PERFORMANCE ANALYSIS OF GENERIC DISCRETE-TIME QUEUES AND APPLICATIONS TO TELECOMMUNICATION NETWORKS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Egemen Değer Kavak
September 2001

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Nail Akar(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Ezhan Karaşan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Tuğrul Dayar

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

PERFORMANCE ANALYSIS OF GENERIC DISCRETE-TIME QUEUES AND APPLICATIONS TO TELECOMMUNICATION NETWORKS

Egemen Değer Kavak
M.S. in Electrical and Electronics Engineering
Supervisor: Assist. Prof. Dr. Nail Akar
September 2001

Discrete-time queues frequently arise in the performance analysis of wireless and high-speed data networks such as ATM (Asynchronous Transfer Mode). In this thesis, we study two queueing analysis problems, namely, the generic discrete-time infinite queue and the generalized QBD (Quasi-Birth-Death) process, and we propose an algorithmic solution technique for each. The solution we propose for the generic discrete-time infinite queue is a novel technique that simultaneously employs the "generating function method" and the computation of "generalized invariant subspaces" via "matrix-sign function" iterations. The generality of the discrete-time infinite queue allows us to analyze a wide variety of queueing systems some of which cannot be carried out using the traditional matrix-analytical methods pioneered by Neuts. This approach also saves us from the burden of forming or representing the block matrices that constitute the transition matrix in the matrix-analytical methods. Besides the algorithm, we also present some network performance examples from recent literature for which we show how one can easily obtain the inputs to our proposed algorithm from the corresponding. The second algorithmic solution we propose for the generalized QBD process also makes use of generalized invariant subspaces. We show that the solution vectors for these two problems can be expressed via simple modified matrix-geometric forms. Each algorithmic solution technique is simple to implement, they are based on proven linear algebra techniques, and are robust and stable.

Keywords: Performance Analysis, Wireless Networks, ATM, Discrete-Time Queues, Generalized Invariant Subspaces, Matrix-Sign Function

ÖZET

GENEL AYRIK ZAMAN KUYRUKLARININ PERFORMANS ANALİZİ VE TELEKOMÜNİKASYON AĞLARINA UYGULAMALARI

Egemen Değer Kavak
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Assist. Prof. Dr. Nail Akar
Eylül 2001

Kablosuz ve yüksek hızlı veri ağlarında ayrık zaman kuyruklarına sıkça raslanır. Bu tezde, genel ayrık-zaman sonzuz kuyruğu ve genelleştirilmiş QBD ("Quasi-Birth-Death") problemleri üzerinde çalışıldı ve her iki problem için de birer algoritmik çözüm tekniği sunuldu. Genel ayrık-zaman sonsuz kuyruğu problemi için önerdiğimiz çözüm tekniği, "olasılık üretim fonksiyonu" yöntemini ve "matris-işareti fonksiyonu" özyinelemeleri yardımıyla "genelleştirilmiş değişimsiz altuzay" yöntemini beraber kullanan yeni bir tekniktir. Üzerinde çalıştığımız ayrık-zaman sonsuz kuyruğunun genelliği, geleneksel matris-analitik yöntemlerle çözülemeyen bazı kuyruk sistemlerinin de içinde bulunduğu birçok değişik kuyruk sistemini analiz edebilmemizi sağlar. Ayrıca matris-analitik yöntemlerde karşılaşılan, geçiş matrisini oluşturan blok matrislerinin oluşturulması problemi bu yaklaşımla ortadan kalkmaktadır. Bu tezde, algoritmanın yanında tekniğimizin genelliğini ve kolay kullanabilirliğini göstermek amacıyla yakın literatürden bazı örnek performans analizi problemleri verilecektir. İkinci problem olan genelleştirilmiş QBD süreci için önerdiğimiz çözüm tekniği yine genelleştirilmiş değişimsiz altuzay yöntemine dayanmaktadır. Her iki problemde bulunan çözüm vektörleri matris geometrik formdadır. Ayrıca bu çözüm teknikleri kolay uygulanabilir, ispatlanmış lineer cebir tekniklerini kullanan ve kararlı yöntemlerdir.

Anahtar Kelimeler: Performans Analizi, Kablosuz Ağlar, ATM, Ayrık-zaman Kuyrukları, Genelleştirilmiş Değişimsiz Altuzay, Matris-işareti Fonksiyonu

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor Assist. Prof. Dr. Nail Akar for his guidance, suggestions and invaluable encouragement throughout the development of this thesis.

I would like to thank Assist. Prof. Dr. Ezhan Karaşan and Assoc. Prof. Dr. Tuğrul Dayar for reading and commenting on this thesis.

Contents

List of Figures

List of Tables

To My Family and Friends . . .

Chapter 1

INTRODUCTION

In the design and development of telecommunication networks, performance evaluation plays a crucial role and its importance increases as the costs and the capabilities of the telecommunication systems increase. Performance evaluation problems arising in telecommunication networks can be divided into two basic categories: traffic modeling and queueing analysis. When a suitable traffic model is available, numerical queueing analysis provides various performance measures such as blocking probability, loss rate, delay, and delay jitter that are also known as QoS (Quality of Service) metrics.

Despite a wide variety of traditional traffic models and numerical queueing analysis techniques existing in the literature, the increase in traffic and the changes in the traffic characteristics have led to new performance evaluation problems that cannot easily be solved via existing methods. For example, the superposition of various types of multimedia sources can no longer be modeled by Poisson arrivals and exponential call holding times since such traffic may exhibit non-trivial higher-order statistics. Also, when the characterization is possible by a probabilistic model, this model may have a large dimensionality which may then require computationally efficient queueing analysis techniques. Another important point is the generality of the analysis techniques used for performance evaluation. Most of the existing techniques are problem specific and it is generally hard, if not impossible, to apply a method devised for one queueing system to another even when the latter is derived from the former. In other words, there is a certain need for a general queueing analysis technique that applies to a large class of queueing systems.

1.2 Discrete-Time Queueing Systems

In the performance evaluation of telecommunication networks using discrete-time models, the following discrete-time queue evolution equation is frequently encountered [1]:

$$Q_{n+1} = min[K, (Q_n + A_n - B_n)^+ + C_n]. \tag{1.1}$$

In this evolution equation, $(\cdot)^+$ stands for the $max(0, \cdot)$ operator, K stands for the queue storage capacity, the random variable Q_n stands for the queue length (e.g., number of packets, messages, or jobs depending on the application) at the beginning of time slot n and the random variable B_n represents the number of packets that can be served by the server during slot n. For the sake of generality, we assume two different types of packet arrivals from processes A and C. The process A represents the type of arrivals that can immediately be served while arrivals from process C must wait for the next slot to be served. Using this description, the random variables A_n and C_n stand for the number of packet arrivals from processes A and C, respectively, during slot n. We assume that each of the arrival and service processes is a Discrete-Batch Markovian Arrival Process (D-BMAP) [2] in which arrivals are allowed to occur in batches and the batch size distribution is dependent on the state of an underlying Markov chain. Such processes are known to be representable by a corresponding probability generating matrix (p.g.m.).

We refer to Eq. (1.1) as the evolution equation of a *generic* discrete-time queue since

- When $K \to \infty$, we have an infinite queue. Although infinite queues do not exist physically, they usually serve as an effective approximation for large buffer sizes. Also, they are used in the analysis of asymptotic queue length behaviour.

- Arrival and service processes are simultaneously allowed to be of D-BMAP type by which one can analyze the interaction of several complex phenomena like the variability of the arrival and server batch size over time, multi-server queues, server interruptions, priority queues, etc. via a unified framework.

- D-BMAP has a very rich structure and although it is mainly used in modeling discrete Markovian processes, it can also be used (although with a

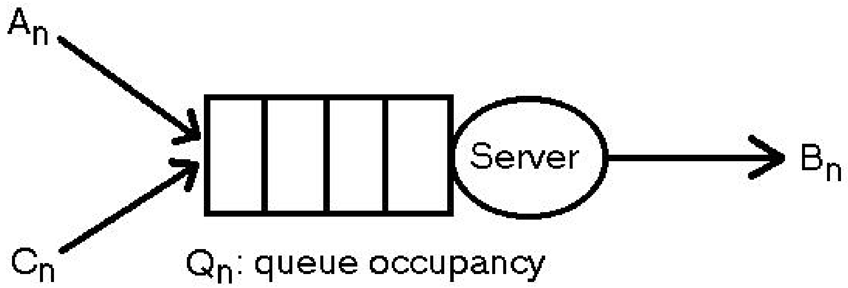

Figure 1.1: A generic discrete-time queueing system.

potentially large state space) to effectively approximate non-Markovian processes such as those with self-similar and heavy tailed distributions. We refer the reader to [3] for a related approach.

- Introduction of two different types of arrival in this evolution equation allows us to build a unified algorithmic solution for the steady-state probabilities, irrespective of when the new arrivals are served, during the current slot or the next slot.

In this thesis, queueing analysis refers to the computation of the steady-state probabilities of Eq. (1.1)

$$q_k^{level} = lim_{n \to \infty} Prob\{Q_n = k\},$$

rather than the corresponding transient behaviour. We assume such probabilities exist. This can be ensured by checking that the underlying discrete-parameter Markov chain is irreducible, positive-recurrent, and aperiodic. The steady-state performance measures like losses or delays can then easily be derived from the stationary probabilities.

There are basically three approaches to obtain stationary distributions, namely the *transform* approach, the *matrix analytical* approach, and the *generalized invariant subspace* approach. In the transform approach, first, an expression for the generating function of the state probabilities is obtained in terms of unknown coefficients, then the zeros of a certain polynomial matrix within the unit disk are determined. These zeros are then used to obtain linearly independent equations, solutions of which give the desired unknown coefficients. Finally, with inverse transformation, stationary probabilities can be determined. Since accurate computation of the zeros inside the unit disk is numerically difficult, this method is not considered to be much appealing in practice [4].

The second method, which is the matrix analytical approach, is pioneered by Neuts for the M/G/1 and G/M/1 type Markov chains [4], [5]. The matrix analytical method basically depends on iterative algorithms with linear convergence rates to find the minimal nonnegative solution of certain nonlinear matrix equations. Then the stationary probabilities are obtained using recursive computations. Although this method has better numerical stability compared to the transform approach, it may be impractical for problems of large dimensionality because of its low convergence rates. We also note that if the Markov chain under study is of type M/G/1 and G/M/1 simultaneously, or equivalently it is a QBD (Quasi-Birth-Death), then there are iterative algorithms to solve such processes with quadratic convergence rates [6].

Another approach is the *generalized invariant subspace* approach. This approach is very different from the two approaches briefly discussed above. In this approach, the problem of obtaining the stationary queue occupancy is considered as a problem of determining the output of a linear, discrete-time system, the initial conditions of which are to satisfy certain constraints. In particular, the stationary probabilities take over the role of the output of the dynamic system. Using the queue evolution equation, one obtains a generalized state-space representation for the described dynamical system [7]

$$z_{k+1}E = z_kA, \; q_k = z_kC,$$

where $\{q_k : k \geq 0\}$ is the output, E, A, and C are matrices of suitable size, z_k is called the descriptor of the system, and z_0 is to satisfy certain linear constraints. Finding z_0 requires the computation of certain *generalized invariant subspaces* which can be done by iterative algorithms such as the matrix-sign function iterations with quadratic convergence rates. However, this approach has only been applied to finite/infinite M/G/1 type Markov Chains and the solution technique proposed in [7] does not address all queueing problems which fall under Eq. (1.1). One main goal of this thesis is to extend the results of [7] to a wider variety of queueing analysis problems dictated by the queueing equation (1.1).

An important property of the invariant subspace approach is that it does not assume any particular Markov chain type such as M/G/1 or G/M/1. It can be built on top of the generic queueing system described by the evolution equation (1.1).

In this thesis, we study two general queueing analysis problems, namely

- Generic Discrete-time Infinite Queue,
- Generalized Discrete-time QBD Process,

and we propose a novel algorithmic solution for each. Next, we describe these two problems and outline our proposed solution techniques and then relate them to the above-mentioned three approaches.

1.3 Generic Discrete-Time Infinite Queue

In this study, we concentrate on the infinite queue capacity case of Eq. (1.1), i.e., $K \to \infty$. Then the evolution equation (1.1) reduces to

$$Q_{n+1} = max(0, Q_n + A_n - B_n) + C_n. \tag{1.2}$$

We propose an algorithmic solution for the generic discrete-time infinite queue through a novel technique that simultaneously employs the "generating function method" and the computation of the "generalized invariant subspace" [8]. This approach saves us from the burden of forming or representing block matrices that constitute the transition matrix in the "structured Markov chain"-based methods. The algorithmic solution is simple to implement, it is based on proven linear algebra techniques, and is robust and stable.

A variety of network performance analysis problems can be studied using the unified queueing model in (1.2) including several wireless multiple access protocols from recent literature [9], [10]. A similar model is used for the performance analysis of PRMA++ in [11]. A considerable amount of literature exists on discrete-time models used in the performance analysis of high-speed networks such as ATM [2], [12], [13]. Priority queues can also be studied using similar discrete-time models [14].

Finding the steady-state probabilities of the queue length in (1.2) is generally reduced to the computation of the invariant vector of an infinite-state discrete-time Markov chain with repeating rows [15], [16]. Such a chain turns out to be a variant of the following canonical probability transition matrix in block-partitioned form:

$$Q = \begin{bmatrix} D_0 & D_1 & D_2 & \cdots & & & \\ E_{-1} & P_0 & P_1 & P_2 & \cdots & & \\ E_{-2} & P_{-1} & P_0 & P_1 & P_2 & \cdots & \\ E_{-3} & P_{-2} & P_{-1} & P_0 & P_1 & P_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{1.3}$$

The first row block and column block of Q constitute the two boundaries of the Markov chain and the remaining submatrix is a block-structured Toeplitz matrix. When $P_i = 0$, $E_i = 0$ for $i < -1$ and $P_{-1} = E_{-1}$, we have a canonical M/G/1 type Markov chain and this chain can be solved using the iterative methods of Neuts [5]. An alternative solution technique using invariant subspaces for M/G/1 type Markov chains is proposed in [17]. When $P_i = 0$, $D_i = 0$ for $i > 1$ and $P_1 = D_1$, we have a canonical G/M/1 type Markov chain for which powerful algorithms exist [4], [18], [17]. Using similar techniques, it is also possible to address non-canonical chains for these two subcases; see [4] and [5] for details. For the general Markov chain in (1.3) with repeating rows, few results exist. In [15], a state reduction method that uses truncation on matrix blocks is proposed whereas [16] extends the generalized invariant subspace approach proposed in [7] to solve for the steady-state vector of the structured Markov chain with repeating rows.

The solution technique we propose is based on a different method, as opposed to the methods that use the structured Markov chain framework. To classify, this solution technique is similar to the transform approach since it uses generating functions, but we avoid the determination of the zeroes of a certain matrix polynomial inside the unit circle which is the most significant drawback of the transform approach. At the same time, our solution technique uses principles similar to those used in the invariant subspace approach. Our contributions can be summarized as follows:

- We bypass the step of constructing the structured Markov chain and instead we directly employ the generating function approach on the probability generating matrices corresponding to the arrival and service processes. This saves us from the burden of compactly representing the boundaries and the block Toeplitz submatrix that would arise in the structured Markov chain method. Our experience with the numerical analysis of queues has shown us that the difficulty, even for a sophisticated user, is to form or represent the blocks or submatrices of the associated Markov chain in their network performance applications. In other words, the network performance problem does not immediately lend itself to a structured Markov chain and forming the latter is generally cumbersome. This feature of our proposed approach will allow researchers and practitioners to use the algorithms we present with little additional effort if their network performance problem fits in the framework (1.2).

- Once a representation of the generating function of the steady-state queue length is determined, we apply the generalized state-space techniques to obtain a modified matrix-geometric representation of the steady-state queue length probabilities. These techniques are already proven to be robust and efficient [7].

- We also present several wireless and ATM network performance problems from recent literature and show how these problems fit naturally in the queueing model (1.2). We also show how to easily derive the probability generating matrices of the arrival and service processes for these network performance problems.

1.4 Generalized Discrete-Time QBD Process

As the second part of this thesis, we propose another algorithmic approach to compute the steady-state probability vector of the generalized discrete-time QBD process. Before describing this problem, let us first examine the discrete infinite QBD Markov chain with state space $(i, j) : 0 \leq i,\ 1 \leq j \leq m$, and which has an irreducible transition probability matrix of the canonical block tridiagonal form

$$P = \begin{bmatrix} B_0 & A_0 & & & \\ B_1 & A_1 & A_0 & & \\ & A_2 & A_1 & \ddots & \\ & & A_2 & \ddots & \\ & & & \ddots & \end{bmatrix}, \tag{1.4}$$

with B_0, B_1, A_0, A_1, and A_2 all being $m \times m$ nonnegative matrices. Note that P is of infinite size. The stationary probability vector x of the infinite QBD process is the unique solution of the equations

$$x = xP, \; xe = 1.$$

where e is an infinite column vector of ones. Let us partition the stationary vector x as

$$x = \begin{bmatrix} x_0 & x_1 & x_2 & \cdots \end{bmatrix},$$

where x_k is of size $1 \times m$.

The infinite QBD with the transition matrix given in (1.4) has been encountered in numerous applications involving infinite queues [19], [20], [4], [21]. In this case, the stationary solution has a matrix-geometric form [4], i.e.,

$$x_k = x_0 R_1^k, \; k \geq 0, \tag{1.5}$$

where the matrix geometric rate matrix R_1 is the unique minimal nonnegative solution of the quadratic matrix equation

$$R = A_0 + RA_1 + R^2 A_2.$$

Once R_1 is known, the boundary vector x_0 in (1.5) can be computed by solving [4]

$$x_0 = x_0 B[R_1], x_0(I - R_1)^{-1} e = 1,$$

where

$$B[R_1] \triangleq B_0 + R_1 B_1.$$

Also in the case of a finite QBD process with transition probability matrix of the form

$$P = \begin{bmatrix} B_0 & A_0 & & & & \\ B_1 & A_1 & A_0 & & & \\ & A_2 & A_1 & \ddots & & \\ & & A_2 & \ddots & A_0 & \\ & & & \ddots & A_1 & C_0 \\ & & & & A_2 & C_1 \end{bmatrix}, \tag{1.6}$$

the stationary solution can be expressed in a variety of matrix-geometric forms [22], [23], [24]. In [25], a constructive proof for the mixed matrix-geometric form

$$x_k = v_1 R_1^k + v_2 R_2^{K-k},\ 0 \le k \le K, \tag{1.7}$$

of the solution vector for the finite QBD is given. In (1.7), R_1 is as defined before, and R_2 is the unique minimal nonnegative solution of the quadratic matrix equation

$$R = A_2 + RA_1 + R^2 A_0,$$

and v_1 and v_2 are row vectors of length m. In [25], R_1 and R_2 are calculated using matrix-sign function iterations on matrices of size $2m \times 2m$. Once R_1 and R_2 are known, v_1 and v_2 follow easily as the solution of a linear matrix equation of also size $2m$.

Let us now consider the more general transition matrix of the form

$$P = \begin{bmatrix} D_0 & P_1 & \cdots & P_\eta & & & & & \\ E_{-1} & P_0 & \cdots & P_{\eta-1} & P_\eta & & & & \\ E_{-2} & P_{-1} & \cdots & P_{\eta-2} & P_{\eta-1} & P_\eta & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & \\ E_{-\gamma} & P_{-\gamma+1} & \cdots & P_{\eta-\gamma} & P_{\eta-\gamma+1} & P_{\eta-\gamma+2} & \cdots & P_\eta & \\ & P_{-\gamma} & \cdots & P_{\eta-\gamma-1} & P_{\eta-\gamma} & P_{\eta-\gamma+1} & \cdots & \ddots & P_\eta \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{1.8}$$

We call this system a *generalized infinite QBD*. One can employ QBD solution techniques to solve for the stationary probability of this system by properly repartitioning the block matrices. Consider the case when $\eta = 3$ and $\gamma = 1$, then the form of the transition matrix is

$$P = \begin{bmatrix} E_0 & P_1 & P_2 & P_3 & & & & \\ E_1 & P_0 & P_1 & P_2 & P_3 & & & \\ & P_{-1} & P_0 & P_1 & P_2 & P_3 & & \\ & & P_{-1} & P_0 & P_1 & P_2 & P_3 & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{1.9}$$

With the following definitions:

$$B_0 \triangleq \begin{bmatrix} E_0 & P_1 & P_2 \\ E_1 & P_0 & P_1 \\ 0 & P_{-1} & P_0 \end{bmatrix}, B_1 \triangleq \begin{bmatrix} 0 & 0 & P_{-1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$A_2 \triangleq \begin{bmatrix} 0 & 0 & P_{-1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, A_1 \triangleq \begin{bmatrix} P_0 & P_1 & P_2 \\ P_{-1} & P_0 & P_1 \\ 0 & P_{-1} & P_0 \end{bmatrix}, A_2 \triangleq \begin{bmatrix} P_3 & 0 & 0 \\ P_2 & P_3 & 0 \\ P_1 & P_2 & P_3 \end{bmatrix},$$

where 0 stands for a zero block matrix of order m, the system in (1.9) is expressed as a QBD of the form (1.4). In other words, generalized infinite QBD systems can be solved by QBD solution techniques. However, we propose a solution technique to solve for the infinite/finite generalized QBD process with a lower computational complexity than that is available using traditional QBD solution techniques. In the solution of generalized QBD systems via QBD techniques, the matrix-sign iterations are carried out on matrices of size $2max(\eta, \gamma)m$, however in the solution technique we propose, the size of the matrices are $(\eta + \gamma)m$. Note that when $\eta = \gamma$, the complexities for both techniques are the same.

Our solution technique is based on generalized invariant subspaces. The solution vector for level k, x_k, $k \geq 1$, is then shown to be in the modified matrix geometric form $x_{k+1} = gF^kY,\ k \geq 0$ for the infinite case; whereas, the solution vector can easily be obtained from a vector z_k which takes the modified mixed matrix geometric form $z_k = g_1F_1^kY_1 + g_2F_2^{K-k}Y_2,\ 0 \leq k \leq R$, for the case of a finite number of levels, where R is the number of repeating columns or rows in the transition matrix. The matrix parameters in the above two expressions can be obtained by decomposing the generalized system into forward and backward systems, or equivalently, by finding bases for certain generalized invariant subspaces of a regular pencil $\lambda E - A$. We note that the computation of such bases can efficiently be carried out using matrix sign function iterations with quadratic convergence rates. The simplicity of the matrix-geometric form of the solution vector allows one to obtain very easily related performance measures of interest (e.g., overflow probabilities, moments of level distribution) which is a significant advantage over conventional matrix analytical methods that use recursive computations to obtain the steady-state probabilities.

1.5 Thesis Outline

In Chapter 2, we first give some preliminaries and notations. Then we present the mathematical model and the solution of the generic discrete-time infinite queue problem. After summarizing the algorithm of the solution technique proposed, we address several network performance examples from existing literature and show how they fit in our algorithmic solution framework. A simple numerical example is provided in the last section.

In Chapter 3, we present the problem formulation and the solution algorithm proposed for the finite generalized QBD problem. Then we briefly explain the solution algorithm for the infinite capacity queue case, which can be considered as a subcase of the finite generalized QBD problem with some reductions. In the last section, we give a simple numerical example showing the accuracy and robustness of the solution technique proposed.

Conclusions and potential future research areas are then presented in Chapter 4.

Chapter 2

GENERIC DISCRETE-TIME INFINITE QUEUE

2.1 Preliminaries and Notation

First we define the probability generating function (p.g.f.) of a discrete random variable g as

$$g(z) \triangleq \sum_{i=-\infty}^{\infty} g_i z^i, \tag{2.1}$$

where

$$g_i \triangleq Pr(g = i).$$

We use the following notations throughout this chapter. A constant, polynomial or rational matrix is called *regular* if it is square and has a nonzero determinant, otherwise it is called *singular*. The degree of a polynomial $p(z)$ is shown with $deg(p)$. The field of real numbers is denoted by R. A subspace $\mathcal{S}$ is a subset of R^m that is closed under the operations of addition and scalar multiplication. $\mathcal{S} + \mathcal{T}$ and $\mathcal{S} \oplus \mathcal{T}$ are the sum and direct sum, respectively, of the subspaces $\mathcal{S}$ and $\mathcal{T}$. I and e denote the identity matrix and a column matrix of ones of appropriates sizes, respectively. When the size of the identity matrix needs to be emphasized, we use the notation I_b for a $b \times b$ identity matrix. Given an $n \times m$ matrix $A = \{a_{ij}\}$ and a $p \times q$ matrix B, their Kronecker product is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{bmatrix}, \tag{2.2}$$

and the size of $A \otimes B$ is $np \times mq$.

Now, we give some mathematical background on the generalized invariant subspaces and their numerical computations. Let A and E be two $n \times n$ matrices. The matrix pencil $\lambda E - A$ is a polynomial matrix in the indeterminate λ with degree one. A complex scalar λ and a nonzero vector x satisfying

$$Ax = \lambda Ex, \quad x \neq 0,$$

are called the generalized eigenvalue of the matrix pencil $\lambda E - A$ and the generalized eigenvector of the matrix pencil $\lambda E - A$ associated with λ, respectively. When $E = I$, we have the definition of ordinary eigenvalue of the matrix A. $\lambda(E, A)$ denotes the set of eigenvalues of the matrix pencil $\lambda E - A$. Any subspace $\mathcal{S}$ satisfying

$$\mathcal{T} = E\mathcal{S} + A\mathcal{S}, \; dim(\mathcal{S}) = dim(\mathcal{T})$$

is called a *generalized invariant subspace* (or a deflating subspace) of the pencil $\lambda E - A$.

Let $\mathcal{S}$ and $\mathcal{S}_c$ be two complementary deflating subspaces of the pencil $\lambda E - A$ such that $\mathcal{S} \oplus \mathcal{S}_c = R^n$. Then the subspaces $\mathcal{T}$ and $\mathcal{T}_c$ which are defined as

$$\mathcal{T} = E\mathcal{S} + A\mathcal{S}, \;\; \mathcal{T}_c = E\mathcal{S}_c + A\mathcal{S}_c, \tag{2.3}$$

are also complementary subspaces of each other as shown in [26]. If the subspaces are represented by matrices such that $\mathcal{S} = Im\ S$, $\mathcal{S}_c = Im\ S_c$, $\mathcal{T} = Im\ T$ and $\mathcal{T}_c = Im\ T_c$, then there exists a decomposition such that

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}, \;\; U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \tag{2.4}$$

where

$$U = \begin{bmatrix} T & T_c \end{bmatrix}, \;\; V = \begin{bmatrix} S & S_c \end{bmatrix}. \tag{2.5}$$

If $\lambda(E_{11}, A_{11})$ ($\lambda(E_{22}, A_{22})$) lies in the closed right-half (open left-half) plane, then $\mathcal{S}$ ($\mathcal{S}_c$) is called the right (left) deflating subspace of the matrix-pencil $\lambda E - A$. When $\lambda(E_{11}, A_{11})$ ($\lambda(E_{22}, A_{22})$) lies outside (in) the open unit disk, then $\mathcal{S}$ ($\mathcal{S}_c$) is called the unstable (stable) deflating subspace of the matrix-pencil $\lambda E - A$. Once the stable and unstable deflating subspaces for the matrix-pencil $\lambda E - A$ are obtained, one can find the matrices U and V that satisfy the decomposition (2.4) such that $\lambda(E_{11}, A_{11})$ lies outside the unit disk and $\lambda(E_{22}, A_{22})$ lies in the unit disk. Different techniques exist in numerical linear algebra for the computation of deflating subspaces (left or right, stable or unstable) such as generalized matrix-sign function iterations, generalized Schur decomposition method, and inverse-free spectral divide and conquer method. Here we present their algorithmic approaches briefly.

2.1.1 Generalized Matrix-Sign Function Approach

For a definition of the matrix-sign function of a matrix, we refer the reader to [25]. The matrix-sign function is defined only for matrices that have no eigenvalue on the imaginary axis; the matrix-sign function of such a matrix M is denoted by $sgn(M)$. Given a regular matrix-pencil $\lambda L - M$ of size $n \times n$ with no generalized eigenvalues on the imaginary axis, two important properties of the matrix-sign function are given in [27] as

- $\mathcal{N}(L - Lsgn(L^{-1}M))$ = right deflating subspace of the matrix pencil $\lambda L - M$,
- $\mathcal{N}(L + Lsgn(L^{-1}M))$ = left deflating subspace of the matrix pencil $\lambda L - M$,

where $\mathcal{N}$ stands for nullspace. Note that the stable (unstable) deflating subspace of a matrix-pencil $\lambda E - A$ is equal to right (left) deflating subspace of a matrix-pencil $\lambda L - M$, where L and M are defined as

$$L \triangleq E + A, \quad M \triangleq A - E. \tag{2.6}$$

For a proof, refer to [28]. Then the stable and unstable deflating subspaces of the matrix-pencil $\lambda E - A$ can be found using this transformation and the properties of the matrix-sign function if the matrix-pencil $\lambda E - A$ have no generalized eigenvalue on the unit circle.

The generalized matrix-sign function iterations for the matrix-pencil $\lambda L - M$ are given as

$$\begin{aligned} Z_0 &= M, \\ Z_{k+1} &= \frac{1}{2c_k}\left(Z_k + c^2 L Z_k^{-1} L\right), \; c_k = \left(\frac{|det\ Z_k|}{|det\ L|}\right)^{\frac{1}{n}}. \end{aligned} \tag{2.7}$$

The stopping criterion we use is

$$||Z_{k+1} - Z_k|| < \epsilon \, ||Z_{k+1} - Z_k|| \, .$$

When $\lambda L - M$ have no eigenvalues on the imaginary axis, the iteration converges with quadratic rates to $Lsgn(L^{-1}M)$:

$$Z_\infty \triangleq \lim_{n\to\infty} Z_k = Lsgn(L^{-1}M).$$

Since the stable (unstable) deflating subspace of the matrix-pencil $\lambda E - A$ is equal to right (left) deflating subspace of the matrix-pencil $\lambda L - M$, one can find the matrix V using rank revealing QR decompositions. Let the rank-revealing QR decomposition of $\mathcal{N}(L - Z_\infty)$ and $\mathcal{N}(L + Z_\infty)$ be

$$\begin{aligned} \mathcal{N}(L - Z_\infty) &= Q_u R_u \Pi_u, \\ \mathcal{N}(L + Z_\infty) &= Q_s R_s \Pi_s. \end{aligned}$$

Next, define

$$\begin{aligned} V_1 &\triangleq \textit{leading } r_u \textit{ columns of } Q_u, \\ V_2 &\triangleq \textit{leading } r_s \textit{ columns of } Q_s, \end{aligned}$$

where $r_u = rank\,(R_u)$ and $r_s = rank\,(R_s)$. Note that the rank of R_u and R_s are the number of generalized eigenvalues outside and inside the unit disk, respectively. Then, the matrix V is found as

$$V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}.$$

The matrix U can be obtained from V, by the equations (2.3) and (2.5). We again use the rank-revealing QR decompositions of the matrices below

$$\begin{aligned} \begin{bmatrix} EV_1 & AV_1 \end{bmatrix} &= \hat{Q}_u \hat{R}_u \hat{\Pi}_u, \\ \begin{bmatrix} EV_2 & AV_2 \end{bmatrix} &= \hat{Q}_s \hat{R}_s \hat{\Pi}_s, \end{aligned}$$

and we define

$$\begin{aligned} U_1 &\triangleq \textit{leading } r_u \textit{ columns of } \hat{Q}_u, \\ U_2 &\triangleq \textit{leading } r_s \textit{ columns of } \hat{Q}_s, \end{aligned}$$

finally, U is set to

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}. \tag{2.8}$$

2.1.2 Inverse-Free Spectral Divide and Conquer Method

This method is based on [29] and requires that the matrix-pencil $\lambda E - A$ have no generalized eigenvalue on the unit circle. The iterations of the method are given as

$$A_0 = A,\ B_0 = E;$$

$$\begin{aligned} A_{j+1} &= Q_{12}^H A_j, \\ B_{j+1} &= Q_{22}^H B_j, \end{aligned}$$

where Q_{12} and Q_{22} come from the rank revealing QR decomposition:

$$\begin{bmatrix} B_j \\ -A_j \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R_j \\ 0 \end{bmatrix}.$$

The stopping criteria are

$$\|R_j - R_{j-1}\|_1 < \tau \|R_{j-1}\|_1,$$

or

$$j < maxit$$

whichever is first satisfied. The convergence parameters used in [29] are $\tau \approx n\epsilon$ (where ϵ is the machine precision and n is the size of the square matrices A and E) and $maxit = 60$ (maximum number of iterations). When the matrix-pencil $\lambda E - A$ have no generalized eigenvalue on the unit circle, the iterations converge to

$$\begin{aligned} A_\infty &\triangleq \lim_{j\to\infty} A_j, \\ B_\infty &\triangleq \lim_{j\to\infty} B_j. \end{aligned}$$

With these definitions, the unstable and the stable deflating subspaces of the matrix-pencil $\lambda E - A$ can be found by rank revealing QR decompositions. Let the rank rank-revealing QR decomposition of $(A_\infty + B_\infty)^{-1} A_\infty$ and $(A_\infty + B_\infty)^{-1} B_\infty$ be

$$\begin{aligned} (A_\infty + B_\infty)^{-1} A_\infty &= Q_u R_u \Pi_u, \\ (A_\infty + B_\infty)^{-1} B_\infty &= Q_s R_s \Pi_s, \end{aligned}$$

Then the matrix V is found from Q_u and Q_s, and the matrix U is obtained from the matrix V as described in the previous section. To make the algorithm inverse-free, consider the computation of the rank revealing QR decomposition of $C^{-1}D$. Let the rank-revealing QR decomposition of D be

$$D = Q_1 R_1 \Pi \tag{2.9}$$

and the RQ decomposition of $Q_1^H C$ be

$$Q_1^H C = R_2 Q_2. \tag{2.10}$$

From (2.9) and (2.10), we have

$$C^{-1}D = Q_2^H (R_2^{-1} R_1)\Pi. \tag{2.11}$$

Observe that (2.11) is a rank-revealing QR decomposition and the unitary matrix Q_2^H can be obtained without explicitly calculating the inverse of the matrix C. The rank of R_1 is also the rank of $C^{-1}D$. So, for the computation of the rank-revealing QR decomposition of $(A_\infty + B_\infty)^{-1} A_\infty$ and $(A_\infty + B_\infty)^{-1} B_\infty$, this inverse-free computation technique can be used.

2.1.3 Generalized Schur Decomposition Approach

Given the matrix pencil $\lambda E - A$, another approach to find the stable and the unstable deflating subspaces that give rise to the decomposition (2.4) is the generalized Schur decomposition approach [30]. Using the generalized Schur decomposition with ordering, one can find the two orthonormal matrices Θ and Ψ that satisfy the decomposition:

$$\Theta^T E \Psi = \begin{bmatrix} E_{11} & E_{12} \\ 0 & E_{22} \end{bmatrix}, \quad \Theta^T A \Psi = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \tag{2.12}$$

such that

1. A_{11} and A_{22} are upper triangular with non-negative diagonals,
2. E_{11} and E_{22} are upper triangular with either 1-by-1 or 2-by-2 blocks (corresponding to complex eigenvalues),
3. $\lambda(E_{11}, A_{11})$ lies outside the open unit disk,
4. $\lambda(E_{22}, A_{22})$ lies in the open unit disk.

Given the above decomposition, we next solve the generalized Sylvester equations:

$$A_{11}Y - XA_{22} = A_{12}, \quad \textit{and} \quad E_{11}Y - XE_{22} = E_{12},$$

for the two matrices X and Y. Finally defining

$$U \triangleq \Theta \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix}, \quad \textit{and} \quad V \triangleq \Psi \begin{bmatrix} I & -Y \\ 0 & I \end{bmatrix},$$

one obtains the decomposition (2.4), i.e., eliminates the upper-diagonal blocks E_{12} and A_{12} in (2.12).

2.2 Mathematical Model and Solution

Let us now consider the generic discrete-time infinite queue evolution equation

$$Q_{n+1} = max(Q_n + A_n - B_n, 0) + C_n \tag{2.13}$$

For the solution of (2.13), the probability generating function approach is used. Let us now focus on a simpler relationship among the following random variables:

$$t = max(x - y, 0) \tag{2.14}$$

where x any y are two positive integer valued random variables. Define:

$$s \triangleq x - y$$

then there is the following relationship between the random variables t and s:

$$Pr(t = i) = \begin{cases} 0, & i < 0 \\ Pr(s \le 0), & i = 0 \\ Pr(s = i), & i > 0 \end{cases} .$$

From the relationship shown above, the probability generating function of the random variable t can be calculated from the probability generating function of s as shown below:

$$\begin{aligned} t(z) &= \sum_{i=0}^{\infty} t_i z^i, \\ s(z) &= \sum_{i=-\infty}^{\infty} s_i z^i, \end{aligned}$$

where

$$Pr(t = i) = t_i = \begin{cases} 0, & i < 0 \\ \sum_{k=-\infty}^{0} s_k, & i = 0 \\ s_i, & i > 0 \end{cases} . \tag{2.15}$$

Throughout the thesis, this relation between the probability generating functions is shown by the $[\,]_s$ operator:

$$t(z) = [s(z)]_s \,,$$

As can be seen from the relation shown in (2.15), the operator $[\,]_s$ maps $s(z)$ to $t(z)$ by putting the weight of the negative part of $s(z)$ on $s(0)$ and leaving the positive part as it is.

One can show by using the definition of the probability generating function in (2.1) that

$$s(z) = x(z)y(z^{-1}).$$

Consequently $t(z)$ can be rewritten as:

$$t(z) = \left[x(z)y(z^{-1})\right]_s .$$

Theorem 1. *Given $x(z)$ and $y(z)$ as right-sided probability generating functions such that:*

$$y(z) = \frac{n(z)}{d(z)}, \tag{2.16}$$

$$deg(n) > deg(d),$$

then

$$\left[x(z)y(z^{-1})\right]_s = x(z)y(z^{-1}) - \frac{u(z^{-1})}{d(z^{-1})}, \tag{2.17}$$

where $u(z)$ is a polynomial of degree $deg(n)$ and $u(1) = 0$.

Proof. Both $x(z)$ and $y(z)$ can be written in the form of infinite series:

$$\begin{aligned} x(z) &= x_0 + x_1 z + x_2 z^2 + \cdots, \\ y(z) &= y_0 + y_1 z + y_2 z^2 + \cdots, \end{aligned}$$

From polynomial multiplication and definition of the $[\,]_s$ operator in (2.15), one can show that

$$\left[x(z)y(z^{-1})\right]_s = x(z)y(z^{-1}) - \sum_{k=0}^{\infty} x_k z^k \Big(\sum_{i=k+1}^{\infty} y_i z^{-i} \Big) + \Gamma,$$

where Γ is the constant

$$\Gamma = \sum_{k=0}^{\infty} x_k \sum_{i=k+1}^{\infty} y_i.$$

Since p.g.f of a random variable evaluated at $z = 1$ must be one, $x(1) = 1$ and $y(1) = 1$. It is easy to see that

$$\left[x(z)y(z^{-1})\right]_{s|_{z=1}} = 1. \tag{2.18}$$

If there is such a $u(z)$ satisfying (2.17), since $d(1) < \infty$, $u(1)$ must be zero. Now the existence of such a $u(z)$ will be shown. Observe that the terms in the parenthesis in (2.19) can be written in a different form:

$$\left[x(z)y(z^{-1})\right]_s = x(z)y(z^{-1}) - \sum_{k=0}^{\infty} x_k z^k \left(y(z^{-1}) - y_k(z^{-1})\right) + \Gamma, \tag{2.19}$$

where

$$y_k(z) \triangleq \sum_{i=0}^{k} y_i z^i \quad . \tag{2.20}$$

If rational form of $y(z^{-1})$ is used,

$$y(z^{-1}) - y_k(z^{-1}) = \frac{n(z^{-1}) - y_k(z^{-1})d(z^{-1})}{d(z^{-1})}. \tag{2.21}$$

From (2.16) it is known that

$$y(z)d(z) = n(z), \tag{2.22}$$

then it is easy to see that

$$y_k(z^{-1})d(z^{-1}) = n_k(z^{-1}) + p_k(z^{-1}), \tag{2.23}$$

where

$$n_k(z) \triangleq \sum_{i=0}^{k} n_i z^i, \tag{2.24}$$

$$p_k(z) \triangleq \sum_{i=k+1}^{k+deg(d)} p^{(k,i)} z^i. \tag{2.25}$$

Note that, $p^{(k,i)}$ is the i'th coefficient of the polynomial $p_k(z)$ whose value does not have significance for the proof. Also one can see that the definition of the polynomial $p_k(z)$ is different from the definitions of $y_k(z)$ and $n_k(z)$. As can be seen from the definition, the polynomial $p_k(z)$ is of degree $k + deg(d)$ whose first $k+1$ coefficients are 0. Using equation (2.23)

$$\begin{aligned} x_k z^k \left(\frac{n(z^{-1}) - y_k(z^{-1})d(z^{-1})}{d(z^{-1})} \right) &= x_k z^k \left(\frac{n(z^{-1}) - n_k(z^{-1}) - p_k(z^{-1})}{d(z^{-1})} \right), \\ &= x_k z^k \left(\frac{\sum_{i=k+1}^{deg(n)} n_i z^{-i} - \sum_{i=k+1}^{k+deg(d)} p^{(k,i)} z^{-i}}{d(z^{-1})} \right), \\ &= x_k \left(\frac{\sum_{i=1}^{deg(n)-k} n_{i+k} z^{-i} - \sum_{i=1}^{deg(d)} p^{(k,i+k)} z^{-i}}{d(z^{-1})} \right). \end{aligned}$$

Then, the equality (2.19) can be written as

$$\left[x(z)y(z^{-1})\right]_s = x(z)y(z^{-1}) - \frac{\overbrace{\sum_{k=0}^{\infty} x_k \left(\sum_{i=1}^{deg(n)-k} n_{i+k} z^{-i} - \sum_{i=1}^{deg(d)} p_i z^{-i} \right) + \Gamma d(z^{-1})}^{u(z^{-1})}}{d(z^{-1})}.$$

Knowing that $deg(n) > deg(d)$, $u(z)$ has a degree of at most $deg(n)$ which concludes the proof. □

The random variables in (2.13) are assumed to be Markov modulated for the generality of the solution. Let us consider the random variable A_n, which corresponds to the arrival process. Let S_n^A, $1 \le S_n^A \le a$, be the state of the modulating chain in slot n. Then the probability generating function of A_n can be represented by its *probability generating matrix* (p.g.m.):

$$A(z) = \{A_{ij}(z)\}_{a \times a},$$

where

$$A_{ij}(z) = E\left[z^{A_{n+1}} 1(S_{n+1}^A = j) | S_n^A = i\right].$$

The indicator function $1(E)$ is equal to one if the event E is true and equal to zero otherwise. Note that the size of $A(z)$ is $a \times a$. The probability generating matrix is assumed to be in rational form as in

$$A_{ij}(z) = \frac{n_A^{ij}(z)}{d_A^i(z)},$$

then one can obtain the right co-prime matrix fraction:

$$A(z) = N_A(z) D_A(z)^{-1}, \tag{2.26}$$

where

$$\begin{aligned} N_A(z) &= \{n_A^{ij}(z)\}_{a \times a}, \\ D_A(z) &= Diag\{d_A^1(z), d_A^2(z), \cdots, d_A^a(z)\}_{a \times a}. \end{aligned}$$

Let the probability generating matrices for the random variables A_n and B_n be $B(z)$ and $C(z)$, respectively. Then $B(z)$ and $C(z)$ have the same format with, respectively, $b \times b$ and $c \times c$ matrices. So there is a three dimensional state space and from the final value theorem, it is known that $deg(N_A) \ge deg(D_A)$, $deg(N_B) \ge deg(D_B)$ and $deg(N_C) \ge deg(D_C)$.

Define q_{jkl}^i as

$$q_{jkl}^i \triangleq lim_{n \to \infty} Prob\{Processes\ A, B,\ and\ C\ are\ in\ states\ j, k, l\ and\ Q_n = i\},$$

and $q(z)$ as

$$q_{jkl}(z) = \sum_{i=0}^{\infty} q_{jkl}^i.$$

Then

$$q_{jkl}(z) = \sum_{j'=1}^{a} \sum_{k'=1}^{b} \sum_{l'=1}^{c} \left[q_{j'k'l'}(z) \frac{n_A^{j'j}(z)}{d_A^j(z)} \frac{n_B^{k'k}(z)}{d_B^k(z)} \right]_s \frac{n_C^{l'l}(z)}{d_C^l(z)}.$$

Using Theorem 1, one can write the equation

$$q_{jkl}(z) = \sum_{j'=1}^{a} \sum_{k'=1}^{b} \sum_{l'=1}^{c} q_{j'k'l'}(z) \frac{n_A^{j'j}(z)}{d_A^j(z)} \frac{\overline{n}_B^{k'k}(z)}{\overline{d}_B^k(z)} \frac{n_C^{l'l}(z)}{d_C^l(z)} + \frac{w_{jkl}(z)}{\overline{d}_B^k(z)} \frac{n_C^{l'l}(z)}{d_C^l(z)}, \quad (2.27)$$

where

$$\overline{n}_B^{k'k}(z) \triangleq z^{deg(n_B^{k'k})} n_B^{k'k}(z^{-1}), \quad (2.28)$$
$$\overline{d}_B^{k}(z) \triangleq z^{deg(n_B^{k'k})} d_B^{k}(z^{-1}), \quad (2.29)$$

and $w_{jkl}(z)$ is a polynomial of z with degree $deg(n_B^{k'k})$. Then equation (2.27) can then be rewritten as:

$$q_{jkl}(z)d_A^j(z)\overline{d}_B^k(z)d_C^l(z) = \sum_{j'=1}^{a} \sum_{k'=1}^{b} \sum_{l'=1}^{c} Q_{j'k'l'}(z)n_A^{j'j}(z)\overline{n}_B^{k'k}(z)n_C^{l'l}(z) + w_{jkl}(z)n_C^{l'l}(z)d_A^j(z). \quad (2.30)$$

One can rewrite the equality (2.30) in matrix form as

$$\begin{aligned} q(z)\left(D_A(z) \otimes \overline{D}_B(z) \otimes D_C(z)\right) &= q(z)\left(N_A(z) \otimes \overline{N}_B(z) \otimes N_C(z)\right) + \\ &\quad w(z)\left(D_A(z) \otimes I \otimes N_C(z)\right), \end{aligned} \quad (2.31)$$

where

$$\overline{N}_B(z) \triangleq z^{deg(N_B)} N_B(z^{-1}), \quad (2.32)$$
$$\overline{D}_B(z) \triangleq z^{deg(N_B)} D_B(z^{-1}).$$

Above, I is the identity matrix of size $b \times b$, $\otimes$ stands for Kronecker product, and both $q(z)$ and $u(z)$ are row vectors of size $f \triangleq abc$ obtained through a suitable enumeration of q_{jkl}'s and w_{jkl}'s:

$$q(z) = \begin{bmatrix} q_{111}(z) & q_{112}(z) & \cdots & q_{11c}(z) & q_{121}(z) & \cdots q_{abc}(z) \end{bmatrix}, \quad (2.33)$$
$$w(z) = \begin{bmatrix} w_{111}(z) & w_{112}(z) & \cdots & w_{11c}(z) & w_{121}(z) & \cdots w_{abc}(z) \end{bmatrix}. \quad (2.34)$$

The equality in (2.31) can be simplified to:

$$q(z)G(z) = w(z)H(z), \quad (2.35)$$

where

$$\begin{aligned} G(z) &= \left(D_A(z) \otimes \overline{D}_B(z) \otimes D_C(z)\right) - \quad (2.36) \\ &\quad \left(N_A(z) \otimes \overline{N}_B(z) \otimes N_C(z)\right), \\ H(z) &= D_A(z) \otimes I \otimes N_C(z). \quad (2.37) \end{aligned}$$

The degree of $G(z)$ is $n \triangleq deg(N_A) + deg(N_B) + deg(N_C)$, the degree of $v(z)$ is $m \triangleq deg(N_B)$ and the degree of $H(z)$ is $deg(H) = deg(N_C) + deg(D_A) \leq n - m$.

Sizes of $G(z)$ and $H(z)$ are both $f \times f$. As the first constraint for the solution of (2.35), $w(1) = 0$ from Theorem 1, which leads to

$$w\bar{I} = [0 \ \ 0 \ \cdots \ 0], \tag{2.38}$$

where

$$w = [w_0 \ \ w_1 \ \cdots \ w_m], \quad \bar{I} = \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}, \tag{2.39}$$

and $\bar{I}$ is of size $(m+1)f \times f$. Consider now the identity (2.35). Also let

$$b(z) = b_0 + b_1 z + b_2 z^2 + \cdots + b_n z^n = w(z)H(z).$$

We first find a generalized state-space representation [31] for q_k as will be given in (2.40). This is to ensure that we can use advanced linear algebra techniques that employ vector-matrix operations. The proof is given in Appendix A, which is along the lines of [7]. This representation has $z_k, \ k \geq 0$ as the semistate [31].

Theorem 2. *The following generalized state-space equations form a representation of the identity (2.35):*

$$\begin{aligned} z_{k+1}E &= z_k A + B\delta(k), \quad k \geq 0, \\ q_k &= z_k C, \quad k \geq 0, \end{aligned} \tag{2.40}$$

where

$$z_k = [q_k \ \ q_{k+1} \ \ \cdots \ \ q_{k+n-1}]$$

$$E = \begin{bmatrix} I & & & \\ & \ddots & & \\ & & I & \\ & & & G_0 \end{bmatrix}, \tag{2.41}$$

$$A = \begin{bmatrix} & & & -G_n \\ I & & & -G_{n-1} \\ & \ddots & & \vdots \\ & & I & -G_1 \end{bmatrix}, \tag{2.42}$$

$$\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}, \tag{2.43}$$

$$B = [0 \ \ 0 \ \cdots \ 0 \ \ b_n], \tag{2.44}$$

$$C = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.45}$$

with the constraint

$$z_0\hat{G} = w\hat{H}, \tag{2.46}$$

where

$$\hat{G} = \begin{bmatrix} G_0 & G_1 & \cdots & G_{n-1} \\ & G_0 & \cdots & G_{n-2} \\ & & \ddots & \vdots \\ & & & G_0 \end{bmatrix}, \tag{2.47}$$

$$\hat{H} = \begin{bmatrix} H_0 & H_1 & \cdots & H_{n-m} & & \\ & H_0 & \cdots & H_{n-m-1} & H_{n-m} & \\ & & \ddots & & & \ddots \end{bmatrix}. \tag{2.48}$$

In these definitions, E and A are $nf \times nf$, z_k, $k \geq 0$, is $1 \times nf$, B is $1 \times nf$, C is $nf \times f$, $\hat{G}$ is $nf \times nf$ and $\hat{H}$ is $(m+1)f \times nf$.

Theorem 2 suggests that the state-space equations (2.40) are equivalent to the identity (2.35). Also note in this representation that

$$B = w\tilde{H}, \tag{2.49}$$

where

$$\tilde{H} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ H_{n-m} \end{bmatrix}. \tag{2.50}$$

To make the realization autonomous, we eliminate the $\delta(k)$ term in the realization (2.40). For this purpose let

$$y_k = z_{k+1}, \quad k \geq 0;$$

then the generalized state-space realization can be rewritten as

$$\begin{aligned} y_{k+1}E &= y_kA, \quad k \geq 0, \\ q_{k+1} &= y_kC, \quad k \geq 0; \end{aligned} \tag{2.51}$$

and the identity governing y_0 becomes

$$y_0E = z_0A + B. \tag{2.52}$$

For the matrices E and A, one can find a decomposition such that

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}, \quad U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \tag{2.53}$$

where $\lambda(E_{11}, A_{11})$ is outside the unit disk (including the unit disk) and $\lambda(E_{22}, A_{22})$ is inside the unit disk [29]. There are mf generalized eigenvalues of the matrix pencil $\lambda E - A$ (one of them is unity) that are outside the unit disk (including the unit disk), and the remaining $(n-m)f$ generalized eigenvalues are inside the unit disk. Then the size of the matrices E_{11} and A_{11} is $mf \times mf$ and the size of the matrices E_{22} and A_{22} is $(n-m)f \times (n-m)f$. Note that nf is the dimension of the generalized system given in (2.51). In order to compute the matrices U and V, numerical linear algebra techniques such as generalized matrix-sign approach, inverse-free divide and conquer, or generalized Schur decomposition approaches explained in Section 2.1 can be used. Remember that these approaches require that the matrix-pencil $\lambda E - A$ have no generalized eigenvalues on the unit circle. We must move the generalized eigenvector at unity to the outside of the unit circle without changing the eigenstructure of the matrix-pencil $\lambda E - A$. This can be done by defining

$$A_e = A + \frac{ExyEx}{yEx},$$

where x and y are the right and left eigenvectors of A such that

$$Ax = 0, \quad yA = 0.$$

One can see that the matrix-pencil $\lambda E - A_e$ has the same generalized eigenvalues and eigenvectors with the matrix-pencil $\lambda E - A$ except that the generalized eigenvector of the matrix-pencil $\lambda E - A$ at unity is moved to $\lambda = 2$. With the inputs A_e and L, one can now use one of the approaches to obtain the matrices U and V that satisfy the decomposition (2.53).

Having found U and V, the matrix U can be partitioned as

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}. \tag{2.54}$$

where U_1 is the first mf columns of U. Define

$$\begin{bmatrix} u_k & v_k \end{bmatrix} \triangleq y_k \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \quad k \geq 0. \tag{2.55}$$

Postmultiplying the generalized state-space model (2.51) by V, and using the property given in (2.53), we have two uncoupled generalized difference equations for u_k and v_k:

$$\begin{aligned} u_{k+1}E_{11} &= u_k A_{11}, \quad k \geq 0, \\ v_{k+1}E_{22} &= v_k A_{22}, \quad k \geq 0. \end{aligned}$$

Since the generalized eigenvalues of $\lambda(E_{22}, A_{22})$ lie inside the open unit disk, E_{22} is nonsingular implying

$$v_k = v_0 F^k, \quad k \geq 0, \tag{2.56}$$

where

$$F = A_{22} E_{22}^{-1}. \tag{2.57}$$

From the definition given in (2.55), y_k can be written in terms of u_k and v_k:

$$\begin{aligned} y_k &= [u_k \;\; v_k] \, U^{-1}, \quad k \geq 0 \\ &= u_k L_1 + v_k L_2, \quad k \geq 0, \end{aligned}$$

where

$$U^{-1} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}. \tag{2.58}$$

However, for the system to be stable, u_0 must be zero since the generalized eigenvalues of $\lambda(E_{11}, A_{11})$ lie outside the open unit disk. Using the definition given in (2.55), the constraint that u_0 must be zero can be written as

$$y_0 U_1 = 0. \tag{2.59}$$

Then,

$$q_0 = z_0 C, \tag{2.60}$$

$$\begin{aligned} q_{k+1} &= y_k C, \quad k \geq 0, \\ &= (u_k L_1 + v_k L_2) \, C, \quad k \geq 0, \\ &= v_k L_2 C, \quad k \geq 0, \\ &= v_0 F^k L_2 C, \quad k \geq 0, \\ &= y_0 U_2 F^k L_2 C, \quad k \geq 0. \end{aligned} \tag{2.61}$$

The only unknowns that remain to complete the solution are z_0 and y_0 which now can be found from the following nullspace using (2.38), (2.46), (2.49), (2.52) and (2.59):

$$xQ = 0, \tag{2.62}$$

where

$$x \triangleq [z_0 \;\; w \;\; y_0], \quad Q \triangleq \begin{bmatrix} 0 & -\hat{G} & A & 0 \\ \bar{I} & \hat{H} & \tilde{H} & 0 \\ 0 & 0 & -E & U_1 \end{bmatrix}. \tag{2.63}$$

Note that Q is a square matrix. Furthermore, the sum of the probabilities $q_k e$ is unity which gives a normalizing equation for z_0 and y_0:

$$\begin{aligned} \sum_{k=0}^{\infty} q_k e &= z_0 C e + y_0 U_2 (I - F)^{-1} L_2 C e \\ &= 1. \end{aligned} \tag{2.64}$$

The unique solution of the equations (2.62) and (2.64) gives us x_0 and y_0 which completes the solution for $q_k, \ k \geq 0$. The overall algorithm to obtain q_k is presented in Table 1. Once $q_k, \ k \geq 0$, is determined, the distribution of the steady-state queue length can easily be determined from

$$q_i^{level} = \lim_{n \to \infty} Pr(Q_n = i) = q_i e. \tag{2.65}$$

The expected value of the queue length, which is $q'(1)e$, can also be found from

$$\begin{aligned} \lim_{n \to \infty} \sum_{k=0}^{\infty} k Pr(Q_n = k) &= \sum_{k=0}^{\infty} (k+1) q_{k+1} e \\ &= y_0 U_2 (I - F)^{-2} L_2 C e. \end{aligned} \tag{2.66}$$

Table 2.1: Algorithm to find q_k given $A(z)$, $B(z)$, and $C(z)$.

1.	Find $N_A(z)$, $D_A(z)$, $N_B(z)$, $D_B(z)$, $N_C(z)$, and $D_C(z)$ which are the polynomial factors of $A(z)$, $B(z)$, and $C(z)$.
2.	Construct $\overline{N}_B(z)$ and $\overline{D}_B(z)$ as defined in (2.32).
3.	Evaluate $G(z)$ and $H(z)$ from (2.36) and (2.37).
4.	Obtain the matrices E, A, B, C, $\hat{G}$, $\hat{H}$, and $\tilde{H}$ using (2.41), (2.42), (2.44), (2.45), (2.47), (2.48), and (2.50).
5.	Find the matrices U and V that satisfy (2.53) using one of the spectral decomposition methods given in section-2.1.
6.	Define the matrix $\bar{I}$ as in (2.39). Obtain the matrices E_{22}, A_{22}, U_1, and L_2 from the matrix partitionings in (2.53), (2.54), (2.58), and calculate F using (2.57).
7.	Construct the matrix Q in (2.63).
8.	Find z_0 and y_0 by evaluating the left nullspace of Q and using the normalizing equation (2.64).
9.	Evaluate $q_k, \ k \geq 0$, from (2.60) and (2.61).

2.3 Applications

The algorithm proposed for the evaluation of the queue length distribution for the generic discrete-time queue finds its applications in many different areas of network performance evaluation problems because of its generality. Once the algorithm is implemented, the only inputs needed to be determined are the probability generating matrices $A(z)$, $B(z)$, and $C(z)$. In this section, two applications on the performance evaluation of ATM networks and wireless multiple access protocols will be given to present the generality and easy to use properties of the proposed solution technique.

2.3.1 Performance Analysis of a Priority Based ATM Multiplexer with Correlated Arrivals

This application is based on [14]. In this work, delay sensitive (e.g., voice with silence detection) and loss sensitive (e.g., data) traffic flows are considered for the performance analysis of an ATM multiplexer. There are two queues in the multiplexer, one for the delay sensitive flows and one for the loss sensitive flows. High priority is given to the delay sensitive flows by transmitting the delay sensitive cells first. The loss sensitive cells are transmitted only when the queue for the delay sensitive cells is empty. A finite capacity queue is assumed for the delay sensitive flows and an infinite queue is assumed for the loss sensitive traffic flows. Both traffic flows are modeled as D-BMAPs. Our objective is to find the steady-state queue length distribution for the loss sensitive class. Defining s as the size of the finite queue for the delay sensitive flows, the queue length equations for the two queues are

$$Q_{n+1}^{(1)} = min\left(max\left(Q_n^{(1)} - 1, 0\right) + C_n^{(1)}, s\right), \tag{2.67}$$

$$Q_{n+1}^{(2)} = max\left(Q_n^{(2)} - 1(Q_n^{(1)} = 0), 0\right) + C_n^{(2)}, \tag{2.68}$$

where $Q_n^{(1)}$ ($Q_n^{(2)}$) is the queue length for the delay sensitive (loss sensitive) class at the beginning of slot n and $C_n^{(1)}$ ($C_n^{(2)}$) denotes the number of arrivals to the delay sensitive (loss sensitive) queue at slot n. The p.g.m.'s associated with the arrival processes of high priority and low priority flows are $C^{(1)}(z)$ and $C^{(2)}(z)$, respectively.

One can use the algorithm proposed for the solution of the steady-state queue length distribution for the loss sensitive (low priority) class by determining the associated probability generating matrices $A(z)$, $B(z)$, and $C(z)$. Since there are no A type arrivals to the loss sensitive queue, $A(z)$ and $C(z)$ are

$$\begin{aligned} A(z) &= 1, \\ C(z) &= C^{(2)}(z). \end{aligned}$$

Now, we will find the p.g.m. $B(z)$ for the service process of the loss sensitive class. From (2.68), one can see that the service process of the low priority queue depends on the queue length evolution process of the high priority queue. This dependency can be exploited by a Markov chain that governs the queue length process of the high priority class. Let $R(z)$ denote the p.g.m. associated with the arrival process of the high priority queue. Writing $R(z)$ in terms of an expansion at $z = 0$

$$R(z) = \sum_{k=0}^{\infty} R_k z^k,$$

the transition matrix of the Markov chain that governs the queue length of the high priority queue turns out to be

$$Q = \begin{bmatrix} R_0 + R_1 & R_2 & R_3 & \cdots & R_s & R_0^B \\ R_0 & R_1 & R_2 & \cdots & R_{s-1} & R_1^B \\ & R_0 & R_1 & \cdots & R_{s-2} & R_2^B \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & & R_0 & R_s^B \end{bmatrix},$$

where R_i^B represents the boundary matrices such that

$$R_i^B = \sum_{k=s-i+1}^{\infty} R_k.$$

With this definition, the p.g.m. $B(z)$ is found to be

$$B(z) = QD(z), \tag{2.69}$$

where

$$D(z) = diag\{zI, I, \cdots, I\},$$

I is the identity matrix of the same size with $R(z)$. The structure of the diagonal matrix $D(z)$ can be explained by the fact that the loss sensitive cells are served only when the high priority queue is empty.

Having obtained $A(z)$, $B(z)$, and $C(z)$, one can evaluate the steady-state queue length distribution for the loss sensitive class, by making use of the proposed algorithm.

2.3.2 Performance of a Priority-Based Dynamic Capacity Allocation Scheme for Wireless ATM Systems

The performance of a priority-based dynamic capacity allocation scheme for wireless ATM systems is presented in [9] where more detail about the protocol description can be found. In [9], a priority-based scheduling for rt-VBR (real-time Variable Bit Rate) and nrt-VBR (nonreal-time Variable Bit Rate) traffic is considered.

As an example, real-time traffic is composed of voice and video cells and nonreal-time traffic is composed of data cells. The communication system is composed of uplink and downlink multiplexing systems which are composed of signaling and traffic slots. In each user terminal, the nrt-VBR traffic is kept in a FIFO queue and the rt-VBR traffic is kept in a buffer for one frame since real-time traffic is not tolerable to delay variation. Each user terminal sends its capacity request to the scheduler, in terms of the number of slots needed for both rt-VBR and nrt-VBR traffic arrived in the previous frame, using the signalling slots. The scheduler receives requests in the signalling slots from user terminals during each frame and keeps the requests in a request table. The scheduler first allocates the slots of the next frame for the rt-VBR requests. If the rt-VBR demand exceeds the capacity of the next frame, the excess rt-VBR traffic packets are dropped, otherwise the remaining slots are assigned to the nrt-VBR traffic. Then the request table is updated by keeping only the unsatisfied nrt-VBR requests in the table. In other words, the nrt-VBR requests are queued in a FIFO request queue at the scheduler which is an aggregation of the nrt-VBR queues in the user terminals.

Our objective for this example is to find the queue length distribution for the nonreal-time queue. The equation governing the length of the nonreal-time queue is

$$Q_{n+1} = max\left(Q_n + A_n - max(s - R_n, 0), 0\right), \tag{2.70}$$

where Q_n is the number of data packets in the queue at the beginning of the n^{th} frame, A_n is the random variable representing the number of nonreal-time cells arriving during the n^{th} frame, R_n is the random variable representing the number of real-time cells arriving during the n^{th} frame and s is the number of traffic slots in a frame. The probability generating matrices, $A(z)$ and $R(z)$, associated with the arrival processes of nonreal-time and real-time flows, respectively are assumed to be given.

The queue length equation (2.70) is very similar to the queue equation (2.13) with the difference that, for this example, B_n and C_n are

$$B_n = max(s - R_n, 0), \tag{2.71}$$

$$C_n = 0. \tag{2.72}$$

From (2.72), one can see that $C(z) = 1$. Then, the only input needed to be determined is $B(z)$ to find the steady-state queue length distribution of the nonreal-time queue. The p.g.m., $R(z)$, associated with the arrival process of real-time traffic can be written in the following form

$$R(z) = \sum_{k=0}^{\infty} R_k z^k.$$

With this information and from (2.71), the p.g.m. $B(z)$ turns out to be

$$B(z) = \sum_{k=0}^{s-1} R_i z^{s-i} + R(1) - \sum_{k=0}^{s-1} R_i.$$

Consequently, the steady-state probabilities of the queue length can be calculated using the algorithm proposed.

2.3.3 Performance Evaluation of a Reservation TDMA Protocol

A random reservation protocol based on a TDMA access technique proposed in [32] is considered for this application. In this protocol, voice communications and data transmissions are considered in particular, where voice terminals have a higher priority assigned than data terminals in accessing the shared channel. Two separated channels, namely uplink and signaling channels are assumed, both of which are slotted with the same duration T_s. Contentionless uplink channel is used for the transmissions of voice packets and data packets. In the uplink channel, N_c slots are grouped into a frame of duration T_f. The signalling channel is used for the reservation of data slots. Contention for slots occur in this channel as in the ALOHA-type random access scheme. There are N_v voice terminals and N_d data terminals linked to a base station.

All VTs (Voice Terminals) have a slow speech activity detector, so that the speech source is modeled by a two-state Markov Chain whose states are talkspurt and silence gap, as shown in Figure 2.1. The lengths of the talkspurts and the silence gaps are assumed to be exponentially distributed. Another assumption to discretize the chain is that the transition between the states occur in slot boundaries. Then

$$\sigma = 1 - e^{-(T_s/Q_s)},$$
$$\delta = 1 - e^{-(T_s/Q_t)},$$

where Q_s and Q_t are the mean lengths of silence gaps and talkspurts respectively. When a talkspurt starts in a VT, it tries to send a request in the signaling channel

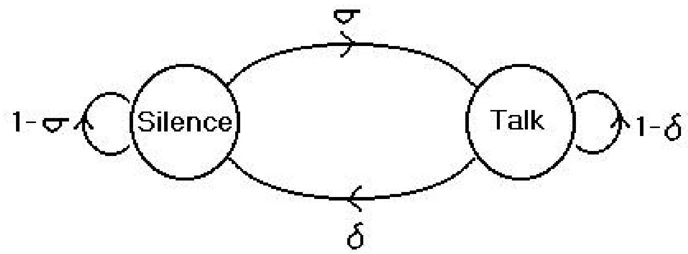

Figure 2.1: Slow speech activity detector model.

to have one slot reserved in the uplink channel every T_f. There is no priority of VTs over DTs (Data Terminals) in possible contention to send request in signaling channel. If a VT is able to send the request (no collusion in the signaling channel), the VT waits for the authorization to send from BS (Base Station) until it arrives or the talkspurt ends.

For the data terminals, the outgoing data traffic is assumed to be a Poisson process with a message arrival rate of λ/N_d messages, where message length is geometrically distributed with mean $\overline{m}$ packets. Data messages are queued in a FIFO buffer for transmission. For each arriving message, a DT sends a request in the signaling channel and waits for the authorization from BS. If there are empty slots, they are assigned for the transmission of the data message.

The BS keeps two request buffers, one for voice and one for data requests. Data requests are served only when the voice request buffer is empty, because of which voice communication has higher priority over data communication. Also there is preemption of data requests. For example, consider there is a voice request but no empty slots. If at least one of the slots is used for data packet transmission, the DT is asked to release one of its slots to assign the released slot to the voice request.

The data subsystem performance depends on the voice subsystem state, however the opposite is not true. That iss why performance evaluation of the data subsystem is important and also it is much more complicated than the evaluation of voice subsystem performance. The performance of the data subsystem is evaluated in terms of the mean data packet delay which consists of two parts: request transmission delay (T_{req}) and request queue delay (T_{BS}), which is the duration a data request waits in the data request buffer at the BS. The evaluation of T_{req} is straightforward as in the ALOHA-type random access scheme which will not be given in this paper. For the evaluation of T_{BS}, the solution tool proposed in this paper will be used.

A virtual FIFO queue that contains the queued data packets from all data terminals is considered for the evaluation of T_{BS}. The equation governing the virtual queue length is

$$Q_{n+1} = max(Q_n + A_n - B_n, 0)$$

where Q_n, A_n and B_n are the number of data packets in the virtual queue at the beginning of the n^{th} slot, the number of data packet arrivals at the n^{th} slot, and the number of data packet departures at the n^{th} slot, respectively. There are no C type arrivals, which means $C(z) = 1$. For the solution of the steady state probability distribution, the p.g.f.'s of the random variables A_n and B_n must be obtained.

The message arrival process is modeled as a Bernoulli process with probability λT_s of having a message arrival in each slot. Since the number of packets in a message is assumed to be geometrically distributed with mean $\overline{m}$, the probability that m packets arrive at the virtual queue is

$$P_a(m) = \begin{cases} 1 - \lambda T_s, & m = 0 \\ \lambda T_s \frac{1}{\overline{m}} \left(1 - \frac{1}{\overline{m}}\right), & m > 0 \end{cases},$$

then the p.g.f. of A_n is given by

$$\begin{aligned} A(z) &= \sum_{m=0}^{\infty} P_a(m) z^m \\ &= \frac{(1 - \lambda T_s)(\overline{m} - (\overline{m} - 1)z) + \lambda T_s z}{\overline{m} - (\overline{m} - 1)z}. \end{aligned}$$

The departure process can be modeled by a Markov chain whose states are the number of active talkspurts. This model is presented in Figure 2.2. As seen from Figure 2.2, the p.g.f. of a state can be written as

$$b_i(z) = \begin{cases} \frac{i}{N_c} + \frac{N_c - i}{N_c} z, & i \leq N_c \\ 1, & i > N_c \end{cases},$$

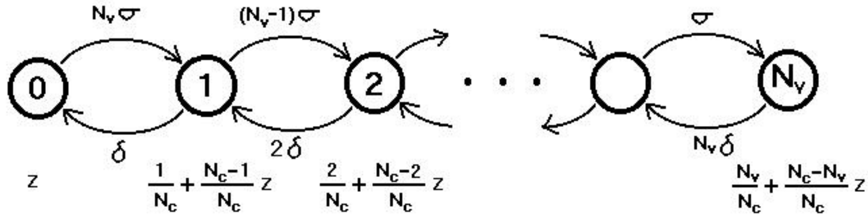

Figure 2.2: Markov chain model for the departure process.

So the p.g.m. of the departure process $B(z)$ can be evaluated as in (2.69)

$$B(z) = QD(z),$$

where

$$D(z) = diag\{b_1(z), b_2(z), \cdots, b_{N_v}(z)\},$$

and Q is the transition matrix of the Markov chain in Figure 2.2.

Once $A(z)$ and $B(z)$ are found, one can find the probability distribution of the virtual queue length using the proposed algorithm. Then the mean value of the virtual queue delay can be derived using Little's formula as

$$T_{BS} = \frac{q'(1)e}{\lambda T_s}.$$

The expected value of the queue length, $q'(1)e$, can be found from equation (2.66).

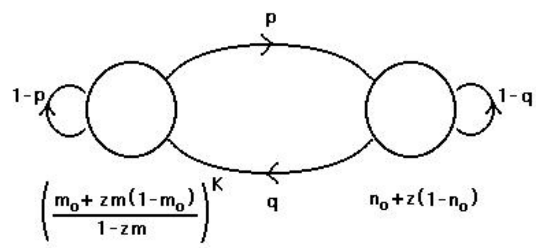

Figure 2.3: Example Markov chain structure with state p.g.f.s.

2.4 Numerical Examples

In this section, we present a set of probability models by which we show the numerical accuracy and numerical stability of the solution technique proposed in this paper. The algorithm is implemented in Matlab. For the numerical examples we assume the Markov chain of the structure shown in Figure 2.3 unless otherwise is stated. The corresponding p.g.m is then can be given as

$$X(z) = N(z)D(z)^{-1}, \tag{2.73}$$

where

$$\begin{aligned} N(z) &= \begin{bmatrix} (1-p)(m_0 + m(1-m_0)z)^K & p(n_0 + (1-n_0)z) \\ q(m_0 + m(1-m_0)z)^K & (1-q)(n_0 + (1-n_0)z) \end{bmatrix}, \\ D(z) &= \begin{bmatrix} (1-mz)^K & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{2.74}$$

By changing the parameters, one can play with the dimensionality and the utilization of the problem.

2.4.1 Example 1

First, we present a simulation result to show that our algorithmic solution works properly for generic discrete-time queues. As an example, we assume that the p.g.m.'s $A(z)$, $B(z)$, and $C(z)$ are in the form (2.73) with the parameter $K = 1$. In the simulation, we counted the number of occurrences of first hundred states and then divided this number by the simulation duration to obtain an approximation for the stationary state probabilities. The simulation duration we used is one million transitions. We present the results of the simulation and the results of our solution technique in Figure 2.4. As can be seen from the figure,

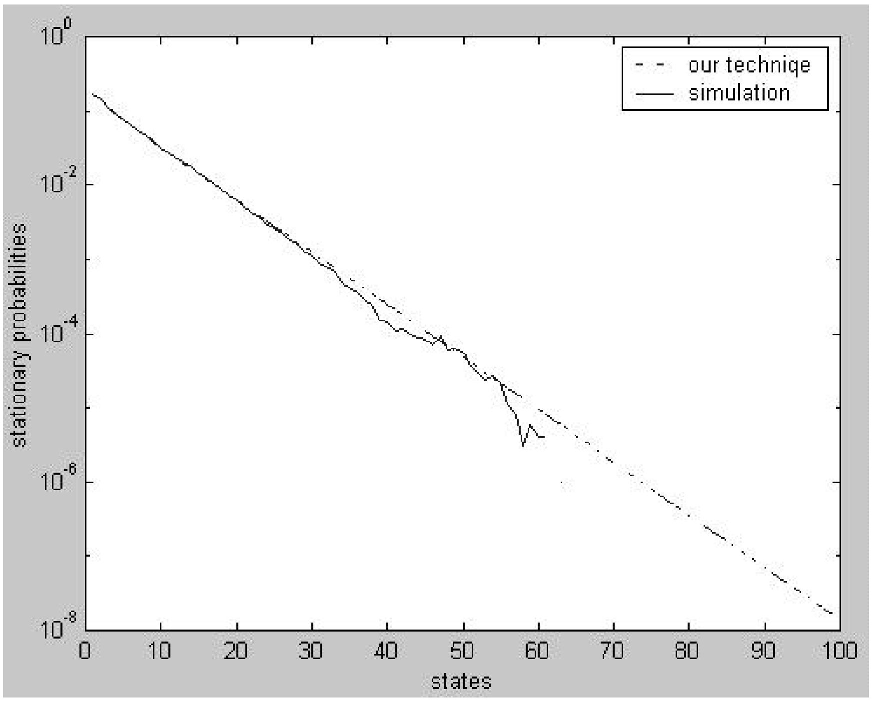

Figure 2.4: Stationary probabilities

the results are very close for lower indexed states and some deviation occurs for higher indexed states. Since we assumed that the queue is empty at the beginning of the simulation and since it takes time to reach high indexed states, the error in the approximation for steady-state probabilities increase for higher indexed states. The results of our algorithm also shows the exponential behaviour of the stationary probabilities that comes from the mixed matrix geometric form.

2.4.2 Example 2

As a second example, we assume a discrete-time infinite G/M/1 type Markov chain probability model to compare the results of the algorithm with the results of a teletraffic analysis software *TELPACK* [1], that provides solution methods for structured Markov chains of generic types M/G/1, G/M/1, and QBD (Quasi-Birth-Death Process). Since the example model is a G/M/1 system and since we assume that the only arrivals are of type A (no C type arrivals), the p.g.m.'s $A(z)$ and $C(z)$ are

$$A(z) = z, \tag{2.75}$$

$$C(z) = 1. \tag{2.76}$$

To make the system more complicated, we assume the departures are modulated by a Markov chain as shown in Figure 2.3 with $K = 2$. Then the p.g.m $B(z)$ is in the form (2.73) with $K = 2$.

The utilization for the queue length equation (2.13) is defined as

$$\rho = \frac{E\left[A_n + C_n\right]}{E\left[B_n\right]}.$$

Then the utilization for the system is

$$\rho = \frac{(p+q)(1-m)^2}{2qm + p(1-n_0)(1-m)^2}$$

For a utilization of 0.99, the results of the TELPACK software and the results of our algorithm for 10 level probabilities are shown in Table 2.2.

Table 2.2: comparison of level probabilities

	level probabilities		
level	TELPACK	ALGORITHM	% |difference|
0	$1.326361497945111\,10^{-2}$	$1.326361497943781\,10^{-2}$	$1.002752636512216\,10^{-12}$
1	$1.193606451338161\,10^{-2}$	$1.193606451336795\,10^{-2}$	$1.144800851651369\,10^{-12}$
2	$1.177993097740582\,10^{-2}$	$1.177993097739479\,10^{-2}$	$9.364321917424439\,10^{-13}$
3	$1.163745720773079\,10^{-2}$	$1.163745720772005\,10^{-2}$	$9.228539231618345\,10^{-13}$
4	$1.149682510218513\,10^{-2}$	$1.149682510217466\,10^{-2}$	$9.110567704441052\,10^{-13}$
5	$1.135789366378532\,10^{-2}$	$1.135789366377511\,10^{-2}$	$8.992910242792781\,10^{-13}$
6	$1.122064113115827\,10^{-2}$	$1.122064113114831\,10^{-2}$	$8.874103213643951\,10^{-13}$
7	$1.108504720352385\,10^{-2}$	$1.108504720351414\,10^{-2}$	$8.758868696530405\,10^{-13}$
8	$1.095109183762674\,10^{-2}$	$1.095109183761728\,10^{-2}$	$8.639487256857900\,10^{-13}$
9	$1.081875523254529\,10^{-2}$	$1.081875523253607\,10^{-2}$	$8.522288448756611\,10^{-13}$

As can be seen from Table 2.2, the difference between the level probabilities using two different approaches is in the order of 10^{-12} even for a high utilization of 0.99. These results show the accuracy of the algorithm proposed.

2.4.3 Example 3

In this final example, we present the robustness of the solution technique we proposed by increasing the utilization and dimensions of the problem. As in subsection 2.4.1, we assume that the p.g.m.'s $A(z)$, $B(z)$, and $C(z)$ are in the form (2.73), but this time we increase the value of the parameter K for each p.g.m. to increase the size of the problem. Since the two computationally intensive parts of the algorithm are the computation of the generalized invariant subspaces for the decomposition (2.53) and the calculation of the nullspace of Q in (2.62), we measure the accuracy of the results by the l_2 norms of the offdiagonal blocks in (2.53) and by the l_2 norm of xQ in (2.53), both of which must be zero in exact arithmetic. In the tables below, S represents the sizes of the square matrices E and A and so the dimension of the system in (2.51).

Table 2.3: maximum of the l_2 norms of offdiagonal blocks and l_2 norm of xQ, for increasing problem dimensionality, ($\rho = 0.7$)

S	maximum of offdiagonal block norms	$norm(xQ)$
48	1.342648214830433e-012	1.242071801639783e-015
72	1.952509714239076e-012	4.126121191903769e-015
96	1.477542394244671e-011	2.556349991273195e-015
120	2.983995273098117e-010	3.979176143021537e-015
144	4.317609200216796e-009	9.403207213477917e-015

Table 2.4: maximum of the l_2 norms of offdiagonal blocks and l_2 norm of xQ, for increasing problem dimensionality, ($\rho = 0.8$)

S	maximum of offdiagonal block norms	$norm(xQ)$
48	6.211781089504598e-012	1.430838752630128e-015
72	1.830138266682799e-011	2.830479414370544e-015
96	4.220257476816869e-011	2.388406387936671e-015
120	8.429692499589692e-010	3.366524795541218e-015
144	5.169730959986296e-008	1.100560956208918e-014

From the Tables 2.3, 2.4, 2.5, and 2.6, we observe that as the dimensionality of the system (2.51) and the utilization of the queue increase, although the norms of the offdiagonal blocks increase, the norm of xQ does not change much and the algorithm still performs well.

Table 2.5: maximum of the l_2 norms of offdiagonal blocks and l_2 norm of xQ, for increasing problem dimensionality, ($\rho = 0.9$)

S	maximum of offdiagonal block norms	$norm(xQ)$
48	2.117625519382216e-011	1.508637156705881e-015
72	1.101571801983246e-010	3.814176520149427e-015
96	4.219467775179453e-009	2.299128748229626e-015
120	8.677716323290952e-009	3.800967849950413e-015
144	2.517828699104494e-007	4.875146501468081e-015

Table 2.6: maximum of the l_2 norms of offdiagonal blocks and l_2 norm of xQ, for increasing problem dimensionality, ($\rho = 0.99$)

S	maximum of offdiagonal block norms	$norm(xQ)$
48	1.932178090484696e-010	2.400034247987217e-014
72	3.872151758260722e-008	3.246679467345070e-015
96	3.453556692175046e-008	6.309391787813667e-015
120	8.909872519780038e-007	3.408378878329309e-014
144	8.382269805189829e-006	5.768479631604270e-015

Chapter 3

GENERALIZED DISCRETE-TIME QBD

Let $P = (p_{ij})$ be the transition matrix of an irreducible Markov Chain with finite state space. Our aim is to find the unique stationary probability vector x, which satisfies the equation

$$x = xP, \quad xe = 1, \tag{3.1}$$

where e is a column vector of ones of suitable size.

We assume the following structure on the transition matrix P:

- P has repeating rows (or columns) such that for some fixed numbers m, a and K

$$p_{ij} = p_{i+m,j+m} \ , \quad a \leq i, j \leq a + m(K-2)$$

- P is banded, that is, for some fixed numbers g and h,

$$p_{ij} = 0, \quad j < i - g \ or \ j > i + h$$

With this structure, P can be written in the following block partitioned form:

$$P = \left[\begin{array}{ccccccccccc}
D_0 & D_1 & \cdots & D_\eta & & & & & & & \\
E_{-1} & P_0 & \cdots & P_{\eta-1} & P_\eta & & & & & & \\
E_{-2} & P_{-1} & \cdots & P_{\eta-2} & P_{\eta-1} & P_\eta & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & & \\
E_{-\gamma} & P_{-\gamma+1} & \cdots & P_{\eta-\gamma} & P_{\eta-\gamma+1} & P_{\eta-\gamma+2} & \cdots & P_\eta & & & \\
 & P_{-\gamma} & \cdots & P_{\eta-\gamma-1} & P_{\eta-\gamma} & P_{\eta-\gamma+1} & \cdots & \ddots & P_\eta & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & & P_{-\gamma} & \cdots & P_{\eta-\gamma} & P_{\eta-\gamma+1} & P\eta-\gamma+2 & \cdots & P_{\eta-1} & F_\eta \\
 & & & & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & & & P_{-\gamma} & P_{-\gamma+1} & P-\gamma+2 & \cdots & P_{-1} & F_1 \\
 & & & & & & C_{-\gamma} & C_{-\gamma+1} & \cdots & C_{-1} & F_0
\end{array}\right], \quad (3.2)$$

where D_0 is an $a \times a$ matrix, D_i, $0 < i \leq \eta$ are $a \times m$ matrices, E_i, $-\gamma \leq i < 0$ are $m \times a$ matrices, P_i are $m \times m$ matrices, F_0 is a $b \times b$ matrix, F_i, $0 < i \leq \eta$ are $m \times b$ matrices, C_i, $-\gamma \leq i < 0$ are $b \times m$ matrices. Also note that the probability transition matrix P is of size $(a+b+m(K-1)) \times (a+b+m(K-1))$.

3.1 Problem Formulation

We first partition

$$x = \left[\begin{array}{ccccc} x_0 & x_1 & \cdots & x_{K-1} & x_K \end{array}\right].$$

For the solution of $x = xP$, one obtains $K+1$ matrix equations one for each x_i, $i = 0, \ldots, K$. Note that x_0 is of size $1 \times a$, x_K is of size $1 \times b$ and x_k's, $0 < k < K$ are of size $1 \times m$. We refer to x_k as the solution vector for level k. In the problem formulation to solve (3.1) for P defined as in (3.2), we concentrate on arbitrary values of η and γ. Then we have the following $\eta + \gamma + 2$ boundary matrix equations:

$$\begin{aligned}
x_0 &= x_0 D_0 + x_1 E_{-1} + x_2 E_{-2} + \cdots + x_\gamma E_{-\gamma} \\
x_1 &= x_0 D_1 + x_1 P_0 + x_2 P_{-1} + \cdots + x_\gamma P_{-\gamma+1} + x_{\gamma+1} P_{-\gamma} \\
x_2 &= x_0 D_2 + x_1 P_1 + x_2 P_0 + \cdots + x_\gamma P_{-\gamma+2} + x_{\gamma+1} P_{-\gamma+1} + x_{\gamma+2} P_{-\gamma} \\
\vdots &= \vdots \\
x_k &= x_0 D_k + \sum_{i=1}^{k+\gamma} x_i P_{k-i}, \quad 1 \leq k \leq \eta \qquad (3.3)
\end{aligned}$$

and

$$
\begin{aligned}
x_k &= x_K C_{k-K} + \sum_{i=1}^{K-k+\eta} x_{K-i} P_{i+k-K}, \quad K-\gamma \le k \le K-1, \\
x_K &= x_{K-\eta} F_\eta + x_{K-\eta+1} F_{\eta-1} + \cdots + x_K F_0.
\end{aligned} \tag{3.4}
$$

The remaining $K-\eta-\gamma-2$ matrix equations for the repeating columns can be immediately rewritten in a linear difference equation framework as follows:

$$
x_k = x_{k-\eta} P_\eta + x_{k-\eta+1} P_{\eta-1} + \cdots + x_k P_0 + x_{k+1} P_{-1} + \cdots + x_{k+\gamma} P_{-\gamma}, \quad \eta+1 \le k \le K-\gamma-1. \tag{3.5}
$$

By defining

$$
z_k = \begin{bmatrix} x_{k+1} & x_{k+2} & \cdots & x_{k+\eta+\gamma-1} & x_{k+\eta+\gamma} \end{bmatrix}, \; k = 0, 1, ..., R,
$$

where $R = K - \eta - \gamma - 1$ is the number of repeating columns (or rows) in the transition probability matrix P, we can write the R matrix equations in (3.5) in the form of the following *generalized state-space representation*:

$$
z_{k+1} E = z_k A, \quad 0 \le k \le R-1, \tag{3.6}
$$

$$
x_{k+1} = \begin{cases} z_k C_1, & 0 \le k \le R, \\ z_R C_{k-R+1}, & R+1 \le k \le K-2, \end{cases} \tag{3.7}
$$

where E, A, and C_i are defined as follows:

$$
E = \begin{bmatrix} I & & & \\ & \ddots & & \\ & & I & \\ & & & P_{-\gamma} \end{bmatrix}, \tag{3.8}
$$

$$
A = \begin{bmatrix} & & & & & -P_\eta \\ I & & & & & -P_{\eta-1} \\ & \ddots & & & & \vdots \\ & & I & & & I - P_0 \\ & & & \ddots & & \vdots \\ & & & & I & -P_{-\gamma+1} \end{bmatrix}, \tag{3.9}
$$

and C_i is a block column vector, whose i^{th} block row is an identity matrix of appropriate size, for example,

$$
C_3 = \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix}.
$$

Note that the size of z_k is $1 \times (\eta+\gamma)m$, E and A are of sizes $(\eta+\gamma)m \times (\eta+\gamma)m$, and C is of size $(\eta + \gamma)m \times m$.

When the matrix E defined above is invertible, then this representation is called a regular state-space representation, but in this study, we will not make this assumption. Here, z_k is called the *descriptor* (or the semi-state) which reduces to the definition of *state* when E is nonsingular.

Our main result is the following: we first decompose the descriptor system (3.6) into its forward and backward subsystems. Using this decomposition, one obtains row vectors g_1 and g_2 and four matrices F_1, F_2, Y_1 and Y_2 of appropriate sizes so that the solution vector for the descriptor z_k can be expressed in mixed matrix geometric form:

$$z_k = g_1 F_1^k Y_1 + g_2 F_2^{R-k} Y_2, \ 0 \leq k \leq R. \tag{3.10}$$

Then it is very easy to obtain the level probabilities x_{k+1}, $0 \leq k \leq K-2$, using equation (3.7). The first term (second term) in (3.10) can be viewed as the output of the forward subsystem (backward subsystem) obtained through the decomposition of the descriptor system.

We will now explain the framework to decompose the descriptor system into its forward and backward subsystems. We first note that the matrix pencil $\lambda E - A$ is known to have one singularity at $\lambda = 1$, γm singularities (including the one at $\lambda = 1$) outside the open unit disk, and ηm singularities in the open unit disk. Note that the number of singularities yields the dimension of the generalized system given in (3.6).

Similar to the work done in Section (2.2), for the matrices E and A, one can find a decomposition such that

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}, \ U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \tag{3.11}$$

where $\lambda(E_{11}, A_{11})$ is outside the unit disk (including the unit disk) and $\lambda(E_{22}, A_{22})$ is inside the unit disk. There are γm generalized eigenvalues of the matrix-pencil $\lambda E - A$ (one of them is unity) that are outside the unit disk (including the unit disk), and the remaining ηm generalized eigenvalues are inside the unit disk. Then the size of the matrices E_{11} and A_{11} is $\gamma m \times \gamma m$ and the size of the matrices E_{22} and A_{22} is $\eta m \times \eta m$. In order to compute the matrices U and V, numerical linear algebra techniques such as generalized matrix-sign approach, inverse-free divide and conquer, or generalized Schur decomposition approaches explained in Section 2.1 can be used. Remember that these approaches require that the matrix-pencil $\lambda E - A$ have no generalized eigenvalues on the unit circle. We must move the generalized eigenvector at unity to the outside of the unit circle without changing the eigenstructure of the matrix-pencil $\lambda E - A$. This can be done by defining

$$A_e = A + \frac{ExyEx}{yEx}, \tag{3.12}$$

where x and y are the right and left eigenvectors of A such that

$$Ax = 0, \quad yA = 0.$$

One can see that the matrix-pencil $\lambda E - A_e$ has the same generalized eigenvalues and eigenvectors as the matrix-pencil $\lambda E - A$ except that the generalized eigenvector of the matrix-pencil $\lambda E - A$ at unity is moved to $\lambda = 2$. With the inputs A_e and L, one can now use one of the approaches to obtain the matrices U and V that satisfy the decomposition (3.11).

Having found U and V, the matrix U can be partitioned as

$$U = \left[\begin{array}{cc} U_1 & U_2 \end{array} \right], \tag{3.13}$$

where U_1 is the first γm columns of U.

Defining

$$\left[\begin{array}{cc} u_k & v_k \end{array} \right] \triangleq z_k \left[\begin{array}{cc} U_1 & U_2 \end{array} \right], \; 0 \le k \le R,$$

and postmultiplying the generalized state-space model (3.6) by V, we have two uncoupled generalized difference equations for u_k and v_k:

$$\begin{aligned} u_{k+1}E_{11} &= u_k A_{11}, \; 0 \le k \le R-1, \\ v_{k+1}E_{22} &= v_k A_{22}, \; 0 \le k \le R-1. \end{aligned}$$

Since $\lambda(E_{22}, A_{22})$ lies inside the open unit disk, E_{22} is nonsingular implying

$$v_k = v_0 F_1^k, \; 0 \le k \le R, \tag{3.14}$$

where

$$F_1 = A_{22}E_{22}^{-1}. \tag{3.15}$$

In equation (3.14), the value of v_k is expressed in terms of its previous values and we therefore call the system in (3.14), the forward subsystem associated with (3.6).

Similarly, since $\lambda(E_{11}, A_{11})$ lies outside the open unit disk, A_{11} is nonsingular implying

$$u_k = u_R F_2^{R-k}, \; 0 \le k \le R, \tag{3.16}$$

where

$$F_2 = E_{11} A_{11}^{-1}. \tag{3.17}$$

In Equation (3.16), the value of u_k is expressed in terms of its future values and we therefore call the system in (3.16), the backward subsystem associated with (3.6).

Let us now partition U^{-1} as

$$U^{-1} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}. \tag{3.18}$$

Then

$$z_k = v_k L_2 + u_k L_1 = \underbrace{v_0}_{g_1} F_1^k \underbrace{L_2}_{Y_1} + \underbrace{u_R}_{g_2} F_2^{R-k} \underbrace{L_1}_{Y_2}, \quad 0 \le k \le R, \tag{3.19}$$

and the level probabilities $x_{k+1}, \;\; 0 \le k \le K-2$, can be obtained easily from equation (3.7). The only unknowns needed to obtain the steady-state vector are the vectors x_0, u_R, v_0 and x_K. In order to find these four unknown vectors, we rewrite the $\eta + \gamma + 2$ boundary matrix equations in (3.4) in terms of the concatenated vector $y = \begin{bmatrix} x_0 & u_R & v_0 & x_K \end{bmatrix}$. Direct substitution into the boundary equations leads to

$$yQ = 0, \tag{3.20}$$

where the transpose of Q is,

$$Q^T = \begin{bmatrix}
(D_0 - I)^T & (\sum_{i=1}^{\gamma} G_i E_{-i})^T & (\sum_{i=1}^{\gamma} H_i E_{-i})^T & 0 \\
D_1^T & (-G_1 + \sum_{i=1}^{1+\gamma} G_i P_{1-i})^T & (-H_1 + \sum_{i=1}^{1+\gamma} H_i P_{1-i})^T & 0 \\
\vdots & \vdots & \vdots & \vdots \\
D_k^T & (-G_k + \sum_{i=1}^{k+\gamma} G_i P_{k-i})^T & (-H_k + \sum_{i=1}^{k+\gamma} H_i P_{k-i})^T & 0 \\
\vdots & \vdots & \vdots & \vdots \\
D_\eta^T & (-G_\eta + \sum_{i=1}^{\eta+\gamma} G_i P_{\eta-i})^T & (-H_\eta + \sum_{i=1}^{\eta+\gamma} H_i P_{\eta-i})^T & 0 \\
0 & (-J_{K-\gamma} + \sum_{i=1}^{\gamma+\eta} J_{\gamma+\eta-i+1} P_{i-\gamma})^T & (-W_{K-\gamma} + \sum_{i=1}^{\gamma+\eta} W_{\gamma+\eta-i+1} P_{i-\gamma})^T & C_{-\gamma}^T \\
\vdots & \vdots & \vdots & \vdots \\
0 & (-J_k + \sum_{i=1}^{K-k+\eta} J_{\gamma+\eta-i+1} P_{i+k-K})^T & (-W_k + \sum_{i=1}^{K-k+\eta} W_{\gamma+\eta-i+1} P_{i+k-K})^T & C_{k-K}^T \\
\vdots & \vdots & \vdots & \vdots \\
0 & (-J_{K-1} + \sum_{i=1}^{1+\eta} J_{\gamma+\eta-i+1} P_{i-1})^T & (-W_{K-1} + \sum_{i=1}^{1+\eta} W_{\gamma+\eta-i+1} P_{i-1})^T & C_{-1}^T \\
0 & (\sum_{i=1}^{\eta} J_{\gamma+\eta-i+1} F_i)^T & (\sum_{i=1}^{\eta} W_{\gamma+\eta-i+1} F_i)^T & (F_0 - I)^T
\end{bmatrix}, \tag{3.21}$$

in which G_i's, K_i's, J_i's and H_i's $(i = 1, 2, \cdots, \eta + \gamma)$ are the matrices defined by the partitionings

$$\begin{aligned}
F_2^R L_1 &= \begin{bmatrix} G_1 & G_2 & \cdots & G_{\eta+\gamma} \end{bmatrix}, \\
F_1^R L_2 &= \begin{bmatrix} K_1 & K_2 & \cdots & K_{\eta+\gamma} \end{bmatrix}, \\
L_1 &= \begin{bmatrix} W_1 & W_2 & \cdots & W_{\eta+\gamma} \end{bmatrix}, \\
L_2 &= \begin{bmatrix} H_1 & H_2 & \cdots & H_{\eta+\gamma} \end{bmatrix}.
\end{aligned} \tag{3.22}$$

Then the null space of the matrix Q gives us the concatenated vector y up to a normalization constant which can be found from the fact that the sum of the probabilities $x_k,\ k = 0,\ \ldots, K$, is unity.

$$\begin{aligned}
\sum_{k=0}^{K} x_k e &= x_0 e + \sum_{k=0}^{K-2} x_{(k+1)} e + x_K e, \\
&= x_0 e + \sum_{k=0}^{R} v_0 F_1^k L_2 C_1 e + \sum_{k=R+1}^{K-2} v_0 F_1^R L_2 C_{k-R+1} e + \\
&\quad \sum_{k=0}^{R} u_R F_2^{R-k} L_1 C_1 e + \sum_{k=R+1}^{K-2} u_R L_1 C_{k-R+1} e + x_K e, \\
&= x_0 e + v_0 S_1 L_2 C e + u_R S_2 L_1 C e + \\
&\quad \sum_{k=R+1}^{K-2} v_0 F_1^R L_2 C_{k-R+1} e + \sum_{k=R+1}^{K-2} u_R L_1 C_{k-R+1} e + x_K e, \\
&= 1, \qquad (3.23)
\end{aligned}$$

where e is column vector of ones of appropriate size and S_1, S_2 are defined as:

$$S_1 = \sum_{k=0}^{R} v_0 F_1^k L_2 C_1, \qquad (3.24)$$

$$S_2 = \sum_{k=0}^{R} u_R F_2^{R-k} L_1 C_1. \qquad (3.25)$$

It is known that, if A is power summable [25], that is if all the eigenvalues of matrix A are inside the open unit disk,

$$S = \sum_{k=0}^{s} A^k = (I - A^{s+1})(I - A)^{-1}.$$

Then, since F_1 is power summable,

$$S_1 = v_0 (I - F_1^{R+1})(I - F_1)^{-1} L_2 C_1. \qquad (3.26)$$

However, F_2 has an eigenvalue at unity, and a simplification of the form (3.26) for S_2 is not immediate. We now define the following matrix

$$M = \frac{xy}{yx},$$

where x and y are right and left invariant eigenvectors of F_2 respectively, such that $F_2 x = x$ and $y F_2 = y$. Note that the matrix $(I - \hat{F}_2)$ with

$$\hat{F}_2 = F_2 - M$$

is invertible since the eigenvalue of F_2 at unity is moved to the origin by this transformation. One can show by induction that

$$F_2^k = \begin{cases} \hat{F}_2^k + M & \text{if } k \geq 1, \\ I & \text{if } k = 0. \end{cases}$$

Then we have the following expression for S_2:

$$\begin{aligned} S_2 &= \sum_{k=0}^{R} u_R F_2^{R-k} L_1 C_1, \\ &= u_R[(I - \hat{F}_2^{R+1})(I - \hat{F}_2)^{-1} + RM] L_1 C_1. \end{aligned}$$

Now, we can rewrite (3.23) as:

$$\begin{aligned} \sum_{k=0}^{K} x_k e &= x_0 e + \sum_{k=R+1}^{K-2} v_0 F_1^R L_2 C_{k-R+1} e + \sum_{k=R+1}^{K-2} u_R L_1 C_{k-R+1} e + x_K e + \\ & \quad v_0 (I - F_1^{R+1})(I - F_1)^{-1} L_2 C_1 e + \\ & \quad u_R[(I - \hat{F}_2^{R+1})(I - \hat{F}_2)^{-1} + RM] L_1 C_1 e, \\ &= 1. \end{aligned} \tag{3.27}$$

The unique solution of (3.20) and (3.27) gives us the vectors x_0, u_R, v_0 and x_K which completes the solution. The overall algorithm is summarized in Table (3.1).

3.2 Infinite Queue Capacity Case

The algorithm given in the previous section can be used to solve both finite and infinite queue steady state level probabilities with some reductions. In other words, the solution of the infinite queue steady state level probabilities is a special case of the solution algorithm given in the previous section. In this section the reductions on the algorithm for the infinite queue will be given briefly.

The form of the transition matrix for the infinite queue is

$$P = \begin{bmatrix} D_0 & D_1 & \cdots & D_\eta & & & & & \\ E_{-1} & P_0 & \cdots & P_{\eta-1} & P_\eta & & & & \\ E_{-2} & P_{-1} & \cdots & P_{\eta-2} & P_{\eta-1} & P_\eta & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & \\ E_{-\gamma} & P_{-\gamma+1} & \cdots & P_{\eta-\gamma} & P_{\eta-\gamma+1} & P_{\eta-\gamma+2} & \cdots & P_\eta & \\ & P_{-\gamma} & \cdots & P_{\eta-\gamma-1} & P_{\eta-\gamma} & P_{\eta-\gamma+1} & \cdots & \ddots & P_\eta \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{3.29}$$

Table 3.1: Algorithm to obtain the level probabilities x_k's for finite generalized QBD.

1.	Construct the matrices E and A as defined in (3.8) and (3.9).
2.	Construct the matrix A_e as defined in (3.12) and obtain the matrices U and V that satisfy (3.11) using one of the spectral decomposition methods given in Section 2.1.
3.	Obtain the matrices E_{11}, A_{11}, E_{22}, A_{22}, L_1, and L_2 from the matrix partitionings in (3.11), and (3.18). Then define F_1 and F_2 as in (3.15) and (3.17).
4.	Construct the matrix Q as defined in (3.21) using G_i, K_i, W_i and H_i as defined in the partitionings in (3.22).
5.	Find the vector spaces of x_0, u_R, v_0 and x_K by evaluating the left nullspace of the matrix Q and normalize them using equation (3.27).
7.	Define g_1, Y_1, g_2 and Y_2 as in (3.19) to obtain the matrix geometric expression $$z_k = g_1 F_1^k Y_1 + g_2 F_2^{R-k} Y_2, \quad 0 \le k \le R. \qquad (3.28)$$
8.	Evaluate the remaining level probabilities x_k, $1 \le k \le K-1$ from (3.7) and (3.28).

and the steady state probability vector x has the form

$$x = \begin{bmatrix} x_0 & x_1 & x_2 & \cdots \end{bmatrix}.$$

From the transition matrix P, we observe that there are only $\eta + 1$ boundary equations:

$$\begin{aligned}
x_0 &= x_0 D_0 + x_1 E_{-1} + x_2 E_{-2} + \cdots + x_\gamma E_{-\gamma}, \\
x_1 &= x_0 D_1 + x_1 P_0 + x_2 P_{-1} + \cdots + x_\gamma P_{-\gamma+1} + x_{\gamma+1} P_{-\gamma}, \\
x_2 &= x_0 D_2 + x_1 P_1 + x_2 P_0 + \cdots + x_\gamma P_{-\gamma+2} + x_{\gamma+1} P_{-\gamma+1} + x_{\gamma+2} P_{-\gamma}, \\
\vdots &= \vdots \\
x_k &= x_0 D_k + \sum_{i=1}^{k+\gamma} x_i P_{k-i}, \quad 1 \le k \le \eta.
\end{aligned}$$

The remaining equations for the solution are

$$x_k = x_{k-\eta} P_\eta + x_{k-\eta+1} P_{\eta-1} + \cdots + x_k P_0 + x_{k+1} P_{-1} + \cdots + x_{k+\gamma} P_{-\gamma}, \quad \eta + 1 \le k. \qquad (3.30)$$

By defining

$$z_k = \begin{bmatrix} x_{k+1} & x_{k+2} & \cdots & x_{k+\eta+\gamma-1} & x_{k+\eta+\gamma} \end{bmatrix}, \; k = 0, 1, ..., \infty,$$

we can write the matrix equations in (3.30) in the form of the following *generalized state-space representation*:

$$z_{k+1} E = z_k A, \quad 0 \le k,$$

$$x_{k+1} = z_k C_1, \;\; 0 \leq k, \tag{3.31}$$

where E, A, and C_i are defined as in the previous section. Note that the formulation of x_{k+1} for the infinite queue case given in equation (3.31) is much more simpler than the formulation of x_{k+1} for the finite queue case. This important difference will let us write not only the descriptor formulation but also the steady state level probabilities in matrix-geometric form. Again for the matrices E and A, one can find a decomposition such that

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}, \;\; U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \tag{3.32}$$

where all the generalized eigenvalues of $\lambda E_{11}, A_{11}$ are outside the unit disk (including the unit disk) and all the generalized eigenvalues of $\lambda E_{22}, A_{22}$ are inside the unit disk as explained in the previous section. Having found U and V, the matrix U can be partitioned as

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \tag{3.33}$$

where U_1 is the first γm columns of U.

Defining

$$\begin{bmatrix} u_k & v_k \end{bmatrix} \triangleq z_k \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \; 0 \leq k$$

and postmultiplying the generalized state-space model (3.6) by V, we have the two uncoupled generalized difference equations for u_k and v_k:

$$\begin{aligned} u_{k+1}E_{11} &= u_k A_{11}, \; 0 \leq k, \\ v_{k+1}E_{22} &= v_k A_{22}, \; 0 \leq k, \end{aligned}$$

Since $\lambda(E_{22}, A_{22})$ lies inside the open unit disk and E_{22} is nonsingular, the forward subsystem associated with (3.31) is

$$v_k = v_0 F^k, \; 0 \leq k \tag{3.34}$$

where

$$F = A_{22}E_{22}^{-1}. \tag{3.35}$$

However, since $\lambda(E_{11}, A_{11})$ lies outside the open unit disk, in the infinite queue case, for the system to be stable, u_0 must be zero implying all u_k's being zero. Partitioning U^{-1} as

$$U^{-1} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}, \tag{3.36}$$

then

$$x_{k+1} = z_k C_1 = v_k L_2 C_1 = \underbrace{v_0}_{g} F^k \underbrace{L_2 C_1}_{Y}, \quad 0 \le k. \tag{3.37}$$

The only unknowns needed to obtain the steady-state vector are the vectors x_0 and v_0. In order to find these unknown vectors, we rewrite the $\eta + 1$ boundary matrix equations in (3.30) in terms of the concatenated vector $y = \begin{bmatrix} x_0 & v_0 \end{bmatrix}$. Direct substitution into the boundary equations leads to

$$yQ = 0,$$

where the transpose of Q is,

$$Q^T = \begin{bmatrix} (D_0 - I)^T & (\sum_{i=1}^{\gamma} H_i E_{-i})^T \\ D_1^T & (-H_1 + \sum_{i=1}^{1+\gamma} H_i P_{1-i})^T \\ \vdots & \vdots \\ D_k^T & (-H_k + \sum_{i=1}^{k+\gamma} H_i P_{k-i})^T \\ \vdots & \vdots \\ D_\eta^T & (-H_\eta + \sum_{i=1}^{\eta+\gamma} H_i P_{\eta-i})^T \end{bmatrix}, \tag{3.38}$$

in which H_i is defined by the partitioning

$$L_2 = \begin{bmatrix} H_1 & H_2 & \cdots & H_{\eta+\gamma} \end{bmatrix}. \tag{3.39}$$

Then the null space of matrix Q gives us the concatenated vector y up to a normalization constant which can be found from the fact that the sum of the probabilities x_k, $k = 0, \ldots, \infty$, is unity.

$$\begin{aligned} \sum_{k=0}^{\infty} x_k e &= x_0 e + \sum_{k=0}^{\infty} x_{(k+1)} e \\ &= x_0 e + \sum_{k=0}^{\infty} v_0 F^k L_2 C_1 e \\ &= 1. \end{aligned}$$

Since F_1 is power summable,

$$\sum_{k=0}^{\infty} v_0 F^k L_2 C_1 = v_0 (I - F)^{-1} L_2 C_1.$$

Then we can write the normalizing equation as

$$x_0 e + v_0 (I - F)^{-1} L_2 C_1 e = 1, \tag{3.40}$$

which concludes the solution. Now, we summarize the solution algorithm for the infinite queue capacity case in table (3.2).

Table 3.2: Algorithm to obtain the level probabilities x_k's for infinite generalized QBD.

1.	Construct the matrices E and A as defined in (3.8) and (3.9).
2.	Construct the matrix A_e as defined in (3.12) and obtain the matrices U and V that satisfy (3.11) using one of the spectral decomposition methods given in Section 2.1.
3.	Obtain the matrices E_{22}, A_{22}, and L_2 from the matrix partitionings in (3.11), and (3.18). Then define F as in (3.35).
4.	Construct the matrix Q as defined in (3.38) using H_i as defined in the partitionings in (3.39).
5.	Find the vector spaces of x_0 and v_0 by evaluating the left nullspace of the matrix Q and normalize them using equation (3.40).
7.	Define g and Y, as in (3.37) to obtain the matrix geometric expression for level probabilities $$x_{k+1} = gF^kY, \quad 0 \leq k. \qquad (3.41)$$

3.3 Numerical Example

In this section, we will show the accuracy and the robustness of the proposed algorithm by a numerical example. A simple discrete-time queueing example with the queue length evolution shown below will be used for this purpose:

$$q_{n+1} = min(K, max(q_n + a_n - b_n, 0)), \quad n \geq 0, \tag{3.42}$$

where q_n is the queue length at the end of the n^{th} slot, K is the buffer size, a_n is the total number of arrivals in the n^{th} slot, b_n is the is the total number of packets served in the n^{th} slot.

The arrival process is derived from a two-state Markov chain in which in state zero, we assume batch arrivals with probability generating function $p + (1-p)z$ and in state one $q+(1-q)z^2$. The server process is assumed to have a probability generating function of $t + wz + (1 - t - w)z^2$.

In the first set of examples, utilization is set to $\rho = E(a_n)/E(b_n) = 0.9548$. In table (3.3), we present the accuracy of the results by l_2 norm of $x - Px$ for increasing K values.

Table 3.3: l_2 norm of $x - Px$ for increasing K values

K	$\parallel x - xP \parallel$
10	1.9708e-015
100	1.0204e-015
1000	1.0191e-015

Chapter 4

CONCLUSION

There is a certain need for a general queueing analysis technique that applies to a large class of queueing systems. Besides generality, the solution technique must be robust and must have good computational properties in order to handle the queueing models that have high dimensionality. Motivated by this need, in this thesis, we studied two queueing analysis problems, namely, the generic discrete-time infinite queue and the generalized discrete-time QBD process.

The solution technique we proposed for the generic discrete time infinite queue simultaneously employs the generating function approach and the generalized invariant subspace approach. In the algorithmic solution, we bypassed the step of constructing the structured Markov chain and instead we directly employed the generating function approach on the probability generating matrices corresponding to the arrival and service processes to obtain the stationary probabilities of the generic discrete-time infinite queue (1.2). This saves us from the burden of compactly representing the boundaries and the block Toeplitz submatrix that would arise in the structured Markov chain method. Then we showed that this representation can be expressed by a generalized-state space representation which allows us to apply the generalized state-space method to obtain a modified matrix-geometric representation of the steady-state queue length probabilities. Besides the proven robustness and efficiency of this method, we also avoided the determination of the zeroes of a certain matrix polynomial inside the unit circle which is the most significant drawback of the transform methods. Then we presented several wireless and ATM network performance problems from the recent literature and showed how easily one can use the solution algorithm for these problems. Finally, by numerical examples, we presented the accuracy and robustness of our solution technique. As explained before, our technique does not assume any particular chain type like M/G/1 or G/M/1 and uses the generality of the discrete-time queue equation (1.2). Arrival and service processes are simultaneously allowed to be of D-BMAP type by which one can analyze the interaction of several complex phenomena like the variability of the batch size over time, multi-server queues, server interruptions, priority queues, etc. via a unified framework. On the other hand, the algorithmic solution technique we proposed uses the generalized invariant subspace method as a mathematical engine to provide robustness and efficiency. We consider the continuous-time extension of this approach, model building, and model reduction methods to handle large state spaces, as potential areas for future research.

For the finite generalized discrete QBD problem, we showed that the repeating row (or column) structure of the transition matrix can be exploited by the matrix difference equation equations (3.5). Then we showed that this equation can be represented by the generalized state space representation which allows us to make use of the generalized invariant subspace method. Applying this method, the descriptor vector can be written in a modified mixed matrix geometric form $z_k = g_1 F_1^k Y_1 + g_2 F_2^{K-k} Y_2,\ 0 \leq k \leq R$, where the parameters are determined by the simultaneous computation of the generalized invariant subspaces and the solution of the boundary equations. The level probabilities can then easily be obtained from the descriptor vector. In the case of an infinite queue, which can be considered as a special case of the finite queue, even the solution vector takes the modified matrix geometric form $x_{k+1} = gF^k Y,\ k \geq 0$. Our solution technique solves the infinite/finite generalized QBD with a lower computational complexity then the solution using traditional QBD solution techniques. We also gave some numerical examples to show the accuracy and robustness of this technique. However, we required that η and γ are finite. Extension to the solution for $\eta \to \infty$ and $\gamma \to \infty$ is a research topic in progress.

APPENDIX A

Proof of Theorem 2

If both sides of (2.35) are multiplied by z^{-n}, one can write (2.35) as

$$q(z)\overline{G}(z) = \overline{b}(z), \tag{A.1}$$

where

$$\begin{aligned} \overline{G}(z) &= G_n + G_{n-1}z^{-1} + \cdots + G_0 z^{-n}, \\ \overline{b}(z) &= b_n + b_{n-1}z^{-1} + \cdots + b_0 z^{-n}. \end{aligned}$$

Now, we'll try to realize (A.1) by a generalized state-space realization:

$$y_{k+1}E = y_k A, \quad k \geq 0,$$

for some E and A matrices where $y(z)$ is such that $q(z)$ can be extracted from $y(z)$. Define

$$q_k^j \stackrel{\triangle}{=} q_{k+j-1}, \quad q^j(z) = \sum_{k=0}^{\infty} q_k^j z^k, \ 1 \leq j \leq n,$$

which yields

$$\begin{aligned} q^1(z) &= q(z), \\ q^j(z) &= dq^{j-1}(z) - z^{-1}q_0^{j-1}, \ 2 \leq j \leq n. \end{aligned}$$

Here, j in q_k^j should be treated as a superscript. We note that

$$q_0^j = \lim_{z \to 0} q_j(z), \ 1 \leq j \leq n,$$

must be a bounded vector by definition. It is also easy to see that

$$z^{-1}q^j(z) = q^{j+1}(z) + z^{-1}q_0^j, \ 1 \leq j \leq n-1. \tag{A.2}$$

One can show by using (A.2) and by algebraic manipulations that

$$z^{-1}q^n(z)G_0 = \bar{b}(z) - \sum_{j=0}^{n-1}\sum_{i=0}^{j} G_{j-i}q_0^{i+1}z^{j-n} - \sum_{i=1}^{n-1} G_{n-i}q_0^i z^{-1} - \sum_{i=1}^{n} G_{n-i+1}q^i(z) \tag{A.3}$$

Since $\lim_{d\to 0} q^n(z)G_0$ exists, (A.3) dictates linear constraints on $q_0^j, 1 \leq j \leq n-1$ such that $z^{-1}q^n(z)G_0$ can not have terms with z^{-i} where $i \geq 2$, in the following manner:

$$q_0^j G_0 = b_{j-1} - \sum_{i=1}^{j-1} q_0^i G_{j-i}. \tag{A.4}$$

Consequently (A.3) becomes

$$\begin{aligned} z^{-1}q^n(z)G_0 &= b_n + b_{n-1}z^{-1} - \sum_{i=1}^{n-1} G_{n-i}q_0^i z^{-1} - \\ &\quad \sum_{i=1}^{n} G_{n-i+1}q^i(z) \\ &= b_n + z^{-1}q_0^n G_0 - \sum_{i=1}^{n} G_{n-i+1}q^i(z), \end{aligned} \tag{A.5}$$

or equivalently,

$$q^{n+1}(z)G_0 = b_n - \sum_{i=1}^{n} G_{n-i+1}q^i(z). \tag{A.6}$$

Let us now define the concatenated transform vector

$$z(z) \triangleq \begin{bmatrix} q^1(z) & q^2(z) & \cdots & q^n(z) \end{bmatrix}, \tag{A.7}$$

which is the z-transform of the sequence

$$z_k \triangleq \begin{bmatrix} q_k^1 & q_k^2 & \cdots & q_k^n \end{bmatrix}.$$

Using (A.2), (A.4) and (A.6) one obtains the generalized system representation

$$\begin{aligned} z_{k+1}E &= z_k A + B\delta(k), \quad k \geq 0, \\ q_k &= z_k C, \quad k \geq 0, \end{aligned}$$

with the constraint

$$\hat{z_0}\hat{G} = v\hat{H}, \tag{A.8}$$

where E, A, B, $\delta(k)$, $\hat{z_0}$, $\hat{G}$, $\hat{H}$, C are defined in (2.41), (2.42), (2.44), (2.47) and (2.48).

Bibliography

[1] N. Akar, N. C. Oguz, and K. Sohraby, "TELPACK: An advanced teletraffic analysis package," *IEEE Communications Magazine*, vol. 36, no. 8, pp. 84–87, 1998.

[2] K. Sohraby, "On the theory of general on-off sources and their multiplexing behavior in high speed networks," in *IEEE INFOCOM*, pp. 401–410, 1993.

[3] A. T. Andersen and B. F. Nielsen, "A Markovian approach for modeling packet traffic with long range dependence," *IEEE JSAC*, vol. 16, no. 5, pp. 719–732, 1998.

[4] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* The Johns Hopkins University Press, 1981.

[5] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications.* Marcel Dekker, N.Y., 1989.

[6] G. Latouche and V. Ramaswami, "A logarithmic reduction algorithm for quasi-birth-death processes," *J. Appl. Prob.*, vol. 30, pp. 650–674, 1993.

[7] N. Akar, N. C. Oguz, and K. Sohraby, "Matrix-geometric solutions of M/G/1 type Markov chains: A unifying generalized state-space approach," *IEEE JSAC*, vol. 16, no. 5, pp. 626–639, 1998.

[8] N. Akar and E. Kavak, "Algorithmic solution for the generic discrete-time queue and its applications in network performance." submitted to INFOCOM 2002.

[9] T. V. J. G. Babu, T. Le-Ngoc, and J. F. Hayes, "Performance of a priority-based dynamic capacity allocation scheme for wireless ATM systems," *IEEE JSAC*, vol. 19, no. 2, pp. 355–369, 2001.

[10] R. Fantacci and S. Nannicini, "Performance evaluation of a reservation TDMA protocol for voice/data transmission in microcellular systems," *IEEE JSAC*, vol. 18, no. 11, pp. 2404–2416, 2000.

[11] L. Lenzini, B. Meini, and E. Mingozzi, "An efficient numerical method for performance analysis of contention MAC protocols: a case study (PRMA++)," *IEEE JSAC*, vol. 16, no. 5, pp. 653–667, 1998.

[12] M. Murata, Y. Oie, T. Suda, and H. Miyahara, "Analysis of a discrete-time single-server queue with bursty inputs for traffic control in ATM networks," *IEEE JSAC*, vol. 8, no. 3, pp. 447–458, 1990.

[13] T. Daniels and C. Blondia, "Asymptotic behavior of a discrete-time queue with long range dependent input," in *IEEE INFOCOM*, 1999.

[14] R. Jafari and K. Sohraby, "Performance analysis of priority-based ATM multiplexers with correlated arrivals," in *IEEE INFOCOM*, 1999.

[15] W. K. Grassmann and D. P. Heyman, "Computation of the steady-state probabilities for infinite-state Markov chains with repeating rows," *ORSA Journal on Computing*, vol. 5, no. 3, pp. 292–303, 1993.

[16] R. Jafari and K. Sohraby, "Combined M/G/1 - G/M/1 type structured chains: A simple algorithmic solution and applications," in *IEEE INFOCOM*, 2001.

[17] N. Akar and K. Sohraby, "An invariant subspace approach in M/G/1 and G/M/1 type Markov chains," *Stochastic Models*, vol. 13, no. 3, pp. 251–257, 1997.

[18] D. Bini and B. Meini, "On the solution of a nonlinear matrix equation arising in queueing problems," *SIAM Jour. on Matrix. Analy. Appl.*, vol. 17, pp. 906–926, 1996.

[19] V. Wallace, *The solution of quasi birth and death processes arising from multiple access computer systems.* PhD thesis, Systems Engineering Laboratory, University of Michigan, 1969.

[20] M. F. Neuts, "Markov chains with applications in queueing theory, which have a matrix-geometric invariant probability vector," *Adv. Appl. Prob.*, vol. 10, pp. 185–212, 1978.

[21] R. L. Tweedie, "Operator-geometric stationary distributions for Markov chains with applications to queueing models," *Adv. Appl. Prob.*, vol. 14, pp. 368–391, 1982.

[22] P. P. Bocharov and V. Naumov, "Matrix-geometric stationary distribution for the PH/PH/1/r queue," *J. Info. Proc. Cyber.*, vol. 22, pp. 179–186, 1986.

[23] P. P. Bocharov, "Analysis of the queue length and the output flow in single server with finite waiting room and phase-type distributions," *Probl. Control Info. Theory*, vol. 16, pp. 211–222, 1987.

[24] L. Gün, "Experimental techniques on matrix-analytical solution techniques - extensions and comparisons," *Commun. Statist. - Stochastic Models*, vol. 5, no. 4, pp. 669–682, 1989.

[25] N. Akar, N. C. Oguz, and K. Sohraby, "A novel computational method for solving finite QBD processes," *Stochastic Models*, vol. 16, no. 2, pp. 273–311, 2000.

[26] J. Demmel and B. Kagström, "Computing stable eigendecompositions of matrix pencils," *Lin. Alg. Appl.*, vol. 88/89, pp. 139–186, 1987.

[27] A. J. Laub, "Invariant subspace methods for the numerical solution of Riccati equations," in *The Riccati Equation* (S. Bittanti, A. J. Laub, and J. C. Willems, eds.), ch. 7, pp. 163–196, Springer-Verlag, Berlin, 1991.

[28] J. D. Gardiner and A. J. Laub, "A generalization of the matrix-sign-function solution for algebraic Riccati equations," *Int. Jour. Contr.*, vol. 44, pp. 823–832, 1986.

[29] Z. Bai and J. Demmel, "Inverse free parallel spectral divide and conquer algorithms for nonsymmetric eigenproblems," Computer Science Division Report CSD-94-793, University of California at Berkeley, Feb. 1994.

[30] B. Kagström, "A direct method for reordering eigenvalues in the generalized schur form of a regular matrix pair (a,b).," in *Linear for Large Scale and Real-time Applications* (in M. S. Moonen et al, ed.), pp. 195–218, Kluwer Academic Publishers, 1993.

[31] P. V. Dooren, "The generalized eigenstructure problem in linear system theory," *IEEE Trans. Autom. Contr.*, vol. 26, no. 1, pp. 111–129, 1981.

[32] R. Fantacci and S. Nannicini, "Performance evaluation of a reservation TDMA protocol for voice/data transmission in microcellular systems," *IEEE JSAC*, vol. 18, no. 3, pp. 2404–2416, 2000.