

The Theory of Direct Probability Redistribution and its Application to Rare Event Simulation ¹

Zsolt Haraszti

J. Keith Townsend *

Center for Advanced Computing and Communication

Department of Electrical and Computer Engineering

North Carolina State University

Box 7914

Raleigh, N.C. 27695-7914

* Corresponding Author: email. jkt@eos.ncsu.edu, tel. (919) 515-7353, fax. (919) 515-2285

Abstract

Estimating rare event probabilities in communication systems using Monte Carlo simulation can require a prohibitively large number of trials for acceptable accuracy. We develop and present a technique in this paper that provides large speedup factors over conventional Monte Carlo simulation based on the theory of “direct probability redistribution” (DPR). We show that DPR can be viewed as a type of importance sampling, albeit “non-traditional”, which requires little additional effort to apply. The theory of DPR is presented for systems that can be described as a discrete time Markov chain (DTMC). We show that the RESTART technique is an important special case of direct probability redistribution. An efficient simulation technique based on DPR is less restrictive than RESTART, and is much less problem specific than traditional IS techniques.

We demonstrate how to use DPR to speedup Monte Carlo simulation of non-trivial systems by applying the technique to two examples: A 64×64 three-stage ATM switch, and an ATM multiplexer with internal flow control. For these systems we estimate small cell loss probabilities and show that the improvement is nearly inversely proportional to the probability estimate.

Keywords: *rare event simulation, importance sampling, RESTART, Markov chain*

¹The authors wish to thank Ericsson Telecom AB, Sweden, for providing financial support to Zsolt Haraszti while on leave at NCSU from Ericsson.

1 Introduction

Performance of communication systems is typically characterized by the occurrence of rare events. For example, cell loss probabilities in ATM switches are typically less than 10^{-9} . Analytical and numerical performance models of such systems are useful, but can require many simplifying assumptions. Monte Carlo simulation, on the other hand, is a practical alternative when the analysis requires fewer simplifying assumptions. However, estimating rare event probabilities with Monte Carlo techniques requires a prohibitively large number of trials in many interesting cases.

Many techniques for reducing the number of trials in Monte Carlo simulation using Importance Sampling (IS) have been proposed. Fundamentally, IS is based on the notion of modifying the underlying probability mass in such a way that the rare events occur much more frequently (see [1] for a good introduction). To correct for this modification, the results are weighted in a way that yields an estimate that is statistically unbiased. The goal of this effort is to have an estimate that achieves the same accuracy (variance) as the conventional Monte Carlo estimate, but with a much smaller number of trials.

More recent work applying IS to the analysis of stochastic systems, including queueing systems, can be found in [2, 3, 4, 5, 6, 7] and references therein. In such applications, modifications are applied to the stochastic “input” of the system (e.g., traffic sources in a communication system model) in order to change the response (“output”) of the system such that rare events occur more often. The *indirect* relationship between the random inputs that can be altered (“biased”) and the system output distribution of interest requires considerable problem specific effort to determine how the probability distributions of input traffic should be modified to yield improvement. This presents the chief limitation of IS to date. At best, IS techniques have been generalized to very narrow classes of problems. IS techniques that require minimal problem specific effort to apply have not been reported.

Another limiting characteristics of IS when applied to stochastic systems with memory is the decreasing likelihood ratios. To avoid the deleterious effect of decreasing likelihood ratios in these systems, IS techniques based on dynamic biasing [3, 6] and regenerative cycles [2, 3] have been developed.

In this paper, we present a technique that redistributes the probability mass of interest *directly* — hence the name “*direct probability redistribution*”. We develop the theory of direct probability redistribution for discrete-time Markov Chains (DTMC), and show how the theory can be applied to the problem of estimating the probability of buffer overflow. Within the scope of DTMC’s, the theory is general — probabilities corresponding to any subset of states can be estimated using the technique. The technique is easy to implement, and can be applied to complex systems, as we demonstrate here.

The rudiments of direct probability redistribution are first suggested by Bayes [8]. From an implementation point of view, the basic idea is to set thresholds at levels representing progressively rarer subsets of states. In a queueing system model, for example, the thresholds are set at different queue lengths. When a given subset of states is visited by the system (characterized by crossing a threshold), numerous random retrials are generated with the initial state for each retrial being the state of the system at the threshold. The basic technique for one threshold and rules for optimal choice of this threshold is named RESTART and presented in [9]. The RESTART technique is extended to multiple thresholds in [10].

A serious limitation of the technique presented in [10] is that it does not handle the case where system can make transitions that result in crossing multiple thresholds. This requires careful problem formulation in setting up the thresholds to ensure that no multiple crossing can occur, and in some cases it makes it difficult to allocate the proper number of thresholds. For example, consider the following simple discrete time queue. There are N independent sources feeding a single queue from which at most one cell is serviced at each time slot. When the load is relatively low, queue length probabilities diminish rapidly with increasing queue size, requiring the distance between thresholds to be small as specified in [10]. However, the non-zero probability that N cells may arrive in a slot sets the minimum threshold distance to N . It is easy to construct a situation when N is considerable higher than the optimal threshold distance.

The theory and techniques developed here using DPR can handle multiple threshold crossings, and simplify application to arbitrary systems. Furthermore, it requires no strict relationship between thresholds (subsets) and the rare event(s) of interest. DPR incorporates a homogeneous weighting factor that directly

redistributes the probability mass of the output distribution and thus avoids the problems associated with a decreasing likelihood ratio.

We develop the theory of direct probability redistribution in Section 2. Due to space limitations, we present the theorems without proof, although we do provide discussion to aid in understanding the theory. We show in this section that RESTART is a special case of direct probability redistribution. (Note that to understand how to *apply* the theory of DPR, Section 2 can be skipped.) Implementation of the technique is presented in Section 3. Here we present a verbal description of the method, and provide a pseudo-code representation. In Section 4, we demonstrate the technique by applying it to two non-trivial examples: A 64×64 three-stage ATM switch, and an ATM multiplexer with internal flow control. Concluding remarks are given in Section 5.

2 Theory of Direct Probability Redistribution

Let $\mathcal{S} = \{s_1, \dots, s_n\}$ denote the states of the homogeneous *discrete time Markov chain* (DTMC), and let $X_k \in [1, \dots, n]$, $k = 0, 1, \dots$ represent the system evolution observed at discrete time instances t_0, t_1, \dots . The chain is uniquely defined by the *transition probability matrix* (TPM) \mathbf{P} . The unique solution of the equation $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ yields the *steady state distribution vector* $\boldsymbol{\pi} = \{\pi_1, \pi_2, \dots, \pi_n\}$. We assume in this paper that the parameter of interest is a function of $\boldsymbol{\pi}$. Using the theory presented here, extreme probabilities of critical subsets of the state space can be increased arbitrarily in a directly controllable way.

We partition \mathcal{S} into $m \leq n$ non-empty *subsets* and denote those subsets by $\mathcal{S}_1, \dots, \mathcal{S}_m$. Let n_1, \dots, n_m be the sizes of the corresponding subsets. ($\sum_{i=1}^m n_i = n$.) Without loss of generality, we assume that states are indexed such that the following condition holds:

$$s_i \in \mathcal{S}_j \iff \sum_{k=1}^{j-1} n_k < i \leq \sum_{l=1}^j n_l \quad (1)$$

Condition (1) simply says that states are labeled in a continuous fashion:

$$\underbrace{s_1, \dots, s_{n_1}}_{\mathcal{S}_1}, \underbrace{s_{n_1+1}, \dots, s_{n_1+n_2}}_{\mathcal{S}_2}, \dots, \underbrace{s_{n-n_m+1}, \dots, s_n}_{\mathcal{S}_m}.$$

The TPM of the partitioned system is divided into $m \times m$ sub-matrices (blocks),

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1m} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{m1} & \mathbf{P}_{m2} & \dots & \mathbf{P}_{mm} \end{pmatrix},$$

where the diagonal blocks $\mathbf{P}_{11}, \dots, \mathbf{P}_{mm}$ are square matrices of size $(n_1 \times n_1), \dots, (n_m \times n_m)$. The off-diagonal blocks are the so-called *coupling matrices*, e.g., \mathbf{P}_{ij} gives transition probabilities from \mathcal{S}_i to \mathcal{S}_j . For a fixed labeling, assuming that condition (1) holds, the vector $\mathbf{n} = \{n_1, \dots, n_{m-1}\}$ uniquely defines the partitioning.

For didactical reasons, we first introduce the theory for two subsets and then extend it to multiple subsets.

2.1 Probability redistribution for two subsets ($m = 2$)

Let we partition the state space of a Markov chain with known TPM, \mathbf{P} , into two subsets. We define the following operation:

Definition 1 Operator $\mathbb{R}_{\mu}^{\mathcal{S}_2}: \mathbf{P}_{n \times n} \rightarrow \mathbf{P}'_{n \times n}$ is defined by the operation:

$$\mathbf{P}' = \mathbb{R}_{\mu}^{\mathcal{S}_2}\{\mathbf{P}\} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \frac{1}{\mu}\mathbf{P}_{21} & \mathbf{P}_{22} + (1 - \frac{1}{\mu})\mathbf{C}_{22} \end{pmatrix} \quad (2)$$

where $\mathbf{C}_{22} = \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{P}_{12}$, and μ is an arbitrary real number.

Note that $\mathbb{R}_1^{\mathcal{S}_2}\{\mathbf{P}\} = \mathbf{P}$. Note furthermore that the lower right block of \mathbf{P}' converges to the so-called *stochastic complement* of \mathbf{P}_{22} as $\mu \rightarrow \infty$ (see Section 6.2 in [11] for more detail on stochastic complements). We shall call the above generalization of stochastic complements “*partial stochastic complement*” for $1 \leq \mu < \infty$. Analogous to Theorems 6.2 and 6.3 in [11], the following two theorems can be formulated on the domain of partial complements:

Theorem 1 *Let \mathbf{P} be a homogeneous TPM, and μ be a finite real, $\mu \geq 1$. Then the transformed TPM, $\mathbf{P}' = \mathbb{R}_\mu^{\mathcal{S}_2}\{\mathbf{P}\}$, is also a homogeneous TPM. If \mathbf{P} is irreducible, \mathbf{P}' is irreducible too.*

Due to space limitations, we omit the proofs in this paper, they will be provided in an upcoming submission.

Theorem 2 *Let $\mathbf{P}' = \mathbb{R}_\mu^{\mathcal{S}_2}\{\mathbf{P}\}$, and let μ be a finite real, $\mu \geq 1$. Let $\boldsymbol{\pi}' = \{\pi'_1, \dots, \pi'_n\}$ be the steady state probabilities of \mathbf{P}' ($\boldsymbol{\pi}' = \boldsymbol{\pi}'\mathbf{P}'$). The following relationship holds between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$:*

$$\pi'_i = \begin{cases} \frac{1}{\theta} \pi_i & \text{if } 1 \leq i \leq n_1, \\ \frac{\mu}{\theta} \pi_i & \text{otherwise,} \end{cases} \quad (3)$$

where θ is a normalization constant $\theta = \sum_{i=1}^{n_1} \pi_i + \mu \sum_{i=n_1+1}^n \pi_i$.

Verbally, probabilities in the coupling matrix \mathbf{P}_{21} are reduced by a constant factor, and the resulting “released” mass is redistributed within \mathbf{P}_{22} . As a result, steady state probabilities are also redistributed, so that every state in \mathcal{S}_2 is visited relatively more often than in the original system (see Fig. 1). The mass in \mathcal{S}_2 is multiplied by μ , then the entire mass is re-normalized.

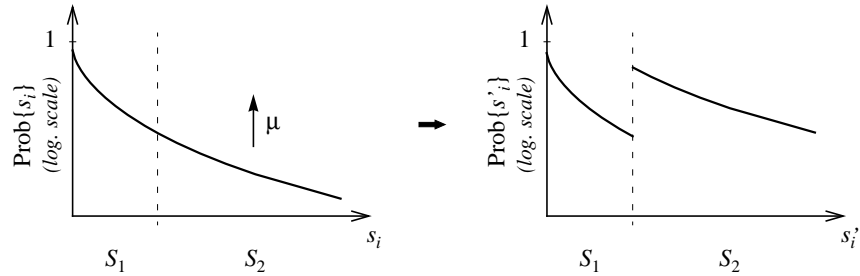


Figure 1: By applying the transformation, probabilities in \mathcal{S}_2 are uniformly increased by a constant factor which approximately equals μ . (Due to the re-normalization, probabilities in \mathcal{S}_1 slightly drop.)

We provide a simple example to help to understand the operation.

Example 1 Consider the following simple chain:

$$\mathbf{P} = \begin{pmatrix} .90 & .05 & .03 & .02 \\ .40 & .40 & .10 & .10 \\ .50 & .35 & .10 & .05 \\ .60 & .38 & .01 & .01 \end{pmatrix}$$

Let $n_1 = 2$, in which case the transformed TPM becomes:

$$\mathbf{P}' = \begin{pmatrix} 0.9 & 0.05 & 0.03 & 0.02 \\ 0.4 & 0.4 & 0.1 & 0.1 \\ \frac{1}{\mu}0.5 & \frac{1}{\mu}0.35 & 0.1 + (1 - \frac{1}{\mu})0.48 & 0.05 + (1 - \frac{1}{\mu})0.37 \\ \frac{1}{\mu}0.6 & \frac{1}{\mu}0.38 & 0.01 + (1 - \frac{1}{\mu})0.554 & 0.01 + (1 - \frac{1}{\mu})0.426 \end{pmatrix}.$$

state	Original system ($\mu = 1$)	$\mu =$		
		2	5	20
1	0.8199	0.7665	0.6412	0.3528
2	0.1105	0.1033	0.0864	0.0475
3	0.0399	0.0747	0.1561	0.3437
4	0.0297	0.0556	0.1163	0.2559

Table 1: Equilibrium probabilities for different μ oversampling ratios for the matrix in Example 1

Table 1 presents the numerical solution of the system for $\mu = 2, 5$ and 20 , as well as the solution of the original matrix ($\mu = 1$). It can be seen that applying equation (3) yields the original probability mass. Please compare the lowest probability in the $\mu = 1$ case (0.0297) to the lowest probability in the $\mu = 5$ case (0.0864). ■

What follows now is a series of modifications to the above theory to make it possible to implement the probability redistribution in a discrete event simulator.

It can be seen that the resulting TPM, \mathbf{P}' in (2), is the linear combination of \mathbf{P} and \mathbf{P}^∞ :

$$\mathbf{P}' = \frac{1}{\mu}\mathbf{P} + \left(1 - \frac{1}{\mu}\right)\mathbf{P}^\infty,$$

where

$$\mathbf{P}^\infty = \mathbb{R}_{\infty}^{\mathcal{S}_2}\{\mathbf{P}\}. \tag{4}$$

Hence, the transition probabilities for the transformed system can be written as

$$p'_{ij} = \begin{cases} p_{ij} & \text{if } Y = 0, \\ p_{ij}^\infty & \text{otherwise,} \end{cases}$$

where $Y \in [0, 1]$ is an i.i.d. binary RV with mean $E[Y] = 1 - 1/\mu$. It can be seen, however, that due to the special relationship between \mathbf{P} and \mathbf{P}^∞ , such a control variable affects the system evolution only in certain cases, namely, when the system is in \mathcal{S}_2 and would transition to \mathcal{S}_1 under the *original* coupling matrix \mathbf{P}_{21} . Let D' (*departure*) denote such an event. Due to the redistribution of a portion of the mass in \mathbf{P}_{21} , this event can be subdivided into two further complementing events, denoted by D and D^∞ : event D denotes a successful transition from subset \mathcal{S}_2 to subset \mathcal{S}_1 , while $D^\infty = D' \setminus D$ represents the event in which the system remains in \mathcal{S}_2 due to the *redistributed* mass resulting from the transformation. Consecutive D' events shall be labeled as D'_1, D'_2, \dots . Let the binary stochastic process $\mathbf{Y}^{D'} = Y_1^{D'}, Y_2^{D'}, \dots$ denote instances of Y in consecutive D' events. Note that $\mathbf{Y}^{D'}$ forms a Bernoulli process, with mean $1 - 1/\mu$.

The following extension of Theorem 2 generalizes the concept, replacing $\mathbf{Y}^{D'}$ by a general stochastic binary process \mathbf{G} with arbitrary correlation structure.

Theorem 3 (First extension of Theorem 2) *Let \mathbf{P} and \mathbf{P}^∞ be defined as before, and let the binary stochastic process $\mathbf{G} = \{G_1, G_2, \dots\}, G_l \in \{0, 1\}$ control the evolution of a DTMC such that*

$$\text{Prob}\{X_{k+1} = j \mid X_k = i\} = \begin{cases} p_{ij}^\infty & \text{if } D'_k \text{ and } G_k = 1, \\ p_{ij} & \text{otherwise} \end{cases}$$

If $E[\mathbf{G}] = 1 - 1/\mu$ then the system converges to a steady state probability vector $\boldsymbol{\pi}'$ which satisfies (3).

The following theorem, which is a second extension of Theorem 2, leads out of the domain of Markov chains. Nevertheless, it provides a very important simplification that makes it practical to implement the operation in discrete event simulators.

Theorem 4 (Second extension of Theorem 2) *Let a_l denote the state where the system last entered \mathcal{S}_2 from \mathcal{S}_1 . Let \mathbf{G} be defined as in Theorem 3, and let $\boldsymbol{\pi}'$ denote the steady state distribution of the system that is governed by \mathbf{G} such that:*

$$\text{Prob}\{X_{k+1} = j \mid X_k = i\} = \begin{cases} 1 & \text{if } D'_l \wedge G_l = 1 \wedge j = a_l, \\ 0 & \text{if } D'_l \wedge G_l = 1 \wedge j \neq a_l, \\ p_{ij} & \text{otherwise} \end{cases}$$

If $E[\mathbf{G}] = 1 - 1/\mu$ then equation (3) holds.

The above theorem says that given the occurrence of a D' event, the current value of the G process determines if the transition is accepted or the system is forced to return to the state where it last entered \mathcal{S}_2 from \mathcal{S}_1 . This mechanism can easily be implemented in an event-driven simulator: *Upon an $\mathcal{S}_1 \rightarrow \mathcal{S}_2$ transition, the system state is saved. When the next D'_l event occurs, the transition is accepted only if $G_l = 0$, otherwise the last saved system state is restored.* Simulation continues from the restored state using a new random number sequence.

This is very similar to the 1-threshold RESTART mechanism [9]. In fact, RESTART is a special case of the above method: in RESTART, μ can only be integer and G is a deterministic sequence producing the following pattern:

$$\dots 0 \underbrace{111\dots 10}_{\mu-1} \underbrace{111\dots 10}_{\mu-1} \underbrace{111\dots 10}_{\mu-1} \dots$$

Another more important difference is in the way the *target event* (or *rare event*) under investigation is being observed. Assume that the goal of the simulation is to estimate the probability of the occurrence of event E , where E is associated to visiting some subset $\mathcal{E} \subset \mathcal{S}$ (formally: $E : s_i \in \mathcal{E} \subset \mathcal{S}$). In RESTART, \mathcal{E} must be a subset of \mathcal{S}_2 , while in our case \mathcal{E} can have arbitrary relation to the partitioning. We know that in the altered system, every state in \mathcal{S}_2 is visited relatively μ times more often than in the original system. Therefore, when an event E occurs while the system is in subset \mathcal{S}_2 , corresponding event counters should be incremented by $1/\mu$, instead of unity as in \mathcal{S}_1 .

2.2 General case: $m \geq 2$

In cases where the state probabilities span many orders of magnitude, it is straight-forward to apply the above operation by assigning different oversampling factors to different subsets, as shown in Fig. 2.

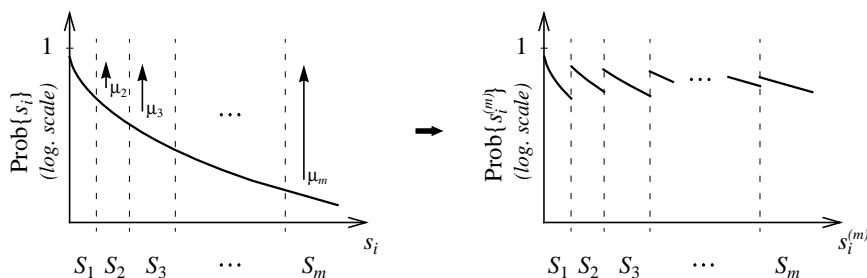


Figure 2: Applying the operation multiple times, low probabilities can be further eliminated.

In this section we develop the underlying theory of *direct probability redistribution* for the more general case of multiple subsets. Before proceeding with the development we shall present some important characteristics of operator \mathbb{R} .

Proposition 1 If μ_1 and μ_2 are real numbers such that $\mu_1, \mu_2 \geq 1$, and $\mathcal{A} \subset \mathcal{S}$ then

$$\mathbb{R}_{\mu_1}^{\mathcal{A}} \left\{ \mathbb{R}_{\mu_2}^{\mathcal{A}} \{ \mathbf{P} \} \right\} \equiv \mathbb{R}_{\mu_1 \mu_2}^{\mathcal{A}} \{ \mathbf{P} \}. \quad (5)$$

Proposition 2 (Commutative characteristics) If $\mu_{\mathcal{A}}$ and $\mu_{\mathcal{B}}$ are real numbers such that $\mu_{\mathcal{A}}, \mu_{\mathcal{B}} \geq 1$, then

$$\mathbb{R}_{\mu_{\mathcal{A}}}^{\mathcal{A}} \left\{ \mathbb{R}_{\mu_{\mathcal{B}}}^{\mathcal{B}} \{ \mathbf{P} \} \right\} \equiv \mathbb{R}_{\mu_{\mathcal{B}}}^{\mathcal{B}} \left\{ \mathbb{R}_{\mu_{\mathcal{A}}}^{\mathcal{A}} \{ \mathbf{P} \} \right\}. \quad (6)$$

for any $\mathcal{A} \subset \mathcal{S}$ and $\mathcal{B} \subset \mathcal{S}$.

Corollary 1 (From Proposition 1 and 2): If $\mathcal{A} \cap \mathcal{B} = \emptyset$, and $\mu_{\mathcal{A}} \leq \mu_{\mathcal{B}}$ then

$$\mathbb{R}_{\mu_{\mathcal{A}}}^{\mathcal{A}} \left\{ \mathbb{R}_{\mu_{\mathcal{B}}}^{\mathcal{B}} \{ \mathbf{P} \} \right\} \equiv \mathbb{R}_{\mu_{\mathcal{A}}}^{\mathcal{A} \cup \mathcal{B}} \left\{ \mathbb{R}_{\mu_{\mathcal{B}}/\mu_{\mathcal{A}}}^{\mathcal{B}} \{ \mathbf{P} \} \right\}. \quad (7)$$

Let the positive real vector $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_m\}$ represent the (desired) oversampling factors for the respective subsets. Without loss of generality, we assume that $\mu_1 = 1$ and $\mu_1 \leq \mu_2 \leq \dots \leq \mu_m$. The former condition can be achieved by normalizing $\boldsymbol{\mu}$ to its smallest element, still keeping the ratio between oversampling factors for any pair of the subsets. The latter condition can be achieved by relabeling the states in \mathcal{S} in a way that condition (1) remains valid.

After performing the operation $\mathbb{R}_{\mu_k}^{\mathcal{S}_k}$ for each $k = (2, 3, \dots, m)$ we can express the resulting TPM as

$$\mathbf{P}^{(m)} = \mathbb{M}_{\boldsymbol{\mu}} \{ \mathbf{P} \} = \mathbb{R}_{\mu_2}^{\mathcal{S}_2} \left\{ \mathbb{R}_{\mu_3}^{\mathcal{S}_3} \left\{ \dots \mathbb{R}_{\mu_{m-1}}^{\mathcal{S}_{m-1}} \left\{ \mathbb{R}_{\mu_m}^{\mathcal{S}_m} \{ \mathbf{P} \} \right\} \right\} \right\}.$$

Using equations (5) and (7), this expression can be rewritten as

$$\mathbf{P}^{(m)} = \mathbb{M}_{\boldsymbol{\mu}} \{ \mathbf{P} \} = \mathbb{R}_{\mu_2}^{\mathcal{G}_2} \left\{ \mathbb{R}_{\mu_3/\mu_2}^{\mathcal{G}_3} \left\{ \dots \mathbb{R}_{\mu_{m-1}/\mu_{m-2}}^{\mathcal{G}_{m-1}} \left\{ \mathbb{R}_{\mu_m/\mu_{m-1}}^{\mathcal{G}_m} \{ \mathbf{P} \} \right\} \right\} \right\}, \quad (8)$$

where \mathcal{G}_k denotes the group of subsets $\mathcal{G}_k = \mathcal{S}_k \cup \mathcal{S}_{k+1} \cup \dots \cup \mathcal{S}_m$.

The following theorem provides the relationship between the steady state probabilities of the original and the altered system.

Theorem 5 Let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^{(m)}$ represent the steady state probability vectors of \mathbf{P} and $\mathbf{P}^{(m)}$, respectively. The following relationship holds for every $i \in \{1, \dots, n\}$:

$$\pi_i^{(m)} = \frac{\mu_{\Gamma(i)}}{\Phi} \pi_i \quad (9)$$

where $\Gamma(i)$ is the so-called subset indicator i.e., $\Gamma(i) = k$ indicates that state s_i is in subset \mathcal{S}_k . Φ is a normalization constant, $\Phi = \sum_{i=1}^n \mu_{\Gamma(i)} \pi_i$.

Now we shall develop the closed formula for $\mathbf{P}^{(m)}$. First, we define the auxiliary mass $\mathbf{D}_{ij}^{(k)}$ in the following way:

When operator \mathbb{R} is applied on the m -partitioned \mathbf{P} such that \mathcal{G}_k is oversampled by a factor of μ_k , in a similar way to (2) the resulting TPM has the form:

$$\mathbb{R}_{\mu_k}^{\mathcal{G}_k} \{ \mathbf{P} \} = \begin{pmatrix} \mathbf{Q}_{11} & & \mathbf{Q}_{12} \\ \frac{1}{\mu_k} \mathbf{Q}_{21} & \mathbf{Q}_{22} + \left(1 - \frac{1}{\mu_k}\right) \mathbf{Q}_{21} (\mathbf{I} - \mathbf{Q}_{11})^{-1} \mathbf{Q}_{12} & \\ & & \end{pmatrix} \quad (10)$$

where

$$\mathbf{Q}_{11} = \begin{pmatrix} \mathbf{P}_{11} & \dots & \mathbf{P}_{1,k-1} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{k-1,1} & \dots & \mathbf{P}_{k-1,k-1} \end{pmatrix} \quad \mathbf{Q}_{12} = \begin{pmatrix} \mathbf{P}_{1k} & \dots & \mathbf{P}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{k-1,k} & \dots & \mathbf{P}_{k-1,m} \end{pmatrix}$$

$$\mathbf{Q}_{21} = \begin{pmatrix} \mathbf{P}_{k1} & \dots & \mathbf{P}_{k,k-1} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{m1} & \dots & \mathbf{P}_{m,k-1} \end{pmatrix} \quad \mathbf{Q}_{22} = \begin{pmatrix} \mathbf{P}_{kk} & \dots & \mathbf{P}_{km} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{mk} & \dots & \mathbf{P}_{mm} \end{pmatrix}.$$

The lower right block of $\mathbb{R}_{\mu_k}^{\mathcal{G}_k}\{\mathbf{P}\}$ in (10) can be reformulated such as:

$$\mathbf{Q}_{22} + \left(1 - \frac{1}{\mu_k}\right) \mathbf{Q}_{21}(\mathbf{I} - \mathbf{Q}_{11})^{-1} \mathbf{Q}_{12} = \begin{pmatrix} \mathbf{P}_{kk} + \left(1 - \frac{1}{\mu_k}\right) \mathbf{D}_{kk}^{(k)} & \dots & \mathbf{P}_{km} + \left(1 - \frac{1}{\mu_k}\right) \mathbf{D}_{km}^{(k)} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{mk} + \left(1 - \frac{1}{\mu_k}\right) \mathbf{D}_{mk}^{(k)} & \dots & \mathbf{P}_{mm} + \left(1 - \frac{1}{\mu_k}\right) \mathbf{D}_{mm}^{(k)} \end{pmatrix}$$

where $\mathbf{D}_{ij}^{(k)}$ is defined for every $(i, j) : i \in \{k, \dots, m\}, j \in \{k, \dots, m\}$ by:

$$\mathbf{D}_{ij}^{(k)} = \begin{pmatrix} \mathbf{P}_{i1} & \dots & \mathbf{P}_{i,k-1} \end{pmatrix} (\mathbf{I} - \mathbf{Q}_{11})^{-1} \begin{pmatrix} \mathbf{P}_{1j} \\ \vdots \\ \mathbf{P}_{k-1,j} \end{pmatrix}. \quad (11)$$

Note that the upper index, k , in the above notation refer to the fact that the operator has been applied to subset \mathcal{G}_k .

Omitting the rather long derivation of the entire transformation $\mathbb{M}_{\mu}\{\mathbf{P}\}$, the blocks of the resulting TPM, $\mathbf{P}^{(m)} = \{\mathbf{P}_{ij}^{(m)}\}$ can be simplified to the following compact form:

$$\mathbf{P}_{ij}^{(m)} = \begin{cases} \mathbf{P}_{ij} + \sum_{k=2}^i \frac{\mu_k - \mu_{k-1}}{\mu_i} \mathbf{D}_{ij}^{(k)} & \text{if } i \leq j, \\ \frac{\mu_j}{\mu_i} \mathbf{P}_{ij} + \sum_{k=2}^j \frac{\mu_k - \mu_{k-1}}{\mu_i} \mathbf{D}_{ij}^{(k)} & \text{if } i > j. \end{cases} \quad (12)$$

The interpretation of (12) is not as straight-forward as it was for the two subset case, therefore we shall present it as a theorem.

Theorem 6 (*Proportional ticketing*) Consider a system that uses TPM, \mathbf{P} , and oversampling vector, μ , according to the following mechanism:

Let the positive integer variable, Ω , called the current ticket (or simply, ticket) control the system evolution and be maintained according to the following three rules:

1. The system, being in some state in \mathcal{S}_i , generates its next “target” state according to TPM \mathbf{P} . Let \mathcal{S}_j denote the subset of the aimed state. The transition is performed only if $j \geq \Omega$. (The ticket identifies the lowest possible index of any subset to which the system is allowed to transition while in possession of said ticket.)
2. If $j < \Omega$, the aimed transition is blocked, and another state is chosen according to TPM, $\mathbb{R}_{\infty}^{\mathcal{G}_\Omega}\{\mathbf{P}\}$. Then system transitions to this state. Let \mathcal{S}_j be reset according to the new choice.
3. In either case, after the transition to \mathcal{S}_j , a new ticket is generated —independently of the current one— with the following distribution:

$$\text{Prob}\{\Omega = k\} = \begin{cases} \frac{\mu_k - \mu_{k-1}}{\mu_j} & \text{if } 1 \leq k \leq j, \\ 0 & \text{otherwise.} \end{cases} \quad (\mu_0 \stackrel{\text{def}}{=} 0) \quad (13)$$

The above system converges with its steady state probabilities to $\boldsymbol{\pi}^{(m)}$.

Since the above theorem represents the most significant result of the theory, mapping a relatively complex formula to a rather simple system control mechanism, we provide the reader with its proof in Appendix A.

Just as for $m = 2$, there are two important generalizations of the above theorem:

The first one is the relaxation of the independent ticket selection mechanism to a generalized stochastic process with arbitrary correlation. The resulting steady state probabilities yield (9) as long as the long term distribution of tickets in subset \mathcal{S}_j follows (13). It can also be proven that the relationship holds even if the ticket is regenerated only after an “upward” transition, that is when system transitions from \mathcal{S}_i to an \mathcal{S}_j such that $j > i$.

It can also be easily seen that the stochastic step based on TPM $\mathbb{R}_{\infty}^{\mathcal{G}_{\Omega}}\{\mathbf{P}\}$, which happens upon a blocked transition, can be replaced by a deterministic transition to the state where system last entered \mathcal{G}_{Ω} from \mathcal{G}_{Ω}^C , resulting in the same steady state distribution as (9). This is an important extension to Theorem 6, offering a restart-type reformulation of DPR that can easily be implemented in discrete event simulation. The algorithm works as follows:

Subsequent states are generated according to \mathbf{P} . When system makes a transition from any state of subset \mathcal{S}_i to a state s_a of a “higher” subset \mathcal{S}_j ($i < j$), a series of independent retrials (*subtrajectories*) are launched from s_a , with new tickets assigned from the set $\{i + 1, \dots, j\}$. The number of subtrajectories assigned with ticket k , denoted by R_k , is determined in a stochastic fashion that ensures that

$$E_{S_i \rightarrow S_j} [R_k] = \frac{\mu_k - \mu_{k-1}}{\mu_i} \quad k = i + 1, \dots, j, \quad (14)$$

where $E_{S_i \rightarrow S_j} [\cdot]$ denotes the mean of R_k 's over all the possible transitions from \mathcal{S}_i to \mathcal{S}_j .

As in the two-subset case, oversampling the subsets means that every state in \mathcal{S}_k is visited μ_k times more often (on average) than in the original system. This implies that any event that is related to \mathcal{S}_k has a weight $1/\mu_k$, instead of unity. Generally, event counters are incremented by $1/\mu_{\Gamma(i)}$ when the system is in state s_i .

3 Implementation

The pseudo code in Fig. 3 introduces an implementation of the simulation mechanism developed above. We used *state vector* \mathbf{V}_k to represent the state of the system in time instant t_k . \mathbf{V} collects all the individual variables of the system model in the simulation program that —together— unambiguously define the system state. In a communication systems model, such variables can represent states of independent Markovian and other stochastic sources, length and/or contents of individual queues, etc.

Simulation is started from an arbitrary initial state, but the initial ticket should be set to $\Omega = 1$. Subsequent states are generated by the function, *transition*. During the simulation course two types of statistics are gathered. The global² vector, $\mathbf{h} = (h_1, \dots, h_m)$, called *hit counter*, counts how many times each subset is visited. As it will be briefly mentioned later in the numerical result section, the hit distribution can aid simulation practitioners to find an appropriate oversampling vector ($\boldsymbol{\mu}$).

A second set of statistics is related to the investigated *target* parameter itself. Here we assume that the target parameter is bound to the probabilities of a set of *exclusive* and *exhaustive* events $E_i \subset \mathcal{S}, i = 1, 2, \dots, n_e$, where n_e denotes the number of such events. The so called *target event indicator* function, denoted and defined as $\Lambda(\mathbf{V}_k) = i \iff X_k \in E_i$, is used for identifying the actual event. The global *target event counter* vector, $\mathbf{e} = (e_1, \dots, e_{n_e})$, is incremented by the current weighting factor $1/\mu_{\Gamma(\mathbf{V}_k)}$.

Should the current subset indicator be higher than in the previous cycle, independent trajectories (retrials) are launched by recursive function calls. To each new trajectory a new ticket is assigned. The number of new trajectories and their tickets are determined in a stochastic fashion according to (14), ensuring that the long term distribution of tickets in any subset of the state space follows the distribution defined by (13). To ensure independence between successive trajectories, the internal state of random number generators are not part of state vector \mathbf{V} .

The *if* condition in the beginning of the outer (while) loop ensures that the trajectory remains in the allowed region of the state space. When the new state falls into a subset that is prohibited by the current ticket, the trajectory is killed.

²By *global* we mean that recursive function calls operate over the same single instance of the said variable.

```

procedure simulate ( $k, \mathbf{V}_k, \Omega$ )                                /* Initial call:  $k = 1, \mathbf{V}_1$ : arbitrary,  $\Omega = 1$  */
  while  $k \leq k_{max}$  do
     $\mathbf{V}_{k+1} := \text{transition}(\mathbf{V}_k)$                                 /* Make state transition(s) */
    if  $\Gamma(\mathbf{V}_{k+1}) < \Omega$  then                                /* Investigate resulting subset indicator */
      return                                                       /* Return if indicator smaller than ticket */
     $h_{\Gamma(\mathbf{V}_{k+1})} := h_{\Gamma(\mathbf{V}_{k+1})} + 1$                 /* Bookkeeping for the new state */
     $e_{\Lambda(\mathbf{V}_{k+1})} := e_{\Lambda(\mathbf{V}_{k+1})} + 1/\mu_{\Gamma(\mathbf{V}_{k+1})}$ 
    if  $\Gamma(\mathbf{V}_{k+1}) > \Gamma(\mathbf{V}_k)$  then                        /* Conduct retries if indicator larger than previous */
       $n_{done} := 1$                                                 /* Number of trajectories done */
      for  $i = \Gamma(\mathbf{V}_k) + 1$  to  $\Gamma(\mathbf{V}_{k+1})$  do                /* One loop for each type of tickets */
         $n :=$  an occurrence of RV  $\mathbf{X}$  such that                    /* Set desired number of traj's */
           $n \in \mathbb{N}^+$  and  $E[\mathbf{X}] = \mu_i/\mu_{\Gamma(\mathbf{V}_k)} - n_{done}$ . /* ... issued with ticket  $\Omega = i$  */
        for  $i = 1$  to  $n$ 
           $h_{\Gamma(\mathbf{V}_{k+1})} := h_{\Gamma(\mathbf{V}_{k+1})} + 1$             /* Bookkeeping for extra trajectory */
           $e_{\Lambda(\mathbf{V}_{k+1})} := e_{\Lambda(\mathbf{V}_{k+1})} + 1/\mu_{\Gamma(\mathbf{V}_{k+1})}$ 
          simulate ( $k + 1, \mathbf{V}_{k+1}, i$ )                          /* Launch trajectory with given ticket*/
        end
       $n_{done} := \mu_i/\mu_{\Gamma(\mathbf{V}_k)}$ 
    end
  end
end
end
end

```

Figure 3: Pseudo-code of the recursive simulation controller. k_{max} , \mathbf{h} and \mathbf{e} are global variables.

At the end of the simulation, target event probabilities can be derived directly from the vector \mathbf{e} . Independent samples for calculating the confidence of the resulting estimates can be obtained via independent replications. The weighting factors are not accumulated as in conventional IS simulation of stochastic systems, therefore they do not diminish by the increasing simulation time. Hence, there is no limitation on the length of replications. This is another substantial simplification over conventional IS.

4 Numerical Results

We have developed and tested the technique on several queueing systems with known solutions. The two problems we have selected to present here represent more non-trivial examples: An 64×64 ATM switching fabric with three stage of switching elements, and an ATM multiplexer with internal flow control. We provide some supporting approximative solutions for both of the presented examples. The ATM multiplexer with flow control represents the first time an IS based simulation is applied to a queueing system with feedback. Simulation speed-up factors that is approximately inversely proportional to the target rare event probabilities have been observed.

4.1 Buffer overflow probabilities in a multistage ATM switch

The switch consists of three stages of 8×8 switching elements (SE), eight of them in each stage (see Fig. 5). Each SE is an output buffered switch with a nonblocking shared bus switching core and with eight separated buffers with uniform buffer size K . Cells arriving in the inlets of a SE in a given time slot are passed to the end of the queue in the appropriate output buffer in a random order.

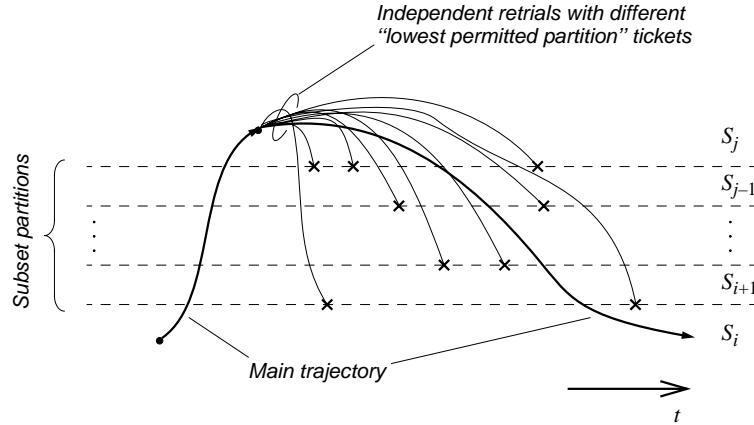


Figure 4: Trajectories of independent retrials after skipping multiple boundaries of state space partitions. Permissions to visit lower indexed partitions are issued proportionally to the desired oversampling factors in the intermediate partitions. (RESTART does not allow multiple threshold skips.)

SE's in subsequent stages are interconnected in a regular mesh as it is shown in the figure. There are eight alternative routes between any pair of input and output ports of the switch, depending which SE in the intermediate stage is used. The switch is used to route bursty data traffic connections, with each port multiplexing several connections to the same cell stream. Routing within the switch is decided on a per connection basis. In order to provide a numerically tractable problem, we assume the homogeneous case where the traffic is entirely symmetric. Thus, between each pair of input-output ports there are eight connections, each of them is routed in a different way in the switch. Connections entering at a given port to the switch are modeled by a single ON-OFF source (i.e., geometrically distributed burst and silence periods, having no idle cell in the bursts), with average burst size, b [cells], and relative load, α . To model the connections, each burst is independently addressed to a randomly selected output port and routed independently within the switch. This represents the 512 simultaneous uniform connections with non-interleaving bursts.

We estimate the probability of cell loss in the third stage of the fabric, approximated via the buffer overflow probability. We partition the state space along the selected buffer, using the actual queue length as subset indicator. (Using RESTART terminology, this yields K thresholds along the buffer.) The selection of oversampling vector, μ , was based on the following intuitive approach. To obtain samples along the entire buffer, the exponentially decaying tail had to be increased to a nearly flat (balanced) hit pmf by probability redistribution. To set μ , we used the inverse of an estimate of the queue length pmf, obtained via short simulation trials according to the following simple iterative procedure: we set $\mu = \mathbf{1}$ (no oversampling) which estimates the portion of the queue length pmf corresponding to small values of queue length. This pmf was used to refine μ for the next simulation trial which provided an estimate of the pmf corresponding to larger values of queue length. The process was repeated until the pmf was “explored” along the entire buffer. This was used to set μ for a set of final simulation replications. The entire initial exploration process was automated. A total of 300,000 slots were used in the exploration process, which is negligible compared to the $50 \times 500,000$ slots to get the final estimates.

Fig. 6 presents the simulation result in terms of the resulting queue length pmf for the following system configuration: $K = 240$ [slots], $\alpha = 90\%$ and $b = 10$. The result was generated from 50 independent replications, each involving a total number of 500,000 cell slots (including the subtrajectories of DPR). Results from conventional Monte Carlo simulation using the same amount of samples is also presented in the figure. The dashed line shows the theoretical approximation of the pmf, assuming that bursts arriving at the third stage remain un-interleaved, thus omitting the smoothing effect of the first two stages. Since

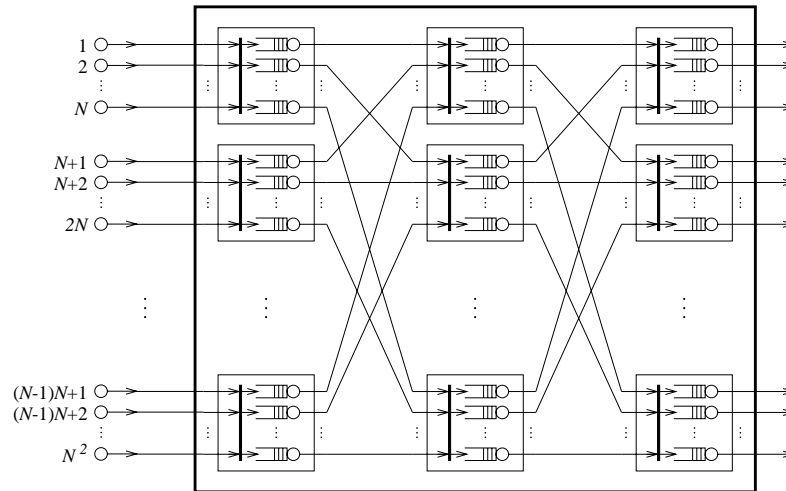


Figure 5: The 64×64 ATM switch with three stages of switching elements.

interleaving effects only cell contention, it has negligible effect on the tail of the queue length distribution. The result shows that this assumption is valid and demonstrates that the simulation provides accurate results.

Fig. 7 shows the resulting buffer overflow probabilities for five different mean burst sizes, $b = 5, 10, 20, 50$ and 100 , with $K = 240$ and $\alpha = 90\%$. The approximated theoretical result falls within the 90% confidence interval at all points. Improvement factors (speed-up) have been estimated by comparing the sample size used by DPR to the (hypothetical) sample size that conventional MC would need to achieve similar confidence, assuming uncorrelated rare events. This leads to a rather conservative estimation of the speed-up, since traffic was indeed strongly correlated, which would require even larger sample size in the conventional MC simulation.

4.2 ATM multiplexer with internal flow control

Our second example is a system with infinite memory: a flow-controlled ATM multiplexer (MUX) with N input ports (Fig. 8). The MUX operates in the following way: Arriving cells are multiplexed to the output port via a central FIFO buffer with K cell slots. To minimize the probability of cell loss a throttling mechanism is employed, which can temporarily prevent cells from being transferred to the congested main buffer. Throttled cells are temporarily stored at the port in a FIFO buffer with L slots. (L is chosen sufficiently high such that the probability of cell loss at the input buffers is negligible compared to the main buffer.) The grade of throttling is adjusted according to the actual queue length, x , via a throttling function, $f(x)$. Forwarding probability, $p = f(x)$, specifies the probability with which a cell is forwarded to the main queue, provided that there is at least one cell available at the port. The decision is made according to an i.i.d. Bernoulli RV, $F = [0, 1]$, with mean, p , where $F = 1$ means that the cell is forwarded. To represent the communication delay between the main queue and the input ports, a fix delay τ is introduced. Input traffic is modeled by independent ON-OFF sources, with average burst length, b , and relative load, α , per source.

The generality of the model enabled us to investigate the performance of different throttling functions and different feedback delays. Fig. 9 and 10 presents the results of the DPR simulation for the configuration $N = 16$, $K = 84$, $L = 500$, $b = 3$, $\alpha = 0.4$ and $\tau = 3 \times$ slot time. The throttling function was defined as:

$$f(x) = \begin{cases} 1 & \text{if } x \leq K - h \\ 0 & \text{otherwise,} \end{cases}$$

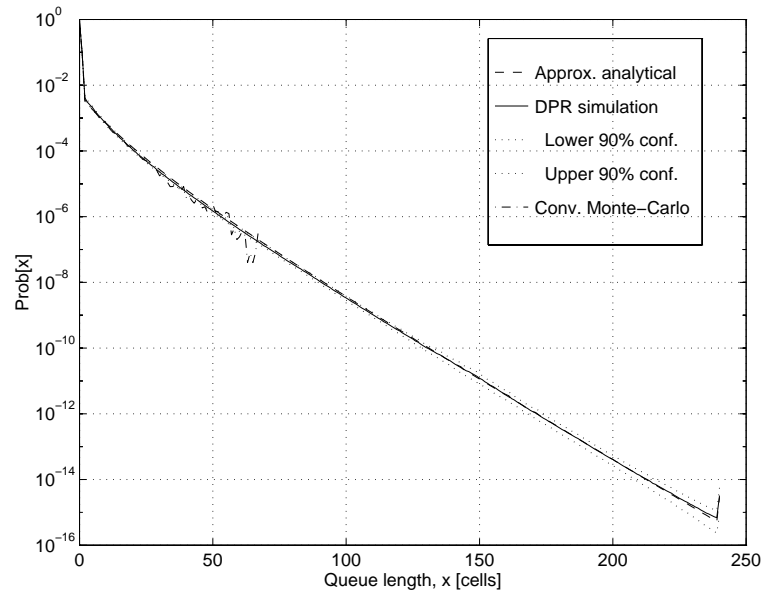


Figure 6: Queue length probability mass function (pmf) in a third-stage buffer, simulated using the DPR method. Confidence intervals (90%) and approximated theoretical distribution are also shown. For comparison, results of conventional MC simulation using the same sample size as DPR is included (no samples were obtained above $x = 67$).

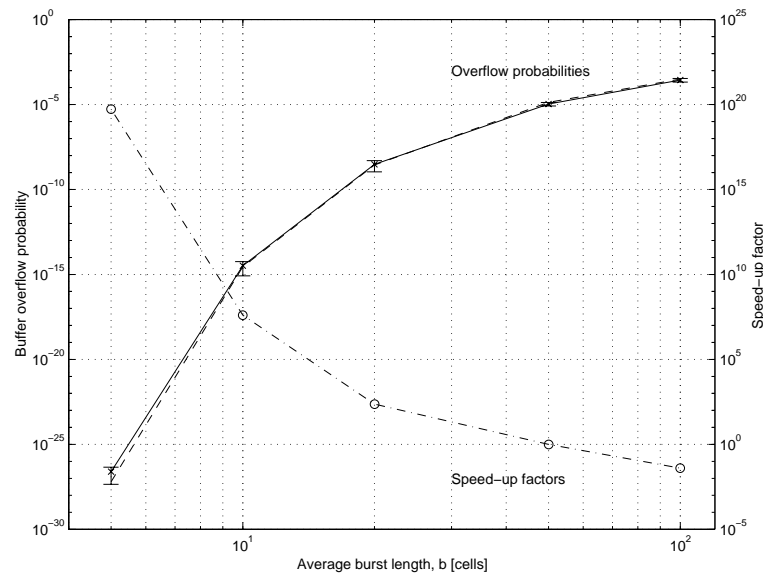


Figure 7: Overflow probabilities in the third-stage buffer vs. five different average burst lengths, simulated by DPR. 90% confidence intervals, approximate theoretical values (dashed line) and estimated speed-up factors (dashdot line) are also presented.

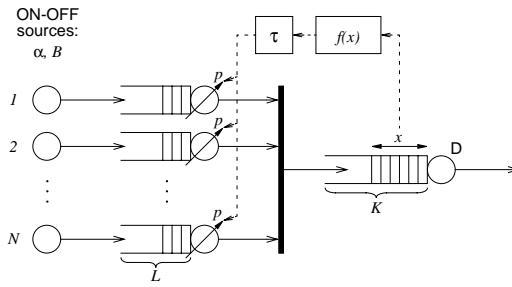


Figure 8: The ATM multiplexer with internal flow-control.

where h is a preset threshold.

Similar to the previous example, queue length, x , has been used as subset indicator for DPR. Oversampling vector, μ , has been selected using the same automated iterative procedure as in the multistage switch case. Fig. 9 shows the resulting queue length pmf for six different threshold settings ($h = 0, 5, 10, 15, 20, 25$), with 90% confidence intervals also displayed. For each threshold, 25 independent replications were ran, each containing less than 10^6 accumulated slots.

Note that the special case $h = 0$ yields the $N \times \text{ON-OFF}/D/1/K$ problem, for which numerical result exists. The analytical pmf perfectly matches with the $h = 0$ curve, therefore it is not visible in Fig. 9.

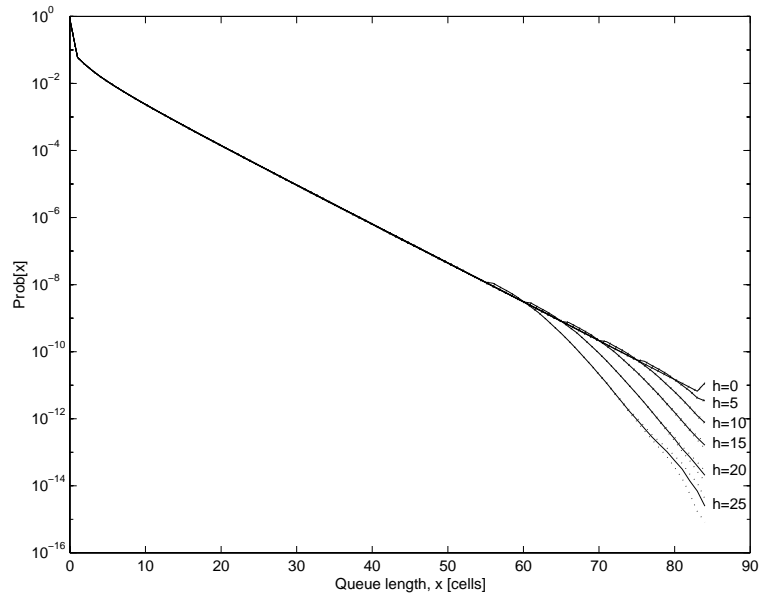


Figure 9: Queue length probability mass functions in the main buffer of the flow-controlled ATM multiplexer, with six different threshold settings. Dotted lines show 90% confidence intervals.

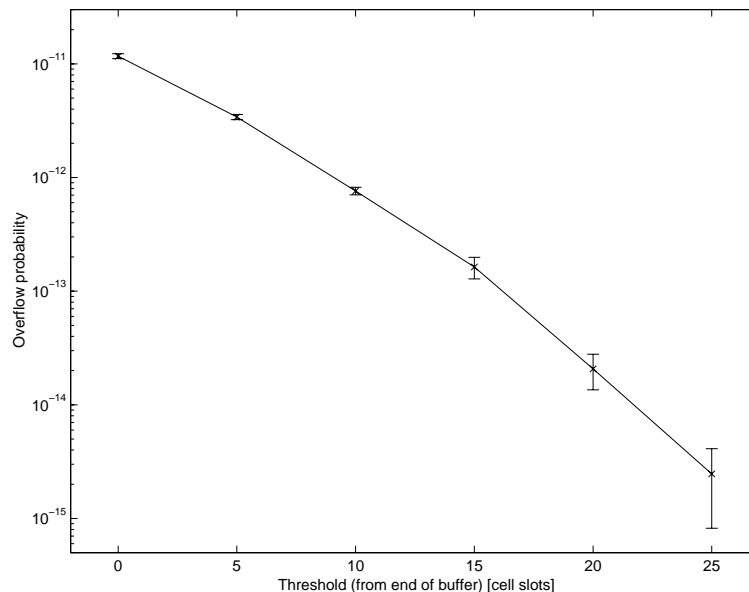


Figure 10: Resulted buffer overflow probabilities in central buffer in the flow-controlled ATM multiplexer vs. threshold values (with 90% confidence intervals)

5 Conclusion

We have developed the theory of direct probability redistribution, an IS technique that provides large speedup factors over conventional Monte Carlo simulation. We focus in this paper on a discrete time Markov chain formulation. Within the scope of DTMC's, the theory is general — probabilities corresponding to any subset of states can be estimated using the technique. DPR is novel in that the IS modifications are applied directly to the system response (“*output*”), which alleviates the problems introduced by decreasing likelihood ratio functions that are present in more “traditional” IS schemes. Also, efficient simulation using DPR is less restrictive than RESTART, and is much less problem specific than traditional IS techniques.

The technique is easy to implement, and can be used with complex systems, as we have demonstrated by applying the technique to two nontrivial examples: A 64×64 ATM switch, and an ATM multiplexer with internal flow control. For these systems we have estimated small cell loss probabilities and have shown that the improvement is nearly inversely proportional to the probability estimate.

References

- [1] C. E. Clark. Importance Sampling in Monte Carlo Analyses. *Operations Research*, pages 603–620, September-October 1961.
- [2] P. W. Glynn and D. L. Iglehart. Importance Sampling for Stochastic Simulations. *Management Science*, 35(11):1367–1392, Nov. 1989.
- [3] M. Devetsikiotis and J. K. Townsend. Statistical Optimization of Dynamic Importance Sampling Parameters for Efficient Simulation of Communication Networks. *IEEE/ACM Trans. Networking*, 1(3):293–305, June 1993.

- [4] Q. Wang and V. S. Frost. Efficient Estimation of Cell Blocking Probability for ATM Systems. *IEEE/ACM Trans. on Networking*, 1(2):230–235, 1993.
- [5] J. S. Sadowsky and J. A. Bucklew. On Large Deviation Theory and Asymptotically Efficient Monte Carlo Estimation. *IEEE Trans. Inform. Theory*, IT-36(3):579–588, May 1990.
- [6] A. Goyal, P. Shahabuddin, P. Heidelberger, V. F. Nicola, and P. W. Glynn. A Unified Framework for Simulating Markovian Models of Highly Dependable Systems. *IEEE Trans. Computers*, 41(1):36–51, Jan. 1992.
- [7] S. Parekh and J. Walrand. A Quick Simulation Method for Excessive Backlogs in Networks of Queues. *IEEE Trans. Automat. Contr.*, AC-34(1):54–66, Jan. 1989.
- [8] B. A. Bayes. Statistical Techniques for Simulation Models. *The Australian Computer Journal*, 2(4):180–184, November 1970.
- [9] M. Villén-Altamirano and J. Villén-Altamirano. RESTART: A Method for Accelerating Rare Event Simulations. In *Proc. 13th Int. Teletraffic Congress, ITC 13 (Queueing, Performance and Control in ATM)*, pages 71–76, Copenhagen, Denmark, June 1991.
- [10] M. Villén-Altamirano *et al.* Enhancement of the Accelerated Simulation Method RESTART by Considering Multiple Thresholds. In *Proc. 14th Int. Teletraffic Congress, ITC 14*, volume 1a, pages 797–810, France, 1994.
- [11] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

A Appendix: Proof of Theorem 6

Let \mathcal{S}_i be the current subset. Let ξ_k denote the probability that the *current ticket*, Ω , equals k . According to the theorem, ξ_k is:

$$\xi_k = \text{Prob}\{\Omega = k\} = \frac{\mu_k - \mu_{k-1}}{\mu_i}, \quad k = 1, \dots, i. \quad (15)$$

Analogous to the 2-subset case, let D'_{ij} , $i < j$ denote the event that the system would make a transition from \mathcal{S}_i to \mathcal{S}_j under the original coupling matrix \mathbf{P}_{ij} . In the modified system, when such a D'_{ij} event occurs, the $\mathcal{S}_i \rightarrow \mathcal{S}_j$ transition is granted only if the current ticket is equal or less than j , which occurs with probability $\sum_{k=1}^j \xi_k = \mu_j / \mu_i$. If $\Omega > j$, the system does not make the transition to \mathcal{S}_j , instead, it immediately goes to one of the subsets in group $\mathcal{G}_\Omega = \mathcal{S}_\Omega \cup \mathcal{S}_{\Omega+1} \cup \dots \cup \mathcal{S}_m$, i.e. in such a time instant the system behaves according to the 2-subset case, where \mathcal{G}_Ω and $\mathcal{G}_\Omega = \mathcal{S} \setminus \mathcal{G}_\Omega$ form the two “subsets”. In other words, the probability mass $(1 - \mu_j / \mu_i) \mathbf{P}_{ij}$ is redistributed to form transition probabilities for $\mathcal{S}_i \rightarrow \mathcal{G}_\Omega$ transitions. The redistributed mass, denoted by \mathbf{C}_{ij}^Ω , can be written as:

$$\mathbf{C}_{ij}^\Omega = \begin{pmatrix} \mathbf{0} & \dots & \mathbf{P}_{ij} & \dots & \mathbf{0} \end{pmatrix} \mathbf{N}_{\Omega-1} \begin{pmatrix} \mathbf{P}_{1\Omega} & \dots & \mathbf{P}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{\Omega-1,\Omega} & \dots & \mathbf{P}_{\Omega-1,m} \end{pmatrix} \quad (16)$$

where $\mathbf{N}_{\Omega-1}$ is the *fundamental matrix* for \mathcal{G}_Ω^C :

$$\mathbf{N}_{\Omega-1} = \left(\mathbf{I} - \begin{pmatrix} \mathbf{P}_{11} & \dots & \mathbf{P}_{1,\Omega-1} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{\Omega-1,1} & \dots & \mathbf{P}_{\Omega-1,\Omega-1} \end{pmatrix} \right)^{-1}.$$

The redistributed mass C_{ij}^Ω consists of a row of blocks

$$C_{ij}^\Omega = \left(C_{ij}^{\Omega,\Omega} \quad \dots \quad C_{ij}^{\Omega,k} \quad \dots \quad C_{ij}^{\Omega,m} \right),$$

where each $C_{ij}^{\Omega,k}$ block has the same size as the corresponding P_{ik} block in P . The block $C_{ij}^{\Omega,k}$, which is the contribution from all D'_{ij} events with ticket Ω to the $P_{ik}^{(m)}$ block in the modified system, is:

$$C_{ij}^{\Omega,k} = \left(\mathbf{0} \quad \dots \quad P_{ij} \quad \dots \quad \mathbf{0} \right) N_{\Omega-1} \begin{pmatrix} P_{1k} \\ \vdots \\ P_{\Omega-1,k} \end{pmatrix}. \quad (17)$$

Under a D'_{ij} event, tickets $\Omega = 1, \dots, j$ contribute to block $P_{ij}^{(m)}$ by a total amount of $(\mu_i/\mu_j)P_{ij}$, while tickets $\Omega = j+1, \dots, i$ contribute to each of the blocks $P_{ik}^{(m)} : k = \Omega, \dots, m$ accordingly to the respective ticket probabilities ξ_Ω . Taking all tickets into consideration, the total contribution from a D'_{ij} event to the $P_{ik}^{(m)}$ block for $k = 1, \dots, m$, denoted by $M_{ik}(D'_{ij})$, becomes:

$$M_{ik}(D'_{ij}) = \begin{cases} \mathbf{0} & \text{if } k < j, \\ \frac{\mu_j}{\mu_i} P_{ij} & \text{if } k = j, \\ \sum_{\Omega=j+1}^k \xi_\Omega C_{ij}^{\Omega,k} & \text{if } k > j. \end{cases} \quad (18)$$

So far, we have discussed only those transitions that decreased the current subset index (*subset indicator*). Note that other transitions i.e. those that result in a non-decreasing subset indicator, are always granted due to the monotonous nature of the applied ticketing mechanism. In such cases, therefore, the mass of P_{ij} contributes totally to $P_{ij}^{(m)}$.

For a given subset i , blocks $P_{ik}^{(m)}, k = 1, \dots, m$ can be calculated by cumulating the respective contributions from each $\mathcal{S}_i \rightarrow \mathcal{S}_j, j = 1, \dots, m$ transitions, which becomes:

$$P_{ik}^{(m)} = \sum_{j=1}^{i-1} M_{ik}(D'_{ij}) + \mathbf{1}(i \leq k) P_{ik} \quad (19)$$

Using (18), (19) can be rewritten as:

$$P_{ik}^{(m)} = \begin{cases} P_{ik} + \sum_{j=1}^{i-1} M_{ik}(D'_{ij}) & \text{if } k \geq i, \\ \sum_{j=1}^{i-1} M_{ik}(D'_{ij}) & \text{if } k < i. \end{cases} \quad (20)$$

For $k \geq i$, the expression $\sum_{j=1}^{i-1} M_{ik}(D'_{ij})$ yields

$$\sum_{j=1}^{i-1} \sum_{\Omega=j+1}^k \xi_\Omega C_{ij}^{\Omega,k},$$

which can be rearranged to

$$\sum_{\Omega=2}^k \left(\xi_\Omega \sum_{j=1}^{\Omega-1} C_{ij}^{\Omega,k} \right). \quad (21)$$

Note, however, that

$$\begin{aligned} \sum_{j=1}^{\Omega-1} C_{ij}^{\Omega,k} &= (P_{i1} \dots P_{i,\Omega-1}) N_{\Omega-1} \begin{pmatrix} P_{1k} \\ \vdots \\ P_{\Omega-1,k} \end{pmatrix} = \\ &= D_{ik}^{(\Omega)}, \end{aligned} \tag{22}$$

which simplifies (21) to the form

$$\sum_{\Omega=2}^k \xi_{\Omega} D_{ik}^{(\Omega)}. \tag{23}$$

Following similar steps, for the $k < i$ case $\sum_{j=1}^{i-1} M_{ik}(D'_{ij})$ can be rewritten as

$$\begin{aligned} \sum_{j=1}^{i-1} M_{ik}(D'_{ij}) &= \frac{\mu_k}{\mu_i} P_{ik} + \sum_{j=1}^{k-1} \sum_{\Omega=j+1}^k \xi_{\Omega} C_{ij}^{\Omega,k} = \\ &= \frac{\mu_k}{\mu_i} P_{ik} + \sum_{\Omega=2}^k \left(\xi_{\Omega} \sum_{j=1}^{\Omega-1} C_{ij}^{\Omega,k} \right) = \\ &= \frac{\mu_k}{\mu_i} P_{ik} + \sum_{\Omega=2}^k \xi_{\Omega} D_{ik}^{(\Omega)}. \end{aligned} \tag{24}$$

Using (15) and substituting (23) and (24) into (20), then changing indices k and Ω to j and k , respectively, yields

$$P_{ij}^{(m)} = \begin{cases} P_{ij} + \frac{1}{\mu_i} \sum_{k=2}^i (\mu_k - \mu_{k-1}) D_{ij}^{(k)} & \text{if } i \leq j, \\ \frac{\mu_i}{\mu_i} P_{ij} + \frac{1}{\mu_i} \sum_{k=2}^j (\mu_k - \mu_{k-1}) D_{ij}^{(k)} & \text{if } i > j, \end{cases}$$

which is identical to equation (12). This proves our theorem. \square