Computing Packet Loss Probabilities in Multiplexer Models Using Rational Approximation

Annie Cuyt, R.B. Lenin, Gert Willems, Chris Blondia, and Peter Rousseeuw

Abstract—A statistical multiplexer is a basic model used in the design and the dimensioning of communication networks. The multiplexer model consists of a single server queue with constant service time and a more or less complicated arrival process. The aim is to determine the packet loss probability as a function of the capacity of the buffer. In this paper, we show how rational approximation techniques may be applied to compute the packet loss efficiently. The approach is based on the knowledge of a limited number of sample values, together with the decay rate of the probability distribution function. A strategy is proposed where the sample points are chosen automatically. The accuracy of the approach is validated by comparison with both analytical results obtained using a matrix-analytic method and simulation results.

Index Terms—Statistical multiplexing, Markovian arrival process, matrix-analytic methods, Newton-Padé approximation.

1 Introduction

Variable bit rate communications with real-time constraints in general, and video communication services (video phone, video conferencing, television distribution) in particular, are expected to be a major class of services provided by the future Quality of Service (QoS) enabled Internet. This network must offer a high degree of flexibility, together with efficiency in resource consumption, by sharing the same network resources (bandwidth and buffers) among several connections with different characteristics (bandwidth, peak bit rate, correlation) requiring different QoS level guarantees. The introduction of statistical multiplexing techniques, such as provided in ATM networks, offers the capability to efficiently support variable bit rate connections by taking advantage of the variability of the bandwidth requirements of individual connections. In this way, connections share a link, of capacity less than the sum of the individual peak bit rates, achieving a more or less significant multiplexing gain while guaranteeing the often stringent QoS requirements with respect to packet loss, end-to-end packet delay, and delay jitter.

In order to assess the multiplexing gain, a variety of techniques have been developed in recent years, based on exact analysis, approximate analysis, and simulation, to study these multiplexer models. In particular, considerable work has been done on the development of analytical techniques for evaluating packet loss probabilities, also called cell loss probabilities (CLP). In these models, the traffic is described by Markovian arrival processes, leading to a Markov model of M/G/1-type [3], [9], [12], [18]. Unfortunately, these techniques incur high computation costs and are therefore sometimes practically impossible. Hence, considerable attention has been paid to the development of techniques that provide approximate estimates for performance metrics. These techniques include methods which approximate the arrival process by fluid models [5], approaches based on generating functions [17], [20], and matrix-analytic methods [12]. However, the computational requirements of the algorithms grow quite rapidly as a function of the system complexity.

Monte Carlo simulation is also used to compute the CLP. If the desired cell loss probability is in the range of 10^{-6} to 10^{-12}, depending on the kind of service, it is, however, computationally impossible to use the conventional Monte Carlo simulation. A simulation technique called Importance Sampling (IS) can speed up simulations involving rare events such as CLP [4]. However, because of the complicated nature of multiplexer queuing models, applying the IS technique is not straightforward.

Recently, another approach to compute the CLP as a function of the system size has become available, based on the use of rational approximation techniques. The motivation behind this approach is that it is computationally feasible to evaluate the CLP as a function of the system size when the system size is small and, moreover, it is often possible to study interesting properties of this function such as monotonicity, convexity, boundedness, and asymptotic behavior [6], [11]. In [8], [21], the authors have employed rational approximants to compute the CLP in ATM networks fed by a population of ON-OFF sources. Their studies were mainly limited to models where the correlation between the cells was ignored, that is, the transition probabilities of the Markov chains which modulate these sources were large. To introduce more correlation between the cells, these transition probabilities should be at least less than 10^{-3} [2]. Considering a high degree of correlation is of

• *The authors are with the Department of Mathematics and Computer Science, University of Antwerp, Universiteitsplein 1, B-2610 Antwerp, Belgium. E-mail: {Annie.Cuyt, Lenin.Bhavanandan, Gert.Willems, Chris.Blondia, Peter.Rousseeuw}@ua.ac.be.*

Manuscript received 10 Jan. 2002; accepted 9 Sept. 2002.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 115675.

major importance when the input consists of more video sources [3].

In [8], [21], the authors have computed the CLP for larger system sizes from the knowledge of the sample values for small sample points and the decay rate of the CLP function. It has been noticed that employing this technique to approximate the function in a multiplexer model with less correlation between the cells is rather straightforward because the graph of the CLP becomes linear in logarithmic scale rather quickly. This property is no longer valid when the correlation between the cells increases. Then, one has to choose a sample point corresponding to a larger buffer size. Choosing sample points is a difficult task since it depends on the various networks and system parameters. In this paper, we propose a strategy where sample points are chosen automatically.

In most real-world network environments, the network load is not close to 1 (heavy traffic). In [8], [21], the authors have chosen numerical examples with fairly heavy traffic, which leads to the case where the graph of the CLP becomes linear rather quickly, facilitating the approximation technique quite a lot. In this paper, we show that there exist networks with both light and heavy loads such that the graph of the CLP becomes linear from a large buffer size on, requiring the method to choose a large sample point. Unlike in [21], we compare the results obtained using the rational approximation approach with results obtained using the matrix-analytic approach proposed in [9], [10] and also with simulation results.

2 Model Description

In the multiplexer environment, cells have the same length and, hence, a fixed service time, which makes the discrete time Markov chain a natural modeling choice. We assume that the arrival of cells which are transmitted by M independent and nonidentical information sources to the multiplexer can be modeled as a discrete time batch Markovian Arrival Process (D-BMAP), the discrete-time version of BMAP. The BMAP is a convenient representation of the versatile Markovian point process which generalizes the Markovian arrival process (MAP) [12]. The D-BMAP is a general process used to model a number of arrival processes, for example, video [3] and periodic processes [7]. Each information source is controlled by a Markov chain, called the background Markov chain. So, the basic queuing system which models the multiplexer is a D-BMAP/D/c/N queue with c discrete time servers, where each server can serve at most one cell per time unit. These servers serve a queue with a capacity of N cells which is fed by M independent information sources. When all the servers are busy, a maximum number of c cells will depart in each slot. Service starts at the beginning of each time slot.

The arrival process associated with a single source is modeled as an Interrupted Bernoulli Process (IBP). This process has two states, 0 and 1. Source i generates a cell with probability $d_i(m)$ when it is in state m $(= 0, 1)$. Source i has the following transition probability matrix:

$$\boldsymbol{Q}_i = \begin{pmatrix} 1-p_i & p_i \\ q_i & 1-q_i \end{pmatrix}. \tag{1}$$

The system can be modeled as a two-dimensional discrete time Markov chain $\{(X_n, Y_n), n \geq 0\}$, where X_n is the number of cells in the buffer and Y_n represents the state of the M sources during the nth time slot. We are interested in the steady state behavior $(X, Y) \equiv \lim_{n\to\infty}(X_n, Y_n)$. Clearly, the state spaces S_X and S_Y of the processes X and Y are given by

$$\begin{aligned} &S_X = \{0, 1, 2, \ldots, N\} \quad \text{and} \\ &S_Y = \{(m_1, m_2, \ldots, m_M) \mid m_i = 0 \text{ or } 1\}. \end{aligned} \tag{2}$$

Let M_i be the background Markov chain for source i. The transition probability matrix $\boldsymbol{Q}_i$ of M_i is given by (1). Significant reduction can be made in the state space Y when the sources are identical. We will discuss this case in Section 2.2.

2.1 Cell Loss Probabilities

The transition probability matrix $\boldsymbol{D}$ of the process Y is given by:

$$\boldsymbol{D} = \bigotimes_{i=1}^{M} \boldsymbol{Q}_i, \tag{3}$$

with dimension $2^M \times 2^M$.

Let $\boldsymbol{D}_m$ be the matrix corresponding to m arrivals during a time slot. Then,

$$\boldsymbol{D}_m = \sum_{\substack{j_1, j_2, \ldots, j_M \\ j_i = 0 \text{ or } 1,\, 1 \leq i \leq M \\ j_1 + j_2 + \cdots + j_M = m}} \bigotimes_{i=1}^{M} \Big[(1-j_i)\boldsymbol{I} + (-1)^{(1-j_i)}\boldsymbol{P}_i\Big]\boldsymbol{Q}_i, \tag{4}$$

where

$$\boldsymbol{P}_i = \begin{pmatrix} d_i(0) & 0 \\ 0 & d_i(1) \end{pmatrix}, \quad i = 1, 2, \ldots, M \tag{5}$$

and $\boldsymbol{I}$ is the identity matrix of order 2×2. The dimension of the matrix $\boldsymbol{D}_m$ is $2^M \times 2^M$ (see, for example, [19]).

Since we assume that each source can generate at most one cell during a time slot and there are M sources, at most M cells can arrive at the multiplexer during a time slot. Therefore, there are $M + 1$ matrices governing the arrivals, namely, $\boldsymbol{D}_0, \boldsymbol{D}_1, \ldots, \boldsymbol{D}_M$.

The average arrival rate of the cells at the multiplexer

$$\lambda = \bar{\boldsymbol{\xi}}\left(\sum_{m=0}^{M} m\boldsymbol{D}_m\right)\bar{\boldsymbol{e}}, \tag{6}$$

where $\bar{\boldsymbol{e}}$ is a column vector of ones and $\bar{\boldsymbol{\xi}}$ is such that $\bar{\boldsymbol{\xi}}\boldsymbol{D} = \bar{\boldsymbol{\xi}}$ and $\bar{\boldsymbol{\xi}}\bar{\boldsymbol{e}} = 1$. The load (traffic intensity) of the network is $\rho = \frac{\lambda}{c}$. Under the condition of ergodicity $(\rho < 1)$ of the chain (X, Y), the stationary distribution vector $\bar{\boldsymbol{\Pi}} := \{\bar{\boldsymbol{\pi}}_0, \bar{\boldsymbol{\pi}}_1, \ldots, \bar{\boldsymbol{\pi}}_N\}$, $(\bar{\boldsymbol{\pi}}_i \in \mathbb{R}^{2^M})$ satisfies

$$\bar{\boldsymbol{\Pi}}\boldsymbol{P} = \bar{\boldsymbol{\Pi}} \quad \text{and} \quad \bar{\boldsymbol{\Pi}}\bar{\boldsymbol{e}} = 1, \tag{7}$$

where the transition probability matrix $\boldsymbol{P}$ of the process (X, Y) is given by [9]

$$P = \begin{pmatrix} D_0 & D_1 & \dots & D_{N-C} & \dots & D_{N-1} & B_N \\ D_0 & D_1 & \dots & D_{N-C} & \dots & D_{N-1} & B_N \\ \vdots & & & & & & \vdots \\ D_0 & D_1 & \dots & D_{N-C} & \dots & D_{N-1} & B_N \\ 0 & D_0 & \dots & D_{N-C-1} & \dots & D_{N-2} & B_{N-1} \\ 0 & 0 & \dots & D_{N-C-2} & \dots & D_{N-3} & B_{N-2} \\ \vdots & & & & & & \vdots \\ 0 & 0 & \dots & D_0 & \dots & D_{C-1} & B_C \end{pmatrix}_{(N+1)2^M \times (N+1)2^M} \tag{8}$$

with

$$B_n := \sum_{j=n}^{M} D_j.$$

The cell loss probability function

$$P_L(N) := \frac{1}{\lambda} \sum_{n=0}^{N} \bar{\pi}_n \sum_{k=0}^{M} [k+n-N]^+ D_k \bar{e}, \tag{9}$$

where $[x]^+ := \max(0, x)$.

2.2 Particular Case

Suppose all the sources are homogeneous (identical). Then, $p_i = p$ and $q_i = q$ for $i = 1, 2, \dots, M$. For this case, the state space of Y is

$$S_Y = \{0, 1, 2, \dots, M\},$$

where $i \in S_Y$ denotes the number of active sources. This drastic reduction in the state space of Y is due to the fact that the sources are identical. The state space S_X remains the same.

Each of the M sources will generate a cell with probability d when it is in active state (or state 1) and no cells when it is in idle state (or state 0), that is, $d_i(0) = 0$ and $d_i(1) = d$ for all $i = 1, 2, 3, \dots, M$. For this case, the (i, j)-th element d_{ij} of the transition probability matrix D of Y is given by

$$d_{ij} = \sum_{k=0}^{i} \binom{i}{k} \binom{M-i}{k+j-i} q^k (1-q)^{i-k} p^{j+k-i} (1-p)^{M-j-k}. \tag{10}$$

When the parameters p and q are very small (more correlation between the arriving cells), then the d_{ij} in (10) can be approximated by the following formula:

$$d_{ij} = \begin{cases} 1 - (M-i)p - iq, & \text{if } j = i \\ (M-i)p, & \text{if } j = i+1 \\ iq, & \text{if } j = i-1. \end{cases} \tag{11}$$

That is, the d_{ij} are one-step transition probabilities and the matrix D corresponds to the transition probability matrix of a birth-death process with birth rate $(M-i)p$ and death rate iq when the process is in state i.

The matrices D_m are given by

$$D_m = \mathrm{Diag}(c_m(0), c_m(1), \dots, c_m(M)) D, \quad m = 0, 1, \dots, M, \tag{12}$$

where

$$c_m(k) = \begin{cases} \binom{k}{m} d^m (1-d)^{k-m}, & \text{if } d \neq 1 \\ \delta_{mk}, & \text{if } d = 1 \end{cases} \tag{13}$$

is the probability of m arrivals during a time slot when the process Y is in state k. The formulae to compute λ, P, and $P_L(N)$ remain the same, namely, (6), (8), and (9), respectively. For this simple case, the (i, j)-th element of D_m equals the probability of m arrivals at the buffer during a time slot when the background Markov chain changes from state i to j.

For this homogeneous case, the matrix P is a square matrix of order $(N+1)(M+1)$.

2.3 Decay Rate

It has been proven that, for infinite M/G/1-type queues, the buffer overflow probability decays exponentially [6]. In [11], the authors have shown that, for Markov modulated queuing models with multiserver and infinite buffer, the queue length distribution has exponential bounds. In [1], the author has studied the exponential decay of the loss probability of the finite MAP/G/1/K queue. In all these papers, the exponential decay rate is studied by providing some conditions on the stationary queue length distribution. We assume that these conditions hold in our D-BMAP/D/*c*/N queuing models and use the approach provided in [6]. Apparently, our numerical results show that the loss probability of D-BMAP/D/*c*/N queues decays exponentially.

We now briefly discuss the approach to compute the decay rate from the knowledge of the parameters for a given model. We first show how we arrange the blocks in the matrix P for the multiserver case so that the structure of P is similar to that of a finite M/G/1-type Markov chain.

Define

$$A_0 := \begin{pmatrix} D_0 & D_1 & \cdots & D_{c-1} \\ 0 & D_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & D_1 \\ 0 & \cdots & 0 & D_0 \end{pmatrix},$$

$$A_i := \begin{pmatrix} D_{i\times c} & D_{i\times c+1} & \cdots & D_{i\times c+c-1} \\ D_{i\times c-1} & D_{i\times c} & \ddots & \vdots \\ \vdots & \ddots & \ddots & D_{i\times c+1} \\ D_{i\times c-c+1} & \cdots & D_{i\times c-1} & D_{i\times c} \end{pmatrix},$$

$$i = 1, 2, 3, \dots, K,$$

where $K = \lceil M/c \rceil$ ($\lceil 2^M/c \rceil$) for homogeneous (heterogeneous) sources.

The matrix A_i is a square matrix of size $c(M+1)$ if the sources are homogeneous and size $2^M c$ if the sources are heterogeneous. If $c = 1$, then $A_i = D_i,\ i = 0, 1, \dots, K$.

Define

$$A(z) := \sum_{n=0}^{K} A_n z^n, \quad 0 < z < R_A, \tag{15}$$

where R_A is the radius of convergence of $A(z)$. Then, for $z \in]1, R_A[$, the exponential decay rate ξ is the Perron-Frobenius eigenvalue of $A(z)$ satisfying [6] the condition

$\xi = z$. Since $P_L(N)$ decays exponentially with decay rate ξ, we have

$$\log P_L(N) \sim \xi N \quad \text{as } N \to \infty. \tag{16}$$

3 Rational Approximation

The new technique to compute $\log P_L(N)$ which is proposed here is a kind of "divide-and-conquer" technique. From [9], [10], we know that the function $P_L(N)$ can easily be evaluated for small values of the buffer length N. Also, the decay rate ξ of $\log P_L(N)$ can easily be obtained [6]. Combining this knowledge into a function model for $\log P_L(N)$ that is validated by one simulation point for a moderate value of N in a reasonable range of $P_L(N)$, will prove to be much more efficient than the traditional techniques used for the computation of $\log P_L(N)$, while the accuracy is comparable.

Because of the fact that the function $\log P_L(N)$ asymptotically behaves as ξN for large N, polynomial approximation techniques for $\log P_L(N)$ are not suitable. However, a rational function $r_n(N)$ of numerator degree $n+1$ and denominator degree n,

$$r_n(N) = \frac{\sum_{i=0}^{n+1} a_i N^i}{\sum_{i=0}^{n} b_i N^i},$$

has a similar asymptotic behavior as that of $\log P_L(N)$. Sometimes we shall denote $r_n(N)$ by $[n+1/n]$. It remains to compute the coefficients a_i and b_i in the numerator and denominator of the rational function from sampled function values $\log P_L(N_j)$ for chosen N_j and to fit its asymptotic behavior to ξ. The rational approximant of the type of $r_n(N)$ can be obtained as the $2n$th convergent of a so-called Thiele type continued fraction [14]:

$$\begin{aligned} r_n(N) &= \varphi[N_0] + \sum_{j=0}^{2n} \frac{N - N_j \quad |}{|\varphi[N_0, \ldots, N_{j+1}]} \\ &= \varphi[N_0] + \cfrac{N - N_0}{\varphi[N_0, N_1] + \cfrac{N - N_1}{\varphi[N_0, N_1, N_2] + \frac{N - N_2}{\cdots}}}, \end{aligned}$$

where the inverse differences $\varphi[N_0, \ldots, N_{j+1}]$ are computed recursively from

$$\begin{aligned} &\varphi[N_j] = \log P_L(N_j) \\ &\varphi[N_0, \ldots, N_{j+1}] = \\ &\frac{N_{j+1} - N_j}{\varphi[N_0, \ldots, N_{j-1}, N_{j+1}] - \varphi[N_0, \ldots, N_{j-1}, N_j]}. \end{aligned} \tag{17}$$

In order to fit the asymptotic behavior of $r_n(N)$ to that of $\log P_L(N)$, we only compute $\varphi[N_0, \ldots, N_{j+1}]$ with $j = 0, \ldots, 2n-1$ from (17). The last inverse difference $\varphi[N_0, \ldots, N_{2n+1}]$ is computed from the following property. The coefficient of highest degree in the numerator of $r_n(N)$, namely, a_{n+1} equals

$$a_{n+1} = \frac{1}{\sum_{j=0}^{n} \varphi[N_0, \ldots, N_{2j+1}]}.$$

For $r_n(N)$ to behave asymptotically like ξN, we need to require $a_{n+1} = \xi$ or, in other words,

$$\varphi[N_0, \ldots, N_{2n+1}] = \frac{1}{\xi} - \sum_{j=0}^{n-1} \varphi[x_0, \ldots, x_{2j+1}].$$

Let us summarize how the function $\log P_L(N)$ can be modeled by a rational function $r_n(N)$. The rational model is fully specified when we know its numerator and denominator coefficients $b_1, \ldots, b_n, a_0, \ldots, a_{n+1}$, which are in total $2n+2$ coefficients (b_0 in the denominator is only a normalization constant for the rational function [14]). Obtaining these coefficients is equivalent to computing the inverse differences $\varphi[N_0, \ldots, N_j]$ for $j = 0, \ldots, 2n+1$ in the continued fraction representation of $r_n(N)$. In total, $2n+1$ of these inverse differences are determined from sampling $\log P_L(N)$ at chosen N_j for $j = 0, \ldots, 2n$, while one value is determined from the asymptotic behavior

$$\log_{N \to \infty} P_L(N) \approx \xi N.$$

Interpolating or approximating an analytic function by polynomials or by rational functions with prescribed poles is rather well understood and has been studied in great detail in [16]. A rather different situation arises if one considers interpolation by rational functions with free poles. Free poles means that both the numerator and denominator coefficients are determined by the interpolation conditions, as is the case here, while, in the case of preassigned poles, this is true only for the numerator coefficients. The theoretical background of rational interpolation with free poles is very similar to that of Padé approximation. Actually, Padé approximants are a special case of rational interpolants with all the interpolation conditions concentrated in one point.

The accuracy of the model $r_n(N)$ is assessed by looking at:

$$\sup_{N \in \mathbb{N}} ||r_n(N) - r_{n+1}(N)||,$$

which tends to zero if $r_n(N)$ converges to $\log P_L(N)$. The convergence of the rational interpolant $r_n(N)$ is guaranteed by the following theorem [15]. Because we include interpolation conditions at infinity, namely,

$$\lim_{N \to \infty} r_n(N) = \infty \qquad \lim_{N \to \infty} r'_n(N) = \xi,$$

the support of the set of interpolation points is given by $[N_{\min}, \infty]$, where

$$N_{\min} = \min\{N_j \mid \exists\, n : N_j \text{ support point for } r_n(N)\}.$$

Theorem 1. *Let the single-valued function f be analytic everywhere in the extended complex plane, except in a compact set E of capacity zero. Let $[N_{\min}, \infty] \cap E = \emptyset$. Then, for every $\varepsilon > 0$ and for every compact set $B \subset \mathbb{C}$, we have*

$$\lim_{n \to \infty} \operatorname{cap}(\{z \in B : |(f - r_n)(z)| > \varepsilon^n\}) = 0.$$

TABLE 1
Strategy for Networks with Homogeneous Sources and Single Server

1	if $(\min(p,q) \geq 1e-1)$
1.1	if $(\lvert\xi\rvert \geq 1e-1)$
	stop
1.2	else
	$K=1;\ L=10;$ support $= \left\{K, \left\lceil \frac{K+L}{2} \right\rceil, L\right\}$
2	elseif $(\min(p,q) \geq 1e-2)$
2.1	if $(\lvert\xi\rvert \geq 1e-1)$
	stop
2.2	else
	$K=1;\ L=20;$ support $= \left\{K, \left\lceil \frac{K+L}{2} \right\rceil, L\right\}$
3	elseif $(\min(p,q) \geq 1e-3)$
	$K=1;\ L=20;$ support $= \left\{K, \left\lceil \frac{K+L}{2} \right\rceil, L\right\}$
4	elseif $(\max(p,q) < 1e-3)$
4.1	if $(\rho \geq 0.6)$
4.1.1	if $(\lvert\xi\rvert > 5e-4)$
	$K=10;\ L=30;$ support $= \left\{K, \left\lceil \frac{K+L}{2} \right\rceil, L\right\}$
4.1.2	else
	$K=1;\ L=30;$ support $= \left\{K, \left\lceil \frac{K+L}{2} \right\rceil, L, 50, 2000\right\}$
4.2	else
4.2.1	if $(\lvert\xi\rvert > 5e-4)$
	$K=10;\ L=30;$ support $= \left\{K, \left\lceil \frac{K+L}{2} \right\rceil, L\right\}$
4.2.2	else
	$K=10;\ L=30;$ support $= \{K, L, 500\}$

The above theorem is a special case of a more general theorem in which the convergence of more close-to-diagonal sequences of rational interpolants is proven under the condition that the support of the set of complex interpolation points does not intersect the exceptional set E. Here, we only need to focus on rational interpolants of numerator degree one more than the denominator degree and we know that the support is a subset of the positive real line where the function $\log P_L(N)$ is well-behaved.

4 Numerical Results

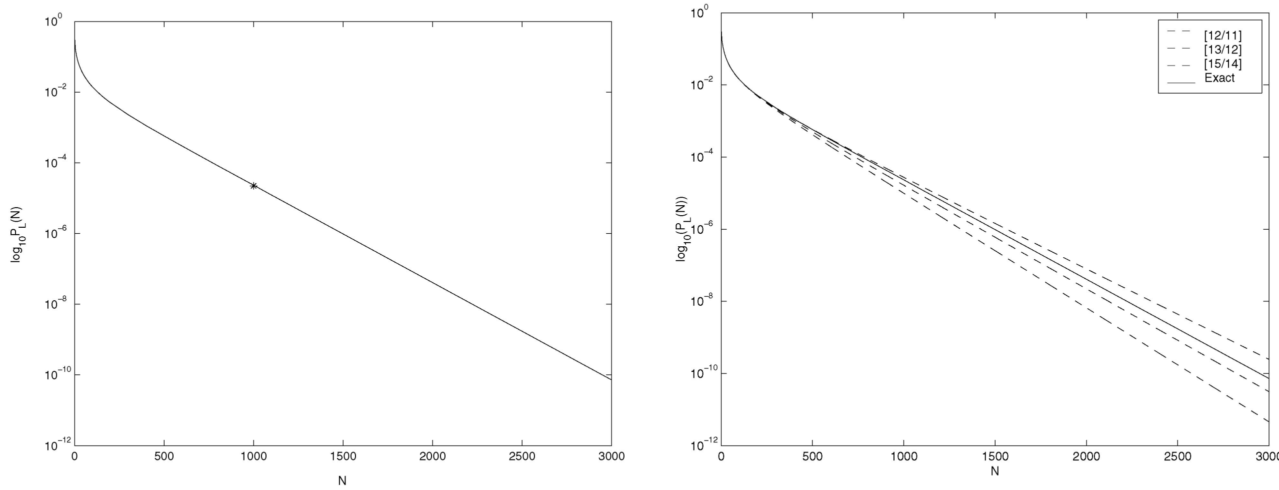
Since $\log P_L(N)$ decays linearly as N tends to infinity, we compute a rational interpolant $[n+1/n]$ to approximate this function. As mentioned in Section 3, we use $2n+1$ support points N_j and the decay rate ξ. Let us now illustrate all this with some numerical examples for networks with homogeneous and heterogeneous sources.

We also want to propose an algorithm that computes the model $r_n(N)$ in a fully automatic way, meaning that it selects the support points N_j automatically, depending on the given parameters M, c, p, q, d of the network with homogeneous sources or $M, c, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{d}$ of the network with heterogeneous sources. The algorithm proceeds as follows: Successive approximants $r_n(N)$ are computed for several values of n. Increasing n by one implies adding two more support points. For $n=1$, only three support points have to be specified to start the procedure. Two of these support points, denoted by K and L, will delimit the sampling range in the sense that all subsequent support points N_j satisfy $K < N_j < L$.

After conducting some numerical experiments, we found that the delimiters K and L can be fixed from the knowledge of the load, decay rate, and the number of servers of a given system so that the function $\log P_L(N)$ switches in the interval $[K, L]$ from a fast decreasing to a slowly decreasing function. When looking at the subsequent figures, it is apparent that the function $\log P_L(N)$ always makes that switch for not too large buffer sizes. For

TABLE 2
Parameter Values for Examples Considered in Section 4.1

parameter	Example 1	Example 2	Example 3	
M	10	15	30	
p	3.5e-2	2.19e-5	2.19e-5	
q	7.5e-2	7.0e-6	4.3e-5	
d	3.1e-1	6.8966e-2	7.6923e-2	
ρ	0.9864	0.6897	0.7787	
ξ	-2.754e-3	-1.332e-3	-1.43e-4	
$[n+1/n]$	[4/3]	[3/2]	[8/7]	
case from Table 1	2.2	4.2.1	4.1.2	
support points	1,6,9,11,14, 16,20	10,15,20,25, 30	case (A)	5,7,9,11, 12,14,15,17, 18,21,24,27, 30,50,60
			case (B)	1,5,9,13, 16,20,23,27, 30,50,500
			case (C)	1,9,16,23 30,50,1500
			case (D)	5,7,9,11, 12,14,15,17 18,21,24,27 30,50,2000



(a) (b)

Fig. 1. CLP for Example 1.

single-server homogeneous systems with light load ρ and large decay rate ξ, $K = 10$ and, in all other cases, $K = 1$. The delimiter L is chosen to be directly proportional to the decay rate ξ.

Below, the pseudocode for initializing the first three support points is given in three different situations: the case of a network with homogeneous sources and single server, that of a network with homogeneous sources and multi-server, and the case of a network with heterogeneous sources. Successive support points are added in the following way: A discrete approximation

$$\max_{N=K,\dots,L} ||r_n(N) - r_{n+1}(N)||$$
$$r_0(N) = (\log P_L(N_0) - \xi N_0) + \xi N$$

of $\sup_{N \in \mathbb{N}} ||r_n(N) - r_{n+1}(N)||$ is computed. The values of N in $[K, L]$ for which the maximum and the second largest value are attained are chosen to be the next two support points.

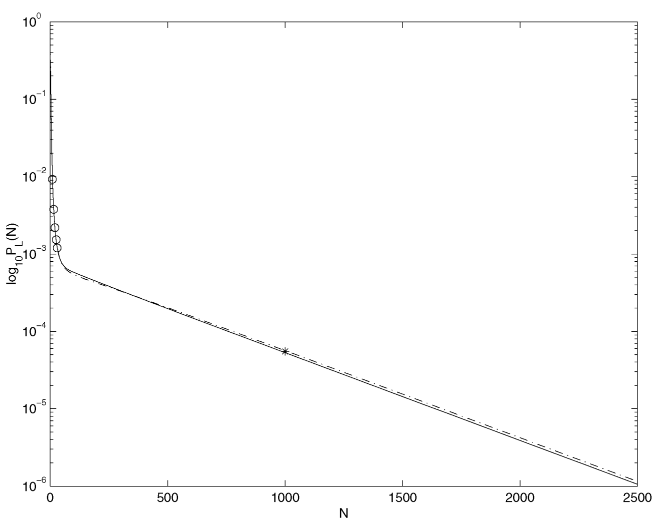


Fig. 2. CLP for Example 2.

The pseudocode is based on an extensive number of numerical experiments, varying the system parameters in all sorts of ways. Our main conclusions are the following:

- For networks with homogeneous sources:
    - When p and q are in the range of 10^{-1} to 10^{-3} and if $|\xi| > 0.1$, then $\log P_L(N)$ becomes smaller than 10^{-12} for small values of N, which is of less practical importance. If $|\xi| < 0.1$, a small number of support points is sufficient to approximate the CLP for large N.
    - Suppose p and q are less than 10^{-3}, which corresponds to long overload periods of the information sources. The graph of $\log P_L(N)$ is now almost parallel to the N-axis for increasing values of N. If the load ρ is close to 1, the loss is heavy and $\log P_L(N)$ remains in between 10^{-1} to 10^{-5}. If $\rho < 0.5$, then $\log P_L(N)$ parallels the N-axis again and stays in between 10^{-5} to 10^{-12} or even less.
- For networks with heterogeneous sources:
    - Immaterial of the values for $\boldsymbol{p}$ and $\boldsymbol{q}$, it has been observed that the quantities ρ and ξ are inversely proportional. Based on this observation, the pseudocode selects the support points automatically.

To compare the model $r_n(N)$ to $\log P_L(N)$, the latter is computed using the algorithm from [10] for Section 4.1 and the algorithm from [9] for Sections 4.2 and 4.3. All numerical experiments (except for Fig. 4) have also been verified using standard Monte Carlo simulation (20 simultaneous runs). The stopping criterion for the simulation guaranteed a maximum relative error of 5 percent (except for Figs. 3, 9, and 10, where it was set to be 1 percent, and Fig. 7, where the relative error was 10 percent). The relative error was computed from the associated confidence interval, which was obtained through the usual normal approximation.

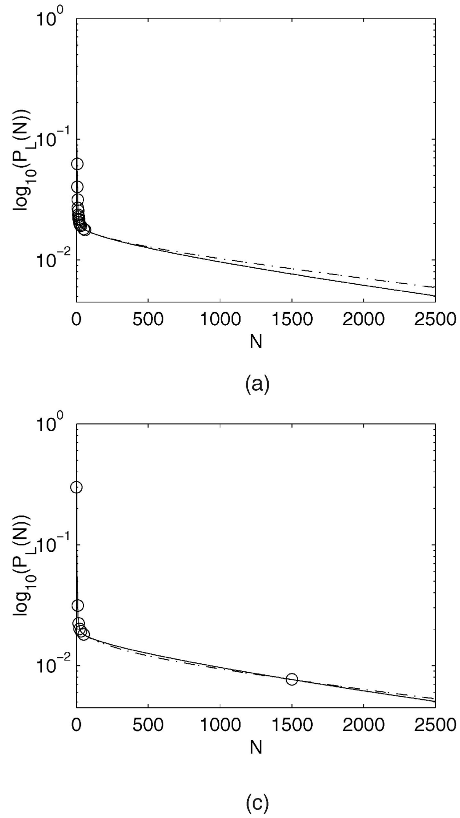


Fig. 3. CLP for Example 3.

In all figures, the values obtained at support points are circled, the computed function $\log P_L(N)$ is graphed using a full line, and the approximation $r_n(N)$ is graphed using a dotted line. An additional simulation point, used merely for validation, is denoted by a $\star$. When only the full line is visible, this means that, on the displayed figure, the approximation and the function $\log P_L(N)$ are graphically indistinguishable.

4.1 Networks with Homogeneous Sources and Single Server

In this section, we compute the CLP for networks with homogeneous sources where the server is capable of serving at most one cell during a time slot. In Table 1, we propose a pseudocode for the algorithm which chooses and adds support points automatically until the required result is achieved up to a prescribed error tolerance for $r_n(N) - r_{n+1}(N)$.

In Table 2, one finds the parameter values for the three different examples which are of interest in this section.

Example 1 (see Fig. 1a) deals with a simple case where the values for p and q are not extremely small and the system load is rather high, namely, almost 99 percent. It can easily be modeled by $r_3(N)$. On the other hand, if the decay parameter is not used as an interpolation condition, then Example 1 cannot easily be modeled accurately, not even by $r_{14}(N)$, as one can see from Fig. 1b).

Example 2 (see Fig. 2) is more difficult because of the small values of p and q. The system load is average. The simulation point confirms both the matrix-analytic computation and the rational model $r_2(N)$.

Example 3 (see Fig. 3) clearly illustrates the influence of the additional support point for large N. The value of $\log P_L(N)$ at this support point can be obtained either using a matrix-analytic technique or simulation. Situation (a) is with $N_{14} = 60$, (b) with $N_{14} = 500$, (c) with $N_{14} = 1,500$, and (d) with $N_{14} = 2,000$. The last choice is clearly the more satisfactory. In Table 3, we compare the CPU time in seconds and the exact values and approximated values of $\log P_L(N)$ for some large N values corresponding to case (d). Note that the CPU time listed for $\log P_L(N)$ relates to its computation for one value of N only, whereas the CPU time needed for the computation of $r_n(N)$ serves to obtain the full function evaluation for a wide range of N values and, hence, it is constant.

TABLE 3
Comparison of Values and CPU Times for Larger N

N	$\log P_L(N)$		CPU Time	
	Exact	[8/7](N)	Exact	[8/7](N)
2100	-2.2263	-2.2256	7092.38	6549.47
2200	-2.2441	-2.2427	7826.95	6549.47
2300	-2.2617	-2.2596	8508.55	6549.47
2400	-2.2792	-2.2764	9046.97	6549.47
2500	-2.2966	-2.2930	9933.09	6549.47
2600	-2.3138	-2.3095	10657.52	6549.47
2700	-2.3309	-2.3258	11433.24	6549.47
2800	-2.3479	-2.3421	12512.29	6549.47

Total CPU time in seconds includes computation of function values at interpolation points.

TABLE 4
Parameter Values for Examples Considered in Section 4.2

parameter	Example			
	4	5	6	7
M	10	15	25	20
p	2.19e-4	2.19e-3	2.5e-3	5e-5
q	1.1e-4	1.1e-6	1.15e-3	6e-5
d	5.5e-1	3.5e-1	6.5e-1	6.5e-1
c	5	5	15	15
ρ	0.7322	0.9998	0.742	0.5455
ξ	-7.614e-4	-1.3e-7	-7.6923e-4	-1.45e-4
$[n+1/n]$	[3/2]	[8/7]	[13/12]	[8/7]
case from Table 5	4.1.1.2	4.1.2.2	3.1	4.1.2.2
support points	1,6,11,20, 500	1,3,5,7, 9,11,13,16 18,20,23,27 30,50,500	1,2,...,16, 18,20,22,23, 25,27,29,30, 300	1,5,9,13, 16,20,23,27, 30,50,500

TABLE 5
Strategy for Networks with Homogeneous Sources and Multiple Servers

1	if $(\min(p,q) \geq 1e-1)$
1.1	if $(\lvert\xi\rvert \geq 1e-1)$
	stop
1.2	else
	$K=1;\ L=10;$ support $=\left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L\right\}$
2	elseif $(\min(p,q) \geq 1e-2)$
2.1	if $(\lvert\xi\rvert \geq 1e-1)$
	stop
2.2	else
	$K=1;\ L=20;$ support $=\left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L\right\}$
3	elseif $(\min(p,q) \geq 1e-3)$
3.1	if $(M > 15)$
	$K=1;\ L=30;$ support $=\{K, L, 300\}$
3.2	else
	$K=1;\ L=20;$ support $=\left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L\right\}$
4	elseif $(\max(p,q) < 1e-3)$
4.1	if $(\rho > 0.5)$
4.1.1	if $(\lvert\xi\rvert \geq 1e-3)$
4.1.1.1	if $(M > 20)$
	$K=1;\ L=30;$ support $=\{K, L, 300\}$
4.1.1.2	else
	$K=1;\ L=20;$ support $=\{K, L, 500\}$
4.1.2	elseif $(\lvert\xi\rvert \geq 1e-4)$
4.1.2.1	if $(M > 20)$
	$K=1;\ L=30;$ support $=\left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L, 50, 300\right\}$
4.1.2.2	else
	$K=1;\ L=30;$ support $=\left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L, 50, 500\right\}$
4.1.3	else
4.1.3.1	if $(M > 20)$
	$K=1;\ L=30;$ support $=\{K, L, 500\}$
4.1.3.2	else
	$K=1;\ L=30;$ support $=\{K, L, 800\}$
4.2	else
4.2.1	if $(\lvert\xi\rvert \geq 1e-3)$
	$K=1;\ L=40;$ support $=\left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L\right\}$
4.2.2	else
4.2.2.1	if $(M > 20)$
	$K=1;\ L=30;$ support $=\{K, L, 500\}$
4.2.2.2	else
	$K=1;\ L=30;$ support $=\{K, L, 800\}$

4.2 Networks with Homogeneous Sources and Multiple Servers

In this section, we compute the CLP for networks with homogeneous sources and c servers, each serving at most one cell during a time slot. Again we propose, in Table 5, pseudocode for the part of the algorithm that selects the support points automatically. The parameter values for the examples considered in this section are tabulated in Table 4.

Example 4 (see Fig. 4) deals with a 5-server system with average load and small p and q. Although the function $\log P_L(N)$ switches to an almost linear and slowly decreasing function before $P_L(N)$ reaches 10^{-3}, it can be modeled quite accurately by $r_2(N)$. In Table 6, we compare the CPU time in seconds and the exact values and approximated values of $\log P_L(N)$ for large N values.

An even more difficult case is that of Example 5 (see Fig. 5). Here, it is very important to obtain an accurate model because $P_L(N)$ only becomes acceptably small for a very large buffer size N.

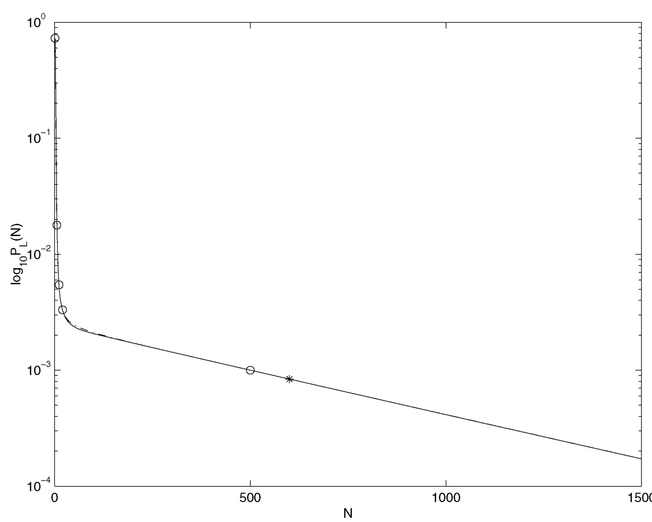


Fig. 4. CLP for Example 4.

TABLE 6
Comparison of Values and CPU Time for Larger N

N	$\log P_L(N)$		CPU Time	
	Exact	[3/2](N)	Exact	[3/2](N)
700	-3.1535	-3.1532	3472.66	1773.17
900	-3.3068	-3.3062	5418.55	1773.17
1100	-3.4596	-3.4590	8108.25	1773.17
1300	-3.6123	-3.6117	11657.1	1773.17
1500	-3.7648	-3.7642	15130.42	1773.17

Total CPU time in seconds include computation of function values at interpolation points.

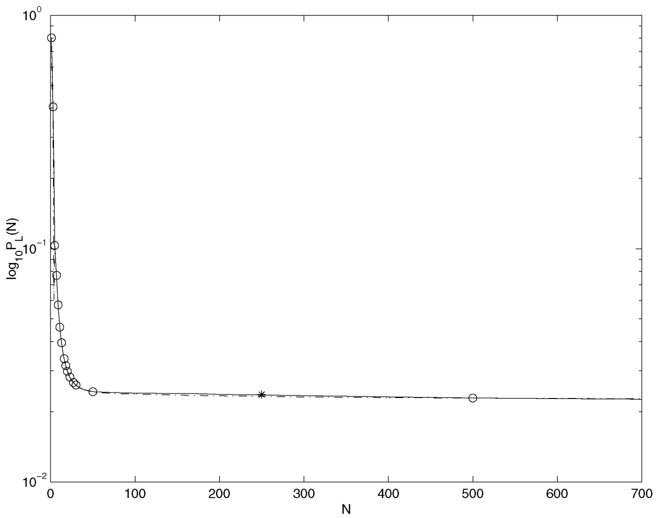


Fig. 5. CLP for Example 5.

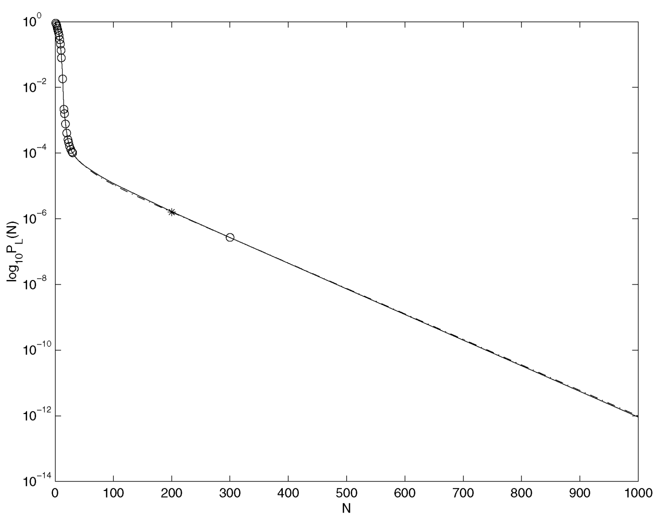


Fig. 6. CLP for Example 6.

Example 6 (see Fig. 6) illustrates a 15-server system with very high load. The aberrant behavior of $\log P_L(N)$ in the first few support points is responsible for the higher degree of the rational model, namely 12, in the denominator.

Example 7 deals with a particularly difficult situation (see Fig. 7). The system load is moderate and the values for p and q are so small that it is impossible to compute $\log P_L(N)$ analytically in a reasonable amount of time (several days on a dual Intel-Pentium 733Mhz system).

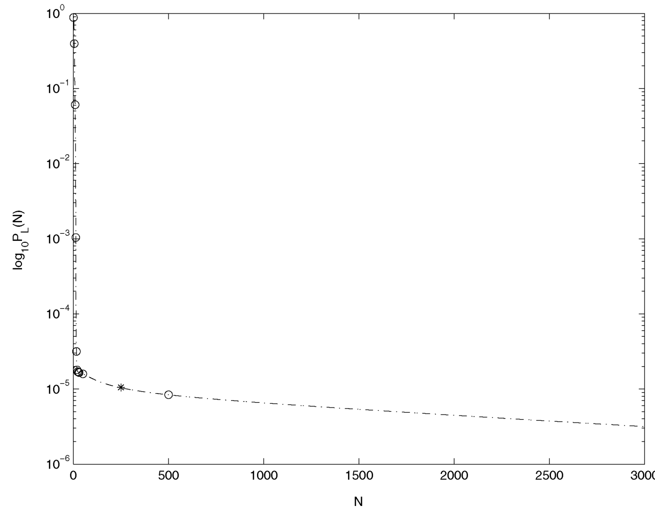


Fig. 7. CLP for Example 7.

TABLE 7
Strategy for Networks with Heterogeneous Sources

1	if $1e-1 \leq \lvert\xi\rvert < 1$
	stop
2	elseif $1e-2 \leq \lvert\xi\rvert < 1e-1$
	$K=1;\ L=20;\ \text{support} = \left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L\right\}$
3	elseif $1e-3 \leq \lvert\xi\rvert < 1e-2$
	$K=1;\ L=30;\ \text{support} = \{K, L, 300\}$
4	elseif $1e-5 \leq \lvert\xi\rvert < 1e-3$
4.1	if $\rho > 0.5$
4.1.1	if $(\max(\boldsymbol{p},\boldsymbol{q}) < 1e-3)$
	$K=1;\ L=30;\ \text{support} = \left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L, 50, 500\right\}$
4.1.2	else
	$K=1;\ L=30;\ \text{support} = \left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L, 50, 1500\right\}$
4.2	else
	$K=1;\ L=30;\ \text{support} = \{K, L, 500\}$
5	else
5.1	if $\rho > 0.5$
5.1.1	if $(\max(\boldsymbol{p},\boldsymbol{q}) < 1e-3)$
	$K=1;\ L=30;\ \text{support} = \left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L\right\}$
5.1.2	else
	$K=1;\ L=30;\ \text{support} = \left\{K, \left\lceil\frac{K+L}{2}\right\rceil, L, 50, 1500\right\}$
5.2	else
	$K=1;\ L=30;\ \text{support} = \{K, L, 500\}$

Therefore, only the function $r_7(N)$ is displayed, which is then validated by more simulation points.

4.3 Networks with Heterogeneous Sources

In this section, we compute the CLP for networks with heterogeneous sources and c servers, each serving at most one cell during a time slot. For this type of network, we propose the pseudocode in Table 7.

For the examples discussed in this section, the parameter values are tabulated in Table 8. Typical values and CPU times for Example 8 are given in Table 9.

Example 8 (see Fig. 8) is a typical example of packet loss probabilities where p_i and q_i are very small while the load is still more than 80 percent. This case is interesting because it deals with a true real-world situation. The function $\log P_L(N)$ switches to a slowly decreasing function for

TABLE 8
Parameter Values for Examples Considered in Section 4.3

parameter	Example		
	8	8	10
M	5	6	6
$\boldsymbol{p}$	$\begin{pmatrix} 6.984e\text{-}5 \\ 2.1e\text{-}7 \\ 8.366e\text{-}5 \\ 8.8894e\text{-}5 \\ 1.98e\text{-}6 \end{pmatrix}$	$\begin{pmatrix} 0.1115 \\ 0.0731 \\ 0.0001 \\ 0.1252 \\ 0.12 \\ 0.1392 \end{pmatrix}$	$\begin{pmatrix} 3.478e\text{-}6 \\ 3.03e\text{-}7 \\ 4.697e\text{-}6 \\ 8.698e\text{-}6 \\ 3.691e\text{-}6 \\ 3.63e\text{-}6 \end{pmatrix}$
$\boldsymbol{q}$	$\begin{pmatrix} 9.84e\text{-}6 \\ 3.742e\text{-}5 \\ 9.675e\text{-}5 \\ 6.196e\text{-}5 \\ 6.7e\text{-}5 \end{pmatrix}$	$\begin{pmatrix} 0.1486 \\ 0.0731 \\ 0.0001 \\ 0.1252 \\ 0.12 \\ 0.1392 \end{pmatrix}$	$\begin{pmatrix} 0.657e\text{-}7 \\ 5.901e\text{-}6 \\ 2.662e\text{-}6 \\ 6.519e\text{-}6 \\ 1.045e\text{-}6 \\ 5.21e\text{-}6 \end{pmatrix}$
$\boldsymbol{d}$	$\begin{pmatrix} 0.4562 & 0.2953 \\ 0.8380 & 0.6022 \\ 0.8231 & 0.1828 \\ 0.5421 & 0.7332 \\ 0.0924 & 0.5489 \end{pmatrix}$	$\begin{pmatrix} 0.1486 & 0.1528 \\ 0.0975 & 0.1721 \\ 0.0001 & 0.2528 \\ 0.1670 & 0.3084 \\ 0.1599 & 0.3071 \\ 0.1856 & 0.0010 \end{pmatrix}$	$\begin{pmatrix} 0.2435 & 0.6723 \\ 0.2105 & 0.3941 \\ 0.0234 & 0.6943 \\ 0.2925 & 0.2596 \\ 0.2016 & 0.1597 \\ 0.1034 & 0.2492 \end{pmatrix}$
c	3	1	2
ρ	0.8128	0.9764	0.9406
ξ	-2.271e-4	-8.945e-5	-3.3e-6
$[n+1/n]$	[7/6]	[14/13]	[4/3]
case from Table 7	4.1.1	4.1.2	5.1.1
support points	1,5,9,13,16,18,20, 23,25,27,30,50,500	1,3,5,7,9,11, 13,…,30,50,1500	1,5,9,16, 23,27,30

average to large N. Yet it can be modeled fully automatically and accurately by $r_6(N)$.

Example 9 (see Fig. 9) shows that, even for large p_i, q_i, and d_i, the graph of $\log P_L(N)$ can be almost linear. The decay rate is close to zero, unlike for a situation with homogeneous sources.

In Example 10 (see Fig. 10), the same effect can be observed for very small p_i and q_i. But, our technique catches $\log P_L(N)$ perfectly, using only seven support points and the decay rate.

5 Conclusion and Future Work

From the examples in the previous section, it is clear that the method is successful. The function $\log P_L(N)$ can be accurately fitted by a rational interpolant of sufficiently low degree in all cases. Of course, the situation where one is dealing with homogeneous sources is easier to deal with than that with heterogeneous sources. The novelty is that we have been able to propose a single rational interpolation technique for $\log P_L(N)$ that is able to model all cases equally well. Whether the parameters p and q or $\boldsymbol{p}$ and $\boldsymbol{q}$ are very small or rather large, whether the load of the system is

TABLE 9
Comparison of Values and CPU Time for Larger N

N	$\log P_L(N)$		CPU Time	
	Exact	[7/6](N)	Exact	[7/6](N)
600	-3.1339	-3.1316	3461.63	2711.00
700	-3.1628	-3.1580	4500.41	2711.00
800	-3.1913	-3.1817	5484.59	2711.00
900	-3.2195	-3.2091	6922.07	2711.00
1000	-3.2474	-3.2340	8188.07	2711.00

Total CPU time in seconds includes computation of function values at interpolation points.

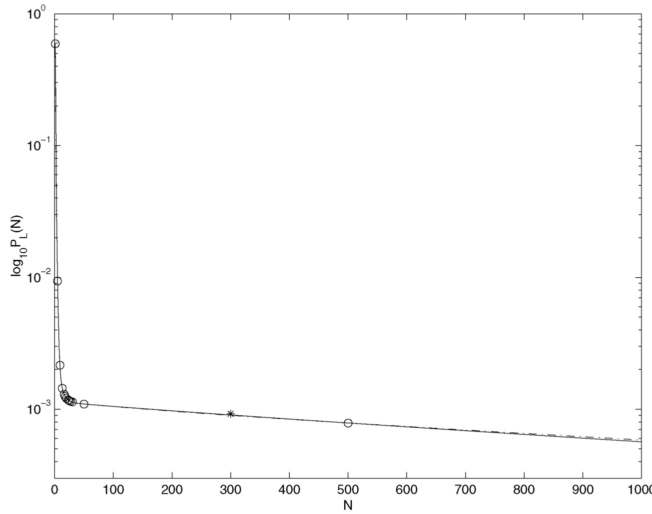


Fig. 8. CLP for Example 8.

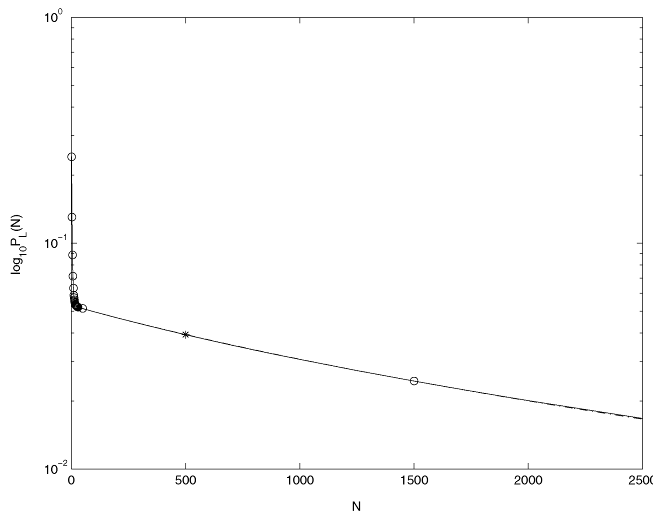


Fig. 9. CLP for Example 9.

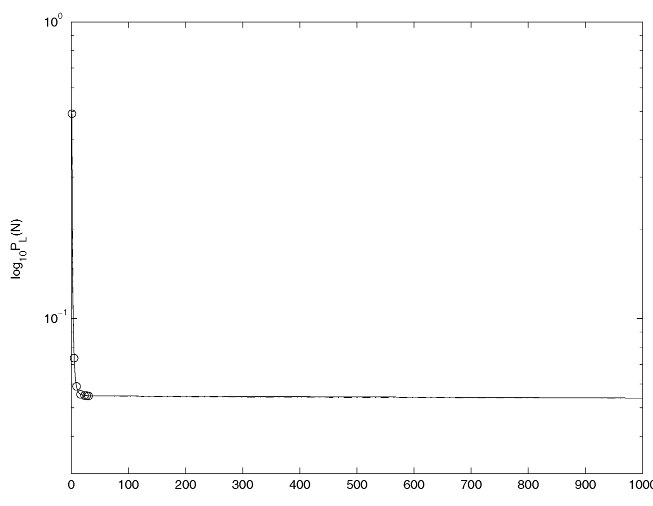


Fig. 10. CLP for Example 10.

low, average, or high, the algorithm finds the correct support points and delivers an approximation for $\log P_L(N)$ within a specified error tolerance.

The attentive reader may have noticed that, in none of the examples, were we bothered by the poles of the rational interpolant, which nevertheless are free. On one hand, the stopping criterion

$$\max_{N=K,\ldots,L} ||r_n(N) - r_{n+1}(N)|| < \varepsilon$$

ensures that, if r_{n+1} has unexpected poles, then the condition will not be satisfied. On the other hand, the technique could be enhanced with an optimal pole assignment procedure, which is the subject of further research.

REFERENCES

[1] A. Baiocchi, "Analysis of the Loss Probability of the MAP/G/1/K Queue, Part I: Asymptotic Theory," *Comm. Statistical Stochastic Models,* vol. 10, no. 4, pp. 867-893, 1994.

[2] C. Blondia, "A Discrete-Time Batch Markovian Arrival Process as B-ISDN Traffic Model," *Belgian J. Operations Research, Statistics and Computer Science,* vol. 32, nos. 3,4, 1992.

[3] C. Blondia and O. Casals, "Performance Analysis of Statistical Multiplexing of VBR Sources: A Matrix-Analytical Approach," *Performance Evaluation,* vol. 16, pp. 5-20, 1992.

[4] C.S. Chang, P. Heidelberger, S. Juneja, and P. Shahabuddin, "Effective Bandwidth and Fast Simulation of ATM Intree Networks," *Proc. Performance '93,* Oct. 1993.

[5] A.I. Elwalid and D. Mitra, "Analysis, Approximations and Admission Control of a Multi-Service Multiplexing System with Priorities," *Proc. INFOCOM '95,* pp. 463-472, 1995.

[6] E. Falkenberg, "On the Asymptotic Behavior of the Stationary Distribution of Markov Chains of M/G/1-Type," *Comm. Statistical Stochastic Models,* vol. 10, pp. 75-97, 1994.

[7] J. Garcia and O. Casals, "A Discrete Time Queueing Model to Study the Cell Delay Variation in an ATM Network," *Performance Evaluation,* vol. 21, pp. 3-22, 1994.

[8] W.B. Gong and H. Yang, "On Global Rational Approximants for Stochastic Discrete Event Systems," *Int'l J. Discrete Event Dynamic Systems,* vol. 7, no. 1, 1997.

[9] A.E. Kamal, "Efficient Solution of Multiple Server Queues with Application to the Modeling of ATM Concentrators," *Proc. IEEE Infocom '96,* pp. 248-254, Mar. 1996

[10] G. Kemeny and J. Snell, *Finite Markov Chains.* New York: van Nostrand-Reinhold, 1960.

[11] Z. Liu, P. Nain, and D. Towsley, "Exponentially Bounds with an Application to Call Admission," Technical Report TR94-63, Computer Science Dept., Univ. of Massachusetts, Amherst, Oct. 1994.

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

[13] T. Thiele, *Interpolationsrechnung.* Leipzig: Teubner, 1909.

[14] D. Warner, "Hermite Interpolation with Rational Functions," PhD thesis, Univ. of California, 1974.

[15] H. Wallin, "Potential Theory and Approximation of Analytic Functions by Rational Interpolation," *Proc. Colloquium Compl. Anal. at Joensuu, Lecture Notes in Math.,* vol. 747, pp. 434-450, 1979.

[16] J.L. Walsh, *Interpolation and Approximation by Rational Functions in the Complex Domain.* Providence, R.I.: Am. Math. Soc. Press, 1969.

[17] S. Witterrongel and H. Bruneel, "Discrete-Time Queues with Correlated Arrivals and Constant Service Times," *Computers and Operations Research,* vol. 26, pp. 93-108, 1999.

[18] K. Wuyts and R.K. Boel, "A Matrix Geometric Algorithm for Finite Buffer Systems with B-ISDN Applications," *Proc. ITC Specialists Seminar Control in Comm.,* pp. 265-276, 1996.

[19] K. Wuyts and R.K. Boel, "Efficient Matrix Geometric Techniques for Performance Evaluation of ATM Buffers, Using Kronecker Product Structures and Spectral Decomposition," submitted, 1997.

[20] Y. Xiong and H. Bruneel, "A Unifying Queueing Model for ATM and Its Analysis," *Int'l J. Comm. Systems,* vol. 9, pp. 253-267, 1996.

[21] H. Yang, "Global Rational Approximation for Computer Systems and Communication Networks," PhD thesis, Computer Science Dept., Univ. of Massachusetts, Amherst, 1996.

Annie Cuyt received the Doctor Scientiae degree in 1982 from the University of Antwerp (UIA), Belgium, where she now teaches several computing courses. She was appointed research director of the FWO-Vlaanderen, the Flemish Science Foundation. She is the author of more than 90 publications in international journals and conference proceedings and the author or editor of several books. Her current interests are in reliable computing and rational approximation theory. She is an editorial board member of the journal *Reliable Computing*. Since 1997, she has also served as a member of the scientific committee of the Flemish Science Foundation.

R.B. Lenin received the BSc degree in mathematics from Madras University, Chennai, India, in 1992. He received the MSc and PhD degrees in mathematics from the Indian Institute of Technology Madras, Chennai, India, in 1994 and 1998, respectively. He is the author of more than 20 publications in international journals and conference proceedings. He worked as a postdoctoral fellow at the University of Twente, The Netherlands, during 1998-1999 on the project "Stochastic Networks" sponsored by NWO, The Netherlands. Since 1999, he has worked as a postdoctoral fellow at the University of Antwerp, Belgium, on the project "Computational Methods for Performance Evaluation and Simulation of Complex Technical Systems," funded by FWO (Flemish Division), Belgium. His recent research interests include extrapolation methods and the computation of rational interpolants with prescribed poles with applications to computer systems and communication networks.

Gert Willems graduated in mathematics from the University of Antwerp, Belgium, in 2000. He is currently a research assistant and PhD student in the Department of Mathematics and Computer Science at the University of Antwerp. His research interests include modeling and simulation of communication networks and computational statistics.

Chris Blondia obtained the PhD degree in mathematics from the University of Ghent (Belgium) in 1982. Between 1986 and 1991, he was a researcher at the Philips Research Laboratory and, from 1991 to 1994, he was an associate professor in the Computer Science Department of the University of Nijmegen (The Netherlands). In 1995, he joined the Department of Mathematics and Computer Science of the University of Antwerp, where he is currently a professor, head of the research group "Performance Analysis of Telecommunication Systems" (PATS), and head of the department. His main research interests are related to both theoretical methods for stochastic modeling (in particular queuing systems) and to the design, modeling, and performance evaluation of telecommunication systems, in particular related to traffic management in broadband networks, IP mobility management, medium access control for wireless and wired access networks, etc. He has published a substantial number of papers in international journals and conference proceedings. He is editor of the *Journal of Network and Computer Applications* and member of IFIP W.G. 6.3 on "Performance of Computer Networks." He has been involved in several European research programs (RACE, ACTS , IST and COST).

Peter Rousseeuw studied mathematics at the Vrije Universiteit Brussel (VUB), Belgium, and did his PhD research in statistics at the Swiss Institute of Technology (ETH) in Zurich, Switzerland. His PhD thesis in 1981 was about robust estimators and tests. Afterward, he performed research on robust regression analysis and on algorithms for cluster analysis. From 1984 on, he was a professor of statistics at the Technical University of Delft (The Netherlands), the University of Fribourg (Switzerland), and (since 1990) at the University of Antwerp. He has supervised 16 PhD dissertations to date. He has authored three books (with Wiley, New York) and more than 150 papers in international journals. Several methods and algorithms he developed are now included in the software packages SAS and S-Plus. He is a fellow of the Institute of Mathematical Statistics, the International Statistical Institute, and the American Statistical Association.