Estimating Small Cell-Loss Ratios in ATM Switches via Importance Sampling

Pierre L'Ecuyer
GERAD and Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
C.P. 6128, Succ. Centre-Ville, Montréal, Canada H3C 3J7
lecuyer@iro.umontreal.ca

Yanick Champoux
Newbridge Networks Corporation
600 March Road, Kanata, Canada K2K 2E6
ychampou@newbridge.com

January, 1999

Les Cahiers du GERAD
G–99–07

Abstract

The cell-loss ratio at a given node in an ATM switch, defined as the steady-state fraction of packets of information that are lost at that node due to buffer overflow, is typically a very small quantity which is hard to estimate by simulation. Cell losses are rare events and importance sampling is normally the appropriate tool in this situation. However, finding the right change of measure is generally hard. In this paper, importance sampling is applied to estimate the cell-loss ratio in an ATM switch modeled as a queueing network fed by several sources emitting cells according to a Markov-modulated *on/off* process, and where all the cells from the same source have the same destination. The numerical experiments show impressive efficiency improvements.

Keywords: Importance sampling, variance reduction, rare events, ATM

Résumé

La fraction de cellules perdues à cause d'un débordement du tampon à un noeud donné d'un commutateur ATM, sur horizon infini, est une quantité habituellement très petite et difficile à estimer par simulation. Les pertes de cellules sont des événements rares et l'échantillonnage stratégique est l'outil habituel pour ce genre de situation. Il est toutefois très difficile en général de trouver un bon changement de loi de probabilité pour léchantillonnage stratégique. Dans cet article, nous appliquons cette méthode pour estimer la fraction de cellules perdues dans un commutateur ATM modélisé par un réseau de files d'attente nourri par plusieurs sources qui émettent des cellules selon un processus Markov-modulé à deux états ("*on/off*"), et pour lequel toutes les cellules d'une même source ont la même destination. Nos résultats numériques montrent une amélioration impressionnante de l'efficacité.

Acknowledgments: This work has been supported by NSERC-Canada grants no. ODGP0110050 and SMF0169893, and FCAR-Québec grant no. 93ER1654, to the first author, as well as a joint scholarship from NSERC-Canada and *Newbridge Networks Corporation* to the second author.

Introduction

An Asynchronous Transfer Mode (ATM) communication switch can be modeled as a network of queues with finite buffer sizes. Packets of information (called *cells*) join the network from several sources according to stochastic processes, and some cells may be lost due to buffer overflow. The long-term (or steady-state) fraction of cells that are lost at a given node is called the cell-loss ratio (CLR) at that node. Typical CLRs are small (e.g., less than 10^{-8}) and the cell losses tend to occur in bunches. Cell losses are thus so rare that estimating the CLR with good precision by straightforward simulation is very time-consuming, and in some cases practically impossible.

Importance sampling (IS) is the method of choice in such a situation. IS changes the probability laws governing the system so that the rare events of interest occur more frequently, eventually to the point of being no longer rare events. The estimator is modified accordingly so that it remains unbiased: It is multiplied by a quantity called the *likelihood ratio*. The hope is that the IS estimator is *more efficient*; i.e., that the product of its variance and its computing cost is smaller than for the regular estimator. The most difficult problem in applying IS is (in general) to figure out how to change the probability laws so that the variance gets reduced to an acceptable level. Theoretically, there always exists a change of measure that reduces the variance to an arbitrary small positive value, but finding it is usually much too complicated and not practical.

[Chang *et al.* 1994] derived an *asymptotically optimal* change of measure, based on the theories of *effective bandwidth* and *large deviations*, for estimating the probability p that a queue length exceeds a given level x before returning to empty, given that the queue is started from empty, for a *single queue* with multiple independent arrival sources. Roughly, *asymptotically optimal* means that the standard error of the IS estimator converges to zero exponentially fast with the same decay rate (exponent) as the quantity to be estimated, as a function of the level x. A precise definition can be found in [Chang *et al.* 1994]. An asymptotically optimal change of measure does not minimize the variance, but it can reduce it by several orders of magnitude. [Chang *et al.* 1994] extended their method to *intree* networks of queues, which are acyclic tree networks where customers flow only towards the root of the tree. For intree networks, they gave an upper bound on the variance of the IS estimator, and conjectured that this estimator is asymptotically optimal (or almost), but did not prove it. In numerical experiments with queueing models with a *single node*, or *two nodes in series*, they observed large variance reductions with their IS estimator.

The probability p just described is closely related to the CLR when x equals the buffer size (it measures almost the same events), so it seems quite reasonable to use the change of measure proposed by [Chang *et al.* 1994] to estimate the CLR as well, as pointed out by these authors themselves.

[Beck *et al.* 1998; Dabrowski *et al.* 1998] also study the application of IS to a discrete-time queueing network model of an ATM switch. Their model is very general. Assuming infinite buffers at all nodes, they obtain the asymptotics of the tail of the queue size distribution in steady-state, and they use that to propose a change of measure for estimating the CLR at a given node. Their IS methodology is related (but different) to that of [Chang *et al.* 1994].

For general background on efficiency improvement (or variance reduction), we refer the reader to [Bratley *et al.* 1987; Fishman 1996; Glynn 1994; L'Ecuyer 1994]. IS is well explained in [Glynn and Iglehart 1989; Heidelberger 1995; Shahabuddin1994] and the several other references given there. Application of IS to the simulation of communication systems is studied by [Bonneau 1996; Chang *et al.* 1994; Chang *et al.* 1995; Heegaard 1998], among others.

In this paper, we consider queueing networks having a large number of nodes, fed by a large number of Markov-modulated *on/off* sources. The nodes are organized in successive layers and each cell (or customer) goes through exactly one node of each layer, following a path uniquely determined by its source. This type of queueing network is a widely used model for the traffic in an ATM switch. We apply IS to estimate the CLR at any pre-specified node of the network, using a change of measure based on the same approach as [Chang *et al.* 1994]. We obtain spectacular efficiency improvements for both small and large networks.

The model is defined is Section 1. Section 2 recalls the A-cycles method and the batch-means method, which we use jointly to compute confidence intervals. In Section 3 we explain how IS is applied to estimate the CLR at a given target node. The idea is to increase the traffic to the target node by increasing the average *on/off* ratio for all the sources (and only those) feeding that node. The exact change of measure is determined by a heuristic. Numerical results are reported in Section 4. In Section 5, we consider various refinements of the basic IS scheme, and test them empirically to see how much additional variance reduction they can bring. Section 6 explains how the CLR can be estimated in functional form, as a function of certain parameters of the model. Additional numerical results and details can be found in [Champoux 1998]. A preliminary report of this work was presented in [L'Ecuyer and Champoux 1996].

1 The Model

We consider an acyclic queueing network with 4 layers of *nodes*, as illustrated in Figure 1. Each node is a single-server FIFO queue with finite buffer size. The ℓ-th layer is called *level* ℓ and the nodes at level 4 transmit cells to *destinations*. Levels 2 and 3 have m_2 nodes each, while levels 1 and 4 have $m_1 m_2$ nodes each. Each level-2 node is fed by m_1 level-1 nodes, while each level-3 node feeds m_1 nodes at level 4. *Cells* (i.e., packets of information) arrive at level 1, visit one node of each level, in succession, then leave the network. Each node at level 1 is fed by m_0 arrival *sources*. These $m_0 m_1 m_2$ sources are assigned to specific destinations; i.e., all the cells produced by a given source follow exactly the same path. The arrival sources are time-synchronized, but otherwise independent, stochastically identical, discrete-time *on/off* Markov modulated processes. A source is *off* for a while, then *on* for a while, then *off* again, and so on. The source produces one cell per unit of time during a *on* period, and none during a *off* period. The durations of *off* and *on* periods are independent geometric random variables with means κ_0 and κ_1, respectively, so the arrival rate is $\rho = \kappa_1/(\kappa_1 + \kappa_0)$. The parameter κ_1 is called the *average burst size*. If we denote *off* and *on* by 0 and 1, respectively, our assumptions imply that the state of a source evolves as a discrete-time Markov chain with two states, 0 and 1, with transition probability matrix

$$R = \begin{pmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{pmatrix} = \begin{pmatrix} 1 - 1/\kappa_0 & 1/\kappa_0 \\ 1/\kappa_1 & 1 - 1/\kappa_1 \end{pmatrix}. \tag{1}$$

These Markov chains comprise all the stochasticity of the model; everything else is deterministic. The arrival sources are numbered from 1 to $m_0 m_1 m_2$ and the nodes are numbered from 1 to $2m_2(1 + m_1)$, level by level. When two or more cells reach a given node simultaneously, they are placed in the queue (the buffer) by order of the number of the node or source where they come from. This deterministic ordering rule is for simplification and tends to favor the cells coming from certain sources and nodes. One could order the cells randomly instead, but that would have no major qualitative impact on our results.

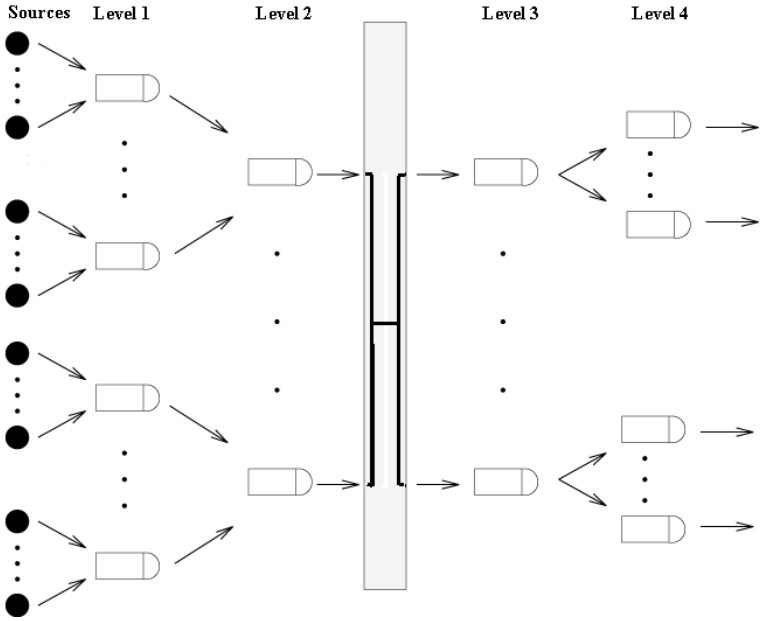

Figure 1: An ATM Switch Modeled as 4 Layers of Queues with Finite Buffer Sizes

All the nodes at level ℓ have the same buffer size B_ℓ and the same constant service time $1/c_\ell$ (so c_ℓ is the service rate). Whenever a cell arrives at a node where the buffer is full, it is *lost* and disappears from the network. Our aim is to estimate the CLR at a given node of the network, say node q^* at level ℓ^*, where the CLR is defined as

$$\mu = \lim_{t \to \infty} \mathrm{E}[N_{\mathrm{L}}(t)]/E[N_{\mathrm{T}}(t)], \tag{2}$$

where $N_{\mathrm{T}}(t)$ is the total number of cells reaching node q^* during the time interval $(0, t]$ and $N_{\mathrm{L}}(t)$ is the number of those cells that are lost due to buffer overflow at node q^*. We assume that the total arrival rate is less than the service rate at each node, so that the network is stable. That is, if the cells from m sources pass through a given node at level ℓ, then $m\rho < c_\ell$, and this holds for all nodes.

To simplify the discussion, we assume that each c_ℓ is an integer. Since the buffers are finite, the model is then a discrete-time Markov chain with finite state space. It is also

aperiodic, and the *zero* state (the state where all sources are *off* and all the nodes are empty) is positive recurrent and is accessible from every other state. As a consequence, there exists a limiting distribution ν over the state space of that chain, defined as

$$\nu(\cdot) = \lim_{n\to\infty} P\{\text{state} \in \cdot \text{ at time } n\}.$$

This model could of course be generalized in several directions and our approach would be easy to adapt for certain types of generalizations. For example, the buffer sizes and constant service times can differ between nodes at a given level, different sources can have different transition probability matrices R, and a source could produce a cell only with some probability when it is *on*. IS would still work nicely in these situations. We keep our simpler model to avoid burying the key ideas under a complicated notation. On the other hand, if the destinations were determined randomly and independently for each cell, or for each *on* period at each source, finding an efficient way of applying IS would be more difficult. Our fixed source-destination assignment model is reasonable because in the ATM switches that we have in mind, a typical connection between a source and a destination lasts for several orders of magnitude longer than the service times $1/c_\ell$.

2 A Regeneration Approach for Confidence Intervals

IS is generally easier to apply to a model defined over a short time horizon or when the model's evolution can be decomposed into short regenerative cycles. Here, the model is over an infinite horizon, and to decompose its trajectory into cycles, we apply a generalization of the classical regenerative method introduced by [Nicola *et al.* 1993; Chang *et al.* 1994], and called the *A-cycle* method. Let A be a subset of the state space of the system. Here we take A as the set of states for which the buffer at q^* is empty. Let $t_0 = 0$ and let $t_1, t_2, \ldots$ be the successive hitting times of the set A; i.e., $t_i = \inf\{t > t_{i-1} : \text{the buffer at } q^* \text{ is empty at time } t \text{ but not at time } t-1\}$. The system state at those hitting times t_i has a pointwise limit distribution π, over A, defined by:

$$\pi(\cdot) = \lim_{i\to\infty} P\{\text{state} \in \cdot \text{ at time } t_i\}.$$

The process over the time interval $(t_{i-1}, t_i]$ is called the ith A-cycle. Let X_i be the number of cells reaching node q^* during the ith A-cycle, and Y_i be the number of those X_i cells that are lost due to buffer overflow at q^*. Let E_π denote the mathematical expectation over an A-cycle when the initial state (at the beginning of the A-cycle) has distribution π. One has:

$$\mu = \frac{\mathrm{E}_\pi[Y_1]}{\mathrm{E}_\pi[X_1]}. \tag{3}$$

In the limit, as the number of A-cycles increases, the *average* distribution of the system states at the times t_i approaches π. By taking the average of the Y_i and X_i over the first n A-cycles, one obtains the consistent estimator of μ:

$$\hat{\mu} = \frac{\sum_{i=1}^n Y_i}{\sum_{i=1}^n X_i}.$$

This estimator is biased unless the initial state at time 0 is generated from π, which is usually much too hard to achieve, but the bias can be reduced by *warming up* the system, e.g., by running $n_0 + n$ A-cycles and discarding the first n_0 from the statistics.

The A-cycles are asymptotically identically distributed (with probability law π for their initial state) but they are *dependent*. To reduce the dependence, and also improve the normality, one can *batch* the cycles, as in the usual batch means method. One then applies the standard methodology for computing a confidence interval for a ratio of expectations, using the batch means as observations, and obtain a confidence interval for μ [Law and Kelton 1991].

3 Applying Importance Sampling

When μ is very small, the vast majority of the Y_i's in (2) are 0 and the *relative error* of $\hat{\mu}$ (i.e., its standard deviation divided by μ) blows up. In (3), the denominator $\mathrm{E}_\pi[X_1]$ is easy to estimate, but the numerator is hard to estimate because it depends on rare events. In fact, denoting $\mu_Y = \mathrm{E}_\pi[Y_1]$ and observing that Y_1 is a non-negative integer, one has $\mathrm{Var}_\pi[Y_1] = \mathrm{E}_\pi[Y_1^2] - \mu_Y^2 \geq \mathrm{E}_\pi[Y_1] - \mu_Y^2 = \mu_Y(1-\mu_Y)$, so the squared relative error satisfies

$$\mathrm{RE}^2[Y_1] = \frac{\mathrm{Var}_\pi[Y_1]}{\mu_Y^2} \geq \frac{1}{\mu_Y} - 1 \to \infty \tag{4}$$

as $\mu_Y \to 0$. Following [Chang *et al.* 1994], we will use IS for the numerator of (3) and not for the denominator.

Let S^* denote the set of sources feeding q^*. The IS strategy for increasing the traffic towards q^* is to increase r_{01} and r_{11} in the matrix R, for all the sources that belong to S^* and only those, so that the total long run arrival rate at q^* becomes larger than the service rate. The system starts with an empty buffer at q^* (a state in A) and the change remains in effect until the buffer at q^* empties again or overflows. When the buffer overflows, R is set back to its original for all the sources until the buffer at q^* empties again, which marks the end of the A-cycle. We call this an *A-cycle with IS*. Under this strategy, if the traffic to q^* can be increased sufficiently, cell losses are no longer rare events. This can certainly be achieved if $m^* > c^*$, where m^* is the cardinality of S^* and $c^* = c_{\ell^*}$ is the service rate at the target node.

It remains to decide how to change R. For a real-valued parameter θ, define

$$\Gamma(\theta) = \begin{pmatrix} r_{00} & r_{01}e^\theta \\ r_{10} & r_{11}e^\theta \end{pmatrix},$$

let $\lambda(\theta)$ be the spectral radius (largest eigenvalue) of $\Gamma(\theta)$, and let $(f_0(\theta), f_1(\theta))$ be the corresponding eigenvector, so that

$$\begin{pmatrix} r_{00} & r_{01}e^\theta \\ r_{10} & r_{11}e^\theta \end{pmatrix} \begin{pmatrix} f_0(\theta) \\ f_1(\theta) \end{pmatrix} \;=\; \lambda(\theta) \begin{pmatrix} f_0(\theta) \\ f_1(\theta) \end{pmatrix}.$$

The eigenvalue $\lambda(\theta)$ can be written explicitly as

$$\lambda(\theta) = \frac{1}{2}\left(r_{00} + r_{11}e^\theta + \sqrt{(r_{00} - r_{11}e^\theta)^2 + 4e^\theta r_{01} r_{10}}\right).$$

For IS, we will change R to the stochastic matrix

$$\tilde{R} = \begin{pmatrix} \tilde{r}_{00} & \tilde{r}_{01} \\ \tilde{r}_{10} & \tilde{r}_{11} \end{pmatrix} = \frac{1}{\lambda(\theta)} \begin{pmatrix} r_{00} & r_{01}e^{\theta} f_1(\theta)/f_0(\theta) \\ r_{10} f_0(\theta)/f_1(\theta) & r_{11}e^{\theta} \end{pmatrix}.$$

This formulation is quite flexible, because the mean arrival rate from a source can be set to an arbitrary value between 0 and 1 by choosing an appropriate θ, and it leads to important simplifications in the likelihood ratio over an A-cycle, as we will see.

During a given A-cycle, let N_{ij} be the number of times a source in S^* goes from state i to state j *while using the probabilities* $\tilde{r}_{ij}$, for $i = 0, 1$ and $j = 0, 1$. The total number of transitions generated from $\tilde{R}$ is then $N_T = N_{00}+N_{01}+N_{10}+N_{11} = m^*t$, where t is the number of time steps where IS is on. The state transitions of the sources are assumed to occur right before the (discrete) times of cell production. The number of cells generated for q^* during the time interval $(0, t]$ is thus $N_{01} + N_{11}$. If the buffer overflows at time t, that number should be approximately equal to the number of cells required to fill up the buffer plus those that are served at q^* during that time period, i.e., approximately $B^* + c^*t$, where B^* is the buffer size at q^*. The difference $\Delta = N_{01} + N_{11} - B^* - c^*t$ can be written as $\Delta = Q_t + L_t - Q_0 - F_t$, where Q_0 and Q_t are the numbers of cells already generated and on their way to node q^* at time 0 and at time t, respectively, L_t is the number of cells headed to q^* but lost due to buffer overflow either at q^* or upstream during $(0, t]$, and F_t is the difference between the total capacity of service c^*t of the server at q^* during $(0, t]$ and the actual number of cells served at q^* during that interval of time. We assume that at the levels upstream of q^*, the increase of traffic when using $\tilde{R}$ instead of R is divided among several nodes and the buffer sizes at these nodes remain almost empty most of the time, whereas the server at q^* is almost always busy, so Q_t, L_t, Q_0, and F_t remain small. This is typical.

The *likelihood ratio* associated with this change of probabilities is

$$\begin{aligned} L &= \left(\frac{r_{00}}{\tilde{r}_{00}}\right)^{N_{00}} \left(\frac{r_{01}}{\tilde{r}_{01}}\right)^{N_{01}} \left(\frac{r_{10}}{\tilde{r}_{10}}\right)^{N_{10}} \left(\frac{r_{11}}{\tilde{r}_{11}}\right)^{N_{11}} \\ &= W(\theta)\lambda(\theta)^{N_T} \exp[-\theta(N_{01} + N_{11})] \\ &= W(\theta) \exp[m^*t \ln \lambda(\theta) - \theta(B^* + c^*t + \Delta)] \end{aligned} \tag{5}$$

where

$$W(\theta) = (f_0(\theta)/f_1(\theta))^{N_{01}-N_{10}}.$$

If V is a random variable defined over an A-cycle with initial state that has distribution π, $\mathrm{E}_\pi[V] = \tilde{\mathrm{E}}_\pi[LV]$, where $\tilde{\mathrm{E}}_\pi$ denotes the expectation under the probabilities $\tilde{R}$, over an A-cycle with IS, with initial state drawn from π. Thus, computing LV over the A-cycle with IS yields an unbiased estimator of $\mathrm{E}_\pi[V]$.

In (5), $|N_{01} - N_{10}|$ in $W(\theta)$ is bounded by m^*, $\exp(-\theta B^*)$ is a constant, and the variance of $\exp(-\theta\Delta)$ is expected to remain under control even for large t. An annoying term that remains is $\exp[t(m^* \ln \lambda(\theta) - \theta c^*)]$, and our strategy is to simply kill it by choosing $\theta = \theta^* > 0$ such that

$$m^* \ln \lambda(\theta^*) = \theta^* c. \tag{6}$$

Note that $\ln \lambda(0) = 0$, $\ln \lambda(\theta)/\theta$ is strictly increasing and differentiable (see, e.g., [Chang *et al.* 1994], Example 2.6), and $\ln \lambda(\theta)/\theta \to 1$ as $\theta \to \infty$. Therefore, this θ^* exists if and only if $m^* > c^*$, which we assume (otherwise, one cannot overload the node q^*). With $\theta = \theta^*$, the likelihood ratio becomes

$$L \;=\; e^{-\theta^*(B^*+\Delta)} W(\theta^*).$$

The variance of the estimator of μ_Y is $\tilde{\mathrm{Var}}_\pi[LY_1] = \tilde{\mathrm{E}}_\pi[L^2 Y_1^2] - \mu_Y^2$ and one has

$$\tilde{\mathrm{E}}_\pi[L^2 Y_1^2] \;=\; e^{-\theta^* B^*} \tilde{\mathrm{E}}_\pi[L Y_1 e^{-\theta^* \Delta} Y_1 W(\theta^*)]. \tag{7}$$

We pursue with *heuristic arguments*. A first observation is that in most cases of interest, $f_0(\theta^*)/f_1(\theta^*) < 1$, in which case $W(\theta^*)$ is almost always less than 1 and usually much smaller than 1. As a second observation, since q^* is stable without IS and since IS is stopped as soon as the buffer overflows, Y_1 should remain "reasonable". Thirdly, by looking at the definition of Δ, the reader would agree that Δ should usually be positive and almost never take large negative values. Moreover, Δ should usually be larger (positive) when Y_1 is larger, because a large Y_1 is much more likely when $Q_t - Q_0$ is large. Therefore, $e^{-\theta^* \Delta} Y_1 W(\theta^*)$ in (7) is expected to remain small. These arguments, together with (7), lead to the *very crude approximation*

$$\tilde{\mathrm{E}}_\pi[L^2 Y_1^2] \;=\; O(e^{-\theta^* B^*} \mu_Y). \tag{8}$$

If (8) holds, then IS provides the approximate variance reduction factor

$$\frac{\tilde{\mathrm{Var}}_\pi[LY_1]}{\mathrm{Var}_\pi[Y_1]} \le \frac{\tilde{\mathrm{E}}_\pi[L^2 Y_1^2] - \mu_Y^2}{\mu_Y - \mu_Y^2} = O(e^{-\theta^* B^*}).$$

Independently of (8), the squared relative error of the IS estimator satisfies

$$\tilde{\mathrm{RE}}^2[LY_1] \;=\; \frac{\tilde{\mathrm{Var}}_\pi[LY_1]}{\mu_Y^2} \;\le\; \frac{\tilde{\mathrm{E}}_\pi[L^2 Y_1^2]}{\mu_Y^2} \;\le\; \frac{\tilde{\mathrm{E}}_\pi[(e^{-\theta^* \Delta} Y_1 W(\theta^*))^2]}{\tilde{\mathrm{E}}_\pi^2[e^{-\theta^* \Delta} Y_1 W(\theta^*)]} \tag{9}$$

The ratio of expectations in (9) is ≥ 1 (by the Cauchy-Schwartz inequality) and should remain under control when B^* increases. Bounding this ratio by a constant independent of B^* would prove that the relative error under IS is bounded, but we do not have the proof. One may be tempted to modify the IS scheme adaptively (e.g., by stopping IS earlier or later) in order to reduce the variability of the quantity $e^{-\theta^* \Delta} Y_1 W(\theta^*)$. We will return to this in Section 5.

What about the variance of the variance estimator, with and without IS? They can be compared by comparing $\tilde{\mathrm{E}}_\pi[L^4 Y_1^4]$ with $\mathrm{E}_\pi[Y_1^4]$. Using the same heuristic argument as in (8) above, one obtain the crude approximation

$$\frac{\tilde{\mathrm{E}}_\pi[L^4 Y_1^4]}{\mathrm{E}_\pi[Y_1^4]} \;=\; \frac{\tilde{\mathrm{E}}_\pi[L^4 Y_1^4]}{\tilde{\mathrm{E}}_\pi[L Y_1^4]} \;=\; O(e^{-3\theta^* B^*}).$$

Not only the estimator itself is less noisy with IS than without, its sample variance is also much less noisy, and by a larger factor.

We now explain how the A-cycles are simulated to estimate both the numerator and the denominator in (3), in the IS case. One simulates two *versions* of each A-cycle, one with IS and the other without, both starting from the same initial state. Thus, the A-cycles come in pairs. For the ith A-cycle pair, one first simulates an A-cycle with IS, which provides an estimation $L_i Y_i$ of the numerator, where L_i and Y_i are the value of the L and the number of cell losses for this cycle. Then, the state of the system is reset to what it was at the beginning of this A-cycle with IS, and a second A-cycle is simulated to obtain an estimator X_i of the denominator. The final state of the no-IS A-cycle, which obeys approximately the distribution π, is then saved and is taken as the initial state for the next pair of A-cycles. After a warmup of n_0 cycles without IS, n pairs of A-cycles are thus simulated and the IS estimator of μ is

$$\hat{\mu} = \frac{\sum_{i=1}^{n} L_i Y_i}{\sum_{i=1}^{n} X_i}.$$

A confidence interval is computed using batch means as explained in Section 2.

Starting the two A-cycles of each pair from the same state means that one must save or reset the entire state of the system after each cycle. This means copying how many cells are at each node of the network, the destinations of these cells, and the state (*on* or *off*) of each source. One can also memorize/reset the state of each random number generator, so that the two A-cycles of a pair use common random numbers. This tends to increase the correlation between $L_i Y_i$ and X_i, and to decrease the variance of $\hat{\mu}$ as a result.

4 Simulation Experiments

4.1 The Setup

For several examples and parameter sets, we ran the simulation first using the standard approach without IS, for C A-cycles, and then with IS for C' pairs of A-cycles. In each case, the values of C and C' were chosen so that the total CPU time was about the same for both IS and no-IS, and these A-cycles were regrouped into $b = 200$ batches. (For sensitivity analysis with respect to b, we tried different values of b ranging from 50 to 3200, for several examples, and found that the variance estimates were practically independent of b, in that range, for the values of C and C' that we use). For $1 \le j \le b$, let $\bar{X}_j$ and $\bar{Y}_j$ denote the samples means of the X_i and Y_i (or of the X_i and $L_i Y_i$, for IS), respectively, within batch j. The tables that follow report the value of the CLR estimator $\hat{\mu}$ and of its variance estimator

$$\hat{\sigma}^2 = (S_Y^2 + \hat{\mu}^2 S_X^2 - 2\hat{\mu} S_{XY})/(b\bar{X}^2), \tag{10}$$

where $\hat{\mu} = \bar{Y}/\bar{X}$, and $\bar{Y}$, $\bar{X}$, S_Y^2, S_X^2, and S_{XY} are the sample means, sample variances, and sample covariance of the $\bar{Y}_j$ and $\bar{X}_j$, respectively. The tables also report the relative half-width $\hat{\delta} = 2.57\hat{\sigma}/\hat{\mu}$ of a 99% confidence interval on μ (under the normality assumption), the CPU time t (in seconds) required to perform the simulation, and the *relative efficiency* (eff.), defined as $\hat{\mu}^2/(t\hat{\sigma}^2)$. These values are all noisy estimates but give a good indication of what happens.

For the cases where no cell loss was observed in all the A-cycles simulated, we put $\hat{\mu} = 0$ and the entries for the variance and efficiency are left blank. The simulation with IS takes more CPU time than no-IS for the same total number of simulated cells, but the relative efficiency takes both the variance reduction and the overhead into account. Beware: Efficiencies and CPU times can be compared within a given table, but not across the tables, because the models are different and the experiments were run on different machines (SUN SparcStations 4, 5, and 20). Within each table, common random numbers were used for the corresponding A-cycles across the different lines of the table.

4.2 CLR Estimation at Level 2

EXAMPLE **1** Let $\ell^* = 2$, $B_1 = 512$, $m_0 = 2$, $m_1 = 25$, $c_1 = 1$, $c_2 = 3$, $\kappa_1 = 50$, $\rho = 1/101$ (i.e., $r_{11} = 49/50 = 0.9800$ and $r_{00} = 0.9998$) and vary the buffer size $B^* = B_2$. There are 50 sources feeding the target node q^*, so the average arrival rate at q^* is $50/101 \approx 0.495$, while the service rate is 3. With these numbers, we compute $\theta^* = 0.018127$, $f_0(\theta^*) = 0.0581$, $f_1(\theta^*) = 0.3676$, $\tilde{r}_{11} = 0.99684$, $\tilde{r}_{00} = 0.99871$, and IS increases the total arrival rate at q^* from 0.495 to 14.48.

We took $C = 7\,200\,000$ for no-IS and $C' = 300\,000$ for IS (note that the IS cycles are much longer than the no-IS on the average, and their average length increases with B^*, because most of them fill up the buffer before emptying it again, whereas for most of the no-IS cycles the buffer empties after just a few cell arrivals). Table 1 gives the results. For $B_2 \geq 512$, without IS, not a single cell loss was observed, so the estimates are useless. On the other hand, the relative error of the IS estimators does not increase significantly as a function of B_2, and these estimators work nicely to estimate very small CLRs. The efficiency decreases slowly with B_2. (The outlier at $B_2 = 768$ will be discussed later on.)

EXAMPLE **2** Same as the preceding example, except that B_2 is now fixed at 512 and we vary the average burst size κ_1. For large κ_1, μ is large and easy to estimate, but not for small κ_1 (the other parameters remaining the same). The results are in Table 2. Without IS, cell losses were observed only for $\kappa_1 \geq 100$, and even in that case IS is much more efficient. The total arrival rate with IS decreases with κ_1: It goes from 22.5 for $\kappa_1 = 25$ to 5.95 with $\kappa_1 = 150$. The squared relative error with IS (not show in the table) is approximately constant as a function of κ_1.

An important question now arises: How noisy are the variance and efficiency estimates given in the tables? One way of estimating the distribution of the variance and efficiency estimators is to bootstrap from the b batch means, as follows. Put the b pairs $(\bar{X}_1, \bar{Y}_1), \ldots, (\bar{X}_b, \bar{Y}_b)$ in a table. Draw b random pairs from that table, with replacement, and compute the quantities $\hat{\sigma}^2$ and eff. that correspond to this sample of size b. Repeat this N times and compute the empirical *distributions* of the N values of $\hat{\sigma}^2$ and of eff. thus obtained. These empirical distributions are bootstrap estimators of the distributions of $\hat{\sigma}^2$ and eff., and the interval between the 2.5th and 97.5th percentiles of the empirical distribution is a 95% bootstrap confidence interval for the variance of $\hat{\mu}$ or for the efficiency. Table 3 gives the xth percentiles Q_x of the

Table 1: CLR estimation at level 2 for different buffer sizes

B_2	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
	no-IS				
128	2.8E-5	2.5E-11	45%	2828	0.0113
256	6.8E-7	4.6E-13	257%	2828	0.0003
512	0			2828	
768	0			2829	
1024	0			2827	
	IS				
128	3.0E-5	6.3E-13	7%	1675	0.838
256	9.8E-7	1.5E-15	10%	1993	0.315
512	2.5E-9	5.4E-20	24%	2593	0.043
768	3.7E-11	5.9E-22	170%	3108	0.001
1024	5.6E-14	3.6E-29	28%	3634	0.023

Table 2: CLR estimation at level 2 for different average burst sizes

κ_1	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
	no-IS				
10	0			1969	
25	0			2245	
50	0			2828	
100	8.1E-5	1.0E-10	32%	4591	0.014
150	3.0E-3	2.9E-9	5%	7641	0.414
	IS				
10	1.1E-39	1.5E-79	87%	2897	0.003
25	2.1E-17	8.6E-36	36%	2884	0.017
50	2.5E-9	5.4E-20	24%	2593	0.043
100	7.2E-5	3.2E-12	6%	2445	0.659
150	3.0E-3	3.9E-9	5%	2585	0.905

bootstrap distributions obtained from the results of Example 1, for $x = 2.5$, 50, and 97.5, with $N = 10\,000$.

We already pointed out the very low empirical efficiency of the IS estimator with $B_2 = 768$ in Table 1. A closer look at the 200 batch means $\bar{Y}_j$ reveals that one of the $\bar{Y}_j$ in that case is 4.19×10^{-8}, whereas all others are less than 10^{-9}, except one which is 1.92×10^{-9}. It seems that a rare event has happened within that particular batch. We did not observe such outliers for the other values of B_2, but we found some in other examples, although rarely as excessive. The presence of these outliers is due to the important tail which remains in the distribution of $\bar{Y}_j$ after IS, despite the large reduction in the variance of $\bar{Y}_j$. (It would have been easy to change the example in the paper for one that gives no outlier. Of course, this would have been misleading. And the current example, with the outlier, is instructive.) This outlier has an important effect not only on the variance and efficiency estimators, but also on the bootstrap distributions, as can be seen from Table 3 (compare the behavior of the quantiles for $B_2 = 768$ with those for the other values of B_2). To assess the effect of the outlier, we repeated the bootstrap after removing it from

Table 3: Bootstrap quantile estimates for Example 1

	$\hat{\sigma}^2$			eff.		
B_2	$Q_{2.5}$	Q_{50}	$Q_{97.5}$	$Q_{2.5}$	Q_{50}	$Q_{97.5}$
128	4.3E-13	6.2E-13	8.8E-13	0.23	0.31	0.42
256	8.1E-16	1.5E-15	2.4E-15	0.08	0.12	0.19
512	6.2E-21	5.4E-20	1.5E-19	7.5E-3	1.6E-2	1.0E-1
768	3.6E-25	5.9E-22	1.8E-21	2.4E-4	3.7E-4	3.7E-3
1 024	9.9E-30	3.4E-29	8.1E-29	5.1E-3	8.7E-3	2.2E-2

the sample (i.e., with the 199 remaining pairs), and obtained the following quantiles for $\hat{\sigma}^2$: $(Q_{2.5}, Q_{50}, Q_{97.5}) = (2.7 \times 10^{-25}, 1.7 \times 10^{-24}, 4.5 \times 10^{-24})$. The effect is significant. The numbers suggest that for $B_2 = 768$, the variance is highly overestimated, that the efficiency is underestimated, and that the bootstrap distribution is more widely spread than the true distribution. To confirm these suspicions, we made 5 additional replications of the entire experiment, independently, with $B_2 = 768$ and IS. The results, in Table 4, give an idea of the variability. Table 5 provides similar results for $B_2 = 512$. One can see that the efficiency estimator is (unfortunately) noisy. On the other hand, $\hat{\mu}$ is (fortunately) much less noisy, and this is reassuring.

Table 4: Five additional independent replications for $B_2 = 768$ with IS

$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
1.2E-11	1.1E-24	22%	3132	0.044
1.2E-11	2.1E-24	30%	3100	0.023
1.1E-11	8.0E-25	21%	3108	0.048
1.1E-11	1.8E-24	31%	3119	0.022
9.5E-12	2.9E-25	14%	3100	0.101

Table 5: Five additional independent replications for $B_2 = 512$ with IS

$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
3.1E-9	1.8E-19	35%	2592	0.020
2.5E-9	2.0E-20	15%	2587	0.115
3.8E-9	2.0E-18	94%	2591	0.003
3.3E-9	3.5E-19	46%	2600	0.012
2.4E-9	1.0E-20	11%	2588	0.223

4.3 CLR Estimation at Level 3

EXAMPLE **3** Let $\ell^* = 3$, $B_1 = B_2 = 512$, $c_1 = 1$, $c_2 = c_3 = 2$, $m_0 = 2$, $m_1 = 3$, $m_2 = 10$, $\kappa_1 = 50$, $\rho = 1/21$, and we vary the buffer size $B^* = B_3$. We assign 6 of the 60 sources to q^*. One node at level 2 is fed by 2 of these 6 *hot* sources, while no other node at levels 1 and 2 is fed by more than 1 of them. Here, $\theta^* = 0.027287$, $f_0(\theta^*) = 0.0851186$, $f_1(\theta^*) = 0.839641$, and the total arrival rate at q^* is $6/21$ without IS and 5.0 with IS. We take $C = 1\,800\,000$ and

$C' = 100\,000$. The results appear in Table 6. Again, IS works nicely while the no-IS observes no cell loss except at the smallest buffer size. With IS, the relative error and the relative efficiency are almost constant with respect to B^*.

EXAMPLE **4** Same as the preceding example, except that B_3 is fixed at 256 and we vary the average burst size κ_1. Table 7 gives the results. While no-IS has difficulty to observe cell losses, IS gives reasonable estimations.

Table 6: CLR estimation at level 3 for different buffer sizes

B_3	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
			no-IS		
128	2.4E-5	1.1E-10	112%	7036	0.002
256	0			7024	
512	0			7059	
768	0			7037	
1024	0			7027	
			IS		
128	4.1E-5	5.3E-12	14%	5779	0.056
256	6.0E-7	7.2E-16	11%	7316	0.069
512	3.4E-10	2.9E-22	13%	10246	0.040
768	2.5E-13	1.7E-28	13%	12930	0.029
1024	2.1E-16	1.4E-34	14%	15649	0.020

Table 7: CLR estimation at level 3 for different average burst sizes

κ_1	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
			no-IS		
10	0			1088	
25	0			1050	
50	0			1125	
100	2.1E-5	2.1E-10	178%	1134	0.007
150	1.1E-4	1.6E-9	94%	1136	0.015
			IS		
10	5.4E-20	8.6E-42	14%	1454	0.24
25	1.3E-10	2.8E-23	11%	1213	0.49
50	6.0E-7	7.2E-16	11%	1042	0.48
100	4.1E-5	6.7E-12	16%	881	0.28
150	1.7E-4	9.1E-11	15%	813	0.38

4.4 CLR Estimation at Level 4

EXAMPLE **5** Let $\ell^* = 4$, $B_1 = B_2 = B_3 = 512$, $c_1 = c_4 = 1$, $c_2 = c_3 = 4$, $m_0 = 5$, $m_1 = 10$, $m_2 = 6$, $\kappa_1 = 50$, $\rho = 1/41$, and we vary the buffer size $B^* = B_4$. We assign 6 of the 300 sources to q^*. They are distributed as in Example 3. Here, $\theta^* = 0.021218$, $f_0(\theta^*) = 0.0813754$, $f_1(\theta^*) = 0.644124$, and the total arrival rate at q^* is $6/41$ without IS and 3.692 with IS. We

take $C = 800\,000$ and $C' = 50\,000$. The results are in Table 8 and they resemble what was observed at level 3. For this example, we also varied κ_1 with B_2 fixed at 512, and the results were qualitatively similar to those of Table 7.

Table 8: CLR estimation at level 4 for different buffer sizes

B_4	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
			no-IS		
128	1.6E-3	7.5E-8	44%	3580	0.004
256	8.3E-6	6.8E-11	255%	3593	0.013
512	0			3586	
768	0			3592	
1024	0			3595	
			IS		
128	1.1E-3	4.0E-9	15%	1881	0.15
256	5.5E-5	1.4E-10	55%	2440	0.008
512	1.4E-7	4.4E-16	39%	3580	0.012
768	3.6E-10	2.7E-21	36%	4488	0.011
1024	1.0E-12	1.5E-26	31%	5550	0.012

4.5 Other Variants of the Model

We made several experiments with variants of the model to explore the effectiveness of the proposed IS strategy in other (sometimes more realistic) situations.

The original model is called *variant A*. For *variant B*, the sources are no longer affected to fixed destinations, but the destination of each cell is chosen randomly, independently of other cells, uniformly over all destinations. *Variant C* is similar except that each *burst* (i.e., all the cells from a source during a given *on* period) has a random destination. The IS approach of Section 3 did very badly for variant B, and gave improvement for variant C only when μ was very small. An appropriate IS strategy for these models should also change the probabilities over the destinations to increase the traffic towards q^*. In any case, variants B and C are not realistic for ATM switches.

In variant D, each node at level 3 has k buffers, the first one receiving the cells originating from the sources 1 to $m_0m_1m_2/k$, the second one taking those from the sources $1 + m_0m_1m_2/k$ to $2m_0m_1m_2/k$, and so on. A server at level 3 takes cells from those buffers according to either a *round robin* or *longest queue first* policy.

In variant E, the sources produce two classes of cells: High priority *constant bit rate* (CBR) cells and low priority *variable bit rate* (VBR) cells. The VBR sources are Markov modulated as before, whereas the CBR sources have constant *on* and *off* periods (they are completely deterministic). Each node has two buffers, one for the CBR cells and one for the VBR cells, and the CBR cells are always served before the VBR ones.

The IS strategy of Section 3 works fine for the variants D and E: It provides reasonable estimates for values of μ that standard simulation cannot handle. We also observed in our empirical results that the longest queue first policy gives a CLR generally smaller than round robin.

5 Refining the Importance Sampling Scheme

5.1 Optimizing θ

The IS approach of Section 3 provides a good change of measure, but based only on a heuristic and asymptotic argument, not necessarily the optimal value of θ for a given buffer size. Moreover, when choosing θ, the approach does not take into account the computational costs which may depend on θ. To evaluate the sensitivity with respect to θ, we performed additional experiments where θ was varied around θ^*, and the variance and efficiency were estimated. As a general rule, we found that the optimal θ was around 20% to 25% less than θ^*, and increased the efficiency by a factor between 2 to 15 compared with θ^*, at level 2 or 3 where m^* is typically large. At level 1 or 4, where m^* is usually small, the optimal θ tends to be much closer to (and no significantly better than) θ^*. We emphasize that there is noise in these estimated factors, due to the variance of the efficiency estimators. However, the tendency persisted when we replicated the experiments. Such factors constitute significant efficiency improvements, so it would make sense to use, e.g., $\theta = (4/5)\theta^*$ instead of θ^* at levels 2 and 3, and perhaps try to optimize θ adaptively in a small neighborhood around that value, during the simulation. It is very dangerous to use $\theta > \theta^*$, because the variance increases very fast with θ in that area, and may even become infinite for finite θ. The next examples illustrate typical behavior at levels 3 and 4.

EXAMPLE **6** Let $\ell^* = 3$, $B_1 = B_2 = B_3 = 256$, $m_0 = 2$, $m_1 = 3$, $m_2 = 10$, $c_1 = 1$, $c_2 = c_3 = 2$, $\kappa_1 = 50$, and $\rho = 1/21$. The node q^* is fed by 6 sources, whose traffic passes through as in example 3. We take $C = 1\,920\,000$. Here, $\theta^* = 0.0272$, and the results for different values of θ around θ^* are in Table 9. Taking $\theta = 0.0185$ improves the empirical efficiency by a factor of approximately 25 compared with θ^*. By examining the data more closely, we found that the efficiency improves because the smaller θ gives a smaller value of $S_Y^2/(b\bar{X}^2)$, which is the dominant term in $\hat{\sigma}^2$. Further replications showed similar results, with $\theta = 0.0185$ registering efficiencies 15 to 60 times higher than θ^*.

EXAMPLE **7** Let $\ell^* = 4$, $B_1 = B_2 = B_3 = B_4 = 512$, $m_0 = 2$, $m_1 = m_2 = 5$, $c_1 = c_4 = 1$, $c_2 = c_3 = 5$, $\rho = 1/41$, and $\kappa_1 = 50$. Only 2 sources feed the node q^*. Both sources feed the same node at level 3, but different nodes at levels 1 and 2. We take $C = 120\,000$. In this case, $\theta^* = 0.0394$, and the results for different values of θ are given in Table 10. In this case, taking $\theta < \theta^*$ brings no significant efficiency improvement. This was confirmed by 4 additional independent replications of this entire experiment. We made similar experiments with exactly the same data as in Example 5, with $B_4 = 256$, and observed an efficiency improvement by a factor between 1.5 and 2.

5.2 Defining the A-Cycles Differently

Instead of starting the A-cycles when the buffer at q^* becomes empty, one can start them when the number of cells in the buffer crosses β upward, where β is a fixed integer. There

Table 9: Comparing different values of θ, for $\ell^* = 3$

θ	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
0.0170	6.3E-7	9.3E-17	3.9 %	16672	0.258
0.0185	6.4E-7	8.0E-17	3.6 %	16744	0.304
0.0200	6.4E-7	1.4E-16	4.8 %	16321	0.176
0.0215	6.2E-7	1.3E-16	4.7 %	15088	0.201
0.0230	6.2E-7	3.4E-16	7.7 %	14307	0.079
0.0245	6.2E-7	1.4E-15	15.5 %	13701	0.020
0.0260	5.7E-7	2.5E-15	22.4 %	15478	0.008
$\rightarrow$ 0.0272	5.2E-7	1.7E-15	20.4 %	12945	0.012
0.0290	5.2E-7	5.1E-15	35.2 %	12667	0.004

is essentially nothing to gain in that direction, however, because when increasing β the no-IS A-cycles tend to become excessively long (typically, the buffer at q^* remains nearly empty most of the time).

Another idea is to impose a lower bound, say t_0, on the length of the A-cycles, to get rid of the extremely short (and wasteful) A-cycles which tend to occur frequently under both the IS and no-IS setup. The A-cycle ends at the maximum time between t_0 and the first time when node q^* becomes empty. How to choose t_0? We want to choose it large enough to make sure that most A-cycles under IS see some overflow, but not too large, so that the A-cycles end at the first return to the empty state after overflow. According to our arguments in Section 3, if overflow occurs at time t_1, then the total production by the twisted sources up to time t_1 should be approximately equal to the number of cells required to keep the server busy until time t_1 and fill up the buffer at node q^*, that is, $m^*\tilde{\rho}t_1 \approx B^* + c^*t_1$, where $\tilde{\rho}$ is the average production rate of a twisted source. The additional time t_2 to empty the buffer (with IS turned off) should satisfy $(c^* - m^*\rho)t_2 \approx B^*$. We want (roughly) $t_0 \leq t_1 + t_2$, i.e.,

$$t_0 \leq \frac{B^*}{m^*\tilde{\rho} - c^*} + \frac{B^*}{c^* - m^*\rho}.$$

We suggest taking t_0 somewhere between 20% and 50% of the value of that upper bound. In our experiments, this always gave efficiency improvement. Since the variance associated with the IS cycles is the dominant term in the variance of $\hat{\mu}$, a good strategy is to choose t_0 just large enough so that most of the IS cycles fill up the buffer. Taking t_0 too large (close to $t_1 + t_2$) is not a good idea because it makes us spend too much time on the no-IS cycles without bringing much additional variance reduction. Beyond a certain point, increasing t_0 eventually *decreases* the efficiency.

EXAMPLE **8** We used the same data as in Example 5 (for $\ell^* = 4$), with $B_4 = 256$ and $C' = 160\,000$, with IS. For $\theta = \theta^*$, we have $t_1 \approx 95$ and $t_2 \approx 300$. For $\theta = 0.80\,\theta^*$, we have a total arrival rate of 1.82 with IS, which give $t_1 \approx 312$ and $t_2 \approx 300$. Table 11 give the results.

Table 10: Comparing different values of θ, for $\ell^* = 4$

θ/θ^*	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
1.00	1.73E-11	7.5E-26	4.1 %	5840	0.68
0.95	1.74E-11	1.2E-25	5.1 %	5810	0.44
0.90	1.74E-11	1.0E-25	4.7 %	5764	0.52
0.85	1.74E-11	9.8E-26	4.6 %	5681	0.54
0.80	1.72E-11	7.2E-26	4.0 %	5569	0.74
0.75	1.73E-11	9.4E-26	4.6 %	5398	0.59
0.70	1.73E-11	8.6E-26	4.4 %	5134	0.67
0.65	1.76E-11	1.2E-25	5.0 %	4767	0.56
0.60	1.76E-11	1.6E-25	5.8 %	4246	0.47
0.55	1.77E-11	2.5E-25	7.3 %	3470	0.36
0.50	1.65E-11	7.8E-25	14 %	2392	0.15

With $\theta = \theta^*$, raising t_0 from 0 to 75 increases the (empirical) efficiency approximately by a factor of 4. With $\theta = 0.8\,\theta^*$, raising t_0 from 0 to 150 improves the (empirical) efficiency by a factor of more than 10. This gain is related to the rapid increase of $\bar{X}$, which decreases $\hat{\sigma}^2$ (see Eq. (10)), when t_0 is small. We made 2 additional replications of this experiments and the results were similar (although the empirical efficiency for $\theta = \theta^*$ and $t_0 = 0$ was 0.02 and 0.04, which suggests that the factor of efficiency improvement from this setup to $\theta = 0.8\,\theta^*$ and $t_0 = 150$ is more around 20 to 30 instead of 10).

EXAMPLE **9** Let $\ell^* = 3$, $B_1 = B_2 = B_3 = 256$, $m_0 = 2$, $m_1 = 3$, $m_2 = 10$, $c_1 = 1$, $c_2 = c_3 = 2$, $\rho = 1/21$ and $\kappa_1 = 50$. Six sources feed the target node q^*, as in example 3, which gives an average arrival rate of $6/21 \approx 0.286$ to that node. We run simulations for different values of t_0 both with the $\tilde{r}_{ij}$ associated to $\theta^* = 2.73 \times 10^{-2}$ (with $\tilde{\rho} = 5/6$, a total arrival rate of 5.00, $t_1 \approx 85$, and $t_2 \approx 150$) and $0.80\,\theta^* = 2.18 \times 10^{-2}$ (with $\tilde{\rho} \approx 0.54$, a total arrival rate of 3.26, $t_1 \approx 200$, and $t_2 \approx 150$). The results are in Table 12. Using $t_0 = 100$ together with $\theta = 0.8\theta^*$ gives the best empirical efficiency in this case, about 20 times the empirical efficiency observed with $t_0 = 0$ and $\theta = \theta^*$.

5.3 Stopping IS Earlier

Suppose that $\ell^* = 4$ and that we use IS. When the target buffer at q^* overflows and IS is turned off, there may be several cells already in the network at previous levels, and this may produce more overflow than necessary. Because of that, it could make sense to turn off IS earlier, e.g., when the total number of cells in buffer q^* or at previous nodes but on their way to q^*, reaches some threshold N_0. [Beck *et al.* 1998; Dabrowski *et al.* 1998] use this criterion for turning off IS, with $N_0 = B^*$. Our experiments with this idea showed no

Table 11: Imposing Lower Bounds on the A-cycle lengths, for $\ell^* = 4$

t_0	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	$\bar{X}$	$\mathrm{Var}(\bar{X})$	CPU	eff.
			$\theta = \theta^*$				
0	4.2E-5	4.2E-11	39.2 %	0.2	0.03	404	0.10
25	5.1E-5	2.9E-12	8.6 %	4.3	0.66	4328	0.20
50	4.8E-5	8.8E-13	5.0 %	8.1	1.25	7089	0.37
75	4.9E-5	5.1E-13	3.7 %	11.7	1.76	9097	0.52
100	4.8E-5	5.0E-13	3.8 %	15.8	2.43	10523	0.43
150	5.0E-5	6.7E-13	4.2 %	23.1	2.90	12501	0.29
200	4.9E-5	7.6E-13	4.5 %	30.7	3.25	13890	0.23
250	5.0E-5	3.0E-12	8.9 %	38.0	5.45	14933	0.06
			$\theta = 0.8\,\theta^*$				
t_0	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	$\bar{X}$	$\mathrm{Var}(\bar{X})$	CPU	eff.
0	5.1E-5	8.2E-11	45.3 %	0.2	0.03	399	0.08
25	4.7E-5	1.6E-12	6.9 %	4.3	0.66	2641	0.52
50	4.8E-5	6.6E-13	4.4 %	8.1	1.24	4646	0.75
75	4.9E-5	4.5E-13	3.5 %	11.7	1.76	6397	0.83
100	4.8E-5	3.0E-13	2.9 %	15.7	2.42	7886	0.98
150	4.8E-5	1.9E-13	2.4 %	23.0	2.90	10468	1.13
200	4.8E-5	2.5E-13	2.6 %	30.7	3.24	12645	0.75
250	4.8E-5	2.2E-13	2.5 %	37.9	5.45	14439	0.73

significant improvement compared with the method which turns off IS when q^* overflows. With $N_0 < B^*$, this idea seems to *reduce* the efficiency instead. Here is a typical illustration.

EXAMPLE **10** Let $\ell^* = 4$, $B_1 = B_2 = B_3 = B_4 = 512$, $m_0 = 2$, $m_1 = 5$, $m_2 = 5$, $c_1 = c_4 = 1$, $c_2 = c_3 = 5$, $\rho = 1/5$, and $\kappa_1 = 50$. Two sources feed the node q^*, which gives an arrival rate at q^* of $2/5 = 0.4$. When IS is applied the arrival rate increases to 1.5887. These 2 hot sources feed different nodes at level 2. In Table 13, CL is the average number of cell losses per cycle with IS and $N_0 = \infty$ corresponds to turning off IS when q^* overflows. Taking N_0 between 520 and 600 appears to be about as good as our usual method, but $N_0 \leq 510$ is definitely worse.

5.4 Retroactive Manipulations to Control the Overflow

The criterion for turning off IS earlier, considered in the previous subsection, is rather blind. Remember that all the randomness in our model is in the state transitions of the sources. It is therefore possible, in principle, to compute at any given point t in time whether or not there will be overflow at q^* caused only by the cells generated so far, and turn off IS as soon as this happens. In this way, IS is turned off *before* the target buffer fills up, but only when overflow is

Table 12: Imposing Lower Bounds on the A-cycle lengths, for $\ell^* = 3$

t_0	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	CPU	eff.
			θ^*		
0	8.6E-7	3.4E-14	55.5 %	451	0.048
25	6.4E-7	1.1E-15	13.3 %	6821	0.054
50	9.2E-7	9.2E-14	84.1 %	8908	0.001
75	6.7E-7	6.1E-16	9.5 %	9823	0.075
100	6.9E-7	2.9E-15	20.0 %	10409	0.016
150	6.9E-7	1.5E-15	14.6 %	11781	0.026
200	6.3E-7	4.3E-15	26.9 %	12118	0.007
250	5.2E-7	1.2E-15	17.0 %	12880	0.018
			$0.8\,\theta^*$		
0	5.5E-7	1.7E-15	19.2 %	455	0.40
25	6.5E-7	1.3E-16	4.5 %	5355	0.60
50	6.4E-7	5.4E-17	3.0 %	8298	0.90
75	6.5E-7	5.3E-17	2.9 %	10209	0.78
100	6.4E-7	3.3E-17	2.3 %	11531	1.07
150	6.5E-7	5.0E-17	2.8 %	13387	0.63
200	6.4E-7	6.1E-17	3.1 %	14784	0.46
250	6.6E-7	1.0E-16	3.9 %	15938	0.27

guaranteed to occur. In practice, this can be implemented by actually running the simulation until there is overflow, and then turning off IS *retroactively* right after the time t when all the cells having reached q^* when the first cell overflows (at time $t+\epsilon$, say) were already produced by a source. This is complicated to implement and implies significant overhead. Despite spending a lot of time on experimenting with this idea, we were unsuccessful in improving the efficiency with it.

5.5 Combining IS with Indirect Estimation

[Srikant and Whitt 1997] proposed the following indirect estimator of the CLR. (This approach was presented by Ward Whitt during the keynote address of the 1997 Winter Simulation Conference.) The CLR at node q^* satisfies

$$\mu = 1 - \lambda/\lambda_0 = 1 - Lc^*/\lambda_0 \tag{11}$$

where $\lambda_0 = m_0\kappa_1/(\kappa_0+\kappa_1)$ is the total (average) production rate of the m_0 sources feeding node q^*, λ is the (average) output rate from node q^*, $1/c^*$ is the service time at node q^*, and L is the steady-state fraction of time where the server is busy at node q^*. The second equality follows from the Lindley equation $L = \lambda/c^*$. Using (11), μ can be estimated indirectly by estimating L. [Srikant and Whitt 1997] showed that the indirect estimator brings substantial

Table 13: Different stopping criteria for IS

N_0	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$	$\bar{Y}$	$\mathrm{Var}(\bar{Y})$	CPU	eff. $(\times 10^{-3})$
500	4.72E-9	6.7E-18	19.8%	1.51	0.188	3308	1.1
510	4.86E-9	1.5E-17	29.2%	1.97	0.148	3354	0.5
520	4.36E-9	1.8E-18	11.3%	2.77	0.322	3397	4.0
530	4.56E-9	2.0E-18	11.1%	3.55	0.340	3433	3.7
540	4.60E-9	2.1E-18	11.3%	4.37	0.515	3508	3.5
550	4.62E-9	1.3E-18	8.9%	5.15	0.469	3539	5.4
560	4.68E-9	1.4E-18	9.3%	6.24	0.610	3659	4.8
570	4.76E-9	1.5E-18	9.4%	7.42	0.854	3699	4.5
580	4.76E-9	1.4E-18	8.9%	8.36	0.935	3603	5.1
590	4.66E-9	1.3E-18	8.8%	9.34	1.235	3667	5.3
∞	4.71E-9	1.4E-18	9.1%	29.5	3.947	3959	4.5

variance reduction in heavy traffic situations, especially for queues with several servers and random service times, but not in light traffic. In our context, the traffic at q^* is light, but becomes heavy when IS is applied, so it was not clear to us *a priori* if the indirect estimator combined with IS could help.

The results of our extensive numerical experiments can be summarized as follows. For a single queue with several servers, without IS, the indirect estimator reduces the variance by large factors when the total arrival rate exceeds the service capacity, and increases the variance by large factors when the total arrival rate is much less than the service capacity. This is true even for constant service times and single-server queues, but less servers or less variability in the service times favors the direct estimator. A larger buffer at q^* tends to accentuate the factor of variance reduction or variance increase. When the indirect estimator was combined with IS, we observed a variance increase instead of a variance reduction, even if the total arrival rate after IS was larger than the service rate. An intuitive explanation seems to be that because IS is turned off as soon as the buffer overflows, the conditions favoring the indirect estimator (sustained overloading at q^*) do not hold for a large enough fraction of the time.

6 Functional Estimation

So far we have considered the problem of estimating the CLR for fixed values of the model parameters. But in real life one is often interested in a wide range of values of the r_{ij}'s and of the buffer sizes. We now examine how the CRL can be estimated in functional form, as a function of the matrix R, from a single simulation, and also as a function of B^* by re-using certain portions of the simulation.

Let R and $\tilde{R}$ be as before, where $\tilde{R}$ is the twisted version of R determined as in Section 3, but suppose that we now want to estimate the CLR μ for R replaced by $\check{R}$, for several $\check{R}$ in some neighborhood of R, by simulating pairs of A-cycles with $\tilde{R}$ and R only. This can be achieved as follows. One simulates pairs of A-cycles and computes X_i, Y_i, and the likelihood ratio L_i for each pair just as before. Afterwards, the estimators L_iY_i and X_i of the numerator and the denominator are multiplied by the likelihood ratios

$$L_i'(\check{R}) = \left(\frac{\check{r}_{00}}{r_{00}}\right)^{N_{00}'} \left(\frac{\check{r}_{01}}{r_{01}}\right)^{N_{01}'} \left(\frac{\check{r}_{10}}{r_{10}}\right)^{N_{10}'} \left(\frac{\check{r}_{11}}{r_{11}}\right)^{N_{11}'}$$

and

$$L_i''(\check{R}) = \left(\frac{\check{r}_{00}}{r_{00}}\right)^{N_{00}''} \left(\frac{\check{r}_{01}}{r_{01}}\right)^{N_{01}''} \left(\frac{\check{r}_{10}}{r_{10}}\right)^{N_{10}''} \left(\frac{\check{r}_{11}}{r_{11}}\right)^{N_{11}''},$$

respectively, where N_{kl}' and N_{kl}'' are the total number of transitions of the sources from state k to state l during the A-cycle with IS and without IS, respectively. The functional estimator of μ is then

$$\hat{\mu}(\check{R}) = \frac{\sum_{i=1}^n L_i'(\check{R}) L_i Y_i}{\sum_{i=1}^n L_i''(\check{R}) X_i}$$

and it can be evaluated *a posteriori* for as many different matrices $\check{R}$ as desired, as long as $\check{R}$ is not too far away from R. The additional overhead during the simulation amounts only to storing the values of N_{kl}' and N_{kl}'', together with those of X_i and L_iY_i, for all the pairs of A-cycles. This type of functional estimator based on a likelihood ratio is discussed in a more general context in [L'Ecuyer 1993] and [Rubinstein and Shapiro 1993], for example.

EXAMPLE **11** We give an example of functional estimation at level 4. Let $B_1 = B_2 = B_3 = B_4 = 500$, $m_0 = 4$, $m_1 = 6$, $m_2 = 8$, $c_1 = c_4 = 1$, $c_2 = c_3 = 3$, $\kappa_1 = 50$, and $\rho = 1/21$. We assign 4 sources to the node q^* and take $C = 150\,000$. We find $\tilde{R}$ and run the simulation as usual, and then compute two functional estimators. For the first one, κ_1 is fixed and μ is estimated as a function of κ_0, whereas for the second one, $\rho = \kappa_1/(\kappa_1 + \kappa_0)$ is fixed and μ is estimated as a function of κ_1. Tables 14 and 15 give a partial view of the results.

The relative half-widths of pointwise 99% confidence interval, $\hat{\delta}$, remain reasonable for a good range of values of κ_0 and κ_1. If one is interested in a wider region, that region can be partitioned into a few subintervals and a different $\tilde{R}$ can be used for each subinterval.

We now consider the estimation of μ as a fonction of B^*. For this, one cannot use the likelihood ratio approach, because B^* is not a parameter of a probability distribution in the model. However, observe that when an A-cycle is simulated, the sample path of the system is independent of B^* as long as there is no overflow at q^*. Therefore, when estimating μ for several large values of B^*, the initial part of the simulation (until overflow occurs) does not have to be repeated for each value. One can start with a *single* simulation (or sample path) and create a new subpath (or branch) each time the number of cells at q^* exceeds one of the buffer sizes of interest. If one is interested in N distinct values of B^*, one eventually ends up with N parallel simulations, but a lot of work is saved by starting these parallel simulations

Table 14: Functional estimation at level 4, for fixed κ_1

κ_0	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$
500	2.4E-6	6.5E-14	27%
541	1.6E-7	1.7E-14	22%
588	9.8E-7	4.0E-15	17%
645	6.0E-7	8.8E-16	13%
714	3.5E-7	1.9E-16	10%
800	2.0E-7	4.3E-17	7%
909	1.1E-7	1.0E-17	7%
1050	5.8E-8	2.7E-18	7%
1250	2.8E-8	7.2E-19	8%
1540	1.2E-8	2.1E-19	10%

Table 15: Functional estimation at level 4, for fixed ρ

κ_1	$\hat{\mu}$	$\hat{\sigma}^2$	$\hat{\delta}$
25.0	2.3E-13	3.7E-27	68%
29.4	1.2E-11	5.8E-24	50%
32.3	7.9E-11	1.2E-22	36%
35.7	5.0E-10	1.6E-21	21%
40.0	3.2E-9	1.9E-20	11%
45.5	2.1E-8	4.2E-19	8%
52.6	1.4E-7	1.4E-17	7%
62.5	9.0E-7	9.0E-16	9%
76.9	6.0E-6	1.8E-13	18%

only when needed. This type of approach is studied in more generality in [L'Ecuyer and Vázquez-Abad 1997]. In our experiments with this method, the savings in CPU time were typically around 50%.

The development of Section 3 suggests an approximately linear relationship between $\ln \mu$ and B^*, at least asymptotically. Our empirical experiments confirm that the linear model $\ln \mu = \beta_0 + \beta_1 B^*$ fits very well indeed, for large enough B^*. We can therefore recommend, for estimating μ as a function of B^*, to perform simulations at 4 or 5 values of B^* only, and fit a linear model to the observations $(B^*, \ln \hat{\mu})$ by least squares regression.

As an illustration, for the same model as in Example 11 and $\ell^* = 4$, Figure 2 shows the 10 points $(B^*, \ln \hat{\mu})$, for $B^* = 250, 300, \ldots, 700$. It is clear from the figure that a linear model is an excellent fit.

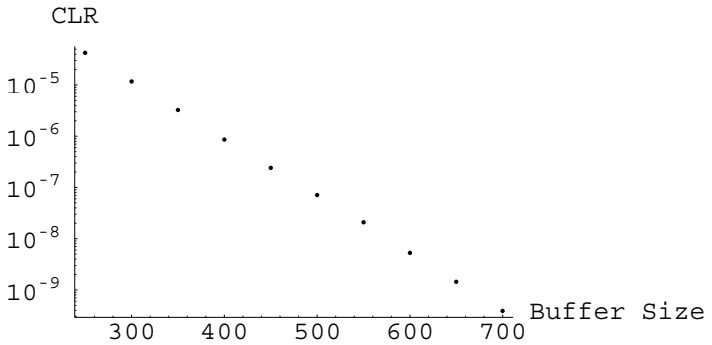

Figure 2: $\hat{\mu}$ as a function of B_4 at level 4

References

Beck, B., Dabrowski, A. R., and McDonald, D. R. 1998. A unified approach to fast teller queues and ATM. *Journal of Applied Probability*. To appear.

Bonneau, M.-C. 1996. Accelerated simulation of a leaky bucket controller. Master's thesis, Department of Mathematics and Statistics, University of Ottawa.

Bratley, P., Fox, B. L., and Schrage, L. E. 1987. *A Guide to Simulation* (Second ed.). Springer-Verlag, New York.

Champoux, Y. 1998. Estimation du taux de perte de réseaux ATM via la simulation et le changement de mesure. Master's thesis, Département d'IRO, Université de Montréal.

Chang, C. S., Heidelberger, P., Juneja, S., and Shahabuddin, P. 1994. Effective bandwidth and fast simulation of ATM intree networks. *Performance Evaluation 20*, 45–65.

Chang, C. S., Heidelberger, P., and Shahabuddin, P. 1995. Fast simulation of packet loss rates in a shared buffer communications switch. *ACM Transactions on Modeling and Computer Simulation 5*, 4, 306–325.

Dabrowski, A. R., Lamothe, G., and McDonald, D. R. 1998. Accelerated simulation of ATM switching fabrics. In *ITC'16* (1998). Submitted.

Fishman, G. S. 1996. *Monte Carlo: Concepts, Algorithms, and Applications.* Springer Series in Operations Research. Springer-Verlag, New York.

Glynn, P. W. 1994. Efficiency improvement techniques. *Annals of Operations Research 53*, 175–197.

Glynn, P. W. and Iglehart, D. L. 1989. Importance sampling for stochastic simulations. *Management Science 35*, 1367–1392.

Heegaard, P. E. 1998. *Efficient Simulation of Network Performance by Importance Sampling.* Ph. D. thesis, Norwegian University of Science and Technology.

Heidelberger, P. 1995. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation 5*, 1, 43–85.

LAW, A. M. AND KELTON, W. D. 1991. *Simulation Modeling and Analysis* (Second ed.). McGraw-Hill, New York.

L'ECUYER, P. 1993. Two approaches for estimating the gradient in a functional form. In *Proceedings of the 1993 Winter Simulation Conference* (1993), pp. 338–346. IEEE Press.

L'ECUYER, P. 1994. Efficiency improvement via variance reduction. In *Proceedings of the 1994 Winter Simulation Conference* (1994), pp. 122–132. IEEE Press.

L'ECUYER, P. AND CHAMPOUX, Y. 1996. Importance sampling for large ATM-type queueing networks. In *Proceedings of the 1996 Winter Simulation Conference* (1996), pp. 309–316. IEEE Press.

L'ECUYER, P. AND VÁZQUEZ-ABAD, F. 1997. Functional estimation with respect to a threshold parameter via dynamic split-and-merge. *Discrete Event Dynamic Systems: Theory and Applications 7*, 1, 69–92.

NICOLA, V. F., SHAHABUDDIN, P., HEIDELBERGER, P., AND GLYNN, P. W. 1993. Fast simulation of steady-state availability in non-markovian highly dependable systems. In *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing* (1993), pp. 38–47. IEEE Computer Society Press.

RUBINSTEIN, R. Y. AND SHAPIRO, A. 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method.* Wiley, New York.

SHAHABUDDIN, P. 1994. Importance sampling for the simulation of highly reliable markovian systems. *Management Science 40*, 3, 333–352.

SRIKANT, R. AND WHITT, W. 1997. Variance reduction in simulations of loss models. Manuscript.