22

Efficient Simulation of Consecutive Cell Loss in ATM Networks

V.F. Nicola and G.A. Hagesteijn
Tele-Informatics and Open Systems
University of Twente
P.O. Box 217, 7500 AE, Enschede, The Netherlands.
Telephone: 053-4894286. Fax: 053-4893247. e-mail: vfn@cs.utwente.nl

Abstract

In some B-ISDN applications running on ATM networks (e.g., for audio/video connections), the occasional loss of a single ATM cell may not affect the user's perceived QoS requirement. However, the QoS may be degraded due to the loss of a multiple (consecutive) ATM cells. As the event of consecutive cell loss is (typically) rare, its probability cannot be estimated efficiently using standard simulation. In this paper we propose a fast simulation method, based on importance sampling, to efficiently estimate the probability of a rare consecutive-cell-loss event. As an example, we consider a queueing model of the Leaky Bucket source policing algorithm, operating in a bursty traffic environment. We present empirical results to demonstrate the validity and effectiveness of our fast simulation method.

Keywords

Rare event simulation, Importance sampling, Cell loss, ATM networks, Quality of service

D. D. Kouvatsos (ed.), *ATM Networks*

1 INTRODUCTION

In an Asynchronous Transfer Mode (ATM) network, data is transported in fixed-size cells. A cell loss may occur due to a variety of reasons, such as buffer overflow in one or more of the network nodes, or as a result of traffic policing at the interface between the user and the network. In any case, the impact of a cell loss on the quality of service (QoS) provided by a given connection depends on the application and its resilience with respect to such a cell loss.

Due to the bursty nature of traffic generated by broadband applications (e.g., multimedia and video conferencing), cells are likely to be lost in multiples (i.e., losing more than one consecutive arriving cells). For example, a buffer overflow at a network node (even if rare) may result in the loss of many consecutive cells. Recovery techniques, such as cell retransmission, may be implemented at the communication protocol level or at the application level. In some applications (such as packet audio/video communication), the occasional loss of one or a few cells may not influence the QoS. Also, extrapolation and/or error correcting techniques can be used to compensate for such cell loss. However, in the absence of cell retransmission or other adequate recovery procedures, the loss of consecutive ATM cells may lead to a remarkable or intolerable degradation of QoS. Therefore, for most applications, it is important to keep the occurrence of consecutive cell loss as rare as possible. This is particularly true for applications with bursty traffic, for which the frequency of consecutive cell loss tend to be (relatively) high. The number of consecutive cell loss that can be tolerated without affecting the QoS depends on the application and/or the supporting recovery (or error correcting) mechanism, if any. For a given application, it is desirable to keep the frequency of losing more than a certain (tolerable) number of consecutive cells below some acceptable threshold. This frequency may be defined as the reciprocal of the steady-state average number of cells between such consecutive-cell-loss events. In a simple queueing model with a finite buffer, this frequency is closely related to another measure of interest; namely, the probability of consecutive cell loss, say, in a busy cycle.

Needless to say, the development of models for the analysis of consecutive cell loss is of much interest for the proper dimensioning of various buffers and other network control parameters. To the best of our knowledge, so far, there has been no analytical results relating to this relevant problem. For a simple $M/M/1$ queue with a finite buffer, we derive analytic closed form expressions for the frequency of consecutive cell loss and the probability of its occurrence in a busy cycle (see Section 2.2 of this paper.) However, for a $GI/GI/1$ queue, the analysis is considerably more difficult, and a useful analytical or algorithmic solution, if at all possible, is not yet available. For the typically correlated and bursty arrival processes, the feasibility of a useful analysis seems even more remote. Furthermore, the probabilities of interest are typically very small, leading to numerical problems.

In order to avoid restrictions necessary for analytic tractability and/or numerical feasibility, simulation is often preferred for the evaluation of realistic models. However, accurate estimation of the frequency of rare events, such as consecutive cell loss, requires observing numerous such events. But, if the frequency of consecutive cell loss is 10^{-9} per cell, then each consecutive-cell-loss event takes place approximately once in 10^9 cells. Observing a sufficiently large number of consecutive-cell-loss events will take extremely long simulation time.

Importance sampling (Hammersley and Handscomb 1964) has been used effectively to achieve significant speed ups in simulations involving rare events, such as failure in a reliable computer system or cell loss in an ATM communication network. See Nicola et al. (1993) for a review of techniques for fast simulation of highly dependable systems, and Heidelberger (1993) for a survey of efficient simulation methods to estimate buffer overflow probabilities in communication systems. The basic idea of importance sampling is to simulate the system under a different probability measure (i.e., with different underlying probability

distributions), so as to increase the probability of typical sample paths involving the rare event of interest. For each sample path (observation) during the simulation, the measure being estimated is multiplied by a correction factor, called the *likelihood ratio*, to obtain an unbiased estimate of the measure in the original system. Asymptotically optimal change of measures (to use in importance sampling) have been found to estimate small probabilities of buffer overflow in relatively simple queueing models (see, Parekh and Walrand (1989), Sadowsky (1991), Chang et al. (1993) and others.) In this paper, we develop heuristics, which are partly based on these optimal change of measures, to estimate very small consecutive-cell-loss probabilities in simple $GI/GI/1/k$ queues (k is the buffer capacity, including the server). We use our heuristics to evaluate a queueing model of the Leaky Bucket (LB) algorithm (see Rathgeb (1991)). Two cell arrival processes are considered; namely, a Poisson process (mainly for validation and experimentation) and a bursty two-phase burst/silence process (see Section 4.4). Empirical results demonstrate the effectiveness of our method to estimate very small consecutive-cell-loss probabilities. These results also show that the simulation time needed to achieve a given accuracy increases (however, slightly) with the number of consecutive cell loss. This increase is attributed to the inherent increase in variability of the probability being estimated, rather than the rarity of the event.

The rest of this paper is organized as follows. In Section 2, we introduce some notation relevant to the study of consecutive cell loss in simple queues, and we carry out the analysis for the $M/M/1/k$ queue. In Section 2.3, we briefly introduce the problem of rare event simulation and review the basic idea of importance sampling. Change of measures used in importance sampling to speed up simulations of simple queues are presented in Section 3; both, a rare full-buffer event and a rare consecutive-cell-loss event, are considered. Validation and experiments with our heuristic change of measure to simulate a queueing model of the LB algorithm are presented in Section 4. Conclusions are given in Section 5.

2 CONSECUTIVE CELL LOSS IN SIMPLE QUEUES

In this section we give brief preliminaries and notation that are needed for the discussion of consecutive cell loss in simple queues. For an $M/M/1/k$ queue, i.e., Poisson cell arrivals and exponential service time distribution, the analysis is not complicated and it is carried out in this section. The results of this analysis are used in Section 4 to validate statistical output obtained from simulation. For general inter-arrival and/or service time distributions, the analysis is considerably more difficult and is not considered here.

2.1 Preliminaries

Consider an $GI/GI/1/k$ queue (k is the buffer capacity, including the server). The probability density function (pdf) of the inter-arrival (resp., service) time is given by $f_A(t)$ (resp., $f_S(t)$.) Define the n-consecutive-cell-loss event to be the (cell arrival) event at which exactly n consecutive cells are lost during a single full-buffer (or overflow) period. (Note that more than n cells may be lost during the same overflow period.) We are interested in the steady-state frequency of this event, i.e., the reciprocal of the average number of arriving cells between two subsequent n-consecutive-cell-loss events; this is denoted by $\mathcal{F}_n$. A closely related measure of interest is the probability of n or more consecutive cell losses in a busy cycle; this is denoted by γ_n.

Let $N(t)$ be the number of items (cells) in the queue (including that in service) at time t, and denote by $t_j, j = 0, 1, 2, ...$, the consecutive instants in time at which $N(t)$ jumps from 0 to 1, i.e., for all $j = 0, 1, 2, ...$,

$N(t_j^-) = 0$ and $N(t_j^+) > 0$. Define a *busy cycle* to be the evolution of the process $N(t)$ between two such consecutive instants, say, t_j and t_{j+1}. Note that $t_j, j = 0, 1, 2, ...,$ constitute renewal points, and, therefore, busy cycles are i.i.d. (independent and identically distributed.) The length of a busy cycle is a r.v. T; for the j-th busy cycle $T_j = t_j - t_{j-1}, j = 1, 2, ...$ The number of arrivals during a busy cycle is a r.v. N which, because of buffer overflow, is not necessarily equal to the number of departures in the same busy cycle; for the j-th busy cycle it is denoted by N_j. Furthermore, denote by $O_{n,j}$ the number of full-buffer periods in the j-th cycle during which n or more cells are lost. $O_{n,j}$ is a realization of the random number O_n. It follows that the reciprocal of the long-run (steady-state) average number of arriving cells between two n-consecutive-cell-loss events, i.e., the frequency $\mathcal{F}_n$, is given by

$$\mathcal{F}_n = \frac{E(O_n)}{E(N)}. \tag{1}$$

Usually, analytic (or numerical) solution for $E(N)$ can be determined. In particular, for an $M/G/1/k$ queue, it is simply given by $1/p_I$, where p_I is the steady-state probability that the server is idle (see, for example, Cooper (1981)). The analysis for $E(O_n)$ is considerably more complicated, mainly because the length of a full-buffer period depends on the sample path (within a busy cycle) leading to that full-buffer. For example, in an $M/G/1/k$ queue, full-buffer periods in the same busy cycle are independent, but the first full-buffer period has a different distribution from that of the second and all subsequent full-buffer periods. However, in an $M/M/1/k$ queue, all full-buffer periods are independent and have the same exponential (service time) distribution, regardless of the sample path leading to the full-buffer. This independence yields significant simplifications leading to the analytical results obtained in the following section.

2.2 Analysis of the $M/M/1/k$ Queue

Consider an $M/M/1/k$ queue with an arrival rate λ and a service rate μ. A busy cycle is defined as above. Define $\pi_i, 0 \leq i \leq k$ as the probability that the number in the system, $N(t)$, moves from level i to level k without hitting level 0. In other words, given that $N(t) = i$, π_i is the probability that the full-buffer state will be reached before the end of the busy cycle. Let γ be the probability of at least one full-buffer period in a busy cycle. Furthermore, given a full-buffer, let ϕ be the probability of yet another full-buffer period in the same busy cycle. It follows that $\gamma = \pi_1$ and $\phi = \pi_{k-1}$. The probabilities $\pi_i, 0 \leq i \leq k$ can be determined from the following equations

$$\pi_i = \frac{\mu}{\lambda + \mu}\pi_{i-1} + \frac{\lambda}{\lambda + \mu}\pi_{i+1}, \quad 1 \leq i \leq k-1, \tag{2}$$

with $\pi_0 = 0$ and $\pi_k = 1$. It follows that

$$\pi_i = \frac{(\frac{\mu}{\lambda})^i - 1}{(\frac{\mu}{\lambda})^k - 1}, \quad 1 \leq i \leq k-1. \tag{3}$$

Now, let p_n be the probability of n or more arrivals (i.e., n or more consecutive losses) in a single full-buffer period. Since full-buffer periods are independent and having the same exponential distribution with a mean $1/\mu$, it follows that

$$p_n = (\frac{\lambda}{\lambda+\mu})^n, \quad 0 \leq n. \tag{4}$$

$P(O_n \geq i)$ is the probability, in a busy cycle, of i or more full-buffer periods, during each of which there are n or more (lost) arrivals. The probability of (at least one) n-consecutive-cell-loss in a busy cycle, γ_n, is given by

$$\begin{aligned} \gamma_n &= P(O_n \geq 1) = \sum_{k=1}^{\infty} \gamma\, \phi^{k-1}\, (1-p_n)^{k-1}\, p_n \\ &= \frac{\gamma\, p_n}{1-\phi(1-p_n)}. \end{aligned} \tag{5}$$

Also, define ϕ_n to be the probability of another n-consecutive-cell-loss in the same busy cycle. Then

$$\begin{aligned} \phi_n &= \sum_{k=1}^{\infty} \phi^k\, (1-p_n)^{k-1}\, p_n \\ &= \frac{\phi\, p_n}{1-\phi(1-p_n)}. \end{aligned} \tag{6}$$

It follows that

$$P(O_n \geq i) = \gamma_n\, \phi_n^{i-1}, \quad i \geq 1, \tag{7}$$

and

$$E(O_n) = \frac{\gamma_n}{1-\phi_n} = \frac{\gamma\, p_n}{1-\phi}. \tag{8}$$

Note that for a sufficiently high number of consecutive losses $p_n \ll 1$ and $E(O_n) \approx \gamma_n$.

The above analysis is not valid for other queues, such as $M/G/1/k$ and $GI/M/1/k$. Appropriate analysis techniques may be developed for these queues, which is a subject for further investigation and is not considered in this paper. For these and other $GI/GI/1/k$ queues, we use simulation to estimate $E(O_n)$ and/or γ_n. However, because the n-consecutive-cell-loss is typically a rare event, $E(O_n)$ and γ_n are very small quantities, difficult to estimate using standard simulation. In the next section, we develop fast simulation methods, based on importance sampling, to efficiently estimate γ_n and/or $E(O_n)$. These methods can be validated by comparing statistical output from simulations of the $M/M/1/k$ queue with the above analytical results.

2.3 IMPORTANCE SAMPLING

In a $GI/GI/1/k$ queue, let us consider the estimation of the probability of reaching full-buffer in a busy cycle, γ (see Section 2 for notation). This probability can be expressed as $\gamma = E_f(I(T_{fb} < T))$, where T_{fb} is a r.v. denoting the time to reach a full buffer in a busy cycle, and T is a r.v. denoting the cycle time (as defined in Section 2). $I(.)$ is the indicator function. Note that $T_{fb} = \infty$ for a busy cycle in which the buffer is never full. The subscript f denotes the underlying original probability measure (i.e., the original arrival and service processes). Using standard simulation we generate n independent busy cycles to obtain samples of $I(T_{fb} < T)$, say, $I_1, I_2, ..., I_n$. Then $\hat{\gamma} = \sum_{i=1}^{n} I_i/n$ is an unbiased estimator of γ. The variance of this estimator is given by $Var_f(I(T_{fb} < T))/n$, where $Var_f(I(T_{fb} < T)) = E_f(I^2(T_{fb} < T)) - E_f^2(I(T_{fb} < T)) = \gamma - \gamma^2$. From the central limit theorem (CLT) we have $\sqrt{n}(\hat{\gamma} - \gamma) \rightarrow N(0, Var_f(I(T_{fb} < T)))$. The CLT approximation can be used to obtain a 99% confidence interval (CI), the half width (HW) of which is given by $2.56\sqrt{Var_f(I(T_{fb} < T))/n}$. The relative error ($RE$) is defined as the ratio $HW/\gamma \approx 2.56/\sqrt{n\gamma}$. Obviously, for a fixed n, $RE \rightarrow \infty$ as $\gamma \rightarrow 0$. This is the problem when using standard simulation to estimate the probability of a rare event, such as γ. Importance sampling can be used to overcome this inherent problem.

Now, let g be another underlying probability measure, and ω be a sample path (e.g., a busy cycle) in the set $\mathbf{\Omega}$ of all possible sample paths. Denote by $dg(\omega)$ the probability of the sample path ω according to the new probability measure g. (Similarly, $df(\omega)$ is the probability of the sample path ω according to the original probability measure f.) Note that γ can be written as follows

$$\begin{aligned} \gamma &= \int_{\omega \in \mathbf{\Omega}} I_\omega(T_{fb} < T)\, df(\omega) = \int_{\omega \in \mathbf{\Omega}} I_\omega(T_{fb} < T)\, \frac{df(\omega)}{dg(\omega)}\, dg(\omega) \\ &= \int_{\omega \in \mathbf{\Omega}} I_\omega(T_{fb} < T) L(\omega)\, dg(\omega) = E_g(I(T_{fb} < T)L), \end{aligned} \tag{9}$$

where $I_\omega(.)$ is the indicator function evaluated for sample path ω, and $L(\omega) = df(\omega)/dg(\omega)$ is the likelihood ratio. It is clear from the above equation that the only condition imposed on the new probability measure g is: $dg(\omega) > 0$ whenever $I_\omega(T_{fb} < T)\, df(\omega) > 0$. It follows that we can simulate the system using the new probability measure g to obtain n independent samples of $I(T_{fb} < T)L$, say, $I_1L_1, I_2L_2, ..., I_nL_n$. An unbiased estimate of γ is given by $\hat{\gamma} = \sum_{i=1}^{n} I_iL_i/n$. The variance of this estimator is $Var_g(I(T_{fb} < T)L)/n = (E_g(I(T_{fb} < T)L^2) - \gamma^2)/n$. Notice that a zero variance estimator is obtained if we choose the new probability measure g such that for all $\omega \in \mathbf{\Omega}$, $dg(\omega) = I_\omega(T_{fb} < T)\, df(\omega)/\gamma$. However, this is not possible, since it requires the knowledge of γ, the quantity we are trying to estimate! The main challenge in importance sampling is to find a robust and easily implementable new probability measure g such that

$$E_g(I(T_{fb} < T)L^2) = E_f(I(T_{fb} < T)L) \ll E_f(I(T_{fb} < T)). \tag{10}$$

This means that the variance of the importance sampling estimate is much less than the variance of the standard simulation estimate. In other words, for the same simulation effort (e.g., the same number of busy cycles n), importance sampling yields an estimate with much smaller relative error than that obtained using standard simulation. (This also implies a significant speed up of simulation time to achieve certain accuracy.) Notice from the above equation that much variance reduction is obtained if $L(\omega) = df(\omega)/dg(\omega) \ll 1$ whenever $I_\omega(T_{fb} < T) = 1$. That is, g should be chosen so as to significantly increase the probability of the rare event $\{T_{fb} < T\}$. An "effective" change of probability measure, g, is one for which the relative error (RE) remains bounded, also as the probability of the rare event tends to zero. This is

a desirable property which implies that the simulation effort (e.g., the number of samples n) to achieve a given relative error remains the same as the rare event becomes rarer. In some cases, this property may be established empirically for a given importance sampling technique, as will be demonstrated in our experimental results of Section 4.

3 FAST SIMULATION OF SIMPLE QUEUES

Consider a simple queue with a finite buffer. The cell arrival "rate" is assumed to be sufficiently smaller than the service "rate", so that reaching a *full-buffer* (or buffer overflow) is a rare event. Efficient simulation involving a rare full-buffer event has been considered by many (see, for example, Parekh and Walrand (1989) and Sadowsky (1991).) Another rare event of interest is the *n-consecutive-cell-loss* event, which may occur only after the full-buffer is reached. In this section we consider these two related rare events, and develop an importance sampling heuristic to speed up simulations involving a rare consecutive-cell-loss event.

3.1 Rare Full-Buffer Event

In a $GI/GI/1/k$ queue, let us again consider the estimation of the probability of reaching full-buffer in a busy cycle, γ. As in Section 2.3, this probability can be expressed as $\gamma = E_f(I(T_{fb} < T))$, where the expectation is taken with respect to the original probability measure f. Since $\{T_{fb} < T\}$ is a rare event (i.e., $\gamma \approx 0$), using standard simulation is very inefficient, as it yields 0 for the indicator function on almost all busy cycles. Using importance sampling, we have $\gamma = E_f(I) = E_g(IL)$, where f and g are the original and the new probability measures, respectively, and L is the likelihood ratio. Denote by $dg(\omega)$ the probability of a sample path ω according to the new probability measure g. (Similarly, $df(\omega)$ is the probability of a sample path ω according to the original probability measure f.) Then $L(\omega) = df(\omega)/dg(\omega)$ is the likelihood ratio associated with a sample path ω; it can be computed easily during the simulation. For example, let $t^i_{A,j}$ (resp., $t^i_{S,j}$), $i = 1, 2, ..., N_j$, be the cell arrival (resp., departure) instants in the j-th busy cycle. Furthermore, let $g^i_{A,j}(t)$ (resp., $g^i_{S,j}(t)$) be the new i-th inter-arrival (resp., service) time density used to simulate the system with importance sampling. The likelihood ratio, L_j, associated with the j-th busy cycle, takes the form

$$L_j = \prod_{i=1}^{N_j} \frac{f_A(t^{i+1}_{A,j} - t^i_{A,j})}{g^i_{A,j}(t^{i+1}_{A,j} - t^i_{A,j})} \times \frac{f_S(t^i_{S,j} - t^{i-1}_{S,j})}{g^i_{S,j}(t^i_{S,j} - t^{i-1}_{S,j})}. \tag{11}$$

Note that $t^0_{S,j+1} = t^{N_j+1}_{A,j} = t^1_{A,j+1}$ is the instant at which the j-th busy cycle ends and the $j+1$-th busy cycle begins. Thus, L_j can be computed recursively at arrival and departure events during the simulation.

Now, let b be the number of independent "biased" (using importance sampling) busy cycles used to obtain estimates for the mean and the variance of the r.v. IL. These estimates are given by

$$\hat{\mu}_I = \sum_{j=1}^{b} I_j L_j / b, \quad \hat{\sigma}^2_I = \sum_{j=1}^{b} (I_j L_j - \hat{\mu}_I)^2/(b-1).$$

From the central limit theorem, for large b, the estimate $\hat{\mu}_I$ is approximately normally distributed. It follows that the relative half-width (in percentage) of the 99% confidence interval for the above estimator is given by $2.56\,(\hat{\sigma}_I/\hat{\mu}_I) \times 100$.

In the following we consider the optimal change of measure (importance sampling distribution) to efficiently estimate γ. Let $F_A(\theta) = \int_{t=0}^{\infty} e^{\theta t} f_A(t)dt$ be the moment generating function of the inter-arrival times. Define $f_A^{\theta}(t) = e^{\theta t} f_A(t)/F_A(\theta)$; this is another pdf obtained by exponentially tilting (twisting) the pdf $f_A(t)$ at a parameter θ. Similarly, $F_S(\theta) = \int_{t=0}^{\infty} e^{\theta t} f_S(t)dt$ is the moment generating function of the service times, and $f_S^{\theta}(t) = e^{\theta t} f_S(t)/F_S(\theta)$ is the corresponding exponentially tilted pdf.

Using heuristic arguments based on the theory of large deviations (Bucklew 1990), Parekh and Walrand (1989) proposed an importance sampling distribution to efficiently estimate the probability of buffer overflow in a $GI/GI/1/k$ queue. In Sadowsky (1991), this distribution was proved to be the unique asymptotically (as $k \to \infty$) optimal change of measure. Let θ^* be the solution of the equation

$$F_A(-\theta^*)\, F_S(\theta^*) = 1. \tag{12}$$

Then the optimal change of measure is obtained by simulating the $GI/GI/1/k$ queue with the exponentially tilted densities $g_A(t) = f_A^{-\theta^*}(t)$ and $g_S(t) = f_S^{\theta^*}(t)$. Importance sampling is "turned on" at the start of each busy cycle, and is "turned off" at the occurrence of the rare event. The moment generating functions for the new (optimal) inter-arrival and service times are given by

$$G_A(\theta) = \frac{F_A(\theta - \theta^*)}{F_A(-\theta^*)}, \quad G_S(\theta) = \frac{F_S(\theta + \theta^*)}{F_S(\theta^*)}. \tag{13}$$

Consider the $M/M/1/k$ queue with its arrival rate λ much smaller than its service rate μ (i.e., $\lambda \ll \mu$), so that a full buffer is a rare event. $F_A(-\theta) = \lambda/(\lambda + \theta)$ and $F_S(\theta) = \mu/(\mu - \theta)$, for $\theta < \mu$. Solving the equation $F_A(-\theta^*)\, F_S(\theta^*) = 1$ for θ^*, we get $\theta^* = \mu - \lambda$. It follows that $G_A(\theta) = \mu/(\mu - \theta)$ and $G_S(\theta) = \lambda/(\lambda - \theta)$, i.e., optimally, the $M/M/1/k$ queue is simulated with arrival rate μ and service rate λ. This change of measure accelerates the arrival process relative to the service process, thus increasing the probability of a full buffer in the simulated system.

In the next section, we use the optimal importance sampling distribution (as outlined above) in a heuristic to estimate very small consecutive-cell-loss probabilities.

3.2 Rare Consecutive-Cell-Loss Event

In this section we consider the estimation of the probability of losing n or more consecutive cells in a busy cycle, γ_n (see Section 2 for notation). This probability can be expressed as $\gamma_n = E_f(I(T_n < T))$, where the expectation is taken with respect to the original probability measure f. T_n is a r.v. denoting the time to the first n-consecutive-cell-loss event in a busy cycle, and T is a r.v. denoting the cycle time (also defined in Section 2). Note that $T_n = \infty$ for a busy cycle in which there is no n consecutive cell loss. Here too, since $\{T_n < T\}$ is a rare event, using standard simulation is very inefficient. In fact, the event $\{T_n < T\}$ must be at least as rare as the event $\{T_{fb} < T\}$, since the former may or may not occur only after the latter has occurred. Using importance sampling, we have $\gamma_n = E_f(I) = E_g(IL)$, where f and g are the original and the new probability measures, respectively, and L is the likelihood ratio. Based on b independent "biased" (using importance sampling) busy cycles, estimates of the mean $\hat{\mu}_I$ and the variance $\hat{\sigma}_I^2$ (and hence confidence intervals) are obtained as described in Section 3.1.

To the best of our knowledge, the problem of estimating the probability of a rare consecutive-cell-loss event (γ_n) using importance sampling has not been considered before. Note that this rare event can only occur during a full-buffer period, i.e., after the occurrence of a typically rare full-buffer event. Therefore, it seems intuitive to use two "biasing" (importance sampling) schemes, one to reach a full-buffer, and another, if necessary, to lose n consecutive cells during that full-buffer period. The main idea of our importance sampling heuristic is to use the optimal change of measure to reach the full-buffer state (as described in Section 3.1.) Once (and every time, until the consecutive loss of n cells) the full-buffer state is reached, additional "biasing" (e.g., by increasing the arrival "rate") is applied (if necessary) to increase the probability of n or more arrivals (losses) during the full-buffer period. "Biasing" is turned off as soon as the rare event of interest occurs, i.e., n arrivals during a full-buffer period. Otherwise, "biasing" is continued according to the optimal change of measure (of Section 3.1) until the next full-buffer period or the end of the busy cycle. The implementation details of "biasing" during full-buffer periods may differ depending on the particular arrival and service processes being considered. These details will be discussed for each of the models used in our experiments of Section 4. Empirical results from these experiments demonstrates the effectiveness of the above importance sampling heuristic to estimate γ_n. The same heuristic can also be used to estimate $\mathcal{F}_n$, the frequency of the n-consecutive-cell-loss event. In either case, several orders of magnitude "speed ups" over standard simulation can be obtained.

It is important to mention that, in general, the simulation effort (with importance sampling) slowly increases with the number of consecutive cell loss of interest, i.e., the importance sampling scheme is not asymptotically (as $n \to \infty$) efficient. (This can, perhaps, be seen from the experimental results for the $M/D/1/k$ queue in Section 4.3.) However, this is not due to the increased rarity of the n-consecutive-cell-loss event, but due to increase in the inherent variance of the probability of n or more arrivals during a full-buffer period. Let V be a r.v. denoting the length of a full-buffer period, then for Poisson arrivals with a rate λ, this probability is given by $P_n(V) = e^{-\lambda V} \sum_{i=n}^{\infty} (\lambda V)^i / i!$. Clearly, the variance of $P_n(V)$ increases with the variance of V and is amplified for high values of n. It is this inherent increase in variability which cannot be reduced by importance sampling. In fact, for an $M/M/1/k$ queue, the full-buffer periods, V, are independent and exponentially distributed with a mean $1/\mu$. In this case, samples of $P_n(V)$ observed during simulation can be replaced by their (deterministic) mean $p_n = (\frac{\lambda}{\lambda+\mu})^n$. This way, the variability of $P_n(V)$ does not affect the simulation results. Indeed, for an $M/M/1/k$ queue, this special implementation of our heuristic is asymptotically efficient (as $n \to \infty$), which is clearly demonstrated by the empirical results in Section 4.1.

4 EXPERIMENTAL RESULTS

In this section we use fast simulation methods discussed in Sections 3.1 and 3.2 to evaluate a model of the Leaky Bucket (LB) algorithm. For validation purposes, the simulation of an $M/M/1/k$ queue is considered in Section 4.1. The operation of the LB algorithm and its model are described in Section 4.2. The evaluation of this model is considered in Sections 4.3 and 4.4, for Poisson and two-phase burst/silence (TPBS) cell arrival processes, respectively. The empirical results displayed here include estimates of γ_n (i.e., the probability of losing n or more consecutive cells in a busy cycle), $E(O_n)$ (i.e., the expected number of n-consecutive-cell-loss events in a busy cycle) and $\mathcal{F}_n$ (i.e., the steady-state frequency of the n-consecutive-cell-loss event.)

4.1 Simulation of the $M/M/1/k$ Queue

In this section we consider the efficient simulation of an $M/M/1/k$ queue to estimate the probability of consecutive cell loss in a busy cycle. For this model, analytical results in Section 2.2 can be used to validate statistical output from simulation. As outlined in Section 3.2 our importance sampling heuristic makes use of two different "biasing" schemes. The first is optimal "biasing" (as described in Section 3.1) to reach the full-buffer state (i.e., the $M/M/1/k$ queue is simulated with arrival rate μ and service rate λ.) The second is "biasing" during full-buffer periods, which in the special case of an $M/M/1/k$ queue can be implemented as follows. As argued in Section 3.2, the probability of n or more arrivals (losses) during a full-buffer period is given by $p_n = (\frac{\lambda}{\lambda+\mu})^n$, which is typically very small in the original queue. In the simulated queue, we increase this probability to p_s (a constant sufficiently higher than p_n; for example, $p_s = 0.5$). With probability p_s, the full-buffer period is considered to be a "successful" overload period (i.e., having n or more arrivals). Let U be a uniform random variable ($0 < U < 1$). Every time (until the consecutive loss of n cells) the full-buffer state is reached, we take a sample u of U. If $u \leq p_s$, then the n-consecutive-cell-loss event is considered to have occurred, and "biasing" is turned off until the end of the current busy cycle. In this case, the likelihood ratio is updated by the multiplication factor p_n/p_s. (Note that in this implementation, a sample of the full-buffer period need not be generated, and the simulation is continued, from the instant of reaching the full-buffer state, as if a departure event has just occurred leaving the queue with $k-1$ cells.) Otherwise, if $u > p_s$, then the n-consecutive-cell-loss event is considered to have not occurred, and "biasing" is continued as described in Section 3.1 until the next full-buffer period or the end of the current busy cycle. In this case, the likelihood ratio is updated by the multiplication factor $(1-p_n)/(1-p_s)$.

Now let us consider the $M/M/1/k$ queue with $\lambda = 0.8$ cells per unit of time, $\mu = 1.0$ cells per unit of time and $k = 25$. In Table 1, for increasing n, we give fast simulation estimates of the cycle-based quantities; namely, the n-consecutive-cell-loss probability (γ_n) and the expected number of n-consecutive-cell-loss events $E(O_n)$. Numerical results from analysis are also displayed. Consistent with our remark in Section 2.2, note that $E(O_n) \approx \gamma_n$ for values of $n \geq 8$. Also, Note that the frequency $\mathcal{F}_n$ can be determined by $E(O_n)/E(N) = P_l\, E(O_n)$, where $P_l = 1 - \frac{\lambda}{\mu}$.

Using different arrival and service rates, experiments indicate that for high n, the lowest relative error can be obtained by setting p_s (approximately) to $1 - \frac{\lambda}{\mu}$. Therefore, the "biasing" probability p_s is heuristically set to $max(p'_n, 1 - \frac{\lambda}{\mu})$, where p'_n is the (new) probability of n or more arrivals during a full-buffer period in the simulated system (i.e., with the optimal change of measure as given in Section 3.1.) For the simulated $M/M/1/k$ queue, it follows that $p'_n = (\frac{\mu}{\lambda+\mu})^n$. 25600 "biased" busy cycles were simulated to get the estimates and their relative error (i.e., the relative half-width of the 99% confidence interval) in percentage. Note that fast simulation results are in good agreement with the numerical results from analysis. Also, the relative error does not increase for larger values of n; this verifies the asymptotic optimality of the particular implementation of our proposed importance sampling method when applied to the $M/M/1/k$ queue.

4.2 The Leaky Bucket (LB) Algorithm

An ATM connection is established with an admission contract which specifies the traffic characteristics of the source and the quality of service (QoS) to be guaranteed by the network. In order for the network to ensure that the admission contract is not violated, the usage parameter control (UPC) procedure is invoked to monitor the actual traffic and to police the excess traffic violating the contract. The Leaky

Bucket (LB) algorithm is a popular UPC procedure and can easily be implemented with counters (see Turner (1986).) Each time a cell arrives, the counter is incremented by one. As long as the counter has a positive value, it is decremented at fixed intervals, d. When the cell arrival "rate" exceeds the periodic decrement "rate," the counter value will increase. If the counter reaches a pre-specified limit, say, k, then the source is considered to have exceeded its admission contract, and subsequent cells are discarded (or marked for policing) until the counter value falls below the limit again. The operation of this LB algorithm can be modeled as a $GI/D/1/k$ queue, in which the service time is deterministic and identical to the decrement interval, d. An arriving cell is lost if it finds a full buffer.

For a two-phase burst/silence source model (see Section 4.4), the stationary cell loss probability can be obtained by a numerical method whose complexity grows in proportion to the value of k (Rathgeb 1991.) No analytical or numerical method is available yet to obtain the probability of consecutive cell loss in a $GI/D/1/k$ queue. In order to avoid restrictions necessary for analytic tractability and/or numerical feasibility, simulation is often preferred for the evaluation of realistic models of the LB algorithm. However, standard simulation is not efficient because consecutive cell loss is a rare event. Accurate and efficient estimation of very small probabilities, such as γ, using importance sampling has been considered in Nicola et al. (1994). In the next two sections, we use the importance sampling heuristic proposed in Section 3 to efficiently estimate γ_n, $E(O_n)$ and $\mathcal{F}_n$ in a model of the LB algorithm with (non-bursty) Poisson and (bursty) TPBS cell arrival processes.

4.3 Poisson Cell Arrival Process

In this section we use importance sampling to efficiently estimate the probability of consecutive cell loss in a busy cycle of an $M/D/1/k$ queueing model of the LB algorithm (i.e., for a Poisson cell arrival process). The arrival rate is λ and the service time is a constant d. As outlined in Section 3.1, the optimal change of measure to reach the full-buffer state can be obtained by solving Equation (12) for θ^*. The corresponding inter-arrival and service time densities can now be determined from their generating functions as given in Equation (13). It follows that the optimal service times are also deterministic and identical to the original (i.e., no change in the service process.) However, the arrival process does change, so as to increase the probability of the rare full-buffer event. We note that full-buffer periods (i.e., the actual remaining service time upon reaching the full-buffer state) in the same busy cycle are neither independent nor identically distributed. Therefore, in this implementation, these full-buffer periods must be simulated (unlike the implementation for the $M/M/1/k$ queue). The probability of n or more arrivals (losses) during a full-buffer period depends on the remaining service time ($r < d$) and is given by $P_n(r) = e^{-\lambda r} \sum_{i=n}^{\infty} (\lambda r)^i / i!$. This probability is typically very small in the original system, and, therefore, "biasing" is necessary to increase the probability of "success" (i.e., n or more arrivals) during the full-buffer period. In the simulated queue, we increase this probability to p_s (a constant sufficiently higher than $P_n(r)$; for example, $p_s = 0.5$). Every time (until the consecutive loss of n cells) the full-buffer state is reached, we take a sample u of a uniform random variable U (defined in Section 4.1). If $u \leq p_s$, then the n-consecutive-cell-loss event is considered to have occurred, and "biasing" is turned off until the end of the current busy cycle. In this case, at the end of the full-buffer period, the likelihood ratio is updated by the multiplication factor $P_n(r)/p_s$. Otherwise, if $u > p_s$, then the n-consecutive-cell-loss event is considered to have not occurred, and "biasing" is continued immediately after the full-buffer period (as described in Section 3.1) and until the next full-buffer period or the end of the current busy cycle. In this case, at the end of the full-buffer period, the likelihood ratio is updated by the multiplication factor $(1 - P_n(r))/(1 - p_s)$.

Note that when the full-buffer period r is very small (i.e., $r \ll d$), "biasing" may yield non-typical sample paths, resulting in extremely small values for the likelihood ratio and leading to unstable estimates.

To overcome this problem, the above heuristic is modified as follows. Upon reaching the full-buffer state, $P_n(r)$ is determined, and "biasing" during the full-buffer period (as outlined above) is activated only if, say, $P_n(r)/P_n(d) \geq 4 \times 10^{-3}$. In this way, "biasing" is activated only when a full-buffer period is sufficiently large to yield a rare (but typical) sample path. As long as the consecutive-cell-loss event did not occur, "biasing" to reach the next full-buffer period is resumed as outlined above. The following example shows that the above heuristic with this modification is quite robust and effective.

Now let us consider the model of the LB algorithm with a Poisson cell arrival process at rate $\lambda = 0.8$ cells per unit of time. The new (optimal) arrival process to reach the full-buffer state is also Poisson, however, at an increased rate $\lambda^* = \lambda + \theta^*$, where (from Equation (12)) θ^* is the non-trivial solution of $\lambda + \theta^* = \lambda e^{d\theta^*}$. The (deterministic) service time is set to $d = 1$ time unit, $k = 10$, and we vary the number of consecutive cell loss, n. In Table 2, we list fast simulation estimates of γ_n and $E(O_n)$ as well as their relative error (i.e., the relative half-width of the 99% confidence interval) in percentage. 25600 "biased" busy cycles were used to get these estimates. Using different arrival rates and/or service times, the best relative error (for high values of n) is obtained by setting p_s (approximately) to $1 - \lambda d$. Therefore, the "biasing" probability p_s is heuristically set to $max(P'_n(r), 1 - \lambda d)$, where $P'_n(r)$ is the (new) probability of n or more arrivals during the full-buffer period r in the simulated system (i.e., with the increased optimal arrival rate λ^*.) For the simulated $M/D/1/k$ queue, it follows that $P'_n(r) = e^{-\lambda^* r} \sum_{i=n}^{\infty} (\lambda^* r)^i / i!$. Note that if "biasing" is not activated in a full-buffer period because $r \ll d$, then $p_s = P_n(r)$, and the likelihood ratio is not updated at the end of the full-buffer period. Using the same effort (in CPU time), standard simulation yields meaningful results for only two entries with relatively high probabilities. As can be seen, the relative error of the fast simulation estimates slowly increases with n, which is an indication that the importance sampling heuristic is not asymptotically efficient with respect to n. As explained in Section 3.2, this is due to the increased variability of $P_n(V)$ for higher n, where V is a r.v. denoting the length of a full-buffer period. Note that $E(O_n) \approx \gamma_n$ for values of $n \geq 4$, which validates our remark in Section 2.2 for queues other than the $M/M/1/k$.

4.4 Bursty Cell Arrival Process

In this section we consider the evaluation of the LB algorithm for a more realistic two-phase burst/silence cell arrival process (see Rathgeb (1991)), which we will refer to as TPBS process. This arrival process has been used to model bursty sources, such as packetized voice (see Heffes and Lucantoni (1986)) and interactive data services, and, therefore, it is often used to compare various policing mechanisms. The number of cells per burst is geometrically distributed with a parameter α, and the inter-cell time during a burst is deterministic given by τ. Therefore, transitions from burst to silence occur with a probability α, only at multiples of τ. The duration of the silence phase is exponentially distributed with a mean β^{-1}. The peak cell arrival "rate" is $1/\tau$, and the average cell arrival "rate" $\lambda = (\tau + \alpha/\beta)^{-1}$. Note that we can increase the burstiness of the cell arrival process by increasing the average burst length (i.e., smaller α) while keeping the average cell "rate" the same (i.e., constant α/β.) The pdf of the TPBS inter-arrival time and its moment generating function are given by

$$f_A(t) = \begin{cases} 0, & \text{if } t < \tau, \\ 1 - \alpha, & \text{if } t = \tau, \\ \alpha\, \beta e^{-\beta(t-\tau)}, & \text{if } t > \tau, \end{cases} \tag{14}$$

$$F_A(\theta) = e^{\theta\tau} \left[(1 - \alpha) + \alpha \frac{\beta}{\beta - \theta} \right]. \tag{15}$$

$g_A(t) = f_A^{-\theta^*}(t)$ is the corresponding exponentially tilted pdf (with a tilting parameter θ^*); its moment generating function is given by $G_A(\theta) = F_A(\theta - \theta^*)/F_A(-\theta^*)$. It can be shown that the tilted pdf, $g_A(t)$, is also a TPBS process with the same deterministic burst inter-cell time τ, and with its parameters, $\beta^* = \beta + \theta^*$, and $\alpha^* = \alpha\beta/(\beta + (1-\alpha)\theta^*)$. The tilted pdf, $g_A(t)$, is used as the (new) inter-arrival time density for simulation with importance sampling to reach the full-buffer state.

For a TPBS cell arrival process, the LB algorithm can be modelled as $TPBS/D/1/k$ queue. Since the full-buffer and the consecutive cell loss are typically rare events, importance sampling is used to efficiently simulate this system. At the beginning of each busy cycle, and after each full-buffer period (as long as the rare consecutive-cell-loss event has not occurred), "biasing" to reach the next full-buffer period is affected as described in Section 3.1. The new "biased" (TPBS) cell arrival process is determined by α^*, β^* and τ, as given above. The service time is deterministic (d), and, therefore, remains unchanged in the simulated system. As soon as the full-buffer state is reached, further "biasing" during the full-buffer period may be necessary to accelerate the n-consecutive-cell-loss event. Since the inter-cell time during a burst (τ) is deterministic, the number of cells that may be lost during a full-buffer period of length r cannot exceed a maximum given by $n_{max} = \lfloor r/\tau \rfloor$. At the beginning of a full-buffer period of length r, if $n \leq n_{max}$, then "biasing" is done by setting the new β to β^*. If α^* is not sufficient to increase the probability of n or more remaining cells in the current burst to a high value, p_s (for example, $p_s = 0.5$.), then the new α is set to α_s as determined from $(1-\alpha_s)^n = p_s$ (i.e., $\alpha_s = 1 - e^{\ln(p_s)/n}$.) In other words, until the consecutive loss of n cells, we use the optimal "biasing" to reach the full-buffer state (i.e., the new α is set to α^* and the new β is set to β^*.) In addition, depending on n and r, more (stronger) "biasing" during the full-buffer period may be necessary (i.e., if $n \leq n_{max}$, then the new α is set to $\min(\alpha^*, \alpha_s)$.) The effectiveness of this heuristic is demonstrated in one example. In another example, we use the heuristic to experiment with the burstiness of the cell arrival process.

In the first experiment, we consider a TPBS cell arrival process with $\alpha = 0.2$, $\beta = 5.0 \times 10^{-4}$ and $\tau = 1$. The (deterministic) service time, d, is set to 100 time units, and k is set to 30. In Table 3, the number of consecutive cell loss, n, is varied, and we give fast simulation estimates of γ_n and $\mathcal{F}_n$, with their percentage relative error (i.e., the relative half-width of the 99% confidence interval.) 25600 "biased" busy cycles were used to get these estimates. For all n, the "biasing" probability, p_s, is set to 0.5. It is not directly seen from the table, however, it is interesting to point out that, for smaller values of n, stronger "biasing" during full-buffer periods is not necessary (i.e., the new α is set to α^*.) For relatively high consecutive-cell-loss probabilities, it was possible to compare with results from standard simulation using the same effort (in CPU time.) Note that the relative error of the fast simulation estimates slowly increases with n, i.e., the importance sampling heuristic is not asymptotically efficient with respect to n. A similar observation was made in the experiment for the $M/D/1/k$ queue in Section 4.3.

In the second experiment, we consider a TPBS arrival process, in which we increase the burstiness, while fixing the average cell arrival "rate." As described earlier in this section, this can be achieved by decreasing α and β, while fixing α/β. We set $\tau = 1$ and $\lambda = 1/50$. It follows that α/β is fixed at 49. The (deterministic) service time, d, is set to 25 time units, and k is set to 100. For a fixed number of consecutive cell loss, $n = 5$, in Table 4 we vary the burstiness and give the fast simulation estimates of γ_n and $\mathcal{F}_n$, with their percentage relative error. 25600 "biased" busy cycles were used to get these estimates. For all values of α, the "biasing" probability, p_s, is set to 0.5. Using the same effort (in CPU time), only for relatively high probabilities, it is possible to obtain meaningful results from standard simulation. As expected, the empirical results in Table 4 indicate a sharp increase in the consecutive-cell-loss probability due to increased burstiness.

5 CONCLUSIONS

In this paper we have proposed a heuristic importance sampling change of measure to efficiently estimate the probability of a rare consecutive-cell-loss event in a $GI/GI/1/k$ queue. This heuristic makes use of the optimal change of measure proposed by Parekh and Walrand (1989) to accelerate the occurrence of a rare full-buffer event in an asymptotically stable queue. However, further "biasing" is necessary to increase the probability of a rare consecutive-cell-loss event during a full-buffer period. Experimental results demonstrate the validity and effectiveness of our fast simulation method, which is used for the evaluation of a $GI/D/1/k$ queueing model of the Leaky Bucket algorithm.

ACKNOWLEDGEMENT

The authors wish to thank Fokke Hoeksema for bringing this problem to their attention, and Erik van Doorn for useful discussions on the analysis.

REFERENCES

Bucklew, J.A. (1990) *Large Deviation Techniques in Decision, Simulation and Estimation.* New York, NY: J. Wiley & Sons, Inc.

Chang, C.S., P. Heidelberger, S. Juneja and P. Shahabuddin. (1993) Effective bandwidth and fast simulation of ATM intree networks. In *Proc. of the Performance '93 conference.*

Cooper, R.B. (1981) *Introduction to Queueing Theory.* London: Arnold.

Hammersley, J.M. and D.C. Handscomb. (1964) *Monte Carlo Methods.* London: Methuen.

Heffes, H. and D.M. Lucantoni. (1986) A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Select. Areas Commun.* **4, 6**: 856-868.

Heidelberger, P. (1993) Fast simulation of rare events in queueing and reliability models. In *Models and Techniques for Performance Evaluation of Computer and Communications Systems*, Springer-Verlag, Lecture Notes in Comp. Sc., **729**: 165-202.

Nicola, V.F., P. Shahabuddin and P. Heidelberger. (1993) Techniques for fast simulation of highly dependable systems. In *Proc. of the Second International Workshop on Performability Modelling of Computer and Communication Systems.*

Nicola, V.F., G.A. Hagesteijn and B.G. Kim. (1994) Fast simulation of the Leaky Bucket algorithm. In *Proceedings of the 1994 Winter Simulation Conference*, IEEE Press, 266-273.

Parekh, S. and J. Walrand. (1989) A quick simulation method for excessive backlogs in networks of queues. *IEEE Trans. Autom. Contr.* **34, 1**: 54-66.

Rathgeb, E.P. (1991) Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE J. Select. Areas Commun.* **9, 3**: 325-334.

Sadowsky, J.S. (1991) Large deviations theory and efficient simulation of excessive backlogs in a $GI/GI/m$ queue. *IEEE Trans. Autom. Contr.* **36, 12**: 1383-1394.

Turner, J.S. (1986) New directions in communications (or which way to the information age?). *IEEE Commun. Mag.* **25, 10**: 8-15.

Table 1 Estimates of γ_n and $E(O_n)$ in an $M/M/1/k$ Queue

	γ_n		$E(O_n)$	
	Fast Sim.	Anal.	Fast Sim.	Anal.
full-buffer	9.45×10^{-4} $\pm$ 3.20%	9.48×10^{-4}	4.66×10^{-3} $\pm$ 4.52%	4.72×10^{-3}
$n = 1$	7.69×10^{-4} $\pm$ 3.16%	7.58×10^{-4}	2.10×10^{-3} $\pm$ 4.20%	2.10×10^{-3}
$n = 4$	1.58×10^{-4} $\pm$ 3.25%	1.59×10^{-4}	1.81×10^{-4} $\pm$ 3.50%	1.84×10^{-4}
$n = 8$	7.12×10^{-6} $\pm$ 3.21%	7.15×10^{-6}	7.15×10^{-6} $\pm$ 3.21%	7.19×10^{-6}
$n = 16$	1.09×10^{-8} $\pm$ 3.21%	1.09×10^{-8}	1.09×10^{-8} $\pm$ 3.21%	1.09×10^{-8}
$n = 32$	2.53×10^{-14} $\pm$ 3.21%	2.54×10^{-14}	2.53×10^{-14} $\pm$ 3.21%	2.54×10^{-14}
$n = 64$	1.36×10^{-25} $\pm$ 3.21%	1.36×10^{-25}	1.36×10^{-25} $\pm$ 3.21%	1.36×10^{-25}

Table 2 Estimates of γ_n and $E(O_n)$ in an $M/D/1/k$ Queue

	γ_n		$E(O_n)$	
	Std. Sim.	Fast Sim.	Std. Sim.	Fast Sim.
full-buffer	1.00×10^{-2} $\pm$ 4.49%	9.92×10^{-3} $\pm$ 2.15%	4.87×10^{-2} $\pm$ 5.99%	4.80×10^{-2} $\pm$ 3.21%
$n = 1$	6.47×10^{-3} $\pm$ 4.76%	6.38×10^{-3} $\pm$ 2.22%	1.43×10^{-2} $\pm$ 5.91%	1.40×10^{-2} $\pm$ 3.00%
$n = 4$		8.20×10^{-5} $\pm$ 3.48%		8.28×10^{-5} $\pm$ 3.50%
$n = 8$		9.68×10^{-9} $\pm$ 4.23%		9.68×10^{-9} $\pm$ 4.23%
$n = 12$	—	2.15×10^{-13} $\pm$ 4.97%	—	2.15×10^{-13} $\pm$ 4.97%
$n = 16$	—	1.49×10^{-18} $\pm$ 5.56%	—	1.49×10^{-18} $\pm$ 5.56%

Table 3 Estimates of γ_n and $\mathcal{F}_n$ in a $TPBS/D/1/k$ Queue

	γ_n		$\mathcal{F}_n$	
	Std. Sim.	Fast Sim.	Std. Sim.	Fast Sim.
full-buffer	6.14×10^{-3} ± 8.46%	6.15×10^{-3} ± 0.87%	1.29×10^{-3} ± 9.50%	1.28×10^{-3} ± 1.86%
$n = 1$	5.14×10^{-3} ± 9.16%	5.19×10^{-3} ± 0.87%	1.01×10^{-3} ± 10.13%	1.02×10^{-3} ± 1.82%
$n = 2$	4.27×10^{-3} ± 9.97%	4.36×10^{-3} ± 0.87%	8.12×10^{-4} ± 10.90%	8.18×10^{-4} ± 1.78%
$n = 4$	2.82×10^{-3} ± 12.06%	2.99×10^{-3} ± 0.89%	4.96×10^{-4} ± 12.88%	5.21×10^{-4} ± 1.73%
$n = 8$	1.18×10^{-3} ± 17.91%	1.31×10^{-3} ± 1.08%	1.88×10^{-4} ± 18.37%	2.10×10^{-4} ± 1.77%
$n = 16$		2.23×10^{-4} ± 1.49%		3.41×10^{-5} ± 2.01%
$n = 32$	—	5.70×10^{-6} ± 2.20%	—	8.62×10^{-7} ± 2.57%
$n = 64$	—	2.76×10^{-9} ± 3.51%	—	4.18×10^{-10} ± 3.75%

Table 4 Estimates of γ_n and $\mathcal{F}_n$ in a $TPBS/D/1/k$ Queue

	γ_n		$\mathcal{F}_n$	
	Std. Sim.	Fast Sim.	Std. Sim.	Fast Sim.
$\alpha = 0.05$	3.07×10^{-2} ± 4.04%	3.12×10^{-2} ± 1.50%	2.20×10^{-3} ± 5.22%	2.22×10^{-3} ± 2.83%
$\alpha = 0.10$	1.55×10^{-3} ± 14.46%	1.53×10^{-3} ± 1.53%	1.61×10^{-4} ± 17.54%	1.53×10^{-4} ± 2.80%
$\alpha = 0.15$	6.72×10^{-5} ± 58.73%	6.39×10^{-5} ± 1.55%	9.48×10^{-6} ± 71.29%	7.77×10^{-6} ± 2.73%
$\alpha = 0.20$	—	2.18×10^{-6} ± 1.60%	—	3.13×10^{-7} ± 2.70%
$\alpha = 0.25$		6.09×10^{-8} ± 1.65%		9.80×10^{-9} ± 3.18%
$\alpha = 0.30$		1.34×10^{-9} ± 1.72%		2.44×10^{-10} ± 3.12%
$\alpha = 0.35$	—	2.26×10^{-11} ± 1.71%	—	4.58×10^{-12} ± 2.54%
$\alpha = 0.40$	—	2.86×10^{-13} ± 1.79%	—	6.42×10^{-14} ± 3.03%

AUTHOR BIOGRAPHIES

VICTOR F. NICOLA holds the Ph.D. degree in computer science from Duke University, North Carolina. From 1979 he held scientific and research staff positions at Eindhoven University and Duke University. In 1987, he joined IBM T.J. Watson Research Center as a Research Staff Member. Since 1993 he holds an Associate Professor position with the group of Tele-Informatics and Open Systems at the University of Twente. His research interests include performance and reliability modeling of computer and communication systems, queueing theory, fault-tolerance and simulation.

GERTJAN A. HAGESTEIJN holds the M.Sc. degree in computer science from the University of Twente, The Netherlands. He is currently involved in the development of fast simulation techniques for the evaluation of ATM-based telecommunication systems.