Matrix-Geometric Solutions of M/G/1-Type Markov Chains: A Unifying Generalized State-Space Approach

Nail Akar, Member, IEEE, Nihat Cem Oğuz, and Khosrow Sohraby, Senior Member, IEEE

Abstract—In this paper, we present an algorithmic approach to find the stationary probability distribution of M/G/1-type Markov chains which arise frequently in performance analysis of computer and communication networks. The approach unifies finite- and infinite-level Markov chains of this type through a generalized state-space representation for the probability generating function of the stationary solution. When the underlying probability generating matrices are rational, the solution vector for level $k$, $x_k$, is shown to be in the matrix-geometric form $x_k = gF^kH$, $k \geq 0$, for the infinite-level case, whereas it takes the modified form $x_k = g_1F_1H_1 + g_2F_2H_2$, $0 \leq k < K$, for the finite-level case. The matrix parameters in the above two expressions can be obtained by decomposing the generalized system into forward and backward subsystems, 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 advanced numerical linear algebra techniques including matrix-sign function iterations with quadratic convergence rates or ordered generalized Schur decomposition. The simplicity of the matrix-geometric form of the solution allows one to obtain various performance measures of interest easily, e.g., overflow probabilities and the moments of the level distribution, which is a significant advantage over conventional recursive methods.

Index Terms—ATM multiplexer analysis, generalized difference equations, generalized invariant subspaces, generalized Schur decomposition, matrix-sign function, M/G/1-type Markov chains, polynomial matrix fractional descriptions.

I. INTRODUCTION

In this paper, we study Markov chains of M/G/1 type with finite or infinite number of levels. The state space of an infinite-level (or simply infinite) M/G/1-type Markov chain consists of integer pairs $(i,j)$ where $i$, the level of the chain, takes on an infinite set of values $(i \geq 0)$, and $j$, the phase of the chain, takes on a finite set of values $(0 \leq j < m)$. The transition probability matrix of this chain has the block-partitioned form [29]

$$P = \begin{bmatrix}
B_0 & B_1 & B_2 & B_3 & \cdots \\
A_0 & A_1 & A_2 & A_3 & \cdots \\
& \ddots & \ddots & \ddots & \ddots \\
& & \ddots & \ddots & \ddots \\
& & & \ddots & \ddots \\
& & & & \ddots 
\end{bmatrix}$$

where $A_i$ and $B_i$, $i \geq 0$, are $m \times m$ matrices. Assuming that $P$ is irreducible and positive recurrent, we find the stationary probability vector $x = [x_0 \ x_1 \ \cdots]$ which satisfies

$$x = xP, \quad xe = 1$$

where $x_i$, $i \geq 0$, is $1 \times m$, and $e$ is an infinite column vector of ones.

When the number of levels is finite, say $K+1$, the transition probability matrix takes the block upper-Hessenberg form

$$P = \begin{bmatrix}
B_0 & B_1 & B_2 & \cdots & B_{K-1} & B_{K-1} \\
A_0 & A_1 & A_2 & \cdots & A_{K-1} & A_{K-1} \\
& \ddots & \ddots & \ddots & \ddots & \ddots \\
& & \ddots & \ddots & \ddots & \ddots \\
& & & \ddots & \ddots & \ddots \\
& & & & \ddots & \ddots 
\end{bmatrix} \begin{bmatrix}
A_0 & A_1 & A_2 & \cdots & A_{K-2} & A_{K-2} \\
& \ddots & \ddots & \ddots & \ddots & \ddots \\
& & \ddots & \ddots & \ddots & \ddots \\
& & & \ddots & \ddots & \ddots \\
& & & & \ddots & \ddots \\
& & & & & \ddots
\end{bmatrix}$$

where $A_i$, $0 \leq i < K$, and $B_{K-1}$ are $m \times m$, and constitute the boundary at level $K$. We then study the solution vector $x = [x_0 \ x_1 \ \cdots x_K]$ which satisfies (2), with $e$ this time being a column vector of ones of length $m(K+1)$. Throughout the paper, $e$ will denote a column vector of ones of suitable size.

Both infinite and finite M/G/1-type Markov chains arise frequently in the performance analysis of ATM (asynchronous transfer mode) networks. In an ATM network, the basic unit of information is a fixed-length cell and the sharing of common network resources (bandwidth, buffers, etc.) among virtual connections is made on a statistical multiplexing basis. Statistical quality of service guarantees are integral to an ATM network, necessitating accurate traffic and performance analysis tools to determine the cell loss rate, cell delay, and cell delay variation in an ATM node (switch, multiplexer, etc.). This is, in general, difficult due to multiplexing of typically a large number of connections and burstiness of individual cell streams at possibly different time scales. One popular approach is to approximate such complex nonrenewal input processes by analytically tractable Markovian models either at
the connection [34], [10] or at the link (physical or logical) level [18], [19], [27]. Markovian arrival process (MAP) [26] and batch Markovian arrival process (BMAP) [24] have been used extensively in ATM performance evaluation in continuous time. For example, the well-known Markov-modulated Poisson process (MMPP) is a sub-case of MAP [18]. Various other Markovian traffic models, including Markov-modulated Bernoulli process (MMBP) or its generalization discrete batch Markovian arrival process (DBMAP), are also used to model the correlated nature of ATM traffic streams in discrete time [34], [35]. We note that the DBMAP model allows batch arrivals in one cell time [30] and is suitable for modeling aggregate traffic. Such processes in continuous or discrete time when fed into a single-server queue are known to give rise to M/G/1-type Markov chains [29], where the phase of the chain represents the state of the underlying Markovian model that governs (or modulates) the arrivals, and the level of the chain represents the queue length.

While the infinite M/G/1 chain seems to lack physical justification due to limited storage capacities in ATM nodes, it usually serves as an efficient approximation to the case of a finite but large number of levels. Infinite M/G/1 models have especially been used in the analysis of asymptotic queue length behavior which is closely linked with effective bandwidth computations for call admission control in ATM networks [34], [35], [10]. Assuming an output buffer capacity of $K$ cells, the infinite M/G/1 chain can be truncated at level $K$ to obtain a finite M/G/1 chain of the form (3). Assuming no particular buffer management scheme in effect, this truncation is generally done by writing the boundary at level $K$ as

$$\overline{A}_i \triangleq \sum_{j=i+1}^{\infty} A_j, \quad 0 \leq i < K,$$

and

$$\overline{B}_{K-1} \triangleq \sum_{j=1}^{K-1} B_j.$$  \hspace{1cm} (4)

On the other hand, the boundary behavior at level 0 is generally captured by defining

$$B_0 \triangleq A_0 + A_1 \quad \text{and} \quad B_i \triangleq A_{i+1}, \quad 1 \leq i < K,$$  \hspace{1cm} (5)

if the node can forward an incoming cell without any delay. In the case that an incoming cell is subject to one cell-time delay even when the buffer is empty, one has

$$B_i \triangleq A_i, \quad 0 \leq i < K.$$  \hspace{1cm} (6)

Other possibilities for the boundary sequence $\{B_i\}$ also exist [29].

For the solution of infinite and finite M/G/1 chains, we take an algebraic approach which is entirely different than the conventional methods. This technique unifies finite and infinite models, and consists of obtaining a generalized state-space representation of the probability generating function of the stationary solution. The generalized system is then decomposed into its forward and backward subsystems which in turn result in a matrix-geometric solution for infinite M/G/1 chains

$$x_{k+1} = g_1 F_1^k H_1 + g_2 F_2^{K-k-1} H_2, \quad 0 \leq k \leq K - 1.$$  \hspace{1cm} (8)

Using the same generalized system and its forward–backward decomposition, we further show that the solution vector for level $k$ for finite M/G/1 chains is expressible as

$$x_{k+N} = g_1 F_1^k H, \quad k \geq 0.$$  \hspace{1cm} (10)

Using invariant subspace computations in the solution of infinite M/G/1- and G/M/1-type Markov chains has been proposed before in [2]. In [3], this approach has been refined in the generalized state-space framework to eliminate recursive computations traditionally required to find the stationary probabilities of an infinite M/G/1 chain. The current paper is an extended version of [3], and presents the unifying generalized state-space approach for the stationary solution of infinite/finite and single-/multiple-boundary M/G/1-type chains arising in the performance analysis of computer and communication systems. Furthermore, we introduce the ordered generalized Schur decomposition in this paper as the numerical engine that implements the generalized state-space approach, as well as the matrix-sign function method which was studied extensively in [2] and [3]. Based on the numerical experiments we have performed, the former method appears to outperform the serial version of the latter in terms of execution times and accuracy. However, we note that the matrix-sign function iterations are parallelizable at the algorithm level, and significant execution
time reductions can potentially be attained by means of parallel implementations [15].

The paper is organized as follows. In Section II, we introduce the generalized state-space approach for solving infinite M/G/1-type Markov chains. Section III describes the two algorithms that implement this approach; one algorithm is based on the matrix-sign function, and the other on the ordered generalized Schur decomposition. Then Sections IV and V extend the formulation to also cover finite M/G/1 chains and the noncanonical case of multiple boundary levels, respectively. Numerical examples are provided in Section VI to demonstrate the accuracy and efficiency of the approach.

II. INFINITE M/G/1-TYPE MARKOV CHAINS

For the mathematical formulation of the problem, we first need to define the two $d$-domain probability generating matrices

$$A(d) = \sum_{i=0}^{\infty} A_i d^{-i} \quad \text{and} \quad B(d) = \sum_{i=0}^{\infty} B_i d^{-i}$$

(11)

which are related to their $z$-domain counterparts as $A(d) = A(z)|_{z=d^{-1}}$ and $B(d) = B(z)|_{z=d^{-1}}$, respectively. We then make the assumption that the transform matrices $A(d)$ and $B(d)$ are rational, i.e., the entries of $A(d)$ and $B(d)$ are rational functions of $d$. This assumption is not restrictive due to the following.

1) Most of the probability models of M/G/1 type encountered in computer and communication systems naturally give rise to rational transform matrices.

2) When the transform matrices are general, conventional methods make use of truncation to replace the infinite matrix sequences $\{A_i\}$ and $\{B_i\}$ appropriately by finite sequences for computational tractability, and this amounts to approximating the transform matrices by rational matrices. Our model avoids truncation by taking advantage of the rational structure of $A(d)$ and $B(d)$, and thus generalizes the existing models.

3) It is, in general, advantageous to use rational functions to approximate general (possibly irrational) probability generating matrices. See, for example, [1] in which the deterministic service time in a MAP/D/1/K queue is approximated by Padé approximations in transform domain to successfully estimate the cell loss rates in an ATM multiplexer.

Under the above assumption, one can express $A(d)$ and $B(d)$ as a stable right coprime polynomial matrix fraction

$$\begin{bmatrix} A(d) \\ B(d) \end{bmatrix} = \begin{bmatrix} P(d) \\ R(d) \end{bmatrix} Q^{-1}(d)$$

(12)

where $P(d), R(d),$ and $Q(d)$ are $m \times m$ polynomial matrices of $d$ [21], with polynomial degrees $p, r,$ and $q$, respectively. We note that $A(d)$ and $B(d)$ are proper rational matrices and, hence, the relations $p \leq q$ and $r \leq q$ hold. Moreover, stable right coprimeness is imposed on the fraction to avoid redundancies in the matrix-fractional description, and implies that all the roots of $\text{det}[Q(d)]$ lie in the open unit disk.$^1$

In the following, we first discuss how the fractional description of (12) can be obtained generically, and provide some teletraffic examples naturally yielding such descriptions. Then, after outlining a slightly modified version of the traditional iterative solution methods, we introduce the generalized state-space approach of this paper.

A. Obtaining Stable Right Coprime Fractions

Consider a stable proper transform matrix, $A(d)$, of size $p \times q$. One can generically obtain a stable right coprime fraction of $A(d)$ as follows. Let $l_i(d)$, $1 \leq i \leq q$, be the least common multiple of all the denominators of the $i$th column entries of $A(d)$. Define $Q(d) \triangleq \text{diag}\{l_i(d)\}$ and $P(d) \triangleq A(d)Q(d)$. It is then clear that the fraction $A(d) = P(d)Q^{-1}(d)$ is a stable right coprime polynomial fraction. As an example, consider a two-state Markov-modulated geometric source. Let $t_{ij}$ be the state transition probabilities of the modulating chain, and $r_{ij}$ be the geometric rate parameter associated with the transitions. Then, the entries $a_{ij}(d)$ of $A(d)$ are given as

$$a_{ij}(d) = t_{ij}(1-r_{ij})d \quad \text{for} \quad i, j = 1, 2.$$ 

If we assume that the $r_{ij}$’s are all different, then the entries $q_{ij}(d)$ and $p_{ij}(d)$ of $Q(d)$ and $P(d)$ are found, respectively, as

$$q_{ij}(d) = (d-r_{1i})(d-r_{2i}) \quad \text{for} \quad i = 1, 2$$

and

$$p_{ij}(d) = t_{ij}(1-r_{ij})d(d-r_{kk}) \quad \text{for} \quad i, j = 1, 2, \quad \text{and} \quad k = 3-i.$$ 

For a wide variety of teletraffic models, however, one may not need to take this generic approach as the fractions can directly be obtained from the problem description. Below, we give three popular models from the teletraffic literature, and find a stable right coprime pair of matrices $P(d)$ and $Q(d)$ for the probability generating matrix $A(d)$. We also note that the fraction for $B(d)$ can generally be obtained through that for $A(d)$ easily as in (5) or (6).

1) Quasi-Birth-and-Death Processes [36], [28], [23]: If, in the structure of $P$ in (1), state transitions are restricted to take place between adjacent levels only, the resulting model is called a quasi-birth-and-death process (QBD). That is, for QBD chains, $A_k = 0$ for $k>2$, and

$$A(d) = A_0 + A_1d^{-1} + A_2d^{-2}.$$ 

The choice of

$$P(d) = A_2 + A_1d + A_0d^2 \quad \text{and} \quad Q(d) = d^2I$$

gives a stable right coprime fraction for $A(d)$. Note that this formulation appropriately extend to obtain fractions for the more general case in which $A_k = 0$ for $k>N$ and $2<N<\infty$.

$^1$Our recent experiments indicate that the generalized state-space method works even when there are redundancies, that is, even when coprimeness is not sought. See Appendix I for a brief mathematical overview of stability and right coprimeness concerning polynomial matrices and polynomial matrix fractions.
2) Single-Server Discrete-Time Queue with Modulated Arrivals: Consider a discrete-time queue with a single server and with arrivals modulated by a finite-state discrete-time Markov chain [34]. Assume that the modulating chain has \( m \) states with transitions occurring at slot boundaries. Let \( t_{ij} \), \( 0 \leq i, j \leq m - 1 \), denote the transition probabilities. Also let \( h_{ik} \) denote the probability of \( k \) arrivals when the modulating chain resides in state \( i \). Assume that

\[
h_k(d) = \sum_{k=0}^{\infty} h_{ik} d^k
\]

is a rational function of \( d \) (for example, a discrete phase-type distribution). Let the queue length and state of the modulating chain be associated with our level and phase definitions. If we write \( h_i(d) = p_i(d)/q_i(d) \), \( 0 \leq i \leq m - 1 \), then \( A(d) \) can be written as

\[
A(d) = Q_0 \hat{P}(d) Q^{-1}(d)
\]

where \( Q_0 = \text{diag}\{p_i(d)\} \), and \( Q(d) = \text{diag}\{q_i(d)\} \). Let \( \lambda \) be a polynomial fraction. For \( \lambda = [t_{ij}] \), \( \hat{P}(d) = \text{diag}\{\hat{p}_i(d)\} \), and \( \hat{Q}(d) = \text{diag}\{\hat{q}_i(d)\} \). Considering the embedded Markov renewal process at departure epochs, we obtain a Markov chain of M/G/1 type with

\[
A(d) = h(\lambda - R - \Delta d^{-1})
\]

which is a rational function of \( d \) [24]. The polynomial fractions of \( A(d) \) can directly be obtained as

\[
A(d) = d^w q(\lambda - R - \Delta d^{-1}) \quad \text{and} \quad \hat{A}(d) = d^w p(\lambda - R - \Delta d^{-1})
\]

where \( w \) is the degree of polynomial \( q(s) \).

B. Matrix-Analytic Method

We now outline an efficient iterative method for finding the stationary solution as in (2) of an infinite M/G/1-type Markov chain. This method is based on the matrix-analytic approach pioneered by Neuts [29] with a slight modification to take advantage of the rationality of \( A(d) \) (also see [25] for a similar approach for the BMAP/G/1 queue). In this method, the key is to find the unique minimal nonnegative solution \( G_\kappa \) of the nonlinear matrix equation

\[
G = \sum_{k=0}^{\infty} A_k G_k
\]

where \( G \) is a polynomial fraction for \( A(d^{-1}) \) or, equivalently, \( A(z) = \hat{Q}^{-1}(z) \hat{P}(z) \) in the \( z \)-domain (note that this is a left polynomial fraction as opposed to (12); also see [2]). Previous numerical experiments indicate that this iteration has a linear convergence rate [2]. It is shown in [33] that \( x_0 \) is equal to the stationary probability vector \( \kappa \) of the stochastic matrix \( \sum_{k=0}^{\infty} B_k G_k^\ast \) normalized as

\[
x_0 = \frac{\kappa}{n}, \\
n = 1 + \frac{\kappa}{1 - \rho} [\beta + [B(1) - I] (I - A(1) + e\pi)^{-1} \alpha]
\]

where

\[
\pi = \text{stationary probability vector of } A(1), \quad \alpha = -A(1)(1 - \rho), \quad \beta = -B(1)(1 - \rho), \quad \rho = \pi \alpha
\]

\( \pi \) is the stationary probability vector of \( A(1) \), and the traffic parameter (or utilization) \( \rho \) is less than unity. Once \( x_0 \) is found, the vectors \( x_k, k \geq 1 \), can be obtained recursively by [31]

\[
x_k = \left[ x_0 \hat{B}_k + \sum_{i=1}^{k-1} x_{i+1} \hat{A}_k \right] (I - \hat{A}_1)^{-1}
\]

where

\[
\hat{A}_k = \sum_{i=1}^{k} A_i G_k^i, \quad \hat{B}_k = \sum_{i=1}^{k} B_i G_k^i.
\]

Note that computation of \( x_k, k > 0 \), as in (18), requires truncation of the infinite matrix sequences \( \{A_i\} \) and \( \{B_k\} \). Due to the low linear convergence rates of the successive substitution iterations to find \( G_\kappa \) and depending on the truncation index required to attain a certain accuracy, the matrix-analytic approach may, in general, incur considerable execution times and storage requirements especially under heavy traffic conditions. The generalized state-space approach differs significantly from the matrix-analytic approach, and is presented in Subsection D after the following brief overview of invariant subspaces.

C. Overview of Invariant Subspaces

Here we give a brief description of ordinary and generalized invariant subspaces based mainly on [12] and [16]. We use the following notation. Uppercase is used for matrices and lowercase for vectors, both being defined over the field of real numbers \( \mathbb{R} \). \( \lambda(A) \) denotes the spectrum, i.e., the set of eigenvalues, of \( A \). A constant, polynomial, or rational matrix is called regular when it is square and has a nonzero determinant. Otherwise, it is called singular. A subspace \( S \) is a subset of \( \mathbb{R}^m \) that is closed under the operations of addition and scalar multiplication. If \( A \) denotes the image (or the column space) of \( A \), \( AS \) is the image of \( S \) under \( A \). An invariant subspace of \( A \) satisfies \( AS \subseteq S \) where \( \subseteq \) denotes inclusion. \( S + T \) and \( ST \) are the sum and direct sum, respectively, of the subspaces \( S \) and \( T \). Let \( S \cap T = \mathbb{R}^m \) and assume that \( S \) and \( T \) are invariant subspaces of a square matrix \( A \) of size \( m \). Then, \( S = \text{Im} S \) and \( T = \text{Im} T \) and defined by \( U = [S \ T] \) satisfy

\[
U^{-1} AU = \begin{bmatrix}
A_{11} & 0 \\
0 & A_{22}
\end{bmatrix}
\]

If \( \lambda(A_{11}) \) (\( \lambda(A_{22}) \)) lies in the closed right-half (open left-half) plane, then \( S \) (\( T \)) is said to be the right (left) invariant
When \( \lambda(A_{11}) \) (\( \lambda(A_{22}) \)) lies outside (in) the open unit disk, then \( S \) is called the unstable (stable) invariant subspace of \( A \). This notation is inherited from the stability of difference systems.

Let us now assume a regular matrix pencil \( \lambda E - A \), which is a polynomial matrix (in the indeterminate \( \lambda \)) of degree one. The \textit{generalized eigenvalue problem} for the matrices \( A \) and \( E \) of size \( m \) is equivalent to finding the scalars \( \lambda \) for which the equation \( A x = \lambda E x \) has solutions \( x \neq 0 \). Such scalars \( \lambda \) are called generalized eigenvalues. A solution \( x \neq 0 \) corresponding to an eigenvalue \( \lambda \) is called a generalized eigenvector. A generalized eigenvalue satisfies the relation

\[
\lambda \in \lambda(E, A) := \{ \mu \in \mathbb{C} | \det(\mu E - A) = 0 \}
\]

where \( \mathbb{C} \) is the field of complex numbers, and \( \lambda(E, A) \) denotes the generalized spectrum of the matrix pair \( (E, A) \). Any subspace \( S \) satisfying

\[
T = ES + AS, \quad \dim(S) = \dim(T)
\]

is called a \textit{generalized invariant subspace} (or a deflating subspace) of the pencil \( \lambda E - A \). When \( E = I \), we indeed have an ordinary invariant subspace.

Let \( S \) and \( S_c \) be two complementary deflating subspaces of the pencil \( \lambda E - A \), i.e., \( S \oplus S_c = \mathbb{R}^n \). Define \( T = ES + AS \) and \( T_c = ES_c + AS_c \). It is shown in [11] that these two subspaces are also complementary. Let \( S = \text{Im} \ S, \quad T = \text{Im} \ T, \quad S_c = \text{Im} \ S_c, \) and \( T_c = \text{Im} \ T_c \). Then there exists a decomposition

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

where

\[
U = \begin{bmatrix} T \\ T_c \end{bmatrix}, \quad V = \begin{bmatrix} S \\ S_c \end{bmatrix}
\]

If \( \lambda(E_{11}, A_{11}) \) (\( \lambda(E_{22}, A_{22}) \)) lies in the closed right-half (open left-half) plane, then \( S \) (\( S_c \)) 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 \( S \) (\( S_c \)) is called the unstable (stable) deflating subspace of the matrix pencil \( \lambda E - A \).

\section*{D. Generalized State-Space Approach}

Now consider the Markov chain with the transition probability matrix given in (1). Define the \( d \)-transform of the sequence \( x_k \) \( k \geq 0 \), as

\[
x(d) \overset{def}{=} \sum_{k=0}^{\infty} x_k d^{-k}.
\]

It is easy to show by (1) that

\[
x(d)[I - dA(d)] = x_0[B(d) - dA(d)].
\]

Also define the sequence

\[
y_k \overset{def}{=} x_{k+1}, \quad k \geq 0
\]

and let \( y(d) \) be its \( d \)-transform. It is not difficult to show that

\[
y(d) = d[x(d) - x_0] = x_0N(d)D^{-1}(d)
\]

where

\[
N(d) = d[N(d) - Q(d)] = N_1 + N_2 d + \cdots + N_f d^f - d \quad \text{with} \quad f = q + 1,
\]

\[
D(d) = Q(d) - dP(d)
\]

\[
= D_0 + D_1 d + D_2 d^2 + \cdots + D_f d^f
\]

Here, \( f \) is called the \textit{degree parameter} of the Markov chain and will play a key role in our approach.

Given the polynomial fraction (22), one can find a generalized state-space realization [12] for \( y_k \) (see Appendix II for a proof)

\[
z_{k+1}E = z_kA, \quad z_0D = x_0N, \quad y_k = z_kC, \quad k \geq 0
\]

where

\[
A = \begin{bmatrix} 0 & 0 \cdots & -D_0 \\ I & 0 \cdots & -D_1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & I - D_{f-1} \end{bmatrix}, \quad C = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad E = \begin{bmatrix} I \\ \vdots \\ I \\ D_f \end{bmatrix}, \quad N = [N_f, N_{f-1}, N_{f-2}, \cdots, N_1]
\]

and

\[
D = \begin{bmatrix} D_f & D_{f-1} & \cdots & D_1 \\ D_f & D_{f-1} & \cdots & D_2 \\ \vdots & \vdots & \ddots & \vdots \\ D_f & D_{f-1} & \cdots & D_{f-1} \end{bmatrix}
\]

Here, \( z_k \) is called the \textit{descriptor} (or the semistate) which reduces to the definition of state when \( E \) is nonsingular [21]. The possible singularity of \( E \) plays a significant role in the problem formulation. Also note that \( z_k \) is of size \( 1 \times mf \), and the matrices \( E \) and \( A \) are of size \( mf \times mf \).

\textbf{Remark:} When the first \( \nu \) coefficients of \( D(d) \) are zero, i.e., \( D_i = 0 \) for \( i = 0, 1, 2, \cdots, \nu - 1 \), a reduced-order generalized state-space representation can be obtained. That is, the problem dimension can be reduced to \( m(f - \nu) \), resulting in an effective degree parameter of \( f' = f - \nu \). In the case of a QBD chain, for example, it turns out that \( D_0 = 0 \). Therefore, the effective degree parameter can be made \( f' = 2 \) as opposed to \( f = 3 \), as (25) suggests. See [3] for details.

We now need to find \( x_0 \) so that none of the unstable modes of the matrix pair \( (E, A) \) is excited, i.e., \( z_k \) of (26) remains finite for all \( k \). The matrix pencil \( \lambda E - A \) has one singularity at \( d = 1 \), say, \( m_0 \) singularities (including the one at \( d = 1 \)) outside the open unit disk, and \( m_s \) singularities in the open unit disk. Note that \( m_0 + m_s \) yields the dimension \( mf \) of the generalized system given in (26). Let \( V_1 \) and \( V_2 \) be the unstable and stable deflating subspaces of the pencil \( \lambda E - A \), respectively. Let \( V_1 = \text{Im} V_1 \) and \( V_2 = \text{Im} V_2 \) for some matrices \( V_1 \) and \( V_2 \) of sizes \( mf \times m_0 \) and \( mf \times m_s \), respectively.
respectively. Also let \( \mathcal{U}_1 := EV_1 + AV_1 = \Im U_1 \) and \( \mathcal{U}_2 := EV_2 + AV_2 = \Im U_2 \) for some matrices \( U_1 \) and \( U_2 \) of sizes \( m_f \times m_a \) and \( m_f \times m_a \), respectively. Define

\[
U \triangleq \begin{bmatrix} U_1 & U_2 \end{bmatrix} \quad \text{and} \quad V \triangleq \begin{bmatrix} V_1 & V_2 \end{bmatrix}.
\]  

(30)

Then, from Section II-C, we have

\[
U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix} \quad \text{and} \quad U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}
\]

(31)

and \( \lambda(E_{11}, A_{11}) \) and \( \lambda(E_{22}, A_{22}) \) lie outside and in the open unit disk, respectively. Defining

\[
\begin{bmatrix} u_k & v_k \end{bmatrix} \triangleq z_k \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \quad k \geq 0
\]

and postmultiplying the generalized state-space model (26) by \( V \), we have two uncoupled generalized difference equations for \( u_k \) and \( v_k \)

\[
u_{k+1}E_{11} = u_k A_{11}, \quad k \geq 0
\]

(32)

\[
u_{k+1}E_{22} = v_k A_{22}, \quad k \geq 0
\]

(33)

In order for \( z_k \) not to diverge as \( k \to \infty \), \( u_0 \) must be the zero vector

\[
u_0 = z_0 U_1 = 0.
\]

(34)

Moreover, since \( \lambda(E_{22}, A_{22}) \) lie in the open unit disk, \( E_{22} \) is non-singular implying

\[
u_k = \nu_0 F^k, \quad k \geq 0
\]

(35)

where the \( m_a \times m_a \) matrix \( F \) is found as

\[
F = A_{22} E_{22}^{-1}.
\]

(36)

Let us now partition \( U^{-1} \) as

\[
U^{-1} = \begin{bmatrix} L_1 & L_2 \end{bmatrix}
\]

(37)

where \( L_1 \) and \( L_2 \) has \( m_a \) and \( m_s \) rows, respectively. Then

\[
x_{k+1} = \gamma_k = z_k C = (u_k L_1 + v_k L_2) C
\]

\[
= \nu_0 U_2 F^k + v_k L_2 C, \quad k \geq 0
\]

(38)

The only unknowns that remain to complete the solution are \( x_0 \) and the initial value \( z_0 \) which, by (26) and (34), satisfy

\[
[x_0 \ z_0] Z = \begin{bmatrix} F & 0 \\ -N & D \end{bmatrix} U_k = 0.
\]

(39)

Note that \( Z \) is \( m \times (f+1) \times m \). Furthermore, the sum of the probabilities \( x_k \) is unity which gives a normalizing equation in terms of \( x_0 \) and \( z_0 \)

\[
\sum_{k=0}^{\infty} x_k C = x_0 C + \sum_{k=0}^{\infty} \gamma_k C
\]

\[
= x_0 C + \nu_0 U_2 (I - F)^{-1} H C = 1.
\]

(40)

The concatenated vector \( [x_0 \ \ z_0] \) is the unique solution to the two equations (39) and (40), which when computed leads to the simple matrix-geometric solution for the stationary probability vector \( x_k \) for level \( k + 1 \)

\[
x_{k+1} = gF^k H, \quad k \geq 0.
\]

(41)

This simple and compact solution form makes it easier to write certain performance measures of interest. For example, the \( r \)th factorial moment, \( L(r) = \sum_{r=0}^{\infty} t^{r-1} \), of the level distribution is readily expressible in closed form as (also see [28])

\[
L(r) = \sum_{r=0}^{\infty} \frac{k!}{(r-1)!} x_k e = r! gF^{n-1} (I - F)^{-1} H C.
\]

(42)

The overflow probabilities, say \( P_B \), are also easy to write

\[
P_B = \sum_{k=B+1}^{\infty} x_k e = g F^B (I - F)^{-1} H C.
\]

(43)

In addition, the queue length distribution is known to exhibit a geometric decay as \( \alpha \) for sufficiently large \( k \) [34], [35], [10]. The form (41) of the solution indicates that the decay rate \( \alpha \) is the dominant eigenvalue of matrix \( F \), which can be computed efficiently by the power method [17, Section 7.3.1]. More importantly, (41) allows computation of the coefficient \( \alpha \) as well. Assuming that \( \Lambda = T^{-1} F T \) is the Jordan form of \( F \), the stationary probability of level \( k + 1 \) can be written as \( x_{k+1} = gTA^k T^{-1} H C \), as \( k \) goes to infinity, this expression reduces to \( x_{k+1} = gT_L \beta^k T_R H C \), where \( T_L \) and \( T_R \) are left- and right-eigenvectors of \( F \) associated with the dominant eigenvalue \( \beta \). Once \( \beta \) is computed, \( T_L \) and \( T_R \) can be found by solving two linear equations, and then the coefficient follows as \( \alpha = gT_L T_R H C \).

This concludes the discussion of the existence of matrix-geometric solutions for infinite M/G/1-type Markov chains when the transform matrices \( A(d) \) and \( B(d) \) are rational functions of \( d \). Two computational algorithms, one based on the matrix-sign function and the other on ordered Schur decomposition, for finding the matrices \( U \) and \( V \) of decomposition (31) are presented in the next section.

III. ALGORITHMS FOR INVARIANT SUBSPACE COMPUTATIONS

The (generalized) invariant subspace computation (left or right, stable or unstable) is a well-known problem of numerical linear algebra [12], [16]. To name a few, (generalized) Schur decomposition methods [17], [20], inverse-free spectral divide-and-conquer methods [6], (generalized) matrix-sign function iterations [14] have been proposed to compute bases for these subspaces which arise for a wide variety of problems in applied mathematics. All of the above approaches can be used to find bases for the stable and unstable deflating subspaces of the matrix pencil \( \lambda E - A \), which is an essential task in the generalized state-space method for solving M/G/1-type Markov chains. Here we present two algorithms. One is based on the ordinary matrix-sign function [32], and the other on the generalized Schur decomposition with ordering. The former algorithm employs certain properties of the matrices \( E \) and \( A \) akin to M/G/1-type models, whereas the latter is quite generic. We also provide a summary of the overall method for infinite M/G/1 chains.
A. Matrix-Sign Function Approach

We first note that the stable (unstable) deflating subspace of the matrix pencil $\lambda E - A$ is equal to the left (right) deflating subspace of the pencil $\lambda L - M$, where the two matrices $L$ and $M$ are defined as

$$ L \triangleq E + A, \quad M \triangleq A - E. $$

For a proof, we refer the reader to [14]. With this transformation, the generalized eigenvalues of the pencil $\lambda E - A$ in (outside) the open unit disk are moved to the open left-half (closed right-half) plane. So there is one generalized eigenvalue of $\lambda L - M$ on the imaginary axis, which is at the origin.

One can also show that $L$ is regular by observing that the pencil $\lambda E - A$ does not have any generalized eigenvalue on the unit circle except one at $\lambda = 1$. In particular, $\lambda E - A$ does not have any generalized eigenvalue at $\lambda = -1$ which clearly shows that $L$ is nonsingular. Let $W = L^{-1}M$. It is not difficult to show that the left (right) invariant subspace of $W$ is also equal to the left (right) deflating subspace of the pencil $\lambda L - M$. Furthermore, $W$ has one eigenvalue on the imaginary axis, which is at the origin. Then let $\gamma$ and $\mu$ be left and right eigenvectors of $W$ corresponding to the eigenvalue at the origin, i.e.,

$$ \gamma W = 0, \quad W \mu = 0. \tag{44} $$

Then, the matrix $W_e$ defined as

$$ W_e \triangleq W + \frac{\mu \gamma}{\gamma \mu} \tag{45} $$

is free of imaginary-axis eigenvalues, and the left (right) invariant subspace of $W_e$ is equal to the left (right) invariant subspace of $W$. It is not difficult to show that the vectors $\gamma$ and $\mu$ defined as

$$ \gamma \triangleq [\pi \pi \cdots \pi]L, \quad \mu \triangleq \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{f-1} \end{bmatrix} \tag{46} $$

where $\pi$ is the stationary probability vector of $A(1)$, i.e.,

$$ \pi A(1) = \pi, \pi e = 1 $$

and

$$ \mu_0 = -D_0Q^{-1}(1)c, $$

$$ \mu_i = \mu_{i-1} - D_0Q^{-1}(1)c, \quad 1 \leq i \leq f-2, $$

$$ \mu_{f-1} = Q^{-1}(1)c $$

satisfy (44).

One can now use matrix-sign function iterations on $W_e$ [2] to find bases for the unstable and stable deflating subspaces of the pencil $\lambda E - A$, leading to construction of the matrices $U$ and $V$ defined as in (30). We now outline this approach.

We refer the reader to [2] for the definition of the matrix-sign function. The basic matrix-sign function algorithm for an $m \times m$ matrix $M$ with no imaginary axis eigenvalues is (see [7] and [14])

$$ Z_0 = M, \quad Z_{k+1} = \frac{1}{2 \gamma_k} (Z_k + \gamma_k^2 Z_k^{-1}), $$

$$ \gamma_k = |\det(Z_k)|^{1/m}. \tag{47} $$

Then

$$ \lim_{k \to \infty} Z_k = Z = sgn(M) $$

where $sgn(M)$ denotes the matrix sign of $M$, and convergence is quadratic. The stopping criterion we use is the one proposed in [5]

$$ ||Z_{k+1} - Z_k||_1 < \epsilon ||Z_k||_1. \tag{48} $$

The most important property of matrix sign is that $\text{Im}(Z - I) (\text{Im}(Z + I))$ is equal to the left (right) invariant subspace of $M$ [32]. Then find

$$ S = sgn(W_e) \tag{49} $$

through the matrix-sign function iterations (47). Recall that there are $m_u$ eigenvalues of $W_e$ in the right-half plane, and $m_s$ eigenvalues in the left-half plane. Let the rank-revealing QR decomposition [8] of $S + I$ be

$$ S + I = Q_rR_r\Pi_r \tag{50} $$

where $R_r$ is upper triangular, $Q_r$ is orthogonal, and $\Pi_r$ is a permutation matrix. Suppose that $\Pi_r$ is chosen so that the rank deficiency of $S + I$ is exhibited in $R_r$ by a smaller lower-right block in norm of size $m_s \times m_s$. Then, let

$$ V_1 \triangleq \text{leading } m_u \text{ columns of } Q_r \tag{51} $$

which span $\text{Im}(S + I)$ or, equivalently, form an orthogonal basis for the left-invariant subspace of $W_e$ or the unstable deflating subspace of the pencil $\lambda E - A$. Similarly, a rank-revealing QR decomposition of $S - I$ yields

$$ S - I = Q_lR_l\Pi_l \tag{52} $$

and with a proper choice of permutation $\Pi_l$, we define

$$ V_2 \triangleq \text{leading } m_u \text{ columns of } Q_l \tag{53} $$

Following Section II-C, with two more rank-revealing QR decompositions

$$ [EV_1 \quad AV_1] = \hat{Q}_r\hat{R}_r\hat{\Pi}_r \tag{54} $$

$$ [EV_2 \quad AV_2] = \hat{Q}_l\hat{R}_l\hat{\Pi}_l \tag{55} $$

we define

$$ U_1 \triangleq \text{leading } m_u \text{ columns of } \hat{Q}_r \tag{56} $$

$$ U_2 \triangleq \text{leading } m_s \text{ columns of } \hat{Q}_l. \tag{57} $$

This concludes the discussion of using ordinary matrix-sign function to find the four key matrices $V_1, V_2, U_1$, and $U_2$ that are used to decompose the generalized system (26) into its forward and backward subsystems through (31).
TABLE I

SUMMARY OF THE OVERALL NUMERICAL ALGORITHM FOR INFINITE M/G/1-TYPE MARKOV CHAINS

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( m = 16, f = 8 )</th>
<th>( m = 32, f = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>1.9e-15</td>
<td>2.5e-15</td>
</tr>
<tr>
<td>0.9</td>
<td>1.3e-14</td>
<td>6.6e-15</td>
</tr>
<tr>
<td>0.999</td>
<td>6.9e-11</td>
<td>9.6e-12</td>
</tr>
<tr>
<td>1.001</td>
<td>1.1e-10</td>
<td>2.9e-11</td>
</tr>
<tr>
<td>1.1</td>
<td>1.3e-14</td>
<td>5.6e-15</td>
</tr>
<tr>
<td>1.4</td>
<td>1.9e-15</td>
<td>1.7e-15</td>
</tr>
<tr>
<td>( T_{1}^{[1]} )</td>
<td>7.166</td>
<td>9.583</td>
</tr>
<tr>
<td>( T_{1}^{[2]} )</td>
<td>7.133</td>
<td>9.366</td>
</tr>
<tr>
<td>( T_{1}^{[3]} )</td>
<td>5.016</td>
<td>4.566</td>
</tr>
</tbody>
</table>

B. Generalized Schur Decomposition Approach

One other approach to find the stable (unstable) deflating subspaces that give rise to decomposition \((31)\) is to use generalized Schur decomposition with ordering \([17], [20]\). Given the two matrices \( E \) and \( A \) one can employ the generalized Schur decomposition method with ordering \([20]\) to compute the two orthonormal matrices \( \Theta \) and \( \Psi \) satisfying

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

such that:

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

Given the above decomposition, we next solve the generalized Sylvester equations \([20]\)

\[
A_{11} Y = X A_{22} = A_{12} \quad \text{and} \quad E_{11} Y = X E_{22} = E_{12}
\]

for the two matrices \( X \) and \( Y \). Finally, defining

\[
U \triangleq \Theta \begin{bmatrix} I & -X \\ 0 & I \end{bmatrix} \quad \text{and} \quad V \triangleq \Psi \begin{bmatrix} I & -Y \\ 0 & I \end{bmatrix}
\]

one obtains the decomposition \((31)\), i.e., eliminates the upper-diagonal blocks \( E_{12} \) and \( A_{12} \) in \((58)\). Here we note that elimination of these blocks is not necessary in the solution of infinite M/G/1 chains. However, for the case of finite M/G/1 chains that will be discussed in the next section, those blocks have to be eliminated.

In Table I, we provide an algorithmic description of the generalized state-space approach for infinite M/G/1 chains based on either the matrix-sign function or the generalized Schur decomposition with ordering. The algorithm assumes that the right polynomial fractions of \((12)\) are still valid for \( 0 \leq k \leq K - 1 \)

\[
z_{k+1} E = z_k A, \quad z_0 D = x_0 N,
\]

\[
y_k = z_k C, \quad 0 \leq k \leq K - 1.
\]

Using the same decomposition \((31)\) as in infinite M/G/1 chains, we have

\[
y_{k+1} E_{11} = y_k A_{11}, \quad 0 \leq k < K - 1
\]

or, equivalently

\[
y_k = y_{k+1} F_2, \quad 0 \leq k < K - 1
\]

where

\[
F_2 = E_{11} A_{11}^{-1}.
\]

The invertibility of \( A_{11} \) follows directly from the fact that the generalized eigenvalues of the pair \((E_{11}, A_{11})\) lie outside the open unit disk. Therefore, the matrix \( F_2 \) has all its eigenvalues in the closed unit disk. We call \((63)\) the backward subsystem of the generalized system \((61)\). It immediately follows from \((63)\) that

\[
y_k = u_{K-1} F_2^{K-k-1}, \quad 0 \leq k < K - 1.
\]

The main difference from the infinite M/G/1 formulation is that the unstable modes of the pair \((E, A)\) may be excited in finite M/G/1 chains and the vector \( y_0 \) is not necessarily the zero vector. On the other hand, the difference equations corresponding to the forward subsystem \((33)\) are still valid for \( 0 \leq k \leq K - 1 \), leading to

\[
y_k = u_k F_1^k, \quad 0 \leq k \leq K - 1
\]

where \( F_1 \triangleq F \). Then, the solution \( y_k, 0 \leq k \leq K - 1 \), of the finite M/G/1 chain of \((3)\) can be written in terms of \( u_{K-1} \) and \( y_0 \)

prove the existence of the modified matrix-geometric form given in \((8)\). What now remains is to find the three unknown
vectors \( x_0, g_1, \) and \( g_2 \). We have two equations to solve for these vectors, the first of which is
\[
g_1 L_2 \mathcal{D} + g_2 F_2^{K-1} L_1 \mathcal{D} = x_0 \mathcal{N}
\]
through straightforward substitution. The second equation derived from the balance equation at level \( K \) is
\[
x_0 \mathcal{B}_{K-1} + \sum_{i=0}^{K-1} y_i K_{i-1} = y_{K-1}
\]
and can be rewritten in terms of \( x_0, g_1, \) and \( g_2 \) by using (67) as
\[
x_0 \mathcal{B}_{K-1} + g_1 \left( \sum_{i=0}^{K-1} F_i^1 H_1 A_{K-i-1} \right) \Delta_1 + g_2 \left( \sum_{i=0}^{K-1} F_i^{K-1} H_2 A_{K-i-1} \right) \Delta_2 = g_1 F_1^{K-1} H_1 + g_2 H_2.
\]
Using (68) and (69), one can then find \( x_0, g_1, \) and \( g_2 \) uniquely by solving the equation
\[
\begin{bmatrix} x_0 & g_1 & g_2 \end{bmatrix} Z_e
\]
\[
= \begin{bmatrix} x_0 & g_1 & g_2 \end{bmatrix} \begin{bmatrix} -\mathcal{N} & L_2 \mathcal{D} \Delta_1 & L_1 \mathcal{D} \Delta_1 - F_1^{K-1} H_1 \\ \Delta_2 & L_1 \mathcal{D} \Delta_2 - H_2 \end{bmatrix} = 0
\]
\]
and normalizing the solution such that the stationary probabilities add up to unity, i.e., \( \Sigma_{k=0}^{\infty} x_k e = 1 \).

V. M/G/1 CHAINS WITH MULTIPLE BOUNDARIES

Based on [3], we outline below the algorithm for finding the matrix-geometric factors of the M/G/1 chain with multiple boundary levels. The proof is similar to that of the canonical M/G/1 chain and is omitted in this paper.

Consider the M/G/1-type Markov chain in (9) with \( N \) multiple boundary levels. First, define
\[
A(d) \triangleq \sum_{j=0}^{\infty} A_j d^{-j} \quad \text{and} \quad B_i(d) \triangleq \sum_{j=0}^{\infty} B_{ij} d^{-j}, \quad 0 \leq i < N.
\]
Then find a stable right coprime fraction as
\[
\begin{bmatrix} \hat{A}(d) \\ \hat{B}_0(d) \\ \hat{B}_1(d) \\ \vdots \\ \hat{B}_{N-1}(d) \end{bmatrix} = \begin{bmatrix} P(d) \\ Q_0(d) \\ R_0(d) \\ \vdots \\ R_{N-1}(d) \end{bmatrix} Q^{-1}(d).
\]

Let \( q \) be the degree of the polynomial matrix \( Q(d) \), and define
\[
D(d) = D_0 + D_1 d + D_2 d^2 + \cdots + D_q d^q = Q(d) - d^q P(d)
\]
and
\[
N(d) = N_1 d + N_2 d^2 + \cdots + N_q d^q
\]
\[
= \begin{bmatrix} R_0(d) d^N - d^N Q(d) \\ R_1(d) d^N - d^{N-1} Q(d) \\ \vdots \\ R_{N-1}(d) d^N - d Q(d) \end{bmatrix}.
\]

Note that \( N(d) \) and \( D(d) \) are polynomial matrices of degree \( f = q + N \). Define the matrices \( E, A, C, \mathcal{N}, \) and \( \mathcal{D} \) in the same way as in (27), (28), and (29) using the polynomials of (73) and (74) above. The rest of the algorithm is the same as that for the canonical M/G/1 chain. We first find the matrices \( U, V, \) and \( F \) as in (30) and (36), and partition \( U^{-1} \) as in (37). We then solve or
\[
[x_0 \ x_1 \ \cdots \ x_{N-1} \ x_0] \begin{bmatrix} -\mathcal{N} & 0 \\ \mathcal{D} & U_1 \end{bmatrix} = 0
\]
and normalize the solution so that
\[
\sum_{i=0}^{N-1} x_i e + x_0 U_2 (I - F)^{-1} H e = 1.
\]

Defining \( g \triangleq x_0 U_2 \) and \( H \triangleq L_2 C \) gives us the matrix-geometric solution
\[
x_{k+N} = g F^k H, \quad k \geq 0.
\]

VI. NUMERICAL EXAMPLES AND DISCUSSION

**Example 1**: We first consider an infinite M/G/1-type Markov chain obtained from the IPP_n/E_r/1 queueing model, where IPP_n stands for the superposition of \( n \) independent identical IPP’s (interrupted Poisson process) [13] and E_r stands for the \( r \)-stage Erlangian distribution. We refer to this chain through the following three parameters: 1) the number of phases \( m \), 2) the degree parameter \( f \) defined by (25), and 3) the traffic parameter (or utilization) \( \rho \) given in (17). Note that since i.i.d. IPP’s are considered, setting \( n = m - 1 \) results in an \( m \)-state Markovian model for the aggregate arrival process. For each IPP source, we fix the transition rates to the idle and active states as 3 and 1, respectively. Therefore, the arrival rate in the active state of each IPP is uniquely determined for any desired aggregate arrival rate \( \lambda \). We fix the mean service rate as \( \mu = 1 \), which implies that \( \rho = \lambda \). Since IPP_n/E_r/1 is a special case of the MMPP/G/1 queueing model, the probability generating matrices \( A(d) \) and \( B(d) \) are found as described in Section II-A. The Laplace–Stieltjes transform of an \( r \)-stage Erlangian distribution with unity mean is given as [22] \( h(s) = (1 + s/r)^{-r} \). Hence, a desired degree parameter \( f \) is met by setting \( r = f - 1 \).

We provide CPU time and error results for the matrix-sign function (MSF), generalized Schur decomposition (GSD) implementations of the generalized state-space approach (see Sections II-D and III), and the truncation-free successive substitution iteration (SSI) method outlined in Section II-B. We measure the CPU time until the point at which the program becomes ready to compute the level probability vectors \( x_k \),...
$k > 0$. This amounts to finding the matrix-geometric factors $g$, $F$, and $H$ in the case of the MSF and GSD methods, and to finding the level-0 probability vector $x_0$ and the matrix sequences $\{\hat{A}_k\}$ and $\{\hat{B}_k\}$ in the case of the SSI method. As for error computations, we consider the infinity norm of the residual of the solution vector: $\|x - xP\|_{\infty}$. However, since $x$ and $P$ are infinite entities, truncation is needed here. We simply consider the balance equation over levels 0 and 1 only. That is, we estimate the error as $\|\overline{x} - \overline{xP}\|_{\infty}$, where $\overline{x} = [x_0 \ x_1 \ x_2]$ and $P$ is the corresponding partition of $P$. Note that all three matrix-geometric factors of the solution are involved in this computation.

All MSF, GSD, and SSI methods are implemented in C, and compiled (by gcc version 2.7.2.1 with optimizer -O3) and run on a DEC Alpha server supporting IEEE standard double-precision arithmetic with machine epsilon $\epsilon \approx 2.2 \times 10^{-16}$. Standard BLAS (Basic Linear Algebra Subroutines) and CLAPACK (Fortran-to-C translated version of LAPACK—Linear Algebra Package) library routines [4] are used to perform all matrix operations. In the SSI implementation, all polynomial matrix evaluations are performed using Horner’s method, and the $\{\hat{A}_k\}$ and $\{\hat{B}_k\}$ matrix sequences are obtained by backward recursions on the $\{A_k\}$ and $\{B_k\}$ matrix sequences which are truncated at indexes $K_1$ and $K_2$ such that $\Sigma_{k > K_1} A_k e$ and $\Sigma_{k > K_2} B_k e$ have negligible components [24]. In all iterations, $\epsilon = 10^{-8}$ is used in the stopping criteria. In the SSI method, since the $G_{\epsilon}$ matrix is known to be stochastic, we use $\|G_{\epsilon} e - e\|_{\infty} < \epsilon$ as the stopping criterion.

In Table II, fixing the utilization as $\rho = 0.6$, we present the CPU time and error results for different values of $m$ and $f$. These results show that the MSF and GSD methods are highly accurate although they may incur more computational complexity than the SSI method as $f$ increases. It is noteworthy that the GSD method outperforms the (serial implementation) of the MSF method in terms of CPU time. Since $f = 4$ is the largest degree parameter considered here, we evaluate polynomials of at most degree three during the truncation-free successive substitution iteration of (15). The number of $A_k$ matrices generated ($K_1 \approx 14–21$) indicates the significant time savings achieved here by exploiting the rational structure in (15) in comparison to the traditional iteration [29].

$$G_0 = 0, \quad G_{j+1} = \sum_{i=0}^{K_1} A_i G_j^i,$$

In Table III, we provide the same set of results as in Table II for utilization $\rho = 0.9$. As these results indicate, the MSF and GSD methods are still very accurate. Furthermore, unlike the case for matrix-sign function iterations, the number of successive substitution iterations increase substantially with utilization, and the MSF method becomes faster than the SSI method as utilization increases unless the degree parameter $f$ is large. To further explore this point, we fix $m$ and $f$ as 32 and 3, respectively, and perform a stress test with respect to utilization. The results shown in Table IV indicate the efficiency and high numerical stability of the MSF and GSD methods under heavy load conditions which yield a very ill-conditioned numerical problem. In fact, Schur decomposition is, in general, known for its remarkable numerical stability [6]. This is observed as the GSD method becomes more accurate than the MSF method with increasing $\rho$. The fact that the CPU time for the GSD method is not affected at all by utilization is particularly noteworthy here.

Note that, in Table IV, the error of the SSI method decreases as utilization is increased. This counter-intuitive trend may be due to the fact that the probability mass moves to higher levels with increasing utilization, which makes error measurements through the balance equations over only levels 0 and 1 less representative of the actual error.

**Example 2:** We now consider a finite M/G/1-type Markov chain of the form (3) with $A_k = 0$ for $k \geq f$. The way $A(d)$ is obtained in this case was discussed in Section II-A. We note that the effective degree parameter in this example can be made less than $f$ (see the remark in Section II-D); however, we did not exploit this possibility in our implementation.
TABLE III
CPU TIME AND ERROR RESULTS FOR $\rho = 0.9$. $I_D$ AND $I_S$ ARE THE NUMBERS OF SUCCESSIVE SUBSTITUTION ITERATIONS AND MATRIX-SIGN FUNCTION ITERATIONS, RESPECTIVELY. $K_1$ AND $K_2$ ARE THE TRUNCATION INDEXES FOR THE $\{A_k\}_j$ AND $\{B_k\}_j$ (AND ALSO $\{A_k\}_j$ AND $\{B_k\}_j$) MATRIX SEQUENCES, RESPECTIVELY.

<table>
<thead>
<tr>
<th>$f$</th>
<th>$m$</th>
<th>$T_{CPU}$</th>
<th>Error</th>
<th>$I_D$</th>
<th>$K_1$</th>
<th>$K_2$</th>
<th>$T_{CPU}$</th>
<th>Error</th>
<th>$I_S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>16</td>
<td>0.800</td>
<td>6.3e-10</td>
<td>165</td>
<td>27</td>
<td>26</td>
<td>0.333</td>
<td>6.0e-16</td>
<td>8</td>
</tr>
<tr>
<td>32</td>
<td>5.383</td>
<td>5.9e-10</td>
<td>102</td>
<td>27</td>
<td>26</td>
<td>2.283</td>
<td>1.0e-15</td>
<td>9</td>
<td>1.033</td>
</tr>
<tr>
<td>64</td>
<td>40.048</td>
<td>6.1e-10</td>
<td>160</td>
<td>27</td>
<td>26</td>
<td>19.833</td>
<td>2.9e-15</td>
<td>9</td>
<td>8.450</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>1.083</td>
<td>5.9e-10</td>
<td>180</td>
<td>21</td>
<td>20</td>
<td>0.867</td>
<td>5.2e-15</td>
<td>8</td>
</tr>
<tr>
<td>32</td>
<td>7.333</td>
<td>6.1e-10</td>
<td>170</td>
<td>20</td>
<td>19</td>
<td>7.233</td>
<td>3.3e-14</td>
<td>8</td>
<td>2.617</td>
</tr>
<tr>
<td>64</td>
<td>53.498</td>
<td>5.7e-10</td>
<td>166</td>
<td>20</td>
<td>19</td>
<td>59.298</td>
<td>8.6e-14</td>
<td>8</td>
<td>22.266</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1.517</td>
<td>6.0e-10</td>
<td>194</td>
<td>18</td>
<td>18</td>
<td>1.933</td>
<td>6.8e-13</td>
<td>8</td>
</tr>
<tr>
<td>32</td>
<td>9.533</td>
<td>6.3e-10</td>
<td>177</td>
<td>18</td>
<td>17</td>
<td>16.749</td>
<td>2.3e-12</td>
<td>8</td>
<td>6.000</td>
</tr>
<tr>
<td>64</td>
<td>70.031</td>
<td>5.7e-10</td>
<td>170</td>
<td>18</td>
<td>17</td>
<td>137.111</td>
<td>8.0e-12</td>
<td>8</td>
<td>48.598</td>
</tr>
</tbody>
</table>

TABLE IV
CPU TIME AND ERROR RESULTS AS FUNCTIONS OF $\rho$ FOR $\nu = 3.2$ AND $f = 3$. $I_D$ AND $I_S$ ARE THE NUMBERS OF SUCCESSIVE SUBSTITUTION ITERATIONS AND MATRIX-SIGN FUNCTION ITERATIONS, RESPECTIVELY. $K_1$ AND $K_2$ ARE THE TRUNCATION INDEXES FOR THE $\{A_k\}_j$ AND $\{B_k\}_j$ (AND ALSO $\{A_k\}_j$ AND $\{B_k\}_j$) MATRIX SEQUENCES, RESPECTIVELY.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$T_{CPU}$</th>
<th>Error</th>
<th>$I_D$</th>
<th>$K_1$</th>
<th>$K_2$</th>
<th>$T_{CPU}$</th>
<th>Error</th>
<th>$I_S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1.533</td>
<td>1.1e-09</td>
<td>22</td>
<td>13</td>
<td>13</td>
<td>6.533</td>
<td>1.7e-14</td>
<td>6</td>
</tr>
<tr>
<td>0.6</td>
<td>2.267</td>
<td>1.7e-09</td>
<td>38</td>
<td>16</td>
<td>15</td>
<td>6.533</td>
<td>3.2e-14</td>
<td>6</td>
</tr>
<tr>
<td>0.8</td>
<td>4.100</td>
<td>1.0e-09</td>
<td>84</td>
<td>19</td>
<td>18</td>
<td>6.866</td>
<td>2.5e-14</td>
<td>7</td>
</tr>
<tr>
<td>0.9</td>
<td>7.233</td>
<td>6.1e-10</td>
<td>170</td>
<td>20</td>
<td>19</td>
<td>7.200</td>
<td>3.3e-14</td>
<td>8</td>
</tr>
<tr>
<td>0.95</td>
<td>13.266</td>
<td>3.0e-10</td>
<td>335</td>
<td>21</td>
<td>20</td>
<td>7.550</td>
<td>1.2e-14</td>
<td>9</td>
</tr>
<tr>
<td>0.99</td>
<td>56.314</td>
<td>5.9e-11</td>
<td>1542</td>
<td>21</td>
<td>21</td>
<td>8.516</td>
<td>1.6e-13</td>
<td>12</td>
</tr>
<tr>
<td>0.995</td>
<td>107.096</td>
<td>2.9e-11</td>
<td>2946</td>
<td>21</td>
<td>21</td>
<td>8.483</td>
<td>7.9e-14</td>
<td>12</td>
</tr>
<tr>
<td>0.999</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9.433</td>
<td>1.7e-12</td>
<td>15</td>
</tr>
<tr>
<td>0.9995</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9.416</td>
<td>3.6e-12</td>
<td>15</td>
</tr>
<tr>
<td>0.9999</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>11.100</td>
<td>3.7e-11</td>
<td>20</td>
</tr>
</tbody>
</table>

The $m \times m$ matrices $A_k$, $0 \leq k < f$, are specified as follows:
1) $A_0$ and $A_{f-1}$ are both diagonal matrices with constant diagonal entries $1 - 2\tau$ and $\tau$, respectively, where $0 < \tau < 0.5$.
2) All other $A_k$, $0 < k < f - 1$, are equal to each other, and tridiagonal with null diagonal entries, i.e., their nonzero entries are $a_{i,j+1} = a_{i,j-1} = 0.5\tau/(f - 2)$, $0 < i < m - 1$, and $a_{0i} = a_{m-1,i} = \tau/(f - 2)$. It can be shown that the utilization of this system is $\rho = 3(f - 1)\tau/2$, irrespective of $m$. So, given $f$ and $\rho$, the parameter $\tau$ is uniquely determined, and the number of phases $m$ can be arbitrarily chosen.

Having defined $A_k$’s, we assume the level-0 boundary behavior of (5), and $B_k$’s follow accordingly. As for the limiting level $K$, we use (4) to obtain $B_{K-1}$ and $A_k$’s. The hardware and software platform used is the same as described in Example 1.

Table V presents the CPU time and error results obtained using the generalized Schur decomposition method (see Sections II-D and III-B). As was observed in Example 1, the CPU time for this method is insensitive to utilization. Therefore, in Table V, we only provide CPU time versus $K$ results, and this time, the measurements are taken both after finding the matrix-geometric factors of the solution form and after all $K+1$ level probability vectors are computed. We also provide the CPU times for the infinite-level counterpart of the original chain. Note that the effect of the number of levels on the CPU time is minimal in this example. In addition, even for $K$ as high as 1000, the solution of the finite chain takes about twice the time required to solve the infinite chain. We are thus led to believe that approximating finite M/G/1 chains by their infinite counterparts may be unnecessary in many circumstances. On the other hand, the error is computed as the infinity norm of the residual of the complete solution vector: $\|x - xF\|_{\infty}$; see (3) and (2). Apart from a reasonable worsening for utilization very close to unity, Table V verifies the accuracy and numerical stability of our method under this ultimate error measure.

Example 3: This example is on an infinite M/G/1 chain with multiple boundary levels arising in queueing systems with multiple servers. We assume a discrete-time, slotted queueing system with the following evolution equation for the queue length:

$$Q_{n+1} = \max(0, Q_n + C_n - N) + r_n, \quad n \geq 0$$

where $Q_n$ is the queue length at the end of the $n$th slot, $N \geq 1$ is the number of servers, $C_n$ is the total number of arrivals of type 1 traffic in the $n$th slot with $C_n \leq N$, ...
TABLE V

<table>
<thead>
<tr>
<th>The Error of the Generalized Decomposition Method for the Finite M/G/1 Chain as a Function of ( \rho ) for ( K = 10, 100, ) and ( 1000 ). ( T_{C^{(3)}}^{(3)} ) is the Total Time Elapsed to Obtain the Complete Solution Vector. ( T_{C^{(3)}}^{(2)} ) is the Time Elapsed to Obtain the Matrix-Geometric Factors for the Finite Chain, Whereas ( T_{C^{(1)}}^{(3)} ) is That for the Corresponding Infinite Chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m = 16, f = 8 )</td>
</tr>
<tr>
<td>( \rho )</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>0.9</td>
</tr>
<tr>
<td>0.999</td>
</tr>
<tr>
<td>1.001</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>1.4</td>
</tr>
</tbody>
</table>

\( T_{C^{(1)}}^{(1)} \) | 7.166 | 9.383 | 15.233 | 6.766 | 7.616 | 11.900 |
| \( T_{C^{(2)}}^{(2)} \) | 7.133 | 9.366 | 13.083 | 6.759 | 7.401 | 9.816 |
| \( T_{C^{(3)}}^{(3)} \) | 5.016 | \( 1.566 \) |

\( r_n \) is the total number of arrivals in the \( n \)th slot due to type 2 traffic in the \( n \)th slot, and type 2 traffic is buffered when the servers are busy. \( C_n \) is modulated by a homogeneous finite-state, aperiodic discrete-time Markov chain with state transitions taking place only at the slot boundaries. Let \( S_n \) be the state of the modulating chain before the end of slot \( n \). We also have

\[
c_{ij}(d) = E[d^{C_n-1}1(S_{n+1} = j|S_n = i)], \quad 1 \leq i, j \leq m,\]

\[
C(d) = \{c_{ij}(d)\}
\]

where \( 1(E) \) is equal to one if the event \( E \) is true and zero otherwise. Note that \( C(d) \) takes the following form

\[
C(d) = \sum_{i=0}^{M} C_i d^{-i}, \quad M \leq N.
\]

We also assume \( r_n \) to be an independent geometric batch process with parameter \( \alpha \), i.e.,

\[
r(d) = E[d^{r_n}] = \frac{(1-\alpha)d}{d-\alpha},
\]

This queueing system is suitable for modeling voice and data traffic multiplexing over a single channel, and falls into the M/G/1 paradigm with multiple boundary levels with the doublet \((Q_m, S_n)\) being Markov and having a transition probability matrix of the form (9). It is not difficult to show that \( B_k(d), 0 \leq k \leq N-1, \) and \( A(d) \) can now be written as

\[
B_0(d) = (C_0 + C_1 + \cdots + C_M)r(d)
\]

\[
B_1(d) = (C_0 + C_1 + \cdots + C_M)r(d)
\]

\[
B_{N-M+1}(d) = (C_Md^{-M} + (C_0 + C_1 + \cdots + C_M^{-1})r(d)
\]

\[
B_{N-M+2}(d) = (C_Md^{-M} + C_1d^{-1} + (C_0 + C_1 + \cdots + C_M^{-2})r(d)
\]

\[
A(d) = (C_Md^{-M} + C_{M-1}d^{-M+1} + \cdots + C_0)r(d).
\]

One can then obtain a stable matrix fraction as in (72) by choosing

\[
Q(d) = \frac{1}{1-\alpha}d^{N-1}(d-\alpha)I
\]

and

\[
R_0(d) = (C_0 + C_1 + \cdots + C_M)d^N
\]

\[
R_1(d) = (C_0 + C_1 + \cdots + C_M)d^N
\]

\[
\vdots
\]

\[
R_{N-M+1}(d) = C_Md^{N-M-1} + (C_0 + C_1 + \cdots + C_{M-1})d^N
\]

\[
R_{N-M+2}(d) = C_Md^{N-M-2} + C_1d^{N-M-1} + (C_0 + C_1 + \cdots + C_{M-2})d^N
\]

\[
\vdots
\]

\[
P(d) = C_Md^{N-M} + C_{M-1}d^{N-M+1} + \cdots + C_0d^N.
\]

With the polynomial matrix fractions obtained as above, we implemented the method outlined in Section V using MATLAB on the same hardware platform as in Example 1. We used the matrix-sign function approach of Section III-A. We take \( N = 2, M = 2, \) and

\[
C(d) = \begin{bmatrix}
9.9996e-1 & 0 \\
1.0000e-5 & 0 \\
0 & 7.4999e-1
\end{bmatrix} d^{-1}
\]

\[
+ \begin{bmatrix}
0 & 1.0000e-5 \\
0 & 2.5000e-1
\end{bmatrix} d^{-2}
\]

which amounts to a bursty traffic of type 1 yielding a load of 50% on the system. The traffic parameter \( \alpha \) is chosen so as to yield a desired overall arrival rate. We vary the total load \( \rho \) on the system and present the results in Table VI. The error is computed as the infinity norm of the difference between the left- and right-hand sides of (75). \( P[empty] = x_0e \) is the probability of empty queue, and \( E[queue] \) is the expected queue length. To give an example about the detailed results, we have obtained the following matrices that constitute the
We note that these matrices do not have a probabilistic interpretation, unlike the case for the matrix-analytic approach for M/G/1- and G/M/1-type Markov chains [28], [29]. We have also found the geometric decay rate for the level distribution, i.e., the eigenvalue of $F$ closest to the unit circle, which happens to be $\beta = 0.999996$ for this example.

Certain advantages of the simple and compact matrix-geometric form for the stationary solution of M/G/1 chains were addressed briefly at the end of Section II-D. The results of this section demonstrate the accuracy and numerical stability of two particular implementations (based on ordinary matrix-sign function and generalized Schur decomposition with ordering) of the generalized state-space approach of this paper. The results also indicate substantial savings in the CPU time and storage requirements compared to conventional recursive techniques, especially when the degree parameter $f$ is not large. However, in the case of intolerably large $f$, rational approximation techniques can be employed to reduce $f$ and decrease the computational complexity with insignificant loss of accuracy. For example, in [1], the deterministic service time in a MAP/D/1/K queueing system, which indeed results in $f = \infty$, is modeled by Padé approximations in transform domain with a reduced degree of $f = 3$, and very accurate estimates for cell loss rates in an ATM multiplexer are obtained efficiently.

**Appendix I**

**Polynomial Matrices and Fractions**

The following material on polynomial matrices and polynomial matrix fractions of a rational matrix $A(d)$ is mainly based on [21] and [9].

A matrix $A(d) = [a_{i,j}(d)]$, where $a_{i,j}(d) = p_{i,j}(d)/q_{i,j}(d)$ for a polynomial pair $p_{i,j}(d)$ and $q_{i,j}(d)$, is called a rational matrix in $d$. If $\deg(p_{i,j}) \leq \deg(q_{i,j})$ ($\deg(p_{i,j}) < \deg(q_{i,j})$) for all $i$ and $j$, then $A(d)$ is called proper (strictly proper). We say $A(d)$ is stable if all roots of $q_{i,j}(d)$ lie in the open unit disk for all $i$ and $j$. A polynomial matrix is one with polynomial entries. Let $Q(d)$ and $P(d)$ be $m_q \times m_q$ and $m_p \times m_p$ polynomial matrices, respectively, and let $Q(d)$ be nonsingular. $Q(d)$ and $P(d)$ are said to be right coprime over an arbitrary region $D$ in the complex plane $C$ if and only if, for every $d \in D$, the matrix $[Q^T(d) \ P^T(d)]^T$ has full column rank $m_q$. If in the above definition $D$ is outside the open unit disk, that is, if $D$ is the set $\{d \in C, |d| \geq 1\}$, then the fraction is called stable right coprime. Consider a rational matrix $A(d)$ of size $m_p \times m_q$. The fraction $A(d) = P(d)/Q^{-1}(d)$ is called a stable right coprime polynomial fraction if the polynomial matrices $Q(d)$ and $P(d)$ are stable right coprime. Given a stable rational matrix $A(d)$, it is always possible to obtain a stable right coprime polynomial fraction [9].

**Appendix II**

**Generalized State-Space Realization**

Here we provide a proof for the generalized state-space realization (26) for $y_k$. Following the treatment of [1] in a similar context, we define

$$y^j_k \triangleq y_{k+j-1}, \quad y^j(d) = \sum_{k=0}^{\infty} y_k^j d^{-k}, \quad 1 \leq j \leq f$$

which yields

$$y^j(d) = y(d) \delta_{0r}^{j-1}, \quad 2 \leq j \leq f.$$  

Here, $j$ in $y^j_k$ should be treated as a superscript. We note that

$$y^j_k = \lim_{d \to \infty} y^j(d), \quad 1 \leq j \leq f$$

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

$$dy^j(d) = y^{j+1}(d) + d_0 y^j_k, \quad 1 \leq j \leq f - 1.$$  

One can also show by using (22) and by algebraic manipulations that

$$dy^{j}(d)D_f = - \sum_{i=0}^{f-1} y^{i+1}(d)D_i + \sum_{i=1}^{f} x_0 N_i d^i$$

$$- \sum_{i=1}^{f-1} \sum_{r=1}^{i} d^{i-r} y^r_k D_{i-r} - d \sum_{i=1}^{f-1} d^{i-1} y^i_k D_f.$$  

Since $\lim_{d \to \infty} y^j(d)D_f$ exists, (79) dictates linear constraints on $y^j_k, 1 \leq j \leq f$, in the following manner:

$$y^j_k D_f = x_0 N_{j-1} - \sum_{i=1}^{j-1} y^i_k D_{j-i}.$$  

Consequently, (79) becomes

$$dy^{j}(d)D_f = - \sum_{i=0}^{f-1} y^{i+1}(d)D_i + d_0 y^{j}(d).$$  

Let us now define the concatenated transform vector

$$z(d) \triangleq \begin{bmatrix} y^1(d) & y^2(d) & \cdots & y^f(d) \end{bmatrix}$$

which is the $d$-transform of the sequence $z_k = [y^1_k \ y^2_k \ \cdots \ y^f_k]$. One can then make use of (78), (80), and (81) to obtain

$$z(d)(dE - A) = d_0 z_0, \quad x_0 D = x_0 N, \quad y(d) = z(d) C$$

where the matrices $A$, $E$, $C$, $D$, and $N$ are as defined in (27), (28), and (29), respectively.

We note that the transform identities in (82) are equivalent to the representation (26), and therefore we have found one particular generalized state-space realization for the transform expression (22) for $y(d)$. 

$$\begin{bmatrix} 638 \ IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 16, NO. 5, JUNE 1998 
\end{bmatrix}$$
REFERENCES


Nail Akar (M’97) received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1987 and M.S. and Ph.D. degrees from Bilkent University, Ankara, Turkey, in 1989 and 1994, respectively, all in electrical and electronics engineering.

From 1994 to 1995, he conducted research as a visiting scholar in Computer Science Telecommunications at the University of Missouri-Kansas City, and in 1996, he was a Visiting Assistant Professor in the same program. Since 1997, he has been with the Technology Planning and Integration, Long Distance Division, Sprint, Overland Park, KS. His current research areas include queueing analysis, IP/ATM networks, flow control, routing, and pricing.

Nihat Cem Oğuz received the B.S. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 1987. He received the M.S. and Ph.D. degrees from Bilkent University, Turkey, in 1990 and 1995, respectively, both in electrical and electronics engineering.

In 1996, he joined the same program as a Visiting Assistant Professor since 1997. His current research interests include network performance evaluation, cell loss process characterization, and forward error correction for lost cell recovery in ATM networks.