, and vice-versa.

We will assume that the exhaustive service priority arrangement is used in the rest of this thesis, and that adequate service of the low priority queue is provided by considerate high priority traffic (using peak rate limited sources for example). In particular, the exhaustive service assumption means that arrivals to the high priority buffer receive uninterrupted service, which allows us to establish the following corollary from Theorem 2.1.

Corollary 2.2 *Consider a discrete-time, dual buffer FIFO queueing system, where the buffers each have infinite capacity. Arrivals to the system are assumed to have either high or low priority, with the high priority arrivals queued in one buffer and the low priority arrivals in the other. The high priority buffer is assumed to receive exhaustive service priority, so that the low priority buffer only receives service when the high priority buffer is empty.*

The average queue population for the low priority buffer is then given by the difference *in the average population between an uninterrupted service queue subject to arrivals firstly from both the low and high priority traffics together, and secondly from just the high priority traffic alone. That is*

$$L_{q_{low}} = L_{q_{both}} - L_{q_{high}}$$

where $L_{q_{low}}$ *denotes the average queue population for the low priority buffer,* $L_{q_{high}}$ *for the high priority buffer, and* $L_{q_{both}}$ *for both traffics sharing the same buffer (which receives uninterrupted service). A similar result applies to the variance, so that*

$$\mathrm{Var}\left[L_{q_{low}}\right] = \mathrm{Var}\left[L_{q_{both}}\right] - \mathrm{Var}\left[L_{q_{high}}\right] - \mathrm{Cov}\left[L_{q_{low}}, L_{q_{high}}\right]$$

where $\mathrm{Cov}\left[L_{q_{low}}, L_{q_{high}}\right]$ *denotes the covariance between the low and high priority buffer queue populations.*

Proof. *Let* x_n *denote the total number of queued arrivals at the beginning of the* n*th time slot (after service but before the next group of arrivals). Let* l_n *and* h_n *denote the corresponding quantities for the low and high priority buffers respectively, so that*

$$x_n = l_n + h_n \tag{2.62}$$

From standard probability theory we then have that

$$\mathrm{E}\left[x_n\right] = \mathrm{E}\left[l_n\right] + \mathrm{E}\left[h_n\right] \tag{2.63}$$

and

$$\mathrm{Var}\left[x_n\right] = \mathrm{Var}\left[l_n\right] + \mathrm{Var}\left[h_n\right] + \mathrm{Cov}\left[l_n, h_n\right] \tag{2.64}$$

Noting that the total queue population of the dual buffer system (represented by x_n*) must be equal to the queue population of a single buffer fed by the same high and low priority arrival processes together (Theorem 2.1) we have*

$$\mathrm{E}\left[x_n\right] = L_{q_{both}}$$

and

$$\mathrm{Var}\left[x_n\right] = \mathrm{Var}\left[L_{q_{both}}\right]$$

so that, with a small change in notation, the form of the corollary result follows.

Since the high priority buffer has exhaustive service priority, it can be regarded as receiving uninterrupted service, and can therefore be analysed separately to provide its queue population average and variance. Thus, the average queue population of the low priority buffer will be given by the difference between the averages for two single buffer systems, while the low priority variance is similarly given in terms of two single buffer variances and a covariance term. ■

This dual buffer problem belongs to a class of queueing systems referred to as a 'cyclic-service'[4] or 'polling' systems [128]. These systems consist in general of a single server that provides service to a number of queues in a cyclically sequential fashion, serving one queue for some period of time before moving on to the next. The mechanism used to decide when service on the current queue is to finish is termed the service or scheduling strategy. Exhaustive and limited-k are two examples of service strategies.

Polling systems generally present considerable analytical difficulties when it comes to obtaining measures of the queue population or delay. Consequently, the majority of the literature deals with either simple arrival models or special scheduling strategies (see for example [78, 90, 118, 132], or for a more extensive survey of the subject see [128]). One interesting result that applies to a wide range of these problems is a 'pseudo-conservation law' [13]. These laws are an exact expression for the weighted sums of the mean queueing delays (or the sum of the average queue populations for each buffer) and are more easily obtained than the individual components of the sum.

The combination of the exhaustive and limited-1 service strategies used in the dual buffer priority scheme discussed here has the advantage that the high priority buffer is effectively independent of the low priority buffer, and can be analysed separately using single buffer techniques. In addition, the average of the queue population for the low priority buffer can also be determined exactly using single buffer analysis techniques, reducing the complexity of this part of the analysis considerably.

In [144], Zhang proposed the equivalent of Corollary 2.2 for use with fluid flow queueing models. The author developed a solution for the covariance term when the arrival process is described by a single Markov chain. Unfortunately, we have not found an equivalent solution for the population analysis techniques considered in this thesis, although such a development appears quite probable.

Noting that the covariance term in the variance relation will always be greater than or equal to zero (the larger the high priority buffer, the larger low priority buffer will be on average also, since it will not receive service until the high priority buffer is empty) we see that ignoring this covariance term provides an upper bound to the variance equation. That is

$$\mathrm{Var}\left[L_{q_{\mathrm{low}}}\right] \leq \mathrm{Var}\left[L_{q_{\mathrm{both}}}\right] - \mathrm{Var}\left[L_{q_{\mathrm{high}}}\right] \tag{2.65}$$

The equality in this expression holds when the high priority buffer has an average queue population of zero (which must mean that the variance and covariance terms will also be zero). Although this situation may seem at first to be useless, we will see in the

[4] This is the common use for the term 'cyclic-service' In Chapter 5 we refer to a single buffer queueing system subject to periodically interrupted service as receiving cyclic service, but this is not the usual meaning.

following that it is actually quite useful for analysing one particular type of queueing problem.

2.9.2 Analysis of Interrupted Service Queueing Problems

Corollary 2.2 can be successfully applied to the analysis of single buffer queueing systems subject to service interruptions when this interruption process is known. The following corollary summarises its application.

Corollary 2.3 *The population average and variance of a discrete-time queueing system subject to service interruptions are given directly by the same quantities for an equivalent uninterrupted service queue subject to arrivals from both the original arrival process and from a process that generates a single arrival when service would normally not occur, and no arrivals when service would normally occur. Service interruptions are assumed to occur independently of the state of the queue buffer.*

Proof. *The interrupted service queue may be thought of as being the lower priority buffer in a dual buffer queueing system with exhaustive service priority. That is, service interruptions of the observed queue occur because a higher priority buffer receives service in those time slots. We can assume that the arrival process to this higher priority buffer is exactly equal to the service interruption process, so that a single high priority arrival occurs in any time slot that the service of the lower priority buffer is to be blocked.*

With at most one arrival in any time slot and exhaustive service priority, the high priority queue population at the beginning of any time slot (i.e. immediately after service) will always be zero. Hence the queue population term represented by h_n in the proof of Corollary 2.2 will be zero for all n, resulting in zero values for the average and variance of the high priority buffer, and a zero covariance between the low and high priority queue populations. Hence from Corollary 2.2

$$L_{q_{interrupted}} = L_{q_{both}}$$

and

$$\mathrm{Var}\left[L_{q_{interrupted}}\right] = \mathrm{Var}\left[L_{q_{both}}\right]$$

where $L_{q_{both}}$ and $\mathrm{Var}\left[L_{q_{both}}\right]$ represent the average and variance of a single buffer, uninterrupted service queue subject to arrivals from both the original arrival process and from an arrival process that describes the service process as indicated. ■

This result can also be arrived at by considering the service and arrival processes in discrete-time queues. Normally, when service occurs, the server removes a single waiting arrival (if there is one) from the queue. When service is blocked (interrupted) no waiting arrivals are removed from the queue, although new arrivals may still occur. In terms of the queue population, this is equivalent to there being a single additional arrival to the queue which is then removed by the normal service process. Since the analysis of the queue population does not require that the arrivals are served on a first come — first served basis, the queue population must be the same regardless of whether service was blocked or an additional arrival occurred and service was still available.

2.9.3 A Simple Interrupted Service Example

As an example of the application of this corollary, we will derive the average and variance of the queue population for an interrupted service queue with arrivals from a purely random sources (the arrivals have no autocorrelation). The service interruptions are such that the queue has the opportunity for service in successive time slots for periods described by a geometric distribution, followed by non-service periods having a phase-type distribution. The service and non-service durations are assumed to be independent.

Let λ, M_2, and M_3 describe the first three moments of the non-autocorrelated arrival process, and in addition let f describe the steady state probability that the queue receives service in any time slot. From the description of the service interruptions, a phase-geom binary process (which we will discuss in detail in Chapter 4) can be used to describe the additional arrivals that mimic the service interruptions, where the average arrival rate of this process is $1-f$.

From the results of Chapter 4, combined with the additional theory relating to mixing random and autocorrelated sources in section 2.7, we obtain directly that

$$L_q = \frac{M_2 + \lambda - 2\lambda f}{2\,(f-\lambda)} + \frac{\lambda\,(1-f)}{f-\lambda}\left(\frac{\gamma}{1-\gamma}\right) \tag{2.66}$$

where γ describes the autocorrelation parameter of the interruption arrival process (which is zero for random interruptions), given by

$$\gamma = 1 - \frac{2\eta_1}{f\,(\eta_1 + \eta_2)} \tag{2.67}$$

where η_r is the rth moment of the duration of the non-service periods. Similarly, the variance of the queue population is given by

$$\begin{aligned}\operatorname{Var}[L_q] \;=\; & \frac{4\,(f-\lambda)\,M_3 + 3M_2^2 + 6f\,(1-2f)\,M_2 + 2f\lambda + \lambda^2 - 12f\lambda^2\,(1-f)}{12\,(f-\lambda)^2} \\ & + \frac{f\,(1-f)\,(M_2-\lambda^2)\,\gamma}{(f-\lambda)^2\,(1-\gamma)} + \frac{f^2\,(1-f)^2\,\gamma}{(f-\lambda)^2\,(1-\gamma)^2} + \frac{2\lambda^2\,(1-f)}{(f-\lambda)\,(1-\gamma)}\end{aligned}$$

$$
\begin{aligned}
& -\frac{2\,(1-f)\,\lambda^2}{f\,(f-\lambda)\,(1-\gamma)^2}-\frac{(1-f)\,(3\lambda-f-5\lambda f+f^2)\,\gamma}{(f-\lambda)\,(1-\gamma)^2} \\
& +\frac{\lambda^2 f\,(1-f)}{3\,(f-\lambda)}\left(\frac{\eta_3}{\eta_1}-1\right) \qquad (2.68)
\end{aligned}
$$

For the case where the interruption durations are geometrically distributed, the variance becomes

$$
\begin{aligned}
\operatorname{Var}[L_q] \;=\; & \frac{4\,(f-\lambda)\,M_3+3M_2^2+6f\,(1-2f)\,M_2+2f\lambda+\lambda^2-12f\lambda^2\,(1-f)}{12\,(f-\lambda)^2} \\
& +\frac{f\,(1-f)\,(M_2-\lambda^2)\,\gamma}{(f-\lambda)^2\,(1-\gamma)}+\frac{f^2\,(1-f)^2\,\gamma}{(f-\lambda)^2\,(1-\gamma)^2} \\
& -\frac{(1-f)\,(3\lambda-f-5\lambda f+f^2)\,\gamma}{(f-\lambda)\,(1-\gamma)^2} \qquad (2.69)
\end{aligned}
$$

In [14], Bruneel extended the work of Hsu [48] and Heines [46] to analyse exactly this type of queueing problem, obtaining the probability generating function of the queue population as a result. Bruneel's result is obtained under the assumption that arrivals occur in continuous time, which increases the average and variance somewhat, but otherwise provides the same results as this example. The above approach, making use of Corollary 2.3, is considerably more straightforward however, and applies equally well to more complicated queueing problems.

2.10 Summary

In this chapter we presented some basic theory relating to the analysis of the queue population of a discrete-time infinite buffer G/D/1 queue fed by a number of batch Markov arrival processes. Following a brief discussion of exact and approximate methods for analysing these types of queueing systems, we have developed relations for the state conditioned average queue population vector, the unconditional average queue population, and unconditional queue population variance. These solutions are based on a probability generating function approach, and rely on being able to characterise the eigenvalues and eigenvectors of the Markov sources. Practical application of these results to a number of queueing problems will be covered in the chapters to follow.

In addition we have shown how the population analysis of multiple buffer queueing systems is related to the single buffer analysis. A dual buffer queueing system, where one buffer has exhaustive service priority is considered in particular. This high priority buffer can be treated as a single buffer system which allows the average and variance of its queue population to be obtained. The average population of the low priority queue is given by the difference between a single buffer system serving both high and low priority arrivals, and the high priority queue average. This method can also be applied to the variance, but yields only an upper bound. This result also extends to interrupted service queues, and provides a simple method for analysing these problems.

Chapter 3

Population Analysis for Geometric-Geometric IBP Arrival Models

The geom-geom IBP is an interrupted Bernoulli process with geometric distributions for the durations of its active and inactive periods. Figure 3.1 illustrates the basic Markov chain that describes the behaviour of this arrival process. During its silent (or *Off* state) the process generates no arrivals, while in its active (or *On* state), the process generates arrivals according to a Bernoulli process.

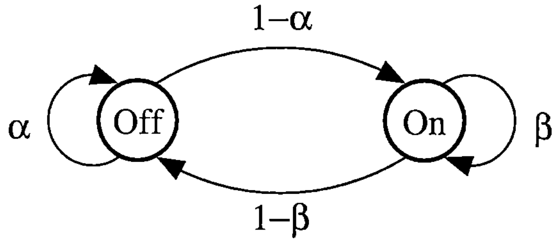

Figure 3.1: *Illustration of the transition probabilities for the Markov chain of a two-state IBP.*

This simple model has been widely considered in the literature, particularly for the analysis of switch and internal network behaviour. Analysis of the population of a queue fed by identical geom-geom IBP sources was considered in [51, 52, 142] and was extended in [83] to include multiple classes of identical sources. Approximate analyses for the queue population with heterogeneous sources were considered in [120] and [141]. Description of the output process of these queueing problems was considered in [30,134], and in [12] and [124] with particular regard to describing the splitting process.

It was noted in [12,124], and can be inferred from the results in [20], that the two-state model is somewhat inadequate to capture the behaviour of internal network traffics. Fonseca and Silvester conclude otherwise however in [30], a conclusion supported to some degree by [134].

There are two main reasons why IBP models of greater complexity than this simple two-state model have not generally been considered in the literature. The first is that the number of states required to describe a heterogeneous mix of N sources increases as m^N, where m is the number of states in the IBP model. Even for two-state IBP sources, this leads to exact solutions only being obtainable for less than 12 sources or so, and at a fairly high computational cost. The second reason is that the number of parameters required to describe the general m state source increases approximately as m^2.

One means of avoiding these problems is to assume that all the sources are identical or belong to classes of identical processes, such as in [126] or [68], where three state models are used. Another is to approximate the behaviour of the queue population. Sohraby uses this approach in [121] where IBP sources with general distributions for the active and silent periods are considered. Unfortunately, the accuracy of this particular approximation is heavily dependent on the queue load being near unity.

We begin this chapter with the exact analysis of the population moments for a queue fed by N independent heterogeneous geom-geom IBPs in section 3.1. Although exact analyses has been performed in a similar fashion previously in [83,142], we present the solution in detail here using the basic probability generating function approach outlined in Chapter 2. Some of these results will be developed further in Chapter 5 when we consider queues with interrupted service.

In the remainder of the chapter we look at the performance of some approximate solutions for this queueing problem, with particular emphasis on the performance when the number of sources is large. Section 3.3 considers the use of the geometric tail approximation for obtaining the average and variance of the queue population, while the performance of the well known MMPP as an approximation for the superposition of the IBP sources is considered in section 3.4. Then in section 3.5 an approximation based on a result by Xiong and Bruneel is discussed. Unfortunately, none of these methods provide particularly good accuracy performance, except at very high queue utilisations. In section 3.6 we then present a completely new approximation method, which achieves very high accuracy at all queue utilisations, for a moderate computational cost.

3.1 Exact Queue Population Analysis

In this section we will look at the application of the theory developed in Chapter 2 to the geom-geom IBP source problem. As we will see, the average and variance are straightforward to calculate once the empty system vector is obtained.

3.1.1 Characterising a Geometric-Geometric IBP Source

We can write the state transition matrix of the IBP Markov chain illustrated in Figure 3.1 for the ith IBP source as

$$\mathbf{A}_i = \begin{bmatrix} \alpha_i & 1-\alpha_i \\ 1-\beta_i & \beta_i \end{bmatrix} \tag{3.1}$$

where α_i represents the probability that source i will remain in state 0 (denoted as the silent state) in the next time slot, given that it is in state 0 in the current time slot. The quantity β_i has the corresponding relation to state 1 (denoted as the active state). Since the matrix $\mathbf{A}_i$ is stochastic, it has one eigenvalue with value 1 and one eigenvalue with value $\alpha_i + \beta_i - 1$, which we will denote by the term γ_i.

Let the average arrival rate from the ith IBP source be denoted by λ_i, and let the corresponding average arrival rate when the source is in its active state be denoted by θ_i. The state based probability generating function matrix $\mathbf{P}_i(z)$ for source i will then be given by

$$\mathbf{P}_i(z) = \begin{bmatrix} 1 & 0 \\ 0 & p_i(z) \end{bmatrix} \tag{3.2}$$

where $p_i(z)$ is the probability generating function of the Bernoulli process for the active state of source i, given by

$$p_i(z) = 1 - \theta_i + z\theta_i \tag{3.3}$$

Since the silent state generates no arrivals, the invariant probability vector of $\mathbf{A}_i$ will be

$$\boldsymbol{\mu}_i = \left[1 - \frac{\lambda_i}{\theta_i}, \frac{\lambda_i}{\theta_i}\right] \tag{3.4}$$

and hence, with a little manipulation we obtain

$$\begin{aligned} \alpha_i &= \left(1 - \frac{\lambda_i}{\theta_i}\right) + \frac{\lambda_i}{\theta_i}\gamma_i && (3.5) \\ \beta_i &= \frac{\lambda_i}{\theta_i} + \left(1 - \frac{\lambda_i}{\theta_i}\right)\gamma_i && (3.6) \end{aligned}$$

The autocorrelation coefficient function for the arrival process from source i can be derived fairly simply (using a process similar to that on page 54 of [33] for example) to give

$$R(m) = \left(\frac{\theta_i - \lambda_i}{1 - \lambda_i}\right) \gamma_i^{|m|} \quad \text{for } m \neq 0 \tag{3.7}$$

which is geometric in γ_i. This provides a physical meaning for γ_i and we will refer to this term in future as the *autocorrelation parameter.* From inspection of $\mathbf{A}_i$ and $\mathbf{P}_i(z)$, and from equations (3.5) and (3.6), we can see that the geom-geom IBP source is completely described by the parameters λ_i, θ_i, and γ_i. These parameters will be used throughout this chapter to describe the geom-geom IBP sources.

The autocorrelation parameter is also related to the average length of the active periods of the IBP source (the average burst length) by the relation

$$\gamma_i = 1 - \frac{\theta_i}{\eta_{i,1}\left(\theta_i - \lambda_i\right)}$$

where $\eta_{i,1}$ denotes the first moment of the durations of the active periods of source i.

3.1.2 Applying the Queue Population Theory

In Appendix C we develop the relevant eigensystem analysis for geom-geom processes. From these results we obtain

$$\delta'(1) = \lambda \tag{3.8}$$

$$\delta''(1) = M_2 - \lambda + 2\sum_{i=1}^{N} \lambda_i \left(\theta_i - \lambda_i\right) \frac{\gamma_i}{1 - \gamma_i} \tag{3.9}$$

$$\delta'''(1) = M_3 - 3\left(M_2 - \lambda\right) - \lambda + 6\sum_{i=1}^{N} \lambda_i \left(\theta_i - \lambda_i\right) \frac{\left(\lambda - \lambda\gamma_i - 2\lambda_i + \theta_i\gamma_i\right)\gamma_i}{\left(1 - \gamma_i\right)^2} \tag{3.10}$$

where $\delta(z)$ represents the Perron–Frobenius eigenvalue for transition probability generating matrix of the combined arrival process. The quantities λ, M_2, and M_3 represent the first three moments of the stationary distribution of the number of arrivals occurring in one time-slot from the combined arrival process. For each individual source we also have

$$\mathbf{h}_{i,1}(1)\mathbf{g}'_{i,0}(1) = \frac{-\lambda_i\gamma_i}{1 - \gamma_i} \tag{3.11}$$

where $\mathbf{h}_{i,n}(z)$ and $\mathbf{g}_{i,n}(z)$ represent the left and right general eigenvectors corresponding to the nth $(n = 0, 1)$ eigenvalue of $\mathbf{A}_i\mathbf{P}_i(z)$ respectively.

Substituting these results into equations (2.36) and (2.51) we obtain

$$L_q = \frac{\mathbf{b}\mathbf{v}'(1)}{1 - \lambda} + \frac{M_2 - \lambda}{2\left(1 - \lambda\right)} + \frac{1}{1 - \lambda}\sum_{i=1}^{N} \lambda_i \left(\theta_i - \lambda_i\right) \frac{\gamma_i}{1 - \gamma_i} \tag{3.12}$$

and

$$\begin{aligned}\operatorname{Var}[L_q] &= \frac{\mathbf{b}\mathbf{v}''(1)}{1-\lambda} + 2\sum_{i=1}^{N} \frac{\lambda_i\gamma_i}{(1-\gamma_i)^2}\mathbf{b}\mathbf{g}_{(i,1)}(1) + \frac{M_3 - 3(M_2-\lambda) - \lambda}{3(1-\lambda)} \\ &+ \frac{2}{1-\lambda}\sum_{i=1}^{N}\lambda_i(\theta_i-\lambda_i)\frac{(\lambda-\lambda\gamma_i-2\lambda_i+\theta_i\gamma_i)\gamma_i}{(1-\gamma_i)^2} \\ &+ \frac{1}{1-\lambda}\left(M_2 - 2\lambda + 1 + 2\sum_{i=1}^{N}\lambda_i(\theta_i-\lambda_i)\frac{\gamma_i}{1-\gamma_i}\right)L_q - L_q^2 \qquad (3.13)\end{aligned}$$

where $\mathbf{b}$ is the empty system probability vector, and $\mathbf{v}(z)$ is the right hand Perron–Frobenius eigenvector of the combined arrival process, with derivatives given by

$$\mathbf{v}'(1) = \sum_{i=1}^{N}\left(\mathbf{e}_{2(i-1)} \otimes \mathbf{v}_i'(1) \otimes \mathbf{e}_{2(N-i)}\right) \qquad (3.14)$$

and

$$\begin{aligned}\mathbf{v}''(1) &= \sum_{i=1}^{N}\left(\mathbf{e}_{2(i-1)} \otimes \mathbf{v}_i''(1) \otimes \mathbf{e}_{2(N-i)}\right) \\ &+ 2\sum_{i=1}^{N-1}\left(\mathbf{e}_{2(i-1)} \otimes \mathbf{v}_i'(1) \otimes \sum_{j=1}^{N-i}\left(\mathbf{e}_{2(j-1)} \otimes \mathbf{v}_{i+j}'(1) \otimes \mathbf{e}_{2(N-i-j)}\right)\right) \qquad (3.15)\end{aligned}$$

where $\mathbf{e}_{2^n}$ is a unit column vector of 2^n elements, and

$$\mathbf{v}_i'(1) = \left[\frac{-\lambda_i\gamma_i}{1-\gamma_i}, \frac{(\theta_i-\lambda_i)\gamma_i}{1-\gamma_i}\right]^T \qquad (3.16)$$

$$\mathbf{v}_i''(1) = \left[\frac{2\lambda_i\gamma_i(\lambda_i-(1+\gamma_i)(\theta_i-\lambda_i))}{(1-\gamma_i)^2}, \frac{2\gamma_i(\theta_i-\lambda_i)(\theta_i\gamma_i-\lambda_i\gamma_i-2\lambda_i)}{(1-\gamma_i)^2}\right]^T \qquad (3.17)$$

Similarly, the vector $\mathbf{g}_{(i,1)}(1)$ is given by

$$\mathbf{g}_{(i,1)}(1) = \mathbf{e}_{2(i-1)} \otimes \mathbf{g}_{i,1}(1) \otimes \mathbf{e}_{2(N-i)} \qquad (3.18)$$

where

$$\mathbf{g}_{i,1}(1) = \left[1, 1 - \frac{\theta_i}{\lambda_i}\right]^T \qquad (3.19)$$

Thus the average and variance of the queue population are straightforward to calculate once $\mathbf{b}$ has been obtained.

3.1.3 Obtaining the Empty System Vector

As discussed in section 2.3.3, the $\mathbf{b}$ vector is obtained by firstly finding the poles of the queue equation that lie within the unit circle, and then solving a linear system of equations constructed from the general right hand eigenvector of the combined arrival process, evaluated at each of these pole positions.

Denote the jth pole of the queue equation by z_j^*, which will be that z satisfying

$$z - \omega_j(z) = 0 \tag{3.20}$$

where $\omega_j(z)$ is the jth general eigenvalue of the combined arrival process. From the Fixed Point Theorem in [47], equation (3.20) will have exactly one solution within the unit circle. Since we can easily show that $\omega_j(0) > 0$ and $\omega_j(1) \leq 1$, this solution must lie on the real axis. As in Chapter 2, we uniquely designate the eigenvalue that takes the value of 1 when $z = 1$ by $j = 0$ (that is $\omega_0(1) = 1$) so that we have $z_0^* = 1$, and $0 < z_j^* < 1$ for $j > 0$. These z_j^* can be easily found using the bisection algorithm or Newton–Raphson method [109].

This eigenvalue $\omega_j(z)$ is given in terms of the eigenvalues of the individual sources by

$$\omega_j(z) = \prod_{i=1}^{N} \omega_{i,r_{i,j}}(z) \tag{3.21}$$

where $r_{i,j}$ is a function that describes which eigenvalue (0 or 1) of source i is indicated when the overall eigenvalue is j, and $\omega_{i,n}$ is the nth ($n = 0, 1$) eigenvalue of the ith source, given by

$$\omega_{i,n}(z) = \frac{\alpha_i + \beta_i p_i(z)}{2} + (-1)^n \sqrt{\left(\frac{\alpha_i + \beta_i p_i(z)}{2}\right)^2 - \gamma_i p_i(z)} \tag{3.22}$$

The $r_{i,j}$ function can be easily realised by assuming that j expresses the selection of the individual eigenvalues using a binary ordering scheme. For example, the lowest order bit (enumerated from zero) of j can be assigned to the first source (enumerated from one) with the remaining bits assigned in increasing source order. The resulting expression for $r_{i,j}$ then becomes

$$r_{i,j} = 2^{1-i} j \bmod 2 \tag{3.23}$$

where $j = 0, 1, \ldots, \left(2^N - 1\right)$.

Once the poles are obtained, the linear system of simultaneous equations describing the empty system vector are given by

$$\mathbf{b}\left[\mathbf{e}, \mathbf{g}_1(z_1^*), \mathbf{g}_2(z_2^*), \ldots, \mathbf{g}_{2^N-1}(z_{2^N-1}^*)\right] = \left[(1-\lambda), 0, 0, \ldots, 0\right] \tag{3.24}$$

where $\mathbf{e}$ is a column vector of 2^N ones, and each of the $\mathbf{g}_j(z_j^*)$ are column vectors of the same size given by the Kronecker product

$$\mathbf{g}_j(z) = \bigotimes_{i=1}^{N} \mathbf{g}_{i,r_{i,j}}(z) \tag{3.25}$$

where $r_{i,j}$ is as defined previously, and

$$\mathbf{g}_{i,n}(z) = \left[1, \frac{1-\beta_i}{\omega_{i,n}(z) - \beta_i p_i(z)}\right]^T \tag{3.26}$$

Once the parameters of the linear system are determined, the system can be solved to find $\mathbf{b}$. Note that $\mathbf{b}$ will have all non-zero elements if all the sources have $\theta_i < 1$. Conversely, if all of the θ_i are equal to 1, then $\mathbf{b}$ will have only one non-zero element, and it is possible to obtain closed form expressions for the average and variance of the queue population. Although it is perhaps not likely that we would have $\theta_i = 1$ for all i (since it would involve all the sources transmitting in bursts at the link rate to the selected output queue), we consider it as a special case in Chapter 4.

3.1.4 Numeric Implementation Issues

There are two main areas where the implementation of this exact solution is susceptible to the finite accuracy of digital computation. The first is in the calculation of the pole locations, and the second is in finding the solution to the linear system. For the real pole situation considered here, very high accuracies in the pole placement are easily and quickly achievable provided that the calculations are performed in double precision (more than 12 significant digits on most computer platforms).

The situation is not so easily defined for the solution of the linear system, although double precision computations should be used throughout. Equation (3.24) can be rewritten as

$$\mathbf{bM} = \mathbf{x} \tag{3.27}$$

where $\mathbf{M}$ is the matrix formed from the $\mathbf{g}_j(z_j^*)$ vectors, and $\mathbf{x}$ is the vector on the right hand side of equation (3.24). The classical linear algebra solution for $\mathbf{b}$ can then be written as

$$\mathbf{b} = \mathbf{M}^{-1}\mathbf{x} \tag{3.28}$$

which requires $\mathbf{M}$ to be invertible. For matrices of any substantial size, direct implementation of this solution method for $\mathbf{b}$ is both inefficient in terms of computation, and highly susceptible to round-off errors. An alternative is to decompose $\mathbf{M}$ into a simpler form, and then use back-substitution to solve for $\mathbf{b}$.

One way to do this is with LU decomposition [109], where $\mathbf{M}$ is decomposed into a lower triangular matrix $\mathbf{L}$ and an upper triangular matrix $\mathbf{U}$, such that $\mathbf{M} = \mathbf{LU}$. The decomposition and subsequent back-substitutions require roughly m^3 operations for a matrix of size $m \times m$ to solve the linear system, which is nearly optimal [109].

Although the LU decomposition avoids determining the matrix inverse $\mathbf{M}^{-1}$ directly, the inverse must of course still exist for there to be a solution for $\mathbf{b}$. One standard way to check if a matrix is singular (or ill-conditioned) is to look at the determinant of the matrix. If the determinant is either very close to zero, or very large, then the matrix is poorly conditioned, and we expect the accuracy of the linear system solution to be poor. (Classically, if the determinant is zero, a solution does not exist, except when $\mathbf{x} = \mathbf{0}$). If this method of checking on the potential accuracy of $\mathbf{b}$ is used however, the results are deceptive. As an example, analysis of a simple queueing problem using 4 geom-geom IBP sources (the parameters are described in Table 3.8) gives a matrix $\mathbf{M}$ with a determinant of 1.1×10^{11}, while another example with 8 sources yields a determinant of 6.6×10^{267}. Both of these are extremely large for determinants, and would normally be a cause for alarm.

A more precise means for determining the accuracy of $\mathbf{b}$ however, is to calculate the residual of the solution. The residual r is defined to be

$$r = |\mathbf{bM} - \mathbf{x}| \tag{3.29}$$

where $|\cdot|$ represents the modulus of the argument (the length of the multidimensional vector). The residual is therefore a least squares *measure* of the difference between the actual and obtained solutions for the $\mathbf{b}$ vector [109]. In the examples above, the residual was returned as 1.3×10^{-16} and 3.6×10^{-15}, implying that both solutions are actually very accurate.

We do not know exactly why the residual can be so small with these extremely large determinants, although the reason is probably closely tied to the fact that the vector $\mathbf{x}$ has only one non-zero entry. That is the empty system vector $\mathbf{b}$ will actually be given by the (appropriately scaled) first row of the inverse matrix $\mathbf{M}^{-1}$ rather than relying on the entire inverted matrix.

3.1.5 Run Times for this Exact Solution Method

Timing results for this exact solution method were obtained for an IBM RS6000/320H and a Sun SPARC 10/402 workstation, using 1000 queueing problems for each N, where the parameters of the sources were obtained using a random generation scheme outlined in the following section. In each case, the time required to actually generate the 1000 random queueing problems and pass them to the calculation program was obtained separately, and subtracted from the observed run times of the entire generation and calculation process. In both cases, the *C* source code for performing the calculations (including the matrix manipulations and decompositions) is identical. These results are presented in Table 3.1.

We have already mentioned that the LU decomposition used to obtain the solution to the linear set of equations describing the $\mathbf{b}$ vector involves approximately m^3 operations,

N	IBM RS6000/320H	Sun SPARC 10/402
3	7.6 msec	6.6 msec
4	16 msec	14 msec
5	40 msec	35 msec
6	105 msec	120 msec
7	640 msec	650 msec
8	4.5 sec	4.9 sec
9	70 sec	54 sec

Table 3.1: *Mean run times for the exact solution method as a function of the number of sources for two common workstations.*

where m is the number of unknowns. For N geom-geom IBP sources, this means that the solution time should vary approximately as 8^N for large N, since the number of states in the combined arrival process is given by 2^N.

Inspection of Table 3.1 shows that the increase in the run times with N is less than a factor of 8 for the smaller N as expected, but on both machines the increase in run times between $N = 8$ and $N = 9$ is considerably more than 8 times. Since memory constraints were suspected on the RS6000, a couple of example 10 and 11 source problems were analysed on the Sun SPARC. The run times of these problems indicated that the increase from 9 to 10 sources and from 10 to 11 sources caused an increase in execution speed very close to a factor of 8. The reason why the increase from 8 sources to 9 is so much more than this is not known, but the result was repeated several times on the Sun SPARC, and simply appears to be an aberration in the solution process.

3.2 Approximate Queue Population Solutions

Considering the time and memory requirements of the above exact solution method (quite apart from the problem of numerical stability), practical use of these methods for connection admission or network analysis seems limited to small numbers of sources. One alternative is to consider the problem for a small number of classes of identical two-state sources [79, 83, 141]. This limits the state space or dimension of the problem to a total of $\prod_{i=1}^{K}(1 + N_i)$ states, where N_i denotes the number of identical sources belonging to connection class i, and K denotes the number of classes. The homogeneous case is of course then described by $K = 1$.

The problem with this approach is that, while it may have some validity at the network edges, it will rarely describe the behaviour of switches within the network. Since exact analytical methods cannot provide practical results for these problems when N is large, the use of approximate solutions must be considered.

One factor that must also be considered in deciding whether to use an approximate or exact analysis technique is that the practical accuracy of any solution is only as useful as the accuracy of the source model used. To illustrate this point, suppose that extensive computation is required to obtain an exact solution for a particular network traffic model. When the results are compared to observations of the 'real world' they might be found to only be accurate to within ±20%. In this situation, approximation methods with errors relative to the exact solution of up to say 10% could replace the exact models without significantly compromising the final applicability of the results.

It is obviously important then that the accuracy of the approximation methods be known. In the following sections we will discuss several approximation techniques based on common approaches, and will study their accuracy relative to the exact solution results. In section 3.6 we will then propose a new approximation method which provides considerably better accuracy for only a slightly higher computational cost.

Details of the Accuracy Study

Rather than constructing arbitrary queueing problems on which to assess the accuracy of the approximation methods discussed in this chapter, we have chosen to use a random generation approach. Each queueing problem then consists of N sources with parameters chosen in the following manner. For each source i, an average arrival rate λ_i is assigned using a uniform distribution in the range of 0 to 1. These λ_i are then scaled[1] so that the sum of these λ_i (which gives the overall arrival rate from the queue λ) has a value chosen uniformly from the range λ_L to λ_U. Once the λ_i are obtained, the average generation rate θ_i of the active state of source i can then be chosen (again randomly) from the range of λ_i to 1. The autocorrelation parameter γ_i of the ith source is chosen independently of the other two parameters, using a uniform distribution in the range γ_L to γ_U.

Unless otherwise noted, all the randomly generated queueing problems in this chapter were obtained using $\lambda_L = 0.1$, $\lambda_H = 0.9$, $\gamma_L = 0$, and $\gamma_H = 0.99$. Since λ and γ must both be less than one for the queuing problem to be stable, these ranges encompass the limits under which these approximation techniques might be expected to usefully operate in practice.

The actual accuracy of the approximation is measured as the relative difference between the approximate results, and the exact results obtained using the approach discussed

[1] This approach provides for the possibility that a single source can be the source of the majority of the arrivals in the arrival process, which is a quite likely occurrence in the real world.

in section 3.1. The relative difference, or error is defined here as

$$\text{Relative error} = \frac{\text{approximation result}}{\text{exact result}} - 1 \tag{3.30}$$

where the order of the terms is chosen so that a positive error corresponds to the approximation result being larger than the exact result. Typically 1000 randomly generated problems were used for each error analysis, and in some cases 10,000 and 100,000 problems are considered for use in scatter plots and histograms. Comparisons between approximation methods are based on the statistics of the observed errors. In particular the mean, standard deviation, and the 1st and 99th percentiles are considered.

3.3 The Geometric Tail Approximation

One well known property of a wide range of queueing systems (see Chapter 6) is that tail distribution of the queue population can be approximated by a function of the form

$$t_n \approx \psi\phi^n \tag{3.31}$$

where we denote by t_n the steady state probability that the queue population is strictly greater than n. This approximation improves as n increases due to the asymptotic nature of this property, and is often referred to as the *geometric tail approximation.* The terms ψ and ϕ are called the *geometric scaling factor* and *geometric coefficient* respectively. Although this approximation is more suitable for estimating loss probabilities (as we shall discuss in Chapter 6), here we will consider its use as a means for estimating the average and variance of the queue population.

3.3.1 Obtaining the geometric coefficient ϕ

A convenient property of queues having this geometric tail property is that

$$\phi^{-1} - \delta(\phi^{-1}) = 0 \tag{3.32}$$

where $\delta(z)$ is the Perron-Frobenius eigenvalue of the arrival process (see Chapter 6 for more details). In other words, ϕ is the reciprocal of the single positive real solution of the equation $z - \delta(z) = 0$ lying outside the unit circle. This then provides us with a means whereby the exact value of ϕ can be calculated quite easily, using either the bisection or Newton–Raphson methods.

An approximate solution for ϕ

The main limitation in obtaining this exact solution for ϕ lies in the requirement that the Perron–Frobenius eigenvector must be known in algebraic form. This restricts the size of the Markov chain of each arrival process to 4 states at most[2], unless the process is a special case (such as the cyclic process dealt with in Chapter 5). In addition, for very large numbers of sources, the time required to find the exact solution may become excessive.

Sohraby [120, 121] presents an approximation for ϕ using the first order Taylor series expansion of equation (3.32) around $\phi = 1$, which can be written as

$$\phi^{-1} \approx 1 + \frac{2\,(1-\lambda)}{\delta''(1)} \tag{3.33}$$

or, more conveniently

$$\frac{\phi}{1-\phi} \approx \frac{\delta''(1)}{2\,(1-\lambda)} \tag{3.34}$$

For the geom-geom IBP sources considered in this chapter, this becomes

$$\frac{\phi}{1-\phi} \approx \frac{M_2-\lambda}{2\,(1-\lambda)} + \frac{1}{1-\lambda}\sum_{i=1}^{N} \lambda_i\,(\theta_i - \lambda_i)\,\frac{\gamma_i}{1-\gamma_i} \tag{3.35}$$

where M_2 is the second moment of the overall stationary arrival process.

This approximation is derived assuming that the queue is heavily loaded, and hence that ϕ will be close to 1. To improve the accuracy of the approximation generally, a more exact approximation can be obtained by increasing the order of the Taylor series expansion. Following the same approach as before, we then obtain

$$\frac{\phi}{1-\phi} \approx \frac{-2\delta'''(1)}{3\delta''(1) - \sqrt{9\delta''(1)^2 + 24\,(1-\lambda)\,\delta'''(1)}} \tag{3.36}$$

where, for the geom-geom IBP sources, $\delta''(1)$ and $\delta'''(1)$ are given by equations (3.9) and (3.10) respectively.

Note that there may be situations where the contents of the square root in equation (3.36) become negative, resulting in a complex valued solution for ϕ. The reason for this is that equation (3.36) represents only a partial power series approximation to the actual ϕ solution, and in the full Taylor series expansion of ϕ, all the complex components would cancel. In practical terms, we found it best to ignore the approximation of equation (3.36) when this problem occurred, and return to the solution given by equation (3.35).

3.3.2 Estimating the Scaling Factor

Although we have these two methods for obtaining the decay coefficient of the geometric tail, there is no simple method to obtain the scaling factor ψ. In [120], Sohraby uses

[2] There is no known closed form algebraic solution for the eigenvalues of a general matrix of size greater than 4 × 4 [139].

the simple assumption that the tail of the queue distribution is geometric for its entire length (from 1 to ∞) which leads to $\psi = \lambda\phi$ in our notation. This solution forms a fairly loose upper bound to the actual tail distribution. Xiong and Bruneel [141] use a more complete analytic approach to provide another upper bound on ψ, which their numeric investigations show to be quite tight. The derivation of the result is straightforward, although too involved to present here, and the interested reader is directed to [141] for the details. We will encounter the main assumption behind Xiong and Bruneel's result again in section 3.5.

Note that an upper bound on the scaling factor ψ does not mean that the average and variance calculated from this bound are guaranteed to be greater than the actual values. Whether this occurs or not is highly dependent on the behaviour of the actual tail distribution for smaller queue population values (where the geometric property does not hold so well).

In our notation, Xiong and Bruneel's result becomes, with some manipulation

$$\psi = \frac{\phi\,(1-\lambda)\,C(\phi^{-1})}{(\delta'(\phi^{-1})-1)\,D(\phi^{-1})} \tag{3.37}$$

where

$$C(z) = \prod_{i=1}^{N} \frac{(\theta_i - \lambda_i)\,(1-\gamma_i)\,z + (1-\theta_i)\,(\delta_i(z)-\gamma_i)}{(1-\lambda_i)\,(1-\gamma_i)\,(1-\theta_i+z\theta_i)} \tag{3.38}$$

$$D(z) = \prod_{i=1}^{N} \frac{2\delta_i(z) - (\lambda_i + \theta_i\gamma_i - \lambda_i\gamma_i)\,(z-1) - 1 - \gamma_i}{\delta_i(z) - (1-\theta_i+z\theta_i)\,\gamma_i} \tag{3.39}$$

and where $\delta_i(z)$ is the Perron–Frobenius eigenvalue of the ith source, and $\delta(z)$ is the overall Perron–Frobenius eigenvalue for the superposition of the N sources. Note that the first derivative of $\delta(z)$ can be written in terms of the individual source eigenvalues as

$$\delta'(z) = \delta(z)\sum_{i=1}^{N}\frac{\delta_i'(z)}{\delta_i(z)} \tag{3.40}$$

As a third alternative, we note that our own observations suggest that

$$\frac{\mathrm{Var}\,[L_q]^{\frac{1}{2}}}{L_q} \approx \frac{\mathrm{Var}\left[L_q^*\right]^{\frac{1}{2}}}{L_q^*} \tag{3.41}$$

where L_q^* and Var $\left[L_q^*\right]$ denote the marginal solutions for the average and variance of the queue population (see Appendix A). In other words, the ratio of the standard deviation of the queue population to the average is approximately equal to the equivalent ratio of the marginal solution. In order to support this claim, we define the variable r to be that which satisfies

$$\frac{\mathrm{Var}\,[L_q]^{\frac{1}{2}}}{L_q} = r \times \frac{\mathrm{Var}\left[L_q^*\right]^{\frac{1}{2}}}{L_q^*} \tag{3.42}$$

where we are suggesting that $r \approx 1$. Figures 3.2 and 3.3 show a scatter plot and histogram of the r values obtained from the analysis of 10,000 and 100,000 randomly generated queueing problems respectively using 4 sources. The average value for r observed over the 100,000 samples was 0.9789 ± 0.0006 (with 99% confidence).

It would appear from these Figures that the assumption that $r = 1$ is acceptable, if not particularly accurate. The obvious trend from the scatter plot is that the approximation performs better for higher utilisations, although the spread of values around the mean is fairly independent of the utilisation. Further observations (although not in as much detail) suggest that the approximation may improve slowly as the number of sources increases. Thus, taking approximation (3.41) as being exact, and assuming the tail of the population distribution is geometric, we obtain

$$\frac{1}{\psi}(1+\phi) - 1 = \frac{\mathrm{Var}\left[L_q^*\right]}{\left(L_q^*\right)^2} \tag{3.43}$$

by making use of the equations (6.7) and (6.8). Alternatively, from the marginal variance and average results in Appendix A we obtain

$$\psi = \frac{3(1+b)(M_2-\lambda)^2}{4(1-\lambda)(M_3-\lambda)+6(M_2-1)(M_2-\lambda)} \tag{3.44}$$

where M_r denotes the rth moment of the net arrival process. We will refer to this solution as the *ratio method.*

3.3.3 Comparison of the Approximations

In order to choose the best geometric tail approximation, we will first consider the effect of the choice of the scaling factor ψ, using the exact numerical solution for the geometric term ϕ discussed in section 3.3.1. Figure 3.4 shows the absolute value of the mean relative error in the approximation for both the average and the variance as a function of the number of sources for the three methods discussed above. The methods for calculating ψ are Sohraby's [120], Xiong and Bruneel's method [141], and the ratio method, described by equation (3.44) above. In each case, the average and variance of the queue population are given by

$$\tilde{L}_q = \frac{\psi}{1-\phi} \quad \text{and} \quad \mathrm{Var}\left[\tilde{L}_q\right] = \frac{\psi(1+\phi-\psi)}{(1-\phi)^2} \tag{3.45}$$

respectively (see Chapter 6).

The results for Sohraby's method [120] (where $\psi = \lambda\phi$) are obviously the worst of the approximations. All three however suffer from increasing mean errors as the number of sources increases, which is particularly undesirable since we ideally wish to use the

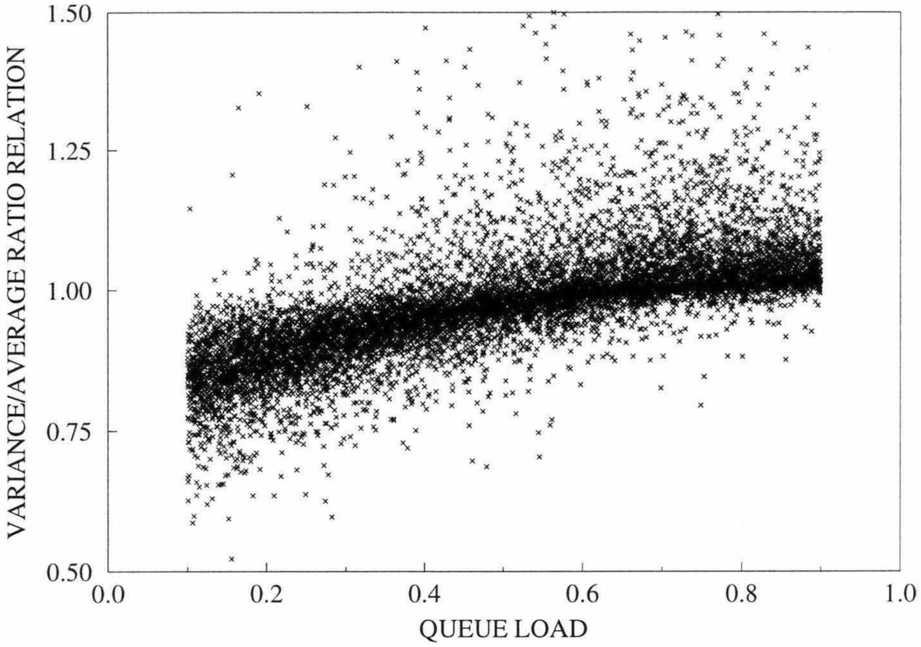

Figure 3.2: *Scatter plot of the ratio variable* r *as a function of the queue load. The results are from observations of* 10,000 *randomly generated geom-geom IBP queueing problems with utilisations between* 0.1 *and* 0.9.

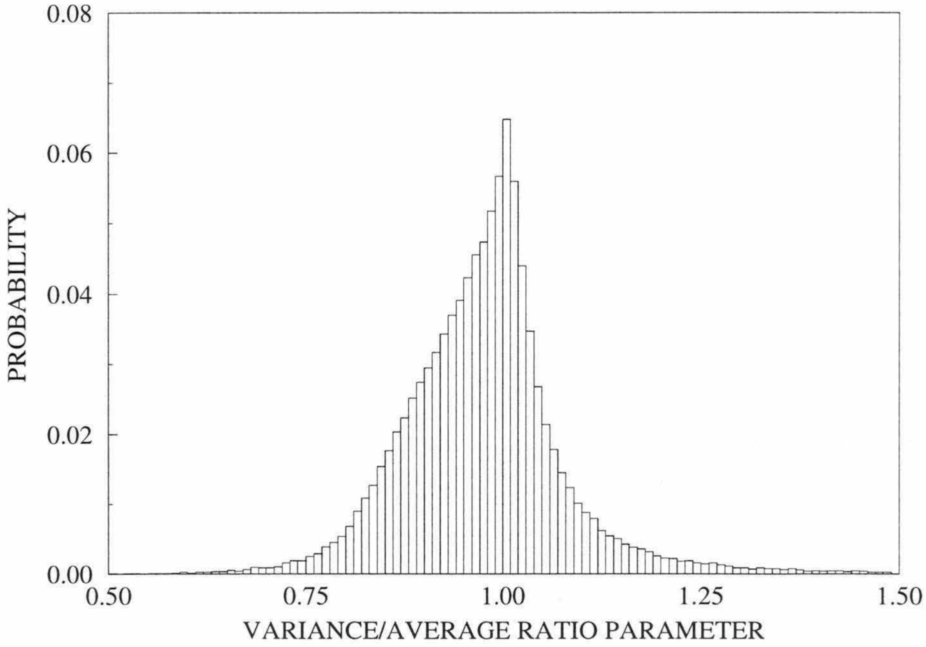

Figure 3.3: *Histogram of the distribution of the ratio variable* r *observed over* 100,000 *randomly generated geom-geom IBP queueing problems with utilisations between* 0.1 *and* 0.9. *The reason for the unequal slopes is readily apparent from Figure 3.2.*

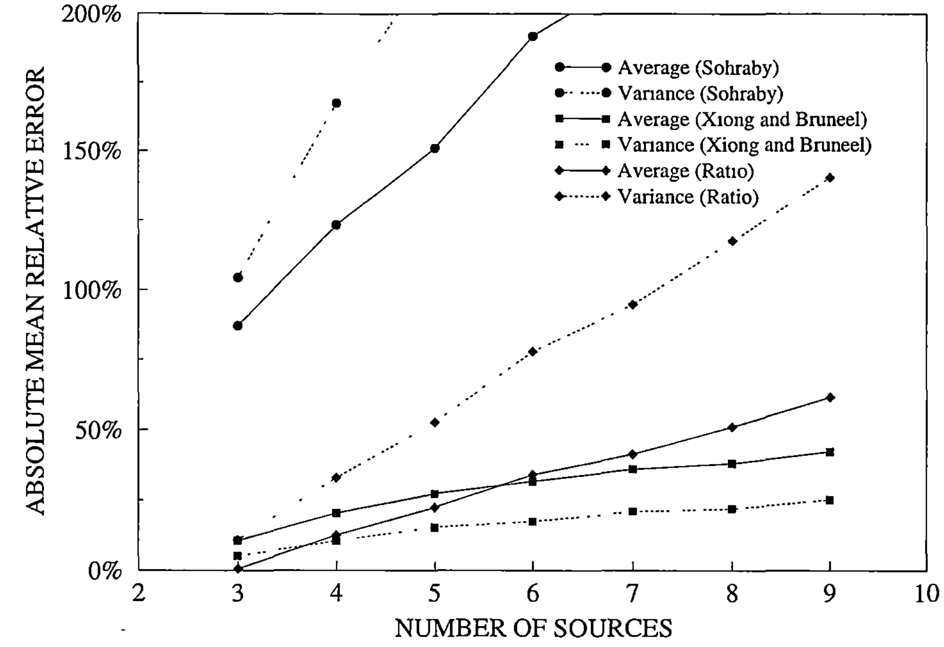

Figure 3.4: *Mean relative error in the average and variance of the queue population as a function of the number of sources for* 1000 *randomly generated queueing problems.*

approximation for large numbers of sources. We suggested earlier that the accuracy of the ratio method might possibly increase with the number of sources, but this observation has not been born out by the results. Of the three methods, the best is the one using Xiong and Bruneel's approximation for ψ and this is the one we will concentrate on for the rest of this discussion.

The accuracy of Xiong and Bruneel's approximation method depends greatly on the accuracy of the ϕ value. For low queue utilisations, the two approximate solutions for ϕ given by equations (3.35) and (3.36) can become quite different to the exact value of the geometric decay coefficient. As a consequence, use of Xiong and Bruneel's equation (3.37) results in very poor approximations for the scaling factor, with negative values of ψ being one occasional consequence. We will not therefore consider the performance of the overall approximation with these two solutions for ϕ, but note that they should still be suitable for analysis of problems when the queue utilisation is quite high.

Tables 3.2 and 3.3 present the statistics[3] on the relative error observed for Xiong and Bruneel's approximation, described by equation (3.37) using the exact solution method for ϕ. We see from the first percentile that the lower limit on the average and variance

[3] Although we present the statistics of the relative errors to three and two significant figures, we do not really have this level of precision. Particularly the 1st and 99th percentiles are very uncertain, since they are based on the least and greatest 10 results out of the 1000 obtained for each N. Ideally we would like 10, 000 sources or more, but this is not very practical for larger numbers of sources.

of the queue population stays close to −100% (approximation of zero) while the upper limit (given by the 99th percentile result) decreases for the average, while staying relatively constant for the variance.

N	Mean	Deviation	1st Percentile	99th Percentile
3	−10.7%	26%	−86%	49%
4	−20.3%	29%	−95%	35%
5	−27.2%	33%	−95%	46%
6	−31.6%	34%	−99%	32%
7	−36.1%	35%	−99%	23%
8	−38.1%	36%	−100%	36%
9	−42.3%	37%	−100%	25%

Table 3.2: *Statistics on the relative error between the approximate and exact solutions for the average queue population obtained from the geometric tail approximation using Xiong and Bruneel's solution for the scaling factor ψ. Each row in the table represents observations from* 1000 *randomly generated problems.*

N	Mean	Deviation	1st Percentile	99th Percentile
3	−5.22%	23%	−82%	41%
4	−10.5%	27%	−92%	49%
5	−15.3%	32%	−93%	49%
6	−17.4%	34%	−99%	44%
7	−21.0%	36%	−98%	35%
8	−21.9%	37%	−99%	45%
9	−25.3%	39%	−99%	49%

Table 3.3: *Statistics on the relative error for the queue population variance corresponding to the average queue population results of Table 3.2.*

To investigate the reason for these observations, Figure 3.5 shows a scatter plot of the relative error observed in the average queue population as a function of the queue load for 10,000 randomly generated queueing problems using 4 sources and Xiong and Bruneel's solution for ψ using exact results for ϕ. It is very clear that the behaviour of the approximation is heavily dependent on the queue load or utilisation, but even at high utilisations there is still a large spread in the possible error in the approximation for the average population.

Xiong and Bruneel's solution for ψ coupled with the exact numeric solution for ϕ represents the best method found for the implementing the geometric tail approximation for moments of the queue population. As can be seen from the data, the results are

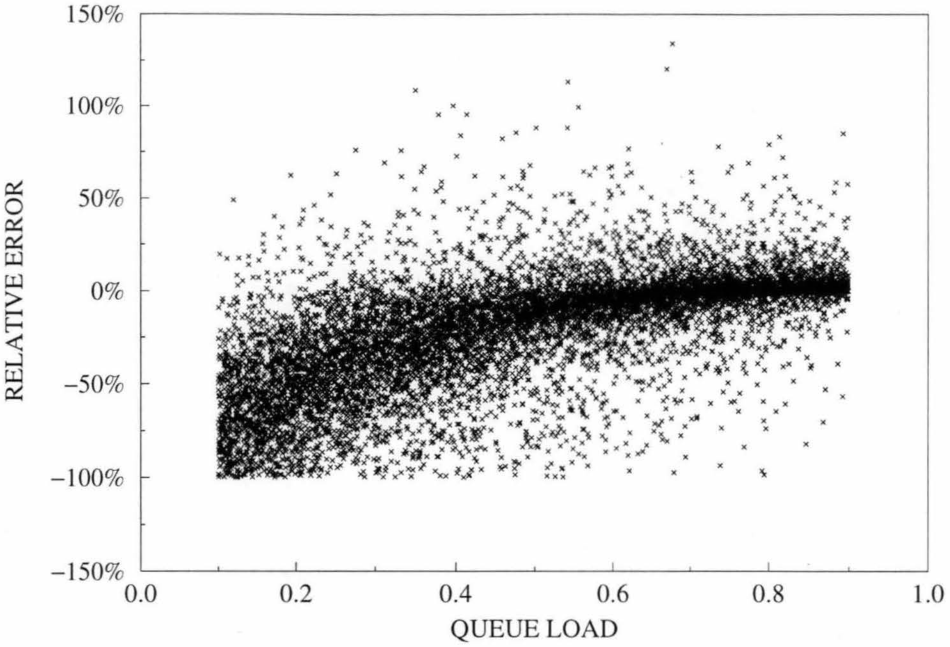

Figure 3.5: *Scatter plot of the relative error in the average queue population approximation as a function of the queue load for* 4 *sources. The results are from observations of* 10,000 *randomly generated queueing problems using Xiong and Bruneel's method for assigning the geometric tail scaling factor.*

far from satisfactory. All of the methods discussed in this section performed poorly for light queue utilisations, suggesting that this is a perhaps a property of the geometric tail approximation itself rather than the methods used to choose the parameters. As we will discuss later in Chapter 6, this is indeed the case, and we expect the geometric approximation to underestimate the average and variance when the utilisation is low — or more specifically when ϕ is small.

3.4 The MMPP Approximation

The discrete time Markov modulated Poisson process (MMPP) or switched Poisson process (SPP) is a Markov modulated process that generates arrivals while in state j of the governing Markov chain according to a Poisson process with average rate parameter λ_j. Its use as an approximation for the net arrival process from on-off sources is popular in the literature, particularly in regard to buffer loss probabilities, and there is a wealth of articles available discussing its use (see [11, 44, 45, 77, 84, 92, 103, 135, 143, 146] and references therein).

The solution for the moments of the population of an infinite buffer queue fed by an

MMPP is straightforward, since the small number of states in the model allows the probability generating function approach to be used with little computational cost. We will only consider the two state MMPP in the following discussion, although models with higher numbers of states have been considered in [143].

3.4.1 Parameter Matching

A two-state MMPP can be characterised by four parameters — the average rate from each of the two states (denoted by p_0 and p_1), the average arrival rate from the overall process (denoted by p), and the autocorrelation between the two states (denoted by γ). The transition matrix for the MMPP is then given by

$$\mathbf{A} = \begin{bmatrix} \mu_0 + \mu_1\gamma & \mu_1(1-\gamma) \\ \mu_0(1-\gamma) & \mu_1 + \mu_0\gamma \end{bmatrix} \tag{3.46}$$

where

$$\mu_0 = \frac{p_1 - p}{p_1 - p_0} \quad \text{and} \quad \mu_1 = \frac{p - p_0}{p_1 - p_0} \tag{3.47}$$

with a corresponding state based probability generating function matrix given by

$$\mathbf{P}(z) = \begin{bmatrix} e^{p_0(z-1)} & 0 \\ 0 & e^{p_1(z-1)} \end{bmatrix} \tag{3.48}$$

so that $\mathbf{AP}(z)$ describes the transition probability generating matrix for the MMPP.

The main task in modelling the superposition of a number of IBP sources by a two state MMPP is in matching these four parameters (p_0, p_1, p, γ) to the statistics of the IBPs and to the net arrival process. Most of the available literature is concerned with one [11, 45, 84, 92] or at most two classes [135, 136] of identical sources, which means it cannot easily be applied to the heterogeneous sources case. Hartanto [44] investigated the performance of three of the more recent matching methods, finding the method of Wang and Silvester [135] to be the most accurate with both one and two classes of identical sources. Unfortunately, these methods do not easily extend to the heterogeneous case [51].

Instead, we consider the method described by Lee and Lee [77] (and also discussed in [44]). This approach matches the average and variance of a superposition of interrupted Poisson processes (IPPs), and additionally the peak to mean ratio and the time constant (or sum) of their autocovariance functions. For the arrival rates of the MMPP, this approach gives

$$p = \lambda \tag{3.49}$$

$$p_0 = \lambda - \sqrt{\frac{\sigma^2 - \lambda}{r}} \tag{3.50}$$

$$p_1 = \lambda + \sqrt{r(\sigma^2 - \lambda)} \tag{3.51}$$

where λ and σ^2 are the required average and variance of the arrival rate, and

$$r = \frac{1}{\lambda}\sum_{i=1}^{N} \lambda_i r_i \tag{3.52}$$

where r_i is the ratio of the peak arrival rate to the average arrival rate of source i.

We cannot simply extend this approach to the IBP case however, because the superposition of any number of IBPs always has $\sigma^2 < \lambda$ while an MMPP always has $\sigma^2 > \lambda$. That is, there is a fundamental difference in the distributions, and we cannot match the variance of the MMPP to that of the IBP source superposition. As an alternative we will use the suggested [44]

$$\begin{aligned} p_0 &= \lambda - \sqrt{\frac{\sigma^2}{r}} && (3.53) \\ p_1 &= \lambda + \sqrt{r\sigma^2} && (3.54) \end{aligned}$$

where σ^2 is the variance of the superposition of IBP's (giving the MMPP a variance of $\sigma^2 + \lambda$). The peak to mean factor r is given simply by

$$r = \frac{1}{\lambda}\sum_{i=1}^{N} \theta_i \tag{3.55}$$

In order to match the autocorrelation parameter of the MMPP, we note that its autocorrelation coefficient function is given by

$$R(m) = \frac{1}{\sigma^2}(p - p_0)(p_1 - p)\gamma^{|m|} \tag{3.56}$$

and that of the ith IBP source is given by

$$R_i(m) = \frac{\theta_i - \lambda_i}{1 - \lambda_i}\gamma_i^{|m|} \tag{3.57}$$

so that, matching the autocorrelation sum of the MMPP to that of the superposed IBPs gives

$$\frac{\gamma}{1-\gamma} = \frac{1}{(p - p_0)(p_1 - p)}\sum_{i=1}^{N} \lambda_i(\theta_i - \lambda_i)\frac{\gamma_i}{1-\gamma_i} \tag{3.58}$$

3.4.2 Queue Population Analysis

The average and variance of the queue population of an infinite buffer queue fed by a two state MMPP is straightforward, requiring a single numeric solution to a pole equation. To simplify the mathematics, we note that the MMPP arrival process can be described by the superposition of a Poisson process with rate parameter p_0 and an interrupted Poisson process with a rate parameter in the active state of $(p_1 - p_0)$.

Denote by z^* the value of z that satisfies

$$z - e^{p_0(z-1)}\omega_1(z) = 0 \tag{3.59}$$

where $\omega_1(z)$ is the eigenvalue of the IPP transition probability generating matrix that has the property $\omega_1(1) \neq 1$. From Appendix C, this eigenvalue is given by

$$\omega_1(z) = \frac{\alpha + \beta a(z)}{2} - \sqrt{\left(\frac{\alpha + \beta a(z)}{2}\right)^2 - \gamma a(z)} \tag{3.60}$$

where

$$a(z) = e^{(z-1)(p_1 - p_0)} \tag{3.61}$$

and

$$\begin{aligned} \alpha &= 1 - (1-\gamma)\frac{p - p_0}{p_1 - p_0} && (3.62) \\ \beta &= \gamma + (1-\gamma)\frac{p - p_0}{p_1 - p_0} && (3.63) \end{aligned}$$

Then, denoting the empty system vector by $\mathbf{b} = [b_0, b_1]$ we obtain

$$b_0 = \frac{(1-p)(1-\beta)}{1 - \beta - \omega_1(z^*) + \beta a(z^*)} \tag{3.64}$$

and

$$b_1 = 1 - p - b_0 \tag{3.65}$$

Substituting in the relevant equations of section 2.7 using the results of Appendix C, we finally obtain (with a little work)

$$L_q = \frac{\lambda(p_0 + p_1) - p_0 p_1}{2(1-\lambda)} + \frac{(p_1 - \lambda + \lambda p_0 - b_0(p_1 - p_0) - p_0 p_1)\gamma}{(1-\lambda)(1-\gamma)} \tag{3.66}$$

and

$$\begin{aligned} \operatorname{Var}[L_q] = {} & \frac{6\lambda(1-\lambda)(p_0+p_1) + \lambda(4-\lambda)(p_0+p_1)^2}{12(1-\lambda)^2} \\ & + \frac{(2\lambda + 4\lambda^2 - 6 - 4p_0 - 2\lambda p_0 - 4p_1 - 2\lambda p_1 + 3p_0 p_1)\, p_0 p_1}{12(1-\lambda)^2} \\ & + \frac{\left(p_0 p_1 \lambda - \lambda^2 + p_0^2(2\lambda - \lambda^2 - 2p_1) + p_1^2(1-p_0)^2 - b_0 p_1^2(1-\lambda)\right)\gamma}{(1-\lambda)^2(1-\gamma)^2} \\ & - \frac{\gamma(1+\gamma)\left(\lambda - p_1 + p_0(\lambda^2 - 2\lambda + 2p_1) + b_0(1-\lambda)(p_0^2 - p_0 + p_1)\right)}{(1-\lambda)^2(1-\gamma)^2} \\ & + \frac{\left(b_0^2(p_0 - p_1)^2 + b_0 p_1(1-\lambda)(2p_0 - p_1) + p_0 p_1 \lambda\right)\gamma^2}{(1-\lambda)^2(1-\gamma)^2} \end{aligned} \tag{3.67}$$

3.4.3 Accuracy Study

Equations (3.66) and (3.67) for the average and variance of the queue population are implemented in this study using equations (3.49), (3.53), (3.54), (3.55), and (3.58) to define the MMPP parameters from the parameters of the IBP sources. Tables 3.4 and 3.5 show the relative error results for the approximation for N IBP sources. As for the previous approximation study, 1000 randomly generated problems with $0.1 \leq \lambda < 0.9$ and $0 \leq \gamma_i < 0.99$ were considered for each N.

N	Mean	Deviation	1st Percentile	99th Percentile
3	401%	430%	44%	2000%
4	346%	420%	32%	1600%
5	317%	350%	38%	1700%
6	307%	340%	29%	1600%
7	301%	330%	33%	1400%
8	287%	290%	32%	1300%
9	289%	310%	30%	1500%

Table 3.4: *Statistics on the relative error between the approximate and exact solutions for the average queue population obtained from the MMPP approximation. Each row in the table represents observations from* 1000 *randomly generated problems.*

N	Mean	Deviation	1st Percentile	99th Percentile
3	1090%	2600%	26%	7200%
4	1010%	3700%	4.8%	6600%
5	892%	2000%	−8.5%	6300%
6	910%	1900%	2.1%	7100%
7	923%	1900%	−7.5%	9400%
8	831%	1200%	−3.6%	4900%
9	905%	1700%	−5.9%	8400%

Table 3.5: *Relative error in the queue population variance corresponding to the average queue population results of Table 3.4.*

The performance of this MMPP approximation is extremely poor, although a very small decrease in the observed errors does occur with increasing numbers of sources. Figure 3.6 shows a scatter plot of the relative error in the average queue population approximation observed from 10,000 randomly generated queueing problems with 4 sources. Obviously the performance improves for very high utilisations, but even here, the mean relative error is of the order of 100%. Part of the reason for this is that the variance

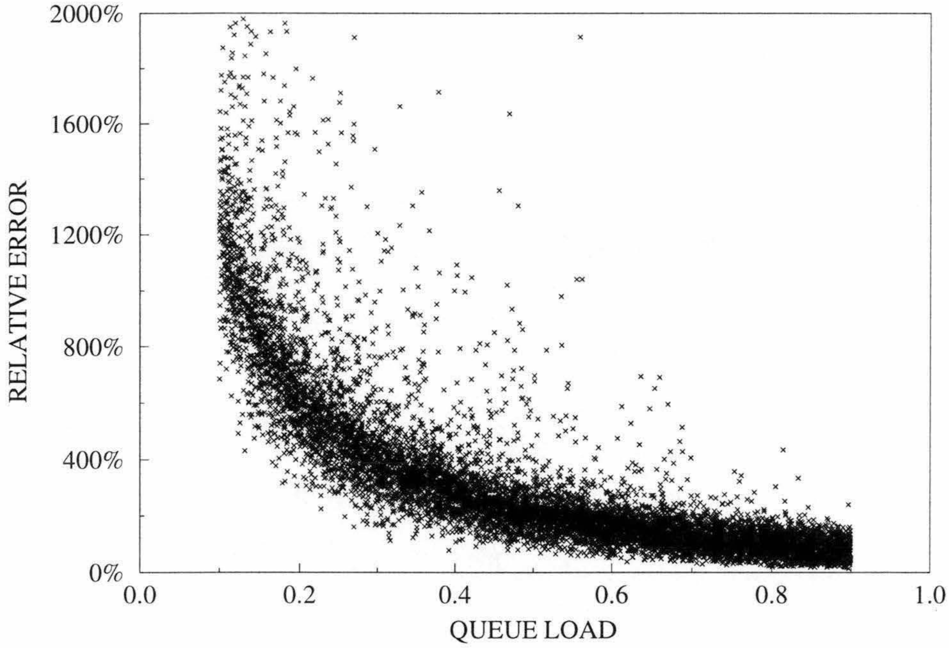

Figure 3.6: *Scatter plot of the relative error in the average queue population as a function of the queue load obtained from the MMPP approximation for* 4 *sources.*

of the MMPP arrival rate is λ greater than the variance of the IBP sources. This will contribute a relative error roughly proportional to $1/\lambda$, which is a good description of the underlying curve visible in Figure 3.6.

This suggests a possible improvement to the approximation. We can replace the marginal components of the MMPP queue population average and variance by the equivalent components of the actual IBP arrival process. Investigation of this improvement shows it to reduce the mean errors to roughly 100% and 600% for the average and variance respectively, with little variation as N increases. These errors are still far too large however to make the MMPP a useful approximation method.

3.5 Empty System Vector Approximation

The primary difficulty in obtaining the exact solution to the IBP queueing problem lies in obtaining the empty system probability vector $\mathbf{b}$. An approximate solution for this vector might however provide another means for estimating the average and variance of the queue population. In [141], Xiong and Bruneel develop a tight upper bound for the scaling factor in the geometric tail approximation by noting that the probability that the combined arrival process is in state i when the system is empty will always be less than or equal to the probability that the state is i when no arrivals are generated.

That is

$$\Pr(s = i \mid y = 0) \leq \Pr(s = i \mid a = 0) \tag{3.68}$$

where s denotes the state of the arrival process, y the system population, and a the number of arrivals in the corresponding time interval. Using the fact that the empty system probability vector of our notation is given by the set of probabilities $\Pr(s = i, y = 0)$ we then obtain an approximation for $\mathbf{b}$ as

$$\widetilde{b}_i = \frac{1 - \lambda}{\prod_{j=1}^{N} (1 - \lambda_j)} \times \Pr(a = 0, s = i) \tag{3.69}$$

where $\widetilde{b}_i$ denotes the approximation for the element of vector $\mathbf{b}$ corresponding to the combined process state i. The joint probability $\Pr(a = 0, s = i)$ is the probability that no arrivals are generated *and* the combined process is in state i, and is given by

$$\Pr(a = 0, s = i) = \prod_{j=1}^{N} \sigma_{j,i} \left(\frac{\lambda_j}{\theta_j} - \lambda_j \right) + (1 - \sigma_{j,i}) \left(1 - \frac{\lambda_j}{\theta_j} \right) \tag{3.70}$$

where $\sigma_{j,1}$ is an indicator that takes the value 1 if source j is active when the combined process state is i, and takes a value of 0 otherwise.

3.5.1 Accuracy Study

Figure 3.7 shows an example approximation to the empty state probability vector for a 4 source problem (the actual source parameters are given in Table 3.8). Although the closeness of the fit is not particularly good, the approximation does follow the same pattern as the exact solution. In this example, using the approximate $\mathbf{b}$ vector in the corresponding equations for the average and the variance of the queue population yields relative errors of 10.9% and 0.2% respectively.

This good result for the variance is unusual, and in most cases studied, the variance approximation was of very poor accuracy, with the approximate result frequently being less than zero. For example, the mean relative error observed for 1000 randomly generated queueing problems with 4 sources was a huge −270%. As an alternative, we have used the result for the average, combined with the exact analysis for the decay coefficient of the geometric tail property (see previous section) to provide an estimate for the geometric tail scaling factor. Thus, with the two geometric tail parameters defined, an approximation for the variance can be obtained.

Figure 3.8 shows a plot of the mean relative error observed for the average and the geometric tail supported variance approximations, as a function of the number of sources. As before, each result is obtained from observation of 1000 randomly generated queueing problems with a queue load between 0.1 and 0.9, and with the individual sources autocorrelation parameters restricted to less than 0.99.

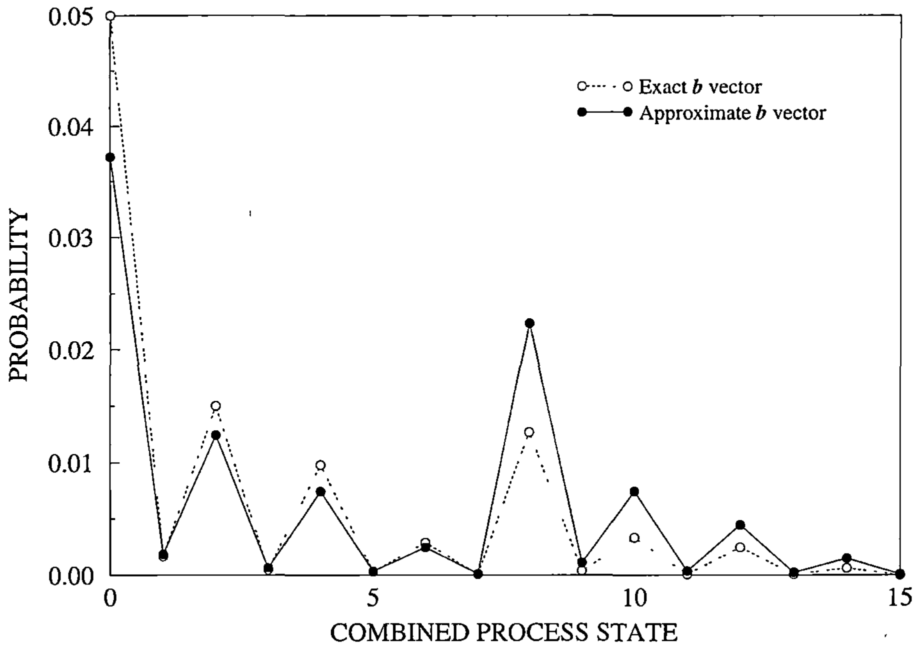

Figure 3.7: *Approximate and exact joint probabilities for the system being empty and the combined arrival process being in the indicated state. This joint probability set is described in the text by the vector* **b**.

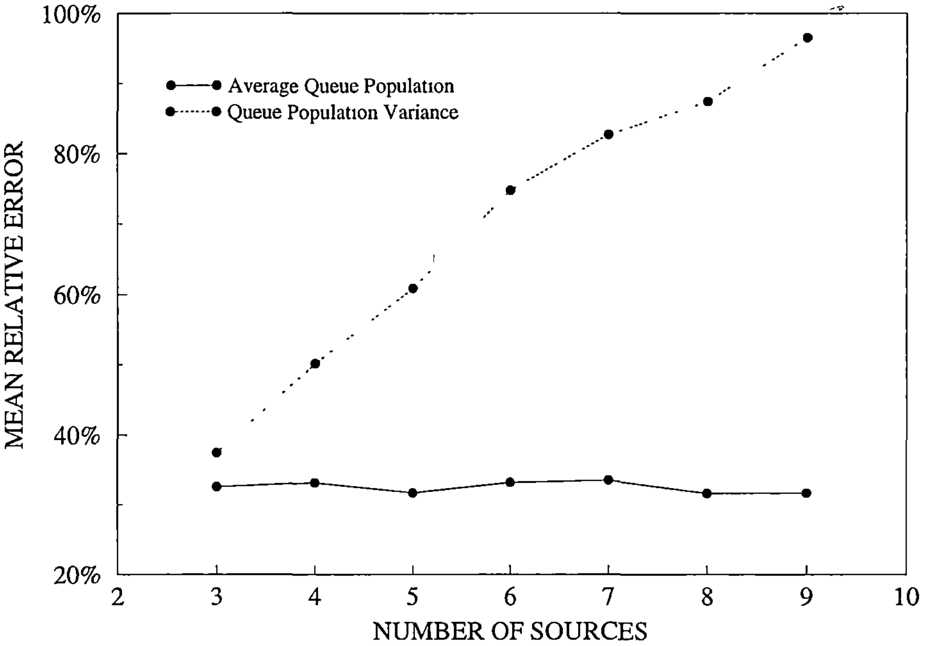

Figure 3.8: *Mean relative error in the average and variance of the queue population as a function of the number of sources for* 1000 *randomly generated queueing problems using the* **b** *vector approximation.*

N	Mean	Deviation	1st Percentile	99th Percentile
3	32.6%	83%	0.31%	380%
4	33.1%	77%	0.44%	360%
5	31.7%	62%	0.41%	310%
6	33.2%	61%	0.60%	360%
7	33.6%	58%	0.82%	330%
8	31.7%	44%	0.89%	220%
9	31.7%	47%	1.2%	270%

Table 3.6: *Statistics on the relative error between the approximate and exact solutions for the average queue population obtained from the* **b** *vector approximation. Each row in the table represents observations from* 1000 *randomly generated problems.*

N	Mean	Deviation	1st Percentile	99th Percentile
3	37.4%	79%	0.15%	330%
4	50.1%	85%	0.40%	390%
5	60.9%	94%	1.1%	490%
6	74.8%	120%	2.0%	640%
7	82.7%	110%	2.3%	520%
8	87.4%	110%	2.5%	520%
9	96.5%	120%	2.9%	630%

Table 3.7: *Statistics on the relative error between the approximate and exact solutions for the queue population variance obtained from combining the* **b** *vector approximation for the average with the geometric tail property.*

Although the mean relative error in the average queue population seems reasonably constant, the error in the variance increases steadily with the number of sources. This unfortunately rules out using this method for large N.

For completeness however, Tables 3.6 and 3.7 present the statistics of the relative error observed from the random generated problems for the average and variance of the queue population respectively. One interesting and perhaps useful property of this method is that the approximate average queue population is always greater than the exact average. This can be related back to the approximation for the **b** vector, which is based on relation (3.68). Whether this relation can guarantee that the approximation will *always* be greater than the exact average is not known.

The fact that the average queue population is always (or nearly always) greater than the actual average could also help explain why the queue population variance is on average so far below the actual variance. The variance is calculated from the second moment of the queue population (which will have some error associated with it) by subtracting the square of the average. If the error in the second moment is not more positive than the error from the square of the average, the approximate variance will underestimate the actual variance. Obviously, the observed results show that this is indeed the case.

3.6 A New Approximation

Although the approximations discussed above provide simpler and above all faster solution methods than the exact solution method, they generally only achieve good accuracy for a small range of system parameters (such as high utilisation, or high peak rates). As we have seen, their 'general' performance using randomly generated problems is extremely poor, which means other approximation methods must be found if high accuracy is desired. In the following we propose a new approximation for these types of problems, which we will introduce by means of an example. The process is fairly straightforward, although somewhat slower than the techniques discussed earlier.

Consider a queue fed by four geom-geom IBP sources, labelled A, B, C, and D. We will assume that the average queue population can be approximated as

$$L_q \approx \mathcal{M} + \mathcal{A} + \mathcal{B} + \mathcal{C} + \mathcal{D} + \mathcal{AB} + \mathcal{AC} + \mathcal{AD} + \mathcal{BC} + \mathcal{BD} + \mathcal{CD} \qquad (3.71)$$

where $\mathcal{M}$ is the queue population obtained when all sources are treated as Bernoulli processes (the marginal solution), $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ represent the increase in the average queue population when only one of source A, B, C, or D respectively becomes autocorrelated, and terms of the form $\mathcal{AB}$ represent the additional increase above that of $\mathcal{A}$ and $\mathcal{B}$ when both source A and B become autocorrelated. That is, terms of the form $\mathcal{AB}$ do not indicate a multiplication operation, but instead represent a logical relationship. This is why a calligraphic type font has been selected for representing these terms.

Thus, we have approximated L_q by a sum of terms due to the interaction of at most two autocorrelated sources. We assume contributions to the queue population due to the interaction of more than two sources are negligible. The motivation for investigating this approximation came from the results of Chapter 4. We consider in that chapter the case where all the θ_i terms of the IBP sources are equal to 1, which allows closed form solutions for the average and variance to be obtained. In those solutions, the average queue population is given exactly by autocorrelation contributions from each source alone, while the variance is given by contributions from pairs.

To make this process a little clearer, we provide numerical results for the case where the parameters of the four sources are described by Table 3.8. The contributions of each component are presented numerically in Table 3.9, to give an approximation for L_q of 6.3306666. Investigation of the exact queueing problem shows that the actual value of L_q is 6.327894 — a relative error in the approximation of only 0.04%.

Source	λ	θ	γ
A	0.1	0.7	0.6
B	0.5	0.8	0.3
C	0.1	0.4	0.1
D	0.2	0.4	0.9

Table 3.8: *Source parameters for approximation example*

Sources	L_q	Term	Expression	Value
none	2.5	$\mathcal{M}$	L_q	2.5
A	3.2799054	$\mathcal{A}$	$L_q - \mathcal{M}$	0.7799054
B	3.0014771	$\mathcal{B}$	$L_q - \mathcal{M}$	0.5014771
C	2.5294422	$\mathcal{C}$	$L_q - \mathcal{M}$	0.0294422
D	4.9295275	$\mathcal{D}$	$L_q - \mathcal{M}$	2.4295275
A and B	3.7839911	$\mathcal{AB}$	$L_q - \mathcal{M} - (\mathcal{A}+\mathcal{B})$	0.0026086
A and C	3.3093844	$\mathcal{AC}$	$L_q - \mathcal{M} - (\mathcal{A}+\mathcal{C})$	0.0000368
A and D	5.7439154	$\mathcal{AD}$	$L_q - \mathcal{M} - (\mathcal{A}+\mathcal{D})$	0.0344825
B and C	3.0310195	$\mathcal{BC}$	$L_q - \mathcal{M} - (\mathcal{B}+\mathcal{C})$	0.0001002
B and D	5.4823335	$\mathcal{BD}$	$L_q - \mathcal{M} - (\mathcal{B}+\mathcal{D})$	0.0513289
C and D	4.9607271	$\mathcal{CD}$	$L_q - \mathcal{M} - (\mathcal{C}+\mathcal{D})$	0.0017574

Table 3.9: *Development of contribution terms*

This same approach can also be applied to the calculation of the variance. For the numeric example discussed above, the approximate variance is 65.467832, while the actual variance is 65.460375 — a relative error in the approximation of just 0.01%. It would appear then that this method has the potential to provide a means whereby the average and queue population can be approximated to a very high degree of accuracy. We will study the accuracy of this approach in more detail in the rest of this chapter.

The strength of this method is that its solution requires the analysis of boundary conditions for at most two sources at once. Thus, with geom-geom IBP sources, we need only implement routines for solving linear systems of 2 and 4 unknowns, both of which can be performed very quickly. The evaluation of the average and variance of the queue population then only requires the modifications to the appropriate theory discussed in section 2.7. Construction of the overall solution is straightforward, using the above

N	Mean	Deviation	1st Percentile	99th Percentile
3	0.031%	0.22%	−0.22%	0.85%
4	0.055%	0.23%	−0.19%	0.96%
5	0.069%	0.29%	−0.49%	1.3%
6	0.083%	0.34%	−0.39%	1.4%
7	0.095%	0.32%	−0.33%	1.6%
8	0.108%	0.36%	−0.41%	1.7%
9	0.103%	0.35%	−0.44%	1.7%

Table 3.10: *Statistics on the relative error between the approximate and exact solutions for the average queue population. Each row in the table represents observations from* 1000 *randomly generated problems.*

N	Mean	Deviation	1st Percentile	99th Percentile
3	−0.051%	0.26%	−1.4%	0.10%
4	−0.102%	0.40%	−1.8%	0.13%
5	−0.131%	0.41%	−2.4%	0.12%
6	−0.190%	0.68%	−3.2%	0.12%
7	−0.205%	0.55%	−2.7%	0.10%
8	−0.203%	0.55%	−2.8%	0.13%
9	−0.228%	0.66%	−3.4%	0.14%

Table 3.11: *Statistics on the relative error between the approximate and exact solutions for the queue population variance. Each row in the table represents observations from* 1000 *randomly generated problems.*

example as a guide. Further enhancements to the method are possible however, and these will be discussed in section 3.6.2.

3.6.1 Accuracy Study

As for the other approximation methods, we assess the accuracy of the new method using randomly generated queueing problems for various numbers of sources with queue utilisations between 0.1 and 0.9. The results are presented in Tables 3.10 and 3.11, where it is immediately obvious that the error terms are at least two or three orders of magnitude better than the previous approximation techniques.

Although both the average and variance exhibit an increasing error magnitude with N, the result should remain small even for significantly more sources than we are able to consider here. We will look again at this trend in the mean relative error for up to 16 sources later in section 3.6.2.

Figures 3.9 and 3.10 show a scatter plot and histogram of the relative error in the average queue population approximation obtained from the analysis of 10,000 and 100,000 randomly generated queueing problems respectively using 4 sources. Although the distribution is very tight around the mean, there are still a significant number of outlying points. Since the queue load is obviously not a contributing factor (at least not by itself) to the instances of these larger errors, there must be other causes.

It turns out that the most significant factor in the accuracy of the approximation is, understandably, the autocorrelation parameter. As γ_i gets closer to 1, the magnitude of the observed errors increases considerably, and is greatly affected by the values of the remaining two parameters. To illustrate this point, Figure 3.11 shows the relative error in the average queue population observed for 4 and 8 identical sources, with autocorrelation parameters of 0.7 and 0.9, and a queue utilisation of 80%. The independent variable is θ_i — the probability that source i generates an arrival while in its active state.

Obviously then, there are situations where this approximation method becomes inaccurate, although the random generation scheme used above for allocating the parameters of the queueing problem finds these situations only infrequently. Observations suggest that if more than 2 sources have γ_i values of 0.9 or greater, then the high accuracy of the approximation may be in doubt. This number of sources is significant because we are approximating the queue behaviour by contributions from at most pairs of sources only. It also suggests a natural extension to the approximation to help improve the solution accuracy.

3.6.2 Improving the Approximation Accuracy

So far we have only considered the approximation using at most pairs of autocorrelated sources (the $\mathcal{AB}$ terms in equation 3.71). We can regard this as a 'second order' approximation, and the case where we only consider each source by itself (the $\mathcal{A}$ terms) as the 'first order' approximation. The marginal case (the $\mathcal{M}$ term resulting from having no sources with autocorrelation) is of course the 'zeroth order' approximation. As with most approximation schemes, we might expect that as the order of the approximation increases, the accuracy also increases. That is, considering additional terms of the form $\mathcal{ABC}$ (third order) would improve the accuracy of the second order method, and so on.

We find that this is indeed the case. Table 3.12 shows the mean and range of the error in the average queue population as a function of both the number of sources and the order of the approximation using highly autocorrelated sources. The γ_i parameters were chosen randomly from the range 0.9 to 0.99 rather than the 0 to 0.99 used previously

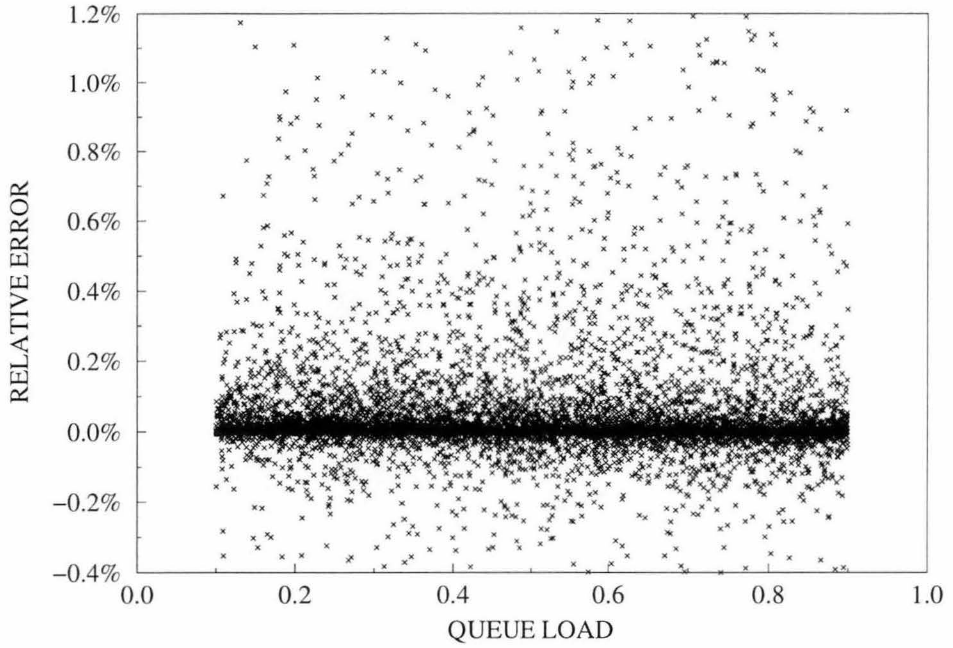

Figure 3.9: *Scatter plot of the relative error in the average queue population approximation as a function of the queue load for* 10, 000 *randomly generated queueing problems using* 4 *sources. The performance of the approximation seems largely unaffected by the queue load.*

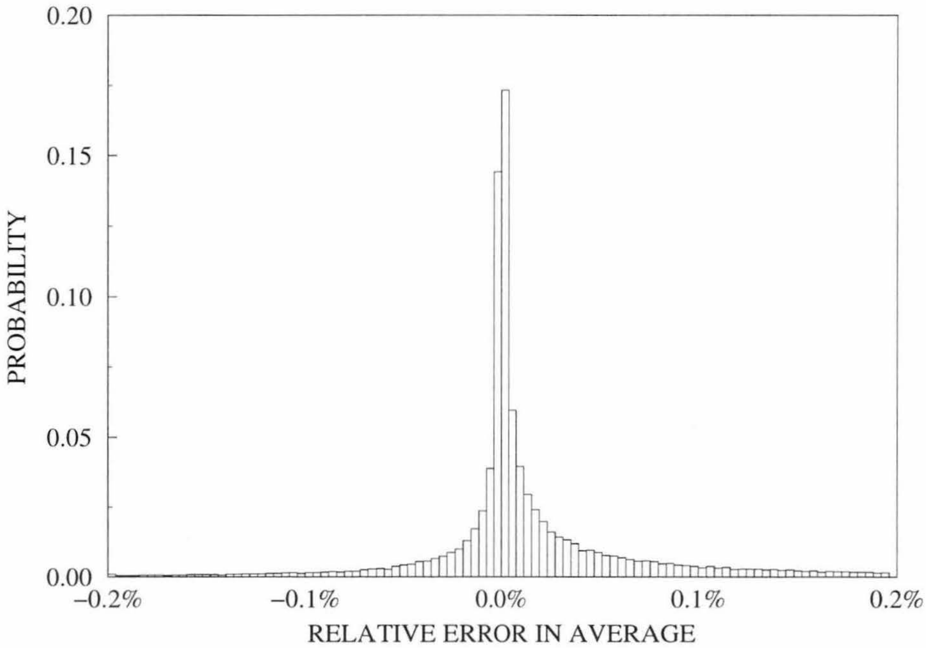

Figure 3.10: *Histogram of the distribution of the relative error for the approximation of the average queue population for* 4 *sources. The results were obtained from* 100, 000 *randomly generated problems, with* $\gamma_i < 0.9$ *in all cases.*

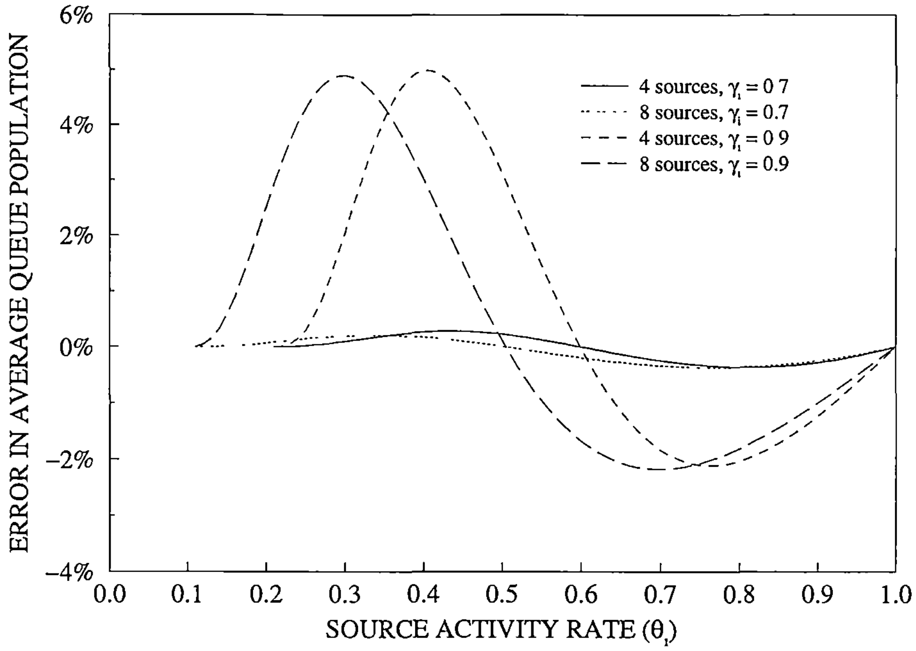

Figure 3.11: *Effect of the autocorrelation parameter on the relative error in the average queue population approximation for* 4 *sources.*

in Tables 3.10 and 3.11 so that the results represent a form of 'worst case' performance. The range is defined here to be the difference between the 99th and the 1st percentiles of the observed errors, and indicates a measure of their spread. The same data sets are used for the analysis of the accuracy of the each order of the approximation, and the entries marked with a '–' indicate that the solution is exact (the order of the approximation is equal to the number of sources).

Direct application of the approach illustrated by the example of Table 3.9 for higher order approximations leads to large memory requirements. This is because the evaluation of the *contribution* of each rth order term (those terms resulting from using r autocorrelated sources out of the N present) requires knowledge of the contributions of all the terms of orders up to $r - 1$. To avoid this problem, we can use the following approach, which we illustrate for the average queue population, but which is equally applicable for the variance.

Denote the sum of the average queue populations obtained from each combination of rth order terms by S_r. Then, by inspection of the sum of the contributions of each of these terms, it is possible to show that the desired kth order approximation for the overall average queue population is given by

$$\widetilde{L}_q^{(k)} = \sum_{r=0}^{k} \left(\sum_{j=0}^{k-r} (-1)^j \binom{N-r}{j} \right) S_r \tag{3.72}$$

	Second Order		Third Order		Fourth Order	
N	Mean	Range	Mean	Range	Mean	Range
3	0.996%	13%	–	–	–	–
4	1.66%	16%	−0.057%	3.0%	–	–
5	2.00%	18%	−0.106%	5.1%	−0.006%	0.88%
6	2.18%	16%	−0.176%	6.2%	−0.005%	1.9%
7	2.25%	19%	−0.221%	7.6%	0.019%	2.8%
8	2.40%	18%	−0.239%	8.7%	0.001%	3.7%
9	2.28%	17%	−0.197%	9.8%	−0.004%	4.8%

Table 3.12: *Mean and range of the relative error in the approximate average queue population as a function of the order of the approximation using highly autocorrelated sources. Each row in the table represents observations from* 1000 *randomly generated problems with* $0.1 \leq \lambda < 0.9$ *and* $0.9 \leq \gamma_i < 0.99$.

where $\binom{n}{r}$ denotes the combination function or binomial coefficient for n and r. Thus, to calculate the kth order result from the $(k-1)$th order result, only the k previous S_r terms need to be stored, and the result is available immediately once S_k is obtained. This method does have the disadvantage that it is more susceptible to round-off errors than calculating the individual contribution terms, and so it should always be used with double precision arithmetic.

One practical concern is knowing what order of approximation is required to provide a certain accuracy or relative error. The purpose of Tables such as 3.10 and 3.12 is to provide a general guide to the accuracies provided by this approximation method. However, it will sometimes be desirable to have better idea of the accuracy of the approximation result, particularly for the purpose of deciding whether an increase in the order of approximation is required. One way to gauge the possible error in the approximation is to compare the difference between the kth and $(k-1)$th order results. If this change is suitably small (depending on the desired accuracy) the kth order result can be accepted.

Alternatively, we might make use of the fact that increasing the order of the approximation always decreases the distance between the approximate result and the exact result. Thus if the magnitude of the kth order result is the greater of those from the $(k-1)$th, kth and $(k+1)$th order approximations, then the exact solution must be less than that of the kth order. Similarly, if the kth order result is the lesser of the three, then the exact solution must be greater than kth order value. Thus, in this fashion we may either bracket the exact result, or use the maximum value as a tight upper bound.

N	Mean	Deviation	1st Percentile	99th Percentile
10	0.102%	0.36%	−0.49%	1.4%
11	0.107%	0.36%	−0.47%	1.8%
12	0.102%	0.34%	−0.48%	1.6%
13	0.098%	0.33%	−0.46%	1.5%
14	0.098%	0.35%	−0.57%	1.2%
15	0.104%	0.37%	−0.65%	1.6%
16	0.097%	0.33%	−0.60%	1.5%

Table 3.13: *Statistics of the error in the average queue population of the second order approximation measured relative to the more exact fourth order approximation. Each row in the table represents observations from* 1000 *randomly generated problems.*

Using high order approximations for 'exact' results

Although high order approximations would be impractical for real time decision making processes, they can serve to provide accurate results where actual exact analysis cannot be used. For example, we might wish to evaluate the performance of the geometric tail approximation for 16 sources or more. Since it is impossible (or at least impractical) with modern computing facilities to try and solve for an exact 16 source solution, current methods have been limited to the assumption that sources can be grouped into classes of identical processes (see [141] for example). The new method discussed here provides an alternative, since most solutions can be obtained to very high accuracy using only third or fourth order approximations.

As a more illustrative example, we are concerned with the trend in the relative errors observed for the second order approximation as the number of sources increases. We have previously obtained results from comparison with the exact analysis of up to 9 sources in Tables 3.10 and 3.11. In Figure 3.12 and in Tables 3.13 and 3.14 we continue the analysis of the mean relative error for up to 16 sources by using a fourth order approximation as the source of the 'exact' results. Importantly we find that the increase in the error of the second order approximation remains quite small. In the case of the average queue population, the relative error appears to have stopped increasing after about 8 sources. We expect that the variance result may also exhibit this behaviour for large enough N, although this is not visible in these results.

Since the errors in the fourth order approximation are known to be several orders of magnitude smaller than those of the second (at least for up to 9 sources) we expect the calculated error results to be fairly precise for all N. Figure 3.12 also includes the mean errors observed for the average and variance relative to the exact results for up to 9 sources, and seems to indicate that the fourth order approximation serves quite

N	Mean	Deviation	1st Percentile	99th Percentile
10	−0.245%	0.65%	−3.1%	0.18%
11	−0.241%	0.63%	−3.2%	0.16%
12	−0.289%	0.70%	−3.1%	0.14%
13	−0.283%	0.67%	−3.1%	0.11%
14	−0.261%	0.56%	−2.6%	0.18%
15	−0.291%	0.71%	−3.7%	0.14%
16	−0.301%	0.67%	−3.3%	0.11%

Table 3.14: *Relative error statistics for the queue population variance corresponding to Table 3.13.*

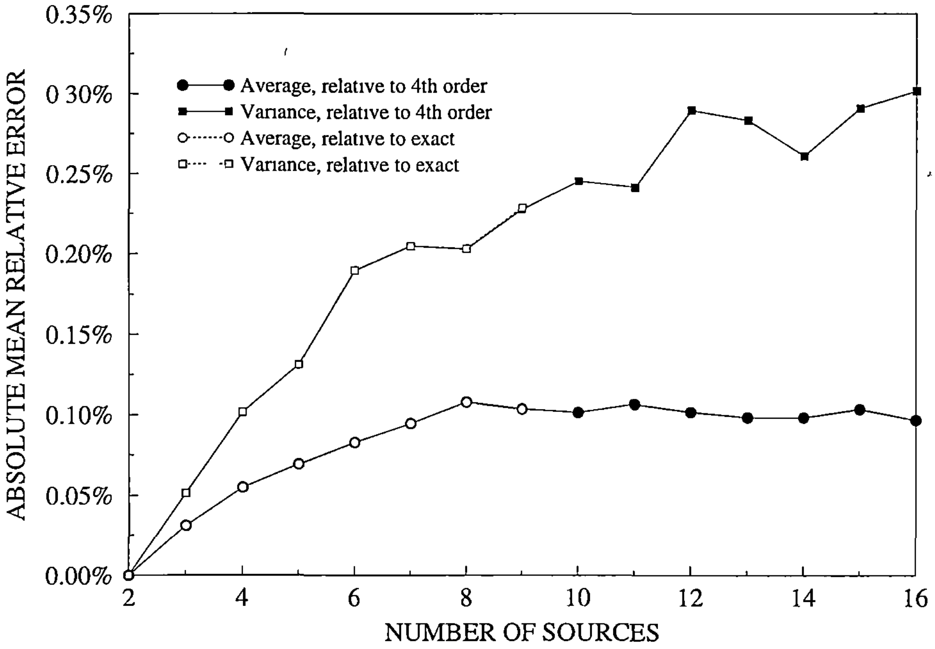

Figure 3.12: *Mean error in the second order approximation for the average and variance of the queue population measured relative to the fourth order approximation results. The results were calculated for queue loads between* 0.1 *and* 0.9 *and with* $0 \leq \gamma_i < 0.99$.

well as an 'exact' solution.

To be more sure of the accuracy of these error results, third and fifth order approximations could be performed, and the change in the results considered. However, since the second order results are so small anyway, we have not bothered to investigate the accuracy any further here.

3.6.3 Approximation Run Times

We have already mentioned that this new approximation, while considerably more accurate than the more common approximation techniques discussed in sections 3.2 to 3.5, is also slower to obtain results. Here we will discuss some of the computational requirements of the approximation, and present some actual run times.

The approximation breaks the solution of each queueing problem into combinations of r terms, where we denote the largest value of $r = k$ as the order of the approximation. Simple algebra shows that for N sources, there will be $\binom{N}{r}$ combinations of r autocorrelated sources, and hence the total run time will be approximately

$$T = \sum_{r=0}^{k} \binom{N}{r} g_{r,N} \tag{3.73}$$

where $g_{r,N}$ is the time required to solve for a queueing problem involving r autocorrelated sources out of N total. In the limit as N and r become large, we can approximate this by

$$T \propto N^k 8^k \tag{3.74}$$

where we have used the fact that the solution of a queue fed by r geom-geom IBP sources varies as 8^r for large r. This usual nomenclature for this type of limiting behaviour is to say that it has a *time complexity* of order $O\left(N^k 8^k\right)$. The results of section 3.1.5 indicate that for small N, and in particular for small k (which is where the greatest time advantage will be obtained in the approximation) the time T to obtain the solution will probably be significantly smaller than indicated by equation (3.74).

On another note, if the sources are all identical, then each term in r sources from 0 to k needs only to be evaluated *once*, significantly reducing the overall execution time. Similarly, for problems involving classes of identical sources, the total number of different combinations involving r sources will be significantly lower than the heterogeneous case. However, as we have mentioned previously, although these pathological cases are popular in the literature, they are not particularly useful for queueing problems involving switches deep within the network.

N	Second	Fourth
10	170 msec	4.9 sec
11	200 msec	7.5 sec
12	230 msec	11 sec
13	270 msec	16 sec
14	320 msec	22 sec
15	360 msec	30 sec
16	410 msec	39 sec

Table 3.15: *Mean run times for the second and fourth order approximations on the IBM RS6000/320H.*

Actual execution times for the approximation were obtained on an IBM RS6000/320H using 1000 randomly generated queueing problems for each N. Figure 3.13 shows a logarithmic plot of the calculation run times as a function of the number of sources. In addition, Table 3.15 shows the run times on the RS6000 for the second and fourth order approximations for 10 to 16 sources.

We note that Figure 3.13 clearly indicates that there is a time advantage to using the approximation *only* when the number of sources is at least three more than the order of the approximation. Thus for problems with only 4 sources, the best solution approach method is the exact one.

3.6.4 Other Variations

A variation on the approach discussed above is to always include one or two particular sources in every calculation used to obtain the approximate queue population moments. That is, referring to the example at the beginning of this section, we might always include source A as an autocorrelated source, resulting in only the terms $\mathcal{A}$, $\mathcal{AB}$, $\mathcal{AC}$, and $\mathcal{AD}$ being used to construct a 'second order' solution to the average and variance (see Table 3.16). In this case, the approximate average is 6.2774801 — an error of 0.8%. The accuracy of the result is dependent to an extent on which source(s) are chosen to be used in each calculation. For example, choosing source D as the one to use in each calculation in this example yields an almost exact result, with an error of just 0.0004%.

We will refer to solutions formed from combinations of k autocorrelated sources but with s of these used in every calculation as 'kth order with s held' approximations. The reason for using this approach rather than the straight kth order method discussed earlier is simply that it is faster, although generally with poorer accuracy. The reason for the increased speed is not hard to see. Since s of the sources must always be used, only combinations involving s or more sources need be considered, and in addition,

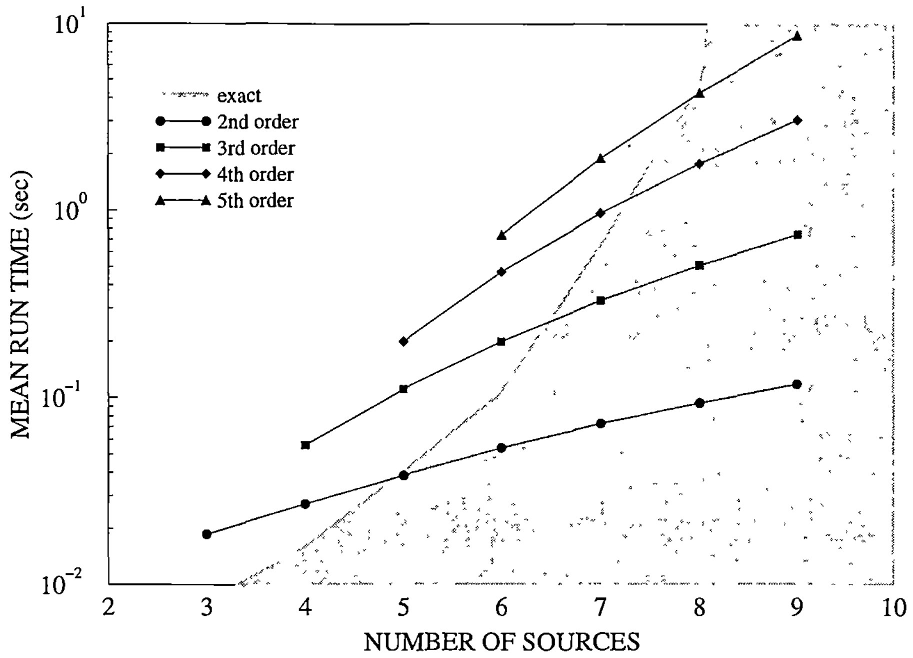

Figure 3.13: *Mean run times for the second to fifth orders of the approximation. Each time was obtained from* 1000 *solutions on an IBM RS6000/320H workstation.*

Sources	L_q	Term	Expression	Value
A	3.2799054	a	L_q	3.2799054
A and B	3.7839911	ab	$L_q - a$	0.5040858
A and C	3.3093844	ac	$L_q - a$	0.0294790
A and D	5.7439154	ad	$L_q - a$	2.4640100

Table 3.16: *Development of contribution terms*

the number of combinations of r sources reduces from $\binom{N}{r}$ to $\binom{N-s}{r-s}$. This means that the upper limit on the time complexity of the approximation becomes $O\left(8^k (N-s)^k\right)$ which can provide a considerable time saving for larger k.

We mentioned above that the choice of which sources are to be 'held' or always included, can affect the accuracy of the final results. From experimenting with various choices, it appears that the best approach is to 'hold' those s sources having the largest autocovariance sums. That is, we choose those sources having the largest values of Υ_i which we define as

$$\Upsilon_i = \lambda_i (\theta_i - \lambda_i) \frac{\gamma_i}{1-\gamma_i} \tag{3.75}$$

As we noted previously, the autocorrelation parameter γ_i plays the largest role in determining the performance of the approximation method. The use of the autocovariance sum in this form provides a convenient metric for determining which sources will have the greatest impact on the approximation accuracy.

In general, a kth order approximation performs better than a kth order with s held. Figures 3.14 and 3.15 compare the absolute value of the mean relative error for the average and variance respectively of the queue population as a function of N for several approximation orders with and without a single held source. In each randomly generated queueing problem, the individual source autocorrelation parameters were selected from the range of 0.9 to 0.99 in order to maximise the resulting approximation errors. As before, 1000 queueing problems make up the observations for each N.

Figure 3.16 compares the run time performances of the kth order approximations with and without a held source as a function of N. As in the previous subsection, the run times are from an IBM RS6000/320H, and the times required to actually generate the 1000 problems and pass them to the calculation program have been subtracted from the measured run times. For comparison, the run times for the exact solution are also shown.

From these results we see that an approximation of order $(k+1)$ with 1 held source has an accuracy and a mean run time that usually falls between those of the kth and $(k+1)$th order approximations without held sources. Thus, if a kth order approximation is not quite accurate enough, but a $(k+1)$th order approximation is too expensive in terms of run time, an alternative would be to use a $(k+1)$th order approximation with 1 held source. We have not investigated the advantages (or otherwise) of using more than one held source.

In Chapter 5, we will consider this approach again in the context of approximating the performance of queues with cyclically interrupted service (or cyclic arrivals).

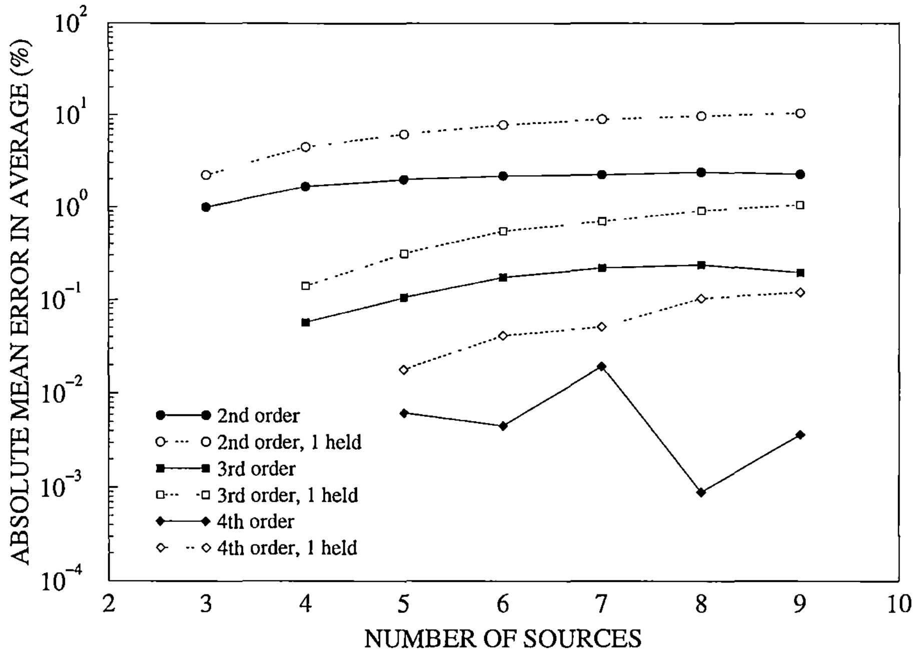

Figure 3.14: *Mean relative error in the average queue population using approximations with and without held sources. Each row in the table represents observations from* 1000 *highly autocorrelated, randomly generated problems with* $0.1 \leq \lambda < 0.9$ *and* $0.9 \leq \gamma_i < 0.99$.

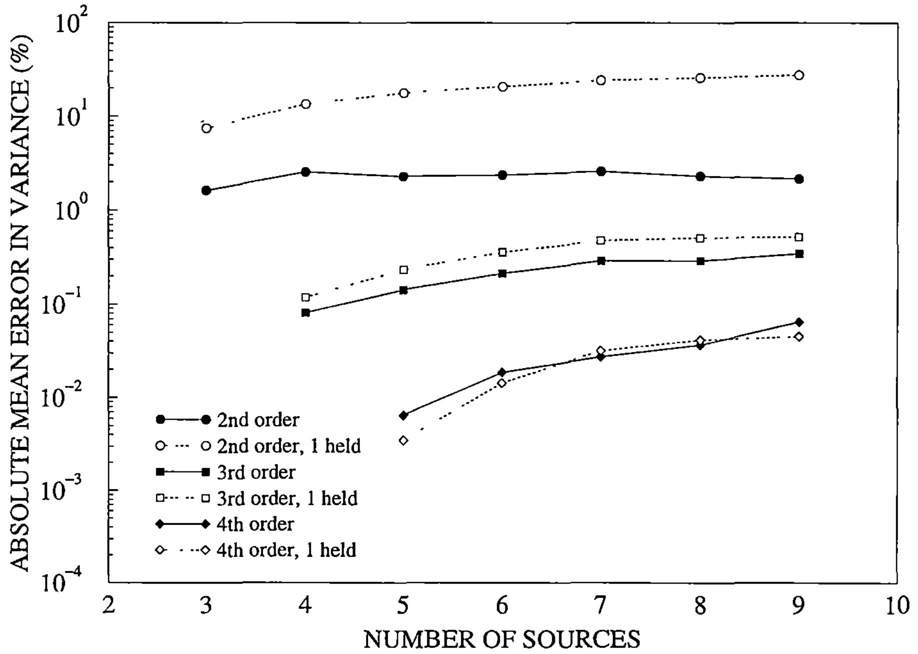

Figure 3.15: *Mean relative error in the queue population variance, corresponding to Figure 3.14.*

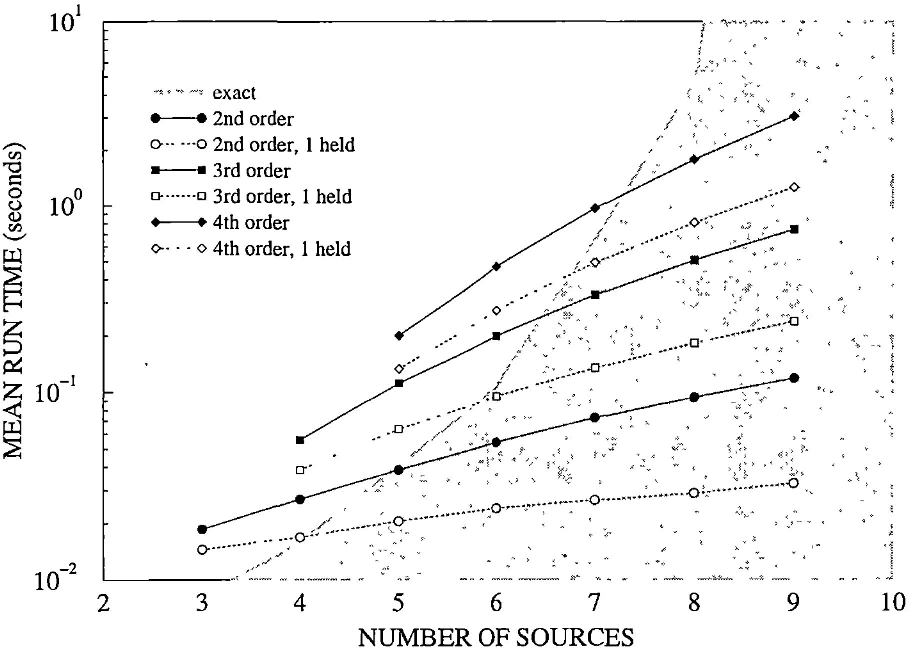

Figure 3.16: *Mean run times for the second, third, and fourth order approximations, with and without a held source. Each time was obtained from* 1000 *solutions on an IBM RS6000/320H workstation.*

3.7 Summary

In this chapter we have discussed the analysis of the moments of the population of a G/D/1 queue fed by a number of generally non-identical geom-geom IBP sources. The average and variance are calculated in a straightforward manner using the results of Chapter 2 applied to this specific problem. For queueing problems involving small numbers of sources, the solution is fast enough to be applied directly. However for larger numbers of sources, the time required to solve the queueing problem becomes quickly impractical, and for more than about 12 sources, the solution simply cannot be obtained. Thus for the larger switch sizes that might be expected in ATM networks approximate solution techniques must be used.

Investigation of approximate solutions based on the geometric tail property, the MMPP model, and an approximation for the empty system vector, were performed for sets of randomly generated queueing problems. Rather than making the assumption that the queue load is high, we have chosen to use a wide range of utilisations and source parameters, as might be expected within a real network switch. As a result, very poor accuracies were observed for all three approximation methods.

A new approximation technique was then presented, which is somewhat slower than the above methods, but has an extremely high accuracy that is independent of the queue utilisation. The second order approximation has a time order complexity of approximately $O\left(N^2\right)$, but only provides faster computation of the population moments than the exact method for queues with 6 or more sources. The accuracy of the method is improved by increasing the order of the approximation, but at the cost of rapidly increasing execution time. The higher order approximations are so accurate that they can be used to provide effectively 'exact' results in situations where the formally exact solution method cannot be applied.

Chapter 4

Population Analysis for Phase-Geometric Binary Arrival Models

A special case of the geom-geom interrupted Bernoulli arrival process considered in the previous chapter is the case where the peak rates of the sources are all equal to the outgoing link rate, so that each source generates a single arrival in every time slot for which it is active (all the θ_i are equal to 1). In order to distinguish this case from the previous one, and to extend the analysis to include phase-type distributions for the active periods, these sources will be referred to as *phase-geom Binary* sources, although in truth both types of source are binary processes (they generate either one or no arrivals in each time slot). The main reason for considering these type of sources separately is that closed form expressions for both the average and the variance of the infinite buffer queue population can be found. This chapter is devoted to presenting these closed form solutions.

4.1 Related Studies

In 1986, Viterbi [133] presented an exact closed form solution for the average population of a discrete-time G/D/1 queue fed by a number of heterogeneous on-off binary sources having geometric on and off periods (geom-geom sources). This solution was expanded by Neuts [98] to include sources where the active periods could be described by a phase type distribution (phase-geom sources). Similarly Dupuis and Hajek [24] rederived Neuts result and presented the equivalent solution both for the continuous time domain, and for sources that transmit one or more arrivals (i.e. non-binary sources) in each active period. Bruneel [15] had earlier considered the non-binary arrivals problem for identical geom-geom sources, also presenting a closed form solution for the average queue population.

The main problem with these closed form results is that they provide *only* the first moment of the queue population. In order to make approximations for the purpose of estimating losses in finite buffers, or estimating bounds on the queueing delays, at least the first and second moments of the queue population are required. In [15], Bruneel indicates that the variance for the homogeneous geom-geom sources problem can be found in the same manner as the average, although this result is not presented in the paper, and investigation into Bruneel's method by this author failed to provide the desired result.

In [33] Gordon considered a heterogeneous mix of non-binary geom-geom sources. The analysis presented explicit derivations for the z-transforms of the queue population and delay, with a reported computational complexity of the order of 2^N. For the case where all the sources are identical (as in [15]), this complexity reduces to order N. As for the IBP problems in the previous chapter, a solution complexity of 2^N is too large to use in all circumstances. As an alternative approach, Sohraby discusses an approximate solution method in [120] and [121] using a geometric approximation to the tail probabilities of the queue (see Chapter 6 for more discussion on this method). The numerical complexity of the solutions is only of order N, although the accuracy is not particularly good, except at very high utilisations.

Thus, until now, no closed form solution for the queue population variance using heterogeneous sources (either the geom-geom or phase-geom) had been established. In a recent paper [108], this author presented a partly empirically observed solution for the variance of the heterogeneous problem using geom-geom Binary sources. That solution was obtained by observation of the symbolic solution to the queueing equation (2.8) for 3 sources using *Mathematica* [139], and then shown to be exact for larger number of sources as well, through comparison with numeric solutions. Although the closed form solution for this case was a useful discovery, a robust proof of the result was still lacking.

In [98], Neuts remarked that "Fluctuations in the queue length (and therefore the delay of packets), which are of the utmost importance to applications, will require investigation by innovative and non-traditional methods." (page 95). In section 4.3 of this chapter, a general solution for the variance of the queue population for this problem is presented. As will be seen in that section, the approach taken to find the solution is definitely non-traditional. The computational complexity of the solutions is of order N for the heterogeneous case, and requires only three parameters per source for phase-geom Binary arrival processes, or two per source for geom-geom processes.

To start with however, we will reproduce Neuts result for the average queue population, but using our notation.

4.2 Average Queue Population for Phase-Geometric Binary Sources

From equation (2.36) of Chapter 2, the general form of the average queue population of a discrete time G/D/1 queue fed by batch Markov arrival processes is given by

$$L_q = \frac{\mathbf{b}\mathbf{v}'(1)}{1-\lambda} + \frac{\delta''(1)}{2\,(1-\lambda)} \tag{4.1}$$

For the general problem, the entries of the 'empty queue' probability vector $\mathbf{b}$ have to be determined using numeric techniques, which precludes any closed form solution for the average (and variance) being obtainable. In the phase-geom special case however, each source generates a single arrival in every time slot for which it is active. Thus, if there is one or more sources active in a time slot, the queue is guaranteed not to be empty immediately prior to service. Thus, the only non-zero entries in the vector $\mathbf{b}$ will correspond to the states for which all arrival processes are silent.

For the phase-geom arrival process problem, each source has a single silent state, resulting in only a single state in the overall arrival process for which all the sources are silent. Thus, the vector $\mathbf{b}$ can be written as

$$\mathbf{b} = [b_0, \mathbf{0}]$$

where the silent state is taken to be state 0 by convention. From equation (2.17) we have $\mathbf{b}\mathbf{e} = 1 - \lambda$, giving $b_0 = 1 - \lambda$ and hence

$$L_q = v_0'(1) + \frac{\delta''(1)}{2\,(1-\lambda)} \tag{4.2}$$

where $v_0(z)$ denotes the element of the right-hand Perron–Frobenius eigenvector corresponding to state 0 of the overall arrival process. From the properties of Kronecker products

$$v_0(z) = \prod_{i=1}^{N} v_{i,0}(z) \tag{4.3}$$

where $v_{i,0}(1)$ denotes the zero state element of the right-hand Perron–Frobenius eigenvector for source i. Thus

$$v_0'(1) = \sum_{i=1}^{N} v_{i,0}'(1) \tag{4.4}$$

since $v_{i,0}(1) = 1$. From Appendix C

$$v'_{i,0}(1) = \frac{-\delta''_i(1)}{2\,(1-\lambda_i)} \tag{4.5}$$

where $\delta_i(z)$ is the Perron–Frobenius eigenvalue for source i.

From Appendix C also, we obtain

$$\delta''(1) = M_2 - \lambda + \sum_{i=1}^{N} \delta''_i(1) \tag{4.6}$$

where $\delta''_i(1)$ is given by

$$\delta''_i(1) = \frac{\varepsilon_{i,2} - 2\varepsilon^2_{i,1} - \varepsilon_{i,1}}{(1+\varepsilon_{i,1})^3} \tag{4.7}$$

for which the $\varepsilon_{i,r}$ are directly related to the physical parameters of the phase-geom arrival process by

$$\varepsilon_{i,r} = \frac{\lambda_i}{1-\lambda_i} \times \frac{\eta_{i,r}}{\eta_{i,1}} \tag{4.8}$$

where $\eta_{i,r}$ describes the rth moment of the duration of the active periods of source i. Alternatively, using Neuts' approach [98] to describing the burstiness or autocorrelation of the binary source, we redefine $\delta''_i(1)$ as

$$\delta''_i(1) = 2\lambda_i\,(1-\lambda_i)\,\frac{\gamma_i}{1-\gamma_i} \tag{4.9}$$

where the parameter γ_i (referred to as the autocorrelation parameter) is given by

$$\gamma_i = 1 - \frac{2\eta_{i,1}}{(1-\lambda_i)\,(\eta_{i,2}+\eta_{i,1})} \tag{4.10}$$

giving the average queue population finally as

$$L_q = \frac{M_2-\lambda}{2\,(1-\lambda)} + \frac{1}{1-\lambda}\sum_{i=1}^{N} \lambda_i\,(\lambda-\lambda_i)\,\frac{\gamma_i}{1-\gamma_i} \tag{4.11}$$

Correcting for the difference between the average queue population and the average system population, equation (4.11) is algebraically equivalent to Neuts solution in [98], although in a simpler form.

The Physical Meaning of the Autocorrelation Parameter

In Chapter 3, the autocorrelation parameter of a geom-geom IBP source is related to the autocorrelation coefficient function $R_i(m)$ by

$$R_i(m) = \left(\frac{\theta_i - \lambda_i}{1_i - \lambda_i}\right)\gamma_i^{|m|} \tag{4.12}$$

where $\theta_i = 1$ for the types of sources considered here. The single-sided autocorrelation coefficient sum $S_\imath$ is then defined for geom-geom Binary source i by

$$S_\imath = \sum_{m=1}^{\infty} R_\imath(m) = \frac{\gamma_i}{1-\gamma_\imath} \tag{4.13}$$

which is a term that appears in the equation for the average queue population (and also in various forms in the variance equation derived later).

Since equation (4.11) applies both to geom-geom and phase-geom Binary sources, we might speculate on whether the relationship of equation (4.13) holds for phase-geom Binary processes as well. Pieloor and Lewis first suggested this relation in [107], although they were unable to provide a proof of the result. The relation can be proved however, and is provided as Theorem D.1 in Appendix D. That is, the autocorrelation parameter for a phase-geom Binary source is directly related to the single-sided sum of the autocorrelation coefficient function by equation (4.13).

The fact that this definition of the autocorrelation parameter is related to the actual autocorrelation function in this manner suggests that the average burst length $\eta_{i,1}$ alone (used in [120] for example) is not an adequate descriptor for general distributions of the active period. The exception to this is when the active periods are geometrically distributed (as commonly used) where knowledge of the average burst length allows the higher moments to be calculated directly.

So how does knowledge of the fact that equation (4.13) holds generally for phase-geom Binary sources help us? We note that the asymptotic variance of a general arrival process is defined by

$$v = \lim_{n\to\infty} \frac{1}{n} \operatorname{Var}\left[N(n)\right] \tag{4.14}$$

where $N(n)$ is the number of arrivals occurring in an arbitrary chosen group of n consecutive time slots. This is related to the single-sided autocorrelation sum S of the process by

$$v = \sigma^2 \left(1 + 2S\right) \tag{4.15}$$

where σ^2 is the variance of the arrival process[1]. Since we can make an approximate measurement of v for a real world source, say by choosing $n = 1000$ or $n = 10000$ in equation (4.14), we obtain an estimate for the autocorrelation parameter that helps describe that source.

[1] This relation can be derived simply from the well known result for the variance of the sum of random variates.

4.2.1 Non-Binary Sources

This result for the average queue population can be easily extended to the non-binary case, where each source generates *at least* one arrival per time slot for which it is active. This is the problem considered by Bruneel [15], Gordon [33], and Dupuis and Hajek [24]. Using basically the same approach as for the binary case above, we obtain

$$L_q = \frac{M_2 - \lambda}{2(1-\lambda)} + \frac{1}{1-\lambda}\sum_{i=1}^{N} \lambda_i \left(\lambda - \lambda_i + p_{i,1} - 1\right) \frac{\gamma_i}{1-\gamma_i} \tag{4.16}$$

where $p_{i,1}$ denotes here the first moment of the arrival process from source i when it is in its active state, and the autocorrelation parameter of each source is now defined by

$$\gamma_i = 1 - \frac{2p_{i,1}\eta_{i,1}}{\left(p_{i,1} - \lambda_i\right)\left(\eta_{i,2} + \eta_{i,1}\right)} \tag{4.17}$$

with the second moment of the combined arrival process given by

$$M_2 = \lambda^2 + \sum_{i=1}^{N} \lambda_i \frac{p_{i,2}}{p_{i,1}} - \sum_{i=1}^{N} \lambda_i^2 \tag{4.18}$$

where $p_{i,2}$ denotes the second moment of the arrival process from source i when it is in its active state.

4.2.2 Identical Sources

For N identical sources, we have $\lambda_i = \lambda/N$ and $\gamma_i = \gamma$, for which L_q reduces to

$$L_q = \frac{\lambda^2}{2(1-\lambda)}\left(\frac{1+\gamma}{1-\gamma}\right)\left(1 - \frac{1}{N}\right) \tag{4.19}$$

or as $N \to \infty$

$$L_q = \frac{\lambda^2}{2(1-\lambda)}\left(\frac{1+\gamma}{1-\gamma}\right) \tag{4.20}$$

which is the upper limiting value for the queue population of a large number of identical sources.

4.3 Queue Population Variance for Phase-Geometric Binary Sources

In Chapter 2, the general form for the population variance of a queue fed by discrete-time batch Markov arrival process was given in equation (2.51). Here we will make use of the special properties of the phase-geom Binary process to provide a unique solution for the queue population variance of a G/D/1 queue fed by these sources. The basic form of the variance solution is

$$\mathrm{Var}\left[L_q\right] = \frac{\mathbf{b}\mathbf{v}''(1)}{1-\lambda} - 2\mathbf{X}'(1)\mathbf{v}'(1) - \mu\mathbf{v}''(1) + \frac{\delta'''(1)}{3\left(1-\lambda\right)} + \left(1 + \frac{\delta''(1)}{1-\lambda}\right) L_q - L_q^2 \quad (4.21)$$

where it was also shown that

$$2\mathbf{X}'(1)\mathbf{v}'(1) - \mu\mathbf{v}''(1) = 2\sum_{i=1}^{N}\sum_{j=1}^{m_i-1}\left(\frac{\mathbf{b}\mathbf{g}_{(i,j)}(1)}{s(1)} \times \frac{\mathbf{h}_{i,j}(1)\mathbf{g}'_{i,0}(1)}{1-\omega_{i,j}(1)}\right) \quad (4.22)$$

In this case, $\mathbf{b} = [1-\lambda, \mathbf{0}]$ and so

$$\frac{\mathbf{b}\mathbf{g}_{(i,j)}(1)}{s(1)} = \frac{1}{s(1)}\left(1-\lambda\right) g_{i,j,0}(1) \prod_{n=1, n\neq i}^{N} s_n(1) \quad (4.23)$$

where $g_{i,j,0}(1)$ is the element of the jth eigenvector of source i corresponding to state 0 of that source, and $s_n(1)$ is obtained from

$$\mathbf{g}_{n,0}(1) = s_n(1)\mathbf{e} \quad (4.24)$$

Then, noting that

$$s(1) = \prod_{n=1}^{N} s_n(1) \quad (4.25)$$

we obtain

$$\frac{\mathbf{b}\mathbf{g}_{(i,j)}(1)}{s(1)} = \left(1-\lambda\right)\frac{g_{i,j,0}(1)}{s_i(1)} \quad (4.26)$$

which is a function of source i and selected eigenvalue j only. Thus we can write

$$\begin{aligned} 2\mathbf{X}'(1)\mathbf{v}'(1) - \mu\mathbf{v}''(1) &= 2\left(1-\lambda\right)\sum_{i=1}^{N}\left(\sum_{j=1}^{m_i-1}\frac{g_{i,j,0}(1)\mathbf{h}_{i,j}(1)\mathbf{g}'_{i,0}(1)}{s_i(1)\left(1-\omega_{i,j}(1)\right)}\right) \\ &= 2\left(1-\lambda\right)\sum_{i=1}^{N} F_i \end{aligned} \quad (4.27)$$

for scalar F_i which is obviously a function only of the parameters of source i, and is independent of the other sources. Thus, if the form of F_i can be established for any one source, the overall solution to $2\mathbf{X}'(1)\mathbf{v}'(1) - \mu\mathbf{v}''(1)$ can be constructed directly. Then, since every other term in the solution for $\mathrm{Var}\left[L_q\right]$ can be established, the variance solution will be complete.

4.3.1 Variance for a single Phase-Geometric Binary source

Consider the queueing problem when only one of the N Binary sources is of phase-geom type, and the remaining $N-1$ sources are described by Bernoulli processes. Then the queue equation (2.8) can be rewritten as

$$\mathbf{X}(z)\left(z\mathbf{I} - p(z)\mathbf{A}_i\mathbf{P}_i(z)\right) = \left(z-1\right)\mathbf{b} \quad (4.28)$$

where $\mathbf{A}_i$ and $\mathbf{P}_i(z)$ describe the behaviour of the phase-geom Binary source i, and $p(z)$ describes the random arrival process resulting from the superposition of the remaining $N-1$ sources, where

$$p'(1) = \lambda - \lambda_i \tag{4.29}$$

$$p''(1) = M_2 - \lambda - 2\lambda_i\left(\lambda - \lambda_i\right) \tag{4.30}$$

$$p'''(1) = M_3 - 3\left(1+\lambda_i\right)M_2 + \lambda\left(2+3\lambda_i\right) + 6\lambda_i^2\left(\lambda - \lambda_i\right) \tag{4.31}$$

so that the superposition of the phase-geom Binary source and the random process $p(z)$ yields the same stationary marginal arrival process moments, denoted by λ, M_2, and M_3 as in the case with N phase-geom Binary sources.

The aim here is to find an expression for $2\mathbf{X}'(1)\mathbf{v}_i'(1) - \boldsymbol{\mu}\mathbf{v}_i''(1)$, where $\mathbf{v}_i(z)$ denotes the right-hand Perron–Frobenius eigenvector of $\mathbf{A}_i\mathbf{P}_i(z)$. In the following, we will write $\mathbf{X}'(1)$ as $[x_0'(1), \mathbf{x}_1'(1)]$, $\boldsymbol{\mu}_i$ as $\left[\mu_{i,0}, \boldsymbol{\mu}_{i,1}\right]$, and $\mathbf{v}_i'(1)$ as $\left[v_{i,0}'(1), \mathbf{v}_{i,1}'(1)\right]$ where the 0 (or $i,0$) subscript denotes the first element of the relevant vector, and the 1 (or $i,1$) subscript and bold face type denotes the remainder of the vector. Then $\mathbf{X}'(1)\mathbf{v}_i'(1)$ will be given by

$$\mathbf{X}'(1)\mathbf{v}_i'(1) = x_0'(1)v_{i,0}'(1) + \mathbf{x}_1'(1)\mathbf{v}_{i,1}'(1) \tag{4.32}$$

and using

$$\mathbf{v}_{i,1}'(1) = \left(v_{i,0}'(1) - 1\right)\mathbf{e} + \left(1-\lambda_i\right)\left(\mathbf{I} - \mathbf{T}_i\right)^{-1}\mathbf{e} \tag{4.33}$$

from Appendix C gives

$$\mathbf{X}'(1)\mathbf{v}_i'(1) = x_0'(1) + \left(1-\lambda_i\right)\mathbf{x}_1'(1)\left(\mathbf{I}-\mathbf{T}_i\right)^{-1}\mathbf{e} + \left(v_{i,0}'(1) - 1\right)L_q \tag{4.34}$$

from which we need to find $x_0'(1)$ and $\mathbf{x}_1'(1)\left(\mathbf{I}-\mathbf{T}_i\right)^{-1}\mathbf{e}$.

The first derivative of equation (4.28) evaluated at $z=1$ is

$$\mathbf{X}'(1)\left(\mathbf{I}-\mathbf{A}_i\right) + \left(1-\lambda+\lambda_i\right)\boldsymbol{\mu}_i - \boldsymbol{\mu}_i\mathbf{P}_i'(1) = \mathbf{b} \tag{4.35}$$

Taking the first element of this vector equation gives

$$\mathbf{X}'(1)\begin{bmatrix} 1-c_i \\ -\mathbf{T}_i^{\mathrm{o}} \end{bmatrix} = \lambda_i\left(\lambda_i - \lambda\right) \tag{4.36}$$

while considering the remaining elements gives

$$\mathbf{X}'(1)\begin{bmatrix} -c_i'\boldsymbol{\alpha}_i \\ \mathbf{I}-\mathbf{T}_i \end{bmatrix} = \left(\lambda - \lambda_i\right)\boldsymbol{\mu}_{i,1} \tag{4.37}$$

which can be written as

$$\mathbf{x}_1'(1)\left(\mathbf{I}-\mathbf{T}_i\right) = x_0'(1)c_i'\boldsymbol{\alpha}_i + \left(\lambda-\lambda_i\right)\boldsymbol{\mu}_{i,1} \tag{4.38}$$

so that

$$\mathbf{x}_1'(1)\left(\mathbf{I}-\mathbf{T}_i\right)^{-1}\mathbf{e} = x_0'(1)c_i'\boldsymbol{\alpha}_i\left(\mathbf{I}-\mathbf{T}_i\right)^{-2}\mathbf{e} + \left(\lambda-\lambda_i\right)\boldsymbol{\mu}_{i,1}\left(\mathbf{I}-\mathbf{T}_i\right)^{-2}\mathbf{e} \tag{4.39}$$

From $\boldsymbol{\mu}_i = \boldsymbol{\mu}_i\mathbf{A}_i$ and using $\mu_{i,0} = 1-\lambda_i$ we obtain

$$\boldsymbol{\mu}_{i,1}\left(\mathbf{I}-\mathbf{T}_i\right) = \left(1-\lambda_i\right)c_i'\boldsymbol{\alpha}_i \tag{4.40}$$

hence

$$\mathbf{x}_1'(1)\left(\mathbf{I}-\mathbf{T}_i\right)^{-1}\mathbf{e} = x_0'(1)c_i'\boldsymbol{\alpha}_i\left(\mathbf{I}-\mathbf{T}_i\right)^{-2}\mathbf{e}+\left(1-\lambda_i\right)\left(\lambda-\lambda_i\right)c_i'\boldsymbol{\alpha}_i\left(\mathbf{I}-\mathbf{T}_i\right)^{-3}\mathbf{e} \tag{4.41}$$

where we note that

$$c_i'\boldsymbol{\alpha}_i\left(\mathbf{I}-\mathbf{T}_i\right)^{-2}\mathbf{e} = \frac{\varepsilon_{i,2}+\varepsilon_{i,1}}{2} \tag{4.42}$$

and

$$c_i'\boldsymbol{\alpha}_i\left(\mathbf{I}-\mathbf{T}_i\right)^{-3}\mathbf{e} = \frac{\varepsilon_{i,3}+3\varepsilon_{i,2}+2\varepsilon_{i,1}}{6} \tag{4.43}$$

Evaluating the second derivative of equation (4.28) at $z = 1$, postmultiplying by the column vector $\mathbf{e}$, and simplifying gives

$$\mathbf{X}'(1)\mathbf{A}_i\mathbf{P}_i'(1)\mathbf{e} = \left(1-\lambda+\lambda_i\right)L_q - \frac{M_2-\lambda}{2} \tag{4.44}$$

where L_q is the average queue population, which can be obtained directly from Neuts' equation (4.11). The term $\mathbf{X}'(1)\mathbf{A}_i\mathbf{P}_i'(1)\mathbf{e}$ can also be written as

$$\mathbf{X}'(1)\mathbf{A}_i\mathbf{P}_i'(1)\mathbf{e} = L_q - \mathbf{X}'(1)\begin{bmatrix} c_i \\ \mathbf{T}_i^\circ \end{bmatrix} \tag{4.45}$$

so that, from equation (4.36) we obtain

$$x_0'(1) = \left(\lambda-\lambda_i\right)\left(L_q-\lambda_i\right) + \frac{M_2-\lambda}{2} \tag{4.46}$$

and hence, with some manipulation

$$\mathbf{X}'(1)\mathbf{v}_i'(1) = \frac{1}{6}\left(\lambda-\lambda_i\right)\left(1-\lambda_i\right)^2\left(\varepsilon_{i,3}-\frac{\lambda_i}{1-\lambda_i}\right) - \frac{\lambda_i^2\left(\lambda-\lambda_i\right)}{\left(1-\lambda_i\right)\left(1-\gamma_i\right)^2} \tag{4.47}$$

where we have used the burstiness parameter γ_i to replace the $\varepsilon_{i,1}$ and $\varepsilon_{i,2}$ terms.

Investigation of the second derivative of $\mathbf{u}_i(z)\mathbf{v}_i(z) = 1$ yields

$$\boldsymbol{\mu}_i\mathbf{v}_i''(1) = -2\mathbf{u}_i'(1)\mathbf{v}_i'(1) \tag{4.48}$$

where, from Appendix C we use the relevant expressions to obtain, with some effort

$$\boldsymbol{\mu}_i\mathbf{v}_i''(1) = 2\lambda_i^2\left(1+\lambda_i\right) - \delta_i''(1) + \frac{\lambda_i\delta_i''(1)\left(3+2\lambda_i\right)}{1-\lambda_i} + \frac{3\delta''(1)^2}{2\left(1-\lambda_i\right)^2} - \frac{\delta_i'''(1)}{3\left(1-\lambda_i\right)} \tag{4.49}$$

and eventually that

$$2\mathbf{X}'(1)\mathbf{v}_i'(1)+\mu_i\mathbf{v}_i''(1)=\frac{2\,(1-\lambda)\,\lambda_i^2}{(1-\lambda_i)\,(1-\gamma_i)^2}-\frac{1}{3}\,(1-\lambda)\,(1-\lambda_i)^2\left(\varepsilon_{i,3}-\frac{\lambda_i}{1-\lambda_i}\right) \quad (4.50)$$

which leads to the solution

$$F_i=\frac{\lambda_i^2}{(1-\lambda_i)\,(1-\gamma_i)^2}-\frac{(1-\lambda_i)^2}{6}\left(\varepsilon_{i,3}-\frac{\lambda_i}{1-\lambda_i}\right) \quad (4.51)$$

in equation (4.27). Alternatively, from equation (4.8) we can write F_i as

$$F_i=\frac{\lambda_i^2}{(1-\lambda_i)\,(1-\gamma_i)^2}-\frac{\lambda_i\,(1-\lambda_i)}{6}\left(\frac{\eta_{i,3}}{\eta_{i,1}}-1\right) \quad (4.52)$$

Note that, if the ith source is geom-geom Binary process, then $\varepsilon_{3,i}$ is defined in terms of the two parameters λ_i and γ_i as

$$\varepsilon_{3,i}=\frac{\lambda_i}{1-\lambda_i}+\frac{6\lambda_i}{(1-\lambda_i)^3\,(1-\gamma_i)^2}-\frac{6\lambda_i}{(1-\lambda_i)^2\,(1-\gamma_i)} \quad (4.53)$$

giving

$$2\mathbf{X}'(1)\mathbf{v}_i'(1)+\mu_i\mathbf{v}_i''(1)=-2\,(1-\lambda)\,\frac{\lambda_i\gamma_i}{(1-\gamma_i)^2} \quad (4.54)$$

or

$$F_i=-\frac{\lambda_i\gamma_i}{(1-\gamma_i)^2} \quad (4.55)$$

for this two-state case.

4.3.2 Construction of the Variance Solution

Construction of the actual variance equation, once the F_i terms are known is straightforward, using the results for the derivatives of $\delta(z)$ and $\mathbf{v}(z)$ at $z=1$ that are given in Appendix C. Consequently, this process will be omitted here, since nothing is used that has not either been discussed above, or mentioned in the appendix.

The solution for the variance using phase-geom sources is given by

$$\begin{aligned}\operatorname{Var}[L_q] \;=\; & \frac{4\,(1-\lambda)\,M_3+3M_2^2-6M_2+\lambda^2+2\lambda}{12\,(1-\lambda)^2}\\ & +\frac{M_2-\lambda^2}{(1-\lambda)^2}\sum_{i=1}^{N}\lambda_i\,(1-\lambda_i)\,\frac{\gamma_i}{1-\gamma_i}\\ & -\frac{1}{1-\lambda}\sum_{i=1}^{N}\frac{2\lambda_i^2\,(\lambda-\lambda_i)^2}{(1-\lambda_i)\,(1-\gamma_i)^2}\\ & +\frac{1}{1-\lambda}\sum_{i=1}^{N}\lambda_i\left(2\lambda-2\lambda^2+2\lambda_i^2-\lambda_i\lambda-\lambda_i\right)\frac{\gamma_i}{(1-\gamma_i)^2}\end{aligned}$$

$$\begin{aligned} &+\frac{1}{(1-\lambda)^2}\left(\sum_{i=1}^{N}\lambda_i\,(1-\lambda_i)\,\frac{\gamma_i}{1-\gamma_i}\right)^2 \\ &+\frac{1}{3\,(1-\lambda)}\sum_{i=1}^{N}\lambda_i\,(\lambda-\lambda_i)^2\,(1-\lambda_i)\left(\frac{\eta_{i,3}}{\eta_{i,1}}-1\right) \end{aligned} \tag{4.56}$$

while, for geom-geom sources, the solution reduces to

$$\begin{aligned} \mathrm{Var}\left[L_q\right] &= \frac{4\,(1-\lambda)\,M_3+3M_2^2-6M_2+\lambda^2+2\lambda}{12\,(1-\lambda)^2} \\ &+\frac{M_2-\lambda^2}{(1-\lambda)^2}\sum_{i=1}^{N}\lambda_i\,(1-\lambda_i)\,\frac{\gamma_i}{1-\gamma_i} \\ &+\frac{1}{1-\lambda}\sum_{i=1}^{N}\lambda_i\left(2\lambda-5\lambda_i\lambda+4\lambda_i^2-\lambda_i\right)\frac{\gamma_i}{(1-\gamma_i)^2} \\ &+\frac{1}{(1-\lambda)^2}\left(\sum_{i=1}^{N}\lambda_i\,(1-\lambda_i)\,\frac{\gamma_i}{1-\gamma_i}\right)^2 \end{aligned} \tag{4.57}$$

This equation is algebraically identical to the empirically derived solution presented in [108], although it is slightly simpler in form.

Notice that the first component of both these solutions, is the contribution that can be attributed to the marginal arrivals solution (see Appendix A).

4.3.3 Numeric Confirmation

Although we have presented a solution for the variance based only on analysis of the relevant queueing equations, numerical confirmation of the accuracy of the resulting equations is still desirable, if only to confirm that the algebraic manipulations have been carried out correctly. In [108], Pieloor and Lewis used numeric results to confirm the accuracy of the then empirically derived solution to equation (4.57) for geom-geom Binary sources.

Gordon [33] provides some independently generated variance results (using a numeric analysis approach) and it was hoped to confirm these using equation (4.56). Unfortunately this author was unable to reproduce even the average queue population results in [33], although they should have been given by Neuts equation in [98], and indeed in one case should have been given by the marginal arrivals solution (Appendix A). The reason for this discrepancy is not known, but makes the variance results in [33] useless as a source of confirmation.

Numeric solutions for the variance were obtained in this case using iterative techniques on the matrix geometric form of the queueing equation. Table 4.1 presents the average

m	N	Average	Maximum
2	2	2.2×10^{-13}	5.1×10^{-12}
	3	3.0×10^{-12}	1.7×10^{-10}
	4	1.6×10^{-12}	3.9×10^{-11}
	5	3.0×10^{-12}	8.6×10^{-11}
3	2	2.1×10^{-10}	4.1×10^{-10}
	3	4.4×10^{-08}	4.2×10^{-06}
	4	3.8×10^{-08}	3.7×10^{-06}
4	2	4.3×10^{-11}	1.4×10^{-09}
	3	7.2×10^{-11}	7.9×10^{-10}

Table 4.1: *Relative error between the queue population variance obtained from numeric analysis and the theoretical variance solution. Each row of the table represents results observed from* 100 *randomly generated problems.*

and maximum relative error observed for 100 randomly generated queueing problems, where the number of sources is denoted by N, and the number of states in each source's Markov process is denoted by m. From the tabulated results and the convergence requirements of each case (discussed below), the accuracy of the two variance equations is beyond doubt.

For the geom-geom Binary sources problem ($m = 2$) the parameters of the sources were chosen by assigning each source randomly generated λ_i and γ_i values, since these define the 4 entries of the transition matrix exactly. In order to avoid excessive convergence times or large round-off errors, the total arrival rate (or the server load) was restricted to lie between of 0.4 to 0.8, while each γ_i was restricted to be less than 0.9. The convergence criterion for the iteration process was a relative change in the variance of less than 10^{-10}, with a maximum permitted loss probability of 10^{-13}.

For the cases where $m = 3$ and $m = 4$, the situation is considerably more complicated because large numbers of physical parameters are required to characterise the transition matrix entries (e.g. 6 parameters for the $m = 3$ case). Thus, one method to generate these sources would be to somehow generate these parameters, and then perform some algebraic transform to obtain each of the required entries. The simpler alternative, used in this study, is to randomly assign a λ_i for each source, and then to randomly generate the entries of the transition matrix under the constraint that this desired average arrival rate λ_i be met. It is then straightforward (see Appendix C) to obtain the various parameters of the phase-geom process required to solve equation (4.56).

Due to the extra complexity of these 3 and 4 state problems, and in particular the large number of states required to describe the transition probabilities of the overall arrival

m	N	Mean	Maximum
3	2	12%	120%
	4	5.8%	31%
	8	3.7%	11%
	16	2.8%	8.5%
4	2	5.4%	39%
	4	3.0%	9.6%
	8	2.1%	5.5%
	16	1.7%	3.6%

Table 4.2: *Relative error between the exact queue population variance obtained from equation (4.56) and the geom-geom approximation obtained from equation (4.57). Each row of the table represents results observed from* 1000 *randomly generated problems.*

process (given by m^N) the convergence criteria were reduced to a relative change in the variance of 10^{-8} with a maximum loss probability of 10^{-11}. As for the $m = 2$ case, the overall arrival rate was restricted to the range of 0.4 to 0.8.

4.3.4 Approximating Phase-Geometric Processes by Geometric-Geometric Processes

Obviously there is a considerable advantage from a computational point of view to modelling a many-state phase-geom Binary arrival process by a two-state geom-geom process. For the average queue population, this is easily achieved by matching the autocorrelation parameter γ_i for each source. The effect on the variance however is harder to describe. In order to provide some quantitative feedback on using this approximation, the variance results of the randomly generated 3 and 4 state queueing problems (as done above) were compared with the equivalent geom-geom variance result. Table 4.2 presents the results of this study, with the actual and approximate variance calculated in each case from equations (4.56) and (4.57) respectively for the randomly generated transition matrices. The variable m indicates the number of states in the transition matrix.

From the tabulated results it would appear that as the number of sources increases, or as the number of states in the transition matrix increases, the accuracy of the approximation improves. The magnitude of the error for the smaller transition matrix size, and for small number of sources is a concern however. Since a maximum error is a poor indicator, Figure 4.1 presents a plot of the approximated variance against the actual variance for the problem above having $m = 3$ and $N = 2$. In order to keep the plot dimensions to a convenient scale, only those problems with a variance

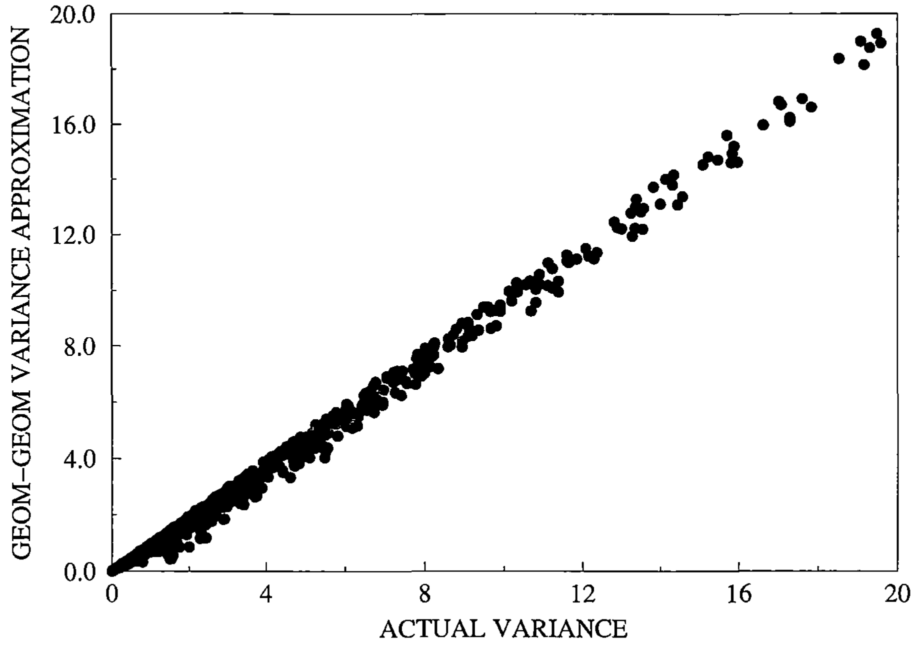

Figure 4.1: *Geom-geom variance approximation as a function of the actual variance for 946 samples of a G/D/1 queue fed by two 3-state sources with randomly generated parameters.*

below 20 were included (which was nearly 95% of the results anyway). The cause of the large error magnitudes is immediately obvious from the plot. The actual variance forms an upper bound to the approximate results, which sit in a small band below this. The width of this band seems fairly constant for most of the plot, and so when the magnitude of the actual variance is small, the geom-geom variance error tends to become quite high.

The observed trends in Table 4.2 (that the error reduces with the number of states and number of sources) are explained then by the fact that, since sources with larger numbers of states, or queues fed by larger numbers of sources, usually experience larger queue population variances, the observed error decreases. So, in practice, if the geom-geom modelled variance is high, then the accuracy of the approximation will be good, and can be confidently used. Conversely, if the approximate variance is small, it will be difficult to estimate how accurate the approximation is.

4.3.5 Identical Sources

For N identical sources, we write $\lambda_i = \lambda/N$, $\gamma_i = \gamma$, $\eta_{i,1} = \eta_1$, and $\eta_{i,3} = \eta_3$, for which $\mathrm{Var}[L_q]$ reduces to

$$\begin{aligned}\mathrm{Var}[L_q] \;=\; & \frac{4\lambda + \lambda^2 - 8\lambda^3 + \frac{3\lambda^2}{N} - \frac{8\lambda^2}{N^2}(1-\lambda)}{12(1-\lambda)^2} \\ & + \frac{\lambda^2\gamma\left(3 - 4\lambda + 2\lambda^2 + \frac{1}{N}(1 - 4\lambda + \lambda^2) + \frac{\lambda^2}{N^2}\right)}{(1-\lambda)^2(1-\gamma)^2} \\ & - \frac{2\lambda^4\left(1 - \frac{1}{N}\right)^2}{(1-\lambda)(N-\lambda)(1-\gamma)^2} \\ & + \frac{\lambda^3\left(1-\frac{1}{N}\right)^2\left(1-\frac{\lambda}{N}\right)}{3(1-\lambda)}\left(\frac{\eta_3}{\eta_1} - 1\right) \end{aligned} \qquad (4.58)$$

or as $N \to \infty$

$$\mathrm{Var}[L_q] = \frac{4\lambda + \lambda^2 - 8\lambda^3}{12(1-\lambda)^2} + \frac{\lambda^2\gamma(3 - 4\lambda + 2\lambda^2)}{(1-\lambda)^2(1-\gamma)^2} + \frac{\lambda^3}{3(1-\lambda)}\left(\frac{\eta_3}{\eta_1} - 1\right) \qquad (4.59)$$

Similarly, for N identical geom-geom sources we obtain

$$\begin{aligned}\mathrm{Var}[L_q] \;=\; & \frac{4\lambda + \lambda^2 - 8\lambda^3 + \frac{3\lambda^2}{N} - \frac{8\lambda^2}{N^2}(1-\lambda)}{12(1-\lambda)^2} \\ & + \frac{\lambda^2\gamma\left(3 - 2\lambda - \frac{1}{N}(1 + 6\lambda - 5\lambda^2) + \frac{\lambda}{N^2}(4 - 3\lambda)\right)}{(1-\lambda)^2(1-\gamma)^2} \end{aligned} \qquad (4.60)$$

so that for $N \to \infty$, we obtain

$$\mathrm{Var}[L_q] = \frac{4\lambda + \lambda^2 - 8\lambda^3}{12(1-\lambda)^2} + \frac{\lambda^2\gamma(3 - 2\lambda)}{(1-\lambda)^2(1-\gamma)^2} \qquad (4.61)$$

4.4 Summary

This chapter has considered a special case of the phase-geom IBP, where each source generates an arrival with probability 1 in every time slot that the source is active. This leads to a completely known 'empty queue' probability vector **b**, and hence to the possibility of closed form expressions for the queue behaviour. In [98], Neuts presented a solution for the average queue population of an infinite buffer, discrete time G/D/1 queue fed by these sources, building on the earlier work of Viterbi [133].

We have presented here a closed form expression for the variance of the average queue population of this queueing problem — a result that has not previously been obtained. The importance of this result is that both the average and the variance of the queue population are required to estimate bounds on the queueing delays and to approximate the loss probabilities of the finite buffer case (see Chapter 6).

The accuracy of approximating a phase-geom Binary source by a simpler geom-geom model was considered for a few cases, and in addition we have shown how the autocorrelation function is directly related to the choice of definition for the autocorrelation parameter originally suggested by Neuts.

Chapter 5

Population Analysis of Cyclic Service Queues

In Chapters 3 and 4 we dealt with the analysis of the population of a uninterrupted service queueing system subject to arrivals from numbers of IBP sources. However, we considered in the introduction to this thesis that there is a need for dual buffer, priority based queueing systems in order to meet the quality of service requirements of both loss and delay sensitive traffics. In this chapter we will consider the analysis of the lower priority queue in a dual buffer system, when the high priority queue is subject to arrivals from a cyclic or periodic process, and hence the lower priority queue receives service in a complementary cyclic fashion. Rather than analysing this interrupted service queue directly, we use Corollary 2.3 and analyse the population of a queueing system subject to arrivals from both a cyclic source and a number of IBP sources. As in Chapter 3, we restrict ourselves to geom-geom IBP sources.

Figure 5.1 illustrates the basic transition process that describes the behaviour of a binary cyclic arrival process. Each state (numbered from zero in diagram) in the process is either active or silent. During its silent states, the process generates no arrivals, while in its active states, a single arrival is generated in every time slot (hence the binary appellation). We assume that at the beginning of each time slot, the process changes to the next state, wrapping around between the last and the first states. The pattern of arrivals from this source is therefore exactly periodic.

Cyclic service (or cyclic arrival) queueing problems are perhaps better known in the literature as hybrid or integrated switching systems [8, 67, 70, 74, 80–82, 114, 123, 137]. These systems provide a number of circuit and packet switched connections using a frame based STM transport mechanism. The circuit switched connections in the frame

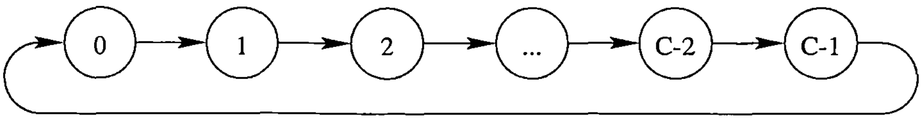

Figure 5.1: *Illustration of the transıtıon probabilities of the Markov chain for a cyclic source. Since only one transition ıs possible from each state, the transition probabilities are all equal to one.*

can be thought of as a cyclic arrival or service process with a period equal to the frame length in cells.

The majority of these hybrid studies are based on the observation of the queueing system at either the end or the beginning of a frame time. That is, the behaviour of the queueing system within the frame is ignored. This unfortunately will lead to underestimation of the actual queue performance [106], particularly as the frame size increases. The analyses in [74, 80–82, 123] are all similar in content, considering the queueing delay of the Poisson distributed 'data' traffic as the number of active states in the cyclic process vary (on-off variation in the 'voice' traffic that describes the framing). Wieselthier and Ephremides [137] present a slightly different slant by considering several different ways in which contention for the available slots in each frame might be resolved. In [8], Arthurs and Stuck present a more in-depth analysis, allowing the 'voice' and 'data' traffics (the number of active states in a cycle and the arrivals from the other non-cyclic sources) to be described by a general Markov chain. Since this model is not tractable, they relax the restrictions to look at several simpler cases.

An alternative approach, used in [67,114] and [41] is to establish the capacity remaining within a cycle, and then analyse the queueing performance using this capacity measure. This approach leads to further underestimation of the actual performance however, since the deterministic nature of the service process is ignored in addition to the framing effects. In [70], Kaudel and Beshai attempt to capture the effect of the framing on the queueing performance for Poisson traffic. Their analysis is based on the assumption that the arrivals from the cyclic process are evenly spaced within the cycle, and that this pattern is then well described by a Poisson process. Unfortunately they do not provide simulation results (or similar) to indicate the accuracy, or otherwise of their approximations.

Of the available literature, the only accurate analysis of the queueing performance of a framed system at the cell level was given by Pieloor in [106]. In this paper, the author considered a cyclic arrival process mixed with Poisson arrivals, using a numeric iterative approach to obtain the complete population and delay distributions, and the finite buffer average loss probabilities. Despite being able to extend the results to geom-

geom IBP sources (see Appendix E) the computational requirements of the numeric iterative method are too high for all but the simplest cases, and alternative approaches are required.

In [79] Li discusses the probability generating function analysis of a queueing system with arrivals from on-off sources that generate a single arrival periodically in their active states. Treating this as a framed system, Li derives expressions for the locations of the poles of the relevant queueing equation and also for the average queue population of the system. Despite the framed nature of Li's approach, it shares many features with the method to be presented here.

Sections 5.1 and 5.2 present the exact solutions for the average and variance of the queue population, and in particular look at the problems in solving this solution due to the finite numeric precision of digital computer implementation. In order to overcome these numerical difficulties, we develop an adaptive solution method in sections 5.2.2 and 5.2.3, and investigate its accuracy.

In section 5.3 we then look at several approximate solution methods. In 5.3.1 we approximate the cyclic source by a random process, and in 5.3.2 we investigate the accuracy of worst case result for the adaptive solution method. Then in section 5.3.3 we investigate the use of the kth order approximation method first discussed in chapter 3. Unsurprisingly, given the results of that chapter, this approximation has the highest accuracy of the three methods investigated.

5.1 Exact Queue Population Analysis

In this section we will apply the population theory of Chapter 2 to the analysis of an infinite buffer queueing system subject to arrivals from the superposition of a single cyclic binary source and N geom-geom IBP sources. Since we have investigated geom-geom IBP sources in detail in Chapter 3, we will start by defining a cyclic binary source.

5.1.1 Characterising the Cyclic Source

The cyclic process illustrated in Figure 5.1 can be characterised both by the period in slot times, denoted by C, and a set of C parameters describing whether the process is active or inactive in each of the C states. Although this is the general form of a cyclic process, we incorporate a further restriction in order to simplify the use of the cyclic model. We assume that all the active states of the cyclic process are consecutive within the overall period. That is, the cyclic process generates some $b < C$ arrivals, one at a time over consecutive time slots, followed by $C - b$ time slots in which no arrivals are generated. Figure 5.3 illustrates the traffic stream from this model, compared to an example of the more general arrival process (Figure 5.2).

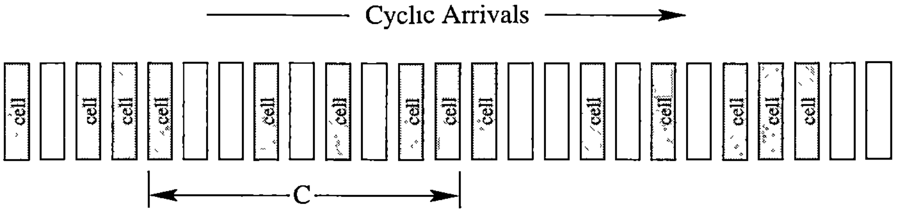

Figure 5.2: *An example general cell arrival pattern for a cyclic source with period C.*

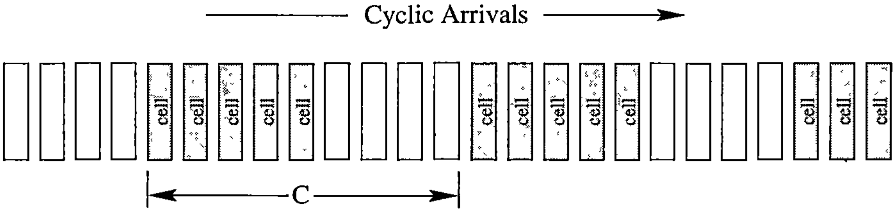

Figure 5.3: *Arrivals constrained to occur consecutively within the cyclic period C.*

There are two basic justifications for making this simplification to the general cyclic binary arrival model. The first is that, at each merging point in the network, the effect of queueing in switch output buffers will cause simultaneous arrivals to be output in random but consecutive order. This effect will be offset somewhat however by splitting at the next switch stage.

The second justification is related, and is based on the fact that we are particularly concerned with the analysis of dual buffer priority systems where one buffer has non pre-emptive service priority over the other. In this arrangement, the queueing of the cyclic process in the high priority queue will cause (in the same manner as before) the high priority buffer to require service for consecutive time slots. That is, the interruption process of the low priority buffer will tend to involve consecutive services followed by consecutive non-services. As indicated by Corollary 2.3, this interrupted service problem can be described by a cyclic binary arrival process of the type proposed above.

A third reason for making this assumption about the behaviour of the cyclic arrival process is that for a given C and number of active states b, it results in the largest queue populations and delays [106]. That is, it forms an upper bound on the queueing performance of the system.

With this simplification, the number of parameters required to describe the cyclic process reduces to just two — b and C. The $C \times C$ state to state transition matrix is then described by

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{5.1}$$

where the subscript 0 is used to indicate the cyclic source. As in Chapter 3, subscripts from 1 to N indicate the geom-geom IBP sources. The stationary probability vector for $\mathbf{A}_0$ is of course given by

$$\boldsymbol{\mu}_0 = \left[\frac{1}{C}, \frac{1}{C}, \frac{1}{C}, \cdots, \frac{1}{C}, \frac{1}{C}\right] \tag{5.2}$$

The $C \times C$ probability generating function matrix $\mathbf{P}_0(z)$ for the cyclic binary arrival process is given by

$$\mathbf{P}_0(z) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & z & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & z & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & z \end{bmatrix} \tag{5.3}$$

where we are assuming, without loss of generality, that a cycle is made up of $C - b$ silent time slots (represented by 1's in the $\mathbf{P}_0(z)$ matrix), followed by b active slots in which a single arrival is generated per time slot (represented by the z terms in the $\mathbf{P}_0(z)$ matrix). The average arrival rate from this cyclic binary source is denoted by λ_0 and has the value b/C.

Note that the autocorrelation coefficient function of a cyclic arrival process is also cyclic, and it is fairly simple to show that

$$R(m) = \begin{cases} \frac{b-m}{b(1-\frac{b}{C})} - \frac{b}{C-b} & \text{for } 0 \leq m + kC < b \\ -\frac{b}{C-b} & \text{for } b \leq m + kC < C - b \\ 1 - \frac{C-m}{b(1-\frac{b}{C})} & \text{for } C - b \leq m + kC < C \end{cases} \tag{5.4}$$

for some integer k such that $0 \leq m + kC < C$. In the case for IBP sources, we were concerned with the quantity known as the single-sided autocorrelation (or autocovariance) sum, which is the sum from $m = 1$ to ∞ of $R(m)$. From relation (5.4) above, it is straightforward to show that the sum across any number of cycles of $R(m)$ is 0. However, the cyclic nature of the autocorrelation function means that the infinite sum does not converge, and hence the single-sided sum of $R(m)$ does not properly exist.

5.1.2 Applying the Queue Population Theory

In Appendix C we develop the relevant eigensystem analysis for cyclic arrival processes. In particular we obtain

$$\delta_0'(1) = \lambda_0 \tag{5.5}$$

$$\delta_0''(1) = \lambda_0 (\lambda_0 - 1) \tag{5.6}$$

$$\delta_0'''(1) = \lambda_0 (\lambda_0 - 1)(\lambda_0 - 2) \tag{5.7}$$

where $\delta_0(z)$ is the Perron–Frobenius eigenvalue of the transition probability generating matrix $\mathbf{A}_0\mathbf{P}_0(z)$. In addition we have

$$\mathbf{h}_{0,n}(1)\mathbf{g}_{0,0}'(1) = \frac{e^{2\varphi_n}\left(1 - e^{b\varphi_n}\right)}{C\left(1 - e^{\varphi_n}\right)^2} \tag{5.8}$$

where

$$\varphi_n = \frac{2\pi n}{C}\sqrt{-1} \tag{5.9}$$

and where $\mathbf{h}_{0,n}(z)$ and $\mathbf{g}_{0,n}(1)$ are the general left and right eigenvectors corresponding to the nth $(n = 0, 1, \ldots, C-1)$ general eigenvalue of $\mathbf{A}_0\mathbf{P}_0(z)$.

Incorporating the cyclic source into equation (3.12) for the average population of a queue subject to arrivals from geom-geom IBP sources only gives

$$L_q = \frac{\mathbf{bv}'(1)}{1-\lambda} + \frac{M_2 - \lambda - \lambda_0(1-\lambda_0)}{2(1-\lambda)} + \frac{1}{1-\lambda}\sum_{\imath=1}^{N} \lambda_\imath (\theta_\imath - \lambda_i) \frac{\gamma_\imath}{1-\gamma_i} \tag{5.10}$$

where $\mathbf{b}$ is the empty system probability vector, and $\mathbf{v}(z)$ is the right hand Perron–Frobenius eigenvector of the combined arrival process, and the λ and M_2 terms are the average and second moment of the combined arrival process (including the cyclic source). Similarly we obtain the queue population variance from equation (3.13) as

$$\begin{aligned}\mathrm{Var}[L_q] \quad = \quad & \frac{\mathbf{bv}''(1)}{1-\lambda} + 2\sum_{\imath=1}^{N}\frac{\lambda_i\gamma_\imath}{(1-\gamma_\imath)^2}\mathbf{bg}_{(\imath,1)}(1) - 2\sum_{n=1}^{C-1}\frac{e^{2\varphi_n}\left(1-e^{b\varphi_n}\right)}{C\left(1-e^{\varphi_n}\right)^3}\mathbf{bg}_{(0,n)}(1) \\ & - \frac{\lambda_0(1-\lambda_0)(3\lambda-2-2\lambda_0)}{3(1-\lambda)} + \frac{M_3 - 3(M_2-\lambda)-\lambda}{3(1-\lambda)} - L_q^2\end{aligned}$$

$$+\frac{1}{1-\lambda}\left(M_2-2\lambda+1-\lambda_0+\lambda_0^2+2\sum_{i=1}^{N}\lambda_i\left(\theta_i-\lambda_i\right)\frac{\gamma_i}{1-\gamma_i}\right)L_q$$
$$+\frac{2}{1-\lambda}\sum_{i=1}^{N}\lambda_i\left(\theta_i-\lambda_i\right)\frac{\left(\lambda-\lambda\gamma_i-2\lambda_i+\theta_i\gamma_i\right)\gamma_i}{\left(1-\gamma_i\right)^2} \tag{5.11}$$

The relevant derivatives of the overall Perron–Frobenius eigenvector $\mathbf{v}(z)$ are given in terms of the derivatives of the individual source eigenvectors as

$$\mathbf{v}'(1)=\mathbf{v}_0'(1)\otimes\mathbf{e}_{2^N}+\sum_{i=1}^{N}\left(\mathbf{e}_{2^{(i-1)}C}\otimes\mathbf{v}_i'(1)\otimes\mathbf{e}_{2^{(N-i)}}\right) \tag{5.12}$$

and

$$\begin{aligned}\mathbf{v}''(1) \quad = \quad & \mathbf{v}_0''(1)\otimes\mathbf{e}_{2^N}+\sum_{i=1}^{N}\left(\mathbf{e}_{2^{(i-1)}C}\otimes\mathbf{v}_i''(1)\otimes\mathbf{e}_{2^{(N-i)}}\right)\\ & +2\sum_{i=1}^{N-1}\left(\mathbf{e}_{2^{(i-1)}C}\otimes\mathbf{v}_i'(1)\otimes\sum_{j=1}^{N-i}\left(\mathbf{e}_{2^{(j-1)}}\otimes\mathbf{v}_{i+j}'(1)\otimes\mathbf{e}_{2^{(N-i-j)}}\right)\right)\\ & +2\mathbf{v}_0'(1)\otimes\sum_{i=1}^{N}\left(\mathbf{e}_{2^{(i-1)}}\otimes\mathbf{v}_i'(1)\otimes\mathbf{e}_{2^{(N-i)}}\right)\end{aligned} \tag{5.13}$$

where $\mathbf{e}_x$ is a column vector of x elements, all of which have the value 1. For the cyclic source we write

$$\mathbf{v}_0'(1)=\left[v_0',v_1',\ldots,v_{C-1}'\right]^T \quad \text{and} \quad \mathbf{v}_0''(1)=\left[v_0'',v_1'',\ldots,v_{C-1}''\right]^T \tag{5.14}$$

where, from Appendix C

$$v_j'=\begin{cases}\lambda_0\left(1-\frac{C-b}{2}\right) & \text{for } j=0\\ v_{j-1}'+\lambda_0 & \text{for } 1\le j<C-b\\ v_{j-1}'+\lambda_0-1 & \text{for } C-b\le j<C\end{cases} \tag{5.15}$$

and

$$v_j''=\begin{cases}\frac{\lambda_0}{6C}\left(C-b\right)\left(3C-6b+2bC-5-2b^2\right) & \text{for } j=0\\ v_{j-1}''+\lambda_0\left(2v_{j-1}'-1+\lambda_0\right) & \text{for } 1\le j<C-b\\ v_{j-1}''+\left(1-\lambda_0\right)\left(2-\lambda_0-2v_{j-1}'\right) & \text{for } C-b\le j<C\end{cases} \tag{5.16}$$

The Perron–Frobenius derivatives for the individual IBP sources are given by equations (3.16) and (3.17).

The vectors $\mathbf{g}_{(0,n)}(1)$ and $\mathbf{g}_{(i,1)}(1)$ are given by

$$\mathbf{g}_{(0,n)}(1)=\mathbf{g}_{0,n}(1)\otimes\mathbf{e}_{2^N} \tag{5.17}$$

and for $i\ge 1$

$$\mathbf{g}_{(i,1)}(1)=\mathbf{e}_{2^{(i-1)}C}\otimes\mathbf{g}_{i,1}(1)\otimes\mathbf{e}_{2^{(N-i)}} \tag{5.18}$$

respectively where

$$\mathrm{g}_{0,n}(1) = \left[1, e^{\varphi_n}, e^{2\varphi_n}, \ldots, e^{(C-1)\varphi_n}\right]^T \tag{5.19}$$

and where $\mathrm{g}_{\imath,1}(1)$ is given by equation (3.19).

The only remaining term required in order to calculate the queue population average and variance is the empty system probability vector **b**. As we shall discuss in the following section, obtaining this vector is not, in many cases, a simple task.

5.2 Obtaining the Empty System Probability Vector

As discussed in Chapters 2 and 3, the **b** vector is obtained by first finding the poles of the queue equation that lie within the unit circle, and then solving a linear system of equations constructed from the general right hand eigenvector of the combined arrival process evaluated at each of the pole positions. Unfortunately, significant numerical difficulties are encountered in attempting to perform these operations.

In the following we will consider the pole finding and linear system solving stages separately, before looking at some results for the accuracy of this method. In order to assess the accuracy of the queue population results, numeric iterative methods are used to generate results with relative convergence errors of less than 10^{-9} and loss probabilities of similar order. Although this method is both highly accurate and robust, it is unfortunately very slow, which is why we prefer to have other means available for analysing the queueing performance. The iterative solution method is discussed in more detail in Appendix E.

5.2.1 Finding the Poles and Zeros of the Queue Equation

As before, we denote the $\jmath$th pole of the queue equation by $z_\jmath^*$, noting that this quantity satisfies

$$z_\jmath^* - \omega_\jmath(z_\jmath^*) = 0 \tag{5.20}$$

where $\omega_\jmath(z)$ is the $\jmath$th general eigenvalue of the transition probability generating matrix of the combined arrival process, given by

$$\omega_\jmath(z) = \prod_{\imath=0}^{N} \omega_{\imath,r_{\imath,\jmath}}(z) \tag{5.21}$$

and where $\omega_{\imath,n}$ is the nth eigenvalue of the $\imath$th source, and $r_{\imath,\jmath}$ is a function that describes which eigenvalue of source $\imath$ is indicated when the overall eigenvalue is $\jmath$. For the N geom-geom IBP sources, $\omega_{i,n}(z)$ is given by equation (3.22) while for the cyclic source we have

$$\omega_{0,n}(z) = e^{\varphi_n} z^{\frac{b}{C}} \tag{5.22}$$

where φ_n is as described in equation (5.9). A convenient definition for the cyclic source problem, and one which is similar to the proposed form of $r_{i,j}$ in chapter 3 is

$$r_{i,j} = \begin{cases} j \bmod C & \text{for } i = 0 \\ 2^{1-i} \left\lceil \frac{j}{C} \right\rceil \bmod 2 & \text{otherwise} \end{cases} \tag{5.23}$$

where $\lceil x \rceil$ represents the largest integer not greater than x, and $j = 0, 1, \ldots, \left(2^N C - 1\right)$.

The are $2^N C$ different eigenvalues that can be formed for the combined cyclic and geom-geom IBP arrivals queueing system. We note however that within every cycle of C periods, there are b time slots in which there is a guaranteed arrival from the cyclic source. This means that there will only be $2^N (C - b)$ unknown probabilities in the empty system vector **b** — the remaining $2^N b$ probabilities will be zero. Thus we only need to find the solution to $2^N (C - b)$ pole equations.

Now, since $|\omega_j(z)| < 1$ provided $|z| < 1$ there must be at least one solution to equation (5.20) according to the Fixed Point Theorem [47]. Obviously $z_j^* = 0$ is one solution to equation (5.20) for all j, but this solution provides no useful information for our purposes. This solution will only be guaranteed unique however if $\omega_j(z)$ is a contraction mapping of z — that is if there exists some constant $0 \leq r < 1$ such that

$$|\omega_j(z_1) - \omega_j(z_2)| < r \, |z_1 - z_2| \tag{5.24}$$

for all z_1 and z_2 within the unit circle. Since $\omega_j(0) = 0$, and since it is always possible to find some z such that $|\omega_j(z)| > |z|$ (consider when $z \to 0$ for example) we know that no suitable value of r exists, and hence $z_j^* = 0$ is not *necessarily* a unique solution to equation (5.20). We are interested then in finding firstly for which j there is a non zero solution to equation (5.20), and secondly the value of that solution.

For convenience we describe the C eigenvalues of the cyclic source corresponding to a single combined eigenvalue of the N geom-geom IBP sources as belonging to a *cycle.* Thus there are 2^N cycles covering every eigenvalue of the queueing system. Within each cycle there will be one positive real solution to equation (5.20) corresponding to $\varphi_n = 0$ (using the same argument as for the IBP sources only in Chapter 3). Due to the complex valued contribution when $\varphi_n \neq 0$ all other solutions will be complex and occur in conjugate pairs — with one possible exception. If $C - b$ is even, then with one positive real pole, and all others appearing as conjugate pairs, there must be a second real valued pole. If present, this pole will occur on the negative real axis, where the imaginary contribution of the complex valued e^{φ_n} term can be cancelled by

Relationship of j to b and C	Location of pole z_j^*
$j \bmod C = 0$	positive real axis
$j \bmod C = \left\lceil \frac{C-b}{2} \right\rceil$ and $C-b$ is even	negative real axis
$\left\lceil \frac{C-b}{2} \right\rceil + 1 \leq j \bmod C \leq \left\lceil \frac{C-b}{2} \right\rceil + b$	z_j^* is zero
otherwise	z_j^* is complex

Table 5.1: *Summary of the relationship between the location of the pole z_j^* and its index j.*

contributions from the complex valued $z^{\frac{b}{C}}$ term of the cyclic process' eigenvalue, and the complex valued eigenvalues of the IBP sources.

As an example, Figures 5.4 and 5.5 show the positions in the complex number plane of e^{φ_n} for $C = 16$, with $b = 5$ and $b = 6$ respectively. In particular we have marked on these plots where the overall eigenvalues associated with each point have non-zero real or complex valued fixed points. These conclusions are based purely on empirical observations, but have consistently proved to be accurate for the many thousands of (randomly generated) queueing problems investigated. A discussion in [79] suggests that these relations can probably be proved, although we have not done so here. Table 5.1 summarises the relationship between the pole location and its index when φ_n is defined by equation (5.9) for general b and C. In addition, we note that if z_j^* is a complex valued pole, then pole z_k^* is its conjugate, where

$$k = j + C - 2(j \bmod C) \qquad (5.25)$$

Thus, for each cycle we need only establish the position of $\left\lceil \frac{C-b}{2} \right\rceil + 1$ poles in order to know all of the poles.

As with the purely real valued problem of Chapter 3, we attempt to find the $2^N\ (C-b)$ required solutions using the Newton–Raphson search algorithm in two dimensions [47, 109]. Identification of those poles having purely real values can aid solution finding by restricting the search to the real axis only. Once the pole is found to within some small error value, the solution can be improved or 'polished' by making use of the iterative process described by

$$z_{n+1} = \omega_j(z_n) \qquad (5.26)$$

which will tend to the solution z_j^* as $n \to \infty$ [47,79]. This iterative process is unfortunately not certain to converge to the correct z_j^* from every starting point z_0, which is why the Newton–Raphson method is used to obtain a close initial guess. This iterative improvement of the Newton–Raphson method is not always necessary, but can help to significantly reduce the solution error when the initial point accuracy is not particularly high.

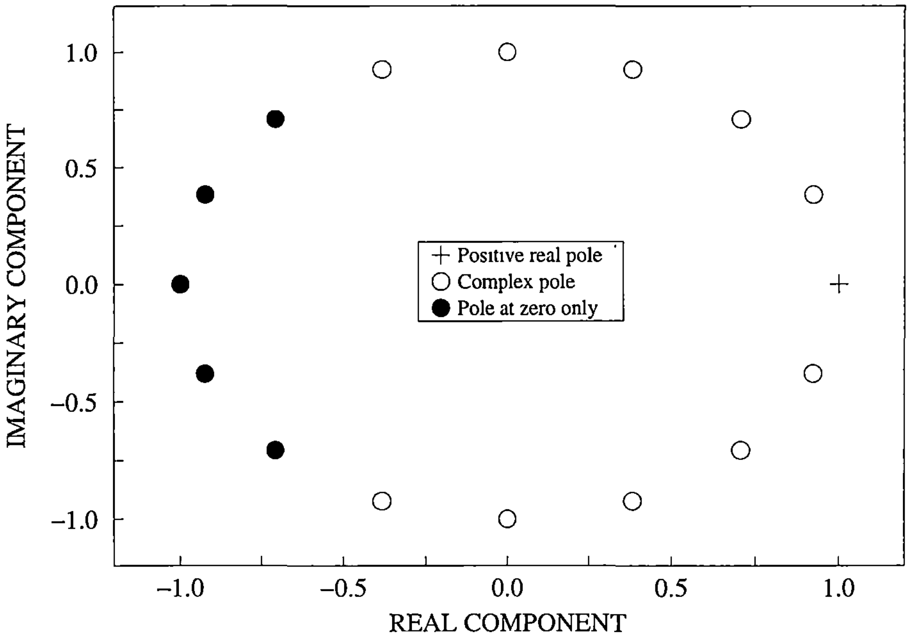

Figure 5.4: *Relationship between e^{φ_n} and the type of pole the corresponding overall eigenvalue will have. These results are for $C = 16$ and $b = 5$ ($C - b$ odd).*

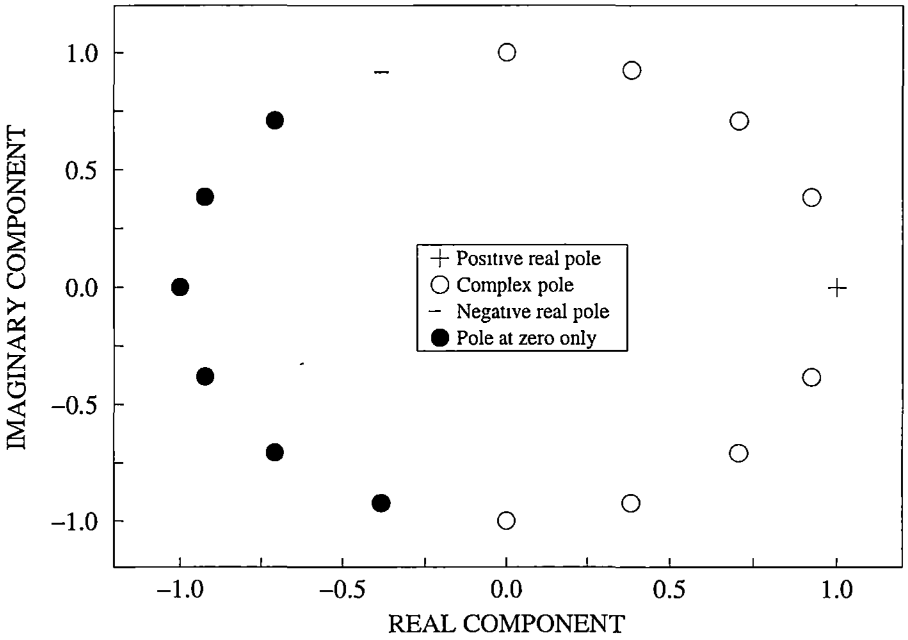

Figure 5.5: *Relationship between e^{φ_n} and the type of pole the corresponding overall eigenvalue will have. These results are for $C = 16$ and $b = 6$ ($C - b$ even).*

Convergence with the Newton–Raphson method is not guaranteed either for every starting point, and it is important that the search algorithm is able to recognise when the process has failed, and take appropriate action. This search for the pole locations fails in several ways, all of which are fairly easy to identify. The first is that the search point begins to move outside the unit circle — there are solutions to equation (5.20) that lie outside the unit circle (termed non-vanishing roots by Li and Sheng in [83]) and these can sometimes be found by the search algorithm instead of the desired solutions. The second is peculiar to searches on the real axis, and occurs when a positive (but very small) solution is found to what should be a negative (but very small) pole, and vice versa. These solutions appear to be due to the finite numeric precision of the computer implementation of the problem. The third type of failure, which has only been observed for the complex valued poles, occurs when the search remains within the unit circle, but does not converge to a solution for equation (5.20), or alternatively converges to $z = 0$. The reason for this is not known.

A suitable approach to providing initial guesses for the search algorithm is based on the observation [79] that the $C - b$ non-zero poles in a single cycle have arguments that are approximately evenly distributed around the unit circle. This is illustrated quite clearly in Figure 5.6 which shows these pole locations for an example queueing system having 4 IBP sources and a cyclic source with $C = 16$ and $b = 4$. In this case there are 12 non-zero poles in every cycle, of which two are on the real axis. Thus we might start the search for the pole z_j^* at some point given by $Ae^{\varrho_j \sqrt{-1}}$ where

$$\varrho_j = \begin{cases} \frac{2\pi}{C-b}\,(j \bmod C) & \text{if } 0 \leq (j \bmod C) \leq \left\lceil \frac{C-b}{2} \right\rceil \\ \frac{2\pi}{C-b}\,(j \bmod C - C) & \text{if } \left\lceil \frac{C-b}{2} \right\rceil + b < (j \bmod C) < C \end{cases} \tag{5.27}$$

and where the magnitude factor A would be say 1 for the first search attempt. Whenever the search fails, A can be reduced (say reduced by half) and the search started again with the new start point.

Occasionally, no solution to equation (5.20) is found, even after many restarted searches, and the only option is to use the closest solution in place of an exact one. By closest solution we mean that z_j^* which minimises $z - \omega_j(z)$. It is a fairly simple matter to have the implementation of the search algorithm keep track of the best solution found at each step, and to return to this if no other pole is found. We will refer to problems of this kind as *pole placement errors*. Fortunately these types of errors do not appear to be very common (see section 5.2.3 below). The exact cause of the errors is not known, and they appear to be unaffected by increasing the numeric resolution, although we have not studied this problem in great detail.

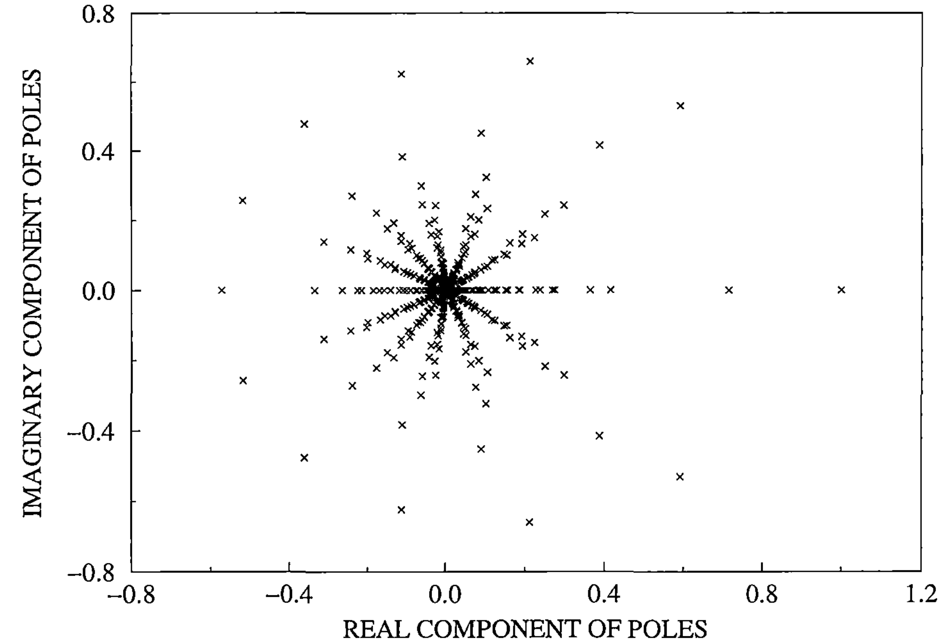

Figure 5.6: *Illustration of the tendency for the poles of the queue equation to be evenly distributed in anglular terms around the unit circle.*

5.2.2 Solving for the Linear System

The state space of the system of the simultaneous equations describing the empty system vector involves $2^N (C - b)$ unknowns, corresponding to the non-zero probabilities in **b**. As in Chapter 3 the coefficients of the linear system are determined from the poles of the queueing equation and the relevant eigenvectors, and the system

$$\mathbf{b}^{\circ}\mathbf{M} = [(1-\lambda), 0, 0, \ldots, 0] \tag{5.28}$$

is solved (using LU decomposition) for $\mathbf{b}^{\circ}$, which is a vector consisting only of the non-zero elements of **b**, referred to as the reduced state empty system vector. The matrix **M** describes the coefficients for the simultaneous equations, where the columns of **M** are made up from those elements of the right hand eigenvector $\mathbf{g}_j(z_j^*)$ corresponding to the non-zero **b** vector probabilities, and only for those j having non-zero z_j^*. The complete empty system vector is then obtained directly from the elements of $\mathbf{b}^{\circ}$.

Eigenvector $\mathbf{g}_j(z)$ is given by

$$\mathbf{g}_j(z) = \bigotimes_{i=0}^{N} \mathbf{g}_{i,r_{i,j}}(z) \tag{5.29}$$

where $r_{i,j}$ is as defined previously in equation (5.23), and where $\mathbf{g}_{i,n}(z)$ describes the general right hand eigenvector corresponding to the nth eigenvector of source i. For the geom-geom IBP sources, eigenvector $\mathbf{g}_{i,n}(z)$ $(i = 1, 2, \ldots, N)$ is given by equation (3.26), while the eigenvector for the cyclic source is given by

$$\mathbf{g}_{0,n}(z) = \left[1, \omega_{0,n}(z), \ldots, z^{-1}\omega_{0,n}(z)^{C-b}, z^{-2}\omega_{0,n}(z)^{C-b+1}, \ldots, z^{-b}\omega_{0,n}(z)^{C-1}\right]^T \tag{5.30}$$

Alternatively, since the non-zero states of $\mathbf{b}$ correspond to the silent periods of the cyclic source, we can use

$$\mathbf{g}^{\circ}_j(z) = \mathbf{g}^{\circ}_{0,r_{0,j}}(z) \otimes \left(\bigotimes_{i=1}^{N} \mathbf{g}_{i,r_{i,j}}(z)\right) \tag{5.31}$$

where

$$\mathbf{g}^{\circ}_{0,n}(z) = \left[1, \omega_{0,n}(z), \ldots, \omega_{0,n}(z)^{C-b-1}\right]^T \tag{5.32}$$

The columns of matrix $\mathbf{M}$ will then be made up of those $\mathbf{g}^{\circ}_j(z^*_j)$ for which z^*_j is non-zero.

One important consideration in the solution of this system is that the $\mathbf{g}_j(z^*_j)$ are generally vectors of complex numbers. Rather than trying to perform complex valued matrix decompositions, we note that the elements of two eigenvectors corresponding to complex conjugate eigenvalues are themselves complex conjugates [36]. Thus, we can construct the linear system from purely real coefficients by taking the real components of $\mathbf{g}_j(z^*_j)$ when the imaginary component of z^*_j is zero or positive, and the imaginary components of $\mathbf{g}_j(z^*_j)$ when it is negative. The reverse arrangement can be used equally well.

An Ill-Conditioned Problem

As for the case where there are only geom-geom IBP sources present, the normal indicators of how singular a matrix is (the determinant and condition number) show that $\mathbf{M}$ is ill-conditioned. Unlike the geom-geom IBP case however, the residual is not (by itself) a good indicator of how accurate the solution actually is. A considerably better indicator is to look for negative entries in the $\mathbf{b}^{\circ}$ vector once it is obtained. These terms begin to appear when the numerical precision required to solve the linear system exceeds the precision used — a problem which can show up when the state space of the linear system is as small as 12 elements.

Figure 5.7 shows the empty system vector obtained using a double precision LU decomposition for an example system with 4 IBP sources and a cyclic source having $C = 16$ and $b = 9$. Obviously this solution is a long way from representing the correct empty system probability vector. Part of the problem is that the relationship of each probability term to some number of other terms means that once one of these becomes significantly negative, it causes adjacent values to be perturbed, giving rise in the extreme to the almost alternating spike effect apparent in Figure 5.7.

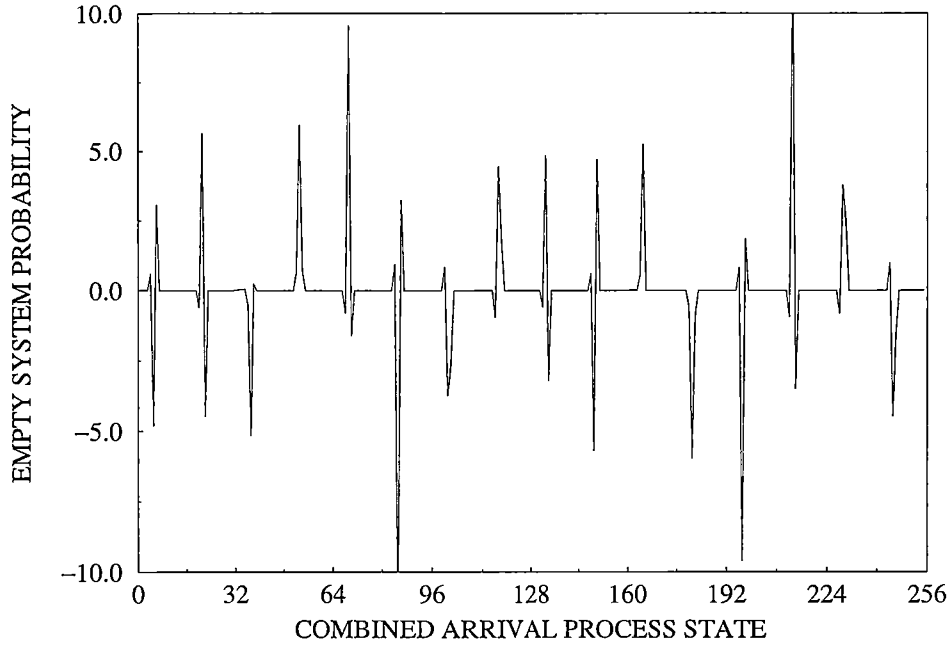

Figure 5.7: *Diagram showing the values obtained for the entries of the empty system probability vector using double-precision LU decomposition. The example system has* 4 *geom-geom IBP sources and a cyclic source with* $C = 16$ *and* $b = 9$.

Increasing the Numeric Precision

One way around this problem is to increase the numeric precision used in the calculations. As an example, Figure 5.8 shows the empty system vector obtained using quadruple precision LU decomposition for the same queueing problem considered in Figure 5.7. The actual probabilities, obtained using the numeric iterative solution method, are shown for comparison. We can see that although the results are much improved, they are still not exact, implying that still higher numeric precision would be required to solve the system exactly.

The difficulty here of course is that increasing the precision means both longer computation times, and increased memory demands. The double precision used as standard for the calculations throughout this thesis has 15 decimal digits of precision, requiring 8 bytes of storage per number. The quadruple precision used to calculate the results show in Figure 5.8 has a precision of 33 decimal digits, and requires 16 bytes per number. More importantly, although double precision floating point calculations are implemented in hardware on most modern computer platforms, quadruple precision floating point hardware is far less common. As a consequence, these types of calculations must then be done in software, resulting in extremely long run times. For example, the results of Figure 5.8 required some 233 seconds computation time compared to just

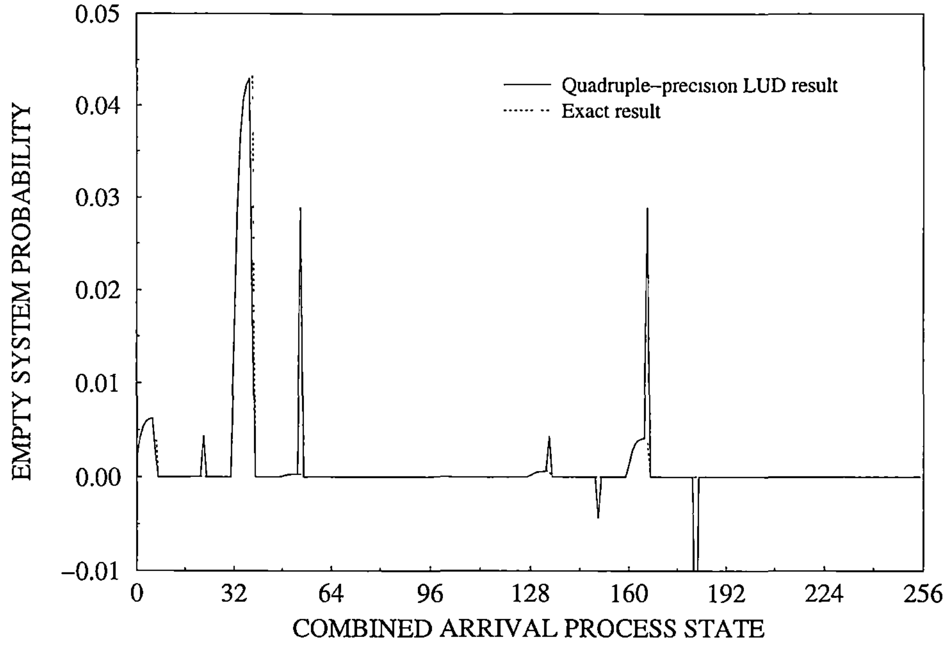

Figure 5.8: *Diagram showing the values obtained for the empty system probability vector using quadruple precision LU decomposition on the queueing problem considered in Figure 5.7. The actual probabilities (obtained from the matrix iterative solution method) are also shown.*

3 seconds for the double precision solution.

Singular Value Decomposition

One common method for dealing with ill-conditioned problems is to perform the singular valued decomposition (SVD) of the matrix rather than the LU decomposition. This method obtains the singular values of the matrix, along with sets of orthogonal vectors describing its range and null space. The explicit knowledge of the singular values allows us to exclude from the calculations those components which either belong in the null space, or are numerically close to this. A more complete description of the properties of the SVD, and in particular an implementation of the SVD algorithm can be found in [109].

Figure 5.9 shows the empty system vector obtained using double precision singular value decomposition for the same queueing problem considered previously in Figure 5.7. In this case, those vector components having singular values smaller than 10^{-14} times the largest singular value were excluded from the calculations. As can be seen, the results are an improvement on the double precision LU decomposition, but there is still a significant deviation from the actual solution.

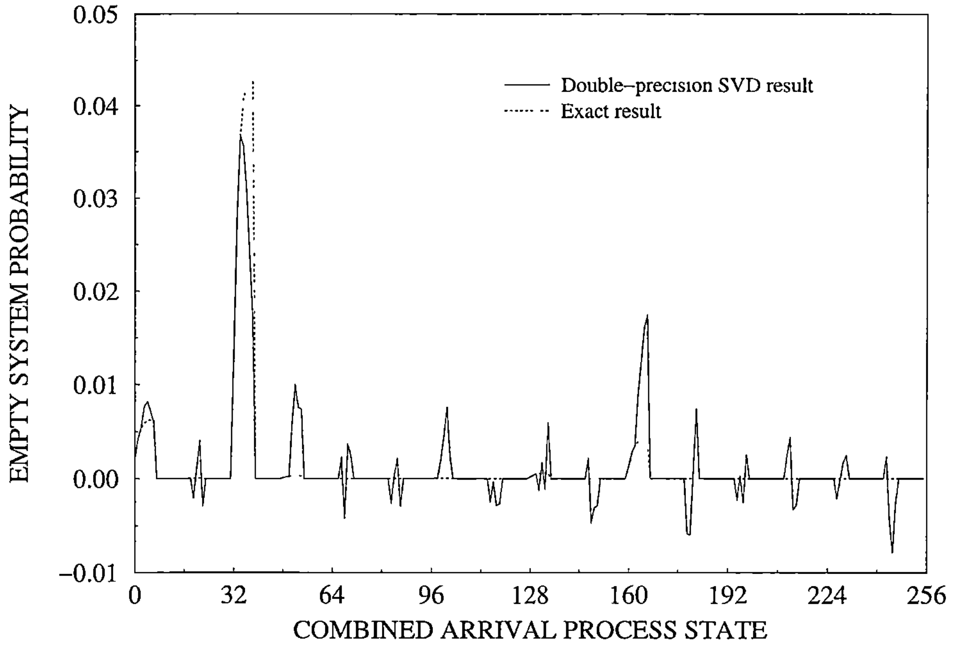

Figure 5.9: *Diagram showing the values obtained for the empty system probability vector using double precision LU decomposition on the queueing problem of Figure 5.7. The actual probabilities (obtained from the matrix iterative solution method) are also shown.*

Although the SVD process is supposed to provide a least squares approximation to the correct solution (in the sense that its residual is minimised) it does not appear to provide particularly accurate solutions for the queueing systems under study. The reason for this poor showing in the method most recognised for solving ill-conditioned problems is probably due to the high precision required to arrive at the SVD form of the matrix in the first place. We have not explored this possibility any further, but it is a likely explanation.

A further disadvantage to using the SVD approach over that of LU decomposition is that the SVD is considerably slower, with run times roughly proportional to m^4 where m is the dimension of the matrix. Thus, where possible it is desirable to perform the LU decomposition rather than the SVD. From our study of those factors that might signal a possibly poor solution, it was found that the dominant factor was the magnitude of the non-zero poles. The smaller these became, the more difficulty was encountered in obtaining an accurate solution. This observation led to the following adaptive approximation method which, as we will see, has a very high accuracy.

An Adaptive Solution Approach

We consider here an solution method which is based on successively removing the probabilities in those cycles having the smallest pole magnitudes until the LU decomposition returns positive probabilities only. We refer to this method as an adaptive solution since it repeats the process only as many times as necessary for each problem.

Once all the poles of the queue equation are located, the maximum magnitude of the $C - b$ poles in each of the 2^N cycles is identified. If the LU decomposition returns negative entries in the reduced state empty system vector $\mathbf{b}^\circ$ then the cycle having the smallest of these maximum magnitudes is removed from the analysis. This simply means that all the probabilities in that cycle are assumed to be zero, so that the state space of the $\mathbf{b}^\circ$ vector is reduced by $C - b$ elements. This process is repeated as many times as necessary until either all the entries in $\mathbf{b}^\circ$ are positive, or there is only one cycle left (which will be the first cycle since it contains one pole with magnitude 1 — the largest of all the poles in the unit circle). We have never observed the situation with only the first cycle remaining to result in negative probabilities, although this might possibly occur for very large C.

Figure 5.10 shows the results for the same problem as discussed previously, but using this adaptive method. The process had to remove 7 of the original 16 cycles in order to obtain the solution. This is clearer in Figure 5.11, which shows the empty system probabilities using a logarithmic scale. Most of the probabilities in the adaptive solution agree fairly closely with the exact solution, with the exception of cycle 5 (counting from 0 at the left). Some deviation is to be expected simply because the sum of all the probabilities is still $1 - \lambda$, even though we have removed some of the cycles in the adaptive case. We expect the magnitude of this deviation to increase as the number of cycles removed increases.

This adaptive process can take a considerable amount of time if the initial number of unknowns is large, since an LU decomposition is performed at each stage. It is an advantage therefore to initially remove those cycles that will probably be removed during the adaptive process anyway. Our results appear to indicate that cycles with the largest pole magnitudes less than 10^{-8} are good candidates for this initial removal when using double precision.

Another adaptive process that was considered involved removing (as required) those cycles with the smallest probabilities — that is, those probabilities that were already very close to zero can be made equal to zero with little consequence to the accuracy. The difficulty with this approach appears to be in deciding which cycles have the smallest probabilities, since we are trying to obtain the probability vector. In this case, Xiong

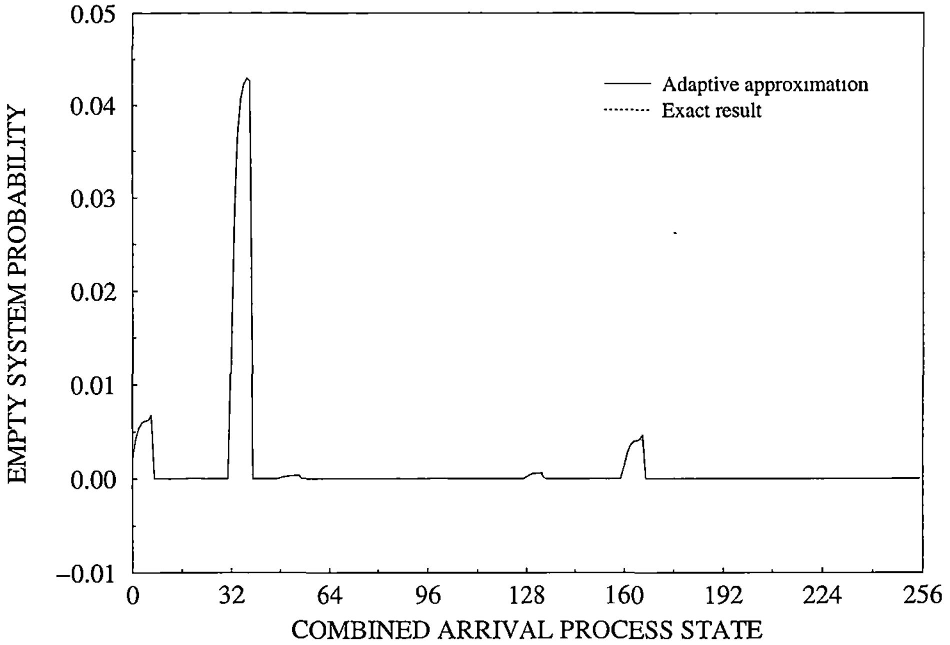

Figure 5.10: *Diagram showing the values obtained for the empty system probability vector using the adaptive approximation technique on the queueing problem of Figure 5.7. The actual probabilities (obtained from the iterative solution method) are also shown.*

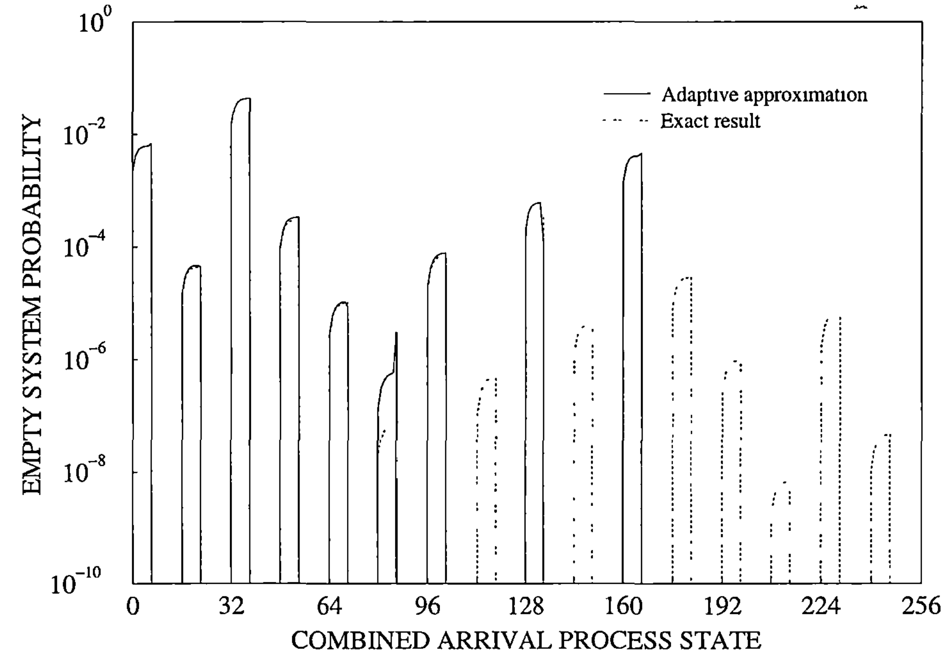

Figure 5.11: *The results of figure 5.10 shown using a logarithmic scale. The zero valued probabilities are the reason for the vertical lines in the plot, as the logarithm result becomes $-\infty$ at these points.*

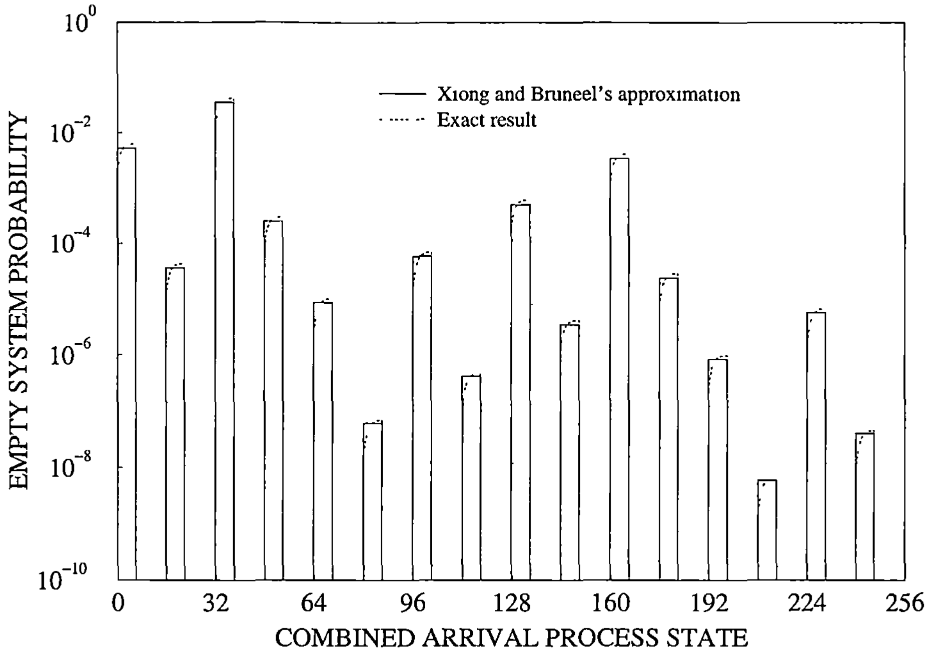

Figure 5.12: *Comparison between the exact empty system probability vector and the approximation of Xiong and Bruneel (discussed in more detail in section 3.5).*

and Bruneel's approximation [141] to the empty system probability vector (discussed in more detail in section 3.5) can be used to estimate these probabilities quite accurately. Figure 5.12 shows the actual probabilities and the approximation obtained from Xiong and Bruneel's work, again using only the non-zero probabilities and a logarithmic scale.

Obviously Xiong and Bruneel's approximation is quite suitable for picking the smallest probabilities. However, in investigating the performance of the adaptive solution based on the smallest probabilities we observed that this method nearly always removed a superset of those cycles that the adaptive process based on the pole magnitudes had removed. That is, more cycles were removed than were usually necessary (compared to the pole based method) and hence the solution accuracy was also lower. In addition, it was not possible to see a pattern in those cycles that would allow likely candidates to be removed before the first LU decomposition was performed in order to speed up run times. The maximum pole magnitude based adaptive solution therefore appears to be superior to the probability based one, and will be used here.

So far we have concerned our discussion to consideration of the accuracy of the solutions for the empty system probability vector. Since our ultimate aim is to determine the average and variance of the queue population for the queueing system, in the following section we will look at the accuracy of the pole based adaptive approximation. Before proceeding to this discussion however, it is interesting to compare the population results obtained from the various methods for obtaining **b** discussed above. These results are presented in Table 5.2, along with the exact solution obtained from numeric iterative analysis[1]. Although the results are all quite close in this particular example, this is not usually the case (with the exception of the two adaptive methods).

Method	Average	Variance
Double Precision LUD	0.3644958	0.5111699
Quadruple Precision LUD	0.4068025	0.5668265
Double Precision SVD	0.3988731	0.5443205
Pole Based Adaptive	0.4070765	0.5676050
Probability Based Adaptive	0.4065476	0.5672422
Iterative Solution (Exact)	0.4070751	0.5675998

Table 5.2: *Comparison of the population average and variance results for the example queueing problem considered in Figures 5.7 to 5.11.*

Other Possible Solution Methods

Neither the LU nor singular value decomposition methods discussed previously have the inherent constraint that the probabilities of the result be non-negative. Adding this constraint to the linear equations would bring it into the field of *linear programming* [122]. Linear programming solution methods basically operate by firstly identifying those variables spanning the range of **M**, and then minimising an objective function (using steepest descent methods or similar) that describes the desired solution. When the system has an equal number of equations and unknowns, but is merely ill-conditioned, identification of the range of **M** (using singular value decomposition for example) can be quite computationally intensive. This is in addition to the time required to then find the optimal solution, which can be considerable for problems with large state spaces. For this reason, and given the high accuracy of the adaptive solution, we will not pursue this approach further.

5.2.3 Investigating the Adaptive Solution Performance

In this section we look at the accuracy and speed of the adaptive solution for finding the empty system probability vector. We have chosen to investigate three different cycle periods ($C = 8$, $C = 12$, and $C = 16$) for 2 and 4 sources, using b values for each C that correspond to 25%, 50%, and 75% of the server capacity (a total of 18 different system combinations).

[1] We refer to this solution as exact since it was constructed using a convergence error of less than 10^{-12}, with a loss probability smaller than this

For each combination, a total of 1000 queueing problems were generated using random selection of the parameters of the geom-geom IBP sources. The average arrival rates λ_i from the IBP sources are chosen under the constraints that the overall load is less than 0.9 and the minimum total load contribution of the IBP sources is 0.05. The peak arrival rate θ_i when each source i is active is chosen from the range $(\lambda_i, 1]$ while the autocorrelation parameters are restricted to the range $0 \leq \gamma_i < 0.99$. These 18,000 randomly generated queueing problems are used for all the results in this section.

Frequency of Errors

Tables 5.3 and 5.4 show the number of inaccurate solutions observed per 1000 randomly generated queueing problems for various combinations of b and C, for two and four IBP sources respectively. The cause of the inaccuracy in the solution is identified in these tables as either a pole placement error (discussed in section 5.2.1) or a numeric precision error (negative entries in the double precision LU decomposition).

b	C	Pole Placement	Numeric Precision
2	8	4	0
4	8	1	1
6	8	0	0
3	12	6	1
6	12	0	30
9	12	0	91
4	16	6	3
8	16	1	159
12	16	0	327

Table 5.3: *Numbers of inaccurate solutions per* 1000 *queueing problems for* 2 *geom-geom IBP sources and a single cyclic source with the indicated parameters. Pole placement inaccuracies refer to those in which the exact pole could not be found while numeric precision inaccuracies indicate those problems generating negative probabilities in the (double precision) LU decomposition.*

Pole placement errors are fortunately fairly rare, but appear to be more common when there are fewer sources, smaller cycle periods, and/or a smaller component of the load contributed by the cyclic source. For the purposes of these results, a pole placement error was deemed to have occurred if the error in the pole equation (5.20) was greater than 10^{-10} for the best z_j^* obtained. Numeric precision errors (which indicate that negative probabilities were obtained for the initial LU decomposition) are obviously a frequent problem however, increasing as both the number of sources and the period of

b	C	Pole Placement	Numeric Precision
2	8	0	0
4	8	0	16
6	8	0	11
3	12	4	7
6	12	1	337
9	12	0	517
4	16	1	86
8	16	1	704
12	16	0	879

Table 5.4: *Numbers of inaccurate solutions per* 1000 *queueing problems for* 4 *geom-geom IBP sources and a single cyclic source with the indicated parameters.*

the cycle increase. Obviously, being able to overcome the limitations of double precision calculations, as discussed in the previous section, is an important requirement.

Error Solution Accuracy

As suggested previously by the results of Table 5.2, the pole magnitude based adaptive solution method is very accurate. To put this claim in more concrete terms, Tables 5.5 and 5.6 present the mean and standard deviation of the relative error[2] in the adaptive results for 2 and 4 IBP sources respectively, obtained from the randomly generated queueing problems of Tables 5.3 and 5.4. The results are for inaccuracies in the solution due to numeric precision only — we will look at errors due to pole placement separately. The exact results used to measure the relative error were obtained from the iterative solution method using a relative convergence error of 10^{-9} in the average population, and loss probabilities below 10^{-8} in each case.

Note that the adaptive solution method was implemented without any initial removal of cycles, so these error results represent this method's best accuracy performance. The error results are also limited to the first 100 inaccurate solutions in every case because of the extremely long run-times required to solve the queueing problems using the numeric iterative method. In addition, relative error statistics of less than 10^{-6} are considered to be zero, and are shown as this in the tables.

In general terms its seems that despite the higher frequency of numeric precision problems for the 4 IBP source case, the adaptive solution method performs better for larger numbers of sources. This may be partly due to the fact that increasing N means there

[2] We have used the same definition for the relative error as discussed on page 66 for queueing systems subject to geom-geom IBP sources only

			Population Average		Population Variance	
b	C	Points	Mean	Deviation	Mean	Deviation
2	8	4	0.02	0.07	−0.01	0.04
4	8	2	0.03	0.04	−0.02	−0.02
6	8	0	–	–	–	–
3	12	1	0	–	0	–
6	12	30	0	7×10^{-5}	0	4×10^{-5}
9	12	91	3.5×10^{-4}	0.0033	2.3×10^{-4}	0.0023
4	16	3	0	0	0	0
8	16	100	-4.5×10^{-5}	0.0014	5.2×10^{-5}	0.0010
12	16	100	0.0017	0.0084	0.0012	0.0062

Table 5.5: *Statistics on the relative error observed for the adaptive solution method applied to those problems encountering numeric difficulties with one cyclic source and* 2 *IBP sources. Note that, as indicated by the Points column, some of the results have very few data points from which to measure the statistics, so care must be taken in interpreting these errors.*

			Population Average		Population Variance	
b	C	Points	Mean	Deviation	Mean	Deviation
2	8	0	–	–	–	–
4	8	16	8×10^{-6}	3×10^{-5}	4×10^{-6}	2×10^{-5}
6	8	11	0	0	0	0
3	12	7	0	0	0	0
6	12	100	-1.4×10^{-5}	3.6×10^{-4}	5.6×10^{-6}	1.2×10^{-4}
9	12	100	7.1×10^{-5}	8.7×10^{-4}	2×10^{-5}	1.8×10^{-4}
4	16	86	-4.6×10^{-5}	4.7×10^{-4}	3×10^{-6}	6×10^{-5}
8	16	100	4.4×10^{-5}	7.1×10^{-4}	2×10^{-5}	2.9×10^{-4}
12	16	100	0.0015	0.0091	8×10^{-4}	0.0056

Table 5.6: *Statistics on the relative error observed for the adaptive solution method. The results are as for Table 5.5, but for queueing problems involving* 4 *IBP sources, rather than two.*

		Two IBP Sources			Four IBP Sources		
b	C	Mean	Deviation	Maximum	Mean	Deviation	Maximum
2	8	–	–	–	–	–	–
4	8	1	–	1	2.2	1.4	5
6	8	–	–	–	2.3	1.3	5
3	12	1	–	1	1.9	0.7	3
6	12	1.2	0.4	2	3.7	2.2	12
9	12	1.4	0.5	2	4.5	2.7	12
4	16	1.3	0.6	2	2.6	2.1	12
8	16	1.3	0.5	3	5.3	2.9	12
12	16	1.5	0.6	3	6.7	3.1	14

Table 5.7: *Mean and standard deviation of the number of cycles removed by the adaptive solution approach for those problems having inaccuracies due to numeric precision.*

are more cycles available in total, and hence possibly more cycles remaining once the adaptive method has found a positive solution for the empty system probability vector. In more quantitative terms, Table 5.7 shows the mean, standard deviation, and maximum for the number of cycles removed by the adaptive method for those randomly generated queueing problems suffering from numeric precision inaccuracies.

Another trend in the adaptive precision method apparent from Tables 5.5 and 5.6 is that its accuracy decreases as b gets larger for fixed C. This may be due in part to the fact that more cycles are removed on average for these cases, implying a corresponding decrease in the solution accuracy.

In contrast to this, it has been observed in some cases that removing one more cycle than is necessary to achieve a positive $\mathbf{b}$ vector can actually improve the accuracy of the population average and variance. An example of this was observed for $b = 12$, $C = 16$, and $N = 2$. The adaptive method only needed to remove one cycle to achieve the positivity requirement, but the resulting average and variance solutions had relative errors of 4.5% and 2.9% respectively. Forcing the adaptive solution to remove a second cycle resulted in errors of only -0.2% and -0.09% respectively. Whether this type of behaviour is merely coincidental, or an indication that there is a superior adaption criteria is an area requiring further research.

Table 5.8 lists the mean relative errors in the average and variance of the queue population for those cases where the solution inaccuracy was due to difficulties in finding the correct pole location. These errors can be quite large, which is why they have been separated from the adaptive solution results, which would otherwise be distorted. Also, due to the very small number of pole placement errors actually observed, we have only

		Two IBP Sources			Four IBP Sources		
b	C	Points	Average	Variance	Points	Average	Variance
2	8	4	2%	−1%	0	–	–
4	8	1	3%	4%	0	–	–
6	8	0	–	–	0	–	–
3	12	6	120%	−120%	4	−0.2%	0.4%
6	12	0	–	–	1	30%	−8%
9	12	0	–	–	0	–	–
4	16	6	0.1%	−0.4%	1	17%	−4%
8	16	1	2%	−2%	1	0.5%	0.8%
12	16	0	–	–	0	–	–

Table 5.8: *Mean relative errors in the average and variance of the predicted queue population due to pole placement inaccuracies. The points column shows how rare these events are, since these are the number of occurrences in* 1000 *samples for each b and C.*

presented the mean relative errors in Table 5.8, but even these figures are only a guide to the actual errors caused by the incorrect pole locations.

Solution Run-Times

An important consideration in the evaluation of the performance of an analytic approach lies in establishing its execution speed. Table 5.9 presents the mean run times (per problem) for each of the 18 queueing system arrangements observed on an IBM RS6000/320H workstation. For comparison, the mean run times for the iterative solution approach are also included for those cases where there were at least 10 solutions generated using this method.

As might be expected there is a decrease in the run times between the cases where the cyclic source load contribution increases from 25% to 50% because the number of states is decreasing. There is a large increase in the run times however as the cyclic source load increases to 75%, which can be attributed to two factors. Firstly, although the number of states (and hence the number of pole positions that must be found) has decreased, it has become more difficult to find each of the poles. A more optimum choice in the parameters of the search algorithm would probably reduce this component considerably. The second factor is the increased proportion of adaptive solutions required, since an adaptive solution requires more time for the multiple LU decompositions.

We have not considered more than 4 IBP sources in our discussion so far, primarily because of the large state space this incurs, and the subsequent difficulties in confirming

		Two IBP Sources		Four IBP Sources	
b	**C**	**Adaptive**	**Iterative**	**Adaptive**	**Iterative**
2	8	179 msec	–	1060 msec	–
4	8	145 msec	–	875 msec	690 sec
6	8	767 msec	–	6800 msec	1100 sec
3	12	251 msec	–	1680 msec	810 sec
6	12	195 msec	9.6 sec	1480 msec	1100 sec
9	12	780 msec	47 sec	6770 msec	1200 sec
4	16	337 msec	–	3460 msec	450 sec
8	16	255 msec	43 sec	3070 msec	1100 sec
12	16	833 msec	41 sec	7810 msec	1400 sec

Table 5.9: *Run times on an IBM RS6000/320H for the adaptive probability generating function approach and the numeric iterative solution method used to obtain the exact results for comparison with the adaptive solutions. If less than* 10 *iterative solutions were used, the run times have been omitted (represented by a – in the table).*

the accuracy of the probability generating function analysis using iterative methods. As can be seen in Table 5.9, the iterative solutions require very long run times. Even restricting the investigation to at most 100 iterative problems per system configuration, the total time to obtain all the exact solutions used in this section amounted to some 156 hours of CPU time.

The trend in the run times of the adaptive solution method is difficult to establish without more results, but we can speculate on the solution behaviour as b, C, and N vary. In Chapter 3 we discussed how the run times for geom-geom IBP sources increases roughly as 8^N for large N due to the requirements of the LU decomposition. Using the same reasoning, we would expect the LU decomposition part of the cyclic source problem to result in run times behaving approximately as $(C - b)^3\, 8^N$ for large N. The overall solution however probably requires multiple LU decompositions, as well as needing to find the $2^N\,(C - b)$ poles of the queue equation in the first place. Even if we only consider a single LU decomposition, we have the same problem for large N or large C as was encountered in Chapter 3 — queueing problems of practical size have impractical computational requirements. As in that chapter, we need to consider approximation methods.

5.3 Approximate Queue Population Solutions

As we have mentioned above, there is a definite need for approximate solution methods that avoid the state explosion problem. In this section we will be discussing three approximation methods. We will not be considering the more traditional methods such as the MMPP and geometric tail approximations, since these showed such poor performance in the studies of Chapter 3.

Unlike the studies of Chapter 3 we have a limited number of exact results available due to the frequency of inaccurate solutions arising from the methods of section 5.2, and the long run times required by more exact solution methods. As a consequence we will limit ourselves to considerations of the performance of the approximate solutions for just two system configurations ($b = 4$ with $C = 8$ and $b = 4$ with $C = 16$) for two and four IBP sources. It is hoped that future research can extend the scope of this study.

5.3.1 Modelling the Cyclic Process by a Random Process

In the introduction to this chapter, we mentioned that the majority of work done in this area has been based around a framed analysis, where the state of the queue is observed only once per frame time. We might consider whether this solution approach provides a simplification to the complexity of the problem. Unfortunately the analysis is no less complicated, and suffers from a similar number of unknowns. Li's analysis in [79] illustrates this point quite clearly.

In the work of Rosenberg and Le Bon [114] and Habibi et al. [41] however, the approach used was to obtain the remaining capacity in a cycle from the blocking probabilities of the periodic processes that made up the cyclic process, and to then use this capacity to obtain the queue performance for the remaining traffic. This suggests that a cyclic process might possibly be adequately described by a Bernoulli process, with the queueing analysis then becoming equivalent to an IBP only system.

Matching the parameters of a Bernoulli process to that of a cyclic process just requires $\lambda_0 = b/C$, where λ_0 is now the average arrival rate from the Bernoulli source. Incorporating this process into the IBP queueing system analysis involves writing $\theta_0 = 1$ and $\gamma_0 = 0$, and solving the resulting $N + 1$ IBP source system. A more efficient approach would be to incorporate the results of section 2.7, solving the N IBP source system and then adding in the Bernoulli source as a marginal arrival process.

We might also attempt to describe the cyclic process as an autocorrelated random process rather than just as a marginal arrival process. That is, we would like to find parameters θ_0 and γ_0 that would describe the cyclic process fairly well. From consideration of the fact that we have a cyclic process that generates b consecutive arrivals every cycle, a natural choice for θ_0 is 1. How do we then choose the autocorrelation parameter γ_0? In the MMPP approximation of section 3.4 we matched the single-sided autocorrelation sums of the superposed IBP sources to that of the MMPP. The same reasoning for the cyclic process cannot be applied however, because it does not have a well defined autocorrelation sum (its autocorrelation function is periodic), and other methods must be used.

One approach is to assume that the contribution of the cyclic process to the queueing performance can be wholly described by its effect on the marginal components of the IBP sources. That is, we look first at the population of the queue when all the IBP sources are marginal ($\gamma_i = 0$), and then when all the IBP sources are marginal but the cyclic process is described by a geom-geom binary source. Since we have a closed form expression for the average queue population in this latter situation, γ_0 can be found quite easily once L_q for the cyclic and marginal arrivals is known. Thus we are reducing the complexity of the problem from that of finding $2^N\ (C - b)$ unknowns in the empty system probability vector to that of finding $C - b$ unknowns in order to establish γ_0, followed by 2^N unknowns in solving the IBP system[3].

A second possible improvement in this geom-geom binary model of the cyclic process would be to somehow include a measure of the autocorrelation of the IBP sources. One way to do this is to describe each geom-geom IBP source by a roughly equivalent geom-geom binary source. That is, we set each θ_i equal to 1, and modify γ_i so that the single-sided autocorrelation sums of the two processes are equal. If we denote by γ_i' the autocorrelation parameter of the geom-geom binary equivalent then this gives

$$\frac{\gamma_i'}{1-\gamma_i'} = \left(\frac{\theta_i - \lambda_i}{1-\lambda_i}\right)\frac{\gamma_i}{1-\gamma_i} \tag{5.33}$$

or

$$\gamma_i' = \frac{\gamma_i\left(\theta_i - \lambda_i\right)}{1-\lambda_i-\gamma_i\left(1-\theta_i\right)} \tag{5.34}$$

Since geom-geom binary sources each generate a single arrival per time slot in their active states, there will still only be $C - b$ unknowns in the analysis of the queue with one cyclic arrival and N geom-geom binary sources.

As an example of the performance of these three approximations, we consider a queue fed by 4 identical IBP sources and a cyclic source with $b = 4$ and $C = 16$. The IBP sources each have $\lambda_i = 0.15$, $\theta_i = 0.6$, and $\gamma_i = \gamma$ (γ is the independent variable of the analysis). Figures 5.13 and 5.14 show the queue population average and variance respectively for the exact method and for the Bernoulli and two geom-geom binary approximations as a function of $\gamma/\left(1-\gamma\right)$. Obviously the second geom-geom binary

[3] Although there are actually $N + 1$ sources in this queueing problem now, there are still only 2^N unknowns because source 0 is a geom-geom binary with $\theta_0 = 1$. That is, every state in which source 0 is active has an empty system probability of zero, leaving only 2^N states in which this probability is unknown.

approximation is the best alternative of the three — particularly as its computational requirements are basically identical to the first geom-geom binary method.

While all three approximations overestimate the population variance in Figure 5.14, the overestimation is nearly constant in the second geom-geom binary approximation. This suggests a simple addition to improve the variance accuracy of this method by calculating the γ_0 term for the first geom-geom binary approximation (which will involve an additional $C-b$ unknowns in a cyclic and marginal arrivals analysis). The difference in the variance between the first geom-geom binary approximation when the IBP sources are marginal, and the actual cyclic process analysis gives an approximately constant difference to improve the second geom-geom binary approximation. In graphical terms we are calculating the difference between the two curves in Figure 5.14 at $\gamma = 0$, and subtracting this from the second geom-geom binary approximation at the desired γ value. We will refer to this as the third geom-geom binary approximation.

An alternative approach again would be to extend the geom-geom binary model to a phase-geom binary model, and hence be able to more accurately describe the variance by incorporating knowledge of the third moment of the active period into the calculations. Although the number of unknowns in solving for the empty system probability vector does not increase when incorporating this type of model, some of the attendant mathematics needs modification. We will not go into this any further here, but note that this approach is worth future consideration.

Note that, although we have not explicitly mentioned it, the analysis of the IBP queueing system could easily be performed using a second or third order approximation (see section 3.6) rather than directly trying to solve for the the 2^N unknowns of the exact system. This approach becomes particularly attractive considering that the expected error of even the second order approximation is a good deal smaller than the overall approximation error, as we shall see.

Comparative Accuracy Study

As mentioned at the beginning of this section, we are considering the accuracy of the four approximations only for 4 system configurations for which exact population results were obtained in section 5.2.3. Tables 5.10 and 5.11 present the mean absolute relative error for the population average and variance respectively for each of the four approximations. The relative error is measured as described in section 3.2 and we use its absolute value[4] in this comparison to avoid the problem where a wide spread of

[4] The relative error used in this thesis is defined to be the difference between the approximation and the exact value, expressed as a proportion of the exact value The sign of this error measure is positive if the approximation is greater than the exact value.

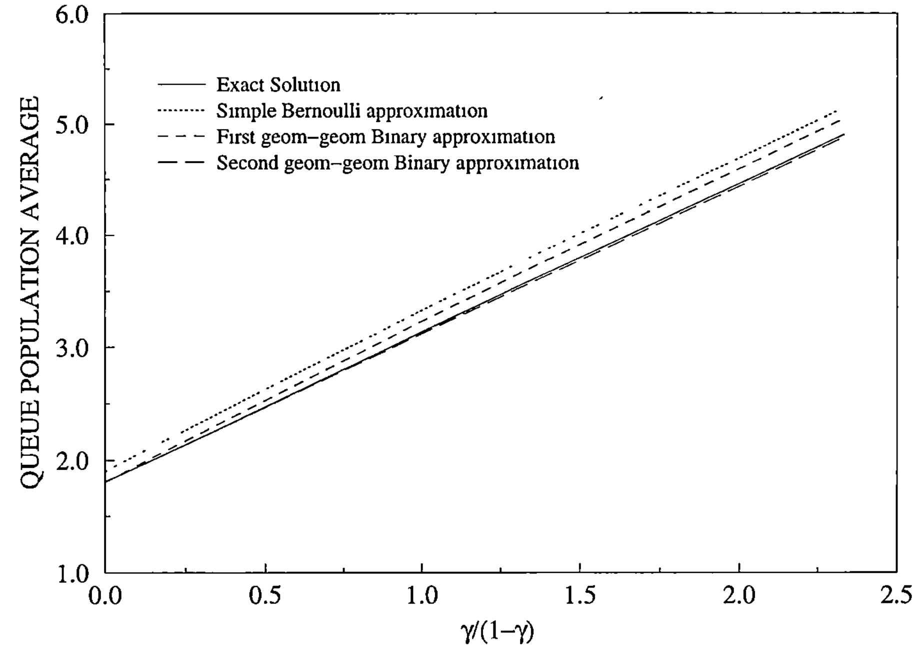

Figure 5.13: *Average queue population for a queue fed by a cyclic arrival process with $b = 4$ and $C = 16$ and four identical IBP sources with $\lambda_i = 0.15$ and $\theta_i = 0.6$. The $\gamma/(1-\gamma)$ independent variable is from the $\gamma_i = \gamma$ of the IBP sources. Note that the first geom-geom binary approximation is calculated by treating the IBP sources as marginals, while the second approximation is calculated by treating the IBP sources as geom-geom binary sources (with suitably modified autocorrelation parameters).*

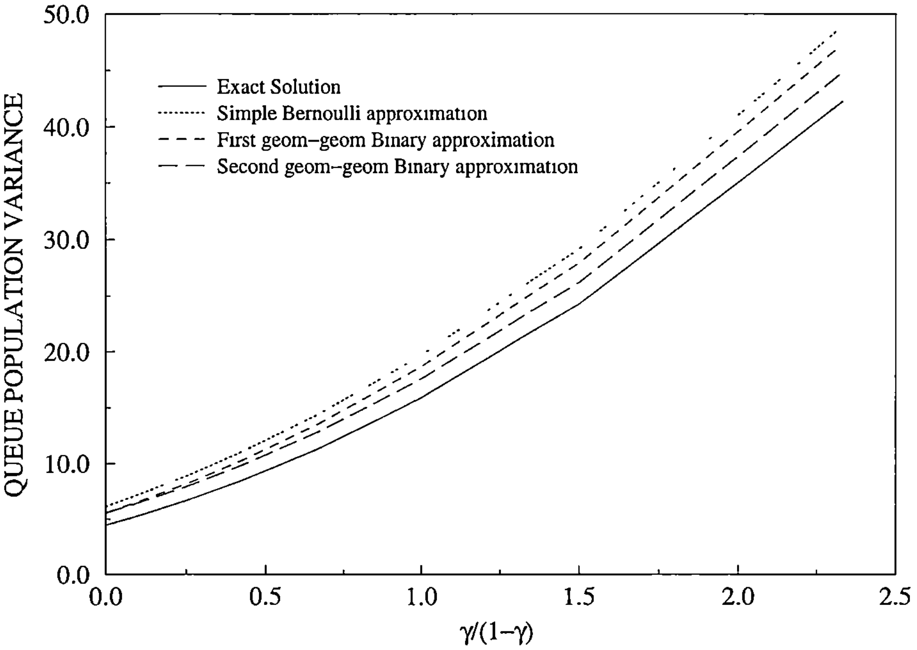

Figure 5.14: *Queue population variance for the queueing problem of Figure 5.13.*

b	C	N	Bernoulli	First	Second	Third
4	8	2	13.6%	5.76%	1.52%	1.52%
4	16	2	15.6%	5.78%	1.96%	1.96%
4	8	4	11.4%	5.26%	1.25%	1.25%
4	16	4	12.1%	5.29%	1.43%	1.43%

Table 5.10: *Mean of the absolute value of the relative error in the average queue population for each of the four approximation methods proposed in this section. Note that the second and third geom-geom binary approximations are identical in terms of the average queue population. Each row of the table represents observations over* 1000 *randomly generated queueing problems with* $\lambda < 0.9$ *and* $0 \leq \gamma_i < 0.99$.

b	C	N	Bernoulli	First	Second	Third
4	8	2	38.6%	22.2%	9.75%	2.82%
4	16	2	23.4%	29.6%	13.8%	4.32%
4	8	4	29.2%	18.1%	8.29%	3.74%
4	16	4	16.1%	23.6%	11.3%	5.15%

Table 5.11: *Mean absolute relative error in the queue population variance for each of the four approximation methods.*

errors might misleadingly result in a mean error very close to zero.

We present a more comprehensive set of error statistics for the third approximation method, which is the best of these four proposed methods. Tables 5.12 and 5.13 present the mean, standard deviation, first and 99th percentiles for the relative error (not the absolute relative error) in the average and variance respectively of the queue population obtained from this approximation method.

The performance of this geom-geom binary approximation is not particularly good, although in some situations it may be adequate. Of real concern is the fact that the

b	C	N	Mean	Deviation	1st Percentile	99th Percentile
4	8	2	−1.49%	2.19%	−9.86%	0.19%
4	16	2	−1.87%	3.71%	−18.7%	1.69%
4	8	4	−1.21%	1.40%	−6.90%	0.68%
4	16	4	−1.32%	1.99%	−9.91%	1.23%

Table 5.12: *Statistics on the relative error in the average queue population for the third geom-geom binary approximation method (which is identical to the second for the average queue population).*

b	C	N	**Mean**	**Deviation**	**1st Percentile**	**99th Percentile**
4	8	2	2.08%	3.76%	−8.46%	16.4%
4	16	2	2.44%	6.64%	−23.8%	21.8%
4	8	4	3.59%	3.15%	−1.67%	15.3%
4	16	4	4.77%	5.12%	−8.30%	23.7%

Table 5.13: *Statistics on the relative error in the queue population variance for the third geom-geom binary approximation method. This method tries to accomodate the inaccuracies in the variance of the two-state approximation for the cyclic process by using a linear comparison with the marginal arrivals problem.*

accuracy of the approximation appears to decrease both with increasing numbers of IBP sources and increasing cycle length. This behaviour is particularly undesirable for an approximation aimed at describing the behaviour of large queueing problems.

5.3.2 Using the Adaptive Solution Approach

The adaptive solution approach presented in section 5.2.2 reduces the number of cycles in the LU decomposition in order to maintain only positive entries in the resulting empty system probability vector. An extension of this idea would be to approximate the solution by using only the first cycle of probabilities, reducing the number of unknowns to $C-b$. This method also represents the worst case situation in the adaptive solution method, and is therefore of interest in this regard.

We note that this method will provide exact results when all of the sources are either marginal ($\gamma_i = 0$) or geom-geom binary ($\theta_i = 1$) since in both of these cases only a single cycle of $C - b$ unknowns is required to establish the empty system probability vector **b**. Generally of course these conditions will not be satisfied, and this method will provide only approximate results.

As a preliminary investigation we consider an example problem with $b = 4$ and $C = 16$ using four identical IBP sources having $\lambda_i = 0.15$, $\theta_i = 0.3$ and $\theta_i = 0.7$, and $\gamma_i = \gamma$ which is the independent variable in the study. The results are shown in Figures 5.15 and 5.16 for the average and variance of the queue population respectively. Surprisingly, although the average queue population is poorly described by this approximation, the variance is very close. Unfortunately, as we shall see in the following more rigorous study, this is not usually the case.

Tables 5.14 and 5.15 present the statistics for the relative error in the approximation for the population average and variance respectively observed for a number of randomly

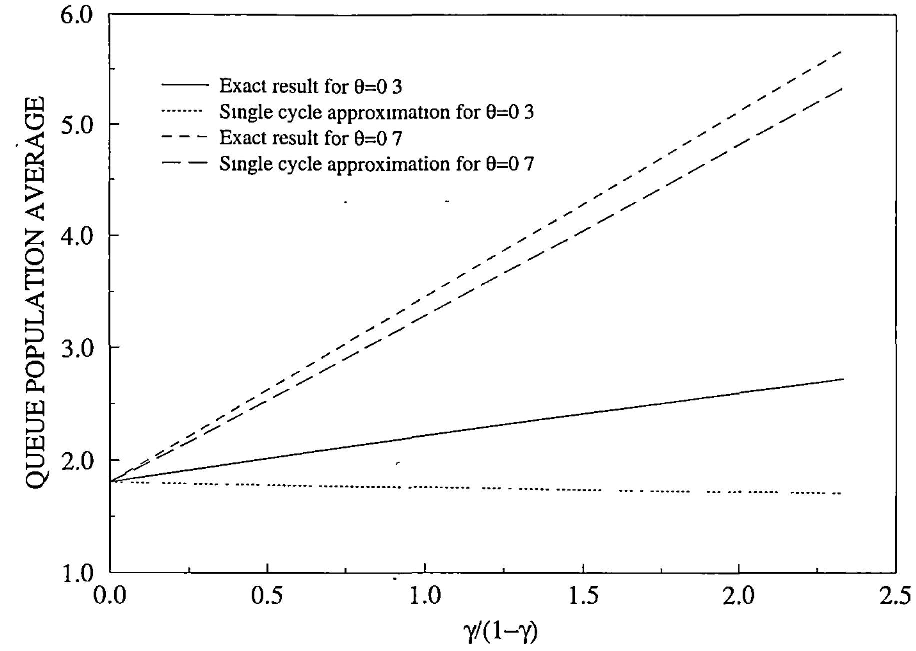

Figure 5.15: *Average queue population for a queue fed by a cyclic arrival process with $b = 4$ and $C = 16$ and four identical IBP sources with $\lambda_i = 0.15$ and $\theta_i = 0.3$ and $\theta_i = 0.7$. The $\gamma/(1-\gamma)$ independent variable is from the $\gamma_i = \gamma$ of the IBP sources.*

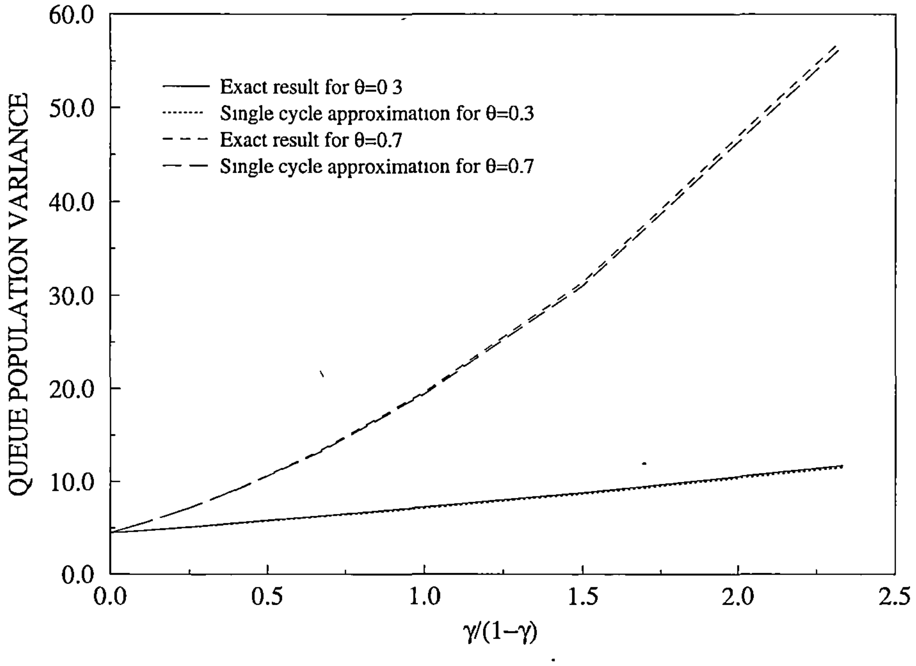

Figure 5.16: *Queue population variance for the queueing problem of Figure 5.15.*

b	C	N	Mean	Deviation	1st Percentile	99th Percentile
4	8	2	−33.6%	120%	−420%	−0.26%
4	16	2	−80.3%	260%	−1000%	−0.54%
4	8	4	−29.7%	88%	−250%	−0.39%
4	16	4	−66.8%	180%	−740%	−0.61%

Table 5.14: *Statistics on the relative error in the average queue population for the adaptive solution approximation. Each row of the table represents observations for* 1000 *randomly generated queueing problems with* $\lambda < 0.9$ *and* $0 \leq \gamma_i < 0.99$.

b	C	N	Mean	Deviation	1st Percentile	99th Percentile
4	8	2	227%	4700%	−15%	3000%
4	16	2	906%	13000%	−11%	15000%
4	8	4	154%	2300%	−13%	1800%
4	16	4	591%	7000%	−9.9%	8600%

Table 5.15: *Statistics on the relative error in the queue population variance for the adaptive solution approximation.*

generated queueing problems. As before, the results are restricted to just 4 system configurations.

The accuracy of this approximation is in general very poor, despite the results of Figure 5.16. In particular negative values for the average queue population, and variances many times that of the actual variance appear to be fairly common. The performance of the approximation appears to improve with the number of sources, and to decrease with increasing cycle period. However, given the magnitudes of the errors, it is doubtful that they could be reduced to acceptable levels for practical sized problems. It is not known what other ranges of parameters might allow this approximation to perform well.

Note that the mean and standard deviation of the relative error are calculated here on the entire data set. An alternative would have been to exclude the results lying outside of the 1st and 99th percentiles (the outliers) so as to describe the statistics of the majority of the relative errors. Although this would provide a more a favourable assessment the approximation, we have not used this option simply because the entire data set consists of valid results, and in addition, we are interested in the worst case performance of the adaptive approximation. In this regard, the worst case performance is quite poor indeed, although the conditions resulting in this situation are infrequent (for 4 IBP sources at least — see section 5.2.3).

5.3.3 The kth Order Approximation Method

In section 3.6 of Chapter 3 we presented a powerful new approximation technique for studying queueing systems involving IBP sources. This method was based around the idea of describing the overall behaviour of the queue by the sum of contributions from sets of at most k autocorrelated IBP sources — a process described as a kth order approximation. In section 3.6.4, a modification to this technique, called a 'kth order approximation with s held' was also presented. In the following we will make use of both of these methods to approximate a queueing system subject to arrivals from both IBP sources and a cyclic source.

Applying the kth Order Approximation

Applying the kth order approximation to the cyclic arrivals problem basically follows the same approach as for the IBP only sources case. We assume the system has $N + 1$ sources, and treat the cyclic source as a Bernoulli source in any combination of s autocorrelated sources ($s = 1, \ldots, k$) which do not include the cyclic source. The contributions from these combinations can therefore be determined using only the IBP source analysis. For those combinations of s sources which do include the cyclic source however, we use the exact cyclic arrivals analysis developed in section 5.1. The parameters for the analysis will involve $s - 1$ autocorrelated IBP sources, $N - s + 1$ marginal sources, and one cyclic source with parameters b and C, requiring the solution of $2^{s-1}(C - b)$ unknowns. The results of this analysis are included in the overall kth order approximation as with any other combination.

Note that for a kth order approximation, there will be $\binom{N}{s-1}$ cyclic source solutions involving the cyclic source and $(s - 1)$ IBP sources for each $s = 1, \ldots, k$.

Holding the Cyclic Source

In this alternative, the cyclic source is included in every calculation. That is, every combination of s sources includes the cyclic source. Otherwise the procedure is the same as for the straight kth order approximation. In fact this approach requires exactly the same number of cyclic source solutions as the kth order approximation, but does not have the additional contributions from combinations of IBP sources only.

Comparative Accuracy Study

We will look here at the population accuracies obtained from four approximations — 1st order, 2nd order, 2nd order with the cyclic source held, and 3rd order with the

b	C	N	1st	2nd	2nd, 1 held	3rd, 1 held
4	8	2	5.71%	0.14%	0.33%	–
4	16	2	3.87%	0.25%	0.79%	–
4	8	4	5.09%	0.14%	0.46%	0.02%
4	16	4	3.29%	0.35%	1.24%	0.05%

Table 5.16: *Mean of the absolute value of the relative error in the average queue population for each of the four approximation methods. Each row of the table represents observations for* 1000 *randomly generated queueing problems with* $\lambda < 0.9$ *and* $0 \leq \gamma_i < 0.99$.

b	C	N	1st	2nd	2nd, 1 held	3rd, 1 held
4	8	2	16.2%	0.14%	3.52%	–
4	16	2	7.68%	0.19%	4.52%	–
4	8	4	12.1%	0.15%	7.05%	0.01%
4	16	4	7.25%	0.27%	9.01%	0.04%

Table 5.17: *Mean absolute relative error in the queue population variance for each of the four approximation methods.*

cyclic source held. Tables 5.16 and 5.17 show the mean values of the absolute relative error in the four approximations for the four system configurations. Note that the third order approximation with one held source will be exact for the system configurations involving two IBP sources, and this is indicated in the tabulated results by a '–'.

As expected from the results in section 3.6.4 for IBP sources, the four approximations can be ranked in terms of increasing accuracy (or decreasing error magnitude) as 1st order, 2nd order with the cyclic source held, 2nd order, and 3rd order with the cyclic source held. This is also illustrated by considering that while both of the 2nd order approximations involve $(N+1)$ solutions having cyclic arrivals, the straight 2nd order approximation additionally includes contributions from combinations of IBP sources only. For a queueing system subject to arrivals from one cyclic source and N IBP sources, this method requires

- N solutions involving the cyclic source, one IBP source, and $N-1$ Bernoulli sources
- One solution involving the cyclic source and N Bernoulli sources
- N solutions involving one IBP source and N Bernoulli sources, and
- $N\left(N-1\right)/2$ solutions involving two IBP sources and $\left(N-1\right)$ Bernoulli sources

b	C	N	Mean	Deviation	1st Percentile	99th Percentile
4	8	2	−0.13%	0.94%	−1.11%	0.08%
4	16	2	−0.23%	0.42%	−1.90%	0.13%
4	8	4	−0.13%	0.21%	−0.90%	0.09%
4	16	4	−0.35%	0.37%	−1.58%	0.08%

Table 5.18: *Statistics on the relative error in the average queue population for the* 2*nd order approximation method.*

b	C	N	Mean	Deviation	1st Percentile	99th Percentile
4	8	2	−0.03%	0.85%	−0.48%	1.73%
4	16	2	0.15%	0.44%	−0.36%	2.08%
4	8	4	−0.10%	0.23%	−0.62%	0.75%
4	16	4	0.15%	0.39%	−0.34%	1.74%

Table 5.19: *Statistics on the relative error in the queue population variance for the* 2*nd order approximation method.*

Tables 5.18 and 5.19 present the mean, standard deviation, and the 1st and 99th percentiles for the relative error in the average and variance respectively of the queue population obtained from the straight 2nd order approximation method. We have singled out this particular approximation basically because it provides fairly high accuracy for relatively low computational requirements. Few practical applications would require the additional accuracy of the 3rd order approximation, while the error magnitudes for the first order approximation are probably a little high.

5.3.4 Approximations Summary

Of the three approximation methods described in this section, only the kth order approximation first proposed in Chapter 3 for queueing problems involving IBP sources only was able to consistently achieve high accuracies over the range of randomly generated problems that have been used throughout this chapter.

The first approximation method examined, which attempted to describe the cyclic source by a geom-geom binary process was able to achieve mean errors of a few percent of so, but with a considerable spread around this. Further investigation with more extensive sets of accurate results may allow the useful range of this approximation to be identified. In addition, there is the possibility for using a phase-geom binary model for the cyclic source to improve the accuracy of the variance approximation.

Investigations into an approximation using a single cycle in the adaptive solution method discussed in section 5.2.2 of this chapter resulted in very poor error performance. Although it is a simple matter to disregard this approach as an approximation option, the implications for the range of errors that might be encountered by the adaptive solution method are not so easily ignored. Fortunately these types of errors in the adaptive solution are rare, and would probably be even more so if the recommended second order approximation is used.

5.4 Summary

The subject of this chapter has been the analysis of queues having arrivals from both a single cyclic source and from a heterogeneous mix of IBP sources. To be exact, in this chapter we have applied the probability generating function theory of Chapter 2 to the analysis of queues subject to both arrivals from IBP sources and to cyclic service by making use of Corollary 2.3. This corollary indicates that the average and variance of the population of a queue subject to cyclic (or periodic) service are identical that of an uninterrupted service queue subject to both the original queue's arrival process and an additional cyclic process having the complementary behaviour to the service process.

The difficulty with implementing the developed theory was found to be in calculating the empty system probabilities. Even for relatively small queueing problems numeric difficulties were encountered, both in terms of being able to correctly identify the poles of the queue equation, and in reducing the resulting system of linear equations. In order to overcome this problem a simple adaptive solution technique was presented and its accuracy investigated for a small set of system configurations. Results on the accuracy of this solution method are encouraging, although there appears to be room for further improvement in the adaption process.

As with most of these types of techniques, the computational complexity of the solution process is geometric in the number of IBP sources, and roughly linear in the length of the cycle period. As a means to overcoming this problem, three approximation techniques were investigated. Although somewhat limited by a fairly small set of accurate solutions with which to compare the approximations, it was found that the kth order approximation technique of Chapter 3 provided the best error performance. In particular the 2nd order approximation appears to be the best option in terms of accuracy and computational complexity.

Chapter 6

Finite Buffer Approximations and Queueing Delays

In the previous chapters we have discussed methods for obtaining the average and variance of the queue population for several queueing systems. The analysis in each case has been based on the assumption that the buffers involved have infinite capacity, which is of course impractical in a real world environment. In particular, the main concern for loss sensitive traffics is the performance of finite buffer queueing systems in regard to overflow behaviour. In the context of connection admission, it is important to be able to accurately predict the loss behaviour of a proposed traffic arrangement.

Another practical concern is that although queue population statistics are measures of network utilisation, it is the queueing delays that directly impact the performance seen by the various traffics using the network. We discuss some of the issues relating to obtaining queueing delays from knowledge of the queue population in sections 6.3 and 6.4.

We begin this chapter with a discussion of a simple approximation for the population distribution of a queue, and show how this may be used to obtain estimates for the average loss probability of a single buffer (uninterrupted service) queueing system. We will also briefly look at the requirements for extending this to dual buffer (prioritised) systems before moving onto a discussion of delay related issues.

6.1 The Geometric Tail Approximation

In Chapter 3 a geometric approximation (section 3.3) to the tail probabilities of a queueing system fed by IBP sources was used to provide approximations for the queue population statistics. This geometric tail property applies to a wide range of queueing systems, and in particular can be used both to estimate loss probabilities, and to estimate quantities such as the 95th or 99th percentiles of the delay.

Queue population and delay distributions are often described by probability density functions for their convenience of use. One useful alternative is the complementary cumulative distribution function, which describes the probability that an observed random variable is *greater* than some desired value — the so called *tail distribution.* We will discuss only the tail distribution of the queue population here, although the reasoning applies equally well to the queueing delay in uninterrupted service systems (see section 6.3).

Let q_n denote the stationary probability that the population of the queue is n, and let t_n denote the stationary probability that the queue population is strictly greater than n, so that

$$t_n = \sum_{i=n+1}^{\infty} q_i = 1 - \sum_{i=0}^{n} q_i \tag{6.1}$$

Desmet et al. [22] showed that any integer-valued and non-negative random variable having a rational probability generating function has a tail distribution that can be approximated by a geometric distribution (the distribution is asymptotically geometric). That is, provided it is possible to show that the probability generating function $q(z)$ of q_n can be described by a rational function, t_n can be approximated by

$$\tilde{t}_n = \psi\phi^n \tag{6.2}$$

for some ψ and ϕ for large n. This approximation is well known in the literature [1,66, 96,120,121,138,141,142] and is often referred to as the *geometric tail approximation.* The ψ and ϕ parameters are referred to as the geometric scaling factor and geometric coefficient.

It is straightforward to show that a queueing system subject to arrivals from a discrete-time Markov modulated process (or D-BMAP) satisfies the criteria of Desmet et al. by considering the form of the queue equation. From Chapter 2 we have $q(z) = \mathbf{X}(z)\mathbf{e}$ or

$$q(z) = (z-1)\,\mathbf{b}\,(z\mathbf{I} - \mathbf{AP}(z))^{-1}\,\mathbf{e} \tag{6.3}$$

The above matrix inverse can be written as

$$(z\mathbf{I} - \mathbf{AP}(z))^{-1} = \frac{1}{|z\mathbf{I} - \mathbf{AP}(z)|} \times \text{Adj}\ (z\mathbf{I} - \mathbf{AP}(z)) \tag{6.4}$$

where the determinant $|z\mathbf{I} - \mathbf{AP}(z)|$ and each of the elements of the adjoint matrix are polynomials in z, since each element of the $\mathbf{AP}(z)$ matrix is either polynomial in z (for finite probability distributions) or can be expressed to arbitrary precision as such (for infinite probability distributions such as the Poisson process). Then, since $\mathbf{b}$ is a row vector of constants, $q(z)$ must also be a rational function of polynomials in z, which is what we wished to establish.

Exact Analysis for ψ and ϕ

The accuracy of the geometric tail approximation increases as n increases, due to the dominance of a single pole in the expression for $q(z)$ having the smallest magnitude, as discussed in [22] and [1]. As it turns out, this pole is given by the single positive real solution to the equation $z - \delta(z) = 0$ lying outside the unit circle [120,141,142], where $\delta(z)$ is the Perron–Frobenius eigenvalue of the probability generating transition matrix of the combined arrival process. The value of ϕ is given simply by the reciprocal of this pole. In other words, ϕ may be obtained from the solution of the equation

$$\phi^{-1} - \delta(\phi^{-1}) = 0 \tag{6.5}$$

using numeric methods on $\phi \in (0,1)$, such as the bisection or Newton–Raphson methods.

Unfortunately there is no simple method to obtain ψ without solving for the entire empty system probability vector, and approximate solutions are usually used (as was discussed in Chapter 3). Note that Xiong and Bruneel present the basics for the calculation of ψ from the system probability vector in [141], but actually utilise an approximation for $\mathbf{b}$ instead. We will look again at the accuracy of this approximation in the context of loss probabilities later.

Approximations for ψ and ϕ

One alternative to obtaining the exact values for ψ and ϕ is to use an approximation that is based on knowing the infinite buffer queue population average and variance, or approximate values for these quantities. For example, in Chapter 3 we proposed a new approximation for L_q and $\mathrm{Var}[L_q]$ for queues subject to arrivals from IBP sources. This new method avoids the major stumbling block of calculating the empty system probability vector. Then in Chapter 5 we showed that this approximation also applies when a cyclic source is mixed with the IBP sources. In addition, Chapter 4 presented closed form solutions for the case when each IBP source i has $\theta_i = 1$.

For this approximation for ψ and ϕ we assume that the tail distribution is quite adequately described by equation (6.2), and an approximation for the probability density distribution of the queue population can then obtained as

$$\tilde{q}_n = \begin{cases} 1-\psi & \text{for } n=0 \\ \psi\,(1-\phi)\,\phi^{n-1} & \text{otherwise} \end{cases} \tag{6.6}$$

The corresponding approximations for the average and variance of the population are given respectively by

$$\tilde{L}_q = \frac{\psi}{1-\phi} \tag{6.7}$$

and

$$\mathrm{Var}\left[\tilde{L}_q\right] = \frac{\psi\,(1+\phi)-\psi^2}{(1-\phi)^2} \tag{6.8}$$

under the assumption that the buffer is infinite. More usefully, equations (6.7) and (6.8) can then be rearranged to provide estimations for ψ and ϕ from the known average and variance as

$$\psi = \frac{2L_q^2}{\mathrm{Var}\left[L_q\right]+L_q^2+L_q} \tag{6.9}$$

and

$$\phi = \frac{\mathrm{Var}\left[L_q\right]+L_q^2-L_q}{\mathrm{Var}\left[L_q\right]+L_q^2+L_q} \tag{6.10}$$

The accuracy of this estimation is dependent basically on the actual value of ϕ. When this quantity is close to one, both L_q and Var $[L_q]$ are dominated by the probabilities in the tail of the queue (q_n for large n). However, when ϕ is small, the main contributions to the queue population moments are from q_n for small n — which is where the geometric tail property does not apply. One way to overcome this problem is to calculate the exact value of ϕ (in general, not a significant computational problem) and combine this with the known average and variance using

$$\psi = \frac{(1-\phi)^2}{2\phi}\left(\mathrm{Var}\left[L_q\right]+L_q^2-L_q\right) \tag{6.11}$$

This approach attempts to bypass as many of the early probabilities as possible by making use of

$$\mathrm{Var}\left[L_q\right]+L_q^2-L_q = \sum_{n=2}^{\infty} n\,(n-1)\,q_n \tag{6.12}$$

Obviously, if higher moments of the queue population are available, more of these early probabilities can be excluded. Conversely, if only the average queue population is known, then equation (6.7) can be rearranged to give ψ from L_q and ϕ, although the accuracy of this result will be poorer.

6.2 Estimating Loss ,Probabilities

Probably the most important application of the geometric tail approximation is in calculating buffer overflow probabilities. Buffer overflow in a finite capacity queueing system occurs when incoming arrivals encounter a full buffer (no remaining storage capacity), resulting in the excess arrivals being discarded. Although there are other reasons for losses in finite buffer systems (such as in prioritised systems where previously queued arrivals may be pushed out by newer arrivals[1]) buffer overflow losses are implied whenever the term 'loss' is used here in an unqualified manner.

Like any stochastic process, buffer overflows may be described or measured in many ways. The most basic measure is the stationary average probability that an arrival to the queue will be lost, denoted here by ξ_K where the K subscript denotes the capacity of the buffer in terms of *waiting* cells — that is, not including the cell that might currently be receiving service.

In a real queueing system, losses are not evenly distributed over time, but tend to be clumped due to autocorrelations in the arrival process and the fact that a full buffer empties slowly. Thus another useful measure of the overflow process is the number of consecutive losses [51, 75, 130] and/or the time between these 'bursts' of lost cells [37].

In the following we will look at how the geometric tail approximation can be applied to estimating the average loss probability. We will not consider other higher order measures of the loss performance for these types of finite buffer problems, although this may be a profitable area for future investigation considering the current interest in the topic.

6.2.1 Predicting the Average Loss Probability for Marginal Arrival Processes

As a starting point we will consider the finite buffer behaviour of a discrete-time deterministic service queueing system fed by a marginal arrival process. As in the rest of this thesis, time is assumed to be divided into equal periods called slots, with a slot being equal to the time required to service one arrival (or cell). Define q_n to be the stationary probability that the queue population immediately after service is n, and we similarly define q_n^+ to be the corresponding probability immediately before service. Let p_k denote the probability that there are exactly k arrivals from the combined marginal arrival process in any time slot.

For a buffer of size K, the *arrival relation* is described simply by

$$q_n^+ = \sum_{i=0}^{n} q_i p_{n-i} \quad \text{for } n < K \tag{6.13}$$

[1] Even in this situation, the losses may still be regarded as being caused by the overflow of the queue buffer, although it is not necessarily the new arrivals that are discarded.

and

$$q_K^+ = \sum_{i=0}^{K} q_i \sum_{j=K-i}^{\infty} p_j \tag{6.14}$$

Similarly, the *service relation* is described by

$$q_n = \begin{cases} q_0^+ + q_1^+ & \text{for } n = 0 \\ 0 & \text{for } n = K \\ q_{n+1}^+ & \text{otherwise} \end{cases} \tag{6.15}$$

where $q_K = 0$ always because there can only be at most K queued arrivals prior to the service, which then removes a single arrival from the buffer.

The q_0^+ term is the probability that the queue is empty immediately prior to service. From equilibrium considerations we must have

$$q_0^+ = 1 - (1 - \xi_K)\,\lambda \tag{6.16}$$

where ξ_K is the loss probability from the queue when the buffer size is K, and λ is the average number of (attempted) arrivals per time slot, given by

$$\lambda = \sum_{k=0}^{\infty} k p_k \tag{6.17}$$

In the following theorem, we will prove that q_0^+ is a common factor for all of the q_n in the marginal arrival process case.

Theorem 6.1 *For a discrete-time deterministic service queueing system with buffer capacity K, subject to marginal arrivals defined entirely by the probability distribution $\mathbf{p} = \{p_k\}$, the stationary probability q_n that the queue population is n immediately after service can be written as*

$$q_n = a_n q_0^+ \tag{6.18}$$

for $n < K$, where q_0^+ is the probability that the queue will be empty immediately prior to service, and a_n is a function of n and of the arrival process probabilities only (i.e. a_n is independent of K).

Proof. *Assume initially that every q_n for $n < T$ for some $T < K$ can be written in the form of equation (6.18). For $T < K$ we have from the arrivals relation of equation (6.13) that*

$$q_T^+ = \sum_{i=0}^{T} q_i p_{T-i} = q_T p_0 + q_0^+ \sum_{i=0}^{T-1} a_i p_{T-i} \tag{6.19}$$

which on rearranging gives

$$q_T = \frac{1}{p_0}\left(q_T^+ - q_0^+ \sum_{i=0}^{T-1} a_i p_{T-i}\right) \tag{6.20}$$

From equation (6.15) however, $q_T^+ = q_{T-1} = a_{T-1}q_0^+$ *resulting in*

$$q_T = \frac{1}{p_0}\left(a_{T-1} - \sum_{i=0}^{T-1} a_i p_{T-i}\right) q_0^+ \tag{6.21}$$

which shows that the initial assumption also holds for $n = T$.

From equation (6.13) it is obvious that $q_0 = q_0^+/p_0$, *which satisfies equation (6.18). Thus, from the above reasoning and the process of mathematical induction we know that equation (6.18) must be satisfied for every* q_n. ■

Theorem 6.1 is an important result because it provides a means whereby knowledge of the q_n and loss probabilities for some particular K allows the distribution of the queue population and the loss probability for any smaller K to be obtained, since the a_n quantities do not change. As an example, let q_n^* denote the steady state probability that the queue population is n for an infinite buffer, so that

$$a_n = \frac{q_n^*}{1-\lambda} \tag{6.22}$$

For a finite buffer, using $q_0^+ = 1 - (1 - \xi_K)\,\lambda$, consider that

$$\begin{aligned}
\sum_{n=0}^{K-1} q_n &= q_0^+ \sum_{n=0}^{K-1} a_n \\
&= \frac{1-(1-\xi_K)\,\lambda}{1-\lambda} \sum_{n=0}^{K-1} q_n^* \\
&= \left(1 + \frac{\lambda\xi_K}{1-\lambda}\right)(1 - t_K)
\end{aligned} \tag{6.23}$$

where t_n denotes the is the probability that the queue population of the infinite buffer problem is greater than n (the tail distribution). Then, since the above sum, being over all the non-zero probabilities of the finite buffer problem, must equal one, we obtain

$$\xi_K = \frac{(1-\lambda)\,t_K}{(1-t_K)\,\lambda} \tag{6.24}$$

which allows us to determine the finite buffer loss probability (or alternatively the finite buffer queue population distribution) from knowledge of the infinite buffer queue population distribution. This approach will apply equally well to a queue distribution obtained from a finite buffer analysis, provided that the length of the buffer used is greater than that of the desired K.

Note that a commonly accepted solution for the average loss probability is to just use $\xi_K = t_K$ [22, 66, 117, 121, 138]. That is, ξ_K is approximated by the proportion of the infinite buffer queue population exceeding the finite buffer size. Obviously this will result in a fairly poor approximation on the basis of equation (6.24) except when $\lambda = 0.5$.

Use of the infinite buffer as a source of queue population probabilities in equation (6.24) above suggests that the geometric tail approximation might be used instead of the actual values of t_K. The approximate loss probability for the finite buffer would then be

$$\tilde{\xi}_K = \frac{(1-\lambda)\,\psi\phi^K}{\left(1-\psi\phi^K\right)\lambda} \tag{6.25}$$

where the ψ and ϕ values are the parameters of the geometric tail approximation described by equation (6.2).

The average and variance of the finite buffer queue population, based on this geometric tail approximation, can also be found using this approach, and are given by

$$\tilde{L}_q = \frac{\psi\left(1 - K\phi^{K-1} - \phi^K + K\phi^K\right)}{(1-\phi)\left(1-\psi\phi^{K-1}\right)} \tag{6.26}$$

and

$$\begin{aligned}\operatorname{Var}\left[\tilde{L}_q\right] &= \frac{\psi\left(1 + \phi - K^2\phi^{K-1} - (K-1)^2\,\phi^{K+1}\right)}{(1-\phi)^2\left(1-\psi\phi^{K-1}\right)} \\ &+ \frac{\psi\left(2K^2 - 2K - 1\right)\phi^K}{(1-\phi)^2\left(1-\psi\phi^{K-1}\right)} - \tilde{L}_q^2 \end{aligned} \tag{6.27}$$

respectively. In the limit as $K \to \infty$, these become equal to the equations (6.7) and (6.8). The accuracy of equations (6.26) and (6.27) is of course dependent on how well the actual tail distribution is described by the geometric tail approximation.

6.2.2 Predicting Average Loss Probabilities for Autocorrelated Arrival Processes

Unfortunately, the approach used in the marginal arrivals case above does not lead to similar observations when the sources are described by autocorrelated arrival processes. To illustrate this point, consider a D-BMAP having m states. When the arrival process is in state i, it generates arrivals according to a probability distribution $\mathbf{P}_i = \{p_{k,i}\}$, where $p_{k,i}$ is the stationary probability that k arrivals will be generated by the D-BMAP whenever it is in state i. The state to state transition probabilities are represented by $\alpha_{i,j}$ which denotes the probability that the next D-BMAP state will be j given that the last state was i.

The time order of events in this queueing system is such that within each time slot, the D-BMAP changes state, arrivals are then generated using the parameters of the new D-BMAP state, and the queue then receives service. Define $q^+_{n,i}$ to be the probability that the queue population is n and the D-BMAP is in state i after the arrivals occur, and define $q_{n,i}$ to be the probability that the queue population is n and the D-BMAP is in state i immediately after the service at the end of the time slot. In addition, let q_n denote the probability that the queue population immediately after service is n, irrespective of the state of the D-BMAP, so that

$$q_n = \sum_{i=0}^{m-1} q_{n,i} \tag{6.28}$$

We will start by considering an infinite buffer in order to reduce the number of separate equations we have to deal with. For this system the arrival relationship of equation (6.13) now becomes

$$q_{n,i}^{+} = \sum_{j=0}^{n} s_{i,j} p_{n-i,j} \tag{6.29}$$

where $s_{n,j}$ represents the probability that the queue population is n and the next D-BMAP state will be j, given by

$$s_{n,j} = \sum_{i=0}^{m-1} \alpha_{i,j} q_{n,i} \tag{6.30}$$

where we also note that

$$q_n = \sum_{i=0}^{m-1} s_{n,i} \tag{6.31}$$

in addition to equation (6.28).

The service relationship of equation (6.15) is similarly redefined to be

$$q_{n,m} = \begin{cases} q_{0,m}^{+} + q_{1,m}^{+} & \text{for } n = 0 \\ q_{n+1,m}^{+} & \text{otherwise.} \end{cases} \tag{6.32}$$

Following the approach used above in the marginal case, and assuming that $p_{0,i}$ is strictly greater than zero for all i gives

$$s_{0,i} = \frac{q_{0,i}^{+}}{p_{0,i}} \tag{6.33}$$

and hence

$$q_0 = \sum_{i=0}^{m-1} \frac{q_{0,i}^{+}}{p_{0,i}} \tag{6.34}$$

From the service and arrivals relations of equations (6.29) and (6.32) we obtain

$$q_1 = \sum_{i=0}^{m-1} \frac{q_{0,i}}{p_{0,i}} - \sum_{i=0}^{m-1} \left(1 + \frac{p_{1,i}}{pi}\right) \frac{q_{0,i}^{+}}{p_{0,i}} \tag{6.35}$$

Unfortunately, we have no straight forward means to establish the $q_{0,i}$ terms individually — they can only be expressed as the solutions of m simultaneous equations relating them to the known $s_{0,i}$ values of equation (6.33). Hence, the q_n probabilities cannot be described directly in terms of the $q_{0,i}^{+}$ values and the arrival process probabilities as was the case for the marginal solution. In addition, the above reasoning assumes that each of the $p_{0,m}$ probabilities are greater than zero. This may not be the case in general, and in particular, is not the case when describing phase-geom binary processes, in which only one of the $p_{0,m}$ is actually non-zero.

Despite this, we will assume that Theorem 6.1 approximates the behaviour of the auto-correlated arrival process case well enough to use the average loss probability equations (6.24) and (6.25) developed in the marginal arrivals case. In the following we will investigate the accuracy of this assumption.

6.2.3 Accuracy Study for the Average Loss Probability Approximation

In the following we will study the accuracy of three approximations for the average loss probability compared with exact losses[2] observed from iterative solutions (see Appendix E). In order to obtain the average loss probability we divide the problem into two parts. The first involves approximating the tail distribution of the queue population, and the second then approximates the average loss from this tail distribution. Unfortunately, even if the tail distribution is perfectly matched by the geometric tail approximation, the correct loss result is not guaranteed (unless the arrival process is marginal). There will therefore be two sources of errors in the average loss probability approximation.

The three approximations under consideration are:

1. Xiong and Bruneel's [141] approximation for ψ along with the exact numerical derivation for ϕ. The actual approximation is given by equation (3.37) in section 3.3.

2. Approximations for ψ and ϕ obtained from L_q and $\mathrm{Var}\left[L_q\right]$ only as given by equations (6.7) and (6.8).

3. Numerical derivation for ϕ, with ψ approximated from ϕ, L_q, and $\mathrm{Var}\left[L_q\right]$ using equation (6.11).

These methods will be referred to as methods 1, 2, and 3 respectively in the following. Methods 2 and 3 rely on knowledge of the average and variance of the queue population

[2] As for the other accuracy studies in this thesis, the loss results from the iterative method, using selected examples, were compared with simulation ın order to confirm the correct ımplementation of the iterative solutıon.

for the queueing system. In this study these are obtained directly from the probability generating function analysis of the queueing system as described in Chapters 3 and 5. In practice, particularly when there are a large numbers of sources, these values could be obtained using a 2nd order approximation instead (see section 3.6).

A Few Examples

As a starting point we will consider the accuracy of the three approximations in terms of how well they describe the tail distribution of an example infinite buffer queueing system. The example system has 4 identical IBP sources with fixed parameters $\theta_i = 0.7$ and $\gamma_i = 0.5$. The average arrival rates from the four sources are given as $\lambda_i = \lambda/4$ where λ is the overall arrival rate, and takes on the values 0.6, 0.7, 0.8, and 0.9. Figure 6.1 shows the tail distribution probabilities as a function of the buffer position for the three approximations compared to the actual results.

Figure 6.2 shows the average loss probabilities calculated from the three approximations for t_K using equation (6.25). We see that although approximation methods 1 and 3 give almost exact results for the tail distribution in Figure 6.1, they both slightly underestimate the average loss probability. This implies that the actual or exact tail probabilities in this autocorrelated arrivals case would also underestimate the loss result when using equation (6.24). The difference between the two is quite small however, indicating that the assumption that Theorem 6.1 approximates the actual loss behaviour is acceptable.

As another example, Table 6.1 shows the parameters for another four IBP sources. Figure 6.3 shows the loss approximations resulting from the actual tail distribution, and the three tail approximations. We note that even the actual tail distribution fails to provide the correct loss probabilities (the relative error in this case being about 52.5%). In other words, this is one example where equation (6.24) does not hold particularly well for the autocorrelated arrivals case. Notice also how poor the second method is in approximating the geometric coefficient of the loss probability, even though the ϕ value in this case is very close to 1.

Source	λ_i	θ_i	γ_i
1	0.33	0.79	0.78
2	0.01	0.93	0.98
3	0.20	0.42	0.56
4	0.11	0.81	0.63

Table 6.1: *Geom-geom IBP source parameters for the example queueing system used to generate the results in Figure 6.3.*

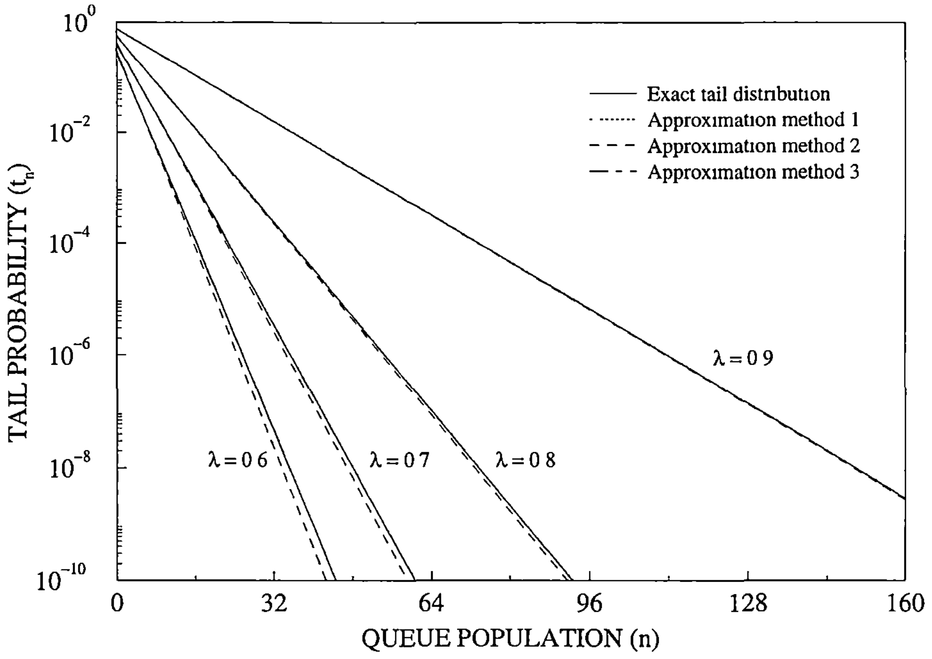

Figure 6.1: *Actual and approximate tail distributions for four average arrival rates using identical IBP sources. The three methods indicate the manner in which the parameters of the geometric tail distribution where obtained.*

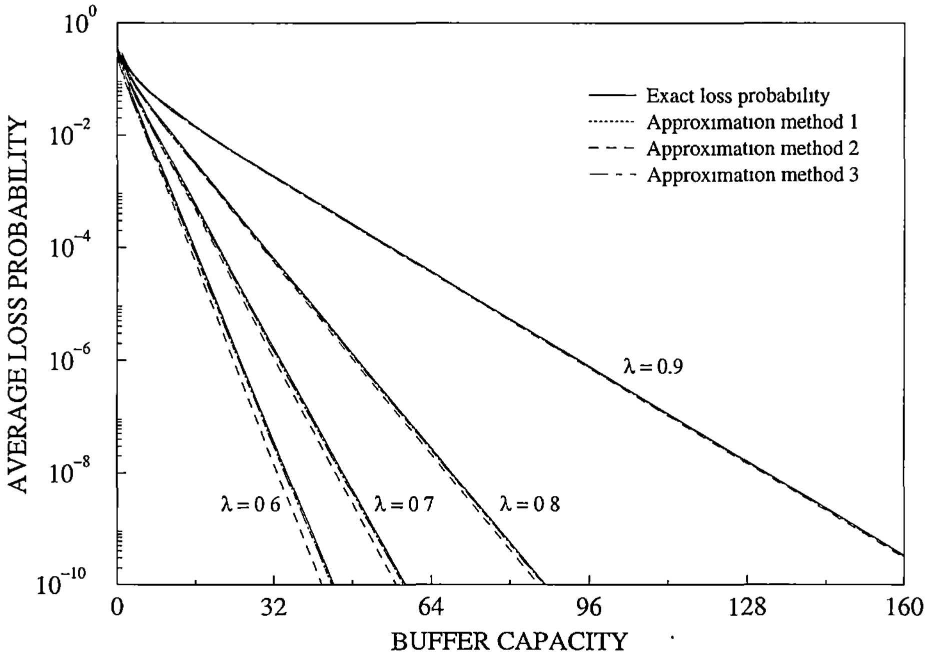

Figure 6.2: *Actual and approximate loss probabilities as a function of the buffer capacity. The loss estimations are performed using equation (6.25) with the tail approximations of Figure 6.1. Note that approximation methods* 1 *and* 3 *are so close to the actual tail distribution in this example that they are not really visible.*

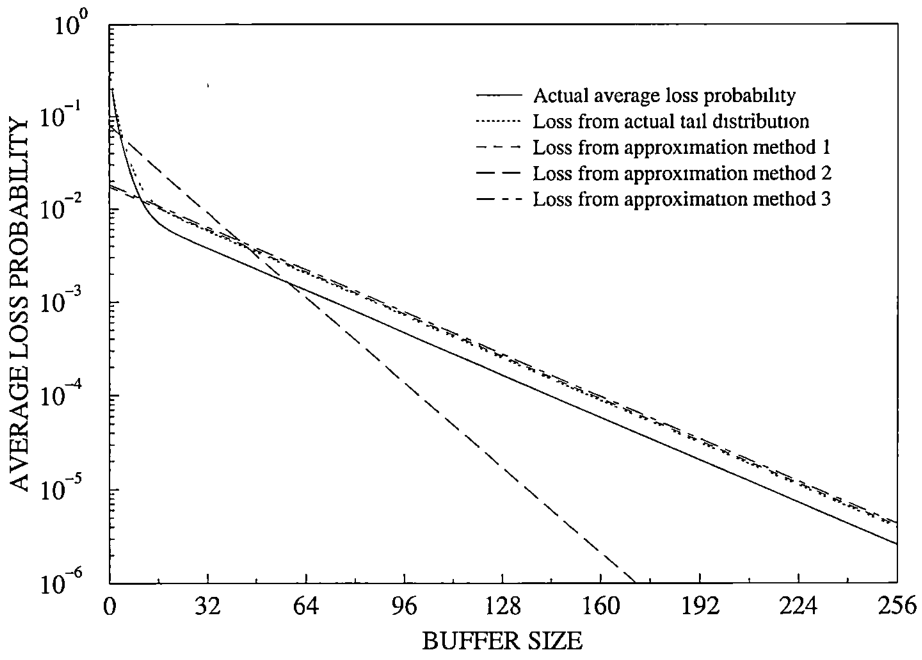

Figure 6.3: *Actual loss probabilities and estimated losses for the example queueing system with IBP source parameters described in Table 6.1.*

So far we have discussed only queueing systems fed by geom-geom IBP sources. As a last example we consider a queueing system fed by four identical IBP sources and a single cyclic source with parameters $b = 2$ and $C = 10$. As for the first example above, the IBP sources each have $\theta_i = 0.7$ and $\gamma_i = 0.5$, but in this case we use $\lambda_i = (\lambda - 0.2)/4$ where λ is the overall arrival rate to the queue from all five sources (with the cyclic source contributing a load of 0.2). The overall arrival rate takes on the value of 0.6, 0.7, 0.8, and 0.9 as before.

Figure 6.4 shows the tail distributions of the four example arrival rates, along with approximations from the second and third methods. We have not re-derived Xiong and Bruneel's result for IBP queueing systems with an additional cyclic source (although this should not be too difficult) so approximation method 1 is not used. Figure 6.5 shows the actual average loss probabilities, and the loss probabilities calculated from the two approximations for the tail distribution. As in Figure 6.2, the losses calculated from the two tail approximations slightly underestimate the actual loss probabilities. The closeness of the third tail approximation method to the actual tail distribution indicates that the difference in the loss probabilities is mostly due to the assumption that equation (6.24) applies to this queueing system.

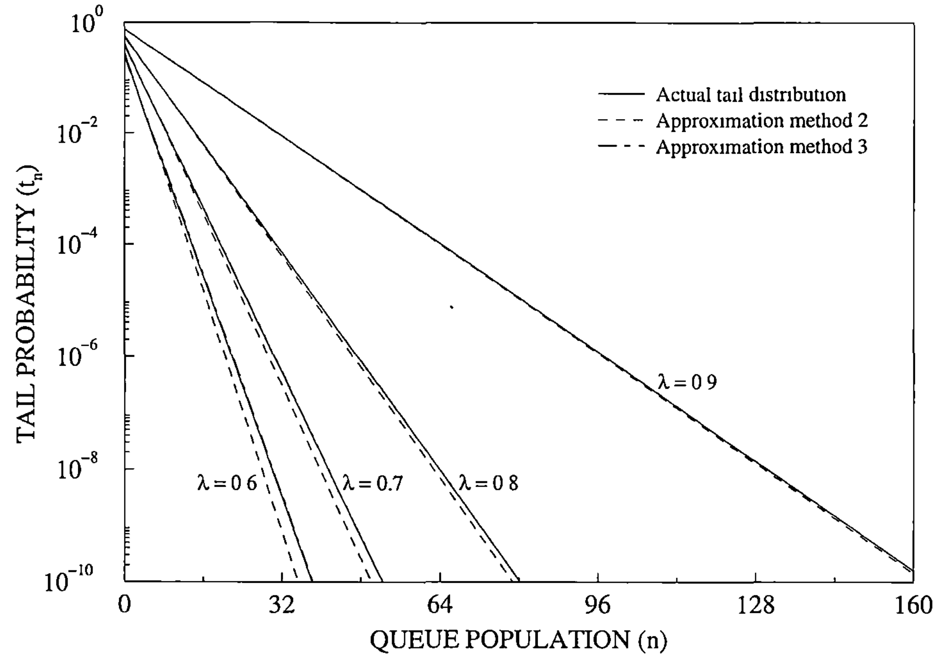

Figure 6.4: *Actual and approximate tail distributions for four average arrival rates using* 4 *identical IBP sources and a single cyclic source with* $b = 2$ *and* $C = 10$.

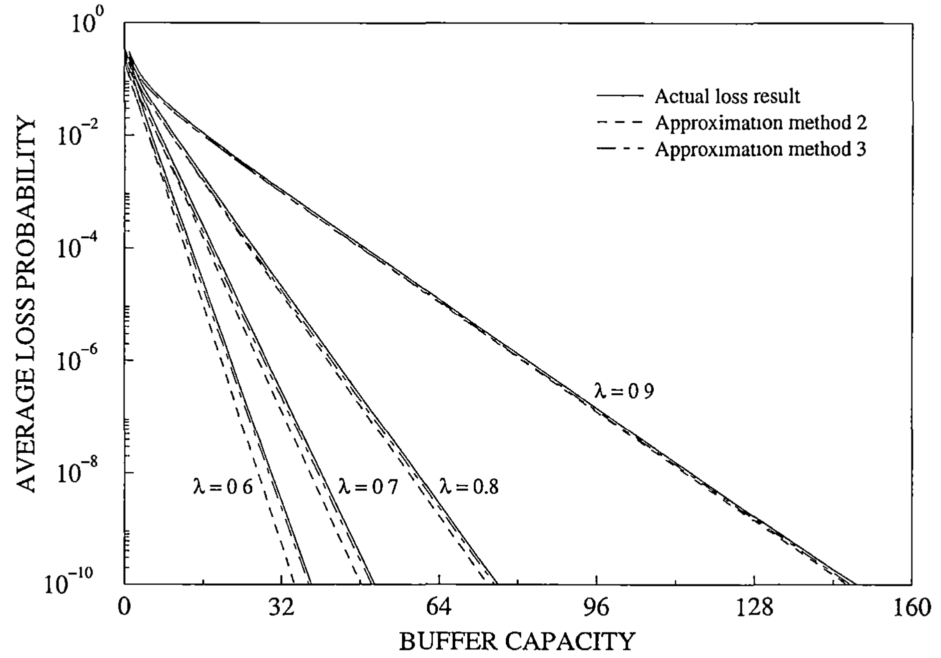

Figure 6.5: *Actual and approximate loss probabilities as a function of the buffer capacity for the queueing problem of Figure 6.4.*

A More Extensive Study

For a more extensive study of the accuracy of the loss estimation and the three methods for approximating the geometric tail parameters, 1000 queueing problems with randomly generated source parameters were generated for each of $N = 2$, 3, and 4 sources. To keep the buffer requirements of these problems large enough to study their loss behaviour accurately, the utilisation for each problem was chosen from the range $0.5 \leq \lambda \leq 0.9$, while the autocorrelation parameters were restricted to be in the range $0.5 \leq \gamma_i < 0.99$ for each source. As with the other studies using randomly generated sources, the average arrival rate θ_i during the active period of the ith source was chosen such that $\lambda_i < \theta_i \leq 1$.

We will not consider queueing systems with an additional cyclic source here since it is apparent that these systems behave similarly to IBP source only systems (see Figure 6.5). In addition, the run times for the iterative solution of these types of problems are considerable, which would restrict the analysis to the only the very simplest cases.

In order to assess the accuracy of the loss estimation we note that distribution of the average loss probability also appears geometric with the buffer capacity (see Figures 6.2, 6.3, and 6.5 for example). That is for buffer sizes above a certain value, the relative error between the actual loss probability and the estimated loss from equation (6.24) will be constant (the loss probabilities have a constant ratio). This same property will also hold true for equation (6.25) when using the first and third tail approximation methods. Since the second tail approximation method does not user the correct value of ϕ, it will see an increasing relative error in the loss estimation as the buffer capacity increases. For this reason we will not be considering this method in the following.

This observation of geometric behaviour provides a possible alternative to the use of importance sampling or large deviation theory in simulation studies involving very small loss probabilities [27, 31, 104]. Rather than perturbing the parameters of the input process in order to study the loss behaviour for short simulation runs, several simulations at different buffer sizes could be performed to obtain the parameters for the geometric approximation for the loss distribution. Another simple approach might be to obtain the probabilities for the infinite buffer tail distribution using simulation, and then obtain the loss behaviour in a manner similar to the one used here.

For this study we have chosen a buffer size of $K = 100$ as the capacity above which we expect the relative error in the loss estimation to remain constant. To insure accuracy in the calculation of the actual loss probabilities, convergence of the iterative solution required the relative change in the loss probability to be less than 10^{-8} over 10 iterations. In addition, the minimum valid loss probability was set to 10^{-10}. Average loss

N	Method	Mean	Deviation	1st Percentile	99th Percentile
2	1	85.7%	89%	1.6%	410%
	3	57.3%	51%	−0.9%	210%
3	1	66.7%	78%	0.5%	450%
	3	46.8%	44%	−3.1%	210%
4	1	61.7%	67%	−0.2%	340%
	3	51.5%	93%	−3.4%	290%

Table 6.2: *Statistics on the relative error in the average loss probability estimation for several queueing problems involving geom-geom IBP sources and two different approximation methods.*

probabilities below this value were taken to be zero, which meant they were ignored for the purpose of calculating relative errors.

Table 6.2 shows the statistics of the relative error in the loss estimation based on the two methods for calculating the parameters of the tail distribution. The relative errors are calculated as described in section 3.2. Note that there were not the full number of results available for each N due to the aforementioned lower limit on the calculation of the actual loss probabilities. The statistics are actually calculated from 295 results for $N = 2$, 516 results for $N = 3$, and 642 results for $N = 4$.

Overall, the third method has the best error performance, although the margin over Xiong and Bruneel's method is not particularly large. From the point of view of computational complexity this result favours Xiong and Bruneel's method since this first method does not require extensive calculations and can be performed very quickly, while the third method relies on knowledge of the queue population average and variance. It also appears that the accuracy of the first method improves as N increases, although the small number of results and the small change in the statistics makes a definite conclusion difficult.

Figure 6.6 shows how the relative error in the loss estimation varies with the average arrival rate to the queue for $N = 2$ using source parameters generated randomly under the same constraints used to obtain the error statistics of Table 6.2. The first approximation method (Xiong and Bruneel's) was used to provide the tail distribution data for calculating the loss estimate. These results were obtained from the analysis of $30,000$ queueing problems, with only some $8,800$ of these resulting in average loss probabilities of greater than 10^{-10} for the buffer capacity of 100 cells. Note that the mean relative error observed for this larger set of results for $N = 2$ was 86.6% for approximation method one, and 57.3% for method three, with corresponding standard deviations of 100% and 54% respectively. These agree very well with the results for the smaller data

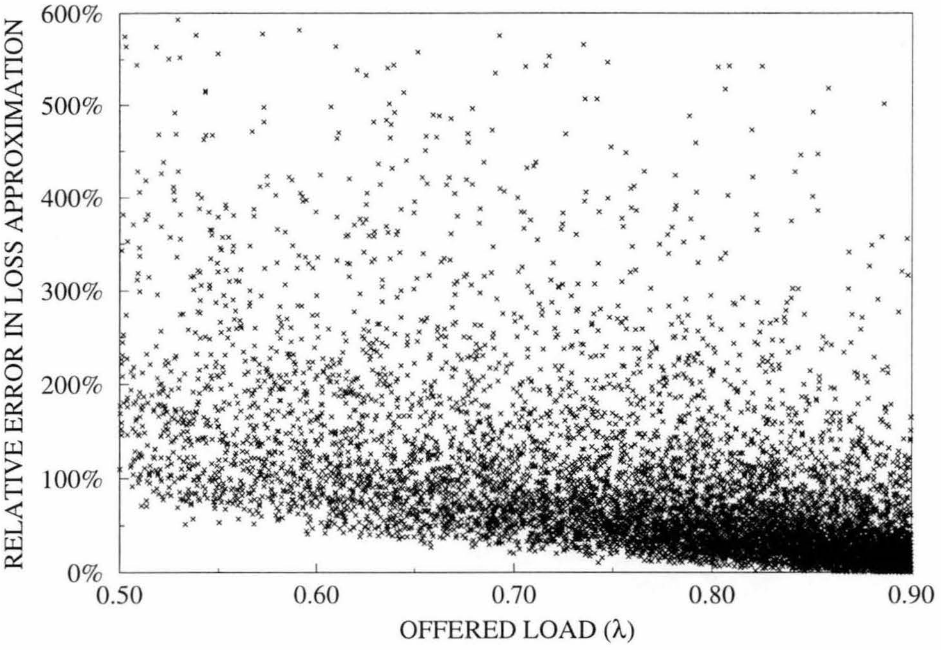

Figure 6.6: *Scatter plot of the relative error in the estimated average loss probability for 2 IBP sources using the first approximation method for the tail distribution. The offered load is restricted to between* 0.5 *and* 0.9*, while the individual source autocorrelations are in the range of* 0.5 *to* 0.99.

set reported in Table 6.2.

In the majority of the results, the approximate loss probability exceeds the actual value (although not by too much), which is a desirable feature of the approach. The clustering of the results at the higher utilisations is an artifact of the buffer size and lower probability limit chosen for the study — higher average arrival rates generally mean higher loss rates for a fixed buffer size. The results also indicate that the approximation will perform worse for lower queue utilisations, although still as an overestimation.

Another Accuracy Measure

An alternative measure of the accuracy of the loss estimation is to consider the error in predicting a buffer capacity to meet some required average loss probability. If ξ denotes the desired loss value, then equation (6.25) can be easily rearranged to provide an estimate for the buffer capacity $\widetilde{K}$ required to meet this loss probability of

$$\widetilde{K} = \frac{\log\left[\lambda\xi\right] - \log\left[\psi\left(1 - \lambda + \lambda\xi\right)\right]}{\log\left[\phi\right]} \tag{6.36}$$

where the exact value of ϕ is used for the geometric tail approximation, along with some approximation for ψ. In general there will be some inaccuracy associated with this estimate due to the scaling error between the tail and loss probability distributions. (We are assuming that the desired loss falls within the geometric region of the loss distribution).

Let us assume that the loss probabilities are geometric in the vicinity of some buffer size T, and denote by ξ_T the actual loss probability and by $\tilde{\xi}_T$ the estimated loss probability at T using equation (6.24) or (6.25). Then the ratio of ξ_T to $\tilde{\xi}_T$ expressed as a power of ϕ is given by

$$\Delta_\xi = \frac{\log\left[\xi_T\right] - \log\left[\tilde{\xi}_T\right]}{\log\left[\phi\right]} \tag{6.37}$$

where $\xi_T = \phi^{\Delta_\xi}\tilde{\xi}_T$. From the geometric property and the order of the terms in equation (6.37) Δ_ξ is also the amount by which the buffer capacity predicted by equation (6.36) will exceed[3] the required buffer size for any $\widetilde{K}$ and $\widetilde{K} - \Delta_\xi$ that fall on the geometric part of the loss distribution.

Table 6.3 presents the statistics on Δ_ξ measured from the same results used to generate Table 6.2. We note that the buffer capacities predicted by equation (6.36) generally overestimate the required capacities. In those few cases where the buffer capacity is underestimated, the difference appears to be very small. These results are not unexpected when considering the results of Table 6.2, since overestimation of the loss for a particular buffer size translates directly to overestimation of the buffer capacity for a particular loss probability, and vice versa.

N	Method	Mean	Deviation	1st Percentile	99th Percentile
2	1	7.3	8.4	0.09	43
	3	5.7	6.8	−0.06	34
3	1	6.8	8.3	0.03	45
	3	5.4	6.6	−0.17	32
4	1	6.9	8.5	−0.01	45
	3	5.8	7.2	−0.16	37

Table 6.3: *Statistics on the amount by which the buffer size required to satisfy a particular average loss probability exceeds the actual required size (Δ_ξ). The units are the same as used to specify the buffer capacity (cells).*

As for the relative error results, the performance of the third approximation method is slightly better than the first, but considering the extra computational complexities involved, the gain does not seem worth the effort. We note also that there is no particular improvement with increasing N, although this was suggested from the results of Table 6.2. Further studies will be required to establish this trend more precisely.

[3] If Δ_K is negative, then the predicted buffer size is smaller than the required buffer size.

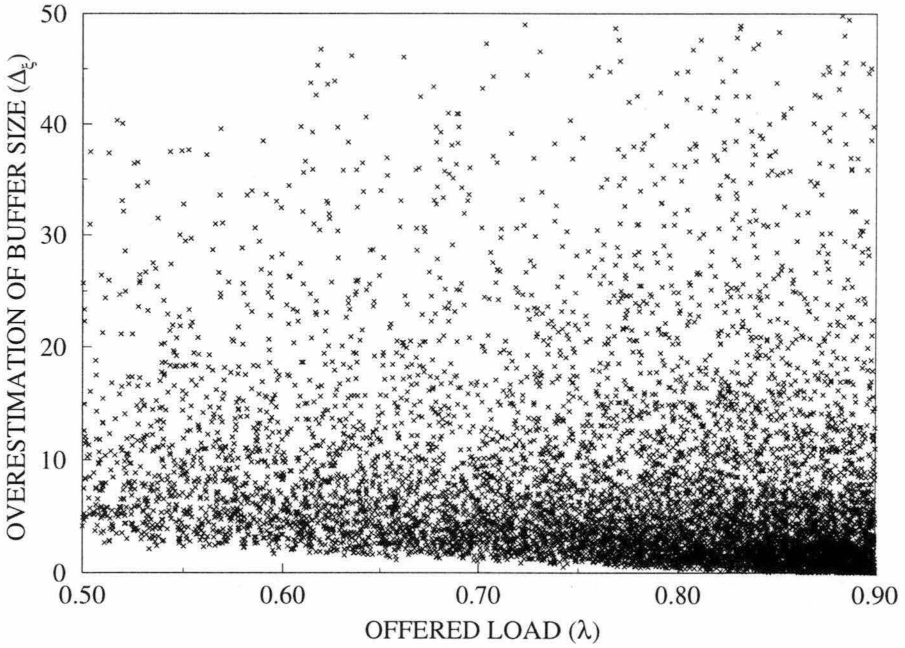

Figure 6.7: *Scatter plot of Δ_ξ against λ for 2 IBP sources using the first approximation method for the tail distribution. The loss results are the same as those used to generate Figure 6.6.*

Figure 6.7 shows a scatter plot of Δ_ξ calculated from the first approximation against the offered load to the queue for $N = 2$ using the same 8,800 exact loss results used to generate Figure 6.6. The mean buffer overshoot for this data set was 7.4 using the first method, and 5.8 for the third with corresponding standard deviations of 9.0 and 7.2 respectively. Again these agree quite well with the results from the smaller data set of only 295 loss results, but the difference is enough to confirm that large data sets are probably necessary to properly establish trends in the accuracy of the approximation methods as N increases.

We close this approximation study by noting that the results of Tables 6.2 and 6.3 are encouraging in that they show that the estimate for the average loss probability (or equivalently the required buffer size) discussed in this chapter is an acceptable one. In particular, the fact that this prediction tends to overestimate the correct solution means that decisions based on these results will err on the side of caution — that is they will generally provide better loss performance than is required.

6.2.4 Considerations for Dual Buffer Systems

An important consequence of Theorem 2.1 in Chapter 2 is that the total buffer space requirements for a queueing system are not affected by the priority scheme used to select which of the queued arrivals will be next to receive service, provided that the total buffer capacity can be fully utilised [35]. That is, as long as arrivals are not rejected from the system when it has available capacity (in any buffer).

In the implementation of a finite buffer queueing system (within an ATM switch for example) there are two basic methods that can be used to create logically separate buffers — the buffers may either be defined over physically separate areas of memory, or they may share the total memory space available. In the first of these approaches, when one queue fills its allocated memory space, any further arrivals to that buffer will be lost from the system, even though there may be capacity available elsewhere. In this case, the total buffer capacity (over all the logical buffers) is not being fully utilised, and we cannot use Theorem 2.1 to aid in predicting loss probabilities.

In the second approach, the buffers share the total memory space allocated to the server. Each logical buffer can grow until there is no capacity available anywhere, so that arrivals are only lost from the system when the total remaining capacity is zero. In this case, the total buffer capacity is fully utilised, and Theorem 2.1 implies that the loss probability from the queueing system will be independent of both the number of logical buffers, and the service orders of those buffers. The manner in which arrivals to a full system are rejected does not affect this result, only the proportion of the total loss shared by each type of arrival.

Thus, for a dual buffer (prioritised) system implemented with full utilisation of the combined memory space, the techniques discussed in this chapter for calculating the average loss probability can be applied to obtain the total loss probability. This simply involves the loss analysis of a single buffer queue subject to arrivals from both priority classes (using Xiong and Bruneel's approximation for the tail distribution say).

This total average loss probability ξ_{total} is related to the average loss probabilities for the high and low priority traffics by the relation

$$\lambda_{\text{low}}\xi_{\text{low}} + \lambda_{\text{high}}\xi_{\text{high}} = \lambda_{\text{total}}\xi_{\text{total}} \tag{6.38}$$

however in general we know no more than this. If the high priority buffer is implemented with head of line service priority (waiting high priority cells are served ahead of low priority cells) *and* we assume that the high priority traffic also has push out priority over the low priority traffic then ξ_{high} can be obtained by considering the high priority traffic on its own using the combined available buffer space. The resulting ξ_{low} is then obtained from equation (6.38).

Alternatively, if there is no push out priority at all, then the two traffics will have the same loss probability such that $\xi_{\text{low}} = \xi_{\text{high}} = \xi_{\text{total}}$. This is easily arrived at by considering that if there is no push out, losses will be independent of the priority of the arrivals, and hence each will see the same loss probability. Then, since the probability that the combined buffer is full is also independent of the arrival priorities, the result follows.

In practice, if we assign the high priority to delay sensitive traffic and the low priority to loss sensitive traffic, it is most likely that we would give the loss sensitive traffic push out priority as well, perhaps combined with a mechanism to guarantee a minimum number of buffer places to the delay traffic. If this is the case, or if no push out is implemented at all, then we cannot say what either ξ_{high} or ξ_{low} will be. We can determine the maximum average loss probability for each traffic class however, by assuming that the other class somehow manages to experience no loss. This then gives

$$\max\left[\xi_{\mathrm{low}}\right] = \frac{\lambda_{\mathrm{total}}}{\lambda_{\mathrm{low}}}\xi_{\mathrm{total}} \tag{6.39}$$

and

$$\max\left[\xi_{\mathrm{high}}\right] = \frac{\lambda_{\mathrm{total}}}{\lambda_{\mathrm{high}}}\xi_{\mathrm{total}} \tag{6.40}$$

If the average loss requirement for one of the traffic classes is known, then the required ξ_{total} can be calculated from equations (6.39) and (6.40) as necessary. When the loss requirements of both traffic classes are specified, then the smallest of the resulting two ξ_{total} values is used. If this combined average loss probability can be guaranteed, then both traffics must experience the required (or better) average loss.

6.3 Obtaining Delays from Populations for Uninterrupted Service Queues

In [125], Steyaert et al. derive a general relationship between the distribution for the queueing delay experienced by arrivals to a discrete-time multi-server FIFO queue and the distribution of the queue population. The arrival process is unconstrained in that it can be generally correlated, although the derivation is based on the assumption of an infinite buffer and a first-in, first-out service mechanism. The main result of interest here is that for a single server

$$d_n = \frac{q_n}{\lambda} \quad \text{for } n > 0 \tag{6.41}$$

where d_n is the steady state probability that the queueing delay is n service periods (time slots), q_n is the steady state probability that the queue population is n, and λ is the average number of arrivals occurring per time slot. It is interesting that statistics of the queueing delay can be obtained from the statistics of the queue population without explicit knowledge of the nature of the arrival process — a feature which holds even for the multi-server case.

From equation (6.41) we then obtain expressions for the average and variance of the queueing delay in terms of the equivalent measures of the queue population as

$$D_q = \frac{L_q}{\lambda} \tag{6.42}$$

which is of course Little's result, and

$$\operatorname{Var}[D_q] = \frac{\operatorname{Var}[L_q]}{\lambda} - \frac{(1-\lambda)\,L_q^2}{\lambda^2} \tag{6.43}$$

where we use D_q to indicate the average queueing delay, and L_q and $\operatorname{Var}[L_q]$ are the average and variance of the queue population.

6.3.1 Estimating the Higher Percentiles of the Delay

From the discussion at the beginning of this chapter we know that q_n is well described by a geometric distribution for large n. This same reasoning must therefore also apply to the distribution of the delay given the result of equation (6.41). This provides a simple means whereby the high percentiles of the queueing delay can be obtained using this geometric property. We will use the 99th percentile in the following, although others are straightforward to accommodate.

The 99th percentile of the queue population can be thought of as that number which the queue population rises above only 1% of the time. In terms of the tail distribution, the 99th percentile is given by the smallest n for which t_n (the probability that the queue population is strictly greater than n) is 0.01 or less. The 99th percentile of the queueing delay is therefore given by the smallest n for which t_n is less than or equal to $0.01/\lambda$.

The geometric tail approximation can be used in place of the actual tail distribution in order to obtain an estimate for the 99th percentile of the queueing delay of

$$\widetilde{n}_{99} = \frac{-2 - \log[\psi\lambda]}{\log[\phi]} \tag{6.44}$$

where $\widetilde{n}_{99}$ is the estimate of the relevant percentile value, and base 10 logarithms are assumed. The ψ and ϕ terms are the parameters of the geometric approximation to the tail probabilities.

Unfortunately, $\widetilde{n}_{99}$ may not always be large enough to fall on the part of t_n which obeys the geometric property (a problem which will be more common for lesser percentiles). However since the geometric assumption generally underestimates the probabilities when n is small, the $\widetilde{n}_{99}$ obtained from equation (6.44) will tend to overestimate the correct 99th percentile value. Thus as a general rule, the larger the value of $\widetilde{n}_{99}$ the more accurate the estimate will be.

6.3.2 Delays in Finite Buffers

One point apparently overlooked by the authors of [125] is that the assumptions under which equation (6.41) is obtained can be exploited in order to extend the result to finite buffer queues as well. Importantly, the derivation in [125] proceeds *without* requiring that the arrivals be independent of the queue population. Consequently, it may be assumed that a finite buffer can be described using an infinite buffer, but with an arrival process which limits the number of arrivals occurring in any time slot so as not to exceed some maximum level in the queue buffer. This arrival process still conforms to the requirements of [125], and hence equation (6.41) still holds, but in the modified form of

$$d_n = \frac{q_n}{(1 - \xi_K)\,\lambda} \quad \text{for } n > 0 \tag{6.45}$$

where ξ_K is the loss probability for a finite buffer with a capacity of K waiting cells, so that $(1 - \xi_K)\,\lambda$ represents the average number of successful arrivals to the queue per time slot.

Applying this result to equations (6.42) and (6.43) for the average and variance of the queueing delay now gives

$$D_q = \frac{L_q}{(1 - \xi_K)\,\lambda} \tag{6.46}$$

and

$$\operatorname{Var}[D_q] = \frac{\operatorname{Var}[L_q]}{(1 - \xi_K)\,\lambda} - \frac{(1 - \lambda + \lambda\xi_K)\,L_q^2}{(1 - \xi_K)^2\,\lambda^2} \tag{6.47}$$

where L_q and $\operatorname{Var}[L_q]$ are the average and variance of the queue population for the finite buffer.

To provide support for this conclusion, Table 6.4 presents results for the average and variance of the delay obtained from both numeric iteration and from simulation for an example queueing system using four identical IBP sources. The IBP sources have parameters given by $\lambda_i = 0.225$, $\theta_i = 0.7$, and $\gamma_i = 0.5$, providing an overall average arrival rate of $\lambda = 0.9$. Very small buffer sizes (K in Table 6.4) were used to provide high loss rates, and hence to make sure any deviations between the predicted and actual results were visible.

The average and variance of the queueing delay obtained by combining the population results of the iterative solution with equations (6.46) and (6.47) agree very well with the simulation results, which supports the reasoning used to develop these equations.

	Analytic			Simulation		
K	ξ_K	D_q	Var $[D_q]$	ξ_K	D_q	Var $[D_q]$
4	0.10572	1.4177	1.3586	0.10578	1.4179 ± 0.0012	1.3586
8	0.04761	2.8819	5.5604	0.04766	2.8821 ± 0.0037	5.5604
16	0.01427	4.9535	18.900	0.01429	4.9587 ± 0.0109	18.920
32	0.00185	6.8143	46.550	0.00186	6.8253 ± 0.0273	46.669

Table 6.4: *Comparison of theoretical and simulation queueing delays for an example finite buffer queueing system involving four identical IBP sources. The* 99% *confidence interval is specified for the average queueing delay obtained from simulation.*

6.3.3 Delays for Individual Traffic Classes

Obtaining queueing delays for an individual traffic class in a queueing system subject to arrivals from multiple classes of traffic is unfortunately not straightforward, and is very much dependent on service considerations. We will not go into this in detail here, but direct the reader to Appendix B, where a general approach and several examples are considered for infinite buffers without service priorities. The results suggest that there are definite limits to the 'best' and 'worst' average queueing delay that any traffic class can achieve, which has interesting impplications for the mixing of traffic in queueing systems.

In the following section we will briefly consider some of the difficulties associated with calculating queueing delays for queueing systems subject to service interruptions, and look at an example problem.

6.4 Delays for Interrupted Service Queues

In a discrete-time dual buffer priority system, where arrivals to one buffer have head-of-line (non pre-emptive) service priority over arrivals to the other buffer, the average and variance of the queueing delay for the high priority arrivals are easily found. Since this buffer has service priority, its queueing behaviour is unaffected by the presence or otherwise of low priority traffic. It can therefore be treated as if it receives uninterrupted service, and the results of the previous section applied to obtain the desired delay measures.

Unfortunately the situation is not so simple for the arrivals to the low priority buffer because it receives interrupted service due to the presence of high priority traffic, and consequently Steyaert et al.'s result does not apply. Although Little's result can still yield the average queueing delay from the average queue population (found using Corollary 2.2), no corresponding solutions have been found in the literature for the variance of the delay.

As we shall see below, the delay variance can be found fairly easily for the simplest case where both the arrivals and the service (or the interruptions) are purely random. This result, and the discussions relating to shared buffer delays in Appendix B, suggest that more general results for autocorrelated arrivals and/or service should be obtainable. This is a topic for further research.

6.4.1 Random Arrivals and Random Service

Here we assume that the arrival process is completely described by its marginal probability distribution, so that the probability of receiving k arrivals in the current time slot is independent of the arrival process at any previous time. Similarly, the probability that the queue receives a visit from the server in the current time slot is also independent of both previous service behaviour and previous arrival behaviour.

The time order of events within a time slot is assumed without loss of generality to be opportunity for service followed by acceptance of any new arrivals. The queue population is always observed immediately after the service opportunity. We assume that if the server visits the buffer and it is not empty, it removes a single queued cell. Delays are measured as the number of complete time slots that an arrival waits in the queue before receiving service, so that an arrival to an empty queue that receives service at the end of that time slot is considered to have zero queueing delay.

As usual, we will denote the average number of arrivals to the queue in one time slot by λ, with second and third moments denoted by M_2 and M_3. In addition, we denote the stationary probability that the queue receives service in any slot by f. Let c_n denote the probability that an arrival to the queue sees c_n cells queued ahead of it, and let d_m denote the probability that an arrival experiences a queueing delay of m slot times.

Consider an arrival that sees n cells queued ahead of itself. In order for it to reach the head of the queue, there must be n services. For the cell to then leave the queue (to be serviced itself) will then require a total of $n+1$ services. Since only whole slot times are considered for the queueing delay, if the arrival receives service in the $(m+1)$th time slot after it arrived, it is considered to only have a queueing delay of m slot times. Thus the probability that this cell has a queueing delay of m slot times is given by the probability that there are n services in the m time slots following its arrival, and there is a further service in the $(m+1)$th time slot. That is

$$\Pr(\text{delay} = m \,|\, n \text{ cells queue ahead}) = \begin{cases} 0 & \text{if } m < n \\ \binom{m}{n} (1-f)^{m-n} f^{n+1} & \text{otherwise} \end{cases} \tag{6.48}$$

giving the probability of a delay of m slot times as

$$d_m = \sum_{n=0}^{m} \binom{m}{n} (1-f)^{m-n} f^{n+1} c_n \tag{6.49}$$

with the first and second moments of the delay then given by

$$\begin{aligned} \sum_{n=0}^{\infty} n d_n &= \sum_{m=0}^{\infty} m \sum_{n=0}^{m} \binom{m}{n} (1-f)^{m-n} f^{n+1} c_n \\ &= \sum_{n=0}^{\infty} c_n f^{n+1} \sum_{i=0}^{\infty} (n+i) \binom{i+n}{n} (1-f)^i \\ &= \frac{1}{f} \left(\sum_{n=0}^{\infty} n c_n + 1 - f \right) \end{aligned} \tag{6.50}$$

and

$$\begin{aligned} \sum_{n=0}^{\infty} n^2 d_n &= \sum_{m=0}^{\infty} m^2 \sum_{n=0}^{m} \binom{m}{n} (1-f)^{m-n} f^{n+1} c_n \\ &= \sum_{n=0}^{\infty} c_n f^{n+1} \sum_{i=0}^{\infty} (n+i)^2 \binom{i+n}{n} (1-f)^i \\ &= \frac{1}{f^2} \left(\sum_{n=0}^{\infty} n^2 c_n + 3(1-f) \sum_{n=0}^{\infty} n c_n + (1-f)(2-f) \right) \end{aligned} \tag{6.51}$$

respectively, where we have made use of Theorems F.2, F.5, and F.6. Thus we have derived the moments of the delay in terms of the probability that an arrival is queued behind so many previous arrivals.

Assume that the buffer contains r cells waiting for service at the instant that the current arrivals are admitted to the queue. Let k be the number of new arrivals, and let s denote the position of any particular arrival within these k, where $s = 0, 1, \ldots, k-1$. The probability that an arrival sees $s + r$ cells queued ahead of is then given by

$$\Pr(s + r \text{ cells ahead}) = \frac{1}{\lambda} \sum_{k=s+1}^{\infty} p_k \tag{6.52}$$

where p_k denotes the probability of there being k arrivals in any time slot. From this we obtain

$$\begin{aligned} c_n &= \frac{1}{\lambda} \sum_{i=0}^{n} \sum_{k=n+1-i}^{\infty} q_i p_k \\ &= \frac{1}{\lambda} \sum_{i=0}^{n} (q_i - q_i') \end{aligned} \tag{6.53}$$

where q_i' denotes the probability that the queue population is i after all the arrivals have been accepted, which is given by

$$q_i' = \sum_{j=0}^{i} q_j p_{i-j} \tag{6.54}$$

For $n > 0$ we can rewrite c_n in terms of c_{n-1} as

$$c_n = c_{n-1} + \frac{1}{\lambda}\left(q_n - q_n'\right) \tag{6.55}$$

In z-transform notation, this becomes

$$c(z) = \frac{1 - p(z)}{\lambda(1-z)} q(z) \tag{6.56}$$

where $c(z)$, $q(z)$, and $p(z)$ are the z-transforms of the c_n, q_n, and p_n probability distributions respectively, and where we note that the post-arrival distribution described by q_n' is given by $q(z)p(z)$. From the second and third derivatives of equation (6.56) we then obtain

$$\sum_{n=0}^{\infty} nc_n = L_q + \frac{M_2 - \lambda}{2\lambda} \tag{6.57}$$

and

$$\sum_{n=0}^{\infty} n^2 c_n = V_q + L_q^2 + \left(\frac{M_2 - \lambda}{\lambda}\right) L_q + \frac{2M_3 - 3M_2 + \lambda}{6\lambda} \tag{6.58}$$

Combining these results for the first two moments of the c_n distribution with equations (6.50) and (6.51) then gives

$$D_q = \frac{1}{f}\left(L_q + \frac{M_2 - \lambda}{2\lambda} + 1 - f\right)$$

and

$$\begin{aligned} \operatorname{Var}[D_q] &= \frac{1}{f^2} V_q + \left(\frac{M_2 + 3\lambda - 4\lambda f}{2\lambda f^2}\right) L_q + \frac{(1-f)(M_2 - \lambda)}{2\lambda f^2} \\ &- \frac{(M_2 - \lambda)^2}{4\lambda^2 f^2} + \frac{2M_3 - 3M_2 + \lambda}{6\lambda f^2} \end{aligned} \tag{6.59}$$

Substituting for the average and variance of the queue population, given by equations (2.66) and (2.69) using $\gamma = 0$, finally yields

$$D_q = \frac{M_2 + \lambda - 2\lambda f}{2\lambda(f - \lambda)} \tag{6.60}$$

and

$$\begin{aligned} \operatorname{Var}[D_q] &= \frac{4\lambda f M_3 - 3(f - \lambda) M_2^2 + 6\lambda f(1-f) M_2 + 2f\lambda^2(1 + 3f - 6f^2)}{12 f^2 \lambda^2 (f - \lambda)} \\ &+ \frac{(M_2 + f - 2f^2)^2}{4f^2(f - \lambda)^2} \end{aligned} \tag{6.61}$$

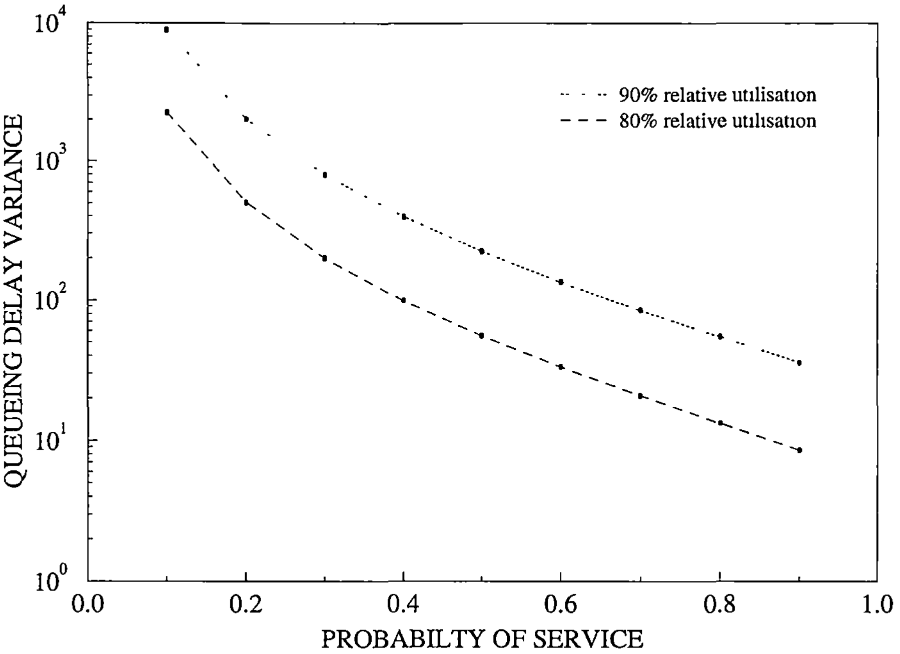

Figure 6.8: *Variance of the queueing delay for Poisson arrivals to a queue with random service. The analytic results are presented as dashed lines, while the vertical bars indicate the error range for the results obtain from simulation.*

To confirm this analysis, Figure 6.8 presents the variance of the queueing delay for an example problem having Poisson arrivals and random service, obtained from both the analysis and simulation. Relative utilisations of 90% and 80% were considered and the results plotted against the probability of service f. The simulation errors are indicated by vertical lines at each sample point and correspond to a 99% confidence interval of ±0.5% for the average queue population. The results obviously support the accuracy of the analysis.

6.5 Summary

We began this chapter with a discussion of the geometric tail approximation for an infinite buffer queueing system, arriving at three methods for calculating the parameters of this curve. Since one of the primary measures of finite buffer performance is the average loss probability we then looked at how this measure might be predicted from knowledge of the tail distribution. For a queue subject to marginally distributed arrivals (no autocorrelation) we proved a simple relation between the tail probabilities and the average loss probability. The solution shows that the commonly accepted approach, where the average loss is approximated by the probability that the infinite buffer population exceeds the finite buffer size, can result in a poor estimation.

Unfortunately this relationship between the loss and the tail distribution does not apply for queues subject to arrivals from autocorrelated sources. However, under the assumption that the relationship applies sufficiently well, the loss prediction accuracy of the three tail approximation methods was investigated. Consideration of three example problems suggested that the average loss probability also exhibits geometric behaviour as the buffer capacity increases. As expected for the marginal arrivals analysis, the geometric decay coefficient of the average loss was found to be equal to that of the population tail. A suitable method for examining the accuracy of the loss predictions was proposed using this observation, and a more exhaustive study using randomly generated sources was performed.

The inexact nature of the geometric decay coefficient calculated by the second method excluded it from this part of the investigation, but results for the other two methods are encouraging. While both usually overestimate the average loss, the magnitude of the relative error is not high. In terms of estimating buffer sizes for a specified loss probability, the performance is similar with again only a relatively small mean overestimation observed. The accuracy of the first method, based on previous work by Xiong and Bruneel [141], is encouraging in particular because the computational complexity of the solution is small. Further work is required however to extend this approximation to include other arrival processes apart from geom-geom IBP sources.

In addition to the loss behaviour, we have also considered the relationship between queueing delays and queue populations in this chapter. For queues subject to uninterrupted service, the average and variance of the queueing delay can be expressed simply in terms of the average and variance of the queue population, the average arrival rate, and for finite buffers, the average loss probability. The geometric tail property can also be applied to obtain an estimate for the 99th percentile of the queueing delay. Queueing delays for individual traffic classes are not considered here, although Appendix B discusses some methods for calculating these results, and provides a few examples.

The use of dual buffer queueing systems (where head of line priority is given to waiting cells in one buffer) complicates the calculation of both finite buffer losses, and queueing delays. For a dual buffer system, the overall average loss probability can be obtained from its single buffer equivalent in the case where the total buffer capacity is shared by both buffers (logically separate buffers rather than physically separate) and rough upper limits can then be placed on the individual buffer loss probabilities. Similarly, average queueing delays can be obtained for the dual buffer system using Little's result and Corollary 2.2 of Theorem 2.1, but higher moments of the delay for the lower priority queue are more complex. A relatively simple example was investigated to illustrate this point.

Chapter 7

Applications

In this chapter we will briefly describe some of the applications of the results presented in the rest of this thesis. In particular we will look at the analysis of a dual buffer priority queueing system in section 7.1, and identify the problem of correlation between sources for class or priority based traffic models. Simulation is used in section 7.2 to provide some insight into this problem for a dual buffer system when the high priority traffic class is described by a cyclic (or periodic) process. When the high priority process is a subclass of the overall arrival process, the correlation effect does not cause modelling problems. We look at an example scheme using random priority assignment for geom-geom IBP sources in section 7.3.

Another aspect of the queueing process that has not yet been considered here is the identification or modelling of the output process of the queue. This characterises the merging process, and in section 7.4 we will look at how the parameters of an output model may be fitted using the population analysis. As an illustrative example, the output of a queue subject to arrivals from geom-geom IBP sources is modelled by a geom-geom IBP process in section 7.5. Finally in section 7.6 we show that the output of a queueing system with either marginal, or phase-geom binary arrivals, is a phase-geom binary process, and determine the moments of the busy and idle periods of this process.

7.1 Analysis of a Dual Buffer Queueing System

Consider the dual buffer queueing system in Figure 7.1, which represents a single output port in a non-blocking ATM switching element. The switch is assumed to have equal speed input and output lines, and a discrete-time analysis is assumed, with one time interval (or slot) equal to the time required to transmit an ATM cell on the output line. Each of the N sources in Figure 7.1 represent a possible source of arrivals from one of the N input ports of the switch after routing (splitting). Since the input lines are assumed to be the same speed as the output line, each source can generate at most one arrival in any time slot. The generated arrivals are then queued in either the high priority buffer (for delay sensitive traffic) or the low priority buffer (for loss sensitive traffic).

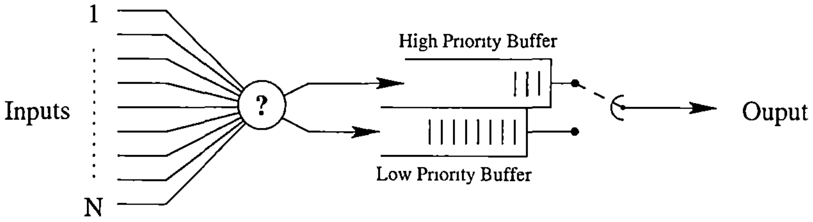

Figure 7.1: *A representation of one of the dual buffer queueing systems present at each of the outputs of an output queueing, non-blocking ATM switch. The question mark represents the priority decision process which is discussed in the text.*

From an analysis point of view, the selection between high or low priority might be regarded as a second splitting process based on priority, so that the arrival processes to the two buffers form subclasses of the arrival processes represented by the N sources. The simplest method, used for example in [10, 12, 39, 88, 127, 144] is to assume that arrivals from source i have high priority with probability h_i and low priority with probability $(1 - h_i)$, independent of the probabilities assigned at other times. The alternative is to assume that the priority selection exhibits some form of autocorrelation, such as in the three-state Markov priority arrival process of [16, 68].

Another approach that can be taken to the analysis of the dual buffer queueing system is to use separate models for the arrival processes to the high and low priority buffers, based on some characteristic of the traffics. For example, in Chapter 1 we proposed that low priority traffic could be described by the geom-geom IBP process, while the high priority processes might be better described by a periodic or cyclic process. In the literature there are similar examples where the arrival processes used in the queueing analysis are based on the particular traffic types being considered — voice sources are modelled as two-state Markov processes [55, 76, 81, 136], video sources are modelled by multi-state Markov processes [76, 136] or more complex models [55], and generic data sources are modelled by the ever present Poisson process [55, 81, 106, 136].

The problem with this approach is that in a real network switching element, the arrival processes on each input line will not be exclusively from a single traffic class (except possibly at the entry points to the network). That is, each of the switch input lines may generate arrivals for several classes of traffic, and since each input line can only generate one arrival per time slot, the arrivals for each class will be *negatively* correlated with the arrivals of other classes. Consequently, the 'traditional' queueing analysis methods (as used in the above references for example) which assume that the sources belonging to the various traffic classes are independent, will tend to overestimate the queue population average and variance.

In regard to the dual buffer problem, the priority splitting approach behaves more realistically in the sense that the arrivals to the two queue buffers are in fact anticorrelated (a high priority arrival from one source means that it did not generate a low priority arrival in the same time slot, and vice versa). We will consider the priority splitting approach in more detail in section 7.3. In the following section we will look at the dual buffer queueing system when the high priority traffic is described by a cyclic arrival process.

7.2 Dual Buffer Analysis with a Cyclic Arrival Process

In Chapter 1 we explained the fact that delay sensitive (high priority) traffic might have long burst periods or holding times, and therefore its periodic nature would not be as distorted by the succession of network queues as other traffic types. As a consequence the cyclic arrival process was suggested to describe the service interruption process of the low priority buffer. If such a model is to be used however, we need to consider how the low priority traffic will be affected. As we have discussed in the above paragraph, the simple geom-geom IBP model considered in Chapter 5 will not accurately describe the low priority arrival process, since correlation with the high priority traffic was not considered.

We are interested in the effect that ignoring this correlation has on the accuracy of the queue behaviour. To do this, the analytical results of Chapter 5 need to be compared with a 'correct' solution — one in which the high and low priority arrival processes exhibit the correct anticorrelation behaviour. Since no suitable analytical models are available that use such a mixture of periodic and autocorrelated random processes, a simulation study has been performed. In the following we will refer to the queueing system wherein each source is modelled independently as the *independent queueing model*, and the queueing system wherein the low priority arrivals are correlated with the high priority arrivals as the *anticorrelated queueing model.*

7.2.1 A Simulation Study

The low priority arrival process needs to be as close to the independent geom-geom IBP model as possible in order for the analytical results to have any application to this queueing problem. To preserve this behaviour, it was decided that the b high priority arrivals every C slot times from the cyclic source were to be 'inserted' into the low priority arrival processes from selected sources. Any low priority arrival that would normally have occurred in the same time slot for the source would be delayed (along with subsequent arrivals as required) for transmission in subsequent slots. This insertion process means that the average arrival statistics of the low priority traffic are unaffected by the high priority arrivals, and during the silent periods of the cyclic source, the geom-geom IBP behaviour is largely unaffected.

As an additional constraint, the high priority arrivals were required to cause b consecutive service interruptions to the low priority buffer, followed by $C - b$ services, as used in the theory of Chapter 5. Three arrangements for the insertion of the high priority arrivals were considered:

1. The b cyclic arrivals come from just one source.
2. The b cyclic arrivals come from b different sources in a consecutive fashion.
3. The b cyclic arrivals come simultaneously from b different sources.

Note that while the first two of these arrangements result in zero queueing delays for the high priority traffic (only one arrival per time slot) high priority arrivals in the third case will experience queueing delays. The theoretical treatment of the independent queueing model in Chapter 5 assumes that the b arrivals are all consecutive, and hence provides the low priority queue population only (using Corollary 2.3). By assuming that $b < C$, the maximum delay any high priority arrival will experience will be bounded simply by b. If a more exact treatment is necessary, the high priority queue can be analysed separately using one of many available methods (see for example [23] and the references therein).

Figures 7.2 and 7.3 show the average and variance respectively of the queue population for the low priority buffer using these three arrangements (referred to as simulation methods 1 to 3) and the theoretical solution of the independent queueing model. The results are for $b = 3$ and $C = 10$, using 4 identical geom-geom IBP sources with parameters $\lambda_i = \lambda/4$, $\theta_i = 0.4$, and $\gamma_i = 0.7$, where λ is the independent variable. The simulation results were obtained using the method of batch means [73] to establish simulation confidence. Using batches equal to observations of 10^6 service periods, all simulation results are accurate to within ±0.5% for the average queue population, with 99% confidence.

In order to make the overestimation of the analytical method clearer, Figures 7.4 and 7.5 show the percentage error[1] in the theoretical results, measured relative to the three simulation results. Although the error appears quite high when the arrival rate from the IBP sources is low, it should be noted that this corresponds to quite small queue populations, as indicated by Figures 7.2 and 7.3.

Note that the percentage error in the theoretical result decreases as the average arrival rate to the queue increases. The reason for this is that the impact of short term differences in the arrival patterns of the two queueing models is less pronounced when the queue populations are large. Since the higher average arrival rate of the IBP sources yields larger queue populations, the decrease in the percentage error between the two models is therefore not unexpected. This indicates that the upper bounds on the average and variance of the queue population represented by the independent queueing model become tighter as the magnitude of the results increase.

We also expect the error in the independent queueing model to decrease as the number of sources increases. The reason for this is that as the number of sources increases, the probability of an arrival being generated by any one particular source decreases. This reduces the probability that the independent queueing model generates more arrivals in one time slot than its anticorrelated counterpart, and hence the arrival processes of the two models become more and more similar. Figure 7.6 shows the change in the percentage error for the average queue population of the low priority buffer as the number of IBP sources (or number of input lines) increases. Using $b = 3$ and $C = 10$, the IBP sources are all identical, with parameters $\lambda_i = 0.35/N$, $\theta_i = 0.4$, and $\gamma_i = 0.7$.

Lastly, to investigate the effect of changing the cyclic source parameters, simulation method 1 was repeated for $b = 1$, 2, 5, and 7 for the same IBP parameters as used to generate Figures 7.2 and 7.3. The percentage error in the average queue population of the low priority buffer for the three values of b is shown in Figure 7.7. Since the possible range of IBP arrival rates varies between the five results, the average arrival rates used to obtain the simulation results have all been normalised to the available low priority capacity $(1 - b/C)$ in order to show all the results on the same graph.

Note that as b increases, the error curve approaches a straight line result with a slope

[1] We will use the term 'percentage error' here instead of 'relative error' as used in the rest of this thesis because we are measuring the performance of the theoretical result relative to the simulation result, which is inherently inaccurate. We will reserve the 'relative error' term for comparisons with exact measures.

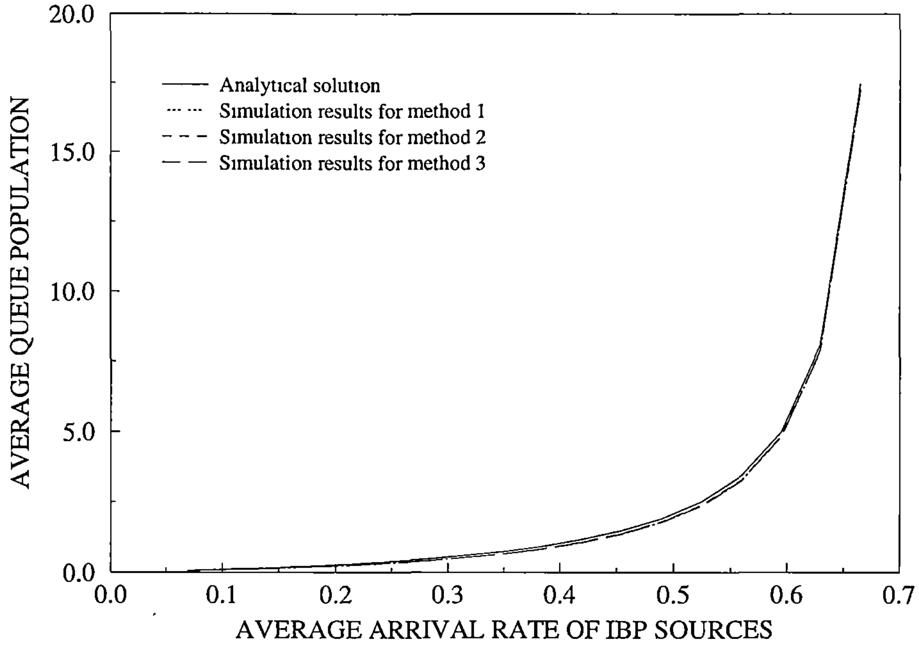

Figure 7.2: *Simulation and theoretical results for the average queue population of the low priority buffer. The cyclic source has parameters $b = 3$ and $C = 10$, while the low priority arrival process is made up of four identical IBP sources with parameters $\lambda_i = 0.25\lambda$, $\theta_i = 0.4$, and $\gamma_i = 0.7$.*

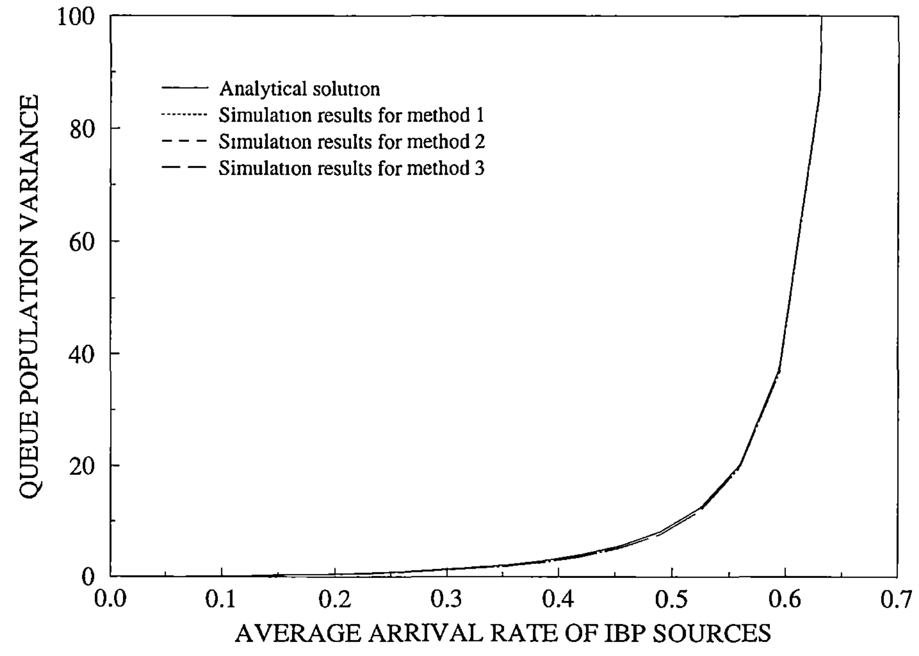

Figure 7.3: *Simulation and theoretical variance results corresponding to Figure 7.2.*

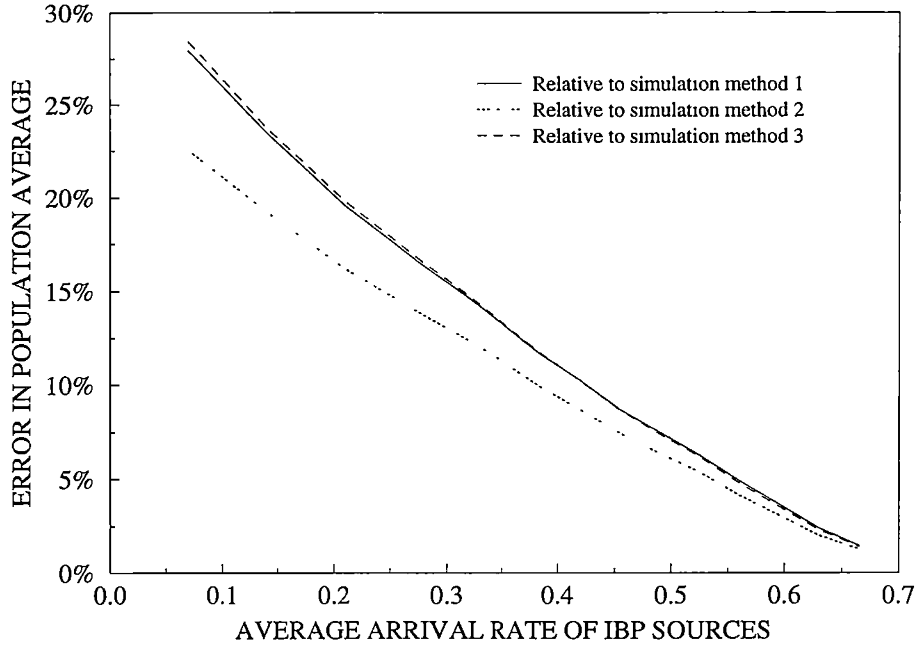

Figure 7.4: *Error in the theoretical average queue population of Figure 7.2 expressed as a percentage of the 3 simulation averages.*

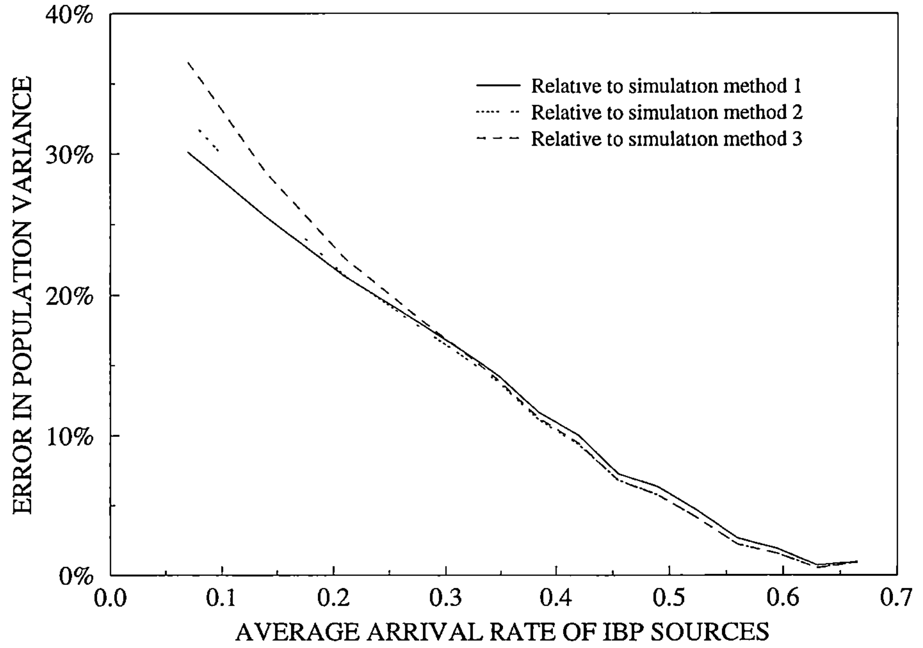

Figure 7.5: *Error in the theoretical queue population variance of Figure 7.3 expressed as a percentage of the three simulation variances.*

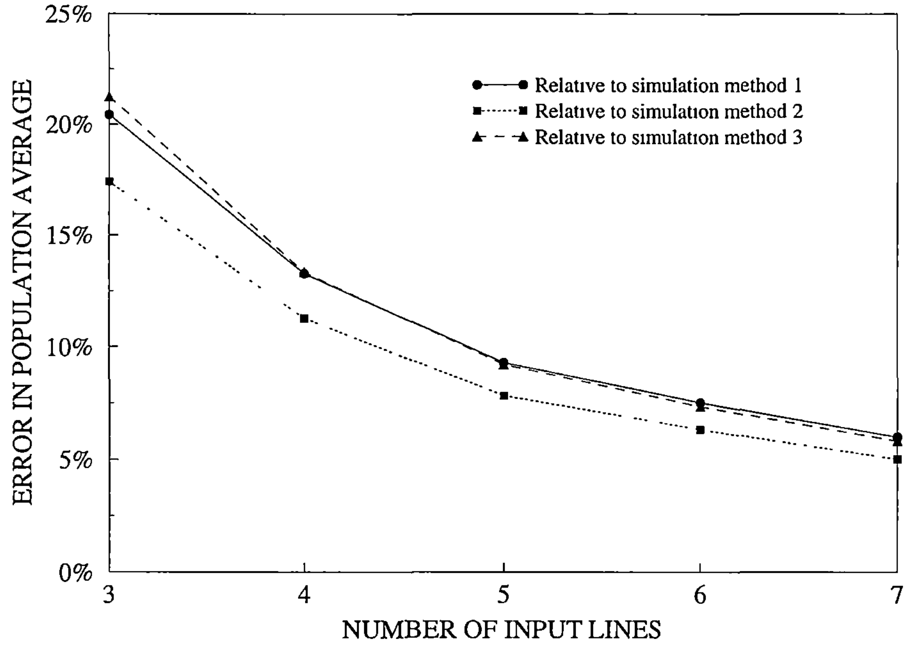

Figure 7.6: *Percentage error in the theoretical average queue population for the low priority buffer as a function of the number of sources. The results are for a cyclic source with* $b = 3$ *and* $C = 10$, *and* N *identical IBP sources with parameters* $\lambda_i = 0.35/N$, $\theta_i = 0.4$, *and* $\gamma_i = 0.7$.

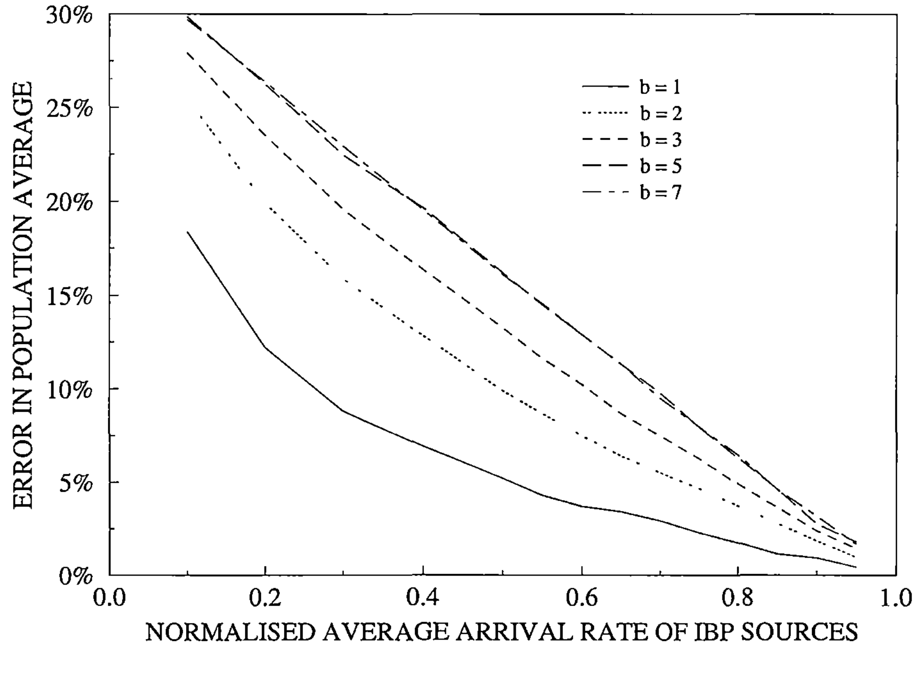

Figure 7.7: *Percentage error in the theoretical average queue population for the low priority buffer measured relative to the first simulation method. The results are for* 3 *values of* b *with* $C = 10$ *and the* 4 *IBP sources defined as for Figure 7.2.*

(in this case) of -0.33. This result is interesting and it appears, from several other simulations investigated by the author, that the slope of this limiting line is given by $-1/(N-1)$ where N is the number of input lines. How and why this straightline behaviour occurs is not understood, however it does provide an upper limit to the error in the independence assumption for the average queue population. Looking at Figure 7.6 we see that $0.5/(N-1)$ is indeed an upper limit for all three simulation methods[2].

This study has not been a particularly intensive one due to the long run times of the simulations. It has shown however that the independent source queueing model can provide reasonable upper bounds on the average and variance of the queue population for the low priority buffer, particularly when the queue load is high or the number of input lines is large.

7.3 Dual Buffer Analysis with IBP Sources Only

The Bernoulli arrival process for the active state of the geom-geom IBP model arises partly from the assumption that the splitting process within a network switch can be reasonably well described by a random allocation of the incoming arrivals to the observed output [126, 142]. We have already mentioned that the assignment of high or low priority to the arrivals of a dual buffer system is also equivalent to a splitting process. Given that the initial splitting was based on an independence assumption, it is natural to assume a corresponding random assignment of priority. We will look at such an analysis for a dual buffer queueing system subject to arrivals from geom-geom IBP sources in the following.

Autocorrelated destination splitting and priority splitting require more than two states to describe the corresponding IBP processes. In this thesis we have considered only two state models, although three state models have been investigated in [16, 68, 124], and most recently by Steyaert and Xiong in [126].

7.3.1 Random Priority Splitting

Consider a non-blocking, output queueing ATM switch with N input lines and N output lines, all having the same transmission speed. Arrivals on switch input line i (prior to passing through the interconnection network of the switching mechanism) are assumed to be described by a geom-geom binary process with average arrival rate Λ_i

[2] We have mentioned that the slope of the limiting line is given by $-1/(N-1)$ where the independent variable is the normalised average arrival rate. Since the example problems of Figure 7.6 all use an average arrival rate of half the available capacity, the upper limit of $0.5/(N-1)$ follows directly.

and autocorrelation parameter γ_i. The switching mechanism directs incoming cells to each of the N output ports at random, according to a specific weighting scheme. Let θ_i denote the probability that an arrival from input line i is directed to the output port of interest. The arrival process to this port from input line i will therefore be described by a geom-geom IBP process with an average arrival rate given by $\lambda_i = \theta_i \Lambda_i$, an average arrival rate in the active state of θ_i, and the same[3] autocorrelation parameter as the input line process of γ_i.

At each output port, arrivals from input line i are assumed to have a high priority with probability h_i, independently of past or concurrent arrivals. Thus the arrival process to the high priority queue can also be described by the superposition of N geom-geom IBP sources, where this arrival process from input line i has an average arrival rate of $h_i\lambda_i = h_i\theta_i\Lambda_i$, an average arrival rate in the active state of $h_i\theta_i$, and again the same autocorrelation parameter as for the entire input line of γ_i.

Thus we know exactly the parameters of the arrival process for both the entire output port, and for the high priority buffer of the output port. Since the high priority buffer is assumed to receive exhaustive priority service (i.e. arrivals to this buffer receive non pre-emptive head of line priority), its queue population statistics can be obtained directly from the high priority arrival process. Corollary 2.2 can the be applied to obtain the equivalent statistics for the low priority buffer from the total arrival process as

$$L_{q_{\text{low}}} = L_{q_{\text{both}}} - L_{q_{\text{high}}} \tag{7.1}$$

and

$$\text{Var}\left[L_{q_{\text{low}}}\right] = \text{Var}\left[L_{q_{\text{both}}}\right] - \text{Var}\left[L_{q_{\text{high}}}\right] - \text{Cov}\left[L_{q_{\text{low}}}, L_{q_{\text{high}}}\right] \tag{7.2}$$

where $L_{q_{\text{low}}}$ denotes the average queue population of the low priority buffer, $L_{q_{\text{high}}}$ the average for the high priority buffer, and $L_{q_{\text{both}}}$ the average queue population for a single buffer queueing system subject to arrivals from both the high and low priority arrival processes — which in this case is simply given by the arrival process to the output port prior to the priority splitting[4]. The variance terms are similarly defined, but involve the covariance of the low and high priority buffer populations. Unfortunately, single buffer theory cannot give this covariance term directly, and hence for the low priority buffer we can only determine the average and an upper limit on the variance of the queue population.

[3] The definition used for the autocorrelation parameter (which is based on the eigenvalues of the state transition probability matrix in Chapter 3) means that it does not change under random splitting.

[4] Corollary 2.2 (and its associated theorem) does not rely on the high and low priority buffer arrival processes being independent.

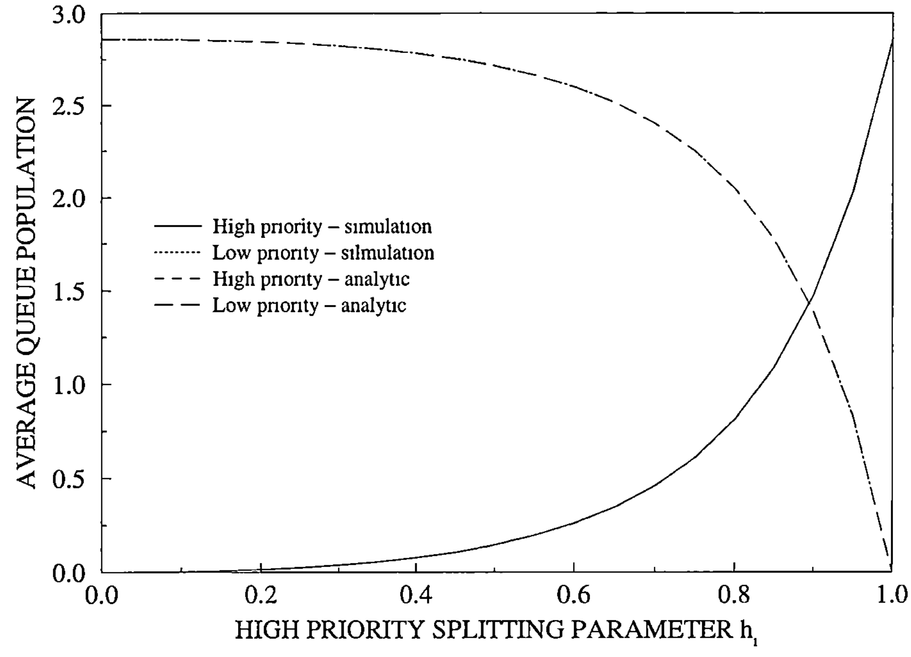

Figure 7.8: *Average queue populations for the low and high priority buffers as a function of the priority splitting probabilty h_i. The simulation results confirm both the IBP theory of Chapter 3 and Corollary 2.2 as it applies to the average queue population.*

A Simple Example

As an example we consider an ATM switch as described above, having 8 input and output ports, and using dual buffer priority queueing at the outputs. For simplicity we assume that the arrival processes at each input port are identical, as are all the splitting probabilities. Of course in practice this will rarely be the case, but the results apply equally well to both the homogeneous and heterogeneous cases, and the identical source assumption allows us to easily specify the arrival processes using the minimum of parameters. For this example we will use $\Lambda_i = 0.4$, $\gamma_i = 0.8$, and $\theta_i = 0.25$ (resulting in each $\lambda_i = 0.1$), and will vary h_i to observe the priority behaviour.

Figure 7.8 shows the average queue population for the two buffers as a function of the h_i value, obtained from both simulation and from the IBP theory of Chapter 3. The simulation results are all accurate to within ±0.5% with 99% confidence, and were obtained using a dual buffer arrangement — that is, they do not rely in any way on Corollary 2.2. Note that the low and high priority average queue populations are symmetric.

In [144], Zhang performs the fluid flow analysis of a dual buffer queueing system using random priority splitting, proposing the equivalent of Corollary 2.2. In a numerical

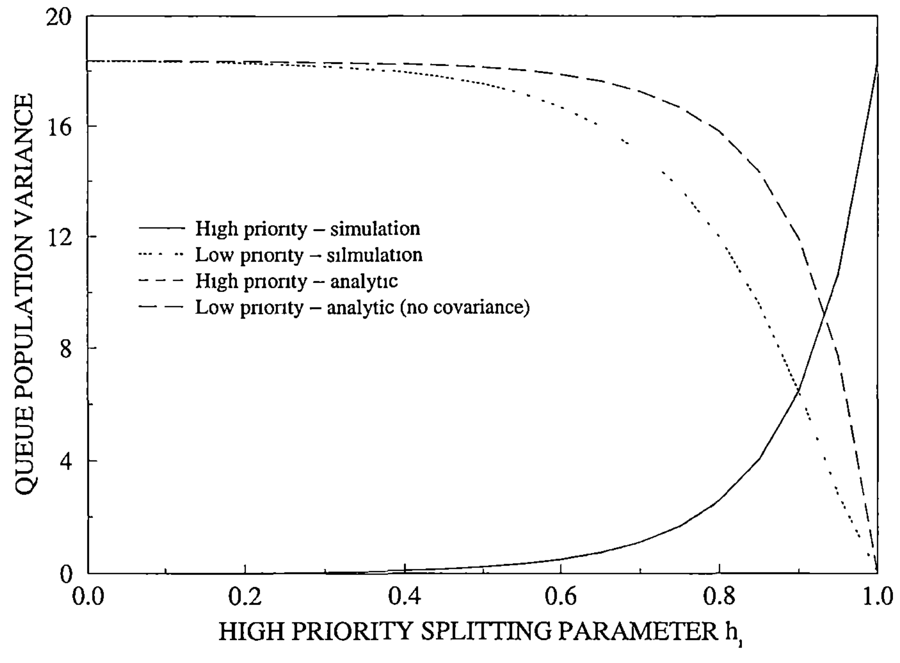

Figure 7.9: *Queue population variance for the low and high priority buffers as a function of the priority splitting probability h_l for the example problem. Although we can calculate the variance result for the high priority buffer using single buffer analysis, we can only approximate the result for the low priority buffer.*

example, Zhang observes that the sum of the low and high priority average queue populations appears to be invariant to the splitting probability, but does not recognise the reason for this. From Corollary 2.2, the sum of the average queue populations of the two buffers will be equal to a single buffer with arrivals from the combined high and low priority arrival processes. Since the combined arrival process is simply the arrival process before the priority splitting, the average queue population of this single buffer equivalent must be invariant to the splitting probability, and hence we have the symmetrical nature of Figure 7.8.

Figure 7.9 shows the corresponding queue population variances for this example problem. However, since we cannot obtain the exact value of the population variance of the low priority queue using single buffer analysis methods alone, we have instead included an analytical approximation for the variance by assuming that the low priority buffer receives service independently of the high priority buffer (zero covariance). Obviously this method overestimates the low priority buffer result by a considerable margin in some cases.

In addition to the average and variance of the queue population (and in turn of the queueing delay) the other main parameter of interest is the loss probability expected

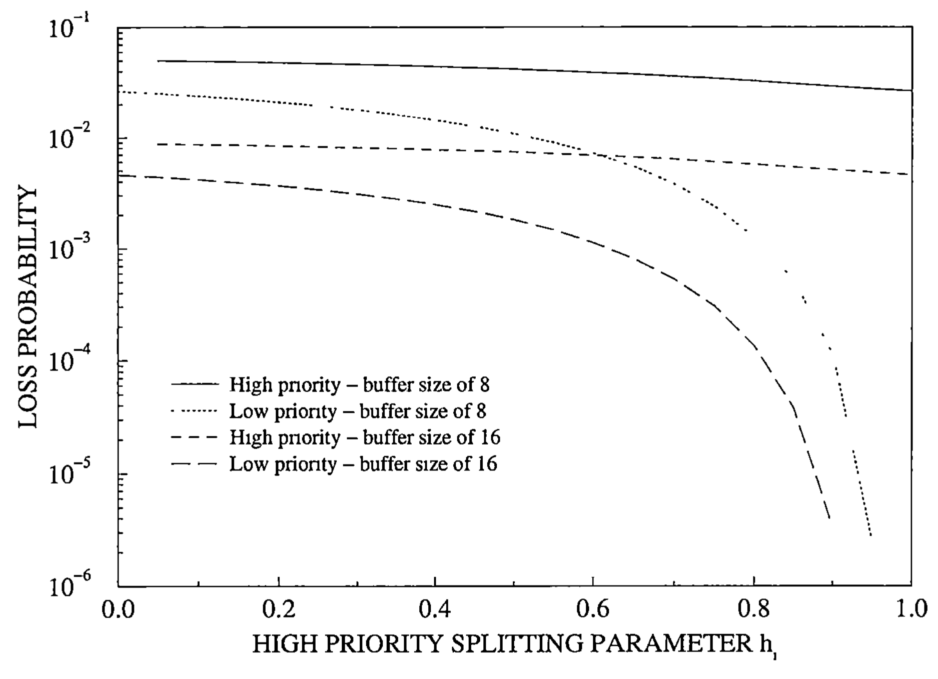

Figure 7.10: *Loss probabilities for the high and low priority buffers obtained from simulation as a function of the priority splitting probability h_l for the example problem. The low priority traffic ıs implemented wıth push out priority over the high priority traffic, which gives it the better loss performance.*

from this queueing system for finite buffer capacities. In chapter 6 we discussed the fact that the overall loss performance for a dual buffer system could be estimated from its single buffer equivalent provided that the available buffer capacity was fully utilised[5]. The individual buffer loss probabilities can only be estimated however when either there is no push out priority (equal loss probabilities), or when the high priority arrivals have push out priority over the low priority traffic. The alternative, where low priority arrivals can push out queued high priority arrivals precludes this individual buffer loss estimation.

Figure 7.10 shows the loss probabilities for the low and high priority buffers obtained from simulation of the example problem using total buffer capacities of 8 and 16 cells. Low priority arrivals, representing loss sensitive traffic, are assumed to have push out priority over the high priority arrivals, resulting in better loss performance for this buffer. We have used these small buffer sizes to keep the loss probabilities large enough to be obtained accurately using simulation without requiring excessively long observation times. Larger buffer capacities would be available in actual switches however, with corresponding smaller loss probabilities.

[5] Cells are only lost from the system when there is no capacıty available in eıther of the switch output buffers

Although we have not included it in the Figure, the relationship between the individual and combined buffer loss probabilities, given by equation (6.38) holds in this case as

$$(1 - h_i)\,\xi_{\text{low}} + h_i\xi_{\text{high}} = \xi_{\text{both}} \tag{7.3}$$

where ξ_{low} and ξ_{high} are the loss probabilities from the low and high priority buffers respectively, and ξ_{both} is the loss probability from the single buffer equivalent system, which is independent of h_i. Thus at $h_i = 0$ we have $\xi_{\text{low}} = \xi_{\text{both}}$ while at $h_i = 1$ we have $\xi_{\text{high}} = \xi_{\text{both}}$. In general however we cannot say what the individual buffer loss probabilities will be.

7.4 Modelling the Outputs of Queues

An important requirement for modelling networks of queues is being able to describe the merging and splitting processes of the carried traffic as it crosses the network. As we discussed in Chapter 1 the introductory chapters, the exact splitting process is the most difficult of these to describe analytically, and few results are available. The merging process is basically described by the queueing of input traffics at the buffers of the network switches, and we have looked at this in terms of queue populations and delays throughout this thesis. Another part of the merging process which we have not so far addressed though is the output traffic resulting from the queueing in the switch buffers. This output process is the aggregate of the individual traffic streams that are split again at the next switching element.

In this section we will look at how this aggregate output process can be described using various traffic models. Although many studies have been devoted to statistical considerations of the behaviour of the output process [49, 50, 102, 119], here we are concerned only with how the parameters of a particular model can be matched to the output process resulting from a queueing system. The parameter matching technique investigated here is based on Theorem 2.1 and its associated corollaries, and provides a general numeric based approach. The number of parameters that can be matched depends on the complexity of the model and how much is known about the input processes and the queue behaviour. In one particular case, which we discuss in section 7.6, the approach provides the moments of the idle and busy periods of the queue output process in analytic form.

Once the aggregate or merged stream is modelled or described in this manner, the splitting process may be accommodated by applying a suitable operation to the merged stream (such as the independent splitting that is the basis of the two-state IBP model, or the correlated splitting process described in [124] for example).

Note that we will only deal with stable infinite buffer queues in this study. Some of the results may also apply to finite buffers, but the infinite buffer assumption will probably hold quite well in those situations where the loss probabilities are very low (such as recommended for the B-ISDN).

7.4.1 The Parameter Matching Method

The basic idea behind the following method is that the output process can be modelled by describing how it affects the behaviour of a queue subject to arrivals from this process. Since we are using discrete-time models in which the slot time is equal to the service period, a queue with arrivals from a previous queue only will have all its population moments equal to zero, which is not much help. Thus we introduce an additional arrival process to overcome this problem. This additional arrival process is called the *test source* although properly speaking it is not the one being tested. To avoid increasing the computational requirements by too much we keep the test source to its simplest form by making it a Bernoulli process with parameter λ_{test}.

To avoid any confusion we will refer to the queue for which we wish to model the output process as the queue under study, or the *target queue*. The queue which is used to test the model is referred to as the *test queue*. We will assume that there are N sources generating arrivals to the target queue, and that the parameters of these sources are known in sufficient detail to determine the queue behaviour. For convenience the total average arrival rate from these N sources will be denoted by λ. The model for the output of the target queue will be simply referred to as the output model, with parameters identified by a subscript 'model'.

Suppose that the queue under study is actually the higher priority buffer of a dual buffer queueing system[6] and that the lower priority buffer is the test queue, which is subject to arrivals from the test source only. From the way the dual buffer system is implemented, the test queue only receives service when the target queue is empty prior to the service instant. Similarly, since the target queue always receives service if it has queued arrivals, the presence of the lower priority buffer will be completely transparent to the target queue and will not affect its performance.

From the point of view of the test queue it is unimportant whether its services are determined by the target queue or by a sufficiently accurate model of that queue. In either case its measurable queueing performance should be the same (see Corollary

[6] Although we have not discussed this here, it is possible to extend this approach to target queueing systems involving multiple buffers themselves All that is important is that the test queue is the lowest priority buffer.

2.3). The aim of this modelling approach then is to find the parameters of the model that provide the same average population for the test queue as the target queue causes. In many cases higher moments of the queue population may also be required in order to match all the parameters of the model. In the following procedural outline we only specify the average and variance of the queue population, but the same approach applies in general.

1. Obtain the average $L_{q_{\text{test}}}$ and variance $\text{Var}\left[L_{q_{\text{test}}}\right]$ of the queue population for the test queue when the service interruptions are caused by the target queue.
2. Assign initial parameters to the output model, and calculate the approximated values for the average and variance of the queue population for the test queue when the service interruptions are caused by the output model. We refer to this queue as the *approximation queue* to distinguish it from the actual test queue, and denote its population average and variance by $L_{q_{\text{approx}}}$ and $\text{Var}\left[L_{q_{\text{approx}}}\right]$ respectively.
3. Adjust the model parameters (using some appropriate method) and return to step 2 until the difference between $L_{q_{\text{approx}}}$ and $L_{q_{\text{test}}}$ and between $\text{Var}\left[L_{q_{\text{approx}}}\right]$ and $\text{Var}\left[L_{q_{\text{test}}}\right]$ is acceptably small.

We have not so far specified how the average and variance of the population for the test queue are calculated. From Corollary 2.2 we have

$$L_{q_{\text{test}}} = L_{q_{\text{both}}} - L_{q_{\text{target}}} \tag{7.4}$$

for the average queue population, where the 'both' subscript indicates the population observed for a single buffer queue subject to arrivals from the N sources of the target queue and the test source as well. Calculation of the variance of the test queue population is not quite so straightforward however due to the presence of a covariance term in the equivalent expression

$$\text{Var}\left[L_{q_{\text{test}}}\right] = \text{Var}\left[L_{q_{\text{both}}}\right] - \text{Var}\left[L_{q_{\text{target}}}\right] - \text{Cov}\left[L_{q_{\text{test}}}, L_{q_{\text{target}}}\right] \tag{7.5}$$

In this thesis we have not looked at how this covariance term can be calculated, leaving at as a topic for future research. A loose upper bound for the test queue variance can be obtained however by assuming the covariance term is zero (as was done in Figure 7.9 for the random priority assignment example). Thus to calculate the average test queue population, and an upper bound for the corresponding variance, requires the solution of the population moments for the target queue, and for a combined arrival process queue involving a total of $N + 1$ sources.

Calculation of the solutions using the output model are similar, but are simplified by the fact that high priority queue, which causes the service interruptions of the approximation queue, has a population average and variance of zero (cf. Corollary 2.2). Hence

$$L_{q_{\text{approx}}} = L_{q_{\text{test+model}}} \tag{7.6}$$

and

$$\text{Var}\left[L_{q_{\text{approx}}}\right] = \text{Var}\left[L_{q_{\text{test+model}}}\right] \tag{7.7}$$

where the 'test+model' subscript indicates that the output model and the test source are treated as the two arrival processes to the queue. Thus each repetition of step 2 in the parameter matching process will involve the analysis of a queueing system subject to arrivals from just two sources, enabling this calculation to be performed very quickly.

In order to choose an appropriate value of λ_{test} we note that it should be large enough that the average and variance of the test queue are of significant magnitude. Noting that these measures will increase rapidly as $(\lambda + \lambda_{\text{test}})$ approaches 1, a good choice of λ_{test} might be

$$\lambda_{\text{test}} = \max\left[0.9 - \lambda, 0.01\right] \tag{7.8}$$

so that the average arrival rate to the combined queue will be at least 0.9 when λ is small, but for λ close to or greater than 0.9, λ_{test} reduces to 0.01. As $(\lambda + \lambda_{\text{test}})$ gets very close to 1.0 (say when $\lambda \approx 0.99$) the extreme sensitivity of the queue population statistics to variations in the average arrival rate also increases, and smaller values of λ_{test} could be easily used. Note also that an appropriate output model should give the same performance regardless of the magnitude of λ_{test} and this can be used as a measure of the accuracy of the model.

Probably the hardest step in the matching process will be determining which parameters are to be adjusted in step 3, and by how much. Some model parameters can be calculated from the parameters of the N sources feeding the target queue (such as the average rate λ_{model} of the output model which will be equal to λ) but the rest must be calculated using a suitable search algorithm. If there is only one unknown parameter, with a specific range of possible values, a binary search might be used. For more than one unknown parameter however, multi-dimensional methods such as the Newton–Raphson will be required.

As an example, and opportunity to discuss some specifics, the following section looks at a queueing system subject to arrivals from geom-geom IBP sources, modelling the output process by a geom-geom IBP also.

7.5 The Geometric-Geometric IBP as an Output Model

In Chapter 3 we looked at queues fed by geom-geom IBP sources. If there is only a single source, then the queue output process will also be a geom-geom IBP process. In this section we will look at how well the geom-geom IBP performs as a model of the output process of this queueing system when the number of sources N is greater than one. Note that the example queueing problems we use here all involve identically distributed sources. This is merely for convenience of representation — the observations apply equally well to the more general heterogeneous case.

The geom-geom IBP model is described by three parameters — its average arrival rate λ_{model}, peak arrival rate θ_{model}, and autocorrelation parameter γ_{model}. Of these the simplest to specify is λ_{model} which will be equal to the average departure rate of the target queue, which in turn is equal to its average arrival rate λ. Thus only θ_{model} and γ_{model} need to be found to match $L_{q_{\text{approx}}}$ to $L_{q_{\text{test}}}$ and $\text{Var}\left[L_{q_{\text{approx}}}\right]$ to $\text{Var}\left[L_{q_{\text{test}}}\right]$. Although this is a straightforward task when $N = 1$, we do not even know whether there is a solution for these two model parameters that will satisfy both the average and variance at the same time for $N > 1$.

Figure 7.11 shows the average and variance parameter curves for an example queueing problem having 4 identical sources for two different values of λ_{test}. The curves were generated by selecting a value of θ_{model} and then finding the value of γ_{model} (using a binary search) that matches either the average or the variance of the approximation queue to that of the test queue[7]. Two facts are immediately apparent from the Figure — there is no θ_{model} and γ_{model} pair that matches both the average and the variance simultaneously, and the solutions vary considerably with λ_{test}. Both of these observations indicate that the geom-geom IBP is a poor model for the output process of these types of queueing problems.

In Chapter 4 we saw that the variance of the queue population when each $\theta_i = 1$ was determined by the first three moments of the active period of the autocorrelated sources, and we can surmise that the same is probably also true for the output process. Figure 7.12 shows the autocorrelation coefficient function $R(m)$ for the output process of an 8 source example problem, as well as for two particular solutions of the geom-geom IBP model parameters. Obviously the actual output process consists of more than a single geometric component, which suggests that at least a three-state IBP model would be required to be able to match the variance.

[7] In order to establish the exact parameter curves for the variance in Figure 7.11, simulation was used to obtain each $\text{Var}\left[L_{q_{\text{test}}}\right]$ rather than using the loose upper bound values calculated from the single buffer theory under the zero covariance assumption. The simulation results were accurate to within ±0.25% with 99% confidence

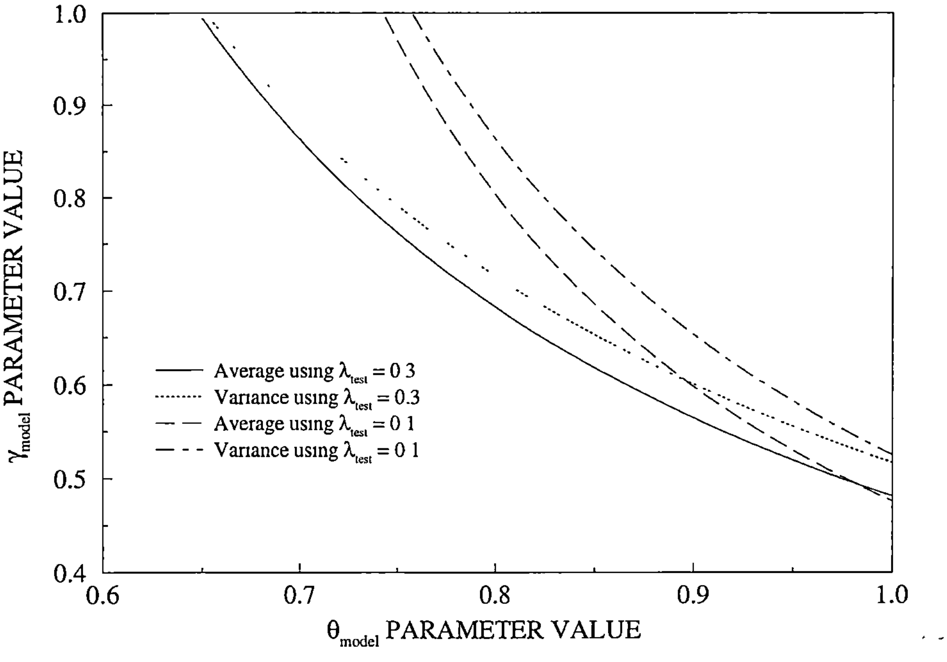

Figure 7.11: *Parameter curves for an example problem involving* 4 *identical sources* ($\lambda_i = 0.15$, $\theta_i = 0.3$, $\gamma_i = 0.5$) *for two different values of* λ_{test}. *The parameter curves describe those values of* θ_{model} *and* γ_{model} *that match either the average or the variance (as specified) of the approximation queue to the test queue.*

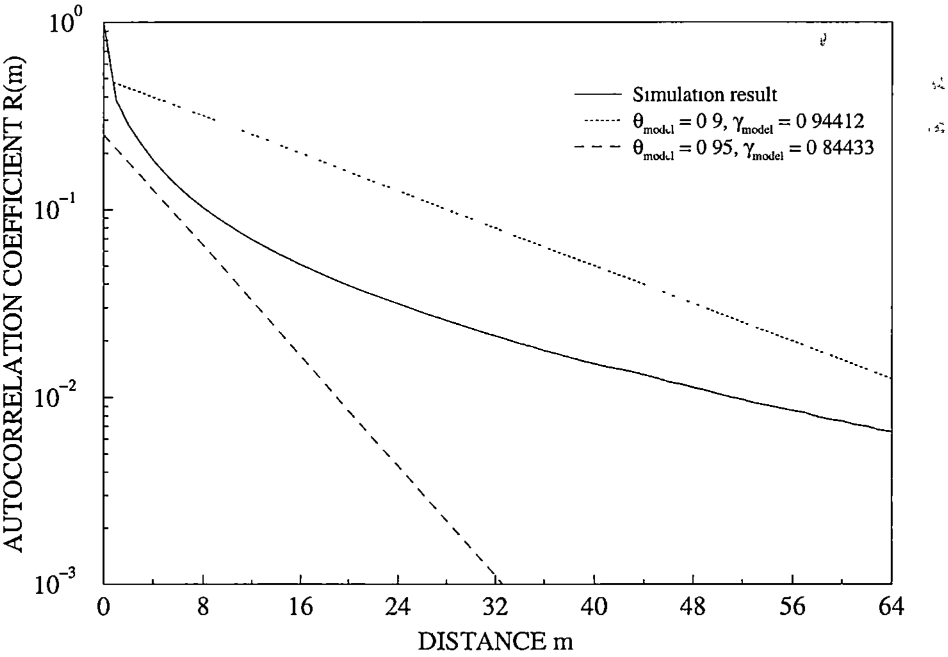

Figure 7.12: *Autocorrelation coefficient function for the output process of a target queue fed by* 8 *identical sources with parameters* $\lambda_i = 0.1$, $\theta_i = 0.3$, *and* $\gamma_i = 0.6$. *The same function is also shown for two geom-geom IBP models of the process with parameters as indicated by the legend.*

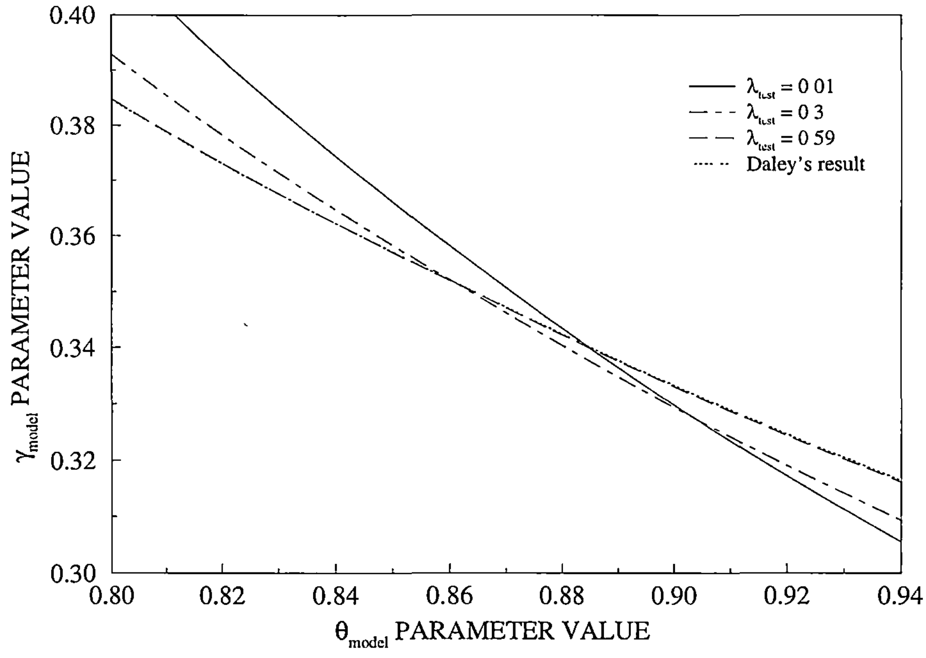

Figure 7.13: *Region in which variations in λ_{test} have the least effect on the parameter curves for the average queue population. The results are for an example problem involving 4 identical sources with $\lambda_i = 0.1$, $\theta_i = 0.2$, $\gamma_i = 0.5$. Daley's result is given in this case by $\gamma_{model} = 0.25/\left(\theta_{model} - 0.15\right)$.*

If a geom-geom IBP is to be used as the output model however, we need to decide which combination of θ_{model} and γ_{model} is the 'best' one to use. From Figure 7.11 we see that there is one point at which the average population parameter curves for $\lambda_{\text{test}} = 0.1$ and $\lambda_{\text{test}} = 0.3$ cross. It would be convenient if the parameter curves for every value of λ_{test} passed through this point, because it would imply that this solution for the IBP model parameters was invariant to λ_{test}, which is desirable for an output model. There is no actual point that all these curves pass through, but instead there is a region where these curves intersect. The size of this region is determined partly by the average arrival rate to the target queue — if λ is close to 1, the region is quite small, while for small λ the region can be quite large. Figure 7.13 illustrates this region effect for the average population parameter curve for an example queueing system involving 4 identical IBP sources with $\lambda = 0.4$.

We see that as λ_{test} increases towards $1 - \lambda$, the parameter curve approaches what we have termed Daley's result in Figure 7.13. This result is based on an observation by Daley in [19] which we will explain in more detail in the following section. It is not known why in particular Daley's result provides this upper limit on the behaviour of the average population parameter curve, but all our studies have shown this to be the case.

Another observation that can be made from Figure 7.11 is that the difference between the parameter curves for the average and variance decreases as λ_{test} increases. In the limit as λ_{test} tends towards $1 - \lambda$ (the approximation queue utilisation tends to unity) these two curves becomes equal, coinciding with the parameter curve described by Daley's result. In section 4.3.4 we investigated the use of a two-state binary model for describing the variance behaviour of three and four-state binary models. It was shown there that the difference in the variance between the two approaches could be roughly bounded by a quantity that was independent of the actual variance magnitude. The consequence of this was that the relative error in the two-state model decreased as the variance magnitude increased. In terms of the parameter curves then, as λ_{test} tends towards $1-\lambda$, and the magnitude of the test queue population variance increases towards infinity, it is perhaps not surprising that the two parameter curves tend to coincide.

7.5.1 Daley's Result

Define the counting process $N(t)$ to be the number of events (arrivals or departures) occurring in an arbitrary time interval of t slots. The *asymptotic variance ratio* of the process $N(t)$ is defined as

$$v = \lim_{t \to \in \infty} \frac{1}{t} \operatorname{Var}\left[N(t)\right] \tag{7.9}$$

for which it can be shown (using the well known result for the variance of the sum — see [6] for example) that

$$v = \sigma^2 \left(1 + 2S\right) \tag{7.10}$$

where σ^2 is the variance of the process and S is the single-sided sum of the autocorrelation coefficient function $R(m)$, given by

$$S = \sum_{m=1}^{\infty} R(m) \tag{7.11}$$

In [19], Daley points out that the asymptotic variance of the net arrival process to an infinite buffer queue is equal to the asymptotic variance of the aggregate server process, or $v_{\text{out}} = v_{\text{in}}$[8]. In terms of the single-sided autocorrelation coefficient function sums, this becomes

$$S_{\text{out}} = \frac{\sigma_{\text{in}}^2 - \sigma_{\text{out}}^2}{2\sigma_{\text{out}}^2} + \frac{\sigma_{\text{in}}^2}{\sigma_{\text{out}}^2} S_{\text{in}} \tag{7.12}$$

where σ_{in}^2 is the variance of the net arrival process and $\sigma_{\text{out}}^2 = \lambda \left(1 - \lambda\right)$ is the variance of the server output process, where λ is the average arrival rate.

[8] Daley actually discusses this result in terms of the index of dispersion of intervals (IDI), while the definition of the asymptotic variance ratio used here is closely related to the index of dispersion of counts (IDC). The IDC and IDI have the same limiting behaviour however [40] and so the result still holds.

Daley's result has been applied elsewhere in the literature for modelling output processes. In [4], Addie and Zukerman used the result to obtain the parameters of a Gaussian output model for queues fed by Gaussian sources, constructing a solution for a tree type network (only merging, no splitting). Theimer [131] used the result to show that for a network of ATM switches with equally loaded inputs and outputs and random path selection, the asymptotic variance of the internal network traffic approaches that of an M/D/1 queue as the number of switching stages passed through increases.

An alternative approach used by Stavrakakis in [124] matches the short term autocorrelation behaviour of the output process. Stavrakakis assumes that the output process of the queue can be modelled by a geom-geom binary process (so that $\theta_{\text{model}} = 1$) and then calculates γ_{model} from the probability that the system is empty in two consecutive time slots. This calculation requires the empty system probability vector to be known, which may present some difficulties when the number of sources is large. In addition, Stavrakakis' approach does not always provide 'legal' values of γ_{model} — for example, the queueing problem of Figure 7.13 results in a $\gamma_{\text{model}} = 1.0802$ using the calculation method in [124].

To apply Daley's result to the IBP arrivals and output model problem, we start by noting that the autocorrelation coefficient function for geom-geom IBP source i is given by

$$R_i(m) = \left(\frac{\theta_i - \lambda_i}{1 - \lambda_i}\right) \gamma_i^{|m|} \tag{7.13}$$

and similarly for the output model by

$$R_{\text{model}}(m) = \left(\frac{\theta_{\text{model}} - \lambda}{1 - \lambda}\right) \gamma_{\text{model}}^{|m|}$$

Combining these using equation (7.12) then yields

$$\gamma_{\text{model}} = \frac{(1 - \lambda)\, S_{\text{out}}}{\theta_{\text{model}} - \lambda + (1 - \lambda)\, S_{\text{out}}} \tag{7.14}$$

where

$$(1 - \lambda)\, S_{\text{out}} = \frac{M_2 - \lambda}{2\lambda} + \frac{1}{\lambda} \sum_{i=1}^{N} \lambda_i \left(\theta_i - \lambda_i\right) \frac{\gamma_i}{1 - \gamma_i} \tag{7.15}$$

and where λ and M_2 are the first and second moments of the combined arrival process. Thus we have a relationship between θ_{model} and γ_{model} based on matching the asymptotic variance of the target queue arrival process to the parameters of the output model using Daley's result.

We have already noted that this result provides a limiting parameter curve for the average queue population, and as such passes through that region in which changes in λ_{test} have the least effect on the model parameters. Thus, the intersection of the parameter curve described by Daley's result and any other parameter curve (say from

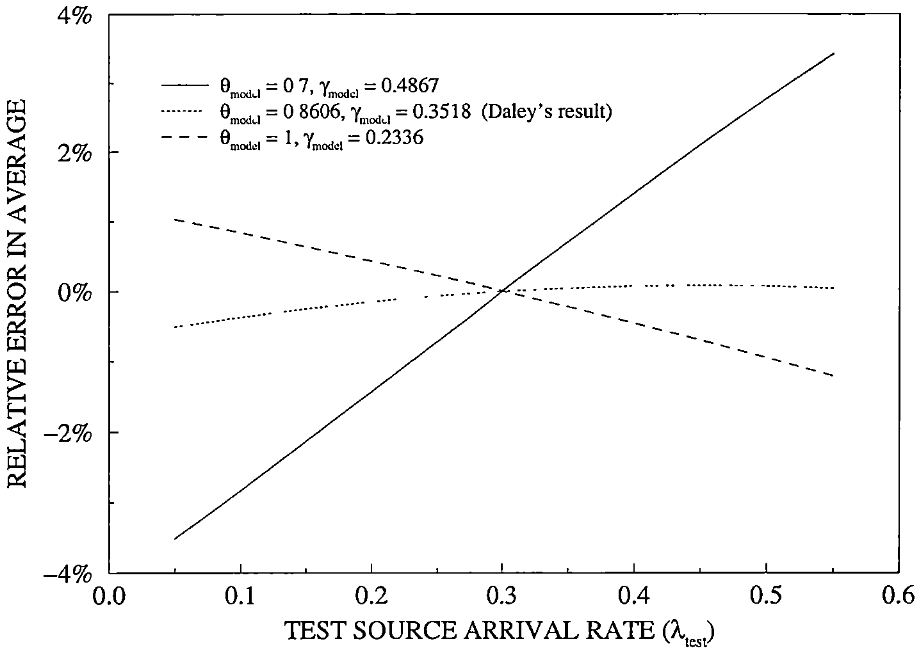

Figure 7.14: *Comparison of the relative error in the average queue population of the approximation queue compared to the test queue as a function of λ_{test} for three sets of model parameters. The arrival process to the target queue consists of* 4 *identical sources each with $\lambda_i = 0.1$, $\theta_i = 0.2$, and $\gamma_i = 0.5$.*

a small value of λ_{test}) will be a solution for the IBP model that will be largely invariant to λ_{test}. That is, the parameters obtained from this intersection point provide an IBP model that will be fairly accurate, in terms of the average queue population, for any value of λ_{test}. This intersection point is easily found by performing a linear search on θ_{model} such that, with γ_{model} calculated from equation (7.14), we obtain $L_{q_{\text{approx}}} = L_{q_{\text{test}}}$.

To verify that this solution for θ_{model} and γ_{model} does provide the 'best' solution in terms of invariance to λ_{test}, Figure 7.14 shows the average queue population for an example queueing system[9], where the target queue is fed by 4 identical IBP sources with $\lambda = 0.4$, and where λ_{test} is varied from 0.05 to 0.55. The three solutions for the geom-geom IBP output model were calculated from the average parameter curve using $\lambda_{\text{test}} = 0.3$ (the midpoint of the range of λ_{test}). They correspond to points obtained from roughly halfway along the parameter curve ($\theta_{\text{model}} = 0.7$, $\gamma_{\text{model}} = 0.4867$), the intersection with Daley's result ($\theta_{\text{model}} = 0.8606$, $\gamma_{\text{model}} = 0.3518$), and the curve's endpoint ($\theta_{\text{model}} = 1.0$, $\gamma_{\text{model}} = 0.2836$).

Daley's result provides two points on the relative error curve for the average queue

[9] We are using fairly small values of λ here because the effects we are trying to illustrate are less pronounced at higher utilisations.

population that are exact — one at the point at which the model is evaluated (in this case at $\lambda_{\text{test}} = 0.3$) and the other at $\lambda_{\text{test}} = 1 - \lambda$ (an unreachable point only since we restrict ourselves to stable infinite buffers). This could be achieved in general for any solution by finding the intersection of the parameter curves for any two λ_{test} values but Daley's result is computationally more efficient and a good deal easier to implement.

Earlier we mentioned that it was best to choose λ_{test} large enough to make the magnitude of the test queue population significant. It was assumed that the model would not be susceptible to this choice explicitly — only indirectly due to round off errors if extreme values of λ_{test} were used. The geom-geom IBP model is a poor one in the sense that its parameters are sensitive to λ_{test} but we have shown that the use of Daley's result can improve the situation somewhat. In terms of a suitable choice of λ_{test} then, Figure 7.14 suggests that the midpoint between 0 and $1 - \lambda$ will probably be best in this case.

7.5.2 Some Trends in the Geometric-Geometric IBP Output Model

We are interested in what general trends may be identifiable in the output modelling process — in particular for the θ_{model} parameter. Stavrakakis proposes in [124] that the output process can be well modelled by a geom-geom binary process, which would imply $\theta_{\text{model}} = 1$. If we can make this assumption, then the problem of finding γ_{model} is considerably simplified. For example, Daley's result provides an immediate value for this quantity through equation (7.14) without actually requiring the evaluation of the target or test queue populations. Alternatively, since λ_{test} is a marginal arrival process, the average queue population of the approximation queue will be given by the closed form equation (4.11), which can be easily rearranged to provide γ_{model} in terms of $L_{q_{\text{test}}}$. The difference in the γ_{model} value obtained from these two solution methods would indicate to some degree how well the assumption that $\theta_{\text{model}} = 1$ applied to this problem.

To understand when this assumption might be used, we will begin by studying the effect on the calculated value of θ_{model} of varying the input parameters of a queueing system fed by four identical geom-geom IBP sources. These θ_{model} values (and all those obtained in this section) are obtained from the intersection of Daley's result with the average queue population parameter curve, using a λ_{test} value equal to the midpoint between 0 and $1 - \lambda$, as we suggested above. Figures 7.15 and 7.16 show θ_{model} as a function of the θ_i and γ_i values respectively.

In Figure 7.15, we see that θ_{model} increases towards 1 as θ_i increases, but also increases towards one as θ_i tends toward λ_i from above. At both extreme values of θ_i, θ_{model}

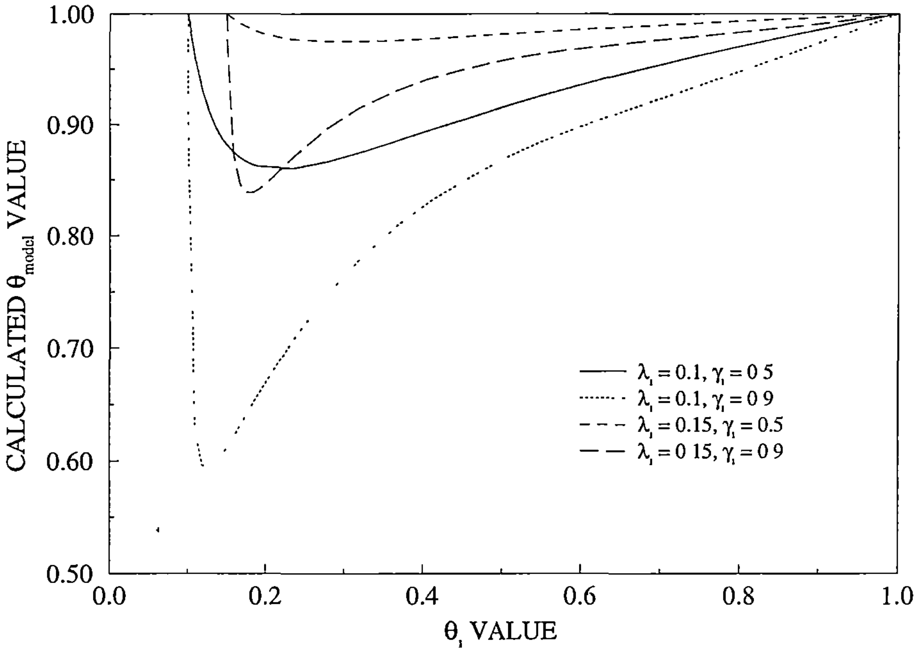

Figure 7.15: *Variation in the calculated value of θ_{model} as the θ_i parameter of the four identical input sources of a queueing system is varied from its minimum value of λ_i to its maximum value of* 1.

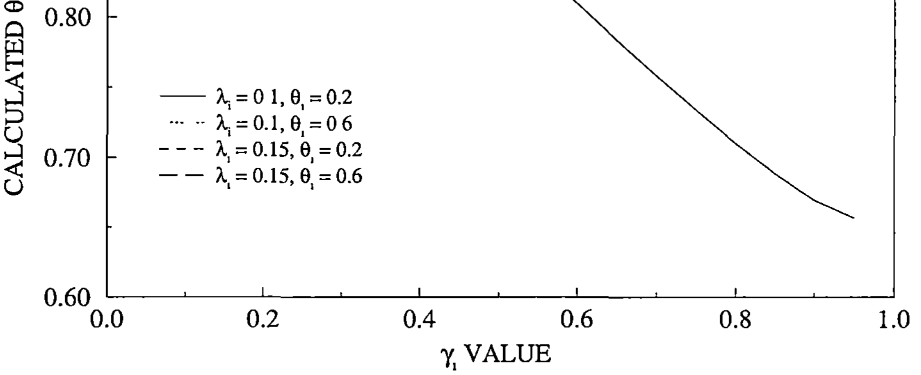

Figure 7.16: *Variation in the calculated value of θ_{model} as the γ_i parameter of the four identical input sources of a queueing system is varied from zero to* 0.95.

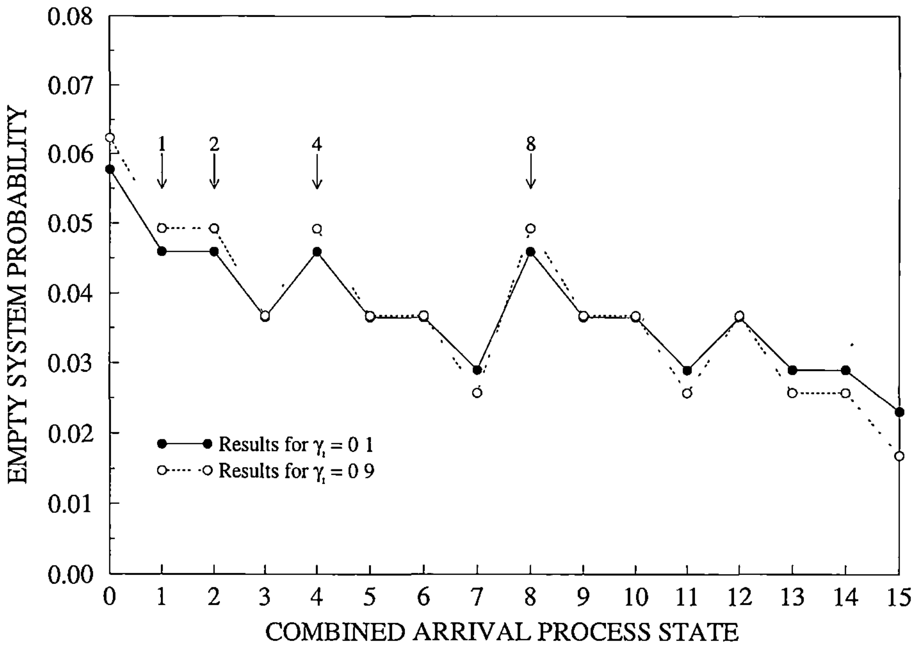

Figure 7.17: *Changes in the empty system probability vector for an example queueing problem involving* 4 *identical IBP sources with* $\lambda_i = 0.1$ *and* $\theta_i = 0.2$ *for the indicated* γ_i *values.*

takes on the value of one, as it does when $\gamma_i = 0$ in Figure 7.16. As θ_i tends towards λ_i, the proportion of time that the source spends in its silent state decreases towards zero, meaning that the source behaviour approaches that of a Bernoulli process with parameter λ_i, which is what happens when $\gamma_i = 0$ also. Thus the results indicate that when the inputs are all Bernoulli, or all geom-geom binary (with $\theta_i = 1$) then the output process is described by a model with parameter $\theta_{\text{model}} = 1$. The reason for this will be explained in section 7.6.

From Figure 7.16 we see that θ_{model} decreases as γ_i increases. The reason for this is not known exactly, but is probably related to the fact that increasing γ_i increases the probability that there will a single source active with an empty queue available. This will cause a decrease in θ_{model} because with only the one source active, once the queue empties, departures will be caused by arrivals from this one source, and hence for this interval, θ_{model} will tend towards this θ_i only. We can provide some support for this claim by investigating the changes in the empty system probability vector as γ_i increases. Figure 7.17 shows the entries of the empty system probability vector for two values of γ_i when $\lambda_i = 0.1$ and $\theta_i = 0.2$. We see that the probabilities at states 1, 2, 4, and 8 (which correspond to a single source being active) all increase as γ_i increases.

The total load on the queue in Figures 7.15 and 7.16 is quite low ($\lambda = 0.4$ and $\lambda = 0.6$),

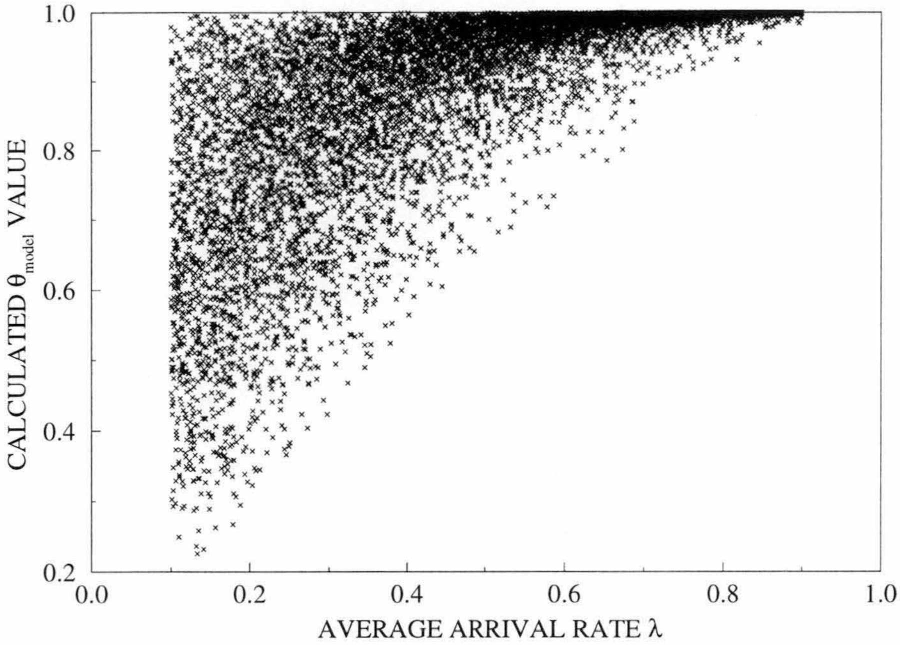

Figure 7.18: *Scatter plot of θ_{model} plotted against λ for 10,000 randomly generated queueing problems involving 4 sources. The autocorrelation parameters for each source in each case were chosen from the range $0 \leq \gamma_i < 0.99$.*

although it is readily apparent that the higher queue load results in θ_{model} values closer to one. This is not unexpected, since higher average arrival rates will require larger values of θ_i, which as indicated by Figure 7.18, will result in larger θ_{model} values. To explore this idea in more detail, Figure 7.15 presents the θ_{model} values obtained from 10,000 randomly generated queueing problems using 4 sources, plotted against the corresponding total average arrival rate. We see that for four sources, θ_{model} is consistently near 1 only for average arrival rates greater than about 0.6.

As a final consideration, we will look at the change in θ_{model} as the number of sources N increases. Again randomly generated queueing problems are used, but the total average arrival rates are restricted to $\lambda = 0.4$ and $\lambda = 0.6$. Figure 7.19 shows the average θ_{model} values observed for 1000 problems for each N from 1 to 7. As we have seen, the greater the arrival rate to the queue, the higher the value of θ_{model} on average. In addition, the more sources generating arrivals to the queue, the higher the value of θ_{model} also, although this trend in the average does not seem to increase much after 4 sources or so.

The fact that θ_{model} tends towards 1 as either the number of sources or the source activity increases lends support to Stavrakakis' assumption that the output process can be well modelled by a geom-geom binary process (at least as far as the average

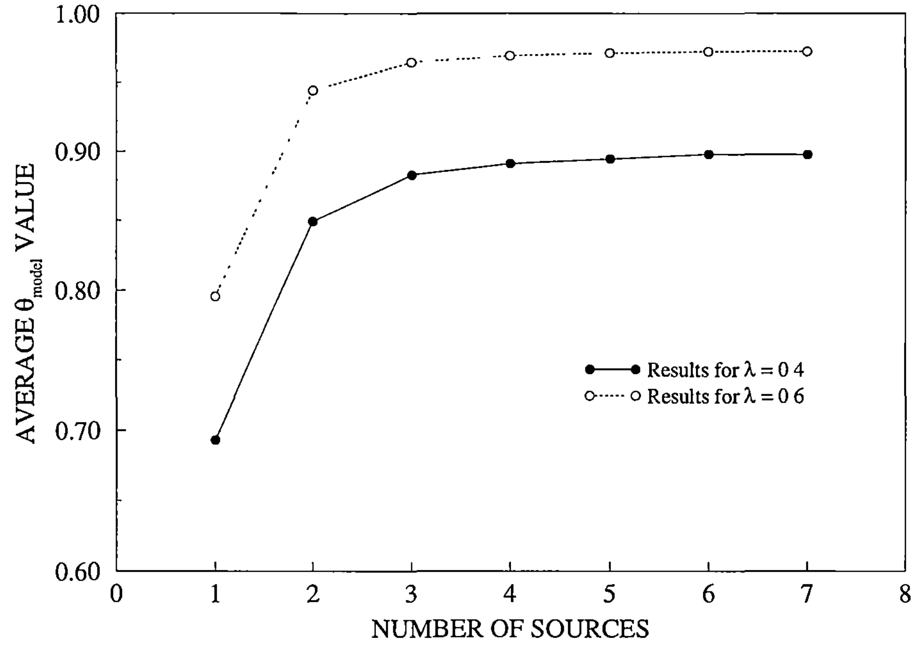

Figure 7.19: *Average value of θ_{model} observed for* 1000 *randomly generated queueing problems, as a function of the number of sources. The total average arrival rate from the sources is restricted to the values indicated, but otherwise the parameters of the individual sources are chosen at random, with the restriction that* $0 \leq \gamma_i < 0.99$.

queue population at following stages is concerned). We have not explicitly looked at the magnitudes of the errors in the average queue population predicted by assuming θ_{model} takes on the value of 1, but the results of Figure 7.14 suggest that these errors will most likely be small.

7.6 Server Idle and Busy Periods for Phase-Geometric Binary Sources

If the arrival processes to the target queue are all marginal (not autocorrelated) or alternatively if they are all phase-geom binary ($\theta_i = 1$) then closed form solutions for the average and variance of the population exist. Thus, for an appropriate model, these closed form solutions will allow us to obtain exact descriptions for the moments of the idle and busy periods of the queue output.

Consider an infinite buffer, discrete-time G/D/1 queue with arrivals from an irreducible, time invariant Markov modulated process (a D-BMAP) having the property that all states except one generate at least one arrival per time slot. It is easy to see that the idle period distribution of the departure process from this queue will be geometric, since only one state can generate zero arrivals per time slot. Furthermore, the durations of the idle period must be independent of those of the busy period because they are governed entirely by a single probability, and transitions from this state to other states must also have fixed probabilities.

The development of the system population for this problem can be written in matrix geometric form [97] as an irreducible stochastic matrix with an infinite state space. The empty system state must therefore be reachable from every other system state, and further must have a non-zero stationary probability. Under these conditions the time to return of the empty state follows a phase-type distribution [95]. Since a non-empty state corresponds to a single departure from the queue in each time slot, the output process has a phase-geom binary distribution[10].

The superposition of a number of phase-geom binary processes has only one state that generates no arrivals. Similarly, a marginal arrival process can be described by a D-BMAP having a single state (which can generate zero arrivals). Thus, the output process for these types of queueing problems will be phase-geom binary, which makes it the obvious choice for the output model. A phase-geom binary process with m distinct states requires $m\,(m-1)$ parameters to exactly describe its state transition parameters. Fortunately, it is not necessary to specify these probabilities explicitly in order to match the average and variance of the test queue — only the average arrival rate and the moments of the active state's phase-type distribution are required.

More specifically, the idle period moments and the first moment of the busy period, which can be obtained without the use of a test queue, require the average arrival rate and the first moment of the active period of each input process to be known. The second moment of the server busy period is obtained from the average population of the test queue, which requires that the second moment of active period of each input process also be known. The third moment of the output process is calculated from the variance of the test queue population, which requires the third moment of the active periods of the input sources to be known. Unfortunately we do not yet have a closed form expression for this quantity, due to the covariance between the low and high priority buffer populations being unknown.

An alternative is to describe each source by only three parameters — the average arrival rate, the ratio of the second moment of the active periods to the first, and similarly the ratio of the third moment to the first. These are the quantities required to calculate the average and variance of the queue population in the single buffer theory, and are the

[10] If the discrete-time Markov arrival process has more than one state in which zero arrivals can be generated (such as when IBP sources are used) a similar argument will show that the server idle periods also have a phase-type distribution, but the busy and idle periods will no longer be independent.

same parameters that can be obtained for the departure process given the test queue population statistics. The moments of the idle period (which are not actually required in the queue population calculations) will not be known in this case however since the first moments of the input active periods are not specified.

For notation we will indicate the parameters of the ith phase-geom binary source ($i = 1, 2, \ldots, N$) by λ_i for the source's average arrival rate, and $\eta_{i,r}$ for the rth moment of the active period of this source. The moments of the idle period for source i are similarly denoted by $\vartheta_{i,r}$. The autocorrelation parameter for source i introduced in Chapter 4 is defined by

$$\gamma_i = 1 - \frac{2}{(1-\lambda_i)\left(\frac{\eta_{i,2}}{\eta_{i,1}} + 1\right)} \tag{7.16}$$

which does not require $\eta_{i,1}$ or $\eta_{i,2}$ to be explicitly specified, only their ratio. The output process will be indicated by the subscript 'out' rather than by the 'model' term used previously because the model is exact, and the properties of the model are equal to the properties of the output process.

7.6.1 Idle Period Distribution and First Moment of the Busy Period

In the following we will use some of the notation of Appendix C as it applies to phase-geom binary processes. In addition, those results that are simply presented here without explanation can either be found in this appendix, or can be simply derived from results presented there.

The phase-geom binary process describing the departures of the observed buffer has a state transition probability matrix defined as

$$\mathbf{A}_{\text{out}} = \begin{bmatrix} c_{\text{out}} & c'_{\text{out}}\boldsymbol{\alpha}_{\text{out}} \\ \mathbf{T}^{\circ}_{\text{out}} & \mathbf{T}_{\text{out}} \end{bmatrix} \tag{7.17}$$

where $c'_{\text{out}} = 1 - c_{\text{out}}$ and $\mathbf{T}^{\circ}_{\text{out}} = (\mathbf{I} - \mathbf{T}_{\text{out}})\,\mathbf{e}$ for the phase type distribution characterised by probability vector $\boldsymbol{\alpha}_{\text{out}}$ and substochastic matrix $\mathbf{T}_{\text{out}}$.

The durations of the idle (or silent periods) of this process are given by

$$l_n = c_{\text{out}}^{n-1} c'_{\text{out}} \tag{7.18}$$

where l_n describes the probability that the process is idle for exactly n periods. The term c_{out} represents the probability that the queue will be empty immediately prior to a service, given that the queue was empty just before the previous service. Since this can only occur when each source stays in its silent state we have

$$c_{\text{out}} = \prod_{i=1}^{N} c_i \tag{7.19}$$

where c_i represents the equivalent component of $\mathbf{A}_{\text{out}}$ but from source i instead of from the output process. Since, for source i we have

$$\lambda_i = \frac{c_i' \eta_{i,1}}{1 + c_i' \eta_{i,1}} \tag{7.20}$$

rearranging gives

$$c_i = 1 - \frac{\lambda_i}{\eta_{i,1}\left(1 - \lambda_i\right)} \tag{7.21}$$

and hence

$$c_{\text{out}} = \prod_{i=1}^{N} \left(1 - \frac{\lambda_i}{\eta_{i,1}\left(1 - \lambda_i\right)}\right) \tag{7.22}$$

where the first three moments of the idle process are given in terms of c_{out} by

$$\vartheta_{0,1} = \frac{1}{1 - c_{\text{out}}} \tag{7.23}$$

$$\vartheta_{0,2} = \frac{1 + c_{\text{out}}}{\left(1 - c_{\text{out}}\right)^2} \tag{7.24}$$

$$\vartheta_{0,3} = \frac{1 + 4c_{\text{out}} + c_{\text{out}}^2}{\left(1 - c_{\text{out}}\right)^3} \tag{7.25}$$

which follow a geometric development.

Note that equation (7.21) applies equally well to the output process, giving on rearrangement

$$\eta_{\text{out},1} = \frac{\lambda}{\left(1 - \lambda\right)\left(1 - c_{\text{out}}\right)} \tag{7.26}$$

where $\lambda = \lambda_{\text{out}}$ is the average arrival and departure rate from the queue.

7.6.2 Second Moment of the Busy Period

Let M_2 and M_3 denote the second and third moments of the stationary arrival process from the N phase-geom binary sources feeding the target queue. From Appendix C we have

$$M_2 = \lambda + \lambda^2 - \sum_{i=1}^{N} \lambda_i^2 \tag{7.27}$$

and

$$M_3 = \lambda + 3\lambda^2 + \lambda^3 - 3\left(1 + \lambda\right) \sum_{i=1}^{N} \lambda_i^2 + 2 \sum_{i=1}^{N} \lambda_i^3 \tag{7.28}$$

When the contribution from the Bernoulli test source is added to the total arrival process we obtain

$$M_{2_{\text{both}}} = M_2 + \lambda_{\text{test}} + 2\lambda\lambda_{\text{test}} \tag{7.29}$$

and

$$M_{3_{\text{both}}} = \lambda_{\text{test}} + 3\lambda_{\text{test}}\left(\lambda_{\text{test}} + M_2\right) + M_3 \tag{7.30}$$

where $\lambda_{\text{both}} = \lambda + \lambda_{\text{test}}$, and the 'both' subscript indicates that the queue is subject to arrivals from both the N binary sources and the single test source.

The solution for the average queue population $L_{q_{\text{test}}}$ of the test queue is given then as

$$L_{q_{\text{test}}} = L_{q_{\text{both}}} - L_{q_{\text{target}}}$$

where the 'target' subscript indicates the queue being studied, which is subject to arrivals from the N phase-geom binary sources only. The closed form solution for the average queue population of these problems is given by equation (4.11) and results in

$$\begin{aligned} L_{q_{\text{test}}} &= \frac{M_2 + 2\lambda\lambda_{\text{test}} - \lambda}{2\left(1 - \lambda - \lambda_{\text{test}}\right)} - \frac{M_2 - \lambda}{2\left(1 - \lambda\right)} \\ &\quad + \frac{1}{1 - \lambda - \lambda_{\text{test}}} \sum_{i=1}^{N} \lambda_i \left(\lambda + \lambda_{\text{test}} - \lambda_i\right) \frac{\gamma_i}{1 - \gamma_i} \\ &\quad - \frac{1}{1 - \lambda} \sum_{i=1}^{N} \lambda_i \left(\lambda - \lambda_i\right) \frac{\gamma_i}{1 - \gamma_i} \end{aligned} \tag{7.31}$$

The average population of the approximation to the test queue is given by the equivalent measure of a queue subject to arrivals from both the test source and the output process model. Thus from equation (4.11) we obtain

$$L_{q_{\text{approx}}} = \frac{\lambda\lambda_{\text{test}}}{\left(1 - \lambda - \lambda_{\text{test}}\right)\left(1 - \gamma_{\text{out}}\right)} \tag{7.32}$$

where γ_{out} is the autocorrelation parameter of the output model. For the output model to provide exactly the same behaviour as the target queue in terms of the average queue population we require $L_{q_{\text{approx}}} = L_{q_{\text{test}}}$ and hence

$$\frac{\gamma_{\text{out}}}{1 - \gamma_{\text{out}}} = \frac{M_2 - \lambda}{2\lambda\left(1 - \lambda\right)} + \frac{1}{\lambda\left(1 - \lambda\right)} \sum_{i=1}^{N} \lambda_i \left(1 - \lambda_i\right) \frac{\gamma_i}{1 - \gamma_i} \tag{7.33}$$

which is independent of λ_{test} as desired.

Note that if the first moment of the active period for each source is known, then $\eta_{\text{out},1}$ can be obtained from equations (7.26) and (7.22). The second moment of the busy period of the output process is then given by

$$\eta_{\text{out},2} = \eta_{\text{out},1} \left(\frac{1 + \lambda}{1 - \lambda} + \frac{2}{1 - \lambda} \left(\frac{\gamma_{\text{out}}}{1 - \gamma_{\text{out}}} \right) \right) \tag{7.34}$$

7.6.3 Third Moment of the Busy Period

The same method used to calculate the second moment of the busy period of the queue output process can be applied to the queue population variance to obtain the third moment of the busy period. Using equation (4.56) to calculate the variance of the approximated test queue, and equating this result to the variance of the actual test queue yields

$$\begin{aligned}\frac{\eta_{\text{out},3}}{\eta_{\text{out},1}} &= 1-\frac{3\left(1-\lambda-\lambda_{\text{test}}+\lambda^2\right)}{\lambda\left(1-\lambda\right)\left(1-\lambda-\lambda_{\text{test}}\right)}+\frac{3\left(1-\lambda-\lambda_{\text{test}}\right)}{\lambda\lambda_{\text{test}}^2\left(1-\lambda\right)}\operatorname{Var}\left[L_{q_{\text{test}}}\right] \\ &\quad -\frac{3\gamma_{\text{out}}\left(1-\lambda_{\text{test}}\right)}{\lambda_{\text{test}}\left(1-\lambda-\lambda_{\text{test}}\right)\left(1-\gamma_{\text{out}}\right)}+\frac{6\lambda}{\left(1-\lambda\right)^2\left(1-\gamma_{\text{out}}\right)^2} \\ &\quad -\frac{3\gamma_{\text{out}}\left(\lambda+2\lambda_{\text{test}}-\lambda^2-5\lambda\lambda_{\text{test}}-2\lambda_{\text{test}}^2\right)}{\lambda_{\text{test}}^2\left(1-\lambda\right)\left(1-\gamma_{\text{out}}\right)^2} \\ &\quad -\frac{3\gamma_{\text{out}}\lambda\left(1-\lambda\right)}{\lambda_{\text{test}}^2\left(1-\lambda-\lambda_{\text{test}}\right)\left(1-\gamma_{\text{out}}\right)^2} \qquad (7.35)\end{aligned}$$

where

$$\operatorname{Var}\left[L_{q_{\text{test}}}\right]=\operatorname{Var}\left[L_{q_{\text{both}}}\right]-\operatorname{Var}\left[L_{q_{\text{target}}}\right]-\operatorname{Cov}\left[L_{q_{\text{test}}},L_{q_{\text{target}}}\right] \qquad (7.36)$$

which in general means that $\operatorname{Var}\left[L_{q_{\text{test}}}\right]$ will be unknown, since although the variance terms on the right are known in closed form, we have not studied the covariance component.

We expect that further study of this problem will result in a closed form expression for this covariance term, possibly from a similar approach as used to obtain the single buffer variance expression.

Given that such a solution becomes available, the resulting expression for equation (7.35) would be quite complicated, so that numeric evaluation and substitution of $\operatorname{Var}\left[L_{q_{\text{test}}}\right]$ becomes the most logical approach to use, particularly since it is likely that $\operatorname{Var}\left[L_{q_{\text{target}}}\right]$ will be required anyway. This approach obviously requires a suitable choice of λ_{test} in order to calculate $\operatorname{Var}\left[L_{q_{\text{both}}}\right]$ although in fact the actual solution for the third moment must be independent of the value of λ_{test} chosen.

7.6.4 Implications for Phase-Geometric Binary Queues

Since the merged output of a queueing system fed by some number of phase-geom binary processes is also a phase-geom binary process, the behaviour of a network of queues in which there is no splitting can be completely known. An example of this is a network structured as a directed tree, where all the network traffic is directed towards the head of the tree (see Figure 7.20).

If the parameters λ, η_2/η_1, and η_3/η_1 are known for every phase-geom binary process entering the network, the equivalent parameters of all the internal network processes at the outputs of the various queues can be calculated as discussed in the preceding sections. Then since the average and variance of the population at each of the queues is known in closed form (chapter 4), the behaviour of the entire queueing system can be

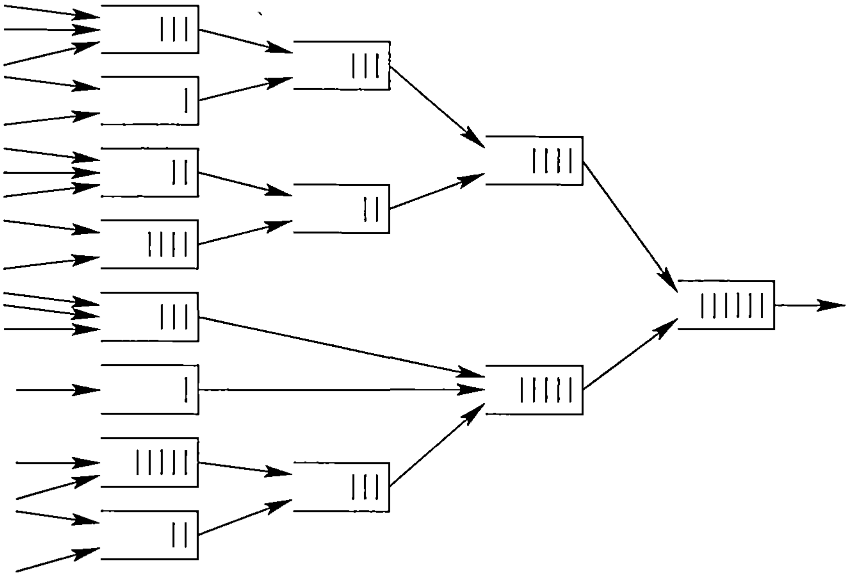

Figure 7.20: *An example of a tree-type network of queues. All traffic in the queueing system is directed towards the 'head' of the tree (the queue on the far right in the diagram) in such a way that only merging operations occur.*

described exactly based on knowledge of the input processes alone. Pieloor and Lewis presented an example of this is [107], although only for the population average. Addie and Zukerman also considered this type of problem in [4], although using a Gaussian traffic model rather than a phase-geom binary one.

If the network edge processes are geom-geom IBP processes rather than phase-geom binary processes, then it will still be possible to construct the network solution in terms of the average queue populations, since we have shown that it is possible to construct an IBP model for the output of a queue fed by geom-geom IBP processes that is quite accurate in the average queue population. In addition, since θ_{model} for the output of the IBP queues will tend towards 1 as the number of queues traversed increases, only those queues subject to arrivals from network entry traffic will incur a high computational cost. The others can be treated as having only geom-geom (or phase-geom) binary arrivals.

Unfortunately, directed tree networks of this sort are not very practical for solving general networking problems, which means that this particular property of the phase-geom binary process is much more interesting than it is useful.

7.7 Summary

This chapter has considered some of the issues relating to the application of the main queueing theory results developed in this thesis to the performance analysis of ATM switches. In particular we have looked at correlation effects in the arrival processes for prioritised output queueing systems. The arrival processes to the queue buffers within the switching system can be described either by identifying and describing the individual traffic types traversing this switch element, or by applying a splitting rule to some traffic model that describes the traffic on the inputs of the switching system. Of the two methods, the first is perhaps more common, but is essentially flawed in that it disregards the correlations between arrivals from each of the various traffic classes.

In chapter 5 it was assumed that the cyclic process describing the high priority arrivals in a dual buffer system was independent of the low priority arrivals, described using geom-geom IBP sources. This is an example of the first method mentioned above, and in section 7.2 we looked at how this assumption leads to overestimation of the queueing performance when compared to a more realistic description of the arrival processes. By more realistic, we mean one where the presence of a high priority arrival on any input line to the switch precludes a low priority arrival on the same input line.

In contrast, splitting and priority splitting models automatically incorporate this anti-correlation behaviour by assigning an arrival on an input line to a single output buffer. As an example of this type of approach, a simple ATM switch using random splitting and random priority was investigated in section 7.3. The single buffer queueing theory developed in chapters 2 to 5 is applied using Corollary 2.2 of Theorem 2.1 to provide the average queue population of the high and low priority queues, but can only provide an exact result for the high priority queue population variance. For the low priority queue, only a loose upper bound on the variance is provided by the relevant single buffer theory.

In the second part of this chapter we looked at how Corollaries 2.2 and 2.3 of Theorem 2.1 can be used to help assign parameters to models for the aggregate output process of a single buffer queueing system. The basic assumption behind the approach is to match the effect on the average and variance of the population of 'test' queue subject to arrivals from the model to that of the actual output process. To describe possible mixing effects in this test queue and to avoid non-zero population results, a Bernoulli test source was superposed with the model and output process.

For a suitable model of the output process, the model parameters obtained using this approach will be independent of the Bernoulli source parameters. As an initial example, in section 7.5 we consider a geom-geom IBP as a model for the output process of a queue fed by a number of geom-geom IBP processes. The results show that the model parameters are effected by the Bernoulli source arrival rate, although use of a result by Daley [19] can minimise this effect as far as the average queue population is concerned. The geom-geom IBP model, having only two states was unable to provide a good matching for the variance except as this quantity became extremely large. Support for the use of a geom-geom Binary process as the output model was also provided.

As a further example of the parameter matching technique introduced in this chapter, we obtained in section 7.6 the first three moments of the idle and busy periods for a server subject to arrivals from phase-geom binary processes. The first and second moments of the busy period were shown explicitly to be independent of the Bernoulli test source's arrival rate.

Chapter 8

Conclusion

This thesis has presented numerous results relating to the discrete-time performance analysis of queueing systems with application to ATM based networks. In this brief chapter we present a summary of the content of the work, a few topics for future research, and some conclusions.

8.1 Thesis Summary

Chapter 1 provided a basic overview of the B-ISDN, and the ATM. In these networks, connection admission control mechanisms are used to maximise the utilisation of resources, while guaranteeing that customers receive their negotiated service quality. In order to perform this function, the CAC requires a means of predicting the performance of the network in quality of service terms. Due to the nature of buffering within the ATM switching elements, this task is accomplished using queueing analysis.

After a discussion of the some of the available analysis methods, some basic theory relating to the discrete-time queue population analysis of an infinite buffer queue was developed in Chapter 2. Consideration was also given to the analysis of dual buffer, and interrupted service queueing systems, showing how these were related to the single buffer analysis.

In Chapter 3, the queueing theory developed in Chapter 2 was applied to a queue with arrivals from a heterogeneous mix of geom-geom IBP sources. The evaluation of the average and variance of the queue population requires the solution of a linear system of equations, with a number of unknowns that grows exponentially with the number of sources. The relevant calculations can be performed very quickly for a small number of sources, however computation time and memory requirements limit the maximum number of sources to about 12 for practical purposes. As an alternative to the exact solution, several approximations based on methods proposed in the literature were discussed, and their accuracy investigated using queueing problems with randomly generated parameters, across a wide range of utilisations. Although a couple of the methods performed adequately at high utilisations, none of the approximations performed well at lower utilisations. A new approximation method was then proposed, which achieves excellent accuracy at all utilisations. In fact the new method can be used to provide 'near exact' results in situations where the formally exact solution cannot be applied.

Chapter 4 was concerned with the queueing behaviour of phase-geom Binary processes, for which a well known closed form solution for the average queue population exists. By making use of the solution method used to obtain this average value, and the theory developed in Chapter 2, a previously unknown, closed form solution for the variance was obtained.

In response to an observation in Chapter 1 that high priority traffic sources within an ATM network that provides delay priority may exhibit cyclic behaviour, Chapter 5 dealt with queues receiving cyclically interrupted service. Using the interrupted service result of Chapter 2, the average and variance of the cyclic service queue were actually obtained by considering a cyclic arrival process. Since numerical difficulties were encountered in establishing the solution to the system of linear equations required by the population results, an innovative adaptive technique was presented that allows accurate estimations for the average and variance to be obtained even when the exact solution method fails. Three approximation solution methods were also investigated, with the new approximation of Chapter 3 shown to provide the best result accuracy.

Chapters 2 to 5 dealt only with the queue population of infinite buffer queueing systems. Since real systems have finite buffer capacities, and experience losses due to buffer overflow, Chapter 6 considered how the average loss probability for a finite buffer system could be calculated from knowledge of its infinite buffer behaviour. A simple relationship was proven between the tail distribution of the population for an infinite buffer queue with marginal arrivals, and the loss experience by the equivalent finite buffer. Although the result does not hold exactly for queues with autocorrelated arrivals, experimental results suggested the error was acceptably small. The accuracy of several approximations to the tail distribution in terms of predicted losses was then investigated, and an approximation method obtained from the literature shown to provide good performance. The relationship of the single buffer loss probability to the loss probabilities of each buffer in a dual buffer system was also discussed.

Queueing delays are another measure of service quality, and the relationship between the average and variance of the queueing delay and the equivalent queue population measures was also considered in Chapter 6. For a single buffer queueing system with uninterrupted service, the relationship is straightforward, and applies equally well to the finite buffer case. For the low priority buffer in a dual buffer system, or a queue with interrupted service, Little's result applies to the average queueing delay. However the relationship for the delay variance is not well defined. Chapter 6 concluded with an example calculation for the delay variance of a queue with marginal arrivals and marginal service.

The last chapter of the main section of this thesis, Chapter 7, began with a consideration of the correlation effects due to splitting and priority splitting within the ATM switching element. For the case where the high priority arrival process in a dual buffer system are cyclic in nature, the theory of Chapter 5 was shown to cause an overestimation in the population average and variance by not taking these correlations into account. The same problem was shown not to arise when only the geom-geom IBP model is used.

The role that modelling the output of a queueing system plays in developing interconnecting networks of queues was then discussed, and a new approach to matching the parameters of a model to the actual output process presented. The geom-geom IBP model was shown to provide a poor match for the output process of a queue fed by geom-geom IBP sources. However, when the sources are phase-geom Binary, the output process was shown to also be phase-geom Binary, with parameters that could be directly calculated from those of the arrival processes.

8.2 Future Directions

In any research work of this nature, there are topics and areas of investigation that are identified, but that lie beyond the scope of the current task. Three such topics are discussed below.

8.2.1 Developing Multi-State IBP Arrival Models

One of the limitations of the geom-geom IBP traffic models is that it is unable to adequately capture variance effects at the output of a queue (see Chapter 7), and hence may be misrepresenting the arrival process at the input of the next queue. To overcome this problem, IBP models with three or more states would be required.

In [126], Steyaert and Xiong develop a probability generating function analysis for the population of a buffer subject to arrivals from a number of independent but identical three-state IBP sources, from which they obtain an expression for the population average. Even in a homogeneous environment, this problem does not appear particularly tractable. An additional problem is the number of parameters required to describe each source — Steyaert and Xiong's model requires eight.

The solutions for the average and variance of the queue population developed in Chapter 2, can be easily adapted for queues with arrivals from heterogeneous three-state IBP sources. However the number of states in the empty system probability vector is 3^N for N sources, which will restrict the maximum number of sources to only about 7 or 8. An additional complication arises if the eigenvalues of the models become complex, although this can be avoided by modifying the structure of the model's transition matrix[1].

To overcome or reduce these numerical difficulties, the approximation method proposed in Chapter 3 could be used to provide good approximations to the exact results. As in that chapter, an extensive investigation of the accuracy of the approximation could be performed to provide guidelines for practical application. Although this author expects somewhat reduced accuracy due to the possibility of long tailed autocorrelations, the improvement in calculation speed will be even more pronounced than for the geom-geom IBP case — run times of approximately $O\left(27^N\right)$ for the exact solution versus $O\left(N^2\right)$ or $O\left(N^3\right)$ for the approximation.

In order to actually make use of the three-state model in a network of queues (as would be the case for CAC) it is important to be able to establish the parameters to model the queue output process, since this is the reason given above for making use of models with more than two states. Methods for doing this will need to be explored.

8.2.2 Extending the Dual Buffer Analysis

An ATM network will only be able to provide good service to delay sensitive and loss sensitive traffics at high levels of network utilisation if preferential service is available in the ATM switches. In Chapter 2 we saw how the single and dual buffer realisations of an output queueing switch are related in terms of their average queue population, but were unable to provide the equivalent result for the variance due to the presence of a covariance term. Further development is required here to complete the population relationship between the two queueing systems. The result is important in particular for the purpose of modelling the output process of a queueing system (see Chapter 7)

The variance of the delay in an interrupted service queue is also an area for further

[1] Which would have the additional benefit of reducing the number of parameters per model.

study. In Chapter 6 we obtained the solution for random arrivals and randomly interrupted service (the simplest case) but more complicated arrival processes have not yet been investigated. In addition, use of the delay average and variance for estimating high percentiles of the delay can be considered, since it is likely that the geometric tail property will not apply to the delay in this situation.

8.2.3 Implementing Performance Results into a CAC Framework

The primary motivation for the queueing performance results developed in this thesis is the role that these play in connection admission control for ATM networks. The analytical approach has considered only a single queueing node however, and although solutions for quality of service parameters (delay and loss) have been presented, their implementation in a wider CAC framework has been not been investigated.

The models of ATM traffic used here have been chosen on the basis of the use of similar models in the literature, computational tractability, and a somewhat less than thorough understanding of the merging and splitting processes occurring in the network. In order to establish the *practical* accuracy, or otherwise of these models, an investigation is required into the overall performance of a simulated or actual network that implements CAC using the results from these models. If the network is unable to consistently meet QoS requirements (too many connections are being accepted) or network utilisation requirements (too few connections accepted) then the traffic models will be known to be inadequate. In either event, compensation mechanisms could then be explored.

Such a CAC framework could also be used to explore the ability of other admission methods, such as peak rate or equivalent bandwidth allocation schemes, to meet the quality and usage objectives of the network.

8.3 Conclusions

In this work we have presented exact numerical solutions for the population average and variance of two single buffer queueing systems, and exact closed form solutions for a third. Due to computational limits in the exact numerical solutions, a number of approximate methods, including a newly proposed method, were investigated for accuracy. We also discussed the relationship between the queue population and queueing delays, and considered methods for estimating loss probabilities in finite buffer systems. The application of these results to dual buffer queueing systems, which are able to provide delay priority service in an ATM network, was also investigated.

In the context of connection admission control, we found no means of accurately predicting queueing delays that could be performed within the time frame that might be required for on-line calculations. A fairly accurate method for calculating average loss probabilities, based on a result from the literature and suitable for on-line use, was presented however. For other applications where on-line calculation is not required, the new approximation method proposed in Chapter 3 is eminently suitable, since it provides accurate results for queue populations and delays to be obtained where exact numerical methods could not normally be used. In addition, the basic structure of the method suggests that it could be applied to the analysis of a wide range of arrival processes.

Providing practical methods for implementing connection admission control for ATM networks has been the driving force for this research, and in this regard several important contributions have been made. However, a great deal more investigation will be required before this target can be fully realised.

Appendix A

Application of the z-Transform to the Marginal Arrivals Problem

In the body of this thesis, the z-transform is used to determine various queue population statistics for autocorrelated arrival processes. In this appendix we will show how the discrete-time z-transform is applied to analyse an infinite buffer queue fed by marginal arrivals — that is, where the probability distribution of the number of arrivals occurring in the current time slot is fixed, and independent of the behaviour of any other component of the queueing system.

A.1 The discrete z-transform

The discrete z-transform of a random variable $\mathrm{x} = \{x_n\}$ is given by [43]

$$x(z) = \sum_{n=0}^{\infty} x_n z^n \tag{A.1}$$

If x is a probability distribution, then

$$\begin{aligned} x(1) &= 1 && \text{(A.2)} \\ x'(1) &= m_1(\mathrm{x}) && \text{(A.3)} \\ x''(1) &= m_2(\mathrm{x}) - m_1(\mathrm{x}) && \text{(A.4)} \\ x'''(1) &= m_3(\mathrm{x}) - 3m_2(\mathrm{x}) + 2m_1(\mathrm{x}) && \text{(A.5)} \end{aligned}$$

where $m_r(\mathrm{x})$ is the rth moment of the distribution $\mathbf{x}$. These very useful properties will be exploited in the application of the z-transform to discrete-time queueing problems as described below.

A.2 The Marginal Arrivals Queue

We start by assuming that time is divided into equally sized units called *slots*. Each slot time is equal to the time required to service or remove from the queue one waiting arrival (called *cells* in the ATM nomenclature). The time ordering of events in the queueing system is such that service of the queue occurs at slot boundaries, while arrivals to the queue occur during the slot times. In this discussion we are only concerned with the number of cells awaiting service and not the cell (if any) that is currently receiving service. That is, we are interested in the queue population, not the entire system population.

Let the distribution of the queue population at the start of a time slot be described by $\mathbf{q} = \{q_n\}$ where q_n denotes the stationary probability that the queue population is n. Similarly, let $\mathbf{q}^+ = \{q_n^+\}$ describe the same quantity at the end of a time slot (after any arrivals have occurred, but before the next service). Additionally let $\mathbf{p} = \{p_k\}$ describe the stationary distribution for the arrival process, where p_k is the probability that there will be k arrivals to the queue in any time slot. For convenience we will denote the first three moments of the arrival process by λ, M_2, and M_3 which are the notation used throughout this thesis.

The relationship between $\mathbf{q}$ and $\mathbf{q}^+$ (the arrival relation) is given by

$$q_n^+ = \sum_{i=0}^{n} q_i p_{n-i} \tag{A.6}$$

which, upon taking z-transforms of both sides and applying Theorem F.4, can also be written as

$$q^+(z) = q(z)p(z). \tag{A.7}$$

where $q(z)$, $q^+(z)$, and $p(z)$ are the z-transforms of $\mathbf{q}$, $\mathbf{q}^+$, and $\mathbf{p}$ respectively. Due to the regular services of the queueing system we have a second relation between the $\mathbf{q}$ and $\mathbf{q}^+$ (the service relation) of

$$q_n = \begin{cases} q_0^+ + q_1^+ & \text{for } n = 0 \\ q_{n+1}^+ & \text{otherwise} \end{cases} \tag{A.8}$$

or in z-transform notation,

$$zq(z) = q^+(z) + (z-1)\, q_0^+ \tag{A.9}$$

Combining equations (A.7) and (A.9) then gives

$$(z - p(z))\, q(z) = (z-1)\, q_0^+ \tag{A.10}$$

where q_0^+ is the probability that the queue is empty immediately prior to service.

The first derivative of equation (A.10) evaluated at $z = 1$ gives

$$q_0^+ = 1 - p'(1) \tag{A.11}$$

where $p'(1) = \lambda$ is the average arrival rate to the queue. We can also arrive at this same conclusion from an equilibrium point of view by considering that the probability that there is at least one queued arrival immediately prior to service is equal to the probability that there is a departure from the system in that time slot. That is, the average departure rate is equal to $1 - q_0^+$ and since the queueing system is work conserving and has infinite capacity this must also be equal to the average arrival rate to the queue. Obviously from equation (A.11) the average arrival rate must also be less than or equal to 1, and for stationary behaviour to exist we must also have λ strictly less than 1.

From the second and third derivatives of equation (A.10) we obtain (with some manipulation)

$$\begin{aligned} L_q &= q'(1) \\ &= \frac{M_2 - \lambda}{2(1-\lambda)} \end{aligned} \tag{A.12}$$

and also

$$\begin{aligned} \mathrm{Var}[L_q] &= q''(1) + L_q - L_q^2 \\ &= \frac{4(1-\lambda)M_3 + 3M_2^2 - 6M_2 + \lambda^2 + 2\lambda}{12(1-\lambda)^2} \end{aligned} \tag{A.13}$$

where L_q denotes the average queue population, and $\mathrm{Var}[L_q]$ denotes its variance. Equation (A.12) may also be written as

$$L_q = \frac{\sigma^2 + \lambda^2 - \lambda}{2(1-\lambda)} \tag{A.14}$$

where σ^2 represents the variance of the number of arrivals occurring per time slot.

Appendix B

Queueing Delays in a Shared Buffer Environment

In this appendix we will show that, given knowledge of the distribution of the population of a shared buffer queueing system, upper and lower limits can be placed on the average queueing delays experienced by individual classes of traffic. That is, the best case and worst case average queueing delays can be established, with the actual queueing delay for a particular class falling midway between these two quantities. A brief discussion of limits for the variance of the queueing delays is also included.

An infinite buffer is assumed for simplicity of the argument, but the same general approach could also be used to obtain limits in the case of a finite buffer. The other important assumption is that the queue receives uninterrupted service (at a rate of one queued arrival removed from the buffer at the beginning of each time slot). Thus an arrival that sees n previously queued arrivals ahead of it, will be at the head of the queue (ready for the next service) n time slots later, and hence has a queueing delay of n time slots.

B.1 Marginal Arrival Processes

Denote the probability that the buffer contains n queued arrivals immediately after the last service by q_n. Let $p_{k,c}$ denote the probability that there will be k arrivals from class c in the current time slot before the next service of the queue. The average number of arrivals per time slot of class c is denoted by λ_c, with variance denoted by σ_c^2. The average number of arrivals from all the classes is denoted by λ and is given by the sum of the individual λ_c.

Denote the average delay for arrivals from traffic class c when the queue population is n by $d_{c,n}$. The best queueing delay for class c traffic (or the minimum value of $d_{c,n}$) occurs when all the new arrivals from that class are queued ahead of any *new* arrivals from the other classes. This is called giving the class *arrival priority.* All arrivals in the current time slot are queued behind those in the previous, so the arrival priority merely determines the position of a class amongst other simultaneous arrivals. Since the actual number of arrivals does not change, and since the service time of each class is equal, the use of arrival priority queueing will not change the steady state behaviour of the queue.

Let k_c denote the number of arrivals from traffic class c in the current time slot. Then assuming that class c has the highest arrival priority, these k_c arrivals contribute a total of $k_c(k_c-1)/2$ service periods in addition to the queueing delay required for the first of these new arrivals to reach the head of the queue. That is

$$\begin{aligned}\min\left[d_{c,n}\right] &= n+\frac{1}{2\lambda_c}\sum_{k=1}^{\infty}k(k-1)\,p_{k,c}\\ &= n+\frac{\sigma_c^2}{2\lambda_c}-\frac{1-\lambda_c}{2} \qquad \text{(B.1)}\end{aligned}$$

and so

$$\min\left[D_{q,c}\right]=L_q+\frac{\sigma_c^2}{2\lambda_c}-\frac{1-\lambda_c}{2} \qquad \text{(B.2)}$$

where $D_{q,c}$ denotes the average queueing delay for the class c traffic, and L_q is the average queue population.

The worst case queueing delay for class c is when it has the lowest queueing arrival priority, and all new arrivals of this class are queued behind new arrivals from the other classes. Since, arrivals from each class are assumed to be independent of each other, this means that there will be on average $\lambda-\lambda_c$ new arrivals to the queue ahead of the arrivals of class c. Thus

$$\max\left[D_{q,c}\right]=L_q+\lambda+\frac{\sigma_c^2}{2\lambda_c}-\frac{1+\lambda_c}{2} \qquad \text{(B.3)}$$

This discussion requires that the arrival processes are marginal in the current time slot only. The reasoning therefore applies to cyclic service and arrival problems, and can be extended to the case of autocorrelated arrival processes, as discussed in the following.

B.2 Autocorrelated Arrival Processes

We consider here the special case where each traffic class generates arrivals according to a discrete-time Markov modulated arrival process (usually called a batch Markov arrival process or D-BMAP) with countable states, wherein each state generates arrivals according to some state and class dependant marginal probability distribution.. The generating processes of each of the individual classes are then combined to create an overall arrival process which will also be a D-BMAP. These types of models can also describe peak rate limited sources.

Without loss of generality, we assume that the time order of events within a time slot is such that the D-BMAP changes state, arrivals are generated according to the new D-BMAP state, and the queue is then serviced. Denote the state transition probabilities of the combined arrival process by $\alpha_{r,s}$, which describes the probability that the D-BMAP will change to state s in the current time slot, given that the last state was r. Let $q_n(r)$ denote the probability that the queue population was n immediately after the queue was serviced, when the last D-BMAP state that generated arrivals was r. Similarly, let $x_n(s)$ denote the probability that the D-BMAP changes to state s in the current time slot from any previous state, and that the queue population immediately after the last service was n, so that

$$x_n(s) = \sum_{r=0}^{m-1} q_n(r)\alpha_{r,s} \tag{B.4}$$

where m is the number of states in the combined Markov arrival process.

Let $p_{k,c}(s)$ denote the probability that class c generates k arrivals in each time slot that the combined arrival process is in state s. The average and variance of the number of arrivals generated by class c in state s are denoted by $\lambda_c(s)$ and $\sigma_c^2(s)$ respectively. Over all of the states, as for the marginal case above, class c generates an average of λ_c arrivals per time slot, with variance σ_c^2, where

$$\lambda_c = \sum_{s=0}^{m-1} \mu_s \lambda_c(s) \tag{B.5}$$

$$\sigma_c^2 = \sum_{s=0}^{m-1} \mu_s \sigma_c^2(s) + \sum_{s=0}^{m-1} \mu_s \lambda_c^2(s) - \lambda_c^2 \tag{B.6}$$

and where μ_s is the stationary probability that the arrival process is in state s. In addition, the quantities $\lambda(s)$ and λ denote the average number of arrivals per time slot from all traffic classes, conditioned on the arrival process state, and independent of the arrival process state respectively.

Within each time slot, arrivals are generated according to a marginal process that is described by the current state of the combined D-BMAP, and hence the deductions of section B.1 can be applied. Assuming that class c traffic has the highest queueing arrival priority gives

$$\min\left[d_{c,n}(s)\right] = n + \frac{\sigma_c^2(s)}{2\lambda_c(s)} - \frac{1-\lambda_c(s)}{2} \tag{B.7}$$

and hence

$$\min\left[D_{q,c}(s)\right] = \frac{1}{\mu_s}\sum_{n=0}^{\infty} n x_n(s) + \frac{\sigma_c^2(s)}{2\lambda_c(s)} - \frac{1-\lambda_c(s)}{2} \tag{B.8}$$

In order to determine $D_{q,c}$ we note that

$$D_{q,c} = \frac{1}{\lambda_c}\sum_{s=0}^{m-1} \mu_s \lambda_c(s) D_{q,c}(s) \tag{B.9}$$

since $D_{q,c}(s)$ represents the average delay seen by arrivals from class c in state s, and there are proportionally $\mu_s\lambda_c(s)/\lambda_c$ arrivals generated by that state. Hence

$$\min\left[D_{q,c}\right] = \frac{1}{\lambda_c}\sum_{r=0}^{m-1}\left(\sum_{s=0}^{m-1}\lambda_c(s)\alpha_{r,s}\right)L_q(r) + \frac{\sigma_c^2}{2\lambda_c} - \frac{1-\lambda_c}{2} \tag{B.10}$$

where $L_q(s)$ denotes the average queue population observed immediately after service when the last state of the arrival process was s. The overall queue population L_q is given by the sum of the individual $L_q(s)$. The maximum or worst case queueing delay is determined in a like fashion to the minimum case above, and yields

$$\begin{aligned}\max\left[D_{q,c}\right] &= \frac{1}{\lambda_c}\sum_{r=0}^{m-1}\left(\sum_{s=0}^{m-1}\lambda_c(s)\alpha_{r,s}\right)L_q(r) + \frac{\sigma_c^2}{2\lambda_c} - \frac{1-\lambda_c}{2} \\ &\quad + \frac{1}{\lambda_c}\sum_{s=0}^{m-1}\mu_s\lambda_c(s)\left(\lambda(s)-\lambda_c(s)\right)\end{aligned} \tag{B.11}$$

The difficulty with the above limit expressions is that to evaluate them, the value of each $L_q(r)$ is required. Some solution methods (such as the iterative numeric approach of Appendix E) will provide these individual $L_q(r)$ and hence allow the limits to be found. However, the closed form solution for the average queue population using phase-geom Binary sources in Chapter 4 only provides the overall L_q value.

B.3 Some Example Applications

It is of interest to determine what parameters give a class of traffic the best average queueing delay (that is, the smallest upper limit) in a shared buffer environment. This is easily determined for marginal arrival processes, where the best performance goes to the source with the least (closest to $-\infty$) value of δ_c where

$$\delta_c = \frac{\sigma_c^2}{2\lambda_c} - \frac{1+\lambda_c}{2} \tag{B.12}$$

which means qualitatively that the traffic class with largest average and smallest variance will receive the best average queueing delay performance. Also, if class c is a binary source, δ_c becomes equal to $-\lambda_c$ which is the least for the traffic class with largest average number of arrivals per time slot.

To investigate what parameters effect the average queueing delay of autocorrelated sources, we will consider three examples. The first example investigates the application of the closed form solution for a single geom-geom source mixed with marginal traffic. The second example uses the iterative numeric approach to investigate the performance seen by a single geom-geom binary source when it is mixed with several other geom-geom binary sources. The third example is similar to the second, but replaces the single geom-geom source with a periodic one.

Note that binary sources have easily exploited properties, resulting in simpler forms for the limit equations (B.10) and (B.11).

B.3.1 Mixing a Geometric-Geometric Source with a Marginal Source

Consider a shared infinite buffer in which arrivals from a total of N traffic classes are statistically multiplexed. Traffic class 1 is described by a geom-geom binary source, while the remaining $N-1$ traffic classes are described by a simple Bernoulli arrival process. The average number of arrivals per time slot from all the traffic classes is denoted by λ with variance σ^2.

From section B.2, the limits of the average queueing delay for traffic class 1 are obtained from equations (B.10) and (B.11) using

$$\begin{aligned} \sigma_1^2 &= \lambda_1 (1-\lambda_1) \\ \lambda_1(s) &= \begin{cases} 0 & \text{for } s=0 \\ 1 & \text{otherwise} \end{cases} \\ \lambda(s) &= \lambda - \lambda_1 + \lambda_1(s) \end{aligned}$$

and also that $m=2$ and $\mu_1 = \lambda_1$, and $L_q = L_q(0) + L_q(1)$, to give

$$\min\left[D_{q,1}\right] = \frac{1}{\lambda_1}\left(\left(1-\alpha_{1,0}\right)L_q + \left(\alpha_{1,0}-\alpha_{0,0}\right)L_q(0)\right) \tag{B.13}$$

and

$$\max\left[D_{q,1}\right] = \min\left[D_{q,1}\right] + \lambda - \lambda_1 \tag{B.14}$$

In addition, note that $\alpha_{0,0}$ and $\alpha_{1,0}$ can be written as

$$\begin{aligned} \alpha_{0,0} &= (1-\lambda_1) + \lambda_1\gamma_1 \\ \alpha_{1,0} &= (1-\lambda_1)(1-\gamma_1) \end{aligned}$$

where γ_1 is the autocorrelation parameter for the geom-geom source (see section C.5).

From equation (4.11) in Chapter 4, we have that

$$L_q = \frac{\sigma^2 + \lambda^2 - \lambda}{2(1-\lambda)} + \frac{1}{1-\lambda}\lambda_1 (\lambda - \lambda_1) \frac{\gamma_1}{1-\gamma_1} \tag{B.15}$$

and from equation (4.46)

$$L_q(0) = (\lambda - \lambda_1)(L_q - \lambda_1) + \frac{\sigma^2 + \lambda^2 - \lambda}{2} \tag{B.16}$$

so that substitution and simplification yields

$$\min[D_{q,1}] = L_q + (\lambda - \lambda_1)\frac{\gamma_1}{1-\gamma_1} \tag{B.17}$$

with the maximum being given simply in terms of $\min[D_{q,1}]$ by equation (B.14).

From the form of equation (B.17) it is readily apparent that a geom-geom binary source with $\gamma > 0$ will obtain worse queueing delay performance than an equivalent Bernoulli source, while $\gamma < 0$ should provide better performance. To illustrate this result, consider an example shared buffer queueing problem having just 2 binary sources, each with an average arrival rate of $\rho/2$, where ρ is the utilisation of the queue. The first source is modelled as a geom-geom type binary process, while the other source is modelled as a simple Bernoulli process. The equations for the limits of the average queueing delay then become

$$\min[D_{q,1}] = L_q + \frac{\rho\gamma_1}{2(1-\gamma_1)} \quad \text{and} \quad \max[D_{q,1}] = L_q + \frac{\rho}{2(1-\gamma_1)} \tag{B.18}$$

where L_q is given by

$$L_q = \frac{\rho^2}{4(1-\rho)(1-\gamma_1)} \tag{B.19}$$

Figure B.1 shows the theoretical limits and simulated average queueing delay of each of the two sources as a function of γ_1 for a utilisation of 0.9. The 99% confidence interval for the simulation results is less than ±0.5% in each case. Note that the range of the autocorrelation parameter includes an area where $\gamma_1 < 0$, in which the autocorrelated source sees better queueing performance than the Bernoulli one. This suggests that negatively autocorrelated binary processes (such as periodic sources) will 'see' better performance than Bernoulli or positively autocorrelated sources. The periodic case will be discussed as an example a little later.

Because the queueing delays are also affected by the increase in the average queue population as γ_1 increases, it is difficult to see exactly where the actual queueing delays lie relative to the calculated limits. Figure B.2 shows the results of Figure B.1 normalised with respect to the overall average queueing delay of L_q/λ. The simulation results are clearly at the midway point of the predicted upper and lower limits, which suggests that the actual queueing delay can be predicted accurately from the limits. This point will be discussed again later.

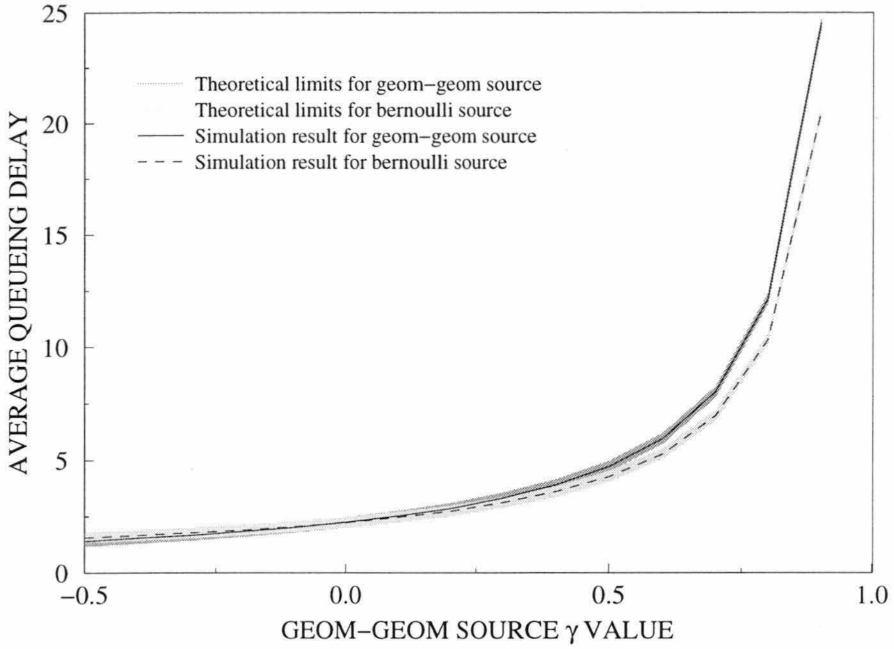

Figure B.1: *Average queueing delay (theoretical limits and simulation results) for two equal rate binary sources, only one of which is autocorrelated with parameter γ_1 given by the independent variable in the figure. The queue utilisation is* 0.9.

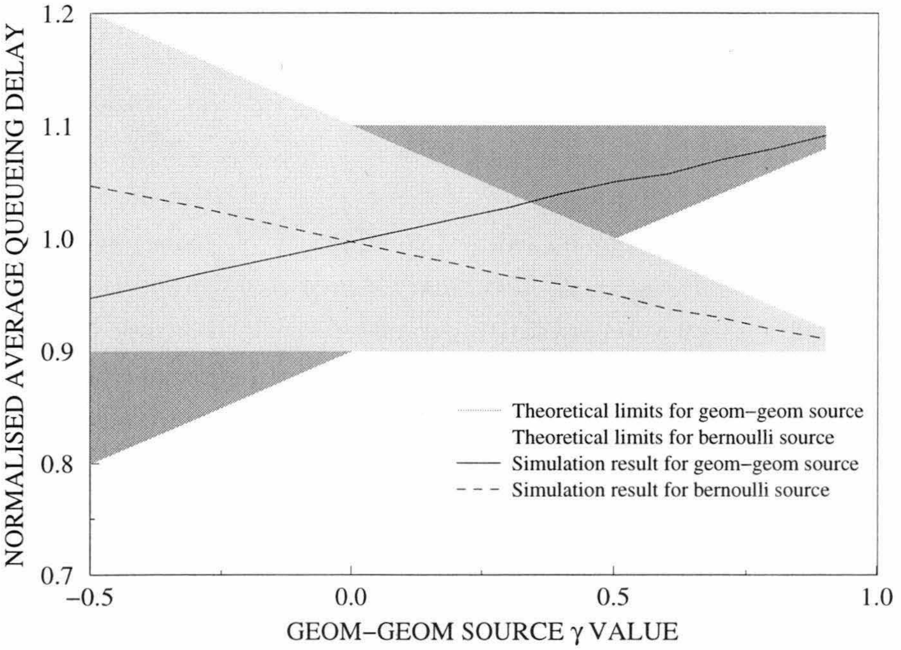

Figure B.2: *Queueing delay results of Figure B.1 normalised to the overall average queueing delay.*

B.3.2 Mixing Geometric-Geometric Sources

Consider a shared infinite buffer in which arrivals from a total of N traffic classes are statistically multiplexed, where each traffic class is described by a single geom-geom binary source. Under this condition we get

$$\min\left[D_{q,c}\right] = \frac{1}{\lambda_c} \sum_{r=0}^{m-1} \left(\sum_{s=0}^{m-1} \lambda_c(s) \alpha_{r,s} \right) L_q(r) \tag{B.20}$$

and

$$\max\left[D_{q,c}\right] = \min\left[D_{q,c}\right] + \frac{1}{\lambda_c} \sum_{s=0}^{m-1} \mu_s \lambda_c(s) \left(\lambda(s) - \lambda_c(s)\right) \tag{B.21}$$

where $\lambda_c(s)$ takes on the value of 1 when $s \in \Theta_c$ and 0 otherwise, for a set Θ_c which specifies those states of the overall D-BMAP in which traffic class c is active (generates a single arrivals). In fact, using this set relation, its is possible to show that

$$\max\left[D_{q,c}\right] = \min\left[D_{q,c}\right] + \lambda - \lambda_c \tag{B.22}$$

is a relation that always holds when using binary sources, regardless of their autocorrelation structure.

When a class is made up of some number S_c of identical binary sources, it is fairly straightforward to show that the lower limit on the average queueing delay becomes

$$\min\left[D_{q,c}\right] = \frac{1}{\lambda_c} \sum_{r=0}^{m-1} \left(\sum_{s=0}^{m-1} \lambda_c(s) \alpha_{r,s} \right) L_q(r) + \frac{\lambda_c}{2} \left(1 - \frac{1}{S_c}\right) \tag{B.23}$$

where, with a little more difficulty we can also show that the upper limit is still given by equation (B.22).

As an example, consider a buffer fed by 4 equal rate geom-geom binary sources, in two classes of 1 and 3 sources each. The binary source belonging to the first traffic class has $\lambda_1 = 0.2$, with an autocorrelation parameter γ_1 which is varied from -0.2 to 0.9 in steps of 0.1. The other three sources are identically distributed with average arrival rates of 0.2 (giving $\lambda_2 = 0.6$) and autocorrelation parameters of $\gamma_2 = 0.3$.

Figures B.3 and B.4 show the calculated limits for each of the two classes along with the actual average queueing delays obtained from simulation of the queueing problem. As before, the 99% confidence interval for the simulation results is less than ±0.5% for each case. The upper and lower average queueing delay limits were calculated by performing an iterative solution to the queueing problem in order to obtain the $L_q(s)$ values used in the equations above. These theoretically predicted limits were confirmed by simulation, by firstly giving traffic class 1 arrival priority, and then by giving the traffic class 2 arrival priority[1].

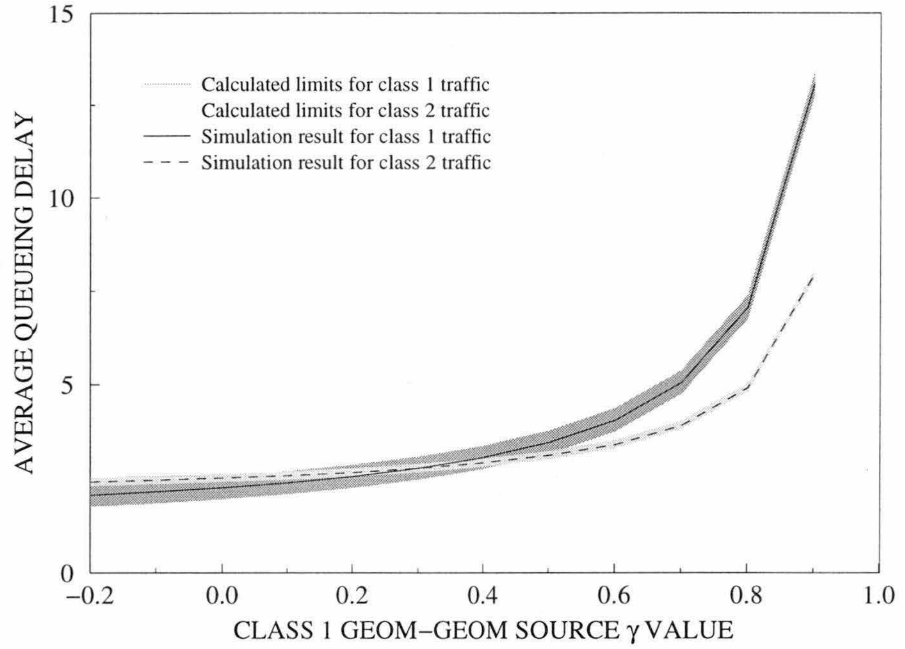

Figure B.3: *Average queueing delay (theoretical limits and simulation results) for two classes of geom-geom binary sources. The first class has one source with* $\lambda_1 = 0.2$ *and* γ_1 *given by the independent variable in the figure. The second class has three identical sources, with each source contributing* 0.2 *arrivals per time slot, with autocorrelation parameter* 0.3*. The overall queue utilisation is* 0.8*.*

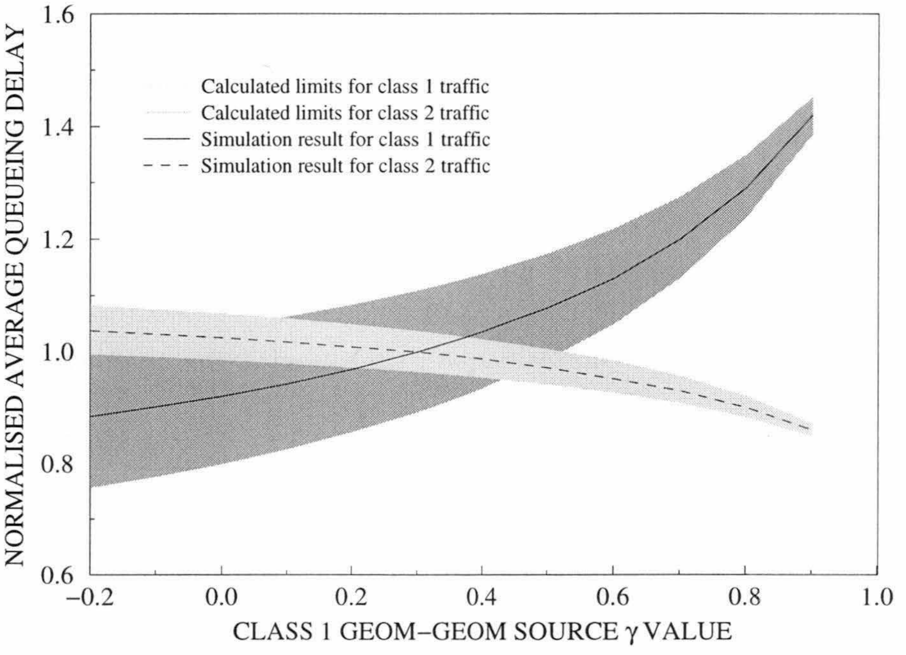

Figure B.4: *Queueing delay results of Figure B.3 normalised to the overall average queueing delay.*

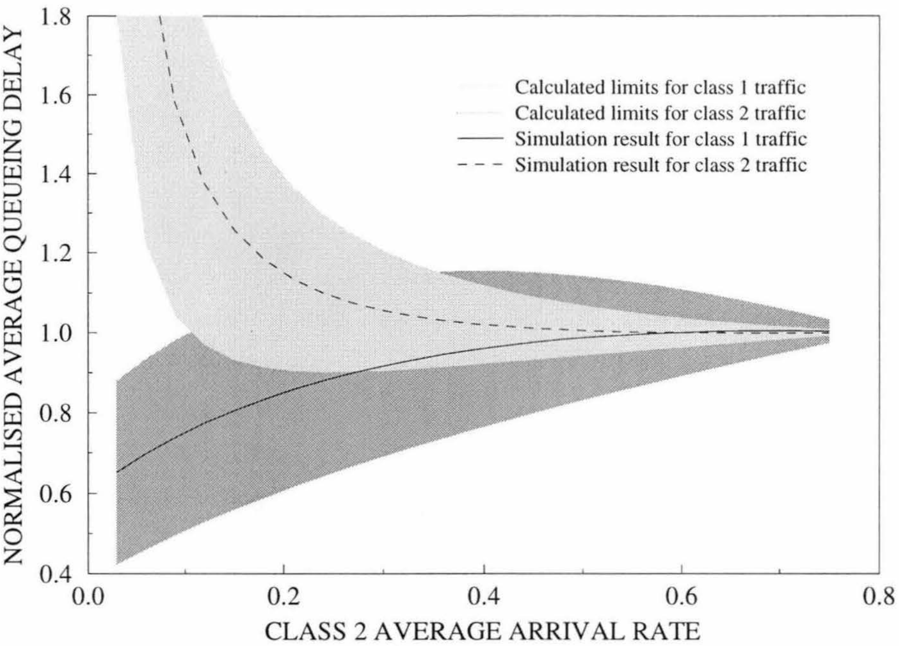

Figure B.5: *Normalised average queueing delay (theoretical limits and simulation results) for two classes of geom-geom binary sources. The first class has one source with* $\lambda_1 = 0.2$ *and* $\gamma_1 = 0.3$. *The second class has three identical sources, each having* $\gamma = 0.3$, *with a total average arrival rate given by the figure's independent variable. The overall queue utilisation varies from* 0.2 *to* 0.95.

The difference between the upper and lower limits for a particular class is actually dependent on the average arrival rates of the remaining classes. This is well illustrated by the above example, where the two traffic classes have considerably different arrival rates. In order to investigate what other effect the average arrival rate of a class has on its queueing delay performance, the same arrangement of sources and traffic classes was considered, but this time all the sources had identical autocorrelation parameters of $\gamma = 0.3$. The average arrival rate of the first class was kept constant at $\lambda_1 = 0.2$ and the average arrival rate of the second traffic class was varied from 0 to 0.75 (0 to 0.25 per source). Only the normalised average queueing delay results are shown here, in Figure B.5.

The graphed results show that the class having sources with the larger arrival rates receives the best queueing delay performance. That is, when the average arrival rate from traffic class 2 is below 0.6 (or below 0.2 per source) then traffic class 1 has a better average queuing delay. After this point, the class 2 traffic receives the better queueing performance, although the normalised delays of the two traffic classes seem

[1] This method was used to successfully confirm the theoretical limit results in all of the examples discussed in this appendix.

to be converging to unity as the utilisation of the queue approaches 100%.

B.3.3 Mixing a Periodic Source with Geometric-Geometric Sources

Here we consider the shared infinite buffer with arrivals from just two classes. The first class is a single periodic source that generates a single arrival every R service periods, while the second class consists of a number S_2 of identical geom-geom binary sources. We will consider the problem presented previously, but with the geom-geom source of the first traffic class replaced by a periodic source with $R = 5$, so that the average arrival rate from this source is still 0.2. The remaining three identical geom-geom sources make up the second arrival class, and have an average arrival rate each of 0.2 (giving $\lambda_2 = 0.6$) and an autocorrelation parameter γ_2 which is varied from -0.2 to 0.9 in steps of 0.1.

As before, Figures B.6 and B.7 show the calculated limits for each of the two classes along with the actual average queueing delays obtained from simulation of the queueing problem. Simulation results again are accurate to ±0.5% with 99% confidence.

The periodic source receives better queueing delays than the geom-geom sources for all of the γ_2 values investigated. Values of γ_2 below -0.25 cannot be used for geom-geom sources with arrival rates of only 0.2 without causing negative probability entries in their transition arrays. Thus it would appear that in this case, the periodic source is equivalent to a much more negatively autocorrelated source than the simpler two-state binary sources can achieve.

B.4 Discussion of the Example Results

Probably the first thing to note about the graphed results is that the simulation results all appear to fall exactly in the middle of the two limits. In fact, a comparison of the average of the theoretically calculated queueing delay limits with the simulation results shows that the difference between the two is less than the simulation accuracy in every case.

This is perhaps not too surprising considering how the limits were obtained, and considering that normally arrivals from all classes will be placed in a random order at the end of the queue buffer. At least this is the assumption used in the simulations, although it may not always be exactly true in practice. In such cases however, the theory can be used to estimate the actual average queueing delay for various traffic classes in a shared buffer, entirely by numeric solution methods.

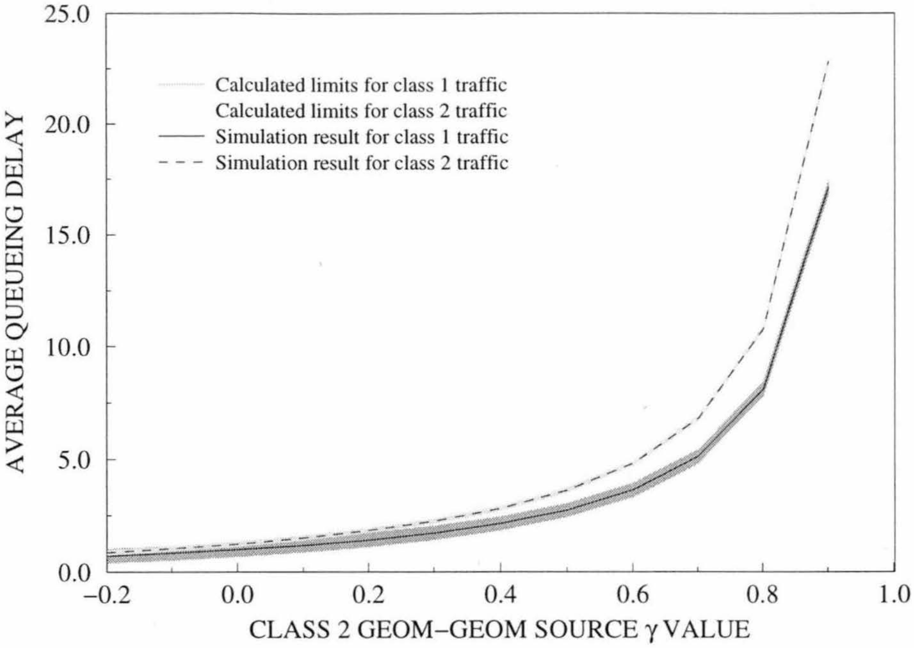

Figure B.6: *Average queueing delay (theoretical limits and simulation results) for two classes of traffic. The first is a periodic source with a fixed period of* 5 *service times. The second class has three identical geom-geom binary sources with each source having an average arrival rate of* 0.2 *with the autocorrelation parameters by the independent variable in the figure. The overall queue utilisation is* 0.8.

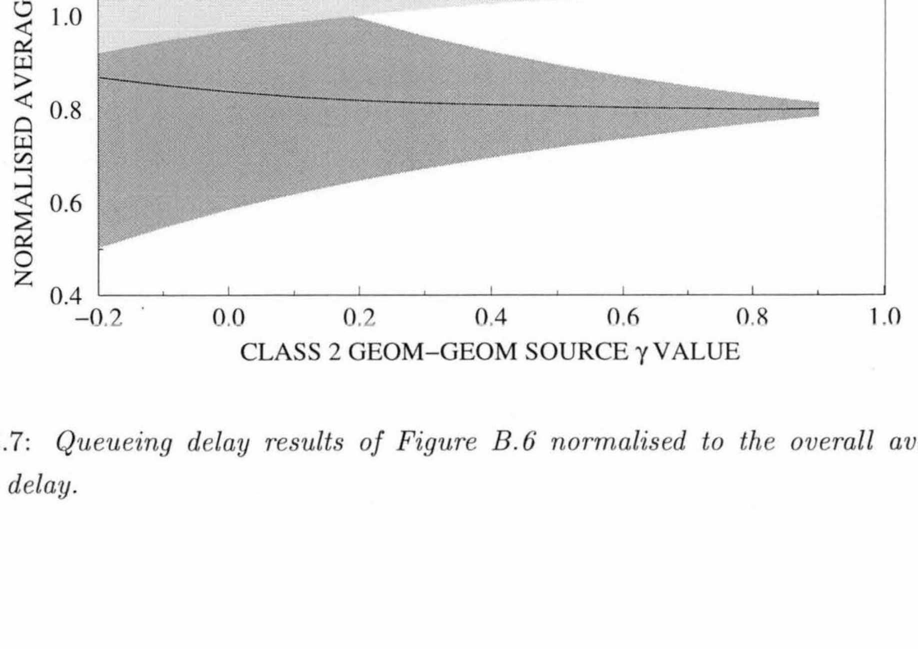

Figure B.7: *Queueing delay results of Figure B.6 normalised to the overall average queueing delay.*

Another result that is not quite so obvious as the midpoint one, but is perhaps more important, is that there appears to be overall limiting values to the average queueing delay that any traffic class can achieve in a shared buffer environment. Referring to Figures B.2, B.4, and B.7, it is apparent that as the γ term of the figure's independent variable tends towards one, the traffic class receiving the better queueing performance, has a normalised average queueing delay that approaches λ, or an average queueing delay that approaches L_q. That is, it appears from the results that the best average queueing delay that any class can have is equal to the average queue population. We propose this as a conjecture, and shall tender a rough proof for the case where the class in question is a periodic or cyclic process.

Conjecture B.1 *In a discrete-time deterministic service queueing environment, where an infinite buffer queue is serving multiple independent classes of traffic without preferences and without interruption, the smallest possible average queueing delay that any class can receive is equal to the average queue population. That is*

$$D_{q,c} \geq L_q$$

for any class c.

Proof. *No general proof for this result has yet been obtained. Consider however when the class in question is a cyclic process that generates b consecutive single arrivals within a period of R slot times. In the steady state, the average queue population after the cyclic process has generated an arrival will increase by* $\lambda - \lambda_c$ *over the value it had in the previous time slot, where* λ *is the overall average arrival rate to the queue, and* λ_c *is the average arrival rate of the cyclic process (given by b/R). During the* $R - b$ *time slots where the cyclic process generates no arrivals, the average queue population will decay to some value* x_0 *that will be the average queue population seen at the beginning of the next cycle. This change in the average queue population over one cycle time is illustrated in Figure B.8 for an example using* $b = 13$, $R = 32$, *and* $\lambda = 0.9$.

The value of L_q *is obtained by taking the mean over the entire cycle, of the average queue population after each service instant. There is no simple way to describe the decay part of the curve in Figure B.8 however, although an upper limit on* L_q *can be obtained by approximating this part of the curve by a straight line, giving*

$$L_q \leq x_0 + \frac{b\,(\lambda - \lambda_1)}{2}$$

The minimum average queueing delay seen by the arrivals from the cyclic traffic class is simply given by the mean of the average queue populations in the time slots preceding the cyclic arrivals. That is,

$$\min\left[D_{q,c}\right] = x_0 + \frac{(b-1)\,(\lambda - \lambda_c)}{2}$$

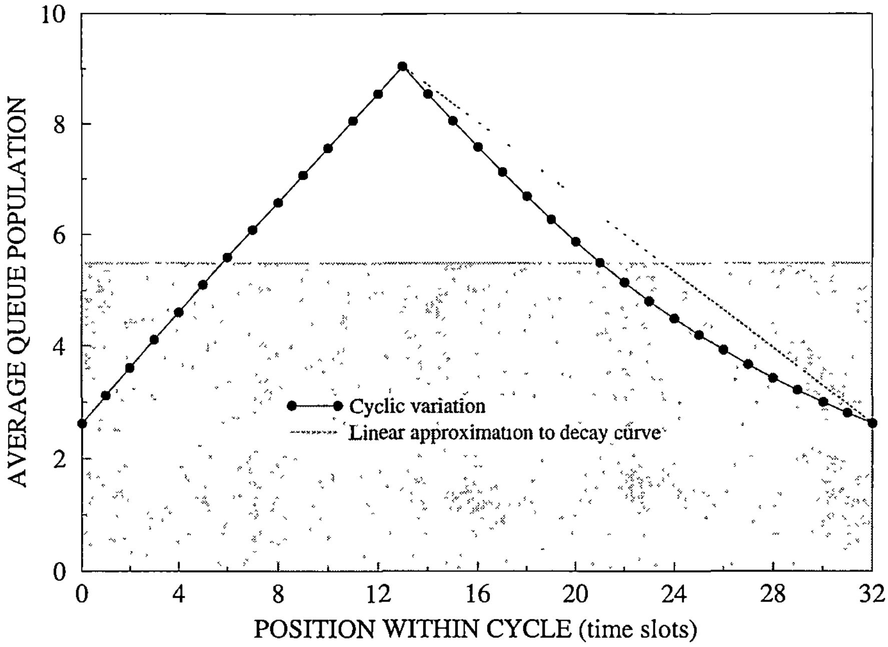

Figure B.8: *Variation in the average queue population as seen from the point of view of the cyclic traffic class c. The shaded area indicates the average population over the entire cycle.*

and since $D_{q,c} = \min[D_{q,c}] + \frac{\lambda - \lambda_c}{2}$ *(the average of the minimum and maximum* $D_{q,c}$ *limits) we obtain*

$$D_{q,c} = x_0 + \frac{b(\lambda - \lambda_1)}{2} \geq L_q$$

which is the required result. ■

Corollary B.2 *The largest possible average queueing delay that any traffic class can receive is* $(1 - \lambda + \lambda_c)/\lambda_c$ *times the average queue population. That is*

$$D_{q,c} \leq L_q + \frac{1-\lambda}{\lambda_c} L_q$$

for any class c. Thus conjecture B.1 might be rewritten as

$$L_q \leq D_{q,c} \leq L_q + \frac{1-\lambda}{\lambda_c} L_q$$

for any class c.

Proof. *The proof follows directly from conjecture B.1, and is deduced by assuming that the other traffic classes all receive the minimum average queueing delay. Then from the conservation law*

$$\lambda_c D_{q,c} = \lambda D_q - \sum_{i \neq c} \lambda_i D_{q,i}$$

and the result follows. ■

B.5 A Brief Look at the Limits of the Queueing Delay Variance

All of the above discussion has focused on defining the limits of the average queueing delay. Since knowledge of the variance is important, the issue of placing limits on the queueing delay variance also needs to be addressed. However, since a discussion of the shared buffer variance is not required in the rest of this thesis, we will restrict this discussion to a brief mention of the results for the marginal arrivals case. These limits are constructed in a manner similar to that used in section B.1 above, and are obtained as

$$\min\left[\mathrm{Var}\left[D_{q,c}\right]\right] = \mathrm{Var}\left[L_q\right] + \frac{M_{3,c}}{3\lambda_c} - \frac{\left(\sigma_c^2 + \lambda_c^2\right)^2}{4\lambda_c^2} - \frac{1}{12} \tag{B.24}$$

and

$$\max\left[\mathrm{Var}\left[D_{q,c}\right]\right] = \min\left[\mathrm{Var}\left[D_{q,c}\right]\right] + \sigma^2 - \sigma_c^2 \tag{B.25}$$

where σ^2 and σ_c^2 are the variance in the number of arrivals per time slot of the overall arrival process, and of traffic class c respectively, and $M_{3,c}$ is the third moment of the number of arrivals per time slot from traffic class c. For binary sources, these two equations become

$$\min\left[\mathrm{Var}\left[D_{q,c}\right]\right] = \mathrm{Var}\left[L_q\right] \tag{B.26}$$

and

$$\max\left[\mathrm{Var}\left[D_{q,c}\right]\right] = \mathrm{Var}\left[L_q\right] + \sigma^2 - \lambda_c\left(1 - \lambda_c\right) \tag{B.27}$$

The autocorrelated arrivals case can be constructed in a manner similar to that used in the average queueing delay discussion above.

We note here that, unlike the average queueing delays, the actual queueing delay variance is not the average of the minimum and maximum variance limits.

Appendix C

Eigensystem Analysis of Arrival Processes

This appendix presents both a general eigensystem analysis, and the Perron–Frobenius eigensystem analysis, for the phase-geometric and the cyclic arrival processes, for a single source. The results are easily extendable to multiple sources.

Additionally, results are also presented explicitly for the case where the phase-geometric arrival process can be represented by a geometric-geometric process.

The analysis is based on a probability generating function approach, where z describes the parameter of the generating function. Each source is assumed to be modelled by a Markov process, with irreducible stochastic transition matrix $\mathbf{A}$ and probability generating matrix $\mathbf{P}(z)$. The process will have some number of eigenvalues $\omega_n(z)$ equal to the number of Markov states required to describe the process. Corresponding to the nth eigenvalue are the left (row) and right (column) eigenvectors, $\mathbf{h}_n(z)$ and $\mathbf{g}_n(z)$ respectively, given by the relations

$$\mathbf{h}_n(z)\mathbf{A}\mathbf{P}(z) = \omega_n(z)\mathbf{h}_n(z) \tag{C.1}$$

and

$$\mathbf{A}\mathbf{P}(z)\mathbf{g}_n(z) = \omega_n(z)\mathbf{g}_n(z) \tag{C.2}$$

where, from the basic properties of eigenvectors, $\mathbf{h}_n(z)\mathbf{g}_n(z) = 1$, and $\mathbf{h}_n(z)\mathbf{g}_m(z) = 0$ for $n \neq m$.

Although the eigenvalues can be arbitrarily assigned to the indices n, we require that the eigenvalue at $z = 1$ that takes the value of 1 will be described by index 0 — that is, we adopt the convention that $\omega_0(1) = 1$. (It is a property of irreducible stochastic matrices that they have exactly one eigenvalue equal to 1 and all other eigenvalues with magnitudes less than or equal to one.)

The Perron–Frobenius eigenvalue and eigenvectors for the source are a special case of the general eigensystem above. The particular $\omega_n(z)$ corresponding to the Perron–Frobenius eigenvalue is denoted by $\delta(z)$, with left and right eigenvectors $\mathbf{u}(z)$ and $\mathbf{v}(z)$ which satisfy

$$\mathbf{u}(z)\mathbf{A}\mathbf{P}(z) = \delta(z)\mathbf{u}(z) \tag{C.3}$$

and

$$\mathbf{A}\mathbf{P}(z)\mathbf{v}(z) = \delta(z)\mathbf{v}(z) \tag{C.4}$$

with the additional constraints that $\mathbf{u}(z)\mathbf{v}(z) = 1$ and $\mathbf{u}(z)\mathbf{e} = 1$, where $\mathbf{e}$ is the unit column vector. As will be discussed in section C.3, and shown for both arrival processes below, this second constraint requires that $\delta(1) = 1$, implying that $\delta(z) = \omega_0(z)$.

Note that some of the notation and part of the basic approach used in this appendix follows Neuts in [98] and [95], and Li in [79] and [83].

C.1 Phase-Geometric Random Processes

In this section we consider a discrete time random process with active periods described by a phase type distribution[1], and silent periods described by a geometric distribution. During the silent periods, the process generates no arrivals, while during the active periods, the process generates arrivals according to some marginal distribution $\{p_k\}$ where p_k denotes the probability that k arrivals are generated in the current time slot. The probability generating function for this distribution is denoted in the following by $p(z)$.

Define the irreducible stochastic state transition matrix $\mathbf{A}$ by

$$\mathbf{A} = \begin{bmatrix} c & c'\boldsymbol{\alpha} \\ \mathbf{T}^{\circ} & \mathbf{T} \end{bmatrix} \tag{C.5}$$

where $c' = 1 - c$, and where the row vector $\boldsymbol{\alpha}$ and substochastic matrix $\mathbf{T}$ describe the irreducible form of a phase type distribution, with column vector $\mathbf{T}^{\circ}$ given by $(\mathbf{I} - \mathbf{T})\,\mathbf{e}$, where $\mathbf{e}$ is a column vector with all components equal to 1. Denote the invariant probability vector of $\mathbf{A}$ by $\boldsymbol{\mu}$, so that $\boldsymbol{\mu}\mathbf{A} = \boldsymbol{\mu}$.

The probability density of the active periods of the phase type distribution is given by

$$a_n = \boldsymbol{\alpha}\mathbf{T}^{n-1}\mathbf{T}^{\circ} \tag{C.6}$$

[1] For an elementary discussion of phase type distributions, refer to Chapter 2 of [95].

where a_n describes the probability that the source is active for exactly n periods, having a probability generating function

$$a(z) = z\boldsymbol{\alpha}\left(\mathbf{I} - z\mathbf{T}\right)^{-1}\mathbf{T}^{\circ} \tag{C.7}$$

with

$$\begin{aligned} a'(z) &= \boldsymbol{\alpha}\left(\mathbf{I} - z\mathbf{T}\right)^{-2}\mathbf{T}^{\circ} && \text{(C.8)} \\ a''(z) &= 2\boldsymbol{\alpha}\left(\mathbf{I} - z\mathbf{T}\right)^{-3}\mathbf{T}\mathbf{T}^{\circ} && \text{(C.9)} \\ a'''(z) &= 6\boldsymbol{\alpha}\left(\mathbf{I} - z\mathbf{T}\right)^{-4}\mathbf{T}^{2}\mathbf{T}^{\circ} && \text{(C.10)} \end{aligned}$$

From these derivatives, the first, second, and third moments of the active period are denoted by

$$\begin{aligned} \eta_1 &= \boldsymbol{\alpha}\left(\mathbf{I} - \mathbf{T}\right)^{-1}\mathbf{e} && \text{(C.11)} \\ \eta_2 &= \boldsymbol{\alpha}\left(\mathbf{I} - \mathbf{T}\right)^{-2}\left(\mathbf{I} + \mathbf{T}\right)\mathbf{e} && \text{(C.12)} \\ \eta_3 &= \boldsymbol{\alpha}\left(\mathbf{I} - \mathbf{T}\right)^{-3}\left(\mathbf{I} + 4\mathbf{T} + \mathbf{T}^{2}\right)\mathbf{e} && \text{(C.13)} \end{aligned}$$

respectively. We can also show that

$$\begin{aligned} \boldsymbol{\alpha}\left(\mathbf{I} - \mathbf{T}\right)^{-2}\mathbf{e} &= \frac{\eta_2 + \eta_1}{2} && \text{(C.14)} \\ \boldsymbol{\alpha}\left(\mathbf{I} - \mathbf{T}\right)^{-3}\mathbf{e} &= \frac{\eta_3 + 3\eta_2 + 2\eta_1}{6} && \text{(C.15)} \end{aligned}$$

Define the probability generating matrix $\mathbf{P}(z)$ by

$$\mathbf{P}(z) = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & p(z)\mathbf{I} \end{bmatrix} \tag{C.16}$$

where $p(z)$ is the probability generating function for the arrival process during the active periods, with $p(1) = 1$, and where the boldface $\mathbf{0}$ indicate row or column vectors as appropriate, and $\mathbf{I}$ indicates the identity matrix. The overall dimension of this probability generating matrix is the same as that of the matrix $\mathbf{A}$ so that the matrix $\mathbf{AP}(z)$ represents the transition probability generating matrix, with $\mathbf{AP}(1) = \mathbf{A}$.

Denote the nth eigenvalue of $\mathbf{AP}(z)$ by $\omega_n(z)$, with corresponding left and right eigenvectors $\mathbf{h}_n(z)$ and $\mathbf{g}_n(z)$, obeying the relations (C.1) and (C.2) respectively. The Perron–Frobenius eigenvalue $\delta(z)$ is given by $\omega_0(z)$, and has left and right eigenvectors $\mathbf{u}(z)$ and $\mathbf{v}(z)$, obeying the relations (C.3) and (C.4) respectively. For convenience of notation, let

$$\mathbf{h}_n(z) = \begin{bmatrix} h_{n,0}(z) & \mathbf{h}_{n,1}(z) \end{bmatrix} \quad \text{and} \quad \mathbf{g}_n(z) = \begin{bmatrix} g_{n,0}(z) \\ \mathbf{g}_{n,1}(z) \end{bmatrix} \tag{C.17}$$

as well as

$$\mathbf{u}(z) = \begin{bmatrix} u_0(z) & \mathbf{u}_1(z) \end{bmatrix} \quad \text{and} \quad \mathbf{v}(z) = \begin{bmatrix} v_0(z) \\ \mathbf{v}_1(z) \end{bmatrix} \tag{C.18}$$

where the boldface symbols indicate vector quantities.

C.1.1 Obtaining $\mathbf{h}_n(z)$ and $\omega_n(z)$

From (C.1) we have

$$ch_{n,0}(z) + \mathbf{h}_{n,1}(z)\mathbf{T}^{\circ} = \omega_n(z)h_{n,0}(z) \tag{C.19}$$

and

$$c'p(z)h_{n,0}(z)\boldsymbol{\alpha}+p(z)\mathbf{h}_{n,1}(z)\mathbf{T} = \omega_n(z)\mathbf{h}_{n,1}(z) \tag{C.20}$$

Post-multiplying equation (C.20) by $\mathbf{e}$, and making use of $\mathbf{T}^{\circ} = (\mathbf{I} - \mathbf{T})\,\mathbf{e}$ gives

$$c'p(z)h_{n,0}(z) - p(z)\mathbf{h}_{n,1}(z)\mathbf{T}^{\circ} = (\omega_n(z){-}p(z))\,\mathbf{h}_{n,1}(z)\mathbf{e} \tag{C.21}$$

where from (C.19)

$$\mathbf{h}_{n,1}(z)\mathbf{T}^{\circ} = (\omega_n(z) - c)\,h_{n,0}(z) \tag{C.22}$$

and hence

$$h_{n,0}(z) = \frac{\omega_n(z) - p(z)}{p(z)\,(1 - \omega_n(z))}\mathbf{h}_{n,1}(z)\mathbf{e} \tag{C.23}$$

From equation (C.20) we also get

$$\mathbf{h}_{n,1}(z) = c'p(z)h_{n,0}(z)\boldsymbol{\alpha}\,(\omega_n(z)\mathbf{I} - p(z)\mathbf{T})^{-1} \tag{C.24}$$

so that, post-multiplying by $\mathbf{T}^{\circ}$ gives, with some manipulation

$$\omega_n(z) = c + c'p(z)\boldsymbol{\alpha}\,(\omega_n(z)\mathbf{I} - p(z)\mathbf{T})^{-1}\,\mathbf{T}^{\circ} \tag{C.25}$$

which can be rewritten as

$$\omega_n(z) = c + c'a\left(\frac{p(z)}{\omega_n(z)}\right) \tag{C.26}$$

where $a\,(z)$ is the probability generating function for the durations of the active periods. It is easy to see that $\omega_0(1) = 1$ is one solution to equation (C.26).

C.1.2 Obtaining $\mathbf{g}_n(z)$

From (C.2) we have

$$cg_{n,0}(z) + c'p(z)\boldsymbol{\alpha}\mathbf{g}_{n,1}(z) = \omega_n(z)g_{n,0}(z) \tag{C.27}$$

and

$$g_{n,0}(z)\mathbf{T}^{\circ}{+}p(z)\mathbf{T}\mathbf{g}_{n,1}(z) = \omega_n(z)\mathbf{g}_{n,1}(z) \tag{C.28}$$

Re-arranging equation (C.28) gives

$$\mathbf{g}_{n,1}(z) = g_{n,0}(z)\,(\omega_n(z)\mathbf{I} - p(z)\mathbf{T})^{-1}\,\mathbf{T}^{\circ} \tag{C.29}$$

while the property $\mathbf{h}_n(z)\mathbf{g}_n(z) = 1$ yields

$$h_{n,0}(z)g_{n,0}(z) + \mathbf{h}_{n,1}(z)\mathbf{g}_{n,1}(z) = 1 \tag{C.30}$$

so that equations (C.24) and (C.29) give

$$h_{n,0}(z)g_{n,0}(z)\left(1 + c'p(z)\boldsymbol{\alpha}\left(\omega_n(z)\mathbf{I} - p(z)\mathbf{T}\right)^{-2}\mathbf{T}^{\circ}\right) = 1 \qquad \text{(C.31)}$$

or

$$h_{n,0}(z)g_{n,0}(z) = \frac{\omega_n(z)^2}{\omega_n(z)^2 + c'p(z)a'\left(\frac{p(z)}{\omega_n(z)}\right)} \qquad \text{(C.32)}$$

where $a'(z)$ is the first derivative of the probability generating function $a(z)$. Consideration of the first derivative of equation (C.26) allows this to be further simplified, to give

$$h_{n,0}(z)g_{n,0}(z) = 1 - \frac{p(z)\omega_n'(z)}{p'(z)\omega_n(z)} \qquad \text{(C.33)}$$

For convenience, we will choose $g_{n,0}(z) = 1$ for all n, giving

$$h_{n,0}(z) = 1 - \frac{p(z)\omega_n'(z)}{p'(z)\omega_n(z)} \qquad \text{(C.34)}$$

and

$$\mathbf{g}_{n,1}(z) = \left(\omega_n(z)\mathbf{I} - p(z)\mathbf{T}\right)^{-1}\mathbf{T}^{\circ} \qquad \text{(C.35)}$$

C.1.3 Simplifying $\mathbf{h}_n(1)\mathbf{g}_0'(1)$

In the analysis of the variance of the queue population of a G/D/1 queue fed by phase-geom sources, terms in $\mathbf{h}_n(1)\mathbf{g}_0'(1)$ for each source are encountered, where $n \neq 0$. Since $n \neq 0$ we have $\mathbf{h}_n(z)\mathbf{g}_0(z) = 0$. Taking the first derivative of this relation gives

$$\mathbf{h}_n'(z)\mathbf{g}_0(z) + \mathbf{h}_n(z)\mathbf{g}_0'(z) = 0 \qquad \text{(C.36)}$$

or at $z = 1$

$$\mathbf{h}_n(1)\mathbf{g}_0'(1) = -\mathbf{h}_n'(1)\mathbf{e} \qquad \text{(C.37)}$$

using $\mathbf{g}_0(1) = \mathbf{e}$, which is obtained from equation (C.35) using the fact that $\omega_0(1) = 1$.

Taking the first derivative of $h_{n,0}(z)$ as given by equation (C.23) yields

$$\begin{aligned} h_{n,0}'(z) &= \left(\frac{p(z)\omega_n'(z)\left(1 - p(z)\right) - p'(z)\omega_n(z)\left(1 - \omega_n(z)\right)}{p(z)\left(1 - \omega_n(z)\right)\left(\omega_n(z) - p(z)\right)}\right) h_{n,0}(z) \\ &\quad + \frac{\omega_n(z) - p(z)}{p(z)\left(1 - \omega_n(z)\right)}\mathbf{h}_{n,1}'(z)\mathbf{e} \end{aligned} \qquad \text{(C.38)}$$

so that at $z = 1$, we obtain with some manipulation

$$\mathbf{h}_n'(1)\mathbf{e} = \frac{p'(1)\omega_n(1)}{1 - \omega_n(1)} h_{n,0}(1) \qquad \text{(C.39)}$$

hence

$$\mathbf{h}_n(1)\mathbf{g}_0'(1) = -\frac{p'(1)\omega_n(1)}{1 - \omega_n(1)} h_{n,0}(1) \qquad \text{(C.40)}$$

or, using equation (C.34)

$$\mathbf{h}_n(1)\mathbf{g}_0'(1) = \frac{\omega_n'(1) - p'(1)\omega_n(1)}{1 - \omega_n(1)} \tag{C.41}$$

which is the desired result.

C.1.4 Obtaining $\mathbf{u}(z)$ and $\delta(z)$

Expressions for $\mathbf{u}(z)$ and $\delta(z)$ can be derived in a similar manner as for $\mathbf{h}_n(z)$ and $\omega_n(z)$, but with the additional constraint that $\mathbf{u}(z)\mathbf{e} = 1$. Writing equation (C.3) in expanded form gives

$$cu_0(z) + \mathbf{u}_1(z)\mathbf{T}^\circ = \delta(z)u_0(z) \tag{C.42}$$

and

$$c'p(z)u_0(z)\boldsymbol{\alpha}+p(z)\mathbf{u}_1(z)\mathbf{T} = \delta(z)\mathbf{u}_1(z) \tag{C.43}$$

from which we obtain

$$u_0(z) = \frac{\delta(z) - p(z)}{\delta(z)\,(1 - p(z))} \tag{C.44}$$

$$\mathbf{u}_1(z) = c'p(z)u_0(z)\boldsymbol{\alpha}\,(\delta(z)\mathbf{I} - p(z)\mathbf{T})^{-1} \tag{C.45}$$

and

$$\delta(z) = c + c'a\left(\frac{p(z)}{\delta(z)}\right) \tag{C.46}$$

Note that if $\delta(1) \neq 1$ in equations (C.44) and (C.45), then the expression for $\mathbf{u}(1)$ becomes infinite which violates the required condition that $\mathbf{u}(1)\mathbf{e} = 1$. Hence, by contradiction, we must have $\delta(1) = 1$, which means that $\delta(z)$ is the same eigenvector as $\omega_0(z)$ according to the convention we have adopted.

C.1.5 Obtaining $\mathbf{v}(z)$

Writing equation (C.4) in expanded form gives

$$cv_0(z) + c'p(z)\boldsymbol{\alpha}\mathbf{v}_1(z) = \delta(z)v_0(z) \tag{C.47}$$

and

$$v_0(z)\mathbf{T}^\circ+p(z)\mathbf{T}\mathbf{v}_1(z) = \delta(z)\mathbf{v}_1(z) \tag{C.48}$$

from which, in a like manner to the derivation of $\mathbf{g}_n(z)$ we obtain

$$\mathbf{v}_1(z) = v_0(z)\,(\delta(z)\mathbf{I} - p(z)\mathbf{T})^{-1}\,\mathbf{T}^\circ \tag{C.49}$$

while the property $\mathbf{u}(z)\mathbf{v}(z) = 1$ yields, with some manipulation

$$u_0(z)v_0(z) = 1 - \frac{p(z)\delta'(z)}{p'(z)\delta(z)} \tag{C.50}$$

or

$$v_0(z) = \frac{(1 - p(z))\,(p'(z)\delta(z) - p(z)\delta'(z))}{p'(z)\,(\delta(z) - p(z))} \tag{C.51}$$

C.1.6 Derivatives of $\delta(z)$ and $v(z)$

Calculating the average and variance of the queue population for a G/D/1 queue fed by phase-geom sources requires the first three derivatives (with respect to z) of the Perron–Frobenius eigenvalue, and the first two derivatives of the corresponding right-hand eigenvector, evaluated at $z = 1$. These derivatives are easily performed, particularly with the use of a symbolic mathematics program such as *Mathematica* [139]. In the equations below, the terms $\varepsilon_r = c'\eta_r$ where η_r is the rth moment of the period of the active periods, is used to simplify the expressions.

For the eigenvalue $\delta(z)$ we have

$$\begin{aligned}
\delta(1) &= 1 && \text{(C.52)}\\
\delta'(1) &= \frac{\varepsilon_1}{1+\varepsilon_1}p'(1) && \text{(C.53)}\\
\delta''(1) &= \frac{\varepsilon_2-\varepsilon_1-2\varepsilon_1^2}{(1+\varepsilon_1)^3}p'(1)^2+\frac{\varepsilon_1}{1+\varepsilon_1}p''(1) && \text{(C.54)}\\
\delta'''(1) &= \frac{2\varepsilon_1+8\varepsilon_1^2+9\varepsilon_1^3-3\varepsilon_2-6\varepsilon_1\varepsilon_2+3\varepsilon_1^2\varepsilon_2-3\varepsilon_2^2+\varepsilon_3+\varepsilon_1\varepsilon_3}{(1+\varepsilon_1)^5}p'(1)^3\\
&\quad+\frac{3\left(\varepsilon_2-2\varepsilon_1^2-\varepsilon_1\right)}{(1+\varepsilon_1)^3}p'(1)p''(1)+\frac{\varepsilon_1}{1+\varepsilon_1}p'''(1) && \text{(C.55)}
\end{aligned}$$

and for the eigenvector $\mathbf{v}(z)$

$$\begin{aligned}
v_0(1) &= 1 && \text{(C.56)}\\
v_0'(1) &= \frac{-\delta''(1)}{2\left(p'(1)-\delta'(1)\right)}+\frac{\delta'(1)}{2p'(1)\left(p'(1)-\delta'(1)\right)}p''(1) && \text{(C.57)}\\
v_0''(1) &= \frac{-3p'(1)\delta''(1)-2\delta'''(1)}{3\left(p'(1)-\delta'(1)\right)}-\frac{\delta''(1)^2}{2\left(p'(1)-\delta'(1)\right)^2}\\
&\quad+\frac{6\delta''(1)p'(1)+3\left(2\delta'(1)p'(1)+\delta''(1)\right)\left(p'(1)-\delta'(1)\right)}{6p'(1)\left(p'(1)-\delta'(1)\right)^2}p''(1)\\
&\quad-\frac{p'(1)+2\left(p'(1)-\delta'(1)\right)}{2p'(1)^2\left(p'(1)-\delta'(1)\right)^2}\delta'(1)p''(1)^2\\
&\quad+\frac{4\delta'(1)}{6p'(1)\left(p'(1)-\delta'(1)\right)}p'''(1) && \text{(C.58)}
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{v}_1(1^-) &= \mathbf{e} && \text{(C.59)}\\
\mathbf{v}_1'(1^-) &= \left(v_0'(1)-p'(1)\right)\mathbf{e}-\left(\delta'(1)-p'(1)\right)\left(\mathbf{I}-\mathbf{T}\right)^{-1}\mathbf{e} && \text{(C.60)}\\
\mathbf{v}_1''(1^-) &= \left(v_0''(1)+2p'(1)^2\right)\mathbf{e}\\
&\quad-\left(\delta''(1)\mathbf{I}+2v_0'(1)\delta'(1)\mathbf{I}-p''(1)\mathbf{T}-2v_0'(1)p'(1)\mathbf{T}\right)\left(\mathbf{I}-\mathbf{T}\right)^{-1}\mathbf{e}\\
&\quad-4p'(1)\left(p'(1)-\delta'(1)\right)\left(\mathbf{I}-\mathbf{T}\right)^{-1}\mathbf{e}\\
&\quad+2\left(\delta'(1)-p'(1)\right)^2\left(\mathbf{I}-\mathbf{T}\right)^{-2}\mathbf{e} && \text{(C.61)}
\end{aligned}$$

We can also write the derivatives of the zero element of the eigenvector in terms of ε_r only, giving

$$\begin{aligned}
v_0'(1) &= \frac{\varepsilon_1 - \varepsilon_2 + 2\varepsilon_1^2}{2\,(1+\varepsilon_1)^2}p'(1) && \text{(C.62)}\\
v_0''(1) &= \frac{-2\varepsilon_1 + 6\varepsilon_2 - 4\varepsilon_3 - 11\varepsilon_1^2 + 18\varepsilon_1\varepsilon_2 + 9\varepsilon_2^2 - 4\varepsilon_1\varepsilon_3 - 18\varepsilon_1^3 - 6\varepsilon_1^2\varepsilon_2}{6\,(1+\varepsilon_1)^4}p'(1)^2\\
&\quad + \frac{2\varepsilon_1^2 + \varepsilon_1 - \varepsilon_2}{6\,(1+\varepsilon_1)^2}p''(1) && \text{(C.63)}
\end{aligned}$$

by substituting from the eigenvalue derivatives.

C.2 Cyclic Processes

In this section we consider a discrete time cyclic process with a period of C time slots. Within these C slots, the process is assumed to be silent for the first $C - b$ time slots, and active for the remaining b slots of the cycle. As with the phase-geom process above, the cyclic process generates no arrivals during silent periods, while during the active periods, the process generates arrivals according to some marginal distribution $\{p_k\}$ where p_k denotes the probability that k arrivals are generated in the current time slot. The probability generating function for this distribution is denoted in the following discussion by $p(z)$.

This process can be represented by a Markov process having C states. Define an irreducible stochastic matrix $\mathbf{A}$ to describes the state to state transition behaviour of the cyclic source, having the form

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \qquad \text{(C.64)}$$

or in more compact form

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I}_{C-1} \\ 1 & \mathbf{0} \end{bmatrix} \qquad \text{(C.65)}$$

where the boldface $\mathbf{0}$'s are row or column vectors of $C - 1$ elements each, and $\mathbf{I}_{C-1}$ is an identity matrix of size $(C - 1) \times (C - 1)$.

Define the probability generating matrix $\mathbf{P}(z)$ by

$$\mathbf{P}(z) = \begin{bmatrix} \mathbf{I}_{C-b} & \mathbf{0}_{(C-b)\times b} \\ \mathbf{0}_{b\times(C-b)} & p(z)\mathbf{I}_b \end{bmatrix} \tag{C.66}$$

where the boldface $\mathbf{0}_{b\times(C-b)}$ and $\mathbf{0}_{(C-b)\times b}$ represent zero matrices of the indicated dimensions. Thus the combined transition probability generating matrix is given by

$$\mathbf{AP}(z) = \begin{bmatrix} 0 & \mathbf{I}_{C-b-1} & \mathbf{0}_{(C-b-1)\times b} \\ \mathbf{0} & \mathbf{0}_{b\times(C-b-1)} & p(z)\mathbf{I}_b \\ 1 & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{C.67}$$

where the boldface $\mathbf{0}$'s without subscripts are column (or row where appropriate) vectors of zeros.

The eigenvalues $\omega(z)$ of the matrix $\mathbf{AP}(z)$ are derived simply owing to the diagonalised nature of the matrix, and are given by the C solutions of

$$\omega_n(z)^C - p(z)^b = 0 \tag{C.68}$$

or

$$\omega_n(z) = e^{\frac{2n\pi}{C}\sqrt{-1}}p(z)^{\frac{b}{C}} \tag{C.69}$$

where $n = 0, 1, \ldots, C-1$. This expression for $\omega_n(z)$ represents an arbitrary allocation of the C eigenvalues to the indices n, but in particular it provides the result $\omega_0(1) = 1$.

C.2.1 Obtaining $\mathbf{h}_n(z)$ and $\mathbf{g}_n(z)$

The left eigenvector $\mathbf{h}_n(z)$ and right eigenvector $\mathbf{g}_n(z)$ of $\mathbf{AP}(z)$ corresponding to $\omega_n(z)$ are as described for the phase-geom process, in equations (C.1) and (C.2). However, in order to simplify the derivation of these vectors, we will rewrite $\mathbf{P}(z)$ as

$$\mathbf{P}(z) = \begin{bmatrix} p_0(z) & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & p_1(z) & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & p_2(z) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & p_3(z) & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & p_4(z) & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & p_{C-1}(z) \end{bmatrix} \tag{C.70}$$

where

$$p_i(z) = \begin{cases} 1 & \text{for } 0 \le i < C-b \\ p(z) & \text{for } C-b \le i < C \end{cases} \tag{C.71}$$

From equation (C.1) we obtain, for $i \ge 1$

$$h_{n,i}(z) = \frac{p_i(z)}{\omega_n(z)}h_{n,i-1}(z) \tag{C.72}$$

where $h_{n,i}(z)$ denotes the element of eigenvector $\mathbf{h}_n(z)$ corresponding to state i. Similarly for the right-hand eigenvector $\mathbf{g}_n(z)$ we obtain from equation (C.2) that

$$g_{n,i}(z) = \frac{\omega_n(z)}{p_i(z)} g_{n,i-1}(z) \tag{C.73}$$

for $i \geq 1$. We choose $g_{n,0}(z) = 1$ for convenience, and in addition, from the relation $\mathbf{h}_n(z)\mathbf{g}_n(z) = 1$ we obtain $h_{n,0}(z) = 1/C$, giving

$$h_{n,i}(z) = \frac{1}{C\omega_n(z)^i} \prod_{j=1}^{i} p_j(z) \tag{C.74}$$

and

$$g_{n,i}(z) = \omega_n(z)^i \prod_{j=1}^{i} \frac{1}{p_j(z)} \tag{C.75}$$

C.2.2 Derivation of $\mathbf{h}_n(1)\mathbf{g}_0'(1)$

Again, we wish to obtain an expression for $\mathbf{h}_n(1)\mathbf{g}_0'(1)$ for $n \neq 0$ to be used in the analysis of the queue population variance of a G/D/1 queue fed by cyclic sources, where terms in $\mathbf{h}_n(1)\mathbf{g}_0'(1)$ for each source are encountered. Since $n \neq 0$ we have $\mathbf{h}_n(z)\mathbf{g}_0(z) = 0$. Taking the first derivative of this relation gives

$$\mathbf{h}_n'(z)\mathbf{g}_0(z) + \mathbf{h}_n(z)\mathbf{g}_0'(z) = 0 \tag{C.76}$$

or at $z = 1$

$$\mathbf{h}_n(1)\mathbf{g}_0'(1) = -\mathbf{h}_n'(1)\mathbf{e} \tag{C.77}$$

using $\mathbf{g}_0(1) = \mathbf{e}$, which is obtained from equation (C.75) using the fact that $\omega_0(1) = 1$.

Taking the first derivative of $h_{n,0}(z)$ as given by equation (C.74) yields

$$h_{n,i}'(z) = \frac{1}{C\omega_n(z)^i} \left(\sum_{k=1}^{i} \frac{p_k'(z)}{p_k(z)} - \frac{i\omega_n'(z)}{\omega_n(z)} \right) \prod_{j=1}^{i} p_j(z) \tag{C.78}$$

or at $z = 1$

$$h_{n,i}'(1) = \frac{1}{C\omega_n(1)^i} \left(\sum_{k=1}^{i} p_k'(1) - \frac{i\omega_n'(1)}{\omega_n(1)^i} \right) \tag{C.79}$$

Thus

$$\begin{aligned} \mathbf{h}_n(1)\mathbf{g}_0'(1) \quad = \quad & \frac{\omega_n'(1)}{\omega_n(1)^C \left(1 - \omega_n(1)\right)} - \frac{\omega_n'(1) \left(1 - \omega_n(1)^C\right)}{C\omega_n(1)^C \left(1 - \omega_n(1)\right)^2} \\ & - \frac{1}{C} \sum_{i=0}^{C-1} \left(\frac{1}{\omega_n(1)^i} \sum_{k=1}^{i} p_k'(1) \right) \end{aligned} \tag{C.80}$$

where using relation (C.71) and expanding in the summation gives

$$\begin{aligned}\mathbf{h}_n(1)\mathbf{g}_0'(1) &= \frac{\omega_n'(1) - \frac{b}{C}\omega_n(1)p'(1)}{\omega_n(1)^C\left(1-\omega_n(1)\right)} \\ &\quad + \frac{p'(1)\omega_n(1)^2\left(1-\omega_n(1)^b\right) - \omega_n'(1)\left(1-\omega_n(1)^C\right)}{C\omega_n(1)^C\left(1-\omega_n(1)\right)^2}\end{aligned} \tag{C.81}$$

From the first derivative of equation (C.69) we can show that

$$\omega_n'(1) = \frac{b}{C}\omega_n(1)p'(1) \tag{C.82}$$

hence

$$\mathbf{h}_n(1)\mathbf{g}_0'(1) = \frac{p'(1)\omega_n(1)^2\left(1-\omega_n(1)^b\right) - \omega_n'(1)\left(1-\omega_n(1)^C\right)}{C\omega_n(1)^C\left(1-\omega_n(1)\right)^2} \tag{C.83}$$

Finally, from equation (C.68) we obtain $\omega_n(1)^C = 1$, to give

$$\mathbf{h}_n(1)\mathbf{g}_0'(1) = \frac{p'(1)\omega_n(1)^2\left(1-\omega_n(1)^b\right)}{C\left(1-\omega_n(1)\right)^2} \tag{C.84}$$

which is the simplest form to which $\mathbf{h}_n(1)\mathbf{g}_0'(1)$ can be reduced.

C.2.3 Obtaining $\mathbf{u}(z)$, $\mathbf{v}(z)$, and $\delta(z)$

The relations for $\mathbf{u}(z)$ and $\mathbf{v}(z)$ are derived in a similar manner to $\mathbf{h}_n(z)$ and $\mathbf{g}_n(z)$ above, giving

$$u_i(z) = \frac{u_0(z)}{\delta(z)^i}\prod_{j=1}^{i} p_j(z) \tag{C.85}$$

where $u_i(z)$ represents the element of the vector $\mathbf{u}(z)$ corresponding to state i of the cyclic process. Similarly

$$v_i(z) = v_0(z)\delta(z)^i\prod_{j=1}^{i}\frac{1}{p_j(z)} \tag{C.86}$$

In order to determine $u_0(z)$ and $v_0(z)$ we note that $\mathbf{u}(z)\mathbf{v}(z) = 1$ yields

$$u_0(z)v_0(z) = \frac{1}{C} \tag{C.87}$$

and from $\mathbf{u}(z)\mathbf{e} = 1$ we obtain

$$\sum_{i=0}^{C-1} u_i(z) = 1 \tag{C.88}$$

Let $S(z)$ denote the sum

$$S(z) = \sum_{i=0}^{C-1}\frac{1}{\delta(z)^i}\prod_{j=1}^{i} p_j(z) \tag{C.89}$$

so that

$$u_0(z) = \frac{1}{S(z)} \qquad \text{and} \qquad v_0(z) = \frac{S(z)}{C} \tag{C.90}$$

and hence

$$u_i(z) = \frac{1}{S(z)\delta(z)^i} \prod_{j=1}^{i} p_j(z) \tag{C.91}$$

where $u_i(z)$ represents the element of the vector $\mathbf{u}(z)$ corresponding to state i of the cyclic process. Similarly

$$v_i(z) = \frac{S(z)\delta(z)^i}{C} \prod_{j=1}^{i} \frac{1}{p_j(z)} \tag{C.92}$$

At $z = 1$ we obtain

$$S(1) = \sum_{i=0}^{C-1} \delta(1)^{-i} \tag{C.93}$$

so that, if $\delta(1) = 1$ we have $S(1) = C$, and if $\delta(1) \neq 1$ we get

$$S(1) = \frac{\delta(1) - \delta(1)^C \delta(1)}{\delta(1)^C \left(1 - \delta(1)\right)} \tag{C.94}$$

but since $\delta(1)^C = 1$ from equation (C.68), this gives $S(1) = 0$ which would lead to an infinite $\mathbf{u}(z)$ and consequently $\mathbf{u}(z)\mathbf{e} \neq 1$. Therefore, as for the phase-geom case, we require $\delta(1) = 1$, corresponding to $\delta(z) = \omega_0(z)$ by our choice of notation, or

$$\delta(z) = p(z)^{\frac{b}{C}} \tag{C.95}$$

C.2.4 Derivatives of $\delta(z)$ and $\mathbf{v}(z)$

Again we require the first three derivatives of the eigenvalue $\delta(z)$ and the first two derivatives of $\mathbf{v}(z)$, evaluated at $z = 1$, for use in the analysis of the queue population of a G/D/1 queue. The eigenvalue derivatives are given by

$$\delta(1) = 1 \tag{C.96}$$

$$\delta'(1) = \frac{b}{C} p'(1) \tag{C.97}$$

$$\delta''(1) = \frac{b}{C}\left(\frac{b}{C} - 1\right) p'(1)^2 + \frac{b}{C} p''(1) \tag{C.98}$$

$$\begin{aligned} \delta'''(1) &= \frac{b}{C}\left(\frac{b}{C} - 1\right)\left(\frac{b}{C} - 2\right) p'(1)^3 \\ &\quad + \frac{3b}{C}\left(\frac{b}{C} - 1\right) p'(1)p''(1) + \frac{b}{C} p'''(1) \end{aligned} \tag{C.99}$$

while the derivatives of $\mathbf{v}(z)$ are given by

$$\mathbf{v}(1) = \mathbf{e} \tag{C.100}$$

$$v_i'(1) = v_{i-1}'(1) + \delta'(1) - p_i'(1) \tag{C.101}$$

$$v_i''(1) = v_{i-1}''(1) + 2\left(p_i'(1) - \delta'(1)\right)\left(p_i'(1) - v_{i-1}'(1)\right) + \delta''(1) \tag{C.102}$$

where

$$v_0'(1) = \frac{1}{C} S'(1) \tag{C.103}$$

$$v_0''(1) = \frac{1}{C} S''(1) \tag{C.104}$$

with

$$S(1) = C \tag{C.105}$$

$$S'(1) = \sum_{i=0}^{C-1} \left(\sum_{k=1}^{i} p_k'(1) - i\delta'(1) \right) \tag{C.106}$$

$$\begin{aligned} S''(1) = & \sum_{i=0}^{C-1} \left(\sum_{k=1}^{i} \sum_{m=1}^{i} p_k'(1) p_m'(1) - \sum_{k=1}^{i} p_k'(1)^2 - 2i\delta'(1) \sum_{k=1}^{i} p_k'(1) \right) \\ & + \sum_{i=0}^{C-1} \left(i\,(i+1)\, \delta'(1)^2 - i\delta''(1) \right) \end{aligned} \tag{C.107}$$

or

$$S(1) = C \tag{C.108}$$

$$S'(1) = b - \frac{b\,(C-b)}{2} \tag{C.109}$$

$$S''(1) = \frac{b\,(C-b)\,(3C - 6b + 2bC - 5 - 2b^2)}{6C} \tag{C.110}$$

C.3 Perron–Frobenius and General Eigensystem results

As already observed for the two arrival processes discussed above, the Perron–Frobenius eigensystem analysis is closely related to the general eigensystem analysis through the eigenvalue that takes on the value of 1 at $z = 1$. This requirement that $\delta(1) = 1$ can be proved very simply by observing that from

$$\mathbf{u}(1)\mathbf{A} = \delta(1)\mathbf{u}(1) \tag{C.111}$$

we have, on postmultiplication by $\mathbf{e}$, that

$$\mathbf{u}(1)\mathbf{e} = \delta(1)\mathbf{u}(1)\mathbf{e} \tag{C.112}$$

which has only two solutions, $\delta(1) = 1$ or $\mathbf{u}(1)\mathbf{e} = 0$. Since the Perron–Frobenius eigensystem analysis explicitly requires $\mathbf{u}(z)\mathbf{e} = 1$, we have $\delta(1) = 1$. Since this eigenvalue corresponds to the general eigenvalue $\omega_0(1)$ there must exist a non-zero scalar function $s(z)$ such that

$$\mathbf{h}_0(z) = \frac{1}{s(z)} \mathbf{u}(z) \tag{C.113}$$

and

$$\mathbf{g}_0(z) = s(z)\mathbf{v}(z) \tag{C.114}$$

giving, with a little manipulation

$$\mathbf{h}_0'(z) = \frac{1}{s(z)}\mathbf{u}'(z) - \frac{s'(z)}{s(z)^2}\mathbf{u}(z) \tag{C.115}$$

$$\mathbf{h}_0''(z) = \frac{1}{s(z)}\mathbf{u}''(z) - 2\frac{s'(z)}{s(z)^2}\mathbf{u}'(z) + \left(2\frac{s'(z)^2}{s(z)^3} - \frac{s''(z)}{s(z)^2}\right)\mathbf{u}(z) \tag{C.116}$$

and

$$\mathbf{g}_0'(z) = s'(z)\mathbf{v}(z) + s(z)\mathbf{v}'(z) \tag{C.117}$$

$$\mathbf{g}_0'(z) = s''(z)\mathbf{v}(z) + 2s'(z)\mathbf{v}'(z) + s(z)\mathbf{v}''(z) \tag{C.118}$$

Note that, using $\delta(1) = 1$ also allows us to easily prove that

$$\mathbf{u}(1) = \boldsymbol{\mu} \tag{C.119}$$

where $\boldsymbol{\mu}$ is the invariant probability vector of the matrix $\mathbf{A}$, and

$$\mathbf{v}(1) = \mathbf{e} \tag{C.120}$$

C.4 Perron–Frobenius Eigenvalue Derivatives for Multiple Sources

In practice there will be a number of sources generating arrivals independently of each other. This overall Markov arrival process can be described by a transition probability generating matrix that is formed from the Kronecker product of the individual source transition probability generating matrices. That is

$$\mathbf{AP}(z) = \bigotimes_{i=1}^{N} \mathbf{A}_i\mathbf{P}_i(z) \tag{C.121}$$

where the i subscript indicates the relevant matrix corresponds to the ith source, and N is the number of sources. From the properties of Kronecker products [34] it is then possible to also establish

$$\delta(z) = \prod_{i=1}^{N} \delta_i(z) \tag{C.122}$$

$$\mathbf{u}(z) = \bigotimes_{\imath=1}^{N} \mathbf{u}_\imath(z) \tag{C.123}$$

$$\mathbf{v}(z) = \bigotimes_{i=1}^{N} \mathbf{v}_i(z) \tag{C.124}$$

The derivatives of the overall eigenvalue $\delta(z)$ are then given on investigation by

$$\begin{aligned}
\delta'(z) &= \sum_{i=1}^{N} \delta_i'(z) && \text{(C.125)}\\
\delta''(z) &= \sum_{i=1}^{N} \delta_i''(z) - \sum_{i=1}^{N} \delta_i'(z)^2 + \delta'(z)^2 && \text{(C.126)}\\
\delta'''(z) &= \sum_{i=1}^{N} \delta_i'''(z) + 3\delta'(z) \sum_{i=1}^{N} \left(\delta_i''(z) - \delta_i'(z)^2\right) \\
&\quad - 3\sum_{i=1}^{N} \delta_i'(z)\delta_i''(z) + \delta'(z)^3 + 2\sum_{i=1}^{N} \delta_i'(z)^3 && \text{(C.127)}
\end{aligned}$$

The first derivative of $\delta_i(z)$ evaluated at $z = 1^-$ is equal to λ_i — the steady state average number of arrivals generated by source i in each time slot. For binary sources (that generate either 0 or 1 arrivals in each time slot) it is a simple matter to show that

$$\begin{aligned}
\delta'(z) &= \lambda && \text{(C.128)}\\
\delta''(1) &= \sum_{i=1}^{N} \delta_i''(1) + M_2 - \lambda && \text{(C.129)}\\
\delta'''(1) &= \sum_{i=1}^{N} \delta_i'''(1) + 3\sum_{i=1}^{N} (\lambda - \lambda_i)\, \delta_i''(1) + M_3 - \lambda - 3\,(M_2 - \lambda) && \text{(C.130)}
\end{aligned}$$

where λ is the stationary average number of arrivals generated by all N sources in total in each time slot, and M_2 and M_3 are the second and third moments of this quantity respectively, given by

$$\begin{aligned}
\lambda &= \sum_{i=1}^{N} \lambda_i && \text{(C.131)}\\
M_2 &= \lambda + \lambda^2 - \sum_{i=1}^{N} \lambda_i^2 && \text{(C.132)}\\
M_3 &= \lambda + 3\lambda^2 + \lambda^3 - 3\,(1+\lambda) \sum_{i=1}^{N} \lambda_i^2 + 2\sum_{i=1}^{N} \lambda_i^3 && \text{(C.133)}
\end{aligned}$$

C.5 The Geometric-Geometric Special Case

The geom-geom arrival process can be treated as a special case of the phase-geom process, where the active periods have only one phase, and hence can be described by a geometric process. These geom-geom processes have the advantage that they have closed form expressions for their eigenvalues, and allow explicit representation of the eigenvectors. These results are presented below, both for the general and Perron–Frobenius eigensystem analyses.

For notation, we represent the geom-geom Markov process by a state transition matrix of the form

$$\mathbf{A} = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{bmatrix} \tag{C.134}$$

The probability generating function matrix is

$$\mathbf{P}(z) = \begin{bmatrix} 1 & 0 \\ 0 & p(z) \end{bmatrix} \tag{C.135}$$

giving

$$\mathbf{AP}(z) = \begin{bmatrix} \alpha & (1-\alpha)\,p(z) \\ 1-\beta & \beta p(z) \end{bmatrix} \tag{C.136}$$

Denoting the average number of arrivals generated in each time slot by λ, and the autocorrelation parameter of the source by γ, gives

$$\alpha \;=\; 1-(1-\gamma)\frac{\lambda}{p'(1)} \tag{C.137}$$

$$\beta \;=\; \gamma+(1-\gamma)\frac{\lambda}{p'(1)} \tag{C.138}$$

where $\gamma = \alpha+\beta-1$.

C.5.1 Geometric-geometric expressions for $\omega_n(z)$, $\mathbf{h}_n(z)$, and $\mathbf{g}_n(z)$

We have for $n = 0, 1$

$$\omega_n(z) = \frac{\alpha+\beta p(z)}{2} + (-1)^n\sqrt{\left(\frac{\alpha+\beta p(z)}{2}\right)^2 - \gamma p(z)} \tag{C.139}$$

with

$$\mathbf{h}_n(z) = (-1)^n\frac{\omega_n(z)-\beta p(z)}{\omega_1(z)-\omega_0(z)}\begin{bmatrix} -1 & \frac{\omega_{1-n}(z)-\beta p(z)}{1-\beta} \end{bmatrix} \tag{C.140}$$

and

$$\mathbf{g}_n(z) = \begin{bmatrix} 1 \\ \frac{1-\beta}{\omega_n(z)-\beta p(z)} \end{bmatrix} \tag{C.141}$$

We also obtain

$$\mathbf{h}_1(1)\mathbf{g}_0'(1) = \frac{-\lambda\gamma}{1-\gamma} \tag{C.142}$$

C.5.2 Geometric-geometric expressions for $\delta(z)$, $\mathbf{u}(z)$, and $\mathbf{v}(z)$ and derivatives

Here we have $\delta(z) = \omega_0(z)$ or

$$\delta(z) = \frac{\alpha+\beta p(z)}{2} + \sqrt{\left(\frac{\alpha+\beta p(z)}{2}\right)^2 - \gamma p(z)} \tag{C.143}$$

with

$$\mathbf{u}(z) = \frac{\delta(z) - p(z)}{\delta(z)\,(1 - p(z))} \begin{bmatrix} 1 & \frac{(1-\alpha)p(z)}{\delta(z)-\beta p(z)} \end{bmatrix} \tag{C.144}$$

and

$$\mathbf{v}(z) = \frac{(1 - p(z))\,(p(z)\delta'(z) - p'(z)\delta(z))}{p'(z)\,(p(z) - \delta(z))} \begin{bmatrix} 1 \\ \frac{1-\beta}{\delta(z)-\beta p(z)} \end{bmatrix} \tag{C.145}$$

Substituting for α and β, and simplifying, the first three derivatives of $\delta(z)$ become

$$\begin{aligned}
\delta'(1) &= \lambda && \text{(C.146)}\\
\delta''(1) &= 2\lambda\,(p'(1) - \lambda)\,\frac{\gamma}{1-\gamma} + \frac{\lambda}{p'(1)}p''(1) && \text{(C.147)}\\
\delta'''(1) &= 6\lambda\,(p'(1) - \lambda)\,\frac{(p'(1) - \lambda)\,\gamma^2 - \lambda\gamma}{(1-\gamma)^2}\\
&\quad + \frac{6\lambda\,(p'(1) - \lambda)\,\gamma}{p'(1)\,(1-\gamma)}p''(1) + \frac{\lambda}{p'(1)}p'''(1) && \text{(C.148)}
\end{aligned}$$

Similarly, the first two derivatives of the right-hand eigenvector become

$$\begin{aligned}
v_0'(1) &= \frac{-\gamma\lambda}{1-\gamma} && \text{(C.149)}\\
v_0''(1) &= \frac{2\lambda\gamma\,(\lambda - (1+\gamma)\,(p'(1) - \lambda))}{(1-\gamma)^2} - \frac{\gamma\lambda}{p'(1)\,(1-\gamma)}p''(1) && \text{(C.150)}
\end{aligned}$$

and

$$\begin{aligned}
v_1'(1^-) &= \frac{(p'(1) - \lambda)\,\gamma}{1-\gamma} && \text{(C.151)}\\
v_1''(1^-) &= \frac{2\gamma\,(p'(1) - \lambda)\,(p'(1)\gamma - \gamma\lambda - 2\lambda)}{(1-\gamma)^2} + \frac{(p'(1) - \lambda)\,\gamma}{p'(1)\,(1-\gamma)}p''(1) && \text{(C.152)}
\end{aligned}$$

Appendix D

The Autocorrelation Sum of a Phase-Geom Binary Source

In this appendix we show that the form of the autocorrelation parameter described by Neuts in [98] for phase-geom binary sources and adopted in this thesis has a direct physical relationship to the observable autocorrelation coefficient function. To be exact, we will prove that the single-sided infinite sum S of the autocorrelation coefficient function is given in terms of the autocorrelation parameter γ by

$$S = \sum_{m=1}^{\infty} R(m) = \frac{\gamma}{1-\gamma} \tag{D.1}$$

where $R(m)$ is the autocorrelation coefficient function (the autocovariance normalised to the variance of the process).

Although this relationship is simple to prove, and indeed is well known, for geom-geom binary sources, the fact that it might apply to the more general phase-geom binary model was first proposed only recently by Pieloor and Lewis in [107]. Although the authors were unable to prove this relation, they successfully used it to accurately predict simulation results in that paper. Previous attempts to prove the relation using matrix algebraic notation were unsuccessful, although they did allow the relation to be confirmed symbolically for transition matrices of size 3 × 3 and 4 × 4 using *Mathematica* [139]. In the following we present a formal proof of the relation.

Theorem D.1 *The single sided autocorrelation sum of a phase-geom binary process with an autocorrelation parameter γ described by*

$$\gamma = 1 - \frac{2\eta_1}{(1-\lambda)(\eta_2+\eta_1)}$$

is given by

$$S = \frac{\gamma}{1-\gamma}$$

where λ is the average generating rate of the process, η_r is the rth moment of its active period, and S is defined by

$$S = \sum_{m=1}^{\infty} R(m)$$

where $R(m)$ is the autocorrelation coefficient function of the phase-geom binary process.

Proof. *Define the counting process $N(t)$ to be the number of events (arrivals or departures) occurring in an arbitrary time interval of t time slots. The* asymptotic variance ratio *of the process $N(t)$ is defined as*

$$v = \lim_{t \to \in\infty} \frac{1}{t} \operatorname{Var}\left[N(t)\right] \tag{D.2}$$

for which it can be shown (using the well known result for the variance of the sum — see [6] for example) that

$$v = \sigma^2 \left(1 + 2S\right) \tag{D.3}$$

where σ^2 is the variance of the process. In [19], Daley points out that the asymptotic variance of the net arrival process to an infinite buffer queue is equal to the asymptotic variance of the aggregate server process, or $v_{out} = v_{in}$[1].

Consider an infinite buffer, single server discrete-time queue with an input process formed from the superposition of N geom-geom binary arrival processes, with average arrival rates and autocorrelation parameters described respectively by λ_i and γ_i for $i = 1, 2, \ldots, N$. The stationary (or marginal) arrival process from these N sources has a combined average arrival rate denoted by λ, with variance σ^2. From Chapter 7 we know that the autocorrelation parameter γ (no subscript) of the phase-geom binary process describing the output *of this queue is given by*

$$\frac{\gamma}{1-\gamma} = \frac{\sigma^2 + \lambda^2 - \lambda}{2\lambda\left(1-\lambda\right)} + \frac{1}{\lambda\left(1-\lambda\right)} \sum_{i=1}^{N} \lambda_i \left(1 - \lambda_i\right) \frac{\gamma_i}{1-\gamma_i} \tag{D.4}$$

It is simple matter to show that the autocorrelation coefficient function of geom-geom binary process i is given by

$$R_i(m) = \gamma_i^{|m|} \tag{D.5}$$

[1] Daley actually discusses this result in terms of the index of dispersion of intervals (IDI), while the definition of the asymptotic variance ratio used here is closely related to the index of dispersion of counts (IDC) The IDC and IDI have the same limiting behaviour however [40] and so the result still holds.

and hence the autocorrelation parameter of the combined arrival process will be described by

$$R(m) = \frac{1}{\sigma^2} \sum_{i=1}^{N} \sigma_i^2 \gamma_i^{|m|} \tag{D.6}$$

where $\sigma_i^2 = \lambda_i (1 - \lambda_i)$ *since the sources are binary, and* $\sigma^2 = \sum_{i=1}^{N} \sigma_i^2$. *Thus, the single sided sum of the autocorrelation function of the input process is given by*

$$S_{in} = \frac{1}{\sigma^2} \sum_{i=1}^{N} \lambda_i (1 - \lambda_i) \frac{\gamma_i}{1 - \gamma_i} \tag{D.7}$$

Rearranging Daley's result gives

$$\begin{aligned} S_{out} &= \frac{\sigma^2 - \sigma_{out}^2}{2\sigma_{out}^2} + \frac{\sigma^2}{\sigma_{out}^2} S_{in} \\ &= \frac{\sigma^2 + \lambda^2 - \lambda}{2\lambda(1-\lambda)} + \frac{1}{\lambda(1-\lambda)} \sum_{i=1}^{N} \lambda_i (1 - \lambda_i) \frac{\gamma_i}{1 - \gamma_i} \end{aligned} \tag{D.8}$$

where $\sigma_{out}^2 = \lambda(1 - \lambda)$ *is the variance of the server output process. Comparing this result to equation (D.4) the desired relation is obtained, and the proof completed.* ■

Appendix E

Iterative Queueing Solutions

One means of providing an independent source of confirmation for queue distribution results is to use iterative solution methods. We will outline this method here firstly for a general discrete-time Markov modulated arrival process, and then discuss some of the specifics of the IBP and cyclic service problems.

E.1 Markov Modulated Arrival Processes

We assume that time is slotted, and that service of the queue occurs at the slot boundaries. The state of the arrival process changes at the start of each time slot, and the arrivals for the current time slot are then generated. Let $q_{n,s}(t)$ denote the probability that the queue population observed at the beginning of the tth time slot (immediately after service) is n and that the state of the arrival process was s at the end of the last time slot. Similarly, let $q^{+}_{n,s}(t)$ denote the same quantity observed at the end of the tth time slot for the current arrival process being s. In both cases, s denotes the state of the arrival process that generated the last arrivals to the queue.

Let $\alpha_{r,s}$ denote the probability that the next state of the combined arrival process will be s given that the last state was r, and let $p_{s,k}$ denoted the stationary probability that there will be k arrivals generated by the combined arrival process when it is in state s. The buffer is assumed to be finite in capacity, with size K (we can hold K cells waiting for service in addition to the cell, if any, currently receiving service). For $n < K$ we obtain

$$q^{+}_{n,s}(t) = \sum_{\imath=0}^{n} \sum_{r=0}^{m-1} q_{i,r}(t)\alpha_{r,s}p_{s,n-\imath} \qquad \text{(E.1)}$$

where m is the number of states in the Markov chain describing the combined arrival process. For $n = K$

$$q^+_{K,s}(t) = \sum_{i=0}^{K} \sum_{r=0}^{m-1} q_{i,r}(t)\alpha_{r,s} \sum_{k=K-i}^{\infty} p_{s,k} \tag{E.2}$$

where $q^+_{n,s}(t)$ is of course equal to zero for $n > K$. These relations describe the effect of the arrival process on the development of the queue population. We can similarly summarise the service process as

$$q_{n,s}(t+1) = \begin{cases} q^+_{0,s}(t) + q^+_{1,s}(t) & \text{for } n = 0 \\ q^+_{n+1,s}(t) & \text{for } 1 \leq n < K \\ 0 & \text{for } n \geq K \end{cases} \tag{E.3}$$

This approach is ideal for the analysis of transient behaviour, since the slot to slot variation in the probability distribution of the queue population is described exactly by equations (E.1), (E.2), and (E.3). For stationary or steady state behaviour we are limited to iterating these relations until the change in some characteristic between iterations (or slots) is small enough to be acceptable. A common choice in this regard is the average queue population, where iteration of the queue problem proceeds until

$$\left| \frac{L_q(t)}{L_q(t-1)} - 1 \right| < \epsilon$$

for some suitably small ϵ, where

$$L_q = \sum_{n=0}^{K} n \sum_{s=0}^{m-1} q_{n,s}(t-1)$$

is the average queue population. Other choices are the variance of the queue population (see Chapter 4) and the average loss probability (see Chapter 6 and the following).

This iterative solution approach is *many* times slower than the probability generating function method discussed in Chapters 2, 3, and 5 when calculating solutions approximating the infinite buffer (large K). It has the advantage however that it is very robust, and returns the entire probability distribution rather than just the average and variance. It is particularly well suited though to calculating loss probabilities for finite buffers, since it is by nature a finite buffer implementation. In this regard it is perhaps the best method available for finite buffer analysis, although matrix geometric techniques can also be applied to this purpose.

Denote the loss probability for a buffer of size K by ξ_K, and denote the average number of cells departing the queueing system in each time slot by ρ. If there are an average of λ attempted arrivals per time slot, then we have

$$\lambda\xi_K = \lambda - \rho \tag{E.4}$$

where $\lambda\xi_K$ denotes the average number of cells per time slot lost due to overflow of the finite buffer. Additionally, since departures from the queueing system can only occur when the server visits a non-empty queue, we see that the probability that the system is empty is given by $1 - \rho$. Hence, the steady state loss probability is obtained simply by

$$\xi_K = 1 - \frac{1}{\lambda}\left(1 - \lim_{t\to\infty} \sum_{s=0}^{m-1} q_{0,s}^{+}(t)\right) \tag{E.5}$$

where, for small K, the iteration process converges quite quickly to the steady state.

E.2 Application to the Geometric-Geometric IBP Queueing Problem

In this section we will present some *C* code fragments for implementing the numeric iterative solution method for queueing systems subject to geom-geom IBP sources. The method used here represents only one of many possible ways to implement the same procedure, but is the one used by the author to generate the various results in this thesis. The details are reproduced here for the interested reader.

C Code Implementation

The following *C* code fragment performs a single iteration of the queueing system. The vectors[1] `q_vector` and `q_plus_vector` describe the $q_{n,s}(t)$ and $q_{n,s}^{+}(t)$ probability distributions respectively by amalgamating the population n and state s into a single index given by $nm + s$, where m is the number of states in the combined arrival process (2^N in this case). The current slot number t is assumed to be dealt with elsewhere, since it has no bearing on the implementation of the iteration process. The array `active_sources` describes the number of active sources in the indexed state, while `transition_matrix` and `arrivals_matrix` describe the state transition and arrival probabilities respectively. Note that `q_vector` and `q_plus_vector` both have $(K + 1)\,m$ elements.

[1] Vectors are defined here as one dimensional arrays of double precision floating point numbers indexed from zero. Matrices are similarly defined, but for two dimensions.

```
for (i=0 ; i<K*m+K ; ++i) q_plus_vector[i] = 0.0;
for (i=0 ; i<m ; ++i) {
    for (j=0 ; j<m ; ++j) {
        for (k=0 ; k<K ; ++k) {
            x = q_vector[k*m+i]*transition_matrix[i][j];
            for (n=0 ; n<=active_sources[j] ; ++n) {
                next_k = k + n;
                if (next_k > K) next_k = K;
                q_plus_vector[next_k*m+j] += x*arrivals_matrix[j][n];
            }
        }
    }
}
for (i=0 ; i<m ; ++i) q_vector[i] = q_plus_vector[i] + q_plus_vector[i+m];
for (i=m ; i<K*m ; ++i) q_vector[i] = q_plus_vector[i+m];
```

For the above process to work of course, the relevant arrays of data must be defined. In the following we list the code for performing this operation under the assumption that Lambda, Theta, and Gamma are arrays of N elements containing λ_i, θ_i, and γ_i for each of the N sources. In addition we require Alpha and Beta to contain α_i and β_i where

$$\alpha_i = 1 - (1 - \gamma_i)\frac{\lambda_i}{\theta_i} \tag{E.6}$$

$$\beta_i = \gamma_i + (1 - \gamma_i)\frac{\lambda_i}{\theta_i} \tag{E.7}$$

We also assume that the state of the overall process describes the state of each of the individual IBP sources using a binary assignment. The order of the assignment is unimportant as long as it is consistent within the program.

With these quantities defined, the $m \times m$ state to state transition probabilities matrix is calculated using

```
for (i=0 ; i<m ; ++i) {
    for (j=0 ; j<m ; ++j) {
        x = 1.0;
        for (k=0 ; k<N ; ++k) {
            last = (i >> k) % 2;
            next = (j >> k) % 2;
            if ((last == 0) && (next == 0)) x *= Alpha[k];
            if ((last == 0) && (next == 1)) x *= (1.0 - Alpha[k]);
            if ((last == 1) && (next == 0)) x *= (1.0 - Beta[k]);
            if ((last == 1) && (next == 1)) x *= Beta[k];
        }
        transition_matrix[i][j] = x;
    }
}
```

The entries in arrivals_matrix are calculated using the following fragment. This is an $m \times N$ matrix describing the probability that a particular number of arrivals is generated when the arrival process is in each state.

```
for (i=0 ; i<m ; ++i) for (j=0 ; j<=N ; ++j) arrivals_matrix[i][j] = 0.0;
arrivals_matrix[0][0] = 1.0;
for (i=1 ; i<m ; ++i) {
    k = 0;
    for (j=0 ; j<N ; ++j) {
        if (i >> j % 2 == 1) {
            temporary_vector[k] = Theta[j];
            ++k;
        }
    }
    num = 1 << active_sources[i];
    for (j=0 ; j<num ; ++j) {
        n = 0;
        x = 1.0;
        for (k=0 ; k<active_sources[i] ; ++k) {
            if (j >> k % 2 == 1) {
                x *= temporary_vector[k];
                ++n;
            }
            else x *= 1.0 - temporary_vector[k];
        }
        arrivals_matrix[i][n] += x;
    }
}
```

The last component is the active_sources array, which is simply calculated due to the binary assignment approach.

```
for (i=0 ; i<m ; ++i) {
    active_sources[i] = 0;
    for (j=0 ; j<N ; ++j) if (i >> j % 2 == 1) ++active_sources[i];
}
```

Initialisation

Before the first iteration is performed, the q_vector must be initialised to some appropriate set of values. Since this vector is the program implementation of the probability distribution $\mathbf{q}(t) = \{q_{n,s}(t)\}$ its entries must all be greater than or equal to zero, with a sum of exactly 1. The most common initial state to use is the empty queue, which still leaves the problem of assigning the probabilities of the m states corresponding to a queue population of zero.

We can of course assume that the starting state of the arrival process is also zero, so that $q_{0,0}(0) = 1$ with all other entries equal to zero. Alternatively, inspection of the arrival and service relations for $q_{n,s}(t)$ as $t \rightarrow \infty$ shows that in the steady state

$$\sum_{n=0}^{K} q_{n,s}(\infty) = \mu_s \tag{E.8}$$

where μ_s is the steady state probability that the combined arrival process will be in state s. Thus we use instead

$$q_{n,s}(0) = \begin{cases} \mu_s & \text{for } n = r \\ 0 & \text{otherwise} \end{cases} \tag{E.9}$$

where r is the starting queue population (i.e. we are not just restricting ourselves to $r = 0$). The μ_s quantities are calculated from the parameters of the geom-geom IBP sources as shown in the following code fragment.

```
for (i=0 ; i<m ; ++i) {
        mu_vector[i] = 1.0;
        for (j=0 ; j<N ; ++j) {
                if (i >> j % 2 == 1) mu_vector[i] *= Lambda[j];
                else mu_vector[i] *= 1.0 - Lambda[j];
        }
}
```

Note that arbitrary assignments of values $q_{n,s}(0)$ can result in phantom convergence — that is, the iterative process stops at an incorrect result even though the convergence criteria has been satisfied. The reason for this is that the observed convergence measure may not proceed from its starting value to its steady-state value in a smooth or direct manner, but may move from the starting point to some other point outside of the range between the starting and steady-state values before then proceeding towards the steady-state. If this happens there will be some point in the iterative process where the change in the observed quantity between iterations reverses sign (i.e. the measure begins to decrease instead of increase). If this reversal is 'slow' enough, it may appear that the observed quantity has converged. This is the main reason why a starting queue population of 0 is most often used.

Other Considerations

As the number of iterations increases, the sum of the probabilities in the q_vector will slowly diverge from 1 due to accumulation of round off errors. With double precision arithmetic this will very likely have a negligible effect for the first 10^5 or so iterations, and rarely will this many be required. If large numbers of iterations are going to be encountered the program should periodically correct for the error in the sum.

Another problem that can occur is that the observed quantity of the iterative process does not continue to converge past a certain point. Combinations of round off errors and other precision effects seem to be the cause of this problem. If the iteration program merely continues until the desired convergence is achieved, the program may iterate indefinitely. Consequently it is a good idea to periodically check that the process is still converging by confirming that the current relative change in the observed variable is less than the relative change observed at the last check time.

E.3 Application to Cyclic Service Queueing Problems

In Chapter 5 we considered the analysis of queueing systems subject to arrivals from both a single cyclic source with parameters b and C, and from N geom-geom IBP sources. Rather than attempt to model the cyclic source as another arrival process like the IBP sources (thereby increasing the number of Markov states from 2^N to $2^N C$) we exploit the time periodic nature of the process. In those time slots for which the cyclic source is active, a single additional arrival to the queue is generated, while for the remaining slots the iterative procedure is identical to that of the IBP sources alone. This results in an implementation of the iterative process itself as given by the following.

```
for (position=0 ; position<C ; ++position) {
    for (i=0 ; i<K*m+K ; ++i) q_plus_vector[i] = 0.0;
    for (i=0 ; i<m ; ++i) {
        for (j=0 ; j<m ; ++j) {
            for (k=0 ; k<K ; ++k) {
                x = q_vector[k*m+i]*transition_matrix[i][j];
                for (n=0 ; n<=active_sources[j] ; ++n) {
                    next_k = k + n;
                    if (next_k > K) next_k = K;
                    q_plus_vector[next_k*m+j] += x*arrivals_matrix[j][n];
                }
            }
        }
    }
    if (position < C-b) {
        for (i=0 ; i<m ; ++i)
            q_vector[i] = q_plus_vector[i] + q_plus_vector[i+m];
        for (i=m ; i<K*m ; ++i) q_vector[i] = q_plus_vector[i+m];
    }
    else {
        for (i=0 ; i<K*m ; ++i) q_vector[i] = q_plus_vector[i];
        for (i=K*m-K ; i<K*m ; ++i) q_vector[i] += q_plus_vector[i+m];
    }
}
```

The constant vectors and matrices are generated in the same manner as for the IBP only case, wit h the main difference in the implementations being one ‘iteration’ now increases the time index by C rather than by 1 as in the previously discussed case.

Appendix F

Miscellaneous Mathematics

Many of the derivations presented in the body of this thesis rely on standard or simple mathematical relations, particularly in regard to multiple sums of probabilities. For convenience, the more useful of these are presented here.

F.1 Sums of Series

Theorem F.1 *For* $|r| < 1$

$$
\begin{aligned}
\sum_{i=1}^{\infty} r^i &= \frac{r}{1-r} \\
\sum_{i=1}^{\infty} i r^i &= \frac{r}{(1-r)^2} \\
\sum_{i=1}^{\infty} i^2 r^i &= \frac{r}{(1-r)^3}(1+r) \\
\sum_{i=1}^{\infty} i^3 r^i &= \frac{r}{(1-r)^4}(1+4r+r^2)
\end{aligned}
$$

Proof. *Only the second case above is proved here, since the other cases are proved in a like fashion.*

$$
\begin{aligned}
S(n) &= \sum_{i=1}^{n} i r^i = r + 2r^2 + 3r^3 + \dots + nr^n \\
S(n) - rS(n) &= (r + 2r^2 + 3r^3 + \dots + nr^n) - (r^2 + 2r^3 + 3r^4 + \dots + nr^{n+1}) \\
&= (r + r^2 + r^3 + \dots + r^n) - nr^{n+1} \\
S(n) &= \tfrac{1}{(1-r)} \left(\sum_{i=1}^{n} r^i - nr^{n+1}\right) \\
S(\infty) &= \tfrac{1}{(1-r)} \sum_{i=1}^{\infty} r^i = \frac{r}{(1-r)^2}
\end{aligned}
$$

■

Theorem F.2 *For $|r| < 1$, and letting $\binom{n}{x}$ denote the binomial coefficient function, we have*

$$\begin{aligned}
\sum_{x=0}^{\infty} \binom{x+c}{c} r^x &= \frac{1}{(1-r)^{c+1}} \\
\sum_{x=0}^{\infty} x \binom{x+c}{c} r^x &= \frac{r\,(c+1)}{(1-r)^{c+2}} \\
\sum_{x=0}^{\infty} x^2 \binom{x+c}{c} r^x &= \frac{r\,(c+1)\,(rc+r+1)}{(1-r)^{c+3}}
\end{aligned}$$

Proof. *As for Theorem F.1, we only present a proof for the second case — the other two are similar. We start by denoting*

$$S_c(n) = \sum_{x=0}^{n} \binom{x+c}{c} r^x$$

then

$$\begin{aligned}
S'_c(n) &= \sum_{x=0}^{n} x \binom{x+c}{c} r^x \\
S'_c(n) - rS'_c(n) &= \sum_{x=0}^{n} x \binom{x+c}{c} r^x - \sum_{x=0}^{n} x \binom{x+c}{c} r^{x+1} \\
&= \sum_{x=0}^{n-1} (x+1) \binom{x+1+c}{c} r^{x+1} - \sum_{x=0}^{n-1} (x+1) \binom{x+c}{c} r^{x+1} \\
&\quad + \sum_{x=0}^{n-1} \binom{x+c}{c} r^{x+1} - n \binom{n+c}{c} r^{n+1} \\
&= \sum_{x=0}^{n-1} (x+1) \left(\binom{x+1+c}{c} - \binom{x+c}{c} \right) r^{x+1} + r \sum_{x=0}^{n-1} \binom{x+c}{c} r^x \\
&\quad - n \binom{n+c}{c} r^{n+1} \\
&= S'_{c-1}(n) + rS_c(n-1) - n \binom{n+c}{c} r^{n+1}
\end{aligned}$$

therefore

$$\begin{aligned}
S'_c(\infty) &= \frac{S'_{c-1}(\infty) + rS_c(\infty)}{1-r} \\
&= \frac{S'_0(\infty)}{(1-r)^c} + \frac{r}{1-r} \sum_{i=1}^{c} \frac{S_{c-i}(\infty)}{(1-r)^i} \qquad \textit{(by inspection)} \\
&= \frac{1}{(1-r)^c} \sum_{x=0}^{\infty} x r^x + \frac{r}{1-r} \sum_{i=1}^{c} \frac{1}{(1-r)^{c+1}} \\
&= \frac{r\,(c+1)}{(1-r)^{c+2}}
\end{aligned}$$

■

Theorem F.3 *For some stochastic matrix* $\mathbf{X}$ *such that* $\mathbf{I}-\mathbf{X}$ *is non-singular*

$$\begin{aligned}
\sum_{i=1}^{\infty}\mathbf{X}^{i} &= (\mathbf{I}-\mathbf{X})^{-1}\mathbf{X}\\
\sum_{i=1}^{\infty}i\mathbf{X}^{i} &= (\mathbf{I}-\mathbf{X})^{-2}\mathbf{X}\\
\sum_{i=1}^{\infty}i^{2}\mathbf{X}^{i} &= (\mathbf{I}-\mathbf{X})^{-3}(\mathbf{I}+\mathbf{X})\,\mathbf{X}\\
\sum_{i=1}^{\infty}i^{3}\mathbf{X}^{i} &= (\mathbf{I}-\mathbf{X})^{-4}\left(\mathbf{I}+4\mathbf{X}+\mathbf{X}^{2}\right)\mathbf{X}
\end{aligned}$$

Proof. *Follows from the same approach as that used in Theorem F.1, keeping in mind the non-commutative nature of matrices.* ■

Theorem F.4 *For some* a_n *and* b_n

$$\sum_{n=0}^{K}\sum_{i=0}^{n}a_{i}b_{n-i}=\sum_{n=0}^{K}a_{n}\sum_{i=0}^{K-n}b_{i}$$

$$\sum_{n=0}^{\infty}\sum_{i=0}^{n}a_{i}b_{n-i}=\sum_{n=0}^{\infty}a_{n}\sum_{n=0}^{\infty}b_{n}$$

Proof. *By inspection.* ■

Theorem F.5 *For some* a_n *and* b_n

$$\sum_{n=0}^{K}\sum_{i=0}^{n}na_{i}b_{n-i}=\sum_{n=0}^{K}a_{n}\sum_{i=0}^{K-n}(n+i)\,b_{i}$$

$$\sum_{n=0}^{\infty}\sum_{i=0}^{n}na_{i}b_{n-i}=\sum_{n=0}^{\infty}b_{n}\sum_{n=0}^{\infty}na_{n}+\sum_{n=0}^{\infty}a_{n}\sum_{n=0}^{\infty}nb_{n}$$

Proof. *By inspection.* ■

Theorem F.6 *For some* a_n *and* b_n

$$\sum_{n=0}^{K}\sum_{i=0}^{n}n^{2}a_{i}b_{n-i}=\sum_{n=0}^{K}a_{n}\sum_{i=0}^{K-n}(n+i)^{2}\,b_{i}$$

$$\sum_{n=0}^{\infty}\sum_{i=0}^{n}n^{2}a_{i}b_{n-i}=\sum_{n=0}^{\infty}n^{2}a_{n}\sum_{n=0}^{\infty}b_{n}+2\sum_{n=0}^{\infty}na_{n}\sum_{n=0}^{\infty}nb_{n}+\sum_{n=0}^{\infty}a_{n}\sum_{n=0}^{\infty}n^{2}b_{n}$$

Proof. *By inspection.* ■

Theorem F.7 *For some a_n and b_n*

$$\sum_{n=0}^{K}\sum_{i=0}^{n} n^3 a_i b_{n-i} = \sum_{n=0}^{K} a_n \sum_{i=0}^{K-n} (n+i)^3 b_i$$

$$\begin{aligned}\sum_{n=0}^{\infty}\sum_{i=0}^{n} n^3 a_i b_{n-i} &= \sum_{n=0}^{\infty} n^3 a_n \sum_{n=0}^{\infty} b_n + 3\sum_{n=0}^{\infty} n^2 a_n \sum_{n=0}^{\infty} n b_n \\ &\quad + 3\sum_{n=0}^{\infty} n a_n \sum_{n=0}^{\infty} n^2 b_n + \sum_{n=0}^{\infty} a_n \sum_{n=0}^{\infty} n^3 b_n\end{aligned}$$

Proof. *By inspection.* ■

F.2 Moments of Sums of Binary Processes

A binary random process X_i can take on one of two values (0 or 1) and is found to have a value of 1 with steady state probability p_i. As a consequence, all the moments around the origin $m_r(X_i)$ have the same value of p_i. Let $m'_r(X_i)$ denote the rth moment of this arrival process around the mean $m_1(X_i)$. Then it is straight forward to show that

$$\begin{aligned} m'_1(X_i) &= 0 \\ m'_2(X_i) &= p_i - p_i^2 \\ m'_3(X_i) &= p_i - 3p_i^2 + 2p_i^3 \end{aligned}$$

Consider now the sum of N independent binary processes, denoted by $S = X_1 + X_2 + \cdots + X_N$. Then it can be shown that

$$\begin{aligned} m_1(S) &= \sum_{i=1}^{N} p_i \\ m'_2(S) &= \sum_{i=1}^{N} \left(p_i - p_i^2\right) \\ m'_3(S) &= \sum_{i=1}^{N} \left(p_i - 3p_i^2 + 2p_i^3\right) \end{aligned}$$

so that, using

$$\begin{aligned} m_2(S) &= m'_2(S) + m_1^2(S) \\ m_3(S) &= m'_3(S) + 3m_1(S)m_2(S) - 2m_1^3(S) \end{aligned}$$

gives

$$\begin{aligned} m_1(S) &= p \\ m_2(S) &= p + p^2 - \sum_{i=0}^{N-1} p_i^2 \\ m_3(S) &= p + 3p^2 + p^3 - 3\,(1+p) \sum_{i=0}^{N-1} p_i^2 + 2 \sum_{i=0}^{N-1} p_i^3 \end{aligned}$$

where $p = \sum_{i=1}^{N} p_i$.

F.3 Binomial Arrival Processes

For the binomial arrival process, the probability of there being i arrivals is given by

$$\Pr(i) = \binom{N}{i} p^i (1-p)^{N-i}$$

where N is the number of sources making up the binomial process, and p is the individual source probability, given by $p = \lambda/N$ for identical sources and a total average number of arrivals per discrete-time interval of λ. The first three moments of the arrival process are therefore given by

$$m_1 = \lambda$$

$$m_2 = \lambda\left(1 - \frac{\lambda}{N} + \lambda\right)$$

$$m_3 = \lambda\left(1 + 3\lambda - 3\frac{\lambda}{N} + \lambda^2 - 3\frac{\lambda^2}{N} + 2\frac{\lambda^2}{N^2}\right)$$

Bibliography

[1] R. G. Addie. Tails of stationary distributions of queued work. In *Proceedings IEEE INFOCOM'94*, pages 1170–1177, June 1994.

[2] R. G. Addie and M. Zukerman. Analysis of connection admission control and multiplexing gain in a B-ISDN based on a Gaussian traffic model. In *Proceedings Australian Broadband Switching and Services Symposium*, pages 1–8, July 1993.

[3] R. G. Addie and M. Zukerman. A Gaussian characterization of correlated ATM multiplexed traffic and related queueing studies. In *Proceedings ICC'93*, pages 1404–1408, May 1993.

[4] R. G. Addie and M. Zukerman. Queueing performance of a tree type ATM network. In *Proceedings IEEE INFOCOM'94*, pages 48–55, June 1994.

[5] H. Ahmadi and W. E. Denzel. A survey of modern high-performance switching techniques. *IEEE Journal on Selected Areas in Communications*, 7(7):1091–1103, Sept. 1989.

[6] A. O. Allen. *Probability, Statistics, and Queueing Theory: With Computer Science Applications.* Academic Press, New York, 1978.

[7] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell Systems Technical Journal*, 61(8):1871–1894, Oct. 1982.

[8] E. Arthurs and B. W. Stuck. Traffic analysis tools for integrated digital time-division link level multiplexing of synchronous and asynchronous message streams. *IEEE Journal on Selected Areas in Communications*, 1(6):1112–1123, Dec. 1983.

[9] ATM Forum. *ATM User-Network Interface Specification: Version 3.1.* 1994.

[10] G. A. Awater and F. C. Schoute. Optimal queueing policies for fast packet switching of mixed traffic. *IEEE Journal on Selected Areas in Communications*, 9(3):458–467, Apr. 1991.

[11] A. Baiocchi, N. B. Melazzi, M. Listanti, A. Roveri, and R. Winkler. Loss performance analysis of an ATM multiplexer loaded with high speed on-off sources. *IEEE Journal on Selected Areas in Communications*, 9(3):388–393, Apr. 1991.

[12] F. Bonomi, S. Mantagna, and R. Paglino. Busy period analysis for an ATM switching element output line. In *Proceedings IEEE INFOCOM'92*, pages 544–551, May 1992.

[13] O. J. Boxma and W. P. Groenendijk. Waiting times in discrete-time cyclic-service systems. *IEEE Transactions on Communications*, 36(2):164–170, Feb. 1988.

[14] H. Bruneel. On the behaviour of buffers with random server interruptions. *Performance Evaluation*, 3:165–175, 1983.

[15] H. Bruneel. Queueing behaviour of statistical multiplexers with correlated inputs. *IEEE Transactions on Communications*, 36(12):1339–1341, Dec. 1988.

[16] C. G. Chang and H. H. Tan. Queueing analysis of explicit policy assignment push-out buffer sharing schemes for ATM networks. In *Proceedings IEEE INFOCOM'94*, pages 500–509, June 1994.

[17] H. J. Chao. A novel architecture for queue management in the ATM network. *IEEE Journal on Selected Areas in Communications*, 9(7):1110–1118, Sept. 1991.

[18] I. Cidon, R. Guérin, and A. Khamisy. On protective buffer policies. *IEEE/ACM Transactions on Networking*, 2(3):240–246, June 1994.

[19] D. J. Daley. Queueing output processes. *Advances in Applied Probability*, 8:395–415, 1976.

[20] M. D'Ambrosio and R. Melen. Evaluating the limit behaviour of the ATM traffic with a network. *IEEE Transactions on Communications*, 3(6):832–841, Dec. 1995.

[21] W. E. Denzel, A. P. J. Engbersen, and I. Iliadis. A flexible shared-buffer switch for ATM at Gb/s rates. *Computer Networks and ISDN Systems*, 27(4):611–624, Jan. 1995.

[22] E. Desmet, B. Steyaert, H. Bruneel, and G. H. Petit. Tail distributions of queue length and delay in discrete-time multiserver queueing models, applicable in ATM networks. In *Proceedings ITC-13*, pages 1–6, June 1991.

[23] L. G. Dron, G. Ramamurthy, and B. Sengupta. Delay analysis of continuous bit rate traffic over an ATM network. *IEEE Journal on Selected Areas in Communications*, 9(3):402–407, Apr. 1991.

[24] H. Dupuis and B. Hajek. Simple formulas for multiplexing delay for independent regenerative sources. In *Proceedings IEEE INFOCOM'93*, pages 28–34, Mar. 1993.

[25] A. Erramilli, J. Gordon, and W. Willinger. Applications of fractals in engineering for realistic traffic processes. In *Proceedings ITC-14*, pages 35–44, June 1994.

[26] A. Erramilli, R. P. Singh, and P. Pruthi. Chaotic maps as models of packet traffic. In *Proceedings ITC-14*, pages 329–338, June 1994.

[27] F. B. A. R. Ferretti, M. Listanti, and G. Zingrillo. ATM system buffer design under very low cell loss probability constraints. In *Proceedings IEEE INFOCOM'91*, pages 929–938, Apr. 1991.

[28] J. Filipiak. *Modelling and Control of Dynamic Flows in Communication Networks.* Springer-Verlag, New York, 1988.

[29] J. Filipiak. M-Architecture: A structural model of traffic management and control in broadband ISDNs. *IEEE Communications Magazine*, 27(5):25–31, May 1989.

[30] N. L. S. Fonseca and J. A. Silvester. Modelling the output process of an ATM multiplexer with Markov modulated arrivals. In *Proceedings ICC'94*, pages 721–725, May 1994.

[31] M. R. Frater, J.Walrand, and B. D. O. Anderson. Optimally efficient simulation of buffer overflows in queues with deterministic service times via importance sampling. *Australian Telecommunications Research*, 24(1):1–8, 1990.

[32] O. Gihr and P. Tran-Gia. A layered description of ATM cell traffic streams and correlation analysis. *Australian Telecommunications Research*, 24(2):9–17, 1990.

[33] J. J. Gordon. Modelling bursty traffic with two-state sources. *Australian Telecommunications Research*, 24(2):51–63, 1990.

[34] A. Graham. *Kronecker Products and Matrix Calculus: with Applications.* John Wiley and Sons, 1981.

[35] A. Gravey and G. Hebuterne. Mixing time and loss priorities in a single server queue. In *Proceedings ITC-13*, pages 47–52, 1991.

[36] S. Grossman. *Multivariable Calculus, Linear Algebra, and Differential Equations.* Harcourt Brace Jovanovich, 2nd edition, 1986.

[37] R. Grünenfelder, C. Gallego, and Y. Gachoud. Measurement and characterization of cell loss in ATM switches. In *Proceedings IEEE GLOBECOM*, pages 1075–1079, Nov. 1994.

[38] R. Guérin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, Sept. 1991.

[39] A. K. Gupta and N. D. Georganas. Priority performance of ATM packet switches. In *Proceedings IEEE INFOCOM'92*, pages 727–733, May 1992.

[40] R. Gusella. Characterising the variability of arrival processes with indexes of dispersion. *IEEE Journal on Selected Areas in Communications*, 9(2):203–211, Feb. 1991.

[41] D. Habibi, D. J. H. Lewis, D. T. Nguyen, and J. Pieloor. Analysis of an access node multiplexer in a system serving CBR and VBR traffic. *Computer Communications*, 16(12):776–780, Dec. 1993.

[42] R. Händel and M. N. Huber. *Integrated Broadband Networks: An Introduction to ATM-Based Networks.* Addison–Wesley Publishers, 1991.

[43] P. G. Harrison and N. M. Patel. *Performance Modelling of Communication Networks and Computer Architectures.* Addison-Wesley, 1993.

[44] V. F. Hartanto. *User-Network Oriented Call Control and Traffic Management in B-ISDNs.* PhD thesis, University of Canterbury, New Zealand, 1994.

[45] H. Heffes and D. M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications*, 4(6):856–868, Sept. 1986.

[46] T. S. Heines. Buffer behaviour in computer communication systems. *IEEE Transactions on Communications*, 28(8):573–576, Aug. 1979.

[47] P. Henrici. *Applied and Computational Complex Analysis*, volume 1. John Wiley and Sons, 1974.

[48] J. Hsu. Buffer behaviour with Poisson arrival and geometric output processes. *IEEE Transactions on Communications*, 22:1940–1941, Dec. 1974.

[49] F. Hübner. *Discrete-Time Performance Analysis of Finite-Capacity Queueing Models for ATM Multiplexers.* PhD thesis, Würzburg, 1993.

[50] F. Hübner. Discrete-time analysis of the busy and idle period distributions of a finite-capacity ATM multiplexer with periodic input. *Performance Evaluation*, 21:23–36, 1994.

[51] D. Hughes, H. Bradlow, and G. Anido. Analysing ATM multiplexing performance. In *Proceedings 6th Australian Teletraffic Research Seminar*, pages 148–157, 1991.

[52] D. A. Hughes, G. Anido, and H. S. Bradlow. Queueing analysis of multiplexed bursty traffic with application to ATM switch performance. In *Proceedings 5th Australian Teletraffic Research Seminar*, pages 1–7, 1990.

[53] J. Hullett, T. Wong, G. Mercankosk, and Z. Budrikis. Packet discard strategy for congestion control of ATM networks. In *Proceedings Australian Telecommunication Networks and Applications Conference*, pages 633–638, Dec. 1994.

[54] J. L. Hullett, S. A. Ivandich, and K.-H. Chen. Architectures for ATM switches handling resources real-time traffic and statistically-multiplexed data traffic. In *Proceedings Australian Telecommunication Networks and Applications Conference*, pages 257–261, Dec. 1995.

[55] J. M. Hyman, A. A. Lazar, and G. Pacifici. Real-time scheduling with quality of service constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063, Sept. 1991.

[56] ITU-T Draft Recommendation E.72x. *Network Grade of Service Parameters and Target Values for B-ISDN.* International Telecommunication Union, 1994.

[57] ITU-T Draft Recommendation E.73x. *Methods for Traffic Control in B-ISDN.* International Telecommunication Union, 1994.

[58] ITU-T Recommendation G.764. *Voice Packetization — Packetized Voice Protocols.* International Telecommunication Union, 1990.

[59] ITU-T Recommendation I.121. *Broadband Aspects of ISDN.* International Telecommunication Union, 1993.

[60] ITU-T Recommendation I.211. *B-ISDN Service Aspects.* International Telecommunication Union, 1993.

[61] ITU-T Recommendation I.321. *B-ISDN Protocol Reference Model and its application.* International Telecommunication Union, 1993.

[62] ITU-T Recommendation I.362. *B-ISDN ATM Adaption Layer (AAL) Functional Description.* International Telecommuncation Union, 1993.

[63] ITU-T Recommendation I.363. *B-ISDN ATM Adaption Layer (AAL) Specifcation.* International Telecommuncation Union, 1993.

[64] ITU-T Recommendation I.371. *Traffic Control and Congestion Control in B-ISDN.* International Telecommunication Union, 1994.

[65] ITU-T Recommendation X.200. *Information Technology — Open Systems Interconnection — Basic Reference Model: The basic model.* International Telecommunication Union, 1994.

[66] M. Ivanovich, T. Neame, T. Theimer, and M. Zukerman. The AR(1) process as a model of mmpp traffic. In *Proceedings Australian Telecommunication Networks and Applications Conference*, pages 833–838, Dec. 1994.

[67] X. Jiang and J. S. Meditch. A high-speed integrated services ATM/STM switch. *Computer Networks and ISDN Systems*, 26:459–477, 1993.

[68] C. G. Kang and H. H. Tan. Queueing analysis of explicit priority assignment partial buffer sharing schemes for ATM networks. In *Proceedings IEEE INFOCOM'93*, pages 810–819, Mar. 1993.

[69] M. J. Karol, M. G. Hluchyj, and S. P. Morgan. Input versus output queueing on a space-division packet switch. *IEEE Transactions on Communications*, 35(12):1347–1356, Dec. 1987.

[70] F. J. Kaudel and M. E. Beshai. Performance of an efficient discipline for hybrid STM–ATM switching and transport. In *Proceedings ITC-14*, pages 1099–1108, June 1994.

[71] L. Kleinrock. *Queueing Systems, Volume II: Computer Applications.* John Wiley and Sons, 1976.

[72] H. Kröner, G. Hébuterne, P. Boyer, and A. Gravey. Priority management in ATM switching nodes. *IEEE Journal on Selected Areas in Communications*, 9(3):418–427, Apr. 1991.

[73] J. F. Kurose and H. T. Mouftah. Computer-aided modeling, analysis, and design of communication networks. *IEEE Journal on Selected Areas in Communications*, 6(1):130–145, Jan. 1988.

[74] R. H. Kwong and A. Leon-Garcia. Performance analysis of an integrated hybrid-switched multiplex structure. *Performance Evaluation*, 4:81–91, 1984.

[75] C. W. Lee and M. S. Andersland. Minimizing consecutive packet loss in real-time ATM sessions. In *Proceedings IEEE GLOBECOM*, pages 935–940, Nov. 1994.

[76] H. W. Lee and J. W. Mark. ATM network traffic characterization using two types of on-off sources. In *Proceedings IEEE INFOCOM'93*, pages 152–159, Mar. 1993.

[77] J. Lee and B. Lee. Performance analysis of ATM cell multiplexer with MMPP input. *IEICE Transactions on Communication*, E75B(8):709–714, Aug. 1992.

[78] K. K. Leung. Cyclic-service systems with probabilitically-limited service. *IEEE Journal on Selected Areas in Communications*, 9(2):185–193, Feb. 1991.

[79] S.-Q. Li. A general solution technique for discrete queueing analysis of multimedia traffic on ATM. *IEEE Transactions on Communications*, 39(7):1115–1132, July 1991.

[80] S.-Q. Li and J. W. Mark. Performance of voice/data integration on a tdm system. *IEEE Transactions on Communications*, 33(12):1265–1273, Dec. 1985.

[81] S.-Q. Li and J. W. Mark. Performance trade-offs in an integrated voice/data services TDM system. *Performance Evaluation*, 8:51–64, 1988.

[82] S.-Q. Li and J. W. Mark. Simulation study of a network of voice/data integrated TDMs. *IEEE Transactions on Communications*, 36(1):126–132, Jan. 1988.

[83] S.-Q. Li and H.-D. Sheng. Discrete queueing analysis of multi-media traffic with diversity of correlation and burstiness properties. *Proceedings IEEE INFOCOM'91*, pages 368–381, 1991.

[84] K.-Q. Liao and L. G. Mason. A discrete-time single server queue with a two-level modulated input and its applications. In *Proceedings IEEE GLOBECOM*, pages 913–918, 1989.

[85] N. M. Marafih, R. L. Pickholtz, and Y.-Q. Zhang. New approximation for analysing the multiplexing of bursty sources in ATM networks. In *Proceedings IEEE GLOBECOM*, pages 1085–1089, Dec. 1994.

[86] K. Mase and S. Shioda. Real-time network management for ATM networks. In *Proceedings ITC-13*, pages 129–136, 1991.

[87] W. Matragi, C. Bisdikian, and K. Sohraby. On the jitter and delay analysis in ATM multiplexer. In *Proceedings ICC'94*, pages 738–744, May 1994.

[88] J. Meyer, S. Montagna, and R. Paglino. Dimensioning of an ATM switch with shared buffer and threshold priority. *Computer Networks and ISDN Systems*, 26:95–108, 1993.

[89] I. Mitrani and R. Chakka. Spectral expansion solution for a class of mrkov models: Application and comparison with the matrix-geometric method. *Performance Evaluation*, 23:241–260, 1995.

[90] N. M. Mitrou and D. E. Pendarakis. Cell-level statistical multiplexing in ATM networks: Analysis, dimensioning, and call-acceptance control w. r. t. QoS critera. In *Proceedings ITC-13*, pages 7–12, 1991.

[91] Y. Miyao. Bandwidth allocation in ATM networks that guarantee multiple QOS requirements. In *Proceedings ICC'93*, pages 1398–1403, May 1993.

[92] R. Nagarajan, J. F. Kurose, and D. Towsley. Approximation techniques for computing packet loss in finite-buffered voice multiplexers. *IEEE Journal on Selected Areas in Communications*, 9(3):368–377, Apr. 1991.

[93] L. Nederlof, K. Struyve, C. O'Shea, H. Misser, Y. Du, and B. Tamayo. End-to-end survivable broadband networks. *IEEE Communications Magazine*, 33(9):63–70, Sept. 1995.

[94] M. F. Neuts. Moment formulas for the Markov renewal branching process. *Advances in Applied Probability*, 8:690–711, 1976.

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

[96] M. F. Neuts. The caudal characteristic curve of queues. *Advances in Applied Probability*, 18:221–254, 1986.

[97] M. F. Neuts. *Structured Stochastic Matrices of the M/G/1 Type and Their Applications.* Marcel Dekker Inc., 1989.

[98] M. F. Neuts. On Viterbi's formula for the mean delay in a queue of data packets. *Stochastic Models*, 6(1):87–98, 1990.

[99] G. F. Newell. *Applications of Queueing Theory.* Chapman and Hall, 2nd edition, 1982.

[100] P. Newman. ATM technology for corporate networks. *IEEE Communications Magazine*, 30(4):90–101, Apr. 1992.

[101] C. Y. Ngo and V. O. K. Li. Poisson approximation of input traffic sources in asynchronous transfer mode (ATM) networks. In *Proceedings IEEE INFOCOM'94*, pages 1046–1053, June 1994.

[102] Y. Ohba, M. Murata, and H. Miyahara. Analysis of interdeparture processes for bursty traffic in ATM networks. *IEEE Journal on Selected Areas in Communications*, 9(3):468–476, Apr. 1991.

[103] C. Ohta, H. Tode, M. Yamamoto, H. Okada, and Y. Tezuka. Peak rate regulation scheme for ATM networks and its performance. In *Proceedings IEEE INFOCOM'93*, pages 680–689, Mar. 1993.

[104] S. P. Parekh. Quick simulation of stationary tail probabilities at a packet switch. In *Proceedings ITC-14*, pages 887–896, June 1994.

[105] A. Pattavina. Nonblocking architectures for ATM switching. *IEEE Communications Magazine*, 31(2):38–48, Feb. 1993.

[106] J. Pieloor. Analysis of an ATM output buffer for a switch carrying both CBR and VBR traffic. In *Proceedings Australian Telecommunication Networks and Applications Conference*, pages 875–880, Dec. 1994.

[107] J. Pieloor and D. Lewis. Queueing behaviour of on-off binary sources. In *Proceedings Australian Telecommunication Networks and Applications Conference*, pages 443–447, Dec. 1995.

[108] J. Pieloor and D. J. H. Lewis. Variance of a discrete-time G/D/1 queue fed by two-state on-off sources. *Electronics Letters*, 32(1):19–20, Jan. 1996.

[109] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992.

[110] G. Ramamurthy and B. Sengupta. An analysis of a variable bit rate multiplexer using loss priorities. *Computer Networks and ISDN Systems*, 28:411–423, 1996.

[111] V. Ramaswami. Nonlinear matrix equations in applied probability — solution techniques and open problems. *SIAM Review*, 30(2):256–263, June 1988.

[112] V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Stochastic Models*, 4(1):183–188, 1988.

[113] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. *IEEE Journal on Selected Areas in Communications*, 13(4):633–641, May 1995.

[114] C. Rosenberg and A. Le Bon. Performance models for hybrid broadband networks. *Computer Networks and ISDN Systems*, 25:1155–1163, 1993.

[115] K. Rothermel. Priority mechanisms in ATM networks. In *Proceedings IEEE GLOBECOM*, pages 847–851, Dec. 1990.

[116] H. Schellhaas. On Ramaswami's algorithm for the computation of the steady state vector in Markov chains of M/G/1 type. *Stochastic Models*, 6(3):541–550, 1990.

[117] M. Schwartz. *Computer Communication: Network Design and Analysis.* Prentice Hall, 1977.

[118] L. D. Servi and D. D. Yao. Stochastic bounds for queueing systems with limited service schedules. *Performance Evaluation*, 9:247–261, 1989.

[119] R. Slosiar. Busy and idle periods at an ATM multiplexer output resulting from the superposition of homogeneous on/off sources. In *Proceedings ITC-14*, pages 431–440, 1994.

[120] K. Sohraby. On the asymptotic behaviour of heterogeneous statistical multiplexer with applications. In *Proceedings IEEE INFOCOM'92*, pages 839–847, May 1992.

[121] K. Sohraby. On the theory of general on-off sources with applications in high speed networks. In *Proceedings IEEE INFOCOM'93*, pages 401–410, Mar. 1993.

[122] D. Solow. *Linear Programming: An Introduction to Finite Improvement Algorithms.* Elsevier Science Publishers, 1984.

[123] K. Sriram, P. K. Varshney, and J. G. Shanthikumar. Discrete-time analysis of integrated voice/data multiplexers with and without speech activity detectors. *IEEE Journal on Selected Areas in Communications*, 1(6):1124–1132, Dec. 1983.

[124] I. Stavrakakis. Efficient modeling of merging and splitting processes in large networking structures. *IEEE Journal on Selected Areas in Communications*, 9(8):1336–1347, Oct. 1991.

[125] B. Steyaert, H. Bruneel, and Y. Xiong. A general relationship between buffer occupancy and delay in discrete-time multiserver queueing models, applicable in ATM networks. In *Proceedings IEEE INFOCOM'93*, pages 1250–1258, Mar. 1993.

[126] B. Steyaert and Y. Xiong. Analysis of a discrete-time queue with general three-state Markovian traffic sources. *Performance Evaluation*, 24:277–294, 1996.

[127] S. Suri, D. Tipper, and G. Meempat. A comparative evaluation of space priority strategies in ATM networks. In *Proceedings IEEE INFOCOM'94*, pages 516–523, June 1994.

[128] H. Takagi. *Analysis of Polling Systems*. MIT Press, Cambridge, MA., 1986.

[129] Y. Takagi, S. Hino, and T. Takahashi. Priority assignment control of ATM line buffers with multiple QOS classes. *IEEE Journal on Selected Areas in Communications*, 9(7):1078–1092, Sept. 1991.

[130] T. Takine, T. Suda, and T. Hasegawa. Cell loss and output process analysis of a finite-buffer discrete-time ATM queueing system with correlated arrivals. In *Proceedings IEEE INFOCOM'93*, pages 1259–1269, Mar. 1993.

[131] T. H. Theimer. Modeling and dimensioning buffers in multistage ATM switch fabrics. In *Proceedings Australian Broadband Switching and Services Symposium*, pages 190–197, July 1993.

[132] P. Tran-Gia. Analysis of polling systems with general input process and finite capacity. *IEEE Transactions on Communications*, 40(2):337–344, Feb. 1992.

[133] A. M. Viterbi. Approximate analysis of time-synchronous packet networks. *IEEE Journal on Selected Areas in Communications*, 4(6):879–890, Sept. 1986.

[134] J. L. Wang, J. P. Zhou, C. Wang, and Y. H. Fan. Interdeparture processes of traffic from ATM networks. In *Proceedings IEEE INFOCOM'93*, pages 1337–1341, Mar. 1993.

[135] S. S. Wang and J. A. Silvester. A fast performance model for real-time multimedia communication. In H. G. Perros and Y. Viniotis, editors, *High Speed Networks and Their Performance*, pages 199–216. Elsevier Science Publishers, 1993.

[136] S. S. Wang and J. A. Silvester. An approximate model for performance evaluation of real-time multimedia communication systems. *Performance Evaluation*, 22:239–256, 1995.

[137] J. E. Wieselthier and A. Ephremides. A movable-boundary channel-access scheme for integrated voice/data networks. In *Proceedings IEEE INFOCOM'91*, pages 721–731, Apr. 1991.

[138] S. Wittevrongel and H. Bruneel. Queue length and delay for statistical multiplexers with variable length messages. In *Proceedings IEEE GLOBECOM*, pages 1080–1084, Nov. 1994.

[139] S. Wolfram. *Mathematica: A system for Doing Mathematics by Computer*. Addison Wesley, 2nd edition, 1991.

[140] M. E. Woodward. Burstiness of interrupted bernoulli process. *Electronics Letters*, 30(18):1466–1467, Sept. 1994.

[141] Y. Xiong and H. Bruneel. A simple approach to obtain tight upper bounds for the asymptotic queueing behaviour of statistical multiplexers with heterogeneous traffic. *Performance Evaluation*, 22:159–173, 1995.

[142] Y. Xiong, B. Steyaert, and H. Bruneel. An ATM statistical multiplexer with on/off sources and spacing: Numerical and analytical performance studies. *Performance Evaluation*, 21:37–58, 1994.

[143] F. Yegenoglu and B. Jabbari. Performance evaluation of MMPP/D/1/K queues for aggregate ATM traffic models. In *Proceedings IEEE INFOCOM'93*, pages 1314–1319, Mar. 1993.

[144] J. Zhang. Performance study of Markov modulated fluid flow models with priority traffic. In *Proceedings IEEE INFOCOM'93*, pages 10–17, Mar. 1993.

[145] Z. Zhang and A. S. Acampora. Equivalent bandwidth for heterogeneous sources in ATM networks. In *Proceedings ICC'94*, pages 1025–1031, May 1994.

[146] M. Zukerman and I. Rubin. Queue size and delay analysis for a communication system subject to traffic activity mode changes. *IEEE Transactions on Communications*, 34(6):622–628, June 1986.