Performance Evaluation 67 (2010) 160–173

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Computing packet loss probabilities of D-BMAP/PH/1/N queues with group services

R.B. Lenin [a,*], A. Cuyt [c], K. Yoshigoe [b], S. Ramaswamy [b]

[a] Department of Mathematics, University of Central Arkansas, Conway, AR 72035, USA
[b] Department of Computer Science, University of Arkansas at Little Rock, Little Rock, AR 72204, U.S.A
[c] Department of Mathematics and Computer Science, University of Antwerp, Campus Middelheim, Middelheimlaan 1, B-2020 Antwerp, Belgium

ARTICLE INFO

Article history:
Received 30 October 2008
Received in revised form 28 August 2009
Accepted 7 October 2009
Available online 22 October 2009

Keywords:
$M/G/1$-type queues
D-BMAP arrival process
Discrete phase-type service
Matrix-analytic methods
Newton–Padé-type approximation

ABSTRACT

Statistical multiplexers have been integral components of packet switches and routers on data networks. They are modeled as queueing systems with a finite buffer space, served by one or more transmission links of fixed or varying capacity. The service structure typically admits packets of multiple sources on a first-come first-serve (FCFS) basis. In this paper, we adhere to D-BMAP/PH/1/N queues with discrete phase-type group service channel which allows the packets to get service in the service channel for a random number of time slots by staying in different phases of the service channel before they leave the switch. The aim is to determine the packet loss probability (PLP) as a function of the capacity of the buffer. Due to the curse of dimensionality of the mathematical model, the numerical computation of the performance measures using the analytical formulas is time and memory consuming. Due to rare events, getting the performance measures by simulation is again time consuming. To overcome this problem, we use the Newton–Padé-type rational approximation technique to compute the PLP more efficiently. Since this technique needs the asymptotic behavior of the PLP, we show a way to regroup the elements of the TPM to obtain the asymptotic behavior of the PLP.

1. Introduction

The Internet is primarily TCP/IP with variable length packets instead of fixed length packets. Variable bit rate (VBR) 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. The introduction of statistical multiplexing techniques offers the capability to efficiently support VBR connections by taking advantage of the variability of the bandwidth requirements of individual connections.

Statistical multiplexers have been integral components in packet switches and routers on data networks. They have gained increased prominence since 1990 with the availability of broadband transmission speeds exceeding 155 Mbps and ranging up to 10 Gbps in the core of the network. The characterization of network traffic with parametric models is a basic requirement to engineer communication networks. Statistical multiplexers in particular are modeled as queueing systems with finite buffer space, served by one or more transmission links of fixed or varying capacity. The service structure typically admits packets of multiple sources on a first-come first-serve (FCFS) basis. The statistical multiplexing gain is an important performance metric that quantifies the multiplexing efficiency.

* Corresponding author. Tel.: +1 5015698130; fax: +1 5015698144.
E-mail addresses: rblenin@uca.edu (R.B. Lenin), annie.cuyt@ua.ac.be (A. Cuyt), kxyoshigoe@ualr.edu (K. Yoshigoe), sxramaswamy@ualr.edu (S. Ramaswamy).

0166-5316/$ – see front matter
doi:10.1016/j.peva.2009.10.001

Next generation Internet services such as on demand high-quality video streaming service will heavily consume the bandwidth of the Internet backbone with their broadcast and multicast traffic. Thus, packet switches, being the core of the Internet, need to efficiently support ever increasing multicast traffic [1].

Two call splitting policies, namely call splitting and no call splitting, are possible for switching multicast packets. No call splitting, or one-shot policy, is the forwarding of a multicast cell to its all destinations at a single time slot. Since transmission will not be allowed until all associated output ports become available, no call splitting can add significant delay to packets with high degree of multicasting. Call splitting, on the other hand, allows forwarding of multicast cells to a set of available output ports in one time slot, and the remaining multicast cells will be forwarded to the rest of the output ports whenever the switch resource associated to them becomes available.

The proposed phase-type service can be used to describe the behavior of a packet switch that supports multicast packets via call splitting. Each packet arriving at an input port of packet switch can have one (in case of a unicast packet) to at most M_s output destinations depending on the degree of multicasting. In our model, each packet is first assigned to a phase i with some probability where $M_s - i + 1$ represents a remaining number of multicasting output ports. Every time a full or partial forwarding of the multicast packet from the input port to output port is completed, the assignment is shifted toward phase M_s with a certain transition probability from phase j to phase k for $k \geq j$ with probability s_{jk}. The process continues until the packet leaves the system at that time the multicast packet is removed from the head-of-line queue.

In order to access the multiplexing gain, a variety of techniques have been developed in recent years, based on exact analysis, approximate analysis and simulation.

In exact analysis, the traffic is described by Markov-modulated arrival processes (MMAP), leading to a Markov model of M/G/1-type [2–6]. These processes find many applications in computer and communication systems. For example, special cases of this model are used to model voice, data and video traffic sources [7–10]. Unfortunately, the curse of dimensionality weighs heavily on many branches of applied probability, and queueing theory is no exception. The complexity of the algorithms used to find the stationary probabilities of M/G/1-type queues is $O(c(M+1)^3 M_S^3 N^2)$, where c is the maximum number of packets the server can serve, M is the number of information sources from which packets are arriving to the system, M_S is the number of phases in the service channel, and N is the buffer size of the queue. This order of complexity does not allow one to compute the PLP for large buffers in real time. Hence, even for special cases of this arrival process, computing the PLP is intensive and impractical, especially when the state space of the aggregate arrival process is large [11]. Another Markov-modulated source that has extensively been used to model various types of traffic is the Markov-modulated fluid (MMF) source [12,11]. In the MMF, information is generated and processed as a continuous (fluid) flow at a rate which depends on the state of the background Markov process. The advantage of this model over traditional queueing models is that the numerical complexity is independent of the buffer size. However, unlike the case of MMAP, where the discrete nature of the packet is preserved, the fluid is unable to capture the effect of cell variability.

Monte Carlo simulation is also used to compute the PLP. If the desired PLP 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 PLP [13,14]. However, because of the complicated nature of multiplexer queueing models, applying the IS technique is not straightforward.

In this paper, we adhere to the discrete phase-type service channel which allows the packets to get service in the service channel for a random number of time slots by staying in different phases of the service channel before they leave the switch unlike the queueing model proposed in [6] where a deterministic service distribution is used to serve the packets. In Section 2, we give the closed form expression for the TPM of the underlying Markov chain. The aim is to determine the packet loss probability $P_L(N)$ as a function of the capacity of the buffer N. In Section 2.3, we find a way to regroup the elements of the TPM to obtain the asymptotic behavior of $P_L(N)$. By suitably modifying the technique proposed in [4], we compute the $P_L(N)$ of the TPM of the queueing model under investigation.

The technique proposed in [6] is a kind of "divide and conquer" technique, in the sense that

- for small values of the buffer capacity N the exact value $P_L(N)$ is computed,
- the function $\log P_L(N)$ is approximated by a suitable rational function $r_n(N)$,
- and the approximate model is validated by simulation for one larger value of the buffer length.

The motivation to compute the PLP using rational approximation comes from the works of Gong and Yang [15] and Yang [16]. They have computed these probabilities for large buffer sizes, from sampled values of $\log P_L(N)$ for small buffer sizes and the decay rate of the loss probability. The technique was applied to multiplexer models with little or no correlation between the cells.

In [6], an automatic procedure to select the sample points (also called support points) is proposed and used for the efficient computation of models $r_n(N)$ in case there is more correlation between the cells. The procedure selects the support points in a region which we determine from the system parameters, until the model $r_n(N)$ is sufficiently accurate, meaning that $|r_n(N) - r_{n+1}(N)|/|r_{n+1}(N)|$ does not exceed a prescribed error threshold. But when encountering positive real poles in the model $r_n(N)$, the procedure has to add more support points and increase n to achieve the desired accuracy.

Here this technique is perfected: besides making use of the slope of the function $\log P_L(N)$ for $N \to \infty$, as explained in Section 2.3, we show in Section 3.2 how one can make very good use of the knowledge of $P_L(N)$ for extremely small values of N. The present approach allows us to construct a model $r_n(N)$ which is free from positive real poles, and is very efficient.

In our model we maximize the information that can be extracted from the data, while minimizing the number of data samples to be collected.

2. D-BMAP/PH/1/N queue with batch service

2.1. Model description

In the multiplexer environment, the arrival of packets to the switch happens in discrete time, and because the service time is discrete, the discrete time Markov chain is a natural modeling choice. We assume that the arrival of packets, which are transmitted by M independent and non-identical information sources to the multiplexer, can be modeled as a discrete time batch Markovian arrival process (D-BMAP), the discrete time version of BMAP. Each information source is controlled by a 2-state (ON and OFF) Markov chain, called the background Markov chain. Each of these background Markov chains has the following TPM

$$\mathbf{Q} = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}, \tag{1}$$

where p (q) is the probability that the chain is changing from OFF to ON state (ON to OFF state).

As an example for an ON–OFF information source, we consider the following practical situation: Consider a population of voice messages serviced by a single T1 channel (whose bandwidth is 1.536 Mbps). We discretize time into 16 ms slots.

$$\text{The mean ON period} = 352 \text{ ms} = \frac{352}{16} \text{ time slots} = 22 \text{ time slots} = \frac{1}{q}.$$

$$\text{Therefore } q = \frac{1}{22} \text{ time slots} \approx 0.04546 \text{ time slots}.$$

$$\text{The mean OFF period} = 650 \text{ ms} = \frac{650}{16} \text{ time slots} = \frac{1}{p}.$$

$$\text{Therefore } p = \frac{16}{650} \text{ time slots} \approx 0.02461 \text{ time slots}.$$

Hence each voice source can be modeled as a homogeneous ON–OFF source with a background Markov chain whose transition matrix is given by [16]:

$$\begin{pmatrix} 0.975 & 0.025 \\ 0.045 & 0.955 \end{pmatrix} \tag{2}$$

where p and q values are substituted in (1) to get the above matrix.

The basic queueing system which models the multiplexer is a D-BMAP/PH/1/N queue with one discrete time server who offers service to groups of varying size and the service times are assumed to be of phase type. When the buffer contains more than c packets, a maximum number of c packets join the service channel.

The service time of the packets has a common phase-type distribution function [17] with a matrix representation $(M_s, \boldsymbol{\alpha}, \boldsymbol{S})$, where M_s is a positive integer denoting the number of phases in the service channel, $\boldsymbol{\alpha}$ is a $1 \times M_s$ nonnegative stochastic vector and $\boldsymbol{S} = (s_{ij})$ is an $M_s \times M_s$ sub-stochastic matrix. The ith component of the vector $\boldsymbol{\alpha}$ is the probability that the packets which join the service channel start being serviced in phase i. With probability s_{ij} these packets change from phase i to phase j in the next time slot. The jth element of the vector $(\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e}$ is the probability that these packets leave the service channel at the jth phase. Here $\boldsymbol{I}$ is an $M_s \times M_s$ identity matrix and $\boldsymbol{e}$ is an $M_s \times 1$ vector with all entries equal to one. The mean service rate is given by

$$\mu = \frac{c}{\boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e}}. \tag{3}$$

When the server is busy servicing the packets (at most c packets), other arriving packets wait in the buffer of capacity N, including the number of packets in the service channel even if the server holds less than c packets. We define $m_{tot} := (M+1)M_s$.

2.2. PLP $P_L(N)$

The D-BMAP/PH/1/N queueing model with group service is basically a Markov chain (MC) with a finite number of states labeled $0, 1, \ldots, (M+1) + Nm_{tot}$. The set of states $\{0, 1, \ldots, M\}$ is referred to as level zero of the MC, whereas the set of states $\{(M+1) + (i-1)m_{tot}, \ldots, (M+1) + im_{tot} - 1\}$ is referred to as level i of the MC for $1 \le i \le N$. The states of level i, with $1 \le i \le N$, are labeled (s, j), where $1 \le s \le M_s$ and $0 \le j \le M$. The state (s, j) of level i corresponds to the situation in which there are packets (at most c packets) in phase s of the service channel and the D-BMAP arrival process is in state j. State j of level zero corresponds to the situation in which the queue and the server are empty while the current state of the D-BMAP arrival process is j. We are interested in computing the stationary probabilities of the MC.

This MC can easily be extended to a model with heterogeneous sources. But in the heterogeneous case the computation of the stationary probability becomes very expensive as there are totally $2^M + 2^M NM_s$ states (see for example [6]). So we restrict our analysis to models with M homogeneous sources. Let $\boldsymbol{D}$ be the TPM of the D-BMAP arrival process and let $\boldsymbol{D_i}$ $(i = 0, 1, \ldots, M)$ denote the matrix corresponding to i arrivals during a time slot. These matrices are of dimension $(M+1) \times (M+1)$ and can be calculated from the following system parameters [6, section 2.2]:

1. the number of sources M,
2. the transition probabilities p and q of the background Markov chains, and
3. the packet generation probability d.

The average arrival rate of packets at the multiplexer is given by

$$\lambda = \boldsymbol{\eta} \left(\sum_{i=0}^{M} i\boldsymbol{D_i} \right) \boldsymbol{e}, \tag{4}$$

where $\boldsymbol{e}$ is a column vector of ones and the vector $\boldsymbol{\eta}$ is such that $\boldsymbol{\eta D} = \boldsymbol{\eta}$ with $\boldsymbol{\eta e} = 1$.

Under the condition of ergodicity of the MC, i.e. the load $\rho = \lambda/(c\mu) < 1$, the stochastic stationary distribution vector $\boldsymbol{\Pi}$ satisfies

$$\boldsymbol{\Pi P} = \boldsymbol{\Pi}, \tag{5}$$

where the (i, j)th element $\boldsymbol{P_{ij}}$ of the transition probability matrix (TPM) $\boldsymbol{P}$ of the MC is given by (6), (7) and (8). The stationary distribution $\boldsymbol{\Pi}$ is given by

$$\boldsymbol{\Pi} = (\boldsymbol{\pi_0}, \ldots, \boldsymbol{\pi_N}),$$

where

$$\boldsymbol{\pi_0} = (\pi_{00}, \ldots, \pi_{0M}) \in \mathbb{R}^{M+1}$$

with

$$\pi_{0j} = P\{\text{buffer is empty and the arrival process is in state } j\}$$

and

$$\boldsymbol{\pi_i} = \left(\boldsymbol{\pi_i^1}, \ldots, \boldsymbol{\pi_i^{M_s}}\right) \in \mathbb{R}^{m_{tot}}, \quad 1 \le i \le N,$$

where

$$\boldsymbol{\pi_i^s} = \left(\pi_{i0}^s, \ldots, \pi_{iM}^s\right) \in \mathbb{R}^{M+1}$$

with

$$\pi_{ij}^s = P\{\text{buffer has } i \text{ packets while the arrival process is in state } j \text{ and the packets are being served in phase } s\},$$

for $1 \le i \le N;\ 0 \le j \le M;\ 1 \le s \le M_s$.

The transition probabilities are given by

$$\boldsymbol{P_{0j}} = \begin{cases} \boldsymbol{D_0}, & j = 0 \\ \boldsymbol{D_j} \bigotimes \boldsymbol{\alpha}, & j = 1, \ldots, N-1, \end{cases} \tag{6}$$

$$\boldsymbol{P_{ij}} = \begin{cases} \boldsymbol{D_0} \bigotimes (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e}, & j = 0 \\ \boldsymbol{D_j} \bigotimes (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e\alpha}, & j = 1, 2, \ldots, i-1 \\ \boldsymbol{D_{j-i}} \bigotimes \boldsymbol{S} + \boldsymbol{D_j} \bigotimes (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e\alpha}, & j = i, i+1, \ldots, N-1 \\ \displaystyle\sum_{k=j-i}^{M} \boldsymbol{D_k} \bigotimes \boldsymbol{S} + \sum_{k=j}^{M} \boldsymbol{D_k} \bigotimes (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e\alpha}, & j = N, \end{cases} \tag{7}$$

for $i = 1, 2, \ldots, c$, and

$$\boldsymbol{P_{ij}} = \begin{cases} \boldsymbol{0}, & j = 0, 1, \ldots, i-c-1 \\ \boldsymbol{D_{j-i+c}} \bigotimes (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e\alpha}, & j = i-c, \ldots, i-1 \\ \boldsymbol{D_{j-i}} \bigotimes \boldsymbol{S} + \boldsymbol{D_{j-i+c}} \bigotimes (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e\alpha}, & j = i, i+1, \ldots, N-1 \\ \displaystyle\sum_{k=j-i}^{M} \boldsymbol{D_k} \bigotimes \boldsymbol{S} + \sum_{k=j-i+c}^{M} \boldsymbol{D_k} \bigotimes (\boldsymbol{I} - \boldsymbol{S})\boldsymbol{e\alpha}, & j = N, \end{cases} \tag{8}$$

for $i = c+1, c+2, \ldots, N$. We use the convention that the summations $(\sum)$ in (7) and (8) will be zero if the lower limit of the running index is greater than the upper limit.

Eqs. (6)–(8) are self-explanatory. For example, the transition from level 0 to level 0 implies that the system is empty in the previous time slot and it remains empty in the current time slot. This can happen when there is no arrival. Hence $\boldsymbol{P_{00}} = \boldsymbol{D_0}$. The transition from level 1 to level N happens in case of any of the following events:

1. One packet was being serviced in the previous time slot and it continues to be in service in the current time slot, but in a different phase with probability matrix $\boldsymbol{S}$. Several situations can occur:
   – We have $N-1$ arrivals in the current time slot, an event of probability $\boldsymbol{D_{N-1}} \bigotimes \boldsymbol{S}$.
   – We have N arrivals in the current time slot, an event of probability $\boldsymbol{D_N} \bigotimes \boldsymbol{S}$.
   $\vdots$
   – We have M arrivals in the current time slot, an event of probability $\boldsymbol{D_M} \bigotimes \boldsymbol{S}$.
2. One packet was being serviced in the previous time slot and it leaves the service in the current time slot with probability vector $(\boldsymbol{I}-\boldsymbol{S})\boldsymbol{e}$. Again different situations can occur:
   – In the current time slot N arrivals occur of which c join the service channel with probability vector $\boldsymbol{\alpha}$. The probability of this event is $\boldsymbol{D_N} \bigotimes (\boldsymbol{I}-\boldsymbol{S})\boldsymbol{e\alpha}$.
   – In the current time slot $N+1$ arrivals occur of which c join the service channel with probability vector $\boldsymbol{\alpha}$. The probability of this event is $\boldsymbol{D_{N+1}} \bigotimes (\boldsymbol{I}-\boldsymbol{S})\boldsymbol{e\alpha}$.
   $\vdots$
   – In the current time slot M arrivals occur of which c join the service channel with probability vector $\boldsymbol{\alpha}$. The probability of this event is $\boldsymbol{D_M} \bigotimes (\boldsymbol{I}-\boldsymbol{S})\boldsymbol{e\alpha}$.

Hence the $(1, N)$th element of $\boldsymbol{P}$ is given by

$$\boldsymbol{P_{1N}} = \sum_{k=N-1}^{M} \boldsymbol{D_k} \bigotimes \boldsymbol{S} + \sum_{k=N}^{M} \boldsymbol{D_k} \bigotimes (\boldsymbol{I}-\boldsymbol{S})\boldsymbol{e\alpha},$$

which is what we get if we substitute $i = 1$ and $j = N$ in (7). We note that the above discussion is valid only when $N < M$. When $N > M$, the jumping of the underlying process from state 1 to state N is not possible and hence $\boldsymbol{P_{1N}} = 0$. This is taken care by the summation convention used in (7) and (8).

The matrix $\boldsymbol{P}$ is a square matrix of order $(M+1) + Nm_{tot}$. The PLP function $P_L(N)$, as a function of the buffer size N, is then given by

$$P_L(N) = \frac{1}{\lambda}\left[\boldsymbol{\pi_0}\sum_{m=0}^{M}[m-N]^{+}\boldsymbol{D_k e} + \sum_{n=1}^{N}\left(\sum_{s=1}^{M_s}\boldsymbol{\pi_n^s}\right)\left(\sum_{m=0}^{M}[m+n-N]^{+}\boldsymbol{D_k e}\right)\right], \tag{9}$$

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

2.3. Asymptotic behavior of $\log P_L(N)$

In the literature, the loss probability $P_L(N)$ in a queueing system with finite buffer size N is often approximated by $P\{Q \geq N\}$, the tail of the queue length distribution in the corresponding infinite buffer queueing system [10]. For infinite buffer queueing systems, it has been shown that $P\{Q \geq N\}$ is asymptotically exponential [18], that is,

$$P\{Q \geq N\} \sim Ae^{\xi N}, \quad N \to \infty.$$

Here ξ is a negative constant called the asymptotic decay rate and A is a positive constant called the asymptotic constant. The classical effective bandwidth approximation assumes that the constant A is 1 [19]. Hence

$$\log P_L(N) \sim \xi N, \quad N \to \infty, \tag{10}$$

where the decay rate ξ is the Perron–Frobenius eigenvalue of the matrix

$$\boldsymbol{A}(z) := \sum_{n=0}^{K}\boldsymbol{A_n}z^n, \quad 0 < z < R_A \tag{11}$$

satisfying the condition $\xi = z$. Here R_A is the radius of convergence of $\boldsymbol{A}(z)$, $K = \lceil \frac{M+c}{c} \rceil$ and

$$\boldsymbol{A_k} := (\boldsymbol{P_{ij}})_{j=kc+1,\ldots,(k+1)c}^{i=c+1,\ldots,2c}, \quad k = 0, 1, \ldots, K.$$

With the regrouping using the matrices $\boldsymbol{A_i}$, the structure of the TPM $\boldsymbol{P}$ is the same as that of the TPM of an M/G/1-type queue and hence we are justified to use the asymptotic exponential decay behavior [18].

It is well known that for Markov-modulated arrival processes there are typically two main regions in which increasing the buffer size reduces the cell loss — the "cell region" and the "burst region" as depicted in Fig. 1. In the cell region, the main component contributing to the loss rate is the cell variability (correlation) within the modulated process. As the buffer size is increased, this variability gets absorbed and the loss rapidly decreases. In the burst region, the loss due to cell variability is negligible and the loss is mainly due to the fact that the rate in one or more of the states (overloaded states) is greater than the link capacity μ. The asymptotic slope of the burst region is the same as the decay rate ξ.

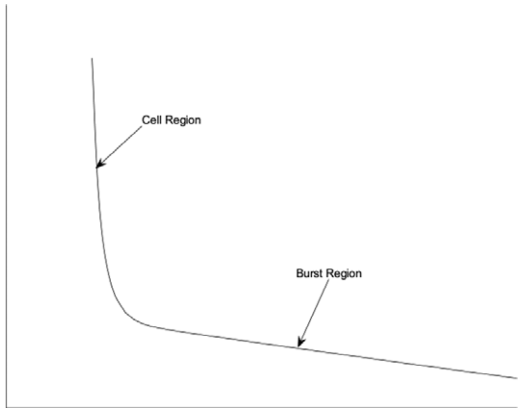

Fig. 1. Cell and burst regions. The $P_L(N)$ plotted on the y-axis is on a log scale and the buffer size plotted on the x-axis is on a linear scale.

It has been observed in [6] that if there is less correlation between packets' interarrival times and the traffic in the network is heavy (ρ is close to 1), the decay rate ξ becomes very large and hence the function $P_L(N)$ decreases quickly as N increases. This means that already for small buffer sizes, say $N = 20$, the loss is very little. Such networks are not of much interest because of their limited practical use. More correlation between the cells is introduced when the transition probabilities $\boldsymbol{p}$ and $\boldsymbol{q}$ (or p and q) are less than 10^{-3}. This case is of major importance when the input consists of more video sources [3]. Also, in most real-world network environments, the network load ρ need not be always close to 1. In such cases, the function $\log P_L(N)$ decreases quickly (cell region) as N increases until a particular buffer size $\tilde{N}$. Beyond $\tilde{N}$, it decreases slowly (burst region) as ξN asymptotically. In Section 3.2 we shall indicate how an approximation for $\tilde{N}$ can be computed and put to good use.

3. Rational approximation

3.1. Free poles

Because of the fact that the function $\log P_L(N)$ asymptotically decays as ξN for large N, polynomial approximation techniques for $\log P_L(N)$ are not suitable. Every polynomial model of degree larger than one, would blow up too quickly for large N. However, a rational function $r_n(N)$ of numerator degree $n+1$ and denominator degree n, has a similar asymptotic behavior as that of $\log P_L(N)$. It remains to compute the coefficients in numerator and denominator of the rational function

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

mostly from computed values of $\log P_L(N_j)$ for small buffer sizes N_j, to fit the behavior of $\log P_L(N)$.

The rational model is fully specified when we know its numerator and denominator coefficients $b_1, \ldots, b_n$ and $a_0, \ldots, a_{n+1}$, a total of $2n+2$ coefficients (the constant term b_0 in the denominator is only a normalization constant for the rational function [20] and is therefore assigned the value 1, whenever possible). These coefficients are determined from sampling $\log P_L(N)$ at chosen N_j for $j = 0, \ldots, 2n$ while a_{n+1} is determined from the asymptotic behavior

$$\lim_{N\to\infty} \log P_L(N) \approx \xi N = \frac{a_{n+1}}{b_n} N. \tag{12}$$

More details and an illustration of this technique can be found in [6]. Functions with poles or at most a countable number of isolated essential singularities, in particular, allow nice convergence properties when approximated by rational functions. In that case, the singularities of the function under consideration, attract the poles of the rational approximant to their position. Although the behavior of the function $\log P_L(N)$ is not such that a rational approximant $r_n(N)$ naturally attracts real positive poles, once in a while it may happen that $r_n(N)$ has one or more poles in the region of interest for N. For instance, with the system parameters given by

$$M = 10, \quad M_s = 5, \quad p = 5 \times 10^{-5}, \quad q = 6 \times 10^{-5},$$
$$d = 0.02, \quad c = 1, \quad \boldsymbol{\alpha} = [1, 0, 0, 0, 0],$$

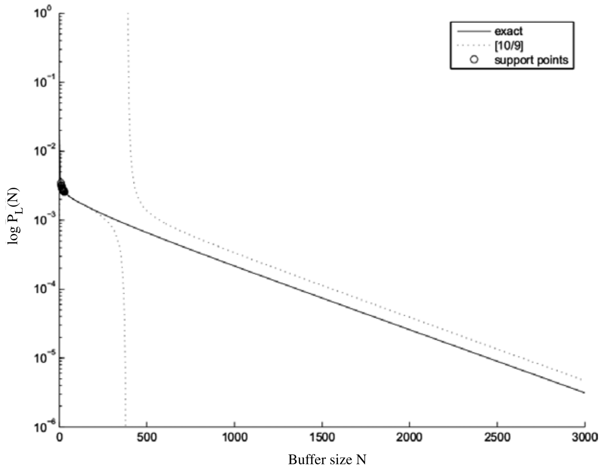

Fig. 2. Rational approximant for $\log P_L(N)$ with a pole around $N = 80$.

$$\boldsymbol{S} = \begin{bmatrix} 0.45 & 0.55 & 0 & 0 & 0 \\ 0 & 0.45 & 0.55 & 0 & 0 \\ 0 & 0 & 0.45 & 0.55 & 0 \\ 0 & 0 & 0 & 0.45 & 0.55 \\ 0 & 0 & 0 & 0 & 0.45 \end{bmatrix},$$

the approximant r_2, which can be seen in Fig. 2, exhibits a pole around $N = 400$. This is of course undesirable. Since a suitable value for the denominator degree n is determined from comparing $\|r_{n+1} - r_n\|/\|r_{n+1}\|$ on the positive real axis to a threshold value ϵ,

$$\sup_{0<N<\infty} |r_{n+1}(N) - r_n(N)| \leq \epsilon \sup_{0<N<\infty} |r_{n+1}(N)|, \tag{13}$$

the occurrence of an undesirable pole slows the method down. Fortunately, it does not make the method unsuitable. Indeed, when $\log P_L(N)$ itself does not have any singularities on the positive real axis, while $r_n(N)$ has a pole at $N = N^\star > 0$, then there is no mathematical argument for the next approximant $r_{n+1}(N)$ to have a pole in the neighborhood of $N = N^\star$ as well. So

$$\sup_{0<N<\infty} |r_n(N) - r_{n-1}(N)| = \sup_{0<N<\infty} |r_{n+1}(N) - r_n(N)| = \infty$$

and (13) is not satisfied, but (13) may be satisfied for a larger value of n. Let us now explain how the occurrence of these undesirable poles can be avoided.

3.2. Curvature maximum of $\log P_L(N)$

Besides the typical asymptotic behavior of $\log P_L(N)$, we also want to capture the neighborhood of N for which the graph turns from a steep descent (cell region) toward its linear asymptotic look (burst region). The curvature of a function $f(x)$ is given by

$$\kappa(x) = \frac{f''(x)}{\left(\sqrt{1 + (f'(x))^2}\right)^3}.$$

Here $'$ and $''$ denote the first and second derivatives, respectively. A discretized version of $\kappa(x)$, which can also be used for $f(N) = \log P_L(N)$, is given by

$$\kappa_j = \frac{(f_{j+1} - 2f_j + f_{j-1})}{\left(\sqrt{1 + (f_{j+1} - f_j)^2}\right)^3}, \quad j \geq 1. \tag{14}$$

where $f_j = \log P_L(j+c)$. We are interested in the point of maximal curvature of the function $\log P_L(N)$, which can be estimated by monitoring κ_j for successive values of j. Let us denote by $\tilde{N}$ the value of $j + 1$ for which κ_j attains its maximum. Then the computation of $\tilde{N}$ can be put to good use, becomes clear from the following observations.

A rational function of the form $1/(N^2 + R^2)$ with $R > 0$ resembles $\log P_L(N)$ on the positive real axis. It has its maximal curvature for positive N in the immediate neighborhood of the point $N = R$, namely in the interval $[R, 1.002R]$ if $R \geq 2$.

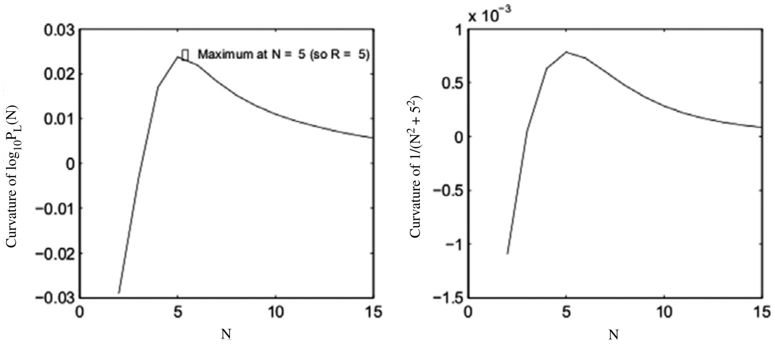

Fig. 3. Curvature of $\log P_L(N)$ and of $1/(N^2+R^2)$.

A typical evolution of the curvature of $\log P_L(N)$ and that of $1/(N^2+R^2)$ can be seen in Fig. 3. A rational function of the form

$$r_2(N) = \frac{\xi N^3 + a_2N^2 + a_1N + a_0}{N^2+R^2}$$

exhibits a limiting behavior of the required type, namely $\lim_{N\to\infty} r_2(N) \approx \xi N$. It also achieves maximal curvature in the neighborhood of $N = R$.

It has been observed in all numerical examples that the maximum of the curvature of $\log P_L(N)$, occurs for $N < M$ where M is the number of sources. The computation of $\tilde{N}$ is detailed in Section 4.1.

3.3. Optimally placed poles

So, when detecting an approximation N_j of the point of maximal curvature of $\log P_L(N)$, through the computation of (14), we can choose $R = N_j$ in $r_2(N)$ and introduce two complex conjugate poles, thereby preventing the occurrence of real poles in $r_2(N)$. It remains to point out which strategy can be followed for larger denominator degrees n.

A rational function $r_n(N)$ with denominator polynomial of the form

$$q_n(N) = \begin{cases} (N^2+R^2)\prod_{j=1}^{k-1}\left(N+Re^{\imath\theta_j}\right)\left(N+Re^{-\imath\theta_j}\right), & n=2k \\ (N+R)(N^2+R^2)\prod_{j=1}^{k-1}\left(N+Re^{\imath\theta_j}\right)\left(N+Re^{-\imath\theta_j}\right), & n=2k+1 \end{cases} \tag{15}$$

has its point of maximal curvature in the neighborhood of $N = R$ only if $\theta_j \approx \pi/2$. So, when increasing the denominator degree of $r_n(N)$, more complex conjugate poles of modulus R can be prescribed, by choosing different $\theta_j \approx \pi/2$. Complex conjugate poles further away from the imaginary axis pull the point of maximal curvature away from $N = R$.

So far we have explained how to use the curvature of $\log P_L(N)$ to fix the coefficients $b_0, \ldots, b_n$ in the rational model, and the slope ξ for the coefficient a_{n+1} determined by (12). Note that the normalization $b_0 = 1$ has been replaced by $b_n = 1$ in (15), an equally simple choice. The remaining coefficients, being the numerator coefficients $a_0, \ldots, a_n$, can be computed from the polynomial data fitting conditions

$$(q_n \log P_L)(N_j) = p_n(N_j) \quad j = 0, \ldots, n \tag{16}$$

which, under the condition that $q_n(N_j) \neq 0$, are equivalent to

$$r_n(N_j) = \log P_L(N_j) \quad j = 0, \ldots, n.$$

The values to be chosen for N_j are further detailed in Section 4.2. This technique is called Newton–Padé-type or multipoint Padé-type approximation [21]. It differs from the standard multipoint Padé approximation because the denominator $q_n(N)$ is not determined by the interpolation conditions but is prechosen.

The convergence results obtained in [21] underline that:

1. the rational approximants $r_n(N)$ need to be uniformly bounded on bounded subsets of the region of interest (here the natural numbers);
2. the interpolation points N_j cannot be scattered around but must be centered in one location (as we describe in Section 4.2).

4. Numerical illustration

4.1. Computation of R for the denominator polynomial

In all numerical experiments we observed that the maximum of the curvature of $\log P_L(N)$ is attained in the interval $[c, M+c-1]$, where M is the number of sources and c is the maximum number of packets served by the server as a group. That is, either the curvature increases from $N = c$ on until it reaches its global maximum and then decreases, or, for some networks, the curvature function is a decreasing function in the interval $[c, M+c-1]$ and then we return $N = 2$ as the argument of the maximum. Since M is relatively speaking rather small, the computation of $R = \tilde{N}$ is almost negligible. The detailed algorithm for the estimation of R as explained in the Sections 3.2 and 3.3, goes as follows:

1. compute $\log P_L(i)$ for $i = c, c, c+1, c+2, c+3$,
2. compute κ_j for $j = 1, 2$,
3. set $i = c+4$ and $j = 3$,
4. while $((\kappa_j - \kappa_{j-1} > 0)$ or $(i < M+c-1))$
   (a) compute $\log P_L(i)$ and κ_j,
   (b) set $j = j+1$ and $i = i+1$,
5. if $(i < M+c-1)$ then $R = i - c + 1$,
6. else $R = 2$.

4.2. Choosing the support points for the numerator polynomial

We start from the approximants r_1 and r_2 whose support points are chosen as follows. We use supp as an abbreviation for the set of support points:

if $(p < 10^{-3})$ and $(q < 10^{-3})$ then

$$\begin{aligned}\texttt{supp} &= \{c, 200, 400\} \text{ for } r_1 \text{ and}\\ \texttt{supp} &= \{c, 40, 200, 400\} \text{ for } r_2,\end{aligned}$$

else

$$\begin{aligned}\texttt{supp} &= \{c, c+1, c+2\} \text{ for } r_1 \text{ and}\\ \texttt{supp} &= \{c, c+1, c+2, c+3\} \text{ for } r_2.\end{aligned}$$

Each subsequent support point is chosen from the set $\{c, \dots, M+c-1\}$ by placing it where

$$\frac{|r_{n+1}(N) - r_n(N)|}{|r_n(N)|}, \quad N = c, \dots, M+c-1$$

attains its maximum value. This process of constructing subsequent approximants continues until

$$\frac{\|r_{n+1} - r_n\|}{\|r_{n+1}\|} \le \epsilon.$$

4.3. Output and figures

To compare the model $r_n(N)$ to $\log P_L(N)$, the latter is computed using the algorithm from [4]. All the numerical experiments have also been verified at one point using standard Monte Carlo simulation (20 simultaneous runs). The stopping criterion for the simulation guaranteed a maximum relative error of 5%. The relative error in the simulation was computed from the associated confidence interval, which was obtained through the usual normal distribution approximation.

Computations are carried out in the CalcUA supercomputer at the University of Antwerp, Belgium [22]. In all figures, the values of $\log P_L(N)$ at support points are circled, the exact 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 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. We consider the following six examples for the numerical illustrations. In all examples, the values of θ_j in (15) are given by

$$\theta_{2i-1} = \frac{\pi}{2} + (i+1)\mathcal{T}, \qquad \theta_{2i} = \frac{\pi}{2} - (i+1)\mathcal{T}, \quad i \ge 1,$$

where the constant $\mathcal{T} \le \pi/12$.

For the rational approximant, the computation time includes the computation time for the support points.

We have chosen the following six examples in order to capture different curvatures of the $\log P_L(N)$ graphs between cell and burst regions (see Fig. 1). As discussed in Section 3.2, capturing approximately the curvature of the graph of $\log P_L(N)$ using the proposed algorithm in Section 4.1 is the crucial step in order to place the poles optimally for the Newton–Pádé-type approximation. Through these examples we show the efficiency of the algorithm to find the curvature and the proposed rational approximation technique. In addition, Examples 5 and 6 are chosen to show that even for moderate values of M, M_s, c and N, the computational time (in seconds) of the PLP using (9) is in the order of 10^4 (see Figs. 8(b) and 9(b)).

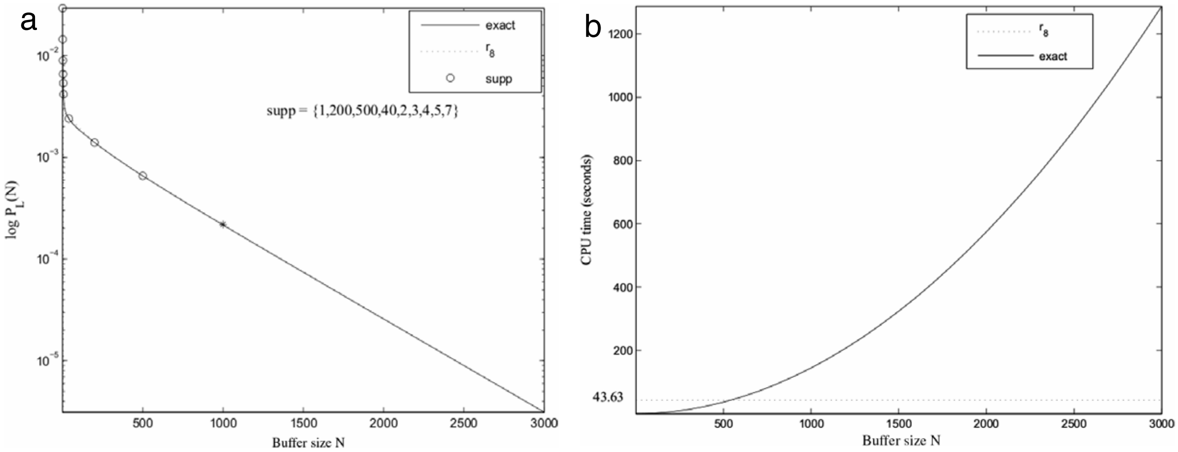

Fig. 4. Packet loss probabilities of the model from Example 1.

Example 1. We consider 10 sources and a service channel with 5 phases where the server is capable of handling at most 1 packet and each arriving packet joins the first phase of the service channel, upon the availability of the server, and goes through all the phases before it leaves the system. The parameter values of this model are:

$$M = 10, \quad M_s = 5, \quad p = 5 \times 10^{-5}, \quad q = 6 \times 10^{-5}, \quad d = 0.02, \quad c = 1, \quad \boldsymbol{\alpha} = [1, 0, 0, 0, 0],$$

$$\boldsymbol{S} = \begin{bmatrix} 0.45 & 0.55 & 0 & 0 & 0 \\ 0 & 0.45 & 0.55 & 0 & 0 \\ 0 & 0 & 0.45 & 0.55 & 0 \\ 0 & 0 & 0 & 0.45 & 0.55 \\ 0 & 0 & 0 & 0 & 0.45 \end{bmatrix}.$$

The mean arrival and service rates are $\lambda = 0.0909$ and $\mu = 0.11$, the load $\rho = 0.8264$, the decay rate $\xi = -9.1415 \times 10^{-4}$ and $R = 2$. In Fig. 4(a), we have used $r_8(N)$ to approximate $\log P_L(N)$. The time taken to compute the PLP using r_8 for $1 \le N \le 3000$ is just 43.64 s as shown in Fig. 4(b).

Example 2. We consider 15 sources and a service channel with 15 phases where the server is capable of handling at most 1 packet. Whenever the service channel is available each arriving packet joins any one of the 15 phases randomly according to the stochastic random vector $\boldsymbol{\alpha}$ and continues to stay in the service channel according to the sub-stochastic random matrix $\boldsymbol{S}$. The parameter values of this model are:

$$M = 15, \quad M_s = 15, \quad p = 2.19 \times 10^{-5}, \quad q = 7 \times 10^{-5}, \quad d = 0.07, \quad c = 1.$$

The mean arrival rate is $\lambda = 0.2502$. The $\boldsymbol{\alpha}$ and $\boldsymbol{S}$ are chosen randomly with the condition that the service rate $\mu = 0.4864$ and the load $\rho = 0.5144$. The decay rate $\xi = -5.1477 \times 10^{-4}$ and $R = 2$. In Fig. 5(a), we have used $r_{10}(N)$ to approximate $\log P_L(N)$. The time taken to compute the PLP using r_{10} for $1 \le N \le 3000$ is 1900.22 s as shown in Fig. 5(b).

Example 3. We consider 10 sources and a service channel with 5 phases where the server is capable of handling at most 5 packets in group. Arriving packets join the first phase of the service channel, upon the availability of the server, and go through all the phases before they leave the system. The parameter values of this model are:

$$M = 10, \quad M_s = 5, \quad p = 9 \times 10^{-3}, \quad q = 6 \times 10^{-3}, \quad d = 0.162, \quad c = 5, \quad \boldsymbol{\alpha} = [1, 0, 0, 0, 0],$$

$$\boldsymbol{S} = \begin{bmatrix} 0.0005 & 0.9995 & 0 & 0 & 0 \\ 0 & 0.0005 & 0.9995 & 0 & 0 \\ 0 & 0 & 0.0005 & 0.9995 & 0 \\ 0 & 0 & 0 & 0.0005 & 0.9995 \\ 0 & 0 & 0 & 0 & 0.9995 \end{bmatrix}.$$

The mean arrival and service rates are $\lambda = 0.9720$ and $\mu = 0.9995$, the load $\rho = 0.9725$, the decay rate $\xi = -0.0026$ and $R = 2$. In Fig. 6(a), we have used $r_3(N)$ to approximate $\log P_L(N)$. The time taken to compute the PLP using r_3 for $1 \le N \le 3000$ is just 3.47 s as shown in Fig. 6(b).

Example 4. We consider 15 sources and a service channel with 5 phases where the server is capable of handling at most 10 packets. Whenever the service channel is available arriving packets join any one of the 15 phases randomly according

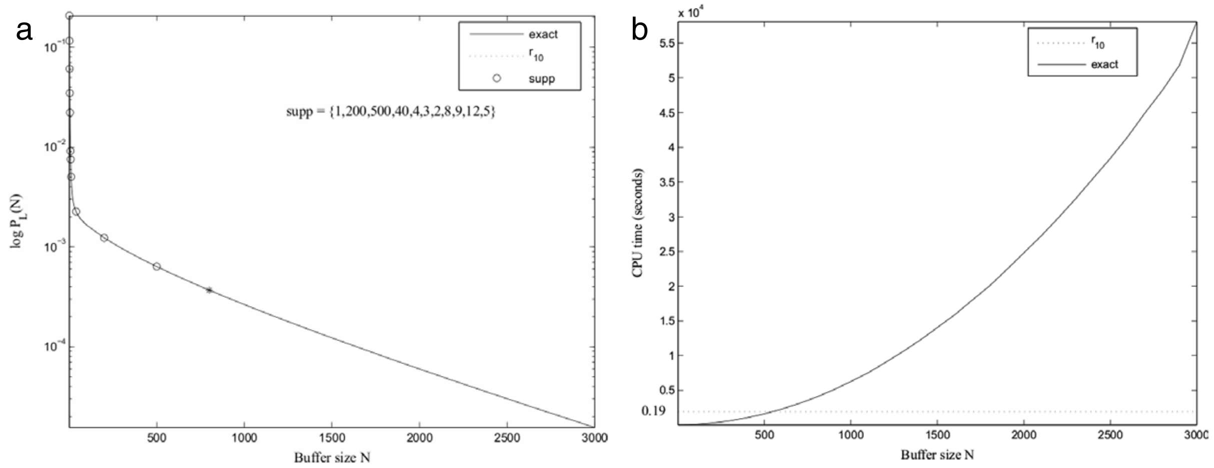

Fig. 5. Packet loss probabilities of the model from Example 2.

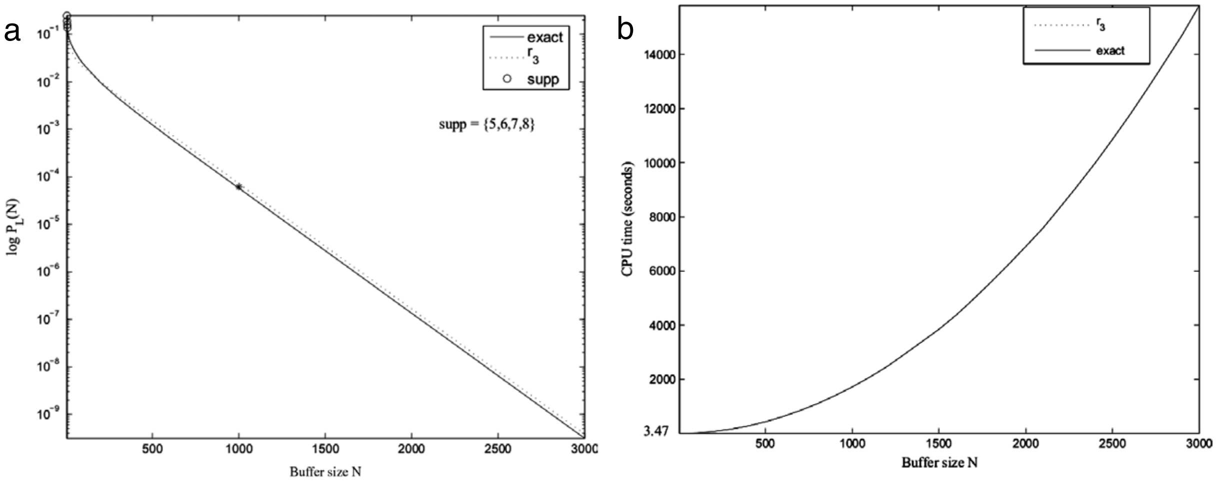

Fig. 6. Packet loss probabilities of the model from Example 3.

to the stochastic random vector $\boldsymbol{\alpha}$ and continue to stay in the service channel according to the sub-stochastic random matrix $\boldsymbol{S}$. The parameter values of this model are:

$$M = 15, \qquad M_s = 5, \qquad p = 3.4 \times 10^{-5}, \qquad q = 5 \times 10^{-5}, \qquad d = 0.6, \qquad c = 10.$$

The mean arrival rate is $\lambda = 3.6429$. The $\boldsymbol{\alpha}$ and $\boldsymbol{S}$ are chosen randomly with the condition that the service rate $\mu = 5.7240$ and the load $\rho = 0.6364$. The decay rate $\xi = -6.0758 \times 10^{-5}$ and $R = 6$. In Fig. 7(a), we have used $r_3(N)$ to approximate $\log P_L(N)$. The time taken to compute the PLP using r_3 for $c \le N \le 3000$ is 2192.33 s as shown in Fig. 7(b).

Example 5. We consider 25 sources and a service channel with 5 phases where the server is capable of handling at most 10 packets in group. Arriving packets join any one of the 5 phases randomly according to the stochastic random vector $\boldsymbol{\alpha}$ and continue to stay in the service channel according to the sub-stochastic random matrix $\boldsymbol{S}$. The parameter values of this model are:

$$M = 25, \qquad M_s = 5, \qquad p = 5 \times 10^{-6}, \qquad q = 6 \times 10^{-6}, \qquad d = 0.2, \qquad c = 5.$$

The mean arrival rate is $\lambda = 2.2727$. The $\boldsymbol{\alpha}$ and $\boldsymbol{S}$ are chosen randomly with the condition that the service rate $\mu = 4.1895$ and the load $\rho = 0.5425$. The decay rate $\xi = -6.6714 \times 10^{-5}$ and $R = 3$. In Fig. 8(a), we have used $r_7(N)$ to approximate $\log P_L(N)$. The time taken to compute the PLP using r_7 for $c \le N \le 3000$ is 9019.4 s as shown in Fig. 8(b).

Example 6. We consider 15 sources and a service channel with 10 phases where the server is capable of handling at most 10 packets. Whenever the service channel is available arriving packets join any one of the 10 phases randomly according to

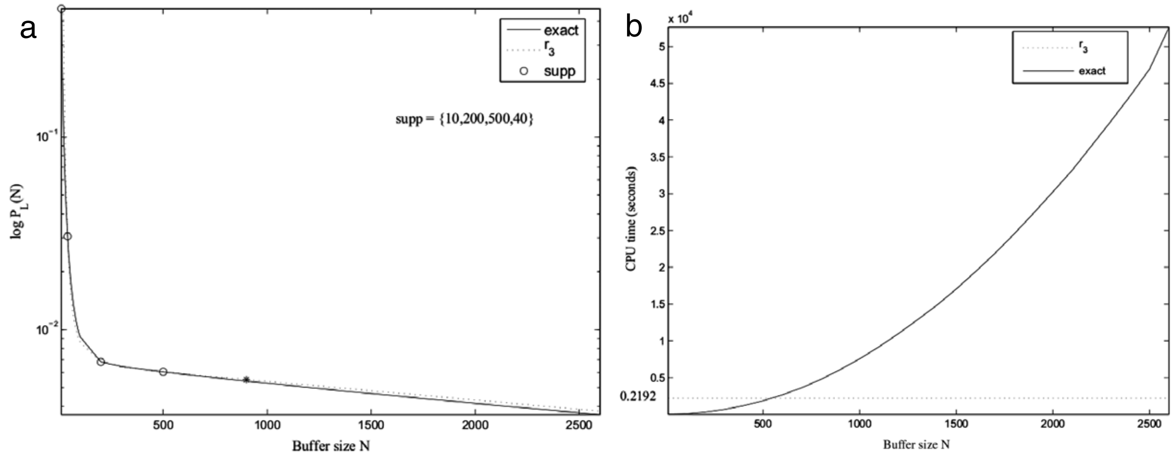

Fig. 7. Packet loss probabilities of the model from Example 4.

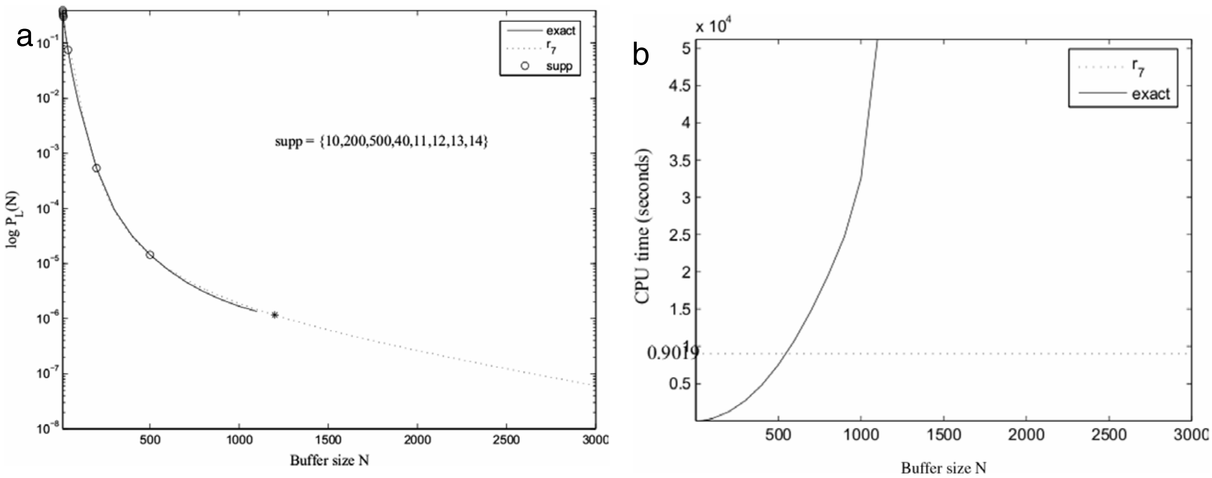

Fig. 8. Packet loss probabilities of the model from Example 5.

the stochastic random vector $\boldsymbol{\alpha}$ and continue to stay in the service channel according to the sub-stochastic random matrix $\boldsymbol{S}$. The parameter values of this model are:

$$M = 15, \qquad M_s = 10, \qquad p = 3.4 \times 10^{-5}, \qquad q = 5 \times 10^{-5}, \qquad d = 0.6, \qquad c = 10.$$

The mean arrival rate is $\lambda = 3.6429$. The $\boldsymbol{\alpha}$ and $\boldsymbol{S}$ are chosen randomly with the condition that the service rate $\mu = 6.0140$ and the load $\rho = 0.6057$. The decay rate $\xi = -7.2007 \times 10^{-5}$ and $R = 6$. In Fig. 9(a), we have used $r_3(N)$ to approximate $\log P_L(N)$. The time taken to compute the PLP using r_3 for $c \le N \le 3000$ is 15250.96 s as shown in Fig. 9(b).

Acknowledgements

This work is based in part, upon research supported by the National Science Foundation (grant nos. CNS-0619069, EPS-0701890 and OISE 0729792), National Board of Higher Mathematics, India (grant no. 48/5/2004/R&D-II/2120), NASA EPSCoR Arkansas Space Grant Consortium (grant no. UALR 16804), and Acxiom Corporation (contract no. 281539). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies. The authors would like to thank the CalcUA supercomputer project of the University of Antwerp, Belgium, for the possibility to carry out the intensive computations required for the illustration of the techniques in Figs. 4(a)–9(b). The authors would like to thank the anonymous referees for their useful suggestions and detailed comments on the draft of this paper which helped in improving the clarity of the paper.

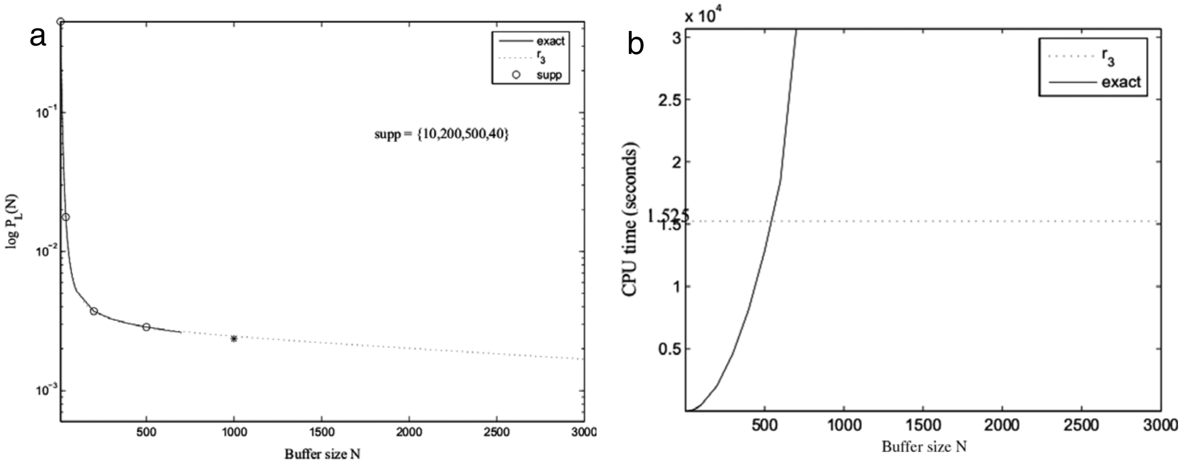

Fig. 9. Packet loss probabilities of the model from Example 6.

References

[1] K. Yoshigoe, Trends in highly scalable crossbar-based packet switch architecture, Elsevier Computer Communications 32 (4) (2009) 740–749.
[2] M. Neuts, Structured Stochastic Matrices of M/G/1 Type and their Applications, Marcel Dekker, 1989.
[3] C. Blondia, O. Casals, Performance analysis of statistical multiplexing of VBR sources: A matrix-analytical approach, Performance Evaluation 16 (1992) 5–20.
[4] A. Kamal, Efficient solution of multiple server queues with application to the modeling of ATM concentrators, in: Proc. IEEE Infocom '96, San Francisco, CA, 1996, pp. 248–254.
[5] K. Wuyts, R. Boel, Efficient matrix geometric methods for B-ISDN by using the spectral decomposition of the rate matrices, in: S. Chakravarthy, A. Alfa (Eds.), Proceedings of the 2nd Conference on Matrix-Analytic Methods in Stochastic Models, Notable Publications, 1998, pp. 341–359.
[6] A. Cuyt, R. Lenin, G. Willems, C. Blondia, P. Rousseeuw, Computing packet loss probabilities in multiplexer models using rational approximation, IEEE Transactions on Computers 52 (5) (2003) 633–644.
[7] N. Akar, E. Arikan, Markov modulated periodic arrival process offered to an multiplexer, in: IEEE GLOBECOM'93, Houston, TX, 1993, pp. 783–787.
[8] A. Elwalid, D. Mitra, Analysis, approximations and admission control of a multi-service multiplexing system with priorities, in: Proc. INFOCOM'95, 1995, pp. 463–472.
[9] K. Sohraby, On the asymptotic analysis of statistical multiplexers with hyper-bursty arrivals, Tech. rep., IBM Watson Res. Center, Yorktown Heights, NY, 1993.
[10] N.B. Shroff, M. Schwartz, Improved loss calculations at an ATM multiplexer, IEEE/ACM Transactions on Networking 6 (4) (1998) 411–421.
[11] A. Elwalid, Markov modulated rate processes for modelling, analysis and control of communication networks, Ph.D. thesis, Grad. Sch. Arts Sci. Columbia Univ., New York, 1991.
[12] D. Anick, D. Mitra, M. Sondhi, Stochastic theory of a data-handling system with multiple sources, Bell Syst. Tech. J. 61 (1982) 1871–1984.
[13] C. Chang, P. Heidelberger, S. Juneja, P. Shahabuddin, Effective bandwidth and fast simulation of ATM intree networks, in: Proceedings of the 16th IFIP Working Group 7.3 International Symposium on Computer Performance Modeling Measurement and Evaluation, Elsevier Science Publishers B. V, Amsterdam, The Netherlands, Rome, Italy, 1994, pp. 45–65.
[14] P. Heidelberger, R. Simha, Fast simulation of a voice-data multiplexer, in: IEEE INFOCOM'95, Boston, 1995, pp. 361–368.
[15] W. Gong, H. Yang, H. Hu, On the convergence of global rational approximants for stochastic discrete event systems, Discrete Event Dynamic Systems - Theory and Applications 7 (1) (1997) 93–116.
[16] H. Yang, Global rational approximation for computer systems and communication networks, Ph.D. thesis, University of Massachusetts, Amherst, MA 01003 (1996).
[17] M. Neuts, Matrix-Geometric Solutions in Stochastic Models, An Algorithm Approach, John Hopkins University Press, 1981.
[18] E. Falkenberg, On the asymptotic behavior of the stationary distribution of Markov chains of M/G/1-type, Commun. Statist. Stochastic Models 10 (1994) 75–97.
[19] P. Glynn, W. Whitt, Logarithmic asymptotics for steady-state tail probabilities in a single-server queue, Adv. Appl. Prob. 26 (1994) 131–156.
[20] D. Warner, Hermite interpolation with rational functions, Ph.D. thesis, University of California, 1974.
[21] J. Van Iseghem, Multipoint Padé-type approximants: Convergence, continuity, in: C. Brezinski, U. Kulisch (Eds.), Computational and Applied Mathematics. I. Algorithms and Theory (Dublin, 1991), North-Holland, Amsterdam, 1992, pp. 237–242.
[22] CalcUA, Calcua supercomputer. http://www.calcua.ua.ac.be/, 2008.

R.B. Lenin received the Ph.D. degree in Mathematics from the Indian Institute of Technology, Madras, India, in 1998. He is currently working as an assistant professor in the department of mathematics at the University of Central Arkansas (UCA), USA. Prior to joining UCA, he was working as a post-doctoral fellow at the University of Arkansas at Little Rock (UALR), USA. Earlier, he had also worked as a post-doctoral fellow at the University of Twente, The Netherlands, and at the University of Antwerp, Belgium for about 5 years. His research interests are in mathematical modeling and simulation of discrete-event systems, stochastic models, optimization techniques, performance analysis of computer and communication networks, rational approximation, information retrieval and network analysis. He has 27 papers published in international journals including Journal of Applied Probability, Queueing Systems, IEEE Transactions on Computers and IEEE Transactions on Microwave Theory and Techniques.

A. Cuyt is full professor at the Department of Mathematics and Computer Science of the University of Antwerp. She received her Doctor Scientiae degree in 1982 from the same university, summa cum laude and with the felicitations of the jury. Subsequently she was a Research fellow with the Alexander von Humboldt Foundation (Bonn, Germany) and she was honoured with a Masuda Research Grant (Kobe, Japan). She is the author of more than 150 publications in international journals and conference proceedings, the author or editor of several books and the organizer of a number of international events. Her current interests are in rational approximation theory and its applications in scientific computing. In view of these interests she is an editorial board member of the journal "Reliable Computing", the "International Journal of Computing Science and Mathematics" and the "International journal of Numerical analysis, Industrial and Applied mathematics". From 1997 to 2006 she also served as a board member on the computer science committee of the Flemish Science Foundation.

K. Yoshigoe (kxyoshigoe@ualr.edu) received a Ph.D. in Computer Science and Engineering from the University of South Florida in 2004. He is currently an assistant professor in the Department of Computer Science at UALR. His research focuses on design and evaluation of systems ranging from high-speed packet switches to wireless sensor networks. He is a member of IEEE and ACM.

S. Ramaswamy is currently Professor and Chairperson of the Computer Science Department at the University of Arkansas at Little Rock. His research interests on behavior modeling, analysis and simulation, software stability and scalability – particularly in the design and development of better software systems, and intelligent and flexible control systems. At UALR, he is currently associated with several research initiatives, which include: the statewide program manager for wireless nanosensors and systems center and the principle investigator at UALR for a high performance computing initiative. He is also the research coordinator for collaboration on "Engineering Innovative Software Systems for Marine Transportation Logistics" with the National Institute of Applied Sciences (INSA) in Rouen, France, where he was a visiting research professor in 2006 and 2007. During the summers of 2003, 2004 and 2007, he was a visiting research professor of Computer Science in the Institute of Software Integrated Systems (ISIS) at Vanderbilt University as part of an NSF ITR project – Foundations of Hybrid and Embedded Software Systems. In 1994–1995, and subsequently during the summer months of 1996 and 1998, he was a post-doctoral research fellow/visiting scientist in the Laboratory for Intelligent Processes and Systems (LIPS) at the University of Texas at Austin where he helped with research efforts on Sensible Agents. Dr. Ramaswamy earned his Ph.D. degree in Computer Science in 1994 from the Center for Advanced Computer Studies (CACS) at the University of Louisiana at Lafayette. He serves as an Associate Editor for the IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews. He is a senior member of the ACM, a member of Society for Computer Simulation International, Computing Professionals for Social Responsibility and a senior member of the IEEE.