Helsinki University of Technology
Department of Electrical and Communications Engineering
Laboratory of Telecommunications Technology

Pasi Lassila

Efficient Simulation Techniques for Multiservice Loss Systems

This thesis has been submitted for official examination for the degree of Licenciate of Technology in Espoo, Finland.

Supervisor Professor Jorma Virtamo
Instructor Ph.D. Samuli Aalto

Espoo January 18, 1999.

Preface

This thesis is a product of the research I've done during 1997–1998. The research was conducted within the COST 257 project funded by Nokia Telecommunications, Sonera and Tekes.

First of all, I want to take this opportunity to thank Prof. Jorma Virtamo for giving me the opportunity to start my postgraduate studies under his supervision. The report at hand here is one milestone on my struggle towards the Ph.D. degree and it would not have been possible without the expertise, helpfulness and patience of Prof. Jorma Virtamo. Also, special thanks go to Ph.D. Samuli Aalto for his invaluable advice and encouragement during the whole process of writing this thesis.

The staff of the laboratory (past and present) also deserve their share of the thanks. The friendly and relaxed atmosphere in the lab has helped in making the concept of work seem less "serious".

Last but not least, I want to thank my beloved Minna and my family for their continuous support and love.

Helsinki, January, 1999

Pasi Lassila

Abstract

Modern broadband networks have been designed to integrate several service types into the same network. On the call scale, the process describing the number of calls present in the network can be modeled by a loss system. In principle, loss systems are mathematically simple and well understood, and one is able to write down exact expressions for such things as the blocking probability of a call belonging to a given class. However, for systems of realistic size in terms of the number and the capacity of the links and the number of traffic classes, such analytical expressions defy a direct evaluation because of the huge size of the state space.

In such situations, one has to resort to simulations in order to obtain estimates of the performance measures of interest. In this thesis we are specifically dealing with identifying efficient simulation methods for estimating the blocking probabilities. To this end, we first consider some basic simulation methods and we show how they are applied for generating samples in the simulation of loss systems. In particular, from the class of static Monte Carlo methods we present a rejection sampling method for generating independent samples and the Gibbs sampler for producing dependent samples. From the class of process simulation methods, we give a number different Markov chain methods. Also, we compare each method with respect to their efficiency in terms of the variance of the estimators and the computational effort associated with the methods.

Then we address the problem of increasing the efficiency of the simulations by using so called variance reduction techniques. In particular, we present two different methods for estimating the blocking probabilities in a multiservice loss system. The first method is an importance sampling method and utilizes some large deviation results for multidimensional random variables. We derive a composite sampling distribution, which is a weighted combination of distributions for effectively sampling the blocking states associated with each link in the network. We also provide heuristics to fix the weights in the composite distribution. This method is shown to be especially useful when estimating small blocking probabilities.

The second method is based on another known method called conditional expectation method, where the idea is to utilize known analytical results to the maximum degree. The method is based on conditioning on the samples hitting certain one-dimensional subsets of the state space for which conditional expectations can be calculated analytically. Moreover these expectations can be precomputed and, hence, the utilization of the method does not cause much extra computational work. In effect, it eliminates the internal variance within each subset. The method is also independent of the method used for generating samples. The numerical results confirm the efficiency of the method.

Contents

Chapter 1

Introduction

1.1 Multiservice Networks

The traditional telephone network offers to its users (or subscribers) a single service - the possibility to make a telephone call. For this call the network reserves a single bandwidth unit (actually a time slot) from each of the links along the path that the call is routed, thus forming a connection through the network. A multiservice network offers exactly the same kind of service to its users, but with the additional possibility that the user can select the *service type* it requires from a predefined set of alternatives. The amount of bandwidth required for the call is then a parameter in the requested service type. Other parameters defining the service type could be a declaration of the delay- or loss sensitivity of the requested connection.

The network offers services with transfer capabilities for traditional voice calls, pure data transmission and video transmission. Furthermore, the last two of these require higher bitrates for their transmission than traditional telephone calls. Hence, a multiservice network is also a broadband network as opposed to the traditional telephone network, which is commonly referred to as a narrowband network.

ITU-T and other standardization bodies, e.g. ATM Forum, have adopted ATM (Asynchronous Transfer Mode) as the chosen technology for the realization of the BISDN (Broadband Integrated Services Network). The standardization of ATM began in the 80's and by now several aspects of the network functionality have been standardized. The standards include, for example, specifications for ATM node interfaces in both public and private networks (see e.g. [Onv97] for a comprehensive coverage of signaling in ATM networks), specifications for traffic- and network management etc.

ATM is a connection oriented technology, where a user can request connections from the network between two ATM addresses. The network then reserves, if possible, the requested resources from the ATM nodes along the selected route towards the destination. After the connection has been established, the data transfer between the two end points is carried out by segmenting the transmitted data into short fixed size *cells* (53 octets), and transporting the cells along the reserved route. The amount of resources a connection needs is defined by a traffic descriptor, which includes information about the required peak and mean rate of the connection, and the tolerable limits on cell loss, cell delay and cell delay variation (for more details on ATM technology, see e.g. [Dep95]).

Since we are assuming that resources are reserved for the connection along its route, this would imply that the underlying network is a circuit switched network, as opposed to a packet switched network, e.g. the Internet. However, lately there has been development in the Internet standardization towards introducing different services and resource reservation into the Internet world as well. The concept corresponding to a connection is then called a *flow* and the network is referred to as the ISI (Integrated Services Internet) [Wro97]. It is aimed at enhancing the capabilities of the Internet such that it could be used for transmitting video and voice streams with quality of service guarantees, instead of just relying on the "best effort" type of service of the traditional Internet.

1.2 Performance of Multiservice Networks

The applications, which the multiservice networks will cater for, have widely differing traffic characteristics. Voice or video applications produce bit streams with either constant or variable bitrate, but the applications require that the network can transfer the bit stream as such without introducing too much extra constant delay or delay variation. On the other hand, data transfer applications have an on/off-type behavior, where cells are either transmitted at full rate or not transmitted at all. They are also much more affected by errors in the content of the received bitstream, i.e. lost or corrupted cells/packets, than the extra delays caused by the queuing in the network nodes. Thus we can see that the network must be able to handle services with different and sometimes even conflicting requirements on the network. Furthermore, these requirements must be met at all times over the lifetime of the connections.

The design and performance analysis of ATM nodes for the multiservice networks would be an impossibly complex task, if the effects of the traffic streams should be modeled with a single model defined over all time scales. Luckily it is possible to separate the phenomena in different time scales from each other. In specific, it is assumed that the higher time scales are quasistatic with respect to the time scale under consideration, and that the lower time scales are stationary. The division into time scales was originally presented in [Hui88] and it allows us to decouple the analysis into three time scales:

- *call scale*: In this time scale we model the process of call arrivals and departures into/from the system. The call attempts and the call durations are assumed to have certain statistical characteristics and also the amount of resources a call requires is assumed to be a fixed number. Of particular importance in this time scale is the concept of *equivalent bandwidth* which allows us to derive the aforementioned resource requirement. This time scale corresponds to the traditional modeling of narrowband networks known as loss systems.

- *burst scale*: At the burst scale we look at the statistical behavior of a call during its lifetime. As mentioned before, many services will have a bursty nature where the source alternates between on and off states, thus introducing a correlation structure into the cell arrivals in an ATM buffer. However, research has shown that for some traffic, there does not even exist a natural length of a burst, since the traffic has been observed to produce similar traffic patterns across time scales ranging from milliseconds to hours. Thus, the traffic apparently has some kind of "fractal"-like behavior, and hence the term self-similar traffic.

- *cell scale*: Here we fix the number of calls and their burst-state and investigate the behavior of a buffer in an ATM node under these conditions. Then the cells arrive from any single source almost equally spaced into the buffer, but, since ATM is an asynchronous technology, there can be several cells arriving to an output buffer from different inputs of the node at the same time.

Models dealing with all these time scales are extensively covered, for example, in the final report of the COST 242 project [Rob96]. In this thesis, however, we will restrict ourselves to deal with the call scale model of a multiservice network and its performance measures.

1.3 Call Performance of Multiservice Networks

On the call scale the process describing the number of calls present in the network can be modeled by a loss system. Associated with each call is the route through the network and the bandwidth requirements on the links. When the call is offered but there is not enough bandwidth on all the links along the requested route, the call is blocked and lost. The specific quantities we are interested in are the blocking probabilities for each call.

This is a natural extension of the model for the traditional single service telephone network. The steady state distribution of the system has a well known product form. A problem with the exact solution is, however, that it requires the calculation of a so called normalization constant, which entails the calculation of a sum over the complete allowed state space of the system. Efficient recursive solutions for calculating the blocking probabilities exist only for the case of multiservice traffic offered to a single link [Kau81, Rob81] and some special topologies, e.g. the tree topology [Ros95]. In a realistic size general topology network with possibly hundreds of classes and high speed links, the state space rapidly becomes astronomical. As a result the exact calculation of the blocking probabilities becomes computationally prohibitively expensive (see e.g. [Lou94] for a modern complexity theoretic analysis on algorithms for calculating exactly blocking probabilities in a general loss system).

However, analytical approximations to the blocking probabilities have been derived. The single link case has been analyzed e.g. in [Gaz93, Lab92, Mit94]. In the network case, the analytical approaches are based either on using the so called reduced load approximation, which leads to a set of fixed point equations [Chu93, Kel86, Kel91b, Hun89, Mit95], or numerical techniques applied to the generating function of the link occupancies [Cho95a, Dow96, Sim97].

1.4 Simulation of Multiservice Loss Systems

An alternative to deriving approximations is to simulate the system to a desired level of accuracy given the constraints on the available computing power. The main emphasis of this thesis is on developing efficient simulation methods for the problem of estimating the blocking probabilities of a multiservice loss system. The performance of the methods will be analyzed with respect to the computational effort required to reach a given accuracy and variance.

Traditionally the simulation approaches have focused on either static Monte Carlo (MC) techniques or the Markov chain simulation techniques. The static MC can be used since the stationary distribution of the system is known and it is possible to generate samples directly from it and we do not need to simulate some stochastic process, e.g. a Markov chain, to generate samples with the desired distribution. Static MC has been extensively studied by Ross [Ros95, chap. 6]. Markov chain simulation methods include the regenerative method, developed by Crane and Iglehart [Cra75, Cra77], which has been lately used in the context of rare event simulation in loss networks by Heegaard [Hee97]. Another method for using a Markov chain for generating the samples is to use the so called Gibbs sampler [Tie94]. However, the method does not belong to the class of process simulation methods like the regenerative method but rather in the class of static Monte Carlo methods since it utilizes conditional distributions of the known stationary distribution. This method is well known in the field of e.g. image analysis and Bayesian statistics, and its application to multiservice loss systems has been presented in [Las98a].

The problem with the aforementioned methods is that they become computationally very intensive as the state space grows, i.e. the simulation becomes inefficient. Known methods for increasing the efficiency of simulation include the use of control variables, antithetic variates, the use of conditional expectations, importance sampling (see e.g. [Ham67, Law91] for surveys) and more lately the so called RESTART method [Vil91, Vil94]. Importance sampling has been used especially in the context of rare event simulation of queuing systems, where the theory of large deviations has helped in the derivation of importance sampling distributions having in some sense optimal performance, see e.g. [Hei95] for a survey or [Fra91, Par89, Sad90, Sad91] for individual results. In loss systems importance sampling has been studied by Ross [Ros95, chap 6], Heegaard [Hee97] and Mandjes [Man97]. All have used importance sampling effectively as the blocking events have become rarer.

The first contribution of this thesis, to be published in [Las99], is the derivation of an efficient IS distribution for estimating the blocking probabilities in the multiservice loss system. We limit ourselves to studying IS distributions which belong to the family of so called exponentially shifted distributions. Previously these have been studied e.g. by Ross in [Ros95, chap. 6] and Mandjes in [Man97]. Ross has presented heuristics which attempt to increase the likelihood of the blocking states while, at the same, trying to limit the likelihood of generating misses from the allowed state space, resulting in a rather conservative shift. Mandjes proposes the use of an importance sampling distribution which shifts the mean of the sampling distribution to match the most probable blocking state.

Our approach is based on using a similar technique, but we extend the approach with ideas suggested by the large deviation results obtained by Sadowsky et al. in [Sad90]. They have shown that sometimes, depending on the shape of the "interesting" set, it is not sufficient to use one shifted distribution to satisfy the conditions of asymptotical optimality for the IS distribution. Instead, one needs to use a composite distribution, which is a weighted combination of several exponentially shifted distributions. The set of blocking states has this kind of shape and a composite distribution is needed. However, the results in [Sad90] leave open the question about the choice of the weights in the composite distribution. We propose heuristics based on attempting to keep the observed variable, i.e. the likelihood ratio, as constant as possible in the set of the blocking states. The slight increase in computational complexity when compared with just using a single shifted distribution appears to be well justified by the gains in the variance reduction and accuracy obtained in our numerical experiments.

A second contribution, published in [Las98b], of this thesis is another variance reduction method for estimating e.g. the blocking probabilities, which is rather different from the ideas behind importance sampling. The rationale behind this method is the realization that sometimes the heart of the problem does not lie in the rarity of the interesting events, but rather in the sheer size of the state space of the system. Importance sampling methods do not necessarily help in weakening the effect of the state space explosion as the size of the system increases. The reason is that these methods affect the distribution from which the samples are generated, but the information is still collected on a state-per-state basis. In contrast, we will present a method that exploits the known analytical results of the system providing a way to more effectively collect information about the state space given the current sample state. The method is an application of a known variance reduction technique called the conditional expectation method, see e.g. [Law91] or [Rub98, p. 97], and it is based on conditioning on the samples hitting certain subsets of the state space for which conditional expectations can be calculated analytically. In effect it eliminates the internal variance within each subset. Furthermore, this method can be used without practically any increase in computational effort and it is independent of the method used for generating the samples.

1.5 Outline

This thesis is organized as follows. In chapter 2 we present the basic model of multiservice loss systems and then consider briefly its applicability for modeling the call scale behavior of ATM networks. Chapter 3 gives a short survey of the literature on analytical approximations on the blocking probabilities in certain asymptotic regimes. In chapter 4 we review some basic approaches for simulating loss systems and give some numerical and analytical results regarding the efficiency of these methods. In chapter 5 we look at different means to increase the efficiency of the simulation process. We first review general known techniques for achieving this. Then a more extensive literature survey is given on the specific problem of so called rare event simulation. Also, we review the literature available on the methods to increase the efficiency of simulating multiservice loss systems in specific. Finally we present the main contributions of this thesis — the two different variance reduction methods, one based on conditioning and the other on using importance sampling.

Chapter 2

The Multiservice Loss System

In this chapter we will first present the stochastic model of the system and its solution from which we can calculate the performance measures of interest. Then we consider the applicability of the system for modeling the call scale behavior of the ATM network.

2.1 The Basic Model for Loss Networks

Consider a network consisting of J links, indexed with $j = 1, \ldots, J$, each having a capacity of C_j resource units. The network supports K classes of calls. The calls from the K classes arrive according to independent Poisson processes with arrival rates $\lambda_k, k = 1, \ldots, K$. The call holding times can, however, have any distribution with a finite mean $1/\mu_k$, due to the so called insensitivity property (see e.g. [Ros95, p. 163]).

Associated with a class-k call, $k = 1, \ldots, K$, is an offered load $\rho_k = \lambda_k/\mu_k$ and a bandwidth requirement of $b_{j,k}$ units on link j. Note that $b_{j,k} = 0$ when class-k call does not use link j. Let the vector $\mathbf{b}_j = (b_{j,1}, \ldots, b_{j,K})$ denote the required bandwidths of the classes in the system on link j. Also, we assume that a call is always accepted if there is enough capacity left and that the blocked calls are cleared. The state of the system is described by the vector $\mathbf{x} = (x_1, \ldots, x_K)$, where element x_k is the number of class-k calls present in the network.

The set of allowed states $\mathcal{S}$ can be described as

$$\mathcal{S} = \{\mathbf{x} \mid \forall j : \mathbf{b}_j \cdot \mathbf{x} \leq C_j\},$$

where the scalar product is defined, as usual, as $\mathbf{b}_j \cdot \mathbf{x} = \sum_i b_{j,i} x_i$, resulting in a coordinate convex state space. Note that a set is coordinate convex if for any $\mathbf{x} \in \mathcal{S}$ and $\mathbf{y} \leq \mathbf{x}$ then $\mathbf{y} \in \mathcal{S}$. This also defines an admission policy for new calls known as complete sharing (CS) in the literature (see e.g. [Lab92, Ros95]).

This system has the well known product form stationary distribution

$$\pi(\mathbf{x}) = \frac{1}{G}\prod_{k=1}^{K}\frac{\rho_k^{x_k}}{x_k!} = \frac{1}{G}\prod_{k=1}^{K} f(x_k,\rho_k) = \frac{f(\mathbf{x})}{G}, \tag{2.1}$$

where $f(\mathbf{x}) = \prod_k \rho_k^{x_k}/x_k!$ denotes the unnormalized state probability,

$$f(x,\rho) = \frac{\rho^x}{x!},$$

and G is the so called normalization constant

$$G = \sum_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}).$$

The set of blocking states for a class-k call, $\mathcal{B}^k$, is

$$\mathcal{B}^k = \{\mathbf{x}\in\mathcal{S} \mid \exists j : \mathbf{b}_j\cdot(\mathbf{x}+\mathbf{e}_k) > C_j\},$$

where $\mathbf{e}_k$ is a K component vector with 1 in the k^{th} component and zeros elsewhere. Also, we will denote with $\mathcal{S}^k$ the subset of the state space where one more class-k call can be admitted, i.e.

$$\mathcal{S}^k = \mathcal{S}\setminus\mathcal{B}^k.$$

Then, let us denote by

$$G_k^B = \sum_{\mathbf{x}\in\mathcal{B}^k} f(\mathbf{x}),$$

the state sum over the blocking states for a class-k call, and by

$$G_k = \sum_{\mathbf{x}\in\mathcal{S}^k} f(\mathbf{x}),$$

the state sum over the admissible set for a class-k call. Note that G_k can be given an alternative expression in the following way. Let $\mathbf{b}^k = \{b_{1,k},\ldots,b_{J,k}\}$ denote the vector for the amount of bandwidth class k requires on each of the links in the network, i.e. it is the k^{th} column from the matrix $\mathbf{b}$. Note that this is not the same as $\mathbf{b}_j$ defined earlier, which is the j^{th} row of $\mathbf{b}$. Now by denoting with $G(\mathbf{C})$ the normalization constant G of a system with link capacity vector $\mathbf{C} = \{C_1,\ldots,C_J\}$, then

$$G_k = G(\mathbf{C}-\mathbf{b}^k),$$

i.e. a normalization constant where for each link j its capacity is diminished by $b_{j,k}$.

The blocking probability of a class-k call, B_k, can be expressed in the following forms

$$B_k = 1 - \frac{G_k}{G} = \frac{G_k^B}{G} = \sum_{\mathbf{x}\in\mathcal{B}^k}\pi(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{S}}\pi(\mathbf{x})1_{\mathbf{x}\in\mathcal{B}^k} = \mathrm{E}\left[1_{\mathbf{x}\in\mathcal{B}^k}\right]. \tag{2.2}$$

It should be noted that the distribution π given by (2.1) represents the truncation of a K dimensional independent Poisson type distribution to the state space $\mathcal{S}$. Then, by defining another state space $\tilde{\mathcal{S}}$ such that $\tilde{\mathcal{S}} \supseteq \mathcal{S}$ and a random vector $\tilde{\mathbf{X}} \in \tilde{\mathcal{S}}$ with a similar product form distribution, $\Pr[\tilde{\mathbf{X}} = \mathbf{x}] \sim f(\mathbf{x})$, we can express the blocking probability as

$$B_k = \mathrm{E}\left[1_{\tilde{\mathbf{X}} \in \mathcal{B}^k} \mid \tilde{\mathbf{X}} \in \mathcal{S}\right] = \frac{\mathrm{E}[1_{\tilde{\mathbf{X}} \in \mathcal{B}^k}]}{\mathrm{E}[1_{\tilde{\mathbf{X}} \in \mathcal{S}}]}. \tag{2.3}$$

The blocking probabilities also have the following so called elasticity property, originally obtained in [Vir88],

$$\frac{\partial B_j}{\partial \rho_i} = \frac{\partial B_i}{\partial \rho_j}.$$

As an example, consider the following network shown in Fig. 2.1 with the state space depicted in the figure on the right. The link capacities appear as linear constraints on the state space and the blocking states are the states on the boundary of the state space. In the figure, the blocking states for class 1 calls have been circled.

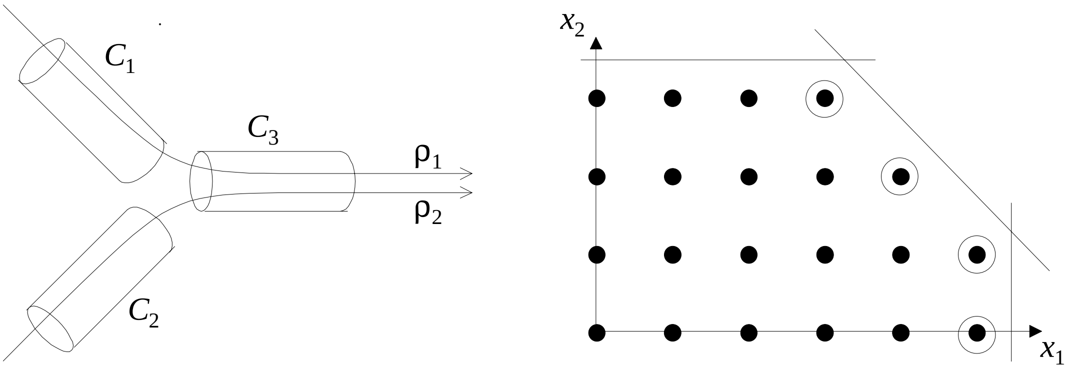

Figure 2.1: Example network and its state space.

2.2 Exact Computation of the Blocking Probabilities

As was noted earlier, the size of the state space of the system becomes intractable very rapidly as the size of the system increases. However, it is still possible to compute the exact blocking probabilities in cases where the number of links and/or the number of traffic classes is not prohibitively large.

The most direct way to perform the calculation would be to brute force perform the summations in (2.2). This approach has the advantage that the memory requirements are indeed very small for the algorithm, since one is essentially just collecting $K + 1$ state sums (K sums for the blocking states of each class and one sum for the normalization constant G itself). The drawback of this approach is that the number of traffic classes in the network easily becomes quite large as the number of nodes in the network increases leading to an excessively large state space.

Now consider a situation where the number of traffic classes is greater than the number of links in the network. Then the state space problem can be alleviated somewhat by changing the state variable of the system from the number of class-k calls present in the network to the number of circuits occupied on each link.

For this let $\mathbf{Y} = \{Y_1, \ldots, Y_J\}$ denote the number of occupied circuits on each of the links. The state space of $\mathbf{Y}$, $\mathcal{Y}$, is the Cartesian product space

$$\mathcal{Y} = \{0, \ldots, C_1\} \times \cdots \times \{0, \ldots, C_J\}.$$

Then we can obtain the distribution of $\mathbf{Y}$ by successively convolving the occupancy distribution caused by each individual traffic class with each other. This so called convolution method was presented in [Ive87], but it was originally presented for the single link case. We will show here how it is generalized to the computation of the joint distribution of several links. Note that once we have the distribution of $\mathbf{Y}$ the blocking probability is obtained simply from

$$B_k = 1 - \sum_{y_1=0}^{C_1-b_{1,k}} \cdots \sum_{y_J=0}^{C_J-b_{J,k}} \Pr\left[\mathbf{Y} = \mathbf{y}\right]. \tag{2.4}$$

To derive a recursive formula for the convolution, we let $\mathbf{Y}_k$ denote the link occupancy vector associated with the traffic class k, i.e. $\mathbf{Y}_k = X_k\, \mathbf{b}^k$. Here X_k is the r.v. for the number of class-k calls in the network. Now we have that

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{Y}_k.$$

Also, we denote by $\mathbf{Y}^{(l)}$ the partial sum of $\mathbf{Y}_k$:s up to l classes, i.e. $\mathbf{Y}^{(l)} = \sum_{k=1}^{l} \mathbf{Y}_k$. Then the following recursion holds:

$$\mathbf{Y}^{(l)} = \sum_{k=1}^{l} \mathbf{Y}_k = \mathbf{Y}^{(l-1)} + \mathbf{Y}_l = \mathbf{Y}^{(l-1)} + X_l \mathbf{b}^l,$$

from which it follows by conditioning on the number of class-l calls that

$$\Pr\left[\mathbf{Y}^{(l)} = \mathbf{y}\right] = \sum_{n=0}^{\min\{\mathbf{y}/\mathbf{b}^l\}} \Pr\left[\mathbf{Y}^{(l-1)} = \mathbf{y} - n\mathbf{b}^l\right] \Pr\left[X_l = n\right], \tag{2.5}$$

where $\mathbf{y}/\mathbf{b}^l$ refers to the componentwise division of the two equal length vectors $\mathbf{y}$ and $\mathbf{b}^l$. In (2.5) the upper limit on the index n is derived from the requirement that $\mathbf{y} - n\mathbf{b}^l \geq 0$. Also, note that the distribution for X_l is just the truncated Poisson distribution truncated at the maximum number of allowed class-l calls.

This gives us directly a computational algorithm for computing the occupancy distribution. However, the resulting convolution algorithm gives us the relative values of the distribution for all the states. From (2.5) we can recursively obtain the unnormalized occupancy distribution $q(\mathbf{y})$ by computing

$$q^{(l)}(\mathbf{y}) = \sum_{n=0}^{\min\{\mathbf{y}/\mathbf{b}^l\}} q^{(l-1)}(\mathbf{y} - n\mathbf{b}^l) f(n, \rho_l), \tag{2.6}$$

with the initial conditions

$$\begin{cases} q^{(0)}(0) = 1. \\ q^{(0)}(\mathbf{y} \neq 0) = 0. \end{cases}$$

Finally, the occupancy probabilities are obtained by normalization

$$\Pr[\mathbf{Y} = \mathbf{y}] = \frac{q(\mathbf{y})}{\sum_{\mathbf{y} \in \mathcal{Y}} q(\mathbf{y})}.$$

Thus, the occupancy distribution is computed by evaluating for a fixed l the recursion (2.6) for all the states resulting in $q^{(l)}(\cdot)$, which gives the unnormalized joint occupancy of all the classes from 1 up to l. The recursion ends when $l = K$, i.e. when all the classes have been convolved and the final step is the normalization. Then the blocking probabilities are obtained from (2.4). Note that this method requires that the whole occupancy distribution must be kept in memory giving a much higher memory requirement for the algorithm than for direct summation. Secondly, during the course of the convolutions, we must go through the entire state space $\mathcal{Y}$ altogether K times (in fact, $K+1$ times including the computation of the normalization constant). Hence, it is quite obvious that also this method becomes intractable as the number of links in the network grows.

2.3 Call Scale Model for Multiservice Networks

The model described above is also applicable almost directly for modeling the call scale process of ATM networks. The only thing that we need to examine closer in the model is the method to calculate the current link occupancies $\sum_{k=1}^{K} b_{j,k} x_k$ for all $j = 1, \ldots, J$, which is required for determining whether a state belongs to the set of allowed states $\mathcal{S}$ or not. In ATM this so called admission region is potentially much more difficult to obtain, because the bandwidth required by a given number of calls can be a complex function depending on all traffic classes and the number of calls of each class.

2.3.1 ATM and Statistical Multiplexing

In ATM the problem of evaluating the required bandwidth given the current state is caused by the network's ability to use *statistical multiplexing*. This is possible because it is anticipated that a large portion of the traffic in the network will have a bursty nature, i.e.

the traffic from a source will consist of activity periods (possibly of varying intensity) and silence periods. Furthermore, the quality of service (QoS) guarantees given by the network for the services are probabilistic in nature, e.g. the cell loss probability (*CLP*) is guaranteed to be less than say 10^{-9}. In statistical multiplexing the sum of the peak rates of the calls occupying a link is allowed to be greater than the link capacity C_j, thus causing some random cell loss, subject to the constraint that the QoS guarantees of all calls have to be met.

Obviously, the cell loss process in the buffers of the ATM nodes is affected by the size of the buffers of the switch. This also results in two types of multiplexing modes (using the terminology of [Rob96]):

- rate envelope multiplexing,
- rate sharing multiplexing.

In rate envelope multiplexing we assume that the buffer of the link is small and capable of only handling the cell scale variations. Consequently, in the multiplexing model the link is assumed to be bufferless and the resulting models have been shown to be relatively insensitive to the assumptions made about the stochastic nature of the rate process. Fig. 2.2 illustrates this. In the figure the rate process $R(t)$ is the instantaneous rate of traffic entering the output buffer, which has service rate equal to the link capacity C_j. In statistical multiplexing $R(t)$ is now allowed to temporarily exceed C_j (e.g. for time T_{loss} in the figure). However, the cell loss probability in this bufferless case, defined as

$$CLP = \frac{\mathrm{E}[(R - C_j)^+]}{\mathrm{E}[R]},$$

must still be below the QoS requirements of the connections.

In rate sharing multiplexing, we assume that the output buffer is capable of storing large amounts of cells with the hope that we can let the rate process exceed the link capacity more often or for longer periods of time, thus achieving a better average link utilization level. However, research has shown that in these so called burst multiplexing models the buffers have to be large in order to gain significant increases in link utilization and, more importantly, the results are very sensitive to the model assumptions about the stochastic nature of the rate process. For more details see e.g. [Rob96].

2.3.2 Link Usage Models for ATM

Irrespective of the nature of the ATM nodes with respect to their buffer sizes, from the point of view of our system model the link utilization model has to be able to answer the following question: How much bandwidth is needed to fulfill the QoS requirements given the current number of calls on the link? The question can be answered by using the following methods:

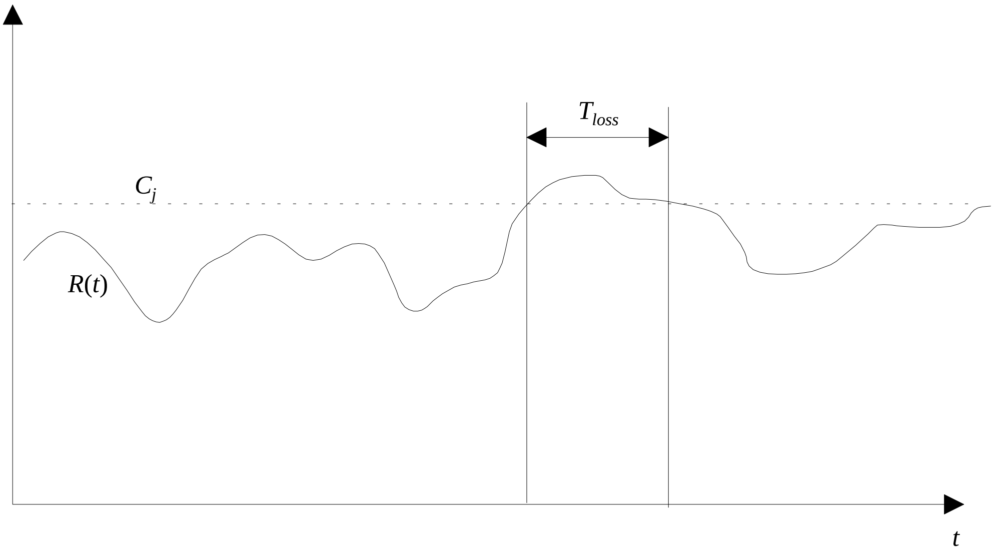

Figure 2.2: Statistical multiplexing in ATM networks.

- exact calculation of the required bandwidth given the QoS constraints and the number of calls of each class,
- a nonlinear approximation to the exact solution,
- a linear approximation using the equivalent bandwidth of a class (see e.g. [Hui88, Kel91a] for an explanation of the concept),
- peak rate allocation (no statistical multiplexing).

These methods will result in different admission regions for the calls. Fig. 2.3 shows the effect of the choice of the bandwidth allocation method on the admission region for a simple case with two traffic classes. The admission region for the exact solution or a nonlinear approximation is concave and rather difficult to obtain. The equivalent bandwidth method is a linear approximation to the concave exact admission region, but as it is a linear approximation, it will not include all allowable states and is thus less efficient from the point of view of the link usage. However, it is still more efficient than just using the peak rates for determining the bandwidth.

In the first case (peak allocation) and the second case (equivalent bandwidth) the link utilization is calculated as previously with the $b_{j,k}$ on each link set equal to the peak rate of the requested call or its equivalent bandwidth, respectively. However, when using the exact solution or any nonlinear approximation, the link usage, given the number calls of each class on the link (state) and their QoS requirements, becomes a function of the given state. Then the set of allowed states is given by

$$\mathcal{S} = \{\mathbf{x} \mid \forall j : \beta_j(\mathbf{x}) \leq C_j \ \} \, ,$$

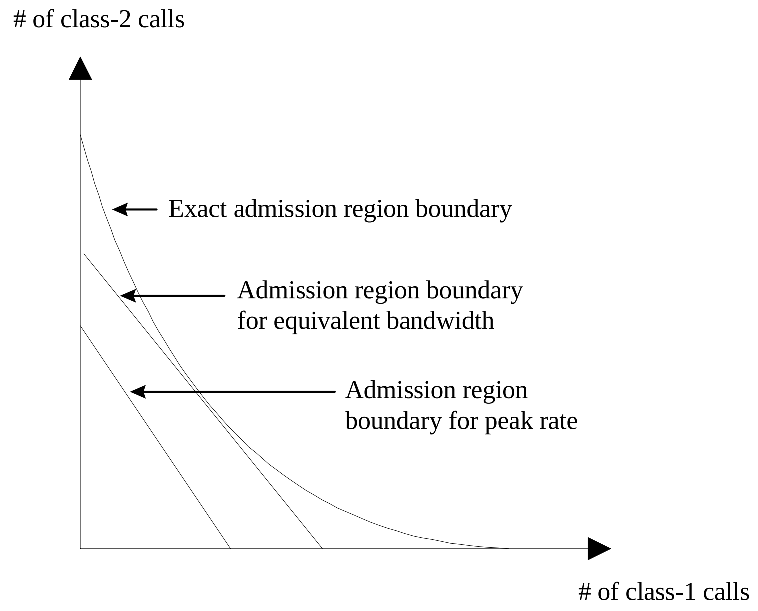

Figure 2.3: Typical admission region for 2 traffic classes.

where $\beta_j(\cdot)$ is some nonlinear function whose value gives for the given argument the required bandwidth to fulfill the QoS requirements on link j. Note that this implies that embedded in the function $\beta_j(\cdot)$ is also the knowledge of whether a class uses link j or not. Typically it is not advantageous to let all different services compete for the complete bandwidth because to determine the $\beta(\cdot)$ function for the whole traffic mix would be difficult. Also, in the literature on connection admission control in ATM networks the models usually assume that the $\beta_j(\cdot)$ function calculates the required bandwidth with respect to one QoS parameter, usually the *CLP*, and a single value for it. Then if the services have different requirements for example on their cell loss probabilities, the resource reservation would always have to be done according to the most demanding requirements. This would be inefficient if the differences in the requirements between the services are several orders of magnitude.

In practice it is most often assumed that the capacity of the link has been partitioned so that services with similar QoS characteristics have been given a share of the total bandwidth. If the ATM nodes only have small buffers for cell scale variations this bandwidth partitioning does not require any extra functionality in the buffers to support it. However, ATM nodes with large buffers require that the buffers have to support the partitioning of the bandwidth through suitable scheduling mechanisms, e.g. weighted round-robin or PGPS (WFQ) [Par93]. If all the services are separated and all the different services share the available bandwidth dynamically (i.e. the partition can be changed with each call acceptance) the set of allowed states can be described as

$$\mathcal{S} = \left\{ \mathbf{x} \mid \forall j : \sum_{k=1}^{K} \beta_{j,k}(x_k) \leq C_j \right\},$$

where we have a $\beta_{j,k}(x_k)$ function depending only on the value of the component x_k for each of the services.

As a conclusion, we notice that the call scale process of the ATM network can be described by a multiservice loss system. However, the method for evaluating the allowed state space in the ATM context can be more difficult. Thus the detailed effects of having an ATM network only affects the shape of the allowed state space.

Chapter 3

Analytical Approximations

In this chapter we review some of the approximation results available in the literature for the blocking probabilities in multiservice loss systems.

3.1 Introduction

The multiservice loss system and its product form solution are mathematically well understood but, as was mentioned earlier, the exact calculation of the blocking probabilities is not feasible due to the state space explosion except for very small networks or some special topologies. Then one alternative to obtain estimates of the blocking probabilities is by deriving analytical approximations. In the literature, the articles basically fall into two categories: ones with approximations for a single link and ones with network approximations. Usually, the novel mathematical approaches have been first applied in the simpler single link case and later applied in the network case.

In the derivation of the approximations it is a standard practice to assume that the links in the network are either in *light, critical* or *heavy* traffic conditions, which are defined as follows: The traffic on link j is said to be

- light if $\sum_{k=1}^{K} b_{j,k}\rho_k < C_j$.
- critical if $\sum_{k=1}^{K} b_{j,k}\rho_k = C_j$.
- heavy if $\sum_{k=1}^{K} b_{j,k}\rho_k > C_j$.

Furthermore, the results are obtained using a suitable scaling of the system corresponding to asymptotical behavior in a large network setting. Formally, we let $\rho_k \to \infty$ and $C_j \to \infty$, but we require that ρ_k/C_j remains bounded.

3.2 Single Link Approximations

In this subsection we will make a slight alteration to the general notation since we will be covering only single link models here as opposed to network models. Hence, we will omit the link dependence from the notation and simply denote with C the capacity of the link, b_k is then the required bandwidth of a class-k call and its offered load is denoted, as usual, by ρ_k. The quantities G, G_k and G_k^B, i.e. the different state sums, are defined analogously with only a single link constraint.

In an early work [Lab92] Labourdette and Hart apply linear system theory to the Kaufman-Roberts recursion for the single link multiservice loss system. The system is transformed into an equivalent discrete time linear system presentation and they are able to establish that under the scaling as described before the aggregate state probabilities (link occupancies) of the link have a product form with respect to the probability of the link being fully occupied, i.e.

$$\frac{\Pr[\sum_{k=1}^{K} b_k x_k = n]}{\Pr[\sum_{k=1}^{K} b_k x_k = C]} = \frac{P_n}{P_C} = \alpha^{C-n},$$

where α is the unique positive root of

$$\sum_{k=1}^{K} b_k \rho_k z^{b_k} = C, \tag{3.1}$$

which is obtained from the eigenvalue problem associated with the linear system form of the problem. The values of α also reflect the loading of the system, i.e. $\alpha < 1$ corresponds to the heavy traffic, $\alpha = 1$ to the critical traffic and $\alpha > 1$ to the light traffic case. With respect to P_C the blocking probabilities can then be expressed as

$$\begin{aligned} B_k &= \sum_{n=C-b_k+1}^{C} P_n \\ &\approx P_C(1 + \alpha + \ldots + \alpha^{b_k - 1}) \end{aligned} \tag{3.2}$$

for all k. The authors derive approximations to P_C by considering an equivalent system with the lowest dimension, i.e. a system with only one traffic class, for which the blocking probability would be given simply by the classical Erlang-B formula $\mathrm{erl}(n, \rho)$, where n is the size of the system and ρ the offered traffic. This is done by matching the parameters of the characteristic polynomial of the equivalent system to the parameters of the original system. As a result they derive that P_C can be approximated by

$$P_C = \begin{cases} \dfrac{1-\alpha}{1-C/m}\mathrm{erl}\left(\dfrac{m}{d}, \dfrac{C}{d}\right), & \text{if } \alpha \neq 1, \\ \dfrac{m}{\sigma^2}\mathrm{erl}\left(\dfrac{m}{d}, \dfrac{C}{d}\right), & \text{if } \alpha = 1, \end{cases}$$

where $m = \sum_{k=1}^{K} b_k \rho_k$ and $\sigma^2 = \sum_{k=1}^{K} b_k^2 \rho_k$ are the mean and variance of the link occupancy when $C \to \infty$ (sum of K Poisson variables), and d is a constant given by

$$d = \frac{\log(C) - \log(m)}{\log(\alpha)}.$$

The approximation (3.2) can be made even simpler when $|\alpha| \leq 1$ (heavy and critical load) and the authors are able to establish the result that the blocking occurs as if the b_k circuits were reserved sequentially and independently giving a new approximation

$$B_k \approx \begin{cases} 1 - \alpha^{b_k}, & \text{if } \alpha < 1. \\ 0, & \text{if } \alpha = 1. \end{cases}$$

However, the authors were not able to give any results on the magnitude of the error terms in the different loading regimes for their approximations due to the assumptions made about the linear system form of the problem.

In a slightly more recent paper [Gaz93] Gazdicki, Lambadaris and Mazumdar derive approximations for all the loading regimes using techniques similar to what are used in large deviation methods (see e.g. [Buc90a] or [Sch95] for details on large deviation theory). Again the analysis is based on considering the scaled system, where we denote by $\rho_k(n) = n\rho_k, C(n) = nC, G(n), G_k^B(n)$ and $B_k(n)$ the corresponding parameters in the scaled system and then we let $n \to \infty$. Also, we assume here that the greatest common divisor of $b_k, k = 1 \ldots, K$, is 1. The problem is transformed into an "unconstrained" problem where the capacity is assumed infinite and the resulting stochastic process will have a multi-dimensional Poisson distribution. Let us first define the aggregate state sets (link occupancy levels) in the scaled system as

$$\mathcal{Y}_m = \left\{ \mathbf{x}, x_k \in \mathbb{N} : \sum_{k=1}^{K} b_k x_k = m \right\}.$$

Then we define the variables in the unconstrained system as

$$\begin{aligned} H(n) &= e^{-n(\rho_1 + \ldots + \rho_K)}\ G(n), \\ H_k^B(n) &= e^{-n(\rho_1 + \ldots + \rho_K)}\ G_k^B(n), \end{aligned}$$

and

$$F_m(n) = e^{-n(\rho_1 + \ldots + \rho_K)} \sum_{\mathbf{x} \in \mathcal{Y}_m} \prod_{k=1}^{K} \frac{\rho_k(n)^{x_k}}{x_k!}.$$

Now, the blocking probabilities can be expressed as

$$B_k(n) = \frac{H_k^B(n)}{H(n)} = \frac{\sum_{m=C(n)-b_k+1}^{C(n)} F_m(n)}{H(n)}.$$

By the properties of the Poisson random variables it is not difficult to show that

$$F_m(n) = \Pr\left[\sum_{i=1}^{n} \eta_i = m \right]$$

and

$$H(n) = \Pr\left[\sum_{i=1}^{n} \eta_i \leq nC\right],$$

where η_i are independent and identically distributed (i.i.d.) random variables defined as $\eta_i \sim \sum_{k=1}^{K} b_k x_k$ and x_k is a Poisson random variable with parameter ρ_k.

By using an exponential change of measure technique and a local limit theorem for sums of i.i.d. random variables on a lattice the authors derive an approximation to $F_m(n)$. Then, recalling that $H_k^B(n) = \sum_{m=nC-b_k+1}^{nC} F_m(n)$, the authors derive that

$$H_k^B(n) = e^{-nI(C)} \frac{1}{\sqrt{2\pi n}\sigma} \left(\frac{1 - e^{\theta_C b_k}}{1 - e^{\theta_C}}\right) [1 + o(1)], \tag{3.3}$$

where $I(C) = C\theta_C - \sum_{k=1}^{K} \rho_k (e^{\theta_C b_k} - 1)$ is the so called rate function for η_i and σ is

$$\sigma = \sqrt{\sum_{k=1}^{K} b_k^2 \rho_k e^{\theta_C b_k}}, \tag{3.4}$$

and θ_C is obtained as a solution to the equation (cf. eq. 3.1)

$$\sum_{k=1}^{K} b_k \rho_k e^{\theta_C b_k} = C.$$

In the critical traffic case, due to the properties of the rate function, we have $I(C) = 0$ and $\theta_C = 0$ and we obtain

$$H_k^B(n) = \frac{b_k}{\sqrt{2\pi n}\sigma} [1 + o(1)]. \tag{3.5}$$

What remains, is to find the asymptotic behavior of $H(n)$. For the light and critical traffic cases they can be derived rather straightforwardly, by the use of Chernoff bound and the central limit theorem, to be

$$\lim_{n\to\infty} H(n) = \begin{cases} 1, & \text{if } \sum_{k=1}^{K} b_k \rho_k < C, \\ 1/2, & \text{if } \sum_{k=1}^{K} b_k \rho_k = C. \end{cases} \tag{3.6}$$

However, for the heavy traffic case, the asymptotics are obtained by the use of the Bahadur-Rao theorem for sums of i.i.d. random variables, which gives a more accurate large deviation approximation than the asymptotic rate given by Cramer's theorem. Then, in the heavy traffic case we have

$$\lim_{n\to\infty} H(n) = e^{-nI(C)} \frac{1}{(1 - e^{\theta_C})\sqrt{2\pi n}\sigma} [1 + o(1)], \quad \text{if } \sum_{k=1}^{K} b_k \rho_k > C. \tag{3.7}$$

Hence from equations (3.3-3.7) we get the following asymptotics for the blocking probabilities in the different loading regimes

$$B_k(n) = \begin{cases} e^{-nI(C)} \dfrac{1}{\sqrt{2\pi n \sigma}} \left(\dfrac{1 - e^{\theta_C b_k}}{1 - e^{\theta_C}} \right) [1 + o(1)]\,, & \text{if } \sum_{k=1}^{K} b_k \rho_k < C, \\ \sqrt{\dfrac{2}{\pi n}} \dfrac{b_k}{\sigma} [1 + o(1)]\,, & \text{if } \sum_{k=1}^{K} b_k \rho_k = C, \\ \left(1 - e^{\theta_C b_k}\right) [1 + o(1)]\,, & \text{if } \sum_{k=1}^{K} b_k \rho_k > C. \end{cases}$$

In the paper itself the results were derived for a general lattice type distribution where the greatest common divisor for b_k is greater than 1.

In [Mit94] Mitra and Morrison use another method, which also allows the derivation of approximations for all loading regimes with rigorous treatment of the error terms. The article considers a finite source loss model applicable e.g. for modeling burst blocking probabilities, where each source sharing the buffer is either in a burst mode or silent mode according to some distribution. The infinite source model, corresponding to the multiservice loss system for a single link, is then obtained as a limiting case of the finite source model by letting the number of sources tend to infinity and letting the probability of being in the burst mode approach 0. However, the limits are approached in such a way that their product will still be finite corresponding to the offered traffic for class-k call in the scaled system.

The approach is based on using numerical methods for calculating normalization constants of a multivariate Poisson distribution using an integral presentation. To be specific, for the normalization constant G of a link with capacity C, the following integral representation holds

$$G = \frac{1}{2\pi i} \oint_{|z|<1} \frac{F(z)}{(1-z) z^{C+1}}\, dz,$$

where $F(z)$ denotes the generating function for the link occupancy in the infinite capacity case

$$F(z) = \exp \left[\sum_{k=1}^{K} \rho_k (z^{b_k} - 1) \right].$$

Then let us define

$$\begin{aligned} f(z) &= \frac{1}{C} (\log F(z) - C \log z) \\ &= \sum_{k=1}^{K} \frac{\rho_k}{C} (z^{b_k} - 1) - \log z. \end{aligned}$$

By the fact that the blocking probabilities can be expressed as $B_k = 1 - G_k/G$, we have the following expression for the blocking probabilities

$$B_k = \frac{1}{2\pi i G} \oint_{|z|<1} \frac{(1 - z^{b_k}) e^{Cf(z)}}{z(1-z)}\, dz, \tag{3.8}$$

where G is given by

$$G = \frac{1}{2\pi i} \oint_{|z|<1} \frac{e^{Cf(z)}}{z(1-z)}\, dz. \tag{3.9}$$

Using the saddle point method the authors derive asymptotic approximations to equations (3.8) and (3.9). The unique positive saddle point z^* is obtained as a solution to the equation

$$\sum_{k=1}^{K} \frac{\rho_k}{C} b_k (z^*)^{b_k} = 1.$$

The case when the saddle point is close to the pole $z^* = 1$ of the integrands in (3.8) and (3.9) happens when $\sum_{k=1}^{K} b_k \rho_k \approx C$, i.e. near the critical traffic case. In order to be able to handle this, the authors use an asymptotic approximation due to Bleistein [Ble66] to derive the following uniform approximation for the blocking probabilities

$$B_k = \frac{e^{Cf(z^*)}\left[1-(z^*)^{b_k}\right]}{\sqrt{2\pi C v(z^*)}(1-z^*)A}\left[1 + O\left(\frac{1}{C}\right)\right],$$

where A is a constant and

$$v(z^*) = \sum_{k=1}^{K} b_k^2 \frac{\rho_k}{C} (z^*)^{b_k}.$$

This approximation can be further developed in the light, critical and heavy traffic cases to get the result

$$B_k = \begin{cases} \left[1-(z^*)^{b_k}\right]\left[1+O\left(\frac{1}{C}\right)\right], & \text{if } \sum_{k=1}^{K} b_k\rho_k > C, \\ \dfrac{A_k'}{\sqrt{C}}\left[1+O\left(\frac{1}{\sqrt{C}}\right)\right], & \text{if } \sum_{k=1}^{K} b_k\rho_k = C, \\ A_k''\dfrac{e^{Cf(z^*)}}{\sqrt{C}}\left[1+O\left(\frac{1}{C}\right)\right], & \text{if } \sum_{k=1}^{K} b_k\rho_k < C, \end{cases}$$

where A_k' and A_k'' are constants. In these results the scaling parameter should be taken as C (instead of n as previously), and, with this in mind, the results are seen to be consistent with the results obtained in [Gaz93].

Earlier results for the critical load case have been obtained by Evans [Eva91] and Reiman [Rei91]. Also, the qualitative behavior in all loading regimes was studied before by Simonian [Sim92] using a saddle point technique on the generating functions.

3.3 Network Approximations

Reduced Load Approximations

The result in the previous section for the heavy traffic case was obtained in an early work by Kelly in [Kel86]. The paper considers a general loss network with static routing and Kelly showed that in the heavy traffic case and under the scaling as before the blocking probabilities of a class-k call (a class consists now of a route and a bandwidth requirement on each of the links on the route) have the following approximation

$$B_k = 1 - \prod_{j=1}^{J} (1 - L_j)^{b_{j,k}}, \tag{3.10}$$

where $L_j \in [0, 1)$ is a parameter obtained from a constrained non-linear optimization problem. The result has the interpretation that blockings seem to happen independently from link to link and, furthermore, that on each link j the $b_{j,k}$ circuits are reserved independently and sequentially with a blocking probability of L_j for each circuit. As a consequence, we can say that the number of free circuits on a link has a geometric distribution with parameter $(1 - L_j)$.

Kelly goes further in developing the approximations and establishes a reduced load approximation leading to a set of fixed point equations whose solution can be obtained e.g. by using repeated substitutions. The motivational arguments are as follows, citing Kelly: "If a request for a circuit on link j is blocked with probability L_j, and if we make the assumption that all such blocking events are independent, then the traffic offered to link j will be Poisson and the level of carried traffic will be $\sum_k b_{j,k} \rho_k \prod_{i \neq j} (1 - L_i)^{b_{i,k}}$." Then we should require that the blocking probability on link j should be consistent with that level of carried traffic. As a result we get the set of fixed point equations

$$E_j = \mathrm{erl} \left(\sum_k b_{j,k} \rho_k \prod_{i \neq j} (1 - E_i)^{b_{i,k}}, C_j \right), \quad j = 1, \ldots, J, \tag{3.11}$$

where $\mathrm{erl}(\cdot, \cdot)$ denotes, again, the Erlang-B formula. Kelly shows that the equations defined by (3.11) converge to a unique solution and that under the asymptotic scaling this solution is the same as obtained from (3.10), i.e. $E_j \to L_j$.

In a more recent paper Kelly and Hunt [Hun89] discuss the effects of having links under critical traffic in the network on the approximations (3.10) and (3.11). They first establish a central limit theorem for the distribution of idle circuits on links with critical traffic. Then they show, by considering a special case with all links under critical traffic, that it is possible to choose the L_j in (3.10) such that the error is of smaller order than $\sqrt{n}$, where n is the scaling parameter. However, it turns out that the fixed point approximation (3.11) does not hold anymore in this case, due to the dependencies between the occupancies of the links under critical traffic. Therefore, the approximation (3.11) becomes more inaccurate as the loading of the network reduces.

In [Chu93] Chung and Ross develop two other reduced load approximations, one for the heavy traffic case and another for the case when none of the links is in the heavy traffic regime. The first approximation (knapsack approximation), originally published in [Dzi87], is based on using the Roberts-Kaufman [Kau81, Rob81] recursive algorithm for obtaining the link blocking probabilities and then invoking the link independence assumption to get the following approximation. Let us denote by $E_{j,k}$ the blocking probability of a class-k call on link j. Note that $E_{j,k} = 0$, if link j is not used by class-k, which happens when $b_{j,k} = 0$. Then the fixed point equations for this reduced load approximation become

$$E_{j,k} = \mathrm{RK}_{j,k}\left[C_j, \left\{\rho_{k'} \prod_{i \neq j}(1 - E_{i,k'}), k' = 1, \ldots, K\right\}\right], \quad j = 1 \ldots, J,\ k = 1, \ldots, K,$$

where $\mathrm{RK}_{j,k}(\cdot,\cdot)$ denotes the blocking probability of a class-k call obtained by using the Roberts-Kaufman algorithm on a link with capacity C_j, offered traffics $\{\rho_k, k = 1, \ldots, K\}$ each thinned by the amount that gets blocked on all other links except link j, and bandwidth requirements $\mathbf{b}_j$. The fixed point equations can then be solved iteratively. Blocking probabilities for each class are then approximated by

$$B_k = 1 - \prod_{j=1}^{J}(1 - E_{j,k}). \tag{3.12}$$

The authors then proceed to show, that this approximation does not retain the uniqueness property of the solution, as opposed to (3.11). However, in the asymptotic limit and under the heavy traffic assumption the approximation converges to the correct values. Also, it is shown by numerical examples that the knapsack approximation gives more accurate results than the approximation of Kelly, although with a considerable increase in the computational complexity of the approximation — recall that Kelly's approximation only requires the calculation of simple Erlang-B functions.

The authors develop also another approximation, Pascal approximation, which is based on approximating the link occupancy distributions with a truncated Pascal distribution (negative binomial distribution). The parameters of the distribution are obtained by using the heuristic that when the link capacity is infinite, the real link occupancy of link j would have mean and variance given by

$$m_j = \sum_k b_{j,k}\rho_k, \quad \sigma_j^2 = \sum_k b_{j,k}^2 \rho_k.$$

Then we can approximate the real finite link occupancy distribution by a Pascal distribution having the same mean and variance, but truncated to the size of link j. Let us denote by $q_j(n), n = 0, \ldots, C_j$, the approximate probability of having n circuits occupied on link j. Also, define

$$p_k(C_j, m_j, \sigma_j^2) = \sum_{n=C_j-b_k+1}^{C_j} q_j(n),$$

as an approximation to the probability of blocking on link j for a class-k call. Now, by again invoking the link independence assumption, we get for m_j and σ_j^2

$$\begin{aligned} m_j &= \sum_k b_{k,j}\rho_k \prod_{i \neq j}(1 - E_{i,k}), \\ \sigma_j^2 &= \sum_k b_{k,j}^2 \rho_k \prod_{i \neq j}(1 - E_{i,k}), \end{aligned}$$

and the reduced load approximation can be expressed as

$$E_{j,k} = p_k\left[C_j, m_j, \sigma_j^2\right], \quad j = 1, \ldots, J, \quad k = 1, \ldots, K.$$

Blocking probabilities are then again approximated by (3.12). The authors show by numerical results that this approximation is more accurate than the knapsack approximation (and thus Kelly's approximation) in the light traffic case but in the critical traffic case the knapsack approximation is again more accurate. In fact, the authors show that asymptotically the Pascal approximation converges to the correct values for the blocking probabilities in light and critical traffic cases. Also, note that the computational complexity of this approximation is equivalent to that of Kelly's since the value of $p_k(\cdot, \cdot, \cdot)$ can be obtained as a result of a one dimensional recursion in the same way as the value of the Erlang-B function, which is used in Kelly's reduced load approximation.

In [Mit95] Mitra, Morrison and Ramakrishnan suggest another reduced load approximation, where the link blocking probabilities are evaluated by using the uniform asymptotic approximation for the single link case as presented in [Mit94] (see the previous section). Then the fixed point equations are obtained by, again, using the link independence assumption.

Methods Using Generating Functions

In [Cho95a] (see also [Cho95b]) Choudhury, Leung and Whitt develop numerical inversion algorithms for the generating functions of normalization constants. The approach uses the fact that the blocking probabilities are expressed as functions of different normalization constants, i.e. $B_k = 1 - G_k/G$, and that their generating functions can be expressed in closed form. Recall from chapter 2.1 that G and G_k are normalization constants calculated with different link capacities and that $G(\mathbf{C})$ denotes the normalization constant G for a system with link capacity vector $\mathbf{C}$. Then, for example, the moment generating function $G(\mathbf{z}) = \sum_{\mathbf{C}} G(\mathbf{C}) \prod_j z_j^{C_j}$ of $G(\mathbf{C})$ is given by

$$\begin{aligned}
G(\mathbf{z}) &= \sum_{C_1=0}^{\infty} \cdots \sum_{C_J=0}^{\infty} \sum_{\mathbf{x}\in\mathcal{S}} \prod_{k=1}^{K} \frac{\rho_k^{x_k}}{x_k!} z_1^{C_1} \cdots z_J^{C_J} \\
&= \sum_{x_1=0}^{\infty} \cdots \sum_{x_K=0}^{\infty} \sum_{C_1=\mathbf{b}_1\cdot\mathbf{x}}^{\infty} \cdots \sum_{C_J=\mathbf{b}_J\cdot\mathbf{x}}^{\infty} \prod_{k=1}^{K} \frac{\rho_k^{x_k}}{x_k!} z_1^{C_1} \cdots z_J^{C_J} \\
&= \frac{\exp\left(\sum_{k=1}^{K} \rho_k \prod_{j=1}^{J} z_j^{b_{j,k}}\right)}{\prod_{j=1}^{J}(1 - z_j)}
\end{aligned}$$

Then the authors devise an algorithm which can numerically invert the generating functions for a given link capacity vector. The main idea in the development of the algorithm is the transformation of the original problem from a K-dimensional recursion into K one-dimensional recursions and the inclusion of a scaling method for improving numerical stability. In their numerical examples the authors show that they are able to handle even large networks with good accuracy.

In [Sim97] Simonian, Roberts, Théberge and Mazumbar continue the work done in [Gaz93] for the single link case and they generalize the results for the network case obtaining approximations for all loading regimes. They consider the multidimensional distribution of the link occupancies to which they apply a probability shift technique similar to that used in [Gaz93]. For this “shifted” link occupancy distribution a local central limit theorem is obtained allowing one to approximate the shifted distribution by a centered Gaussian distribution having the covariance of the shifted link occupancy distribution. Using this result and deriving approximations to the normalization constant, the authors obtain approximations for the class-k call blocking probabilities assuming that all the links in the network are either under light, critical or heavy traffic, thus generalizing the earlier results for the single link case. Also, a uniform estimate is derived for the case when the links in the network are either under light or critical traffic. This corresponds to a uniform asymptotic estimate for the light-critical traffic case, as opposed to [Kel91a], where Kelly obtains a similar uniform asymptotic estimate for the critical-heavy traffic case.

In [Dow96] Down and Virtamo develop another method based on using the contour integral presentation of the generating function for normalization constants. It continues the work done in [Mit94] and [Sim97] and attempts to develop a uniform approximation for the blocking probabilities that would work well also in the case when there is clear dependence between the activity on different links. The basic idea is here the same as in [Mit94], i.e. the development of asymptotic approximations for different normalization constants. Now, with the help of the generating function for the link occupancies in the infinite capacity case

$$F(\mathbf{z}) = \exp\left[\sum_k \rho_k \left(\prod_j z_j^{b_{j,k}} - 1\right)\right],$$

the normalization constant G has the following contour integral expression

$$G = \frac{1}{2\pi i}\oint \frac{dz_1/z_1}{z_1^{C_1}(1-z_1)} \cdots \oint \frac{dz_J/z_J}{z_J^{C_J}(1-z_J)} F(\mathbf{z}).$$

Next, define the function

$$f(\mathbf{z}) = \log F(\mathbf{z}) - \sum_j (C_j + 1)\log z_j.$$

Expanding this function around the saddle point $\mathbf{z}^* > 0$ and using some changes of variables, the authors derive the following uniform asymptotic estimate for the case when $z < 1$ (the heavy traffic case)

$$G = e^{f(\mathbf{z}^*)+\frac{1}{2}(1-\mathbf{z}^*)^T\cdot\mathbf{D}^2\cdot(1-\mathbf{z}^*)} \int_{v_1}^{\infty} dy_1 \cdots \int_{v_J}^{\infty} dy_J \frac{e^{-\frac{1}{2}\mathbf{y}^T\cdot\mathbf{D}^{-2}\cdot\mathbf{y}}}{(2\pi)^{J/2}|\mathbf{D}|}, \tag{3.13}$$

where $\mathbf{D}^2$ is the symmetric matrix of second derivatives with

$$(\mathbf{D}^2)_{ij} = \frac{\partial^2}{\partial z_i \partial z_j} f(\mathbf{z})|_{\mathbf{z}=\mathbf{z}^*},$$

and v_j are the limits of integration given by the components of

$$\mathbf{v} = \mathbf{D}^2 \cdot (1 - \mathbf{z}^*).$$

The integral part in (3.13) is now in the form of a multidimensional integration over a probability density function of a multivariate Gaussian distribution. The authors then derive a simple multivariate Gaussian distribution as an approximation to the integrand in (3.13), which is simple to integrate and still has the correct covariance matrix leading to a computable approximation for the blocking probabilities. However, in the paper it is noted that the method, as it is presented, does not give very good numerical results and can hence be considered as a first step in a method for analyzing the effect of dependence between links on the blocking probabilities in multiservice loss systems.

Chapter 4

Basic Simulation Methods

In this section we review some basic methods for obtaining estimates of the blocking probabilities by using simulation. The methods are roughly divided into three categories differing in the way that the samples are generated. Their efficiency is evaluated through numerical examples by e.g. examining the correlation structure of the samples for the different methods.

4.1 General

An alternative to deriving analytical approximations is to obtain estimates of the performance measures of the system through simulation. Generally, simulation can be used advantageously as a complementary method to analytical methods when modeling complex systems. The advantages of the simulation approach include:

- It is possible to construct a simulation model of (almost) any complex abstract or real world system.
- Simulation allows great flexibility in the choice of the level of detail for modeling the system.
- Simulating any given model is a relatively simple computer programming problem, as opposed to problems in solving analytical models for which analytical results do not exist.

On the other hand the simulation approach has its weaknesses:

- Obtaining reliable estimates can be computationally very expensive, despite the tremendous increase in computing power of modern workstations.
- Simulation gives quantitative results for a given set of model parameters and, hence, it is difficult to obtain insight into the qualitative behavior of a system.

In our case the simulation problem can be considered as being stochastic mathematical simulation (using the terminology of [Mitr82]). This is because we are here considering, in general, models of a stochastic nature, where the system's behavior is determined by random events. Secondly, the system we are ultimately interested in simulating, the multiservice loss system, is itself a purely mathematical object representing an abstraction suitable for modeling a situation where a large population of customers enter the system requesting resources from several shared resources and the customers stay in the system for a random period of time.

To be precise, our problem is now of the following type. We want to evaluate some quantity H defined as the expectation of a random variable $h(\mathbf{X})$, where $\mathbf{X}$ has distribution P defined in the state space $\mathcal{S}$,

$$H = \mathrm{E}\left[h(\mathbf{X})\right]. \tag{4.1}$$

Then an unbiased estimator for H when N samples have been drawn is

$$\hat{H} = \frac{1}{N}\sum_{n=1}^{N} h(\mathbf{X}_n). \tag{4.2}$$

In principle simulation would allow one to obtain estimates with arbitrary accuracy, but in practice this is limited by the available computing power and hence the time it takes to reach a certain precision. Therefore, when comparing one method to another, their relative efficiency depends on:

1. the effort required to generate the samples from P,
2. the covariance of the generated samples $\mathbf{X}_n$,
3. the effort to evaluate the function $h(\cdot)$,
4. the variance of $h(\mathbf{X})$.

In this chapter we will compare the efficiency of different simulation methods with respect to the first two items in the above list. Typically there is a trade-off between these two items. For this, we note first that the variance of (4.2) is

$$\mathrm{Var}\left[\hat{H}\right] = \frac{1}{N^2}\sum_{n=1}^{N}\left\{\mathrm{Var}\left[h(\mathbf{X}_n)\right] + \sum_{m\neq n}\mathrm{Cov}\left[h(\mathbf{X}_n), h(\mathbf{X}_m)\right]\right\}. \tag{4.3}$$

From this it can readily be seen that positive correlation between the samples makes the estimator less efficient from the point of view of the variance.

4.2 The Static Monte Carlo Method

In the static Monte Carlo method the idea is to generate i.i.d. samples $\mathbf{X}_n \in \mathcal{S}$ with distribution P also implying that the method requires explicit knowledge of the distribution P. From (4.3) it is seen that from the point of view of variance this method gives the most efficient samples, but this happens typically at the expense of having a higher computational effort per generated sample. Well known sample generation methods that fall in the class of static Monte Carlo methods are described e.g. in [Rub98] and include the inverse transform method or the rejection method. However, we can note here that as a method the static Monte Carlo does not require the samples to be independent. In fact, to be able to use the static Monte Carlo method it is only required that the distribution P must be known. Later on we will see how we can generate dependent samples with a class of algorithms called Markov Chain Monte Carlo methods.

In the case of multiservice loss systems a natural method for generating the samples belonging in the class of static Monte Carlo methods is the following rejection method. It consists of generating samples of $\tilde{\mathbf{X}}$ in a larger space $\tilde{\mathcal{S}} \supseteq \mathcal{S}$ and rejecting those samples which fall outside of the allowed state space $\mathcal{S}$. The accepted samples $\mathbf{X}_n$ will then have the correct distribution π and we can estimate the blocking probabilities from (2.2) by

$$\hat{B}_k = \frac{1}{N}\sum_{n=1}^{N} 1_{\mathbf{X}_n \in \mathcal{B}^k}, \tag{4.4}$$

where N is the number of those samples falling inside the allowed state space $\mathcal{S}$.

A particularly suitable choice for $\tilde{\mathcal{S}}$ is the Cartesian product space limited by the maximum number of allowed class-k calls $N^k_{\max}$. Formally this state space is defined as

$$\tilde{\mathcal{S}} = \{0, \ldots, N^1_{\max}\} \times \cdots \times \{0, \ldots, N^K_{\max}\}.$$

This state space has the nice property that the product form solution (2.1) in $\tilde{\mathcal{S}}$ means that the different components are independent and hence the samples are easy to generate. Then the sampling distribution is given by:

$$\tilde{\pi}(\mathbf{x}) = \frac{1}{\tilde{G}}\prod_{k=1}^{K}\frac{\rho_k^{x_k}}{x_k!} = \frac{f(\mathbf{x})}{\tilde{G}}, \quad \mathbf{x} \in \tilde{\mathcal{S}}, \tag{4.5}$$

where

$$\tilde{G} = \prod_{k=1}^{K}\sum_{n=0}^{N^k_{\max}}\frac{\rho_k^n}{n!}.$$

4.3 Process Simulation

In many cases we are given a process $\mathbf{X}_t$ and it is possible to directly simulate the process itself. The use of static Monte Carlo requires explicit knowledge of the stationary distribution P of the process, which is, in general, unknown. Using process simulation it is possible to avoid that. Then we simulate the process to generate samples $\mathbf{X}_n$ with the distribution P to estimate (4.2). However, now the samples will be correleted. Typically the correlation is positive and hence, by (4.3), the sampling is less efficient from the point of view of the variance.

The loss system is an exception, since it is a process for which the stationary distribution is also known, as discussed e.g. in [Ros95]. Next we describe several methods for estimating the blocking probabilities by means of Markov chain techniques.

4.3.1 Continuous Time Markov Chain Simulation

In the case of the multiservice loss system we have K independent Poisson arrival streams with intensities $\lambda_k, k = 1, \ldots, K$. Recall that the network consists of J links each with capacity C_j and that associated with each class-k arrival is its bandwidth requirement $b_{j,k}$ on each of the links in the network. Also, we know that the stationary distribution depends on the service time distribution only through its mean and we are therefore free to choose the service times to have a negative exponential distribution. Then the system is itself described by a multidimensional continuous time Markov chain (CTMC) $\mathbf{Y}_t$. Associated with the arrival and departure epochs of the CTMC is the embedded discrete time Markov chain (DTMC) $\mathbf{X}_n$. In Fig. 4.1 we have shown a transition diagram for a two traffic class example. Note that even the transitions corresponding to arrivals in the blocking states are included in the jump chain. We will call this as the *full jump chain* of the process. From this, we are interested in estimating the blocking probability of a traffic class-k call. This is given by the steady state performance measure $\lim_{t\to\infty} \mathrm{E}[1_{\mathbf{Y}_t \in \mathcal{B}^k}]$ corresponding to the so called time congestion of the process, i.e. the proportion of time the process spends in the blocking states for traffic class k.

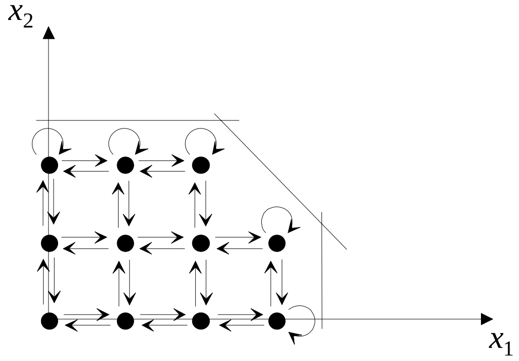

Figure 4.1: Transition diagram for the 2 traffic class example.

The branching probabilities $p(\mathbf{x}, \mathbf{y})$ from state $\mathbf{x}$ to state $\mathbf{y}$ of the DTMC are

$$\begin{cases} p(\mathbf{x}, \mathbf{x}+\mathbf{e}_k) &= \dfrac{\lambda_k}{\sum\limits_k \lambda_k + \sum\limits_k x_k\mu_k}, & \text{if } \mathbf{x} \in \mathcal{S}^k, \\ p(\mathbf{x}, \mathbf{x}) &= \dfrac{\lambda_k}{\sum\limits_k \lambda_k + \sum\limits_k x_k\mu_k}, & \text{if } \mathbf{x} \in \mathcal{B}^k, \\ p(\mathbf{x}, \mathbf{x}-\mathbf{e}_k) &= \dfrac{x_k\mu_k}{\sum\limits_k \lambda_k + \sum\limits_k x_k\mu_k}, & \text{if } \mathbf{x} \in \mathcal{S}, \\ p(\mathbf{x}, \cdot) &= 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

Also, the lifetime of each state, $T(\mathbf{x}_n)$, has an exponential distribution

$$T(\mathbf{x}_n) \sim \mathrm{Exp}(\sum_k \lambda_k + \sum_k x_k\mu_k).$$

The simulation of the CTMC consists of generating a sample path $\mathbf{X}_n$ of the DTMC with branching probabilities given by (4.6) and in each state the generation of the random time $T(\mathbf{X}_n)$ until the next event (arrival or departure). To estimate the blocking probability we generate N samples from the DTMC and collect the sum of the random times the process spent in each state, i.e. the total simulated process time, and the sum of the times the process spends in the blocking states. The blocking probability is then estimated by

$$\hat{B}_k = \frac{\sum_{n=1}^{N} 1_{\mathbf{X}_n \in \mathcal{B}^k} T(\mathbf{X}_n)}{\sum_{n=1}^{N} T(\mathbf{X}_n)}. \tag{4.7}$$

Note that when using this formulation, the estimator is not unbiased: If the process is started from the steady state distribution of the CTMC, it is not in the steady state of the embedded DTMC and hence there is a transient period when the embedded DTMC reaches steady state. On the other hand, if the process is started from the steady state of the DTMC it is not the steady state of the CTMC and, therefore, it is also biased. The estimator is strongly consistent though. An unbiased formulation would be to fix the total simulation time, instead of having the time depend on e.g. the number of generated transitions from the DTMC, to start the process from the steady state of the CTMC, and then to estimate the blocking probability as the ratio of the time the process spends in the blocking states and the total simulation time.

However, it can be noted that from the point of view of the Markov chain and its properties the self-transitions corresponding to blocked arrivals can be removed. We call the resulting process as the *partial jump chain.* This of course affects the arrival process by effectively removing the blocked arrivals, but since we are only interested in the amount of time spent in each state then it is sufficient if this property is retained after removing the transitions.

Intuitively, it is obvious that this can be done, since the effect of the removal of a self-transition from the full jump chain will simply be "merged" with the prolonged average lifetime of the state in the partial jump chain.

4.3.2 Discrete Time Markov Chain Simulation

Now, let us turn our attention to the question of using the embedded DTMC $\mathbf{X}_n$, i.e. the full jump chain, for generating samples with the stationary distribution π given by (2.1). By simulating the embedded DTMC $\mathbf{X}_n$ we can use the samples in the following ways:

1. The chain $\mathbf{X}_n$ weighted with the expected lifetime of the state directly has the distribution π.
2. The subchain $\mathbf{X}_{A(m)}$, where $A(m)$ is the index of the state that the m^{th} arriving customer sees, has the distribution π (PASTA property of the system). Also, by time reversibility of the chain $\mathbf{X}_{A(m)}$, the subchain $\mathbf{X}_{D(m)}$, where $D(m)$ is the index of the state after the m^{th} departure, has the distribution π.
3. The subchain $\mathbf{X}_{A_k(m)}$, where $A_k(m)$ is the index of the state that the m^{th} arriving class-k customer sees, has the distribution π (PASTA property of the system). Also, by time reversibility of the chain $\mathbf{X}_{A_k(m)}$, the subchain $\mathbf{X}_{D_k(m)}$, where $D_(m)$ is the index of the state after the m^{th} class-k departure, has the distribution π.

The Weighted Samples Method

The first method is a consequence of a known relation between the stationary distribution $\pi(\mathbf{x})$ of the CTMC, given by (2.1), and the stationary distribution $\pi^*(\mathbf{x})$ of the embedded DTMC:

$$\pi(\mathbf{x}) = \frac{\pi^*(\mathbf{x})\mathrm{E}[T(\mathbf{x})]}{\sum_{\mathbf{y}\in\mathcal{S}} \pi^*(\mathbf{y})\mathrm{E}[T(\mathbf{y})]}.$$

This yields directly a simulation method, which we call the *weighted samples method.* It is quite similar to the one presented in the previous section for the CTMC. In this method, we also generate samples $\mathbf{X}_n$ from the embedded DTMC. Instead of generating in each state the exponentially distributed lifetime of the state, we use directly its expected value. During the simulation we collect the sum of the expected lifetimes of the generated samples $\mathbf{X}_n$ and separately for each traffic class the expected lifetimes of the samples that hit the blocking states. The blocking probability is then estimated by

$$\hat{B}_k = \frac{\sum_{n=1}^{N} 1_{\mathbf{X}_n \in \mathcal{B}^k} \mathrm{E}[T(\mathbf{X}_n)]}{\sum_{n=1}^{N} \mathrm{E}[T(\mathbf{X}_n)]}. \tag{4.8}$$

Again, we can choose to use the full jump chain or to remove the self-transitions. However, in this case, from the point of view of simulation, there is a slight difference. The removal of a self-transition effectively removes the "variability" in the process caused by that transition arc. Hence, when using the partial jump chain one can expect to have a slightly smaller variance in the estimates when compared with the ones obtained by using the full jump chain for the same number of generated samples. We will give some numerical examples on the effect of removing the self-transitions later in this chapter.

A more important property of this method is that the estimator (4.8) has (in most cases) lower variance than the estimator of the CTMC method (4.7). This can be explained as above by noting that by using the expectations we have effectively removed the variability in each sample associated with the random time the process spends in each state, and only the variability caused by the random number of hitting different states is left. We will return to this also in the numerical results section of this chapter.

The Subchain Methods

The two other methods are from the simulation point of view similar — in both cases we simulate the full jump chain with the self-transitions and choose the suitable states. If we use the complete arrival subchain $\mathbf{X}_{A(n)}$, the blocking probability is estimated by generating a sample path from the full jump chain and then choosing the states $\mathbf{X}_{A(n)}$ as samples. The estimator is then simply

$$\hat{B}_k = \frac{1}{N} \sum_{n=1}^{N} 1_{\mathbf{X}_{A(n)} \in \mathcal{B}^k},$$

where N is the total number of generated arrivals. We will call this method as the *arrival subchain method*. The estimator for the departure subchain $\mathbf{X}_{D(n)}$ can be defined in a similar manner.

If we use the class-k arrival subchain $\mathbf{X}_{A_k(n)}$, the simulation of the blocking probabilities is, again, carried out by generating a sample path from the full jump chain and then choosing the states $\mathbf{X}_{A_k(n)}$ as samples. This, in fact, corresponds to the simulation of the call congestion of the system, i.e. the proportion of lost calls of class-k to the total number of class-k arrivals. The estimator is then

$$\hat{B}_k = \frac{1}{N} \sum_{n=1}^{N} 1_{\mathbf{X}_{A_k(n)} \in \mathcal{B}^k},$$

where N is the total number of class-k arrivals in the simulation. We will call this method as the *class-k arrival subchain method*. Again, this method can be defined, as earlier, for the reverse time model in terms of departing class-k customers.

In these subchain methods we do not have the alternative of omitting the self-transitions, because we are now explicitly relying on the PASTA property of Poisson arrivals, which would be violated if the self-transitions corresponding to blocked arrivals were removed. An interesting question is which method — the weighted samples method or the subchain methods — gives lower variance for the same number of generated samples from the embedded DTMC. Some numerical results related to this question will be given later in this chapter.

4.3.3 Regenerative Simulation

The regenerative simulation method has been developed for the analysis of so called regenerative stochastic processes, see for example [Cra75, Cra77]. Heuristically, a regenerative process is a stochastic process, which starts probabilistically afresh at certain points in time, in other words regenerates itself. Then we are able to break the (theoretically) infinite length steady-state simulation into distinct independent identically distributed finite length "cycles", which start from the regeneration state and end there. This is shown for a one dimensional process in Fig. 4.2, where the horizontal line represents the regeneration state and the vertical lines indicate the starting of a new cycle.

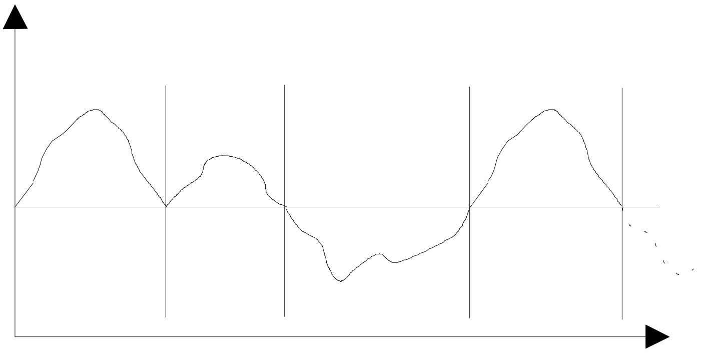

Figure 4.2: Regenerative simulation.

To be precise, the regenerative simulation method for discrete time simulations is defined as follows, see e.g. [Cra75]. Let $\mathbf{X}_n$ now denote the irreducible embedded Markov chain of a stochastic process with finite state space and transition matrix $\mathbf{P}$. The goal is, again, to estimate the expectation $H = \mathrm{E}[h(\mathbf{X})]$ of some function $h(\mathbf{X})$. From the theory of regenerative processes we know that this expectation can be expressed as a ratio of the expectations of two random variables

$$\mathrm{E}\left[h(\mathbf{X})\right] = \frac{\mathrm{E}[F]}{\mathrm{E}[G]},$$

where the random variable $F = \sum_{n=1}^{G} h(\mathbf{X}_n)$ is the value of $h(\cdot)$ over the length of a cycle and G is the "length" of the cycle, i.e. the number of samples generated from the DTMC between two successive regeneration epochs. From this we get the estimator

$$\hat{H} = \frac{1/M \sum_{m=1}^{M} F_m}{1/M \sum_{m=1}^{M} G_m} = \frac{\sum_{m=1}^{M} F_m}{\sum_{m=1}^{M} G_m}, \tag{4.9}$$

where F_m and G_m are i.i.d. observations of F and G during the m^{th} simulated regeneration cycle starting from the regeneration state and ending there and M is the total number of simulation cycles. Note that for Markov chains every state is a regeneration state. This estimator is biased since, in general, $\mathrm{E}[F/G] \neq \mathrm{E}[F]/\mathrm{E}[G]$. However, the estimator is strongly consistent, i.e. $\lim_{M\to\infty} \hat{H} \to H$ with probability 1, since by the law of large numbers the numerator and denominator both converge with probability 1 to their expectations. Regenerative simulation provides a nice way of obtaining i.i.d. samples during a one long simulation run and we are able to use these samples to construct confidence intervals for the ratio estimator (4.9). However, the usefulness of this method relies heavily on the fact that the regeneration state is visited relatively frequently. This is a very strong assumption and it is not easily fulfilled in systems with large multidimensional state spaces, e.g. the multiservice loss system. In comparision with the traditional DTMC methods, it can be noted that in the regenerative method the bias of the estimator is affected by two transients: the so called initial transient resulting from the fact that the chain must be started from a fixed state, and from an extra final transient since we have also fixed the final state of the chain. In contrast, the traditional DTMC methods only have the initial transient.

In the case of the multiservice loss system, we can use the regenerative simulation to estimate the blocking probability for a class-k call in the following way. Let $\mathbf{P}$ denote transition matrix of the embedded DTMC (the full jump chain), with transition probabilities given by (4.6), defined in the state space $\mathcal{S}$. We will now use the class-k arrival subchain method to generate samples with the distribution $\pi(\mathbf{x})$ as in (2.1) by choosing those states just prior to an occurrence of a class-k arrival in the generated path. As before, this chain is denoted by $\mathbf{X}_{A_k(n)}$. Then we can define the random variable F^k simply as

$$F^k = \sum_{n=1}^{G^k} 1_{\mathbf{X}_{A_k(n)} \in \mathcal{B}^k},$$

where G^k is the number of class-k arrivals in a regeneration cycle, i.e. F^k is the number of blocked class-k arrivals. Then the class-k blocking probability is expressed as

$$B_k = \frac{\mathrm{E}[F^k]}{\mathrm{E}[G^k]}.$$

The simulation then consists of generating samples $\mathbf{X}_{A_k(n)}$ for a period of M regenerative cycles starting from a chosen regeneration state and ending there. During the simulation we collect i.i.d. samples of F^k_m and G^k_m, which denote the m^{th} samples of F^k and H^k, respectively. The blocking probability is then estimated simply by

$$\hat{B}_k = \frac{1/M \sum_{m=1}^{M} F^k_m}{1/M \sum_{m=1}^{M} G^k_m}.$$

As was mentioned earlier, the simulation method calls for the choice of a suitable regeneration state, and it leaves us with one free parameter. In [Cra75] the following has been shown. Let $I(t)$ and $I'(t)$ denote the width of the confidence interval of an estimator for two alternative choices of regenaration states when t time units have been simulated. Then $I(t)/I'(t) \to 1$ with probability 1 as $t \to \infty$. This means that when the length of the simulation is long enough the width of the obtained confidence intervals become approximately equal with high probability for simulations starting from different regeneration states. The regenerative simulation literature traditionally use the heuristics of choosing the regeneration state to be the one having the smallest mean cycle length. However, in [Gly93] it is shown that the rate of convergence of the estimator's standard deviation is indeed affected by the choice of the regeneration state. It is also noted there that the standard deviation is not necessarily minimized by this particular choice and that the question of optimal regeneration state choice remains open. Analytic methods for finding the most efficient state for the multiservice loss system will be discussed later in this chapter.

4.4 Markov Chain Monte Carlo Simulation

Markov Chain Monte Carlo (MCMC) methods form a large class of simulation methods, see e.g. [Bro98, Tie94] for surveys. In the MCMC methods the idea is to simulate some Markov chain having a known stationary distribution. In particular, the MCMC methods involve the solution of the problem whereby the stationary distribution is known and we need to identify a transition matrix with the given stationary distribution, as opposed to traditional Markov chain methods, where the transition matrix is already known from the start. Thus we can note that essentially the MCMC methods belong to the class of static Monte Carlo methods despite that the method produces correlated samples.

In general, the MCMC methods are utilized in the field of so called Bayesian statistics, where the problem frequently is as follows. Although the form of the stationary distribution of some multivariate random variable is known, it is not possible to generate samples from the distribution using the most common static Monte Carlo methods producing independent samples. Then the MCMC methods attack the problem by constructing an "artificial" Markov chain with the desired stationary distribution and from which it is also easy generate samples. The most popular methods can be split into two distinct classes [Bro98]: the Gibbs sampler type methods and, the more general, Metropolis-Hastings algorithms. In this section, we briefly describe the Metropolis-Hastings algorithm and its background, and, more thoroughly, the original Gibbs sampler, which can be applied in a very elegant way in the context of the multiservice loss system as has been shown in [Las98a].

4.4.1 Metropolis-Hastings Algorithm

The so called Metropolis algorithm was originally published in [Met53] for computing properties of substances composed of interacting individual molecules and since then it has been used extensively in the field of statistical physics, see e.g. [Ham67]. The algorithm was later generalized by Hastings in [Has70]. Here we follow the description of the method given in [Bro98], where other variants of the Metropolis-Hastings method can also be found.

The idea is based upon deriving a Markov chain to generate samples with a desired distribution which also satisfies the conditions for detailed balance. Recall that if a Markov chain $\mathbf{X}_n$ with transition probabilities $q(\mathbf{x}, \mathbf{y})$ satisfies the detailed balance for distribution $p(\mathbf{x})$, i.e.

$$q(\mathbf{x}, \mathbf{y})p(\mathbf{x}) = q(\mathbf{y}, \mathbf{x})p(\mathbf{y}), \tag{4.10}$$

then the Markov chain has the stationary distribution $p(\mathbf{x})$.

The method begins by choosing some $q(\mathbf{x}, \mathbf{y})$ for generating candidate observations and we allow this candidate generation distribution to depend on the current state of the chain $\mathbf{X}_n$. For this let us denote by $q(\mathbf{x}, \mathbf{y})$ for the transition probability from state $\mathbf{x}$ at to state $\mathbf{y}$. In general the $q(\cdot, \cdot)$ chain associated with the candidate generation distribution itself does not need to be reversible. Using this we generate a next candidate observation and we introduce an acceptance function $\alpha(\mathbf{x}, \mathbf{y})$ so that the new candidate is accepted with probability $\alpha(\mathbf{x}, \mathbf{y})$ in which case the chain moves to the state $\mathbf{X}_{n+1} = \mathbf{y}$. Otherwise the candidate is rejected and the chain remains in state $\mathbf{X}_n$.

It has been shown that the optimal form for the acceptance function, in the sense that suitable candidates are rejected least often and computational efficiency is maximized, is given by

$$\begin{aligned}\alpha(\mathbf{x}, \mathbf{y}) &= \min\left[1, \frac{p(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})q(\mathbf{x}, \mathbf{y})}\right] \\ &= \min\left[1, \frac{p(\mathbf{y})}{p(\mathbf{x})}\right] \quad \text{(if } q(\mathbf{x}, \mathbf{y}) \text{ is symmetric).}\end{aligned}$$

Then the complete transition probability distribution is given by $q(\mathbf{x}, \mathbf{y})\alpha(\mathbf{x}, \mathbf{y})$, which also satisfies the reversibility condition (4.10). It can be shown that the resulting Markov chain $\mathbf{X}_n$ has the stationary distribution $p(\mathbf{x})$, see e.g. [Tie94]. Note that, for implementation, we only need to know the distribution $p(\mathbf{x})$ up to a constant of proportionality.

The only open remaining question is how to choose the candidate generating distribution? If $q(\mathbf{x}, \mathbf{y}) = \gamma(\mathbf{y} - \mathbf{x})$ for some arbitrary density γ, then the kernel driving the chain is a random walk. Common choices for γ include the uniform distribution on the unit disk, a multivariate normal distribution or a t-distribution. Alternative choices for $q(\mathbf{x}, \mathbf{y})$ are discussed e.g. in [Bro98] and [Tie94].

4.4.2 Gibbs Sampler

Again, let $\mathbf{X} = (X_1, \ldots, X_K) \in \mathcal{S}$ denote the vector random variable with the distribution $P : p(\mathbf{x}) = \Pr[\mathbf{X} = \mathbf{x}]$. Another method for constructing a Markov chain $\mathbf{X}_n$ having the invariant distribution P is to use transition probabilities based on conditioning, as defined in the following theorem (taken with slight modification from [Tie94]).

Theorem 1: Let the sets $\mathcal{A}_1, \ldots, \mathcal{A}_I$ form a partition of the state space $\mathcal{S}$ and let $\iota(\mathbf{x})$ denote the unique index of the set to which the state $\mathbf{x}$ belongs. Let $\mathbf{X}$ be a random variable with distribution P. Then the Markov chain $\mathbf{X}_n$ with the transition probability

$$\Pr\left[\mathbf{X}_{n+1} = \mathbf{y} \mid \mathbf{X}_n = \mathbf{x}\right] = \Pr\left[\mathbf{X} = \mathbf{y} \mid \mathbf{X} \in \mathcal{A}_{\iota(\mathbf{x})}\right] \tag{4.11}$$

has the invariant distribution P.

Proof:

$$\begin{aligned}
\Pr\left[\mathbf{X}_{n+1}=\mathbf{y}\right] &= \sum_{\mathbf{x}\in\mathcal{S}} \Pr\left[\mathbf{X}_{n+1}=\mathbf{y}\mid\mathbf{X}_n=\mathbf{x}\right]\Pr\left[\mathbf{X}_n=\mathbf{x}\right]\\
&= \sum_i \sum_{\mathbf{x}\in\mathcal{A}_i} \Pr\left[\mathbf{X}=\mathbf{y}\mid\mathbf{X}\in\mathcal{A}_{\iota(\mathbf{x})}\right]\Pr\left[\mathbf{X}_n=\mathbf{x}\right]\\
&= \sum_i \Pr\left[\mathbf{X}=\mathbf{y}\mid\mathbf{X}\in\mathcal{A}_i\right]\sum_{\mathbf{x}\in\mathcal{A}_i}\Pr\left[\mathbf{X}_n=\mathbf{x}\right]\\
&= \sum_i \Pr\left[\mathbf{X}=\mathbf{y}\mid\mathbf{X}\in\mathcal{A}_i\right]\Pr\left[\mathbf{X}_n\in\mathcal{A}_i\right]\ .
\end{aligned}$$

Now, if $\mathbf{X}_n$ has the distribution π, so does $\mathbf{X}_{n+1}$ because then

$$\Pr\left[\mathbf{X}_{n+1}=\mathbf{y}\right]=\sum_i \Pr\left[\mathbf{X}=\mathbf{y}\mid\mathbf{X}\in\mathcal{A}_i\right]\Pr\left[\mathbf{X}\in\mathcal{A}_i\right]=\Pr\left[\mathbf{X}=\mathbf{y}\right]=p(\mathbf{y})\ \square$$

Let $\mathbf{P}^{(1)}$ denote the transition probability matrix with the components given by eq. (4.11). The Markov chain generated by this transition matrix is not irreducible, because there are no transitions between different sets. However, by defining several partitions $1,\ldots,M$ we can construct an irreducible Markov chain $\mathbf{X}_n$. Let $\mathbf{P}^{(m)}$, $m=1,\ldots,M$, denote the corresponding transition matrices. Then, with a suitable choice of the partitions, the Markov chain $\mathbf{X}_n$ corresponding to the compound transition matrix $\mathbf{P}=\mathbf{P}^{(1)}\cdots\mathbf{P}^{(M)}$ will be irreducible. Since each $\mathbf{P}^{(m)}$ has the invariant distribution P also the compound matrix $\mathbf{P}$ will have the invariant distribution P, and because $\mathbf{X}_n$ is now irreducible, P is also its unique stationary distribution.

In the case of the multiservice loss system we have a product form solution $\pi(\mathbf{x})$ given by (2.1) and it is natural to define the sets in a partition to consist of points in coordinate directions. This leads to the so called Gibbs sampler, which was introduced to the Bayesian image analysis literature in [Gem84] and subsequently generalized in [Gel90]. Other variants of the Gibbs sampler method can be found in the survey papers mentioned earlier.

For the purpose of estimating the blocking probability of class-k calls in the multiservice loss system we define the partition to consist of sets in the coordinate direction of traffic class k. Considering all the traffic classes we have altogether K partitions. Let us denote by $\mathcal{A}_i^k$ the i^{th} set in partition k. Using the example of two traffic classes, shown in Fig. 4.3 (the left figure) for traffic class 2, the partition consists of the vertical columns. Each set $\mathcal{A}_i^k$ consists of states where the number of calls of all other classes are fixed, but the k^{th} component varies. In general, we refer to sets $\mathcal{A}_i^k$ as k-columns. The set of blocking states $\mathcal{B}^k$ for the class-k calls consists of the end points of the k-columns. Associated with each partition is a transition matrix $\mathbf{P}^{(k)}$. The Markov chain $\mathbf{X}_n$ generated by the compound transition matrix $\mathbf{P}=\mathbf{P}^{(1)}\cdots\mathbf{P}^{(K)}$ is irreducible since it is possible to move from any state $\mathbf{x}$ in the coordinate convex state space $\mathcal{S}$ to any other state $\mathbf{y}$ with at most K steps in alternating directions.

The simulation of the Markov chain $\mathbf{X}_n$ consists of making transitions with the transition matrices $\mathbf{P}^{(k)}$ in cyclical order. This is illustrated for the two traffic class example in Fig. 4.3 (the right figure). In transitions generated with $\mathbf{P}^{(k)}$ only the component x_k changes. Starting from state $\mathbf{X}_n$ the value of x_k of the next state is obtained by drawing it from the one- dimensional truncated Poisson distribution $f(x_k, \rho_k)/g(L^k(\mathbf{X}_n), \rho_k)$ with $x_k \in (0, \ldots, L^k(\mathbf{X}_n))$, $L^k(\mathbf{X}_n)$ denoting the length of the k-column to which the state $\mathbf{X}_n$ belongs, and $g(L, \rho)$ denoting the normalization sum

$$g(L, \rho) = \sum_{l=0}^{L} \frac{\rho^l}{l!}.$$

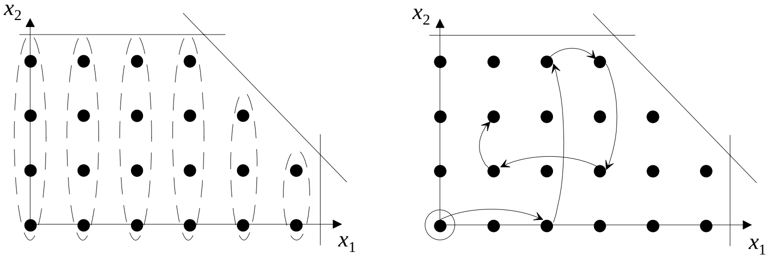

Figure 4.3: State space partitioning and Gibbs sampler example.

The Gibbs sampler provides a way of generating Monte Carlo samples from the state space $\mathcal{S}$, which is simple requiring only the generation of random variables from univariate truncated Poisson distributions for each transition. The advantage it has is that it manages to eliminate the problem of generating 'misses' from the state space $\mathcal{S}$, as happens with the traditional Monte Carlo techniques. On the other hand, the generation of transitions from the Markov chain of the Gibbs sampler is almost as easy as for generating them from the embedded Markov chain associated with the process. The samples generated with the Gibbs sampler are, however, less correlated than the samples from the embedded Markov chain. These issues will be elaborated upon in the next section.

4.5 Numerical and Analytical Studies

Here we make numerical and analytical studies related to some of the questions raised during this chapter. First we compare some of the basic simulation methods introduced in this chapter with respect to the variance of the methods and their computational complexity. Then we present a method for explicitly computing the moments of the ratio estimator associated with the regenerative method for simulating the loss system.

4.5.1 Continuous Time Simulation vs. Discrete Time Simulation

Earlier we made the claim that using the weighted samples method leading to the estimator (4.8) has (in most cases) lower variance than the estimator of the CTMC method (4.7). For this, let us first consider the numerator in (4.8) and (4.7), when there is just one blocking state. For this, let us denote by V the random variable for the time the process spends in the blocking state. Then the numerator in (4.7) is simply a random sum of independent (Markov property) and identically (one state in $\mathcal{B}^k$) distributed random variables V_i with variance

$$\mathrm{Var}\left[\sum_{i=1}^{I} V_i\right] = \mathrm{E}\left[I\right]\mathrm{Var}\left[V\right] + \mathrm{E}\left[V\right]^2\mathrm{Var}\left[I\right],$$

where I denotes the random number of times the condition $1_{\mathbf{X}_n\in\mathcal{B}^k}$ is true for the N samples. Calculating the same for (4.8) gives

$$\mathrm{Var}\left[I\mathrm{E}\left[V\right]\right] = \mathrm{E}\left[V\right]^2\mathrm{Var}\left[I\right].$$

Thus, by replacing V_i with its expectation, we have eliminated the first term in the variance of the numerator of (4.7). In the case when the set $\mathcal{B}^k$ consists of many states, we can show the same in the following way. The variance of the numerator in (4.7) can be computed by conditioning the covariance of each sample on the pair $(\mathbf{X}_i, \mathbf{X}_j)$, i.e.

$$\begin{aligned}
\mathrm{Var}\left[\sum_{n=1}^{N} 1_{\mathbf{X}_n\in\mathcal{B}^k}T(\mathbf{X}_n)\right] &= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{Cov}\left[1_{\mathbf{X}_i\in\mathcal{B}^k}T(\mathbf{X}_i), 1_{\mathbf{X}_j\in\mathcal{B}^k}T(\mathbf{X}_j)\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{E}\left[\mathrm{Cov}\left[1_{\mathbf{X}_i\in\mathcal{B}^k}T(\mathbf{X}_i), 1_{\mathbf{X}_j\in\mathcal{B}^k}T(\mathbf{X}_j) \mid \mathbf{X}_i, \mathbf{X}_j\right]\right] \\
&\qquad + \mathrm{Cov}\left[\mathrm{E}\left[1_{\mathbf{X}_i\in\mathcal{B}^k}T(\mathbf{X}_i) \mid \mathbf{X}_i\right], \mathrm{E}\left[1_{\mathbf{X}_j\in\mathcal{B}^k}T(\mathbf{X}_j) \mid \mathbf{X}_j\right]\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{E}\left[1_{\mathbf{X}_i\in\mathcal{B}^k}1_{\mathbf{X}_j\in\mathcal{B}^k}\mathrm{Cov}\left[T(\mathbf{X}_i), T(\mathbf{X}_j)\right]\right] \\
&\qquad + \mathrm{Cov}\left[1_{\mathbf{X}_i\in\mathcal{B}^k}\mathrm{E}\left[T(\mathbf{X}_i)\right], 1_{\mathbf{X}_j\in\mathcal{B}^k}\mathrm{E}\left[T(\mathbf{X}_j)\right]\right] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{E}\left[1_{\mathbf{X}_i\in\mathcal{B}^k}1_{\mathbf{X}_j\in\mathcal{B}^k}\delta_{ij}\mathrm{Var}\left[T(\mathbf{X}_i)\right]\right] \\
&\qquad + \mathrm{Cov}\left[1_{\mathbf{X}_i\in\mathcal{B}^k}\mathrm{E}\left[T(\mathbf{X}_i)\right], 1_{\mathbf{X}_j\in\mathcal{B}^k}\mathrm{E}\left[T(\mathbf{X}_j)\right]\right].
\end{aligned}$$

In the final step we have utilized the fact that in a CTMC the random life times of each state are independent of each other. From this it can be seen that if we use the expected values of the life times of the states as in (4.8), instead of drawing them from the exponential distributions of the states, we effectively remove the first term in the result above since $\mathrm{Var}[\mathrm{E}[T(\mathbf{X}_i)]] = 0$. The above result holds also for the case when considering the denominator of (4.7). Hence, by using (4.8) we have reduced the variance of the estimator both in the numerator and denominator. However, this does not imply that the ratio estimator's variance is reduced, since the covariance of the numerator and denominator also affects the results.

In [Goy92] this result is shown in the context of regenerative simulation for an arbitrary steady state performance measure of a Markov chain and in [Fox96] this has been extended to semi-Markov processes. However, the proof in [Goy92] also leaves open the effect of the covariance between the numerator and the denominator. In regenerative simulation the dependence is easily demonstrated by considering the asymptotic variance of the regenerative estimator, see e.g. [Cra77, p. 39]. For this let r be the expectation of some steady state performance measure of a regenerative process. Then from the regenerative theory we know that $r = \mathrm{E}[X]/\mathrm{E}[\alpha]$, where X is a cumulative value calculated over one regenerative cycle and α is the length of a regenerative cycle. Then we have the corresponding regenerative estimator $\hat{r} = \bar{X}/\bar{\alpha}$, where $\bar{X} = 1/N \sum_{n=1}^{N} X_n$ and $\bar{\alpha} = 1/N \sum_{n=1}^{N} \alpha_n$ are the sample averages after N samples have been drawn. Then if we consider the sequence of i.i.d. variables $Z_n = X_n - r\alpha_n$ and denote with $\bar{Z} = 1/N \sum_{n=1}^{N} Z_n$, we have by the central limit theorem

$$\sqrt{N}\ \bar{Z} = \sqrt{N}(\bar{X} - r\bar{\alpha}) \to \mathrm{N}(0, \sigma^2), \quad \text{when } N \to \infty,$$

where

$$\sigma^2 = \mathrm{Var}\left[X\right] - 2r\mathrm{Cov}\left[X, \alpha\right] + r^2\mathrm{Var}\left[\alpha\right].$$

By dividing with $\bar{\alpha}$ we get

$$\sqrt{N}(\hat{r} - r) \to \mathrm{N}(0, \sigma^2/\bar{\alpha}), \quad \text{when } N \to \infty.$$

From this we can see that the asymptotic width of the confidence intervals depends not only on the variances of the numerator and denominator, but also on their covariance.

Next we experiment with a few numerical examples on the single link Erlang system using either the CTMC method or the weighted samples method. We used two systems with different capacities and for each two different loads, which were chosen such that the blocking probability is almost equal in the two systems. The arrival and departure intensities were also set such that $\lambda = \rho$ and $\mu = 1$. The exact system parameters are shown in Table 4.1. To compare the results we estimated the standard deviation of the estimators ($\hat{\sigma}_{\mathrm{CTMC}}$ and $\hat{\sigma}_{\mathrm{Weighted}}$ in the table) when 1000 samples are generated from the embedded DTMC (the full jump chain). For the smaller system ($C = 5$) the standard deviation was estimated from 1000 independent simulation runs and for the larger system ($C = 50$) we used 10 000 simulation runs. The results show that the variance of the estimator of the weighted samples method is slightly smaller than that of the CTMC method.

C	ρ	$\hat{\sigma}_{\mathrm{CTMC}}$	$\hat{\sigma}_{\mathrm{Weighted}}$
5	2	0.0094	0.0082
5	1	0.0017	0.0014
50	43	0.0233	0.0226
50	35	0.0060	0.0057

Table 4.1: The standard deviation of the estimators

4.5.2 Comparisons with DTMC Methods

Then we experiment with the difference between using the full jump chain or the partial chain in the weighted samples method for simulating the blocking probabilities. Again we performed tests on the simple Erlang single link model for a small system ($C = 5$) and a larger system ($C = 50$). For both systems we used the same two different loads which were chosen to give a roughly equal blocking probability in both systems and in each case the process was scaled in such a way that the average service rate $\mu = 1$ and the average arrival rate $\lambda = \rho$.

The idea was to have the blocking probability equal in both the small and the larger system to investigate whether the size of the system has any effect on the results. To compare the two methods we calculated in each case the relative efficiency defined as the ratio of the estimator's standard deviation and expected value ($\sigma_{\hat{B}}/\mathrm{E}[\hat{B}]$). In each case we simulated the results of the weighted samples method using the partial chain to a relative efficiency of approximately 21% requiring N (in the table) samples to be generated from the embedded DTMC. Then we tested what is the corresponding figure when using the full jump chain with the same simulation parameters. The exact simulation parameters and the results can be seen in Table 4.2. To estimate the standard deviation we used 10 000 independent simulation runs.

C	ρ	B	N	Partial Chain	Full Jump Chain
5	2	0.0367	1000	0.2095	0.2256
5	1	0.0031	5000	0.2081	0.2186
50	43	0.0376	8500	0.2021	0.2116
50	35	0.0033	60000	0.2140	0.2214

Table 4.2: The relative efficiency of the estimators

From the results it can be seen that the relative efficiency is slightly smaller for the partial chain, although the difference is very small. This result is also "understandable" in the sense that when the self-transition is removed it corresponds to the simulation of the full chain where the effect of the self-transition is calculated analytically.

This can be shown quite easily in the following way and, for generality, we do it for the multiservice case. Let us consider some state $\mathbf{x}$ which is also a blocking state for traffic class m but not for the other classes $k \neq m$. In the partial chain, the lifetime of the state has the expectation

$$\mathrm{E}\left[T(\mathbf{x})\right] = \frac{1}{\sum_{k \neq m} \lambda_k + \sum_k x_k \mu_k}.$$

Calculating the effect of the self-transition in the full jump chain is tantamount to calculating the expectation of the so called first exit time from the state $\mathbf{x}$, denoted by $\tilde{T}(\mathbf{x})$. Let $\tilde{t}(\mathbf{x})$ denote the lifetime of the state $\mathbf{x}$ in the full jump chain. The mean of $\tilde{T}(\mathbf{x})$ can be calculated from the recursion

$$\mathrm{E}\left[\tilde{T}(\mathbf{x})\right] = \mathrm{E}\left[\tilde{t}(\mathbf{x})\right] + \frac{\lambda_m}{\sum_k \lambda_k + \sum_k x_k \mu_k} \mathrm{E}\left[\tilde{T}(\mathbf{x})\right],$$

from which we get directly

$$\begin{aligned}
\mathrm{E}\left[\tilde{T}(\mathbf{x})\right] &= \frac{\sum_k \lambda_k + \sum_k x_k\mu_k}{\sum_{k\neq m} \lambda_k + \sum_k x_k\mu_k}\ \mathrm{E}\left[\tilde{t}(\mathbf{x})\right] \\
&= \frac{\sum_k \lambda_k + \sum_k x_k\mu_k}{\sum_{k\neq m} \lambda_k + \sum_k x_k\mu_k}\ \frac{1}{\sum_k \lambda_k + \sum_k x_k\mu_k} \\
&= \frac{1}{\sum_{k\neq m} \lambda_k + \sum_k x_k\mu_k} = \mathrm{E}\left[T(\mathbf{x})\right].
\end{aligned}$$

Next we turn to the question whether the weighted samples method with the partial chain or the simulation of the full jump chain with the subchain methods gives a lower variance for the same number of generated samples from the embedded DTMC. Again, we use the same methodology as the previous test. We performed tests on the simple Erlang single link model for a small system ($C = 5$) and a larger system ($C = 50$) with the process scaled in such a way that the average service rate $\mu = 1$ and the average arrival rate $\lambda = \rho$. Note that, in this case the arrival subchain method and the class-k arrival subchain methods reduce to the same method. This time we used for both systems three different loads corresponding to very high load, high load and low load conditions.The idea was again to have the blocking probability equal in both the small and the larger system to investigate whether the size of the system has any effect on the results. In each case we simulate the results of the weighted samples method to a relative efficiency of approximately 21% and test what is the corresponding figure for the subchain method with the same simulation parameters. The exact simulation parameters and the results can be seen in Table 4.3. To estimate the sample standard deviation the process was simulated starting from steady state until the number of generated samples indicated in the table (N samples) from the embedded DTMC, and the standard deviation was obtained from 10 000 independent replicas of simulations having the indicated length.

C	ρ	B	N	Weighted Samples	Arrival Subchain
5	3	0.1101	450	0.1989	0.2774
5	2	0.0367	1000	0.2095	0.3179
5	1	0.0031	5000	0.2081	0.4419
50	50	0.1048	2300	0.2104	0.2295
50	43	0.0376	8500	0.2021	0.2204
50	35	0.0033	60 000	0.2140	0.2476

Table 4.3: The relative efficiency of the estimators.

The results clearly show that for a given computational effort the variance is smaller when using the weighted samples method than for both subchain methods. Although, it appears that the difference decreases as the capacity increases. This variance reduction is brought about by the fact that in the weighted samples method we are able to use each generated sample in the estimator, whereas in the subchain consisting of the states just prior to an arrival we can only use a portion of the generated states. However, this is a fair comparison in the sense that the computational effort is equal for both methods, when the extra effort required to weight the samples with the average life time of the state is considered negligible. This advantage would be canceled if we were to compare the methods for the same number of "accepted" samples, i.e. we would generate N arrivals for the subchain method. Also, the results would suggest that the advantage of the weighted samples method seems to increase as the blocking probability becomes smaller. This is explained, at least partly, by the fact that as the blocking probability becomes smaller, the smaller is the number of arrival events for a fixed number of transitions from the DTMC and, hence, the less there are samples to be used in the subchain method.

Then we experiment with small systems having 2 traffic classes to be able to differentiate between the arrival subchain method and the class-k arrival subchain method. However, now we will use the full jump chain in the weighted samples method to be able to use a common realization of the embedded DTMC process for making the comparison of the methods as effective as possible. Intuitively it would seem plausible that the weighted samples method would still be most efficient, since in that method we utilize every sample that is generated from the embedded DTMC, whereas in the subchain methods we only use a subset of the samples.

To support our intuition we use the following single link, two traffic class systems:

- Example 1: $C = 10, \rho = [2, 1], \mathbf{b} = [1, 2]$
- Example 2: $C = 50, \rho = [20, 10], \mathbf{b} = [1, 2]$

The blocking probabilities in these systems were in the range $0.01 \ldots 0.05$ and in both cases the blocking probability of traffic class 2 was higher than for traffic class 1. To compare the methods for Example 1 we estimated the standard deviation of the estimator for 5000 generated samples from the embedded DTMC. For Example 2 we estimated the standard deviation of the estimator for 20 000 generated samples from the embedded DTMC. The deviation was estimated in both cases from 100 independent simulation runs. Also, in both cases the processes were scaled in such a way that the average service rate $\mu = 1$ and the average arrival rate $\lambda = \rho$. The results are shown in Table 4.4.

Example	Class	Weighted Samples	Arrival Subchain	Class-k Arrival Subchain
1	1	0.0023	0.0034	0.0036
1	2	0.0044	0.0062	0.0085
2	1	0.0033	0.0036	0.0039
2	2	0.0062	0.0067	0.0067

Table 4.4: The standard deviation of the estimators for different embedded DTMC methods

The results, again, clearly show that for a given computational effort, i.e. the number of samples generated from the embedded DTMC, the variance is smaller when using the weighted samples method than for both subchain methods. Also, the complete arrival subchain appears to be more effective than the class-k arrival method, which can be again at least partly explained by noting that in the class-k arrival method we use only a subset of the samples used for the estimator of the arrival method. In addition, the arrival subchain method appears to be the better the smaller is the offered load of the traffic class under study with respect to the other traffic class. Also, we can note that the differences seem to become smaller as the system size is increased.

As a conclusion from the experiments made in this section, we can say that from all the Markov chain methods the simulation of the partial chain with the weighted samples method gives the best variance performance. However, at the same time, it seems that the differences may not be so substantial as the system size increases.

4.5.3 Correlation of Sample Generation Methods

In this section we investigate the correlation structure of different sample generation methods when estimating the blocking probabilities. Specifically, we consider the estimator

$$\hat{B}_k = \frac{1}{N}\sum_{n=1}^{N} h(\mathbf{X}_n) = \frac{1}{N}\sum_{n=1}^{N} 1_{\mathbf{X}_n \in \mathcal{B}^k},$$

where the samples $\mathbf{X}_n$ have the distribution π as in (2.1) and $h(\mathbf{X}) = 1_{\mathbf{X}\in\mathcal{B}^k}$. The following sample generation methods are studied:

- the rejection sampling method as described in section 4.2,
- the subchain method corresponding to all arrival events (case “d1” in the figures)
- the class-k subchain method corresponding to arrival events of class k (case “d2” in the figures),
- the weighted samples method with full jump chain (case “d3” in the figures), and
- the Gibbs sampler (case “g” in the figures).

The rejection sampling method, as has been noted before, gives independent samples and hence by (4.3) the most efficient samples in terms of the total variance of the estimator. Thus we only need to study the DTMC methods and the Gibbs sampler. For this we consider some numerical examples and try to infer from them some remarks on the efficiency of the methods.

Our first two examples consist of the same simple two traffic examples that were used in the previous section, i.e. :

- Example 1: $C = 10, \rho = [2, 1], \mathbf{b} = [1, 2]$

- Example 2: $C = 50, \rho = [20, 10], \mathbf{b} = [1, 2]$

The idea is to show how the increase of traffic intensity affects the correlation of the DTMC methods for the same type of system. For this, we have plotted the estimated covariance function for traffic class 2, i.e. $\mathrm{Cov}[h(\mathbf{X}_n), h(\mathbf{X}_{n+m})]$ for $m = 0, \ldots, 20$, for the four methods in Fig. 4.4, where the figure on the left corresponds to Example 1 and the figure on the right corresponds to Example 2. Note that for $m = 0$ the covariance is simply the variance of each sample in the estimator. From the figures we can first see that the variance of each sample is lowest for the weighted samples method and that this advantage appears to diminish as the size of the system is increased. The highest covariance between samples is for the weighted samples method (especially in Example 2), which is also intuitively clear since in this method we use every state generated from the jump chain as samples. The covariance of the arrival subchain method is quite close to the covariance of the weighted samples method in Example 1 for $m \geq 4$, but in Example 2 the arrival subchain has a lower covariance for $m \geq 2$. The class-k subchain method gives almost as good results as the Gibbs sampler in example 1, where the traffic intensity is low. However, when intensity is increased, the covariance is increased for the subchain method as well, but the Gibbs sampler's performance is not affected so much by the increase in traffic intensity. In these cases the correlation of the Gibbs sampler becomes practically negligible after 2 samples, i.e. a full cycle.

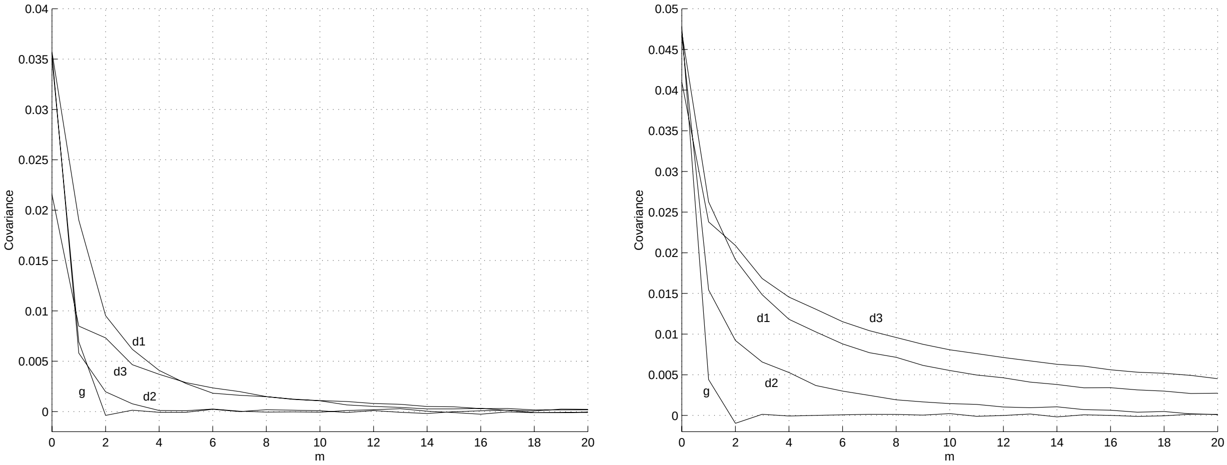

Figure 4.4: Covariance function of $h(\mathbf{X}_n)$ for Example 1 and Example 2. In the figure "g" corresponds to the Gibbs sampler method, "d1" to the arrival subchain method, "d2" to the class-k arrival subchain method and "d3" to the weighted samples method.

Next we tested how the methods perform when we have a larger system with multiple traffic classes and several links. For this, we use the four link star network with 12 traffic classes studied by Ross in [Ros95, chap. 6] in the heavy load case (Example 3). In Fig. 4.5 we have, again, plotted the covariance function for the estimator for traffic class 2 (left figure) and traffic class 8 (right figure), for $m = 0, \ldots, 30$. Again, we can see that the difference in the variance of each sample between the weighted samples method and the other methods is practically negligible and that the covariance of the weighted samples method is highest. Interestingly, we can also see that the covariance drops faster for the class-k subchain method than for the Gibbs sampler. This can, however, be understood by noting that in this case a lot of transitions have to be generated from the DTMC between successive class-k arrivals and hence the covariance between successive such points is also quite low. In the case of the Gibbs sampler, it seems that the length of the "cycles" K, produces cyclical behavior in the covariance plots, too. This can be seen in the right hand figure, where the traffic is "super heavy" and the blocking probability is approximately 23%. There two cycles are clearly visible — the first one for lags $m = 0, \ldots, 12$ and the second one for lags $m = 12 \ldots, 24$.

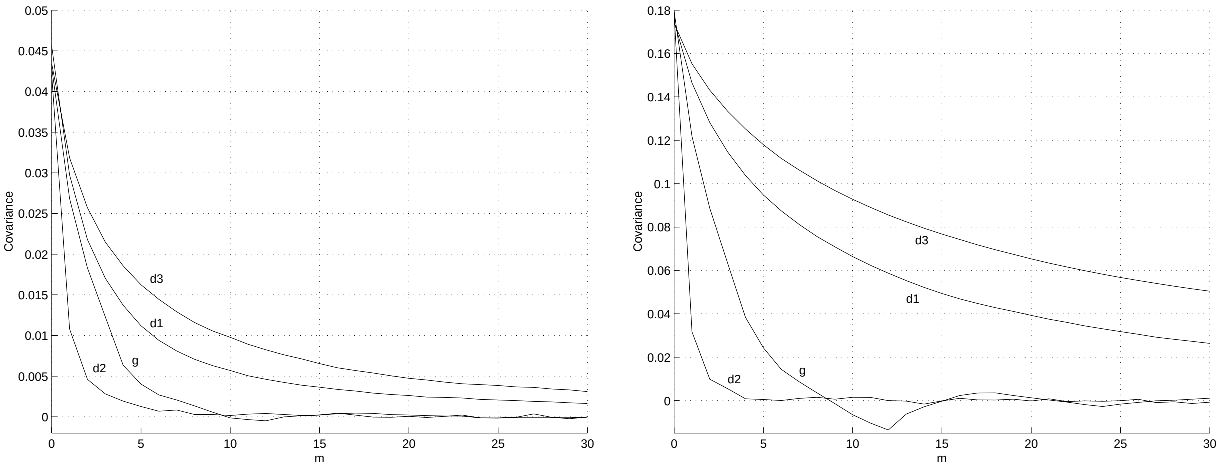

Figure 4.5: Covariance function of $h(\mathbf{X}_n)$ for Example 3. In the figure "g" corresponds to the Gibbs sampler method, "d1" to the arrival subchain method, "d2" to the class-k arrival subchain method and "d3" to the weighted samples method.

Having now examined the covariance of the samples for each method we can make some comments on the complexity of each method. The lowest complexity is for the full jump chain method with weighted samples. In this method the complexity of each sample generation is $O(K)$, since at each state there is at most $2 \cdot K$ possible jump directions and the determination of which event takes place can be done in $O(K)$ time. The computational work is, however, increased by the fact that at each state one has to also calculate the different state dependent branching probabilities, for which precomputation is not feasible because of excessive memory needs.

The simulation of the arrival subchain method consists of using the full jump chain, but only picking the suitable samples. Hence, the computational complexity is very much affected by the mean number of transitions between successive arrivals (or departures). We can make a rough approximate analysis on this in the following way. Assuming an infinite capacity the system behaves as K independent Poisson processes each with offered load $\rho_k = \lambda_k/\mu_k$. Then we have for each class on the average λ_k arrivals per time unit. Since the system is assumed infinite, every arriving customer enters the system and leaves the system as well, i.e. we have also on the average λ_k departures per time unit for each traffic class. Thus we can say that on the average half of the events are arrivals and half are departures, i.e. on the average 2 transitions are required between each arrival from the full jump chain giving a rough complexity of $O(2K)$ for the arrival subchain method, since each transition can be generated in $O(K)$ time.

For the class-k arrival subchain method we use the same assumptions as above and we can say that we have in the continuous time process events (arrivals or departures) taking place at the rate $2 \cdot \sum_k \lambda_k$ per time unit. Note that this is an upper bound on the rate, because in the finite system not all arrivals get service. However, the average time between arrivals of traffic class k is $1/\lambda_k$. Then we have on the average during that time

$$\left(\sum_{m=1}^{K} 2\, \lambda_m - \lambda_k \right) \cdot \frac{1}{\lambda_k} = O(K)$$

other events than arrivals from traffic class k (total average number of events precluding the arrivals from traffic class k). Thus we can deduce that the complexity of the class-k arrival subchain method is of the order $O(K^2)$, since we have $O(K)$ events between successive arrivals of traffic class k and each event generation can be done in $O(K)$ time.

In the rejection sampling method we need to generate K Poisson type random variables from distributions with length $N_{\max}^k$. Therefore, it has complexity $O(KN_{\max})$, where $N_{\max}$ denotes the largest value of $N_{\max}^k$ for $k = 1, \ldots, K$. However, by using advanced methods for generating the K random variables, the complexity can be decreased to mere $O(K)$ [Ros95, p. 233] ($O(1)$ for each traffic class). Despite its higher complexity in comparison with the DTMC methods, the rejection sampling method has the advantage that the distributions from which the samples are generated can be precomputed and stored into arrays. The Gibbs sampler basically consists of generating samples from univariate Poisson distributions with different "lengths" and the distributions can be, again, precomputed and stored into arrays. Hence the complexity of the method is $O(N_{\max})$, but by using similar methods as for the rejection sampling method, the lookup time from the arrays can be reduced to $O(1)$.

4.5.4 Bias Analysis of the Regenerative Simulation Method: The Single Link and Single Traffic Class Case

In this section we address the problem of finding the most efficient regeneration state for simulating the classical single link Erlang model. For this, we derive a method for analyzing the estimator's distribution as a function of the number of simulation cycles for any regeneration state choice. This allows us to examine in detail the effect of the choice of the regeneration state on the accuracy of the estimator in terms of its expected value and standard deviation.

Consider the single link, single traffic type case. Let

- n = the current state of the system,

- N = the size of the system (number of trunks),
- ρ = the offered traffic to the link.

The regenerative method for simulating the blocking probability of this system is as follows: Define a regenerative cycle as a path generated from the embedded DTMC, which starts from the regeneration state and ends there. Along the generated path we collect samples of two random variables: G, the number of arrivals (including blockings) in a cycle, and F, the number of blockings in a cycle. The observations of these variables in m^{th} cycle are denoted as F_m and G_m. The simulation is stopped after M cycles. Then our M-cycle estimator for the blocking probability $\hat{B}_M$ becomes

$$\hat{B}_M = \frac{1/M \sum_{m=1}^{M} F_m}{1/M \sum_{m=1}^{M} G_m} = \frac{\hat{F}_M}{\hat{G}_M} \tag{4.12}$$

Derivation of the Distribution

For deriving the probability generating function (pgf) of the estimator (4.12), let z be the variable in the pgf associated with arrivals, and y with blockings. The joint probability of $G = k$ and $F = l$ is given by

$$p(k,l) = \begin{cases} \Pr[G = k, F = l \mid\uparrow] \Pr[\uparrow] + \Pr[G = k \mid\downarrow] \Pr[\downarrow], & l = 0, \\ \Pr[G = k, F = l \mid\uparrow] \Pr[\uparrow], & l > 0, \end{cases}$$

where ↑ and ↓ denote cycles which start from the regeneration state and on the next transition proceed upwards and downwards respectively. Notice that when the cycle proceeds downwards from the regeneration state then we will of course have no blockings. The pgf for the joint probability $p(k,l)$ is defined as

$$g(z,y) = \sum_{k,l} p(k,l) z^k y^l. \tag{4.13}$$

Now, to facilitate the analysis we will divide the analysis into two parts: 1) for cycles that proceed upwards from the regeneration state and 2) for cycles that proceed downwards, respectively. For this, we denote with $G_\downarrow^n$ and $G_\uparrow^n$ the number of arrivals during a cycle that proceeds downwards and upwards, respectively, starting from state n. For F there is only need for $F_\uparrow^n$ to be defined. We also introduce similar notation for the pgf's of the random variables: $g_\uparrow^n(z,y)$ and $g_\downarrow^n(z)$ denote the pgf's for cycles proceeding upwards or downwards from state n. Finally, we shall denote with $p_\downarrow^n = n/(\rho+n)$ and $p_\uparrow^n = \rho/(\rho+n)$ the probabilities for moving downwards or upwards, respectively, from state n.

From Fig. 4.6 it can be seen that the $G_\uparrow^n$ and $F_\uparrow^n$ have a recursive structure such that for the cycles proceeding upwards from state n, the number of arrivals consists of the one coming from the transition upwards to state $n+1$ plus a random sum of arrivals from cycles beginning from and ending in state $n+1$. Now in this case, the number of such

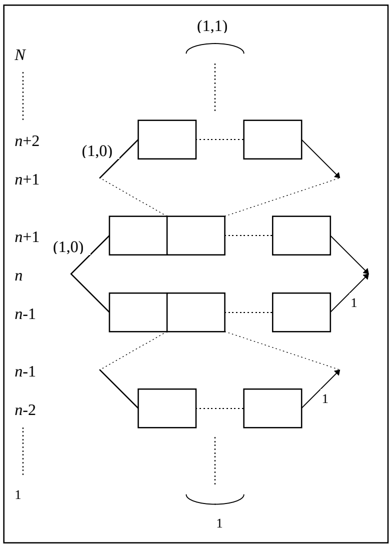

Figure 4.6: Recursive regeneration cycles.

cycles, denoted with $\xi_\uparrow^{n+1}$, is geometrically distributed with success probability $p_\downarrow^{n+1}$. For the blockings the reasoning is the same, but there are no blockings until for the cycle beginning from state N and ending there, in which case the cycle contains exactly one arrival and blocking. Then we can derive the following recursive equations for the $G_\uparrow^n$ and $F_\uparrow^n$

$$\begin{cases} (G_\uparrow^n, F_\uparrow^n) = (1,0) + \sum_{i=1}^{\xi_\uparrow^{n+1}} (G_{\uparrow,i}^{n+1}, F_{\uparrow,i}^{n+1}), \quad n < N, \\ (G_\uparrow^N, F_\uparrow^N) = (1,1), \\ \xi_\uparrow^n \sim \text{Geom}(p_\downarrow^n). \end{cases} \tag{4.14}$$

Note that when $\xi_\uparrow^{n+1} = 0$ (with probability $p_\downarrow^{n+1}$), i.e. when there is an arrival and a departure immediately after that the sum term vanishes.

Similarly we get for the number arrivals during cycles which proceed downwards (remember there are no blockings in these cycles)

$$\begin{cases} G_\downarrow^n = 1 + \sum_{i=1}^{\xi_\downarrow^{n-1}} G_{\downarrow,i}^{n-1}, \\ G_\downarrow^1 = 1, \\ \xi_\downarrow^n \sim \text{Geom}(p_\uparrow^n). \end{cases} \tag{4.15}$$

Now, in order to get the pgf for the above equations, we use the fact that for the sum of i.i.d. variables B_i with common pgf $B(z)$, $A = B_1 + \cdots + B_C$, where C itself is a random variable with pgf $C(z)$ and is independent of the B_i, has the pgf $A(z) = C(B(z))$. Specifically, in the case where $C \sim \text{Geom}(p)$, the pgf is $A(z) = \frac{p}{1-(1-p)B(z)}$. Then we will get the following recursive equations for the pgf:s

$$\begin{cases} g_{\uparrow}^{n}(z,y) = z\dfrac{p_{\downarrow}^{n+1}}{1-p_{\uparrow}^{n+1}g_{\uparrow}^{n+1}(z,y)}, \\ g_{\downarrow}^{n}(z) = z\dfrac{p_{\uparrow}^{n-1}}{1-p_{\downarrow}^{n-1}g_{\downarrow}^{n-1}(z)}, \\ g_{\uparrow}^{N}(z,y) = zy, \\ g_{\downarrow}^{1}(z) = z. \end{cases} \tag{4.16}$$

Then the complete pgf $g^n(z,y)$ for a simulation cycle starting from state n and ending there is expressed as

$$g^n(z,y) = p_{\uparrow}^{n}g_{\uparrow}^{n}(z,y) + p_{\downarrow}^{n}g_{\downarrow}^{n}(z). \tag{4.17}$$

Because the random variables in different regeneration cycles are i.i.d., the probability generating function for the joint probability distribution of the arrivals and blockings during M cycles will simply become

$$g_M^n(z,y) = (g^n(z,y))^M. \tag{4.18}$$

Mean and Variance of the Estimator

In order to calculate the mean of the estimator (4.12) and its variance, we utilize some properties of the pgf. Here we simplify the notation a little bit in order to avoid confusing and overly complex notation. We simply denote with $g(z,y)$ the pgf obtained for the estimator (4.12) through equations (4.16) - (4.18), i.e. we omit the dependence on M and n. Also, $g^{(i,j)}(z,y)$ is used to denote the i^{th} and j^{th} partial derivatives with respect to z and y.

Then for the numerator part, i.e. random variable F, we shall need the following relations

$$\begin{aligned} \mathrm{E}\left[F\right] &= \sum_{k,l} l\ p(k,l) = \frac{\partial}{\partial y}\left(\sum_{k,l} p(k,l)z^k y^l\right)\Bigg|_{z=1,y=1} \\ &= g^{(0,1)}(1,1), \end{aligned} \tag{4.19}$$

$$\begin{aligned} \mathrm{E}\left[F^2\right] &= \sum_{k,l} l^2\ p(k,l) = \sum_{k,l} l(l-1)\ p(k,l) + \sum_{k,l} l\ p(k,l) \\ &= \frac{\partial}{\partial y^2}\left(\sum_{k,l} p(k,l)z^k y^l\right)\Bigg|_{z=1,y=1} + \frac{\partial}{\partial y}\left(\sum_{k,l} p(k,l)z^k y^l\right)\Bigg|_{z=1,y=1} \\ &= g^{(0,2)}(1,1) + g^{(0,1)}(1,1). \end{aligned} \tag{4.20}$$

For calculating the statistics of the arrivals G, we need to be able to evaluate the following

$$\mathrm{E}\left[\frac{1}{G}\right] = \sum_{k}\frac{1}{k}\sum_{l} p(k,l)$$

$$
\begin{aligned}
&= \int_0^1 \frac{g(z,1)}{z}\,dz, && (4.21)\\
\mathrm{E}\left[\frac{1}{G^2}\right] &= \sum_k \frac{1}{k^2}\sum_l p(k,l) \\
&= \int_0^1 \frac{1}{z}\int_0^z \frac{g(u,1)}{u}\,du\,dz \\
&= \int_0^1 \frac{1}{u}\,du \int_u^1 \frac{1}{z} g(u,1)\,dz \\
&= -\int_0^1 \frac{\ln u}{u} g(u,1)\,du, && (4.22)
\end{aligned}
$$

where in equation (4.22) we have changed the order of integration to simplify the integration to a single integral.

Eq. (4.21) can be proved quite easily. The pgf of the arrivals G is derived from the total pgf $g(z,y)$ as $g(z,1) = \sum_k z^k \sum_l p(k,l) = \sum_k p^*(k) z^k$, where we have denoted with $p^*(k)$ the marginal probabilities of G. Also, note that $p^*(0) = 0$. Then we have

$$
\begin{aligned}
\int_0^1 \frac{g(z,1)}{z}\,dz &= \int_0^1 \left(\sum_{k=1}^{\infty} p^*(k) z^{k-1}\right) dz \\
&= \sum_{k=1}^{\infty} \frac{1}{k}\,p^*(k).
\end{aligned}
$$

Eq. (4.22) can be proved in a similar fashion by only doing the integration twice. Then by using methods as in equations (4.19) - (4.22) we can derive the following

$$
\begin{aligned}
\mathrm{E}\left[\hat{B}_M\right] &= \mathrm{E}\left[\frac{F}{G}\right] = \int_0^1 \left(\frac{1}{z} g^{(0,1)}(z,1)\right) dz, && (4.23)\\
\mathrm{E}\left[\hat{B}_M^2\right] &= \mathrm{E}\left[\frac{F^2}{G^2}\right] = -\int_0^1 \left(\frac{\ln z}{z} g^{(0,2)}(z,1) + g^{(0,1)}(z,1)\right) dz. && (4.24)
\end{aligned}
$$

Now, by using equations (4.23) and (4.24) we are able to calculate the mean and variance of the distribution for estimator (4.12) after M simulation cycles.

Numerical Example

In this section we present the results of using equations (4.16) - (4.18) and then evaluating equations (4.23) and (4.24) on a system with $N = 6$ and $\rho = 2$. The exact blocking probability is $\mathrm{erl}(2,6) = 0.0121$ in this case. To compare the effect of different regeneration states, we must be able to compare the mean and variance of the estimator for equal number of generated events from the DTMC of the process. However, equations (4.23) and (4.24) only give us values as a function of the number of simulated cycles.

For this we notice that the mean number of arrivals in a cycle is given by

$$\mathrm{E}\left[G\right] = g^{(1,0)}(1,1). \tag{4.25}$$

Also, we know that on the average the number of departures during a cycle is $(1-B)\mathrm{E}[G]$, i.e. when the blocking probability becomes smaller the closer the average number of arrivals during a cycle is to the average number of departures. Thus, it is sufficient to compare the efficiency of the estimator for different starting states as a function of the mean number of arrivals needed to achieve certain precision. Therefore, when comparing the estimators we just need to evaluate (4.23) and (4.24) until a sufficient number of cycles, so that for each regeneration state, the mean number of arrivals needed to achieve this accuracy is enough.

In Fig. 4.7 we have plotted the results for all possible regeneration states as a function of the number simulation cycles. However, the x-axis has been scaled to correspond to the average number of arrivals that need to be generated from the jump chain by using (4.25). In all the graphs, the lowest curve corresponds to the results for having the regeneration state as state 1, and the next upper curve corresponds to state 2 etc. In Fig. 4.7 on the left the horizontal line represents the true value of the blocking probability.

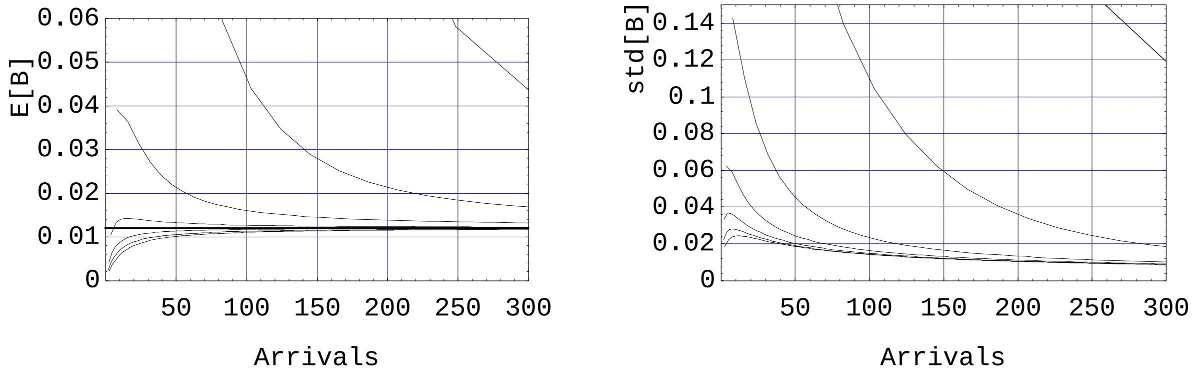

Figure 4.7: Expected value and the standard deviation of the estimator.

The curves clearly show that the chosen regeneration state affects the bias and the standard deviation of the etimator considerably. On the other hand they show rapid convergence as the number of arrivals increases, i.e. simulation time, showing good agreement with the asymptotic results. What is perhaps most surprising in the curves is that the standard deviation $\sigma_{\hat{B}_M}$ increases when the chosen regeneration state becomes higher.

4.5.5 Bias Analysis of the Regenerative Simulation Method: The General Multiservice Loss System Case

In the previous section we were able to derive an analytic expression for the pgf of the estimator as a function of the number of the regeneration cycles for the single link and single traffic type case. The crucial point was that the cycles had a recursive structure in the simple one-dimensional path state space. This property is lost when we are dealing with a multi-dimensional state space, since now the paths can wander in different directions in the state space. However, we can still derive the pgf for the cycles in the multi-dimensional space by using a Markov chain method.

Derivation of the PGF

The idea is based on the following property. Let $\mathbf{X}_n$ be an irreducible Markov chain with a multi-dimensional state space $\mathcal{S}$ and transition matrix $\mathbf{P}$. This transition matrix is constructed for the multi-dimensional process by counting the number of states in the state space and labeling each of them from 0 to n, where n equals one less than the total number of states in the whole state space. Then each entry p_{ij} in the matrix corresponds to the transition probability of moving from the state labeled i to the state labeled j. The states of the state space can of course be labeled in any order. Now, assume that we are interested in finding out the probability of starting the chain from some initial state "0" in the state space and that the process returns to this state in I transitions. Then by denoting with r all the other states than state 0, the transition matrix $\mathbf{P}$ has the following partition

$$\mathbf{P} = \begin{pmatrix} p_{00} & \mathbf{P}_{0r} \\ \mathbf{P}_{r0} & \mathbf{P}_{rr} \end{pmatrix}, \tag{4.26}$$

where $\mathbf{P}_{0r}$ and $\mathbf{P}_{r0}$ are vectors containing the transition probabilities away from state 0 and back to state 0, respectively, and the submatrix $\mathbf{P}_{rr}$ contains the transition probabilities between all the other states than state 0 (see Fig. 4.8).

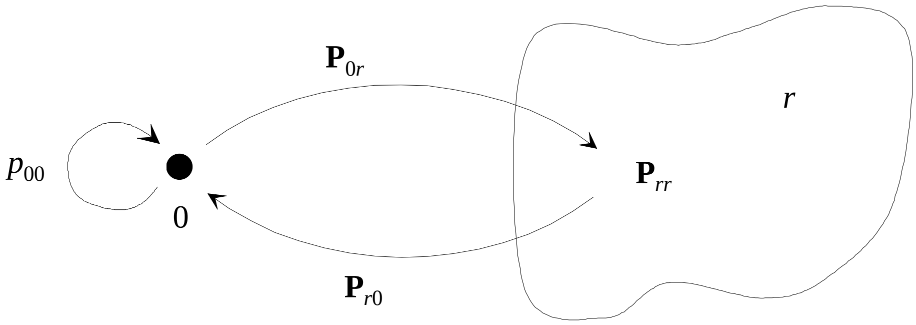

Figure 4.8: Partition of the transition matrix $\mathbf{P}$.

Using the partitioned matrix we note that the probability of re-entering the initial state 0 in i transitions is given by

$$\Pr\left[I = i\right] = \begin{cases} p_{00} & i = 1, \\ \mathbf{P}_{0r} \cdot \mathbf{P}_{rr}^{i-2} \cdot \mathbf{P}_{r0} & i > 1. \end{cases} \tag{4.27}$$

This can be used to construct the pgf for the estimator of the blocking probabilities in a multiservice loss system when using the regenerative simulation method. Then we are again able to use similar methods as in the previous section for investigating the bias and the standard deviation of the estimator as a function of increasing number of regeneration cycles.

To this end, let us now use $\mathbf{P}$ to denote the transition matrix corresponding to the full jump chain of the multiservice loss system. The regenerative simulation of the system consists of starting the chain from some initial state 0 until it returns back to that state. Along the generated path we collect samples of the random variables for the number of arrivals of each traffic class during a cycle, denoted by G_k^m, and for the number of blockings of each traffic class during a cycle, denoted by F_k^m. The simulation is stopped after M such cycles have been generated. The blocking probability for traffic class k is then estimated by

$$\hat{B}_k^M = \frac{\sum_{m=1}^M F_k^m}{\sum_{m=1}^M G_k^m}.$$

To obtain the pgf for this estimator, we first denote by $g(z_k, y_k)$ the pgf for the joint distribution of class-k arrivals and blockings during one regenerative cycle when the process is started from state 0 with z_k corresponding to the arrivals and y_k to the blockings. Now, $g(z_k, y_k)$ can be obtained using conditioning on the length of the cycles I by

$$g(z_k, y_k) = \sum_{i=1}^{\infty} g^i(z_k, y_k)\mathrm{Pr}\left[I = i\right], \tag{4.28}$$

where $g^i(z_k, y_k)$ is the conditional pgf for the arrivals and blockings of traffic class k when the length of the cycle equals i.

Using (4.27) we can obtain the probability of entering the regeneration state 0 in a given number of transitions. However, for the estimator we are only interested in the number of transitions corresponding to arrivals. To derive its pgf, we replace those transitions p_{ij} in $\mathbf{P}$ which correspond to an arrival of a class-k call with $z_k p_{ij}$ and those transitions corresponding to an arrival and a blocking transition with $z_k y_k p_{ij}$. If we then use (4.27) with the modified transition probabilities, it gives the i^{th} term in the summation of (4.28), i.e.

$$\mathbf{P}_{0r} \cdot \mathbf{P}_{rr}^{i-2} \cdot \mathbf{P}_{r0} = g^i(z_k, y_k)\mathrm{Pr}\left[I = i\right]. \tag{4.29}$$

Hence we are able to obtain an approximation for the one cycle pgf $g(z_k, y_k)$ by evaluating (4.29) until a sufficient number of terms from (4.28) have been included, e.g. until $\mathrm{Pr}[I > i] < 0.001$. The M cycle pgf $g_M(z_k, y_k)$ is by the independence of the cycles then simply

$$g_M(z_k, y_k) = g(z_k, y_k)^M. \tag{4.30}$$

Now, we can use the results from the previous section and use (4.23) and (4.24) to calculate the mean and variance of the estimator from

$$\mathrm{E}\left[\hat{B}_k^M\right] = \int_0^1 (\frac{1}{z_k}\left(g_M^{(0,1)}(z_k, 1)\right) dz, \tag{4.31}$$

$$\mathrm{E}\left[(\hat{B}_k^M)^2\right] = -\int_0^1 \left(\frac{\ln z_k}{z_k} g_M^{(0,2)}(z_k, 1) + g_M^{(0,1)}(z_k, 1)\right) dz, \tag{4.32}$$

where $g^{(i,j)}(z, y)$ is used to denote the i^{th} and j^{th} partial derivatives with respect to z and y, respectively.

Numerical example

As a numerical example we use a small tandem link network with two traffic classes and the following parameters:

- Link capacities $C_1 = C_2 = 5$
- Bandwidth requirements: $\mathbf{b}_1 = [2, 3]$, $\mathbf{b}_2 = [0, 3]$
- Offered load: $\rho_1 = \rho_2 = 0.1$

The network and the resulting state space are shown in Fig. 4.9.

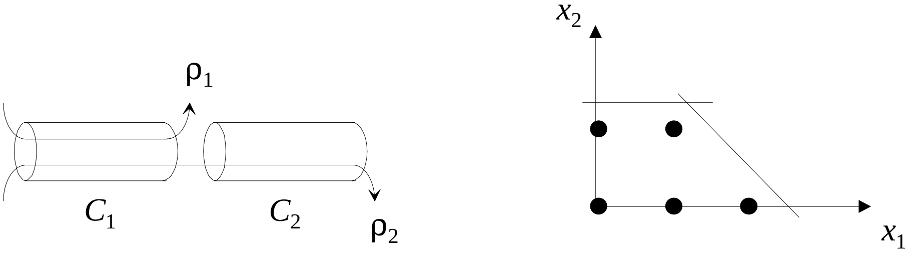

Figure 4.9: Network example and its state space.

As a first example consider the case of choosing state $[0, 0]$ as the regeneration state 0 and assume that we are interested in the results for traffic class 1. We approximated $g(z_1, y_1)$ by including the 13 first terms of the sum (4.28) resulting in a very accurate approximation since now $\Pr[I \leq 13] = 0.999984$. The average number of transitions to enter the regeneration state is from (4.27)

$$\mathrm{E}\left[I\right] = p_{00} + \sum_{i=2}^{\infty} i(\mathbf{P}_{rr} \cdot \mathbf{P}_{0r}^{i-2} \cdot \mathbf{P}_{r0}).$$

In this case $\mathrm{E}[I] \approx 2.36$. In Fig. 4.10 we can see the expectation and the standard deviation of the estimator for the blocking probability of traffic class 1. From the figure we can see that the estimator is strongly biased downwards in the beginning, but approaches the correct value ($B_1 = 0.0123$, straight line at the top of the figure) rapidly as the number simulated cycles increases. Also, aside for the slight increase when only a few cycles have been simulated, the standard deviation decreases rapidly as the number of cycles increases.

Next we consider the case of choosing state $[1, 0]$ as the regeneration state 0 and, again, assume that we are interested in the results for traffic class 1. We approximated $g(z_1, y_1)$ by

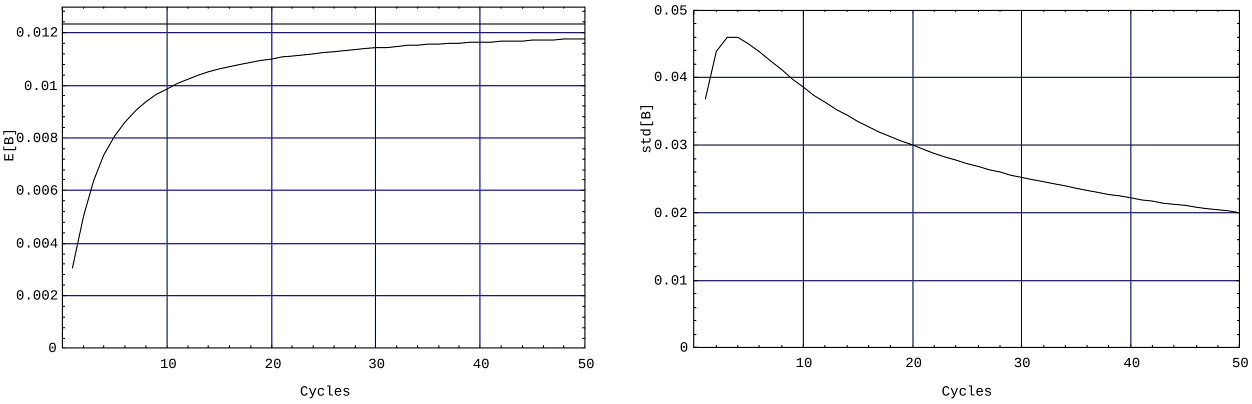

Figure 4.10: Expected value and the standard deviation of the estimator for state $[0, 0]$.

including the 17 first terms of the sum (4.28) resulting in a slightly less accurate approximation than before but still we have $\Pr[I \leq 17] = 0.995788$. In this case $\mathrm{E}[I] \approx 3.94$. In Fig. 4.11 we have the expectation and the standard deviation of the estimator for the blocking probability of traffic class 1. From the figure we can see that the estimator is strongly biased downwards in the beginning, but approaches the correct value (again the straight line at the top of the figure) rapidly as the number simulated cycles increases. Also, the standard deviation decreases rapidly as the number of cycles increases.

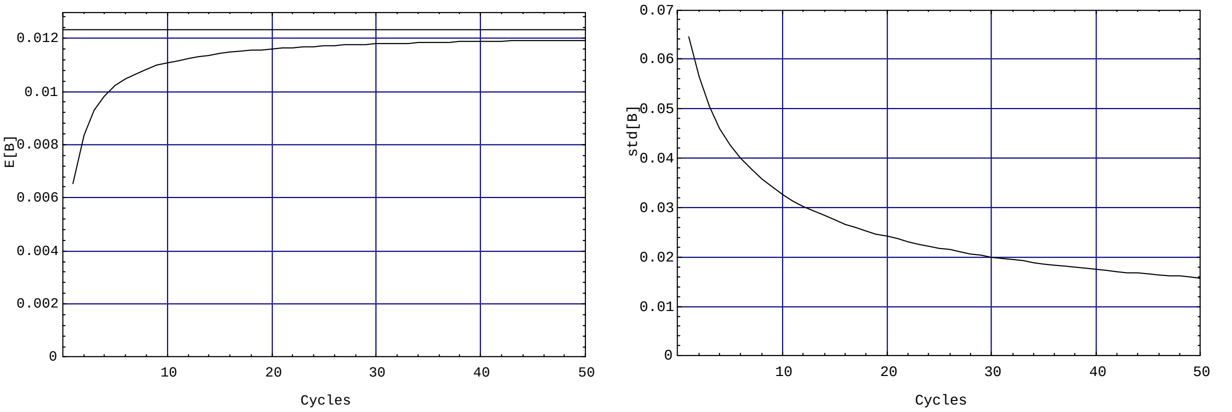

Figure 4.11: Expected value and the standard deviation of the estimator for state $[1, 0]$.

In comparison with the first example we can see from the figures that, for the same number of cycles, the choice $[1, 0]$ for the regeneration state would be better in terms of the bias and the standard deviation of the estimator. However, this is not the whole truth since the average length of a cycle for state $[0, 0]$ was only 2.36 whereas for state $[1, 0]$ it was 3.94. Hence, the average time to simulate a fixed number of cycles is for the state $[0, 0]$ only $2.36/3.94 \approx 60\%$ of the time it would take for the choice $[1, 0]$. When we take this scaling of the results into consideration the two alternatives become almost equally good.

Chapter 5

Simulation Speedup Techniques

In this chapter we first review briefly known general techniques from the simulation literature for increasing the simulation efficiency, i.e. for reducing the variance of the estimator. Then a review is given on the material available on importance sampling applied in communication system simulation. Finally, we give two novel variance reduction methods in the context of the multiservice loss system.

5.1 General Variance Reduction Methods

In general, obtaining results with simulation requires a large amount of computational work and, hence, time. Also, as we know from the central limit theorem, no matter what kind of simulation method is used, the width of the confidence interval of the results depends on the standard deviation of the result, which, in turn, is inversely proportional to the square root of the sample size. Therefore, if one is somehow able to obtain samples which have a smaller variance, this will result in a more accurate estimate with a given number of samples. If the reduction of variance can be obtained without causing excessive additional computational complexity, the savings in simulation run time can be significant. Such methods are generally known as *variance reduction techniques*, and they are based on utilizing known analytical results and/or properties of the system under study.

Next we briefly describe the following methods: common random numbers, antithetic variates, control variables, conditional expectations and importance sampling (IS). An early work reviewing these methods, except the common random numbers method, is [Ham67, chap. 6] describing their usage in the context of Monte Carlo simulation. More recent reviews can be found e.g. in [Mitr82, chap. 6], [Law91, chap. 11] and [Rub98, chap. 4].

5.1.1 Common Random Numbers

This method is applicable when comparing the performance of two similar systems. For example we might be interested in the performance difference of a queue under two different scheduling policies. To be precise, assume we have a performance measure X and we want to estimate the difference in the expectation of X under these two different policies

$$\Delta m = m_A - m_B = \mathrm{E}\left[X^A\right] - \mathrm{E}\left[X^B\right],$$

where X^A and X^B denote the r.v. X under policy A, and B, respectively. To estimate the difference Δm it is advantageous to use the same realizations of the input process, i.e. the same random numbers are used to generate the input process realization for policy A as for policy B, since if e.g. under policy A there appears an arrival burst, the same burst will also appear in the simulation under policy B. During a simulation we obtain N samples of X_n^A and X_n^B, $n = 1, \ldots, N$. Let $\bar{X}^A = 1/N \sum_{n=1}^N X_n^A$ denote the sample mean for policy A and $\bar{X}^B$ the corresponding variable for policy B. Then

$$\Delta \hat{m} = \bar{X}^A - \bar{X}^B \tag{5.1}$$

is still an unbiased estimator for Δm. However, the variance of this is

$$\mathrm{Var}\left[\Delta \hat{m}\right] = \mathrm{Var}\left[\bar{X}^A\right] + \mathrm{Var}\left[\bar{X}^B\right] - \mathrm{Cov}\left[\bar{X}^A, \bar{X}^B\right].$$

Now the samples $X_1^A, \ldots, X_N^A$ are independent and so are $X_1^B, \ldots, X_N^B$, but when using the same arrival realizations the X_n^A and X_n^B, $n = 1, \ldots, N$ are *positively* correlated. Hence, the sample averages $\bar{X}^A$ and $\bar{X}^B$ are positively correlated and the estimator (5.1) has lower variance than when using independent arrival realizations.

In general, it can be noted that when the performance of any two systems are compared, it is always advantageous to try to make the comparison under as identical circumstances as possible. However, the implementation of the common random numbers is not always so straight forward and may require some effort in the synchronization of the simulations of the alternative systems.

5.1.2 Antithetic Variates

Let us again consider the r.v. X and estimating its expectation $m = \mathrm{E}[X]$. Instead of just generating independent samples of X during the simulation the idea is now to generate pairs of samples such that if either value of a pair happens to have a large value the other value will tend to have a small value and vice versa. This will induce negative correlation between the pairs of samples. Such a pair is called a pair of antithetic random variables. Let us denote by $(X_n^{(1)}, X_n^{(2)})$ the n^{th} such pair of samples and during the simulation N such pairs are generated. Also, we assume that the different pairs are independent and that $X_n^{(1)} \sim X_n^{(2)} \sim X$. Then let

$$X_n = \frac{X_n^{(1)} + X_n^{(2)}}{2}.$$

Now $\bar{m} = 1/N \sum_{n=1}^{N} X_n$ is an unbiased estimate for m, but its variance is

$$\mathrm{Var}\left[\bar{m}\right] = \frac{\mathrm{Var}[X_n^{(1)}] + \mathrm{Var}[X_n^{(2)}] + \mathrm{Cov}[X_n^{(1)}, X_n^{(2)}]}{4N}.$$

From this we can see that, if we manage to induce negative correlation between $X_n^{(1)}$ and $X_n^{(2)}$, we have smaller variance for $\bar{m}$ than when they are independent.

To create the negative correlation we try to choose the sequences of the $U_k \sim \mathrm{U}(0,1), (k = 1, \ldots)$ distributed random variables driving the simulation runs appropriately. In practice this means that if some r.v. U_k is used for some specific purpose in generating $X_n^{(1)}$ (typically to generate some r.v. using e.g. the so called inverse transform method, see e.g. [Rub98, p. 18] for details on this method), the r.v. $1 - U_k$ is used in the generation of $X_n^{(2)}$ for exactly the same purpose. Then, if U_k contributes to a large value of $X_n^{(1)}$, $1 - U_k$ contributes to a small value of $X_n^{(2)}$. In general, it is, however, difficult to quantify the amount of variance reduction obtained with this method and, moreover, the reduction is not even guaranteed in all cases. Also, the application of this method requires careful synchronization between the generation of the $(X_n^{(1)}, X_n^{(2)})$ pairs to ensure that the random number sequences $U_1, U_2, \ldots$ and $1 - U_1, 1 - U_2, \ldots$ are used for exactly the same purpose.

5.1.3 Control Variables

The basic control variables method can be described in the following way. The key idea is to utilize strong positive correlation between the r.v. under study X and another so called control variable Y for which the expectation $y = \mathrm{E}[Y]$ is assumed to be known. Again the goal is to estimate $m = \mathrm{E}[X]$. During the simulation we generate samples of both X and Y. The samples X_n and Y_n can be used to construct an estimate with lower variance. Consider the following estimator

$$V = \bar{X} - \bar{Y} + y,$$

where $\bar{X}$ and $\bar{Y}$ denote sample averages. This is clearly an unbiased estimator and its variance is

$$\mathrm{Var}\left[V\right] = \mathrm{Var}\left[\bar{X}\right] + \mathrm{Var}\left[\bar{Y}\right] - 2\mathrm{Cov}\left[\bar{X}, \bar{Y}\right].$$

Thus the variance is smaller than $\mathrm{Var}[\bar{X}]$ if $\mathrm{Cov}[\bar{X}, \bar{Y}] > \mathrm{Var}[\bar{Y}]/2$.

This basic method may be improved by including a free parameter a in the definition of V, i.e. we let

$$\tilde{V} = \bar{X} - a(\bar{Y} - y).$$

The variance of this estimator has a quadratic form with respect to the parameter a and from that it is easily computed that the minimum variance choice for the parameter a is given by $a^* = \mathrm{Cov}[\bar{X}, \bar{Y}]/\mathrm{Var}[\bar{Y}]$. Then the variance of $\tilde{V}$ is

$$\mathrm{Var}\left[\tilde{V}\right] = (1 - \rho_{\bar{X},\bar{Y}}^2)\mathrm{Var}\left[\bar{X}\right],$$

where $\rho_{\bar{X},\bar{Y}}$ denotes the cross correlation of $\bar{X}$ and $\bar{Y}$. From this it can be readily seen that the variance is always reduced irrespective of the sign of the correlation and that the larger $|\rho_{\bar{X},\bar{Y}}|$ is, the greater the variance reduction. However, in many cases the covariance cannot be analytically computed and, hence, it must be estimated prior to the actual simulation. In general, this method already applies the principle that if any analytical results are available of the system under study that are directly or indirectly related, this prior information can be used to construct more efficient estimators.

We can note here that the use of control variables requires that during the simulation both the actual observed r.v. X and the control r.v. Y are simulated. Henderson [Hen97] has recently published a Ph.D. thesis on a related method, whereby one approximates the r.v. X by another r.v., say X'. Then during the simulation only X' is simulated. It turns out that sometimes X' can be chosen such that $\mathrm{E}[X'] = \mathrm{E}[X]$ but $\mathrm{Var}[X'] < \mathrm{Var}[X]$. Henderson considers specifically the case when X and X' are Markov chains.

5.1.4 Conditional Expectations

In this method we utilize known analytical results of the simulated system in a similar manner as before, but this time we replace some r.v. with the value of its conditional expectation. Again we are estimating $m = \mathrm{E}[X]$. Let us assume now that we are able to calculate analytically the conditional expectation $Z = \mathrm{E}[X \mid Y]$, which is also a r.v. whose value changes as Y changes, but when Y is fixed Z has a fixed value. Then during the simulation we generate samples Z_n, instead of samples X_n. Note that then each sample is still an unbiased sample of X since

$$\mathrm{E}[Z] = \mathrm{E}[\mathrm{E}[X \mid Y]] = \mathrm{E}[X].$$

Recall the formula for the unconditional variance in terms of conditional variance and expectation

$$\mathrm{Var}[X] = \mathrm{E}[\mathrm{Var}[X \mid Y]] + \mathrm{Var}[\mathrm{E}[X \mid Y]],$$

from which follows that

$$\mathrm{Var}[Z] = \mathrm{Var}[\mathrm{E}[X \mid Y]] = \mathrm{Var}[X] - \mathrm{E}[\mathrm{Var}[X \mid Y]]$$

showing that the conditioning eliminates the variance of X for all those values of X where the value of Y is fixed. This means that by conditioning we always reduce the variance of each sample.

However, since our final estimator is the sample average $\bar{Z} = 1/N \sum_{n=1}^{N} Z_n$, the reduction of the variance for the sample average is only guaranteed when the samples Z_n are independent. When the samples are positively correlated, which is quite typical e.g. when using process simulation, the reduction is not guaranteed, since it may happen that the expectations will increase the positive correlation such that the variance reduction of the samples is canceled.

We can here also note that the weighted samples method presented in the previous chapter can be interpreted as resulting from the application of this method. By using the expected values of the lifetimes of each state we are using as samples the conditional expectation for the lifetime of the state conditioned on the embedded DTMC being in a given state. Later in this chapter we will show how this method can be applied in the case of the multiservice loss system to give substantial variance reduction.

5.1.5 Importance Sampling

Consider again the following estimation problem. Let $p(x)$ denote the density of the r.v. X obtaining values in the set $\mathcal{S}$ and we are interested in estimating the expectation $\mathrm{E}[f(X)]$. We can express the expectation

$$m = \mathrm{E}\left[f(X)\right] = \sum_{x\in\mathcal{S}} f(x)\, p(x) = \sum_{x\in\mathcal{S}} f(x)\, \frac{p(x)}{q(x)}\, q(x) = \mathrm{E}_q\left[f(X)\, w(X)\right], \qquad (5.2)$$

where $q(x)$ is some other density defined in the set $\mathcal{S}$ and in addition having the property that $q(x) > 0, \forall x \in \mathcal{S} : p(x) > 0$. Also, E_q denotes expectation with respect to the density $q(x)$ and $w(x) = p(x)/q(x)$ is the so called likelihood ratio. This leads to the simulation method known as importance sampling. We generate samples X_n, $n = 1, \ldots, N$, using the density $q(x)$. Then from (5.2) an unbiased estimator for m is

$$\hat{m} = \frac{1}{N} \sum_{n=1}^{N} f(X_n) w(X_n).$$

That is we can estimate m by using another density and then unbiasing the result by multiplying the samples with the likelihood ratio.

Then the problem is to find among the class of all possible densities the one that will produce minimum variance for the estimator (5.2). A well known result for the optimal choice of $q(x)$, i.e. giving the minimum variance for the estimator, is to set

$$q(x) = \frac{f(x)\, p(x)}{m}.$$

This choice would make the variance of $\hat{m}$ zero, since it has the property of making each sample $f(X_n)w(X_n) = m$ with probability 1. However, the optimal choice is of course not a feasible density, since it requires knowledge of the unknown quantity m that we are attempting to estimate. Then the next approach is to try to approximate the optimal choice in some way. This has been subject to a considerable amount of research and the results that apply to simulation of queuing systems will be presented in the next section. Also, later in this chapter we will derive a composite IS distribution for estimating the blocking probabilities in the multiservice loss system, with the guideline of trying to approximate the optimal distribution.

5.2 Rare Event Simulation

Next we will review the literature on variance reduction in the simulation of so called rare events. This particular class of problems has attracted a lot of attention in the research community and is perhaps the one for which the theory has been most throughly developed. For this let us first define the rare event simulation problem. Let X be a r.v. with density $p(x)$ and consider estimating the probability γ that X is in some set A, i.e. as in (5.2) with $f(X) = 1_{X \in A}$,

$$\gamma = \mathrm{E}_p \left[1_{X \in A} \right]. \tag{5.3}$$

The static MC method to estimate the probability is to generate N i.i.d. samples of X from $p(x)$ and use the estimator $\hat{\gamma}_N = 1/N \sum_{n=1}^{N} 1_{X_n \in A}$, which has the variance $\sigma^2(\hat{\gamma}_N) = \gamma(1-\gamma)/N$. Let $RE(\hat{\gamma}_N) = \sigma(\hat{\gamma}_N)/\gamma$ denote the relative error of the estimator. Then

$$RE(\hat{\gamma}_N) = \sqrt{\frac{1-\gamma}{\gamma N}} \approx \frac{1}{\sqrt{\gamma N}} \to \infty \quad \text{when } \gamma \to 0.$$

Thus when using standard simulation the relative error is unbounded as the event becomes rarer. This also implies that in order to get an estimate with fixed relative error the required number of samples N to reach this accuracy goes to infinity as $\gamma \to 0$. Srinivas shows the same result for Markov chains in [Sri96].

It is in this context that several techniques have been applied to identify methods that attempt to solve the problem of unbounded relative error. For the most part the literature deals with methods to identify efficient IS distributions. Another approach is the so called splitting method.

5.2.1 Importance Sampling Methods for Queuing Systems

An early review of IS in static Monte Carlo methods is given in [Ham67]. More recent reviews are [Gly89] dealing with general stochastic processes and [Hei95] reviewing IS results in the context of queuing and reliability models. Another very good overview can be found in [Sri96, chap. 1]. Some of the discussion here is taken from [Hei95] and [Sri96]. Also, we limit ourselves to consider the IS problem from the point of view of simulating queuing systems.

Consider the equation (5.3). Applying IS to this we get the following estimator

$$\hat{\gamma}_N(q) = \frac{1}{N} \sum_{n=1}^{N} 1_{X_n \in A} \frac{p(X_n)}{q(X_n)} = \frac{1}{N} \sum_{n=1}^{N} 1_{X_n \in A} w(X_n), \tag{5.4}$$

where the samples $X_1, \ldots, X_N$ are i.i.d. and generated from the density $q(x)$. Then the optimal zero variance IS density $q(x)$ is

$$q(x) = \begin{cases} p(x)/\Pr[A], & \text{if } X \in A, \\ 0, & \text{otherwise,} \end{cases}$$

i.e. the original density $p(x)$ conditioned on A. If the sequence $\{X_1, \ldots, X_N\}$ is a Markov chain under measure P with transition probabilities $p(\cdot,\cdot)$ and initial distribution ν the corresponding estimator becomes

$$\hat{\gamma}_N(Q) = \frac{1}{N}\sum_{n=1}^{N} 1_{X_n \in A} \frac{\nu(X_1)}{\eta(X_1)} \prod_{n=2}^{N} \frac{p(X_{n-1}, X_n)}{q(X_{n-1}, X_n)} = \frac{1}{N}\sum_{n=1}^{N} 1_{X_n \in A} L_N, \tag{5.5}$$

where η denotes the initial distribution and $q(\cdot,\cdot)$ denotes the transition probabilities under the IS measure Q, and L_N is the likelihood ratio. The zero variance IS measure would in this case be the joint probability of the sequence $\{X_1, \ldots, X_N\}$ conditioned on A, requiring, again, knowledge of the estimated parameter. For our purposes it is sufficient to consider the i.i.d. case and the Markov chain case when using IS (see [Gly89] for information on IS with more general stochastic processes). In general, it can be noted that for any sequence $\{X_1, \ldots, X_N\}$, the optimal IS distribution exhibits the tautology of requiring explicit knowledge of the estimated quantity, making it an unrealizable IS distribution.

How should the IS measure Q then be chosen (or the density q in the i.i.d. case) ? The estimator (5.4) is clearly unbiased and so is (5.5), when ν is the stationary distribution of the chain under the measure P. Then the variance of the estimator is

$$\operatorname{Var}[\hat{\gamma}_N(\cdot)] = \operatorname{E}\left[(\hat{\gamma}_N(\cdot))^2\right] - \Pr[A]^2.$$

Then, by the positivity of this variance for any measure, to minimize the mean square error of the estimator it is sufficient to minimize the first term $\operatorname{E}[(\hat{\gamma}_N(\cdot))^2]$. For this a good IS measure is one which makes the likelihood ratio small in the set A, i.e. under the IS measure the events $X_n \in A$ have a high probability. Srinivas [Sri96] has shown a related monotonicity result for discrete state space Markov chains regarding the properties of a good $\eta, q(\cdot,\cdot)$ pair forming the IS measure of the Markov chain: Given that the chain visits the set A during a regenerative cycle with T_A denoting the time it happens, if $x_0, \ldots, x_{T_A}$ and $x'_0, \ldots, x'_{T_A}$ are such that

$$\nu(x_0) \prod_{n=0}^{T_A - 1} p(x_n, x_{n+1}) > \nu(x'_0) \prod_{n=0}^{T_A - 1} p(x'_n, x'_{n+1})$$

then

$$\eta(x_0) \prod_{n=0}^{T_A - 1} q(x_n, x_{n+1}) > \eta(x'_0) \prod_{n=0}^{T_A - 1} q(x'_n, x'_{n+1}),$$

which states that the IS measure Q should preserve the likelihood ordering of those paths leading to the event A under the original measure P.

At this point we can also make a remark about the increase in computational cost associated with using IS. Generally, the IS distributions that are used increase the computational costs somewhat, but the considerable reduction in the necessary sample sizes amply make up for the extra computational effort. Glynn and Whitt give a theoretical treatment of this issue in [Gly92].

Next we discuss the application of IS in the context of steady state simulation and, specifically, using Markov chains for that. Then we discuss the different approaches to choosing a good IS measure in the literature.

Dynamic Importance Sampling

When using IS in Markov chain simulation and estimating steady state performance measures, one issue is that the likelihood ratio $L_N \to 0$ in (5.5) when $N \to \infty$. Thus, $\hat{\gamma}_N(Q) \to 0$ as $N \to \infty$, i.e. the estimator (5.5) is not consistent, although it is unbiased (note that the Monte Carlo estimator with i.i.d. samples (5.4) does not suffer from this). This is shown formally in [Gly89] for a broad class of processes called regenerative generalized semi-Markov processes of which a DTMC is a special case. A more exact analysis of this is given by Glynn in [Gly94] for the case of Markov chains. Intuitively it is quite obvious, since the IS measure Q should be chosen to make the “less likely” events under P more likely under Q. Then the value of each term in the product of the likelihood ratio L_N corresponding to transitions from the Markov chain under Q is “usually” less than 1. Luckily it is possible to avoid this problem by using regenerative simulation, where the theoretically infinite length steady state simulation problem is transformed into independent finite length cycles.

Then the rare event problem (5.3) for DTMCs is transformed into the estimation of the ratio

$$\gamma = \frac{\mathrm{E}[\sum_{n=0}^{\tau-1} 1_{X_n \in A}]}{\mathrm{E}[\tau]}, \tag{5.6}$$

where τ is the first time that the DTMC enters the regeneration state. This issue has been addressed in the context of reliability models by Goyal et. al. [Goy92]. There it has been noted that under the original measure the estimation of the denominator does not involve a rare event and hence it can be estimated using normal simulation. To derive the IS measure for the simulation of the numerator, the authors consider a simple three state reliability model, which is simple enough to permit analytical solutions for the optimal IS measure. In essence, their conclusion is then that the importance sampling measure for the numerator should move the process quickly to the rare set A and once it has been reached the importance sampling should be turned off, i.e. the simulation should be carried out under the original measure to drive the process back to the regenerative state. Thus, the method has been named as *dynamic importance sampling*, since the process is not anymore time homogenous under the IS measure. Devetsikiotis and Townsend have made similar conclusions in [Dev93a]. Goyal et. al. also note that it is only the estimation of the numerator in (5.6) which involves a rare event and it is for this that an efficient IS distribution should be found. The estimation of the denominator can be done efficiently without the use of IS. In [Goy92] the authors propose to use known methods such as failure biasing (see e.g. [Sha94] or [Hei95] and the references there in) to achieve this, but we will not describe these methods here since, as mentioned earlier, we are here more interested in queuing systems rather than reliability models, which have slightly different properties.

The fact that the estimation of the numerator in (5.6) can be split into two parts, where only one involves a rare event can be illustrated in the following way. Let us denote by B the event that X_n reaches the set A for the first time before returning to the regenerative state and let τ_B denote the time that this happens. Then by conditioning on event B the numerator of (5.6) can be expressed as

$$\begin{aligned}
\mathrm{E}\left[\sum_{n=0}^{\tau-1} 1_{X_n\in A}\right] &= \mathrm{E}\left[\sum_{n=\tau_B}^{\tau-1} 1_{X_n\in A} \mid B\right] \Pr\left[B\right] \\
&= \Pr\left[B\right] + \mathrm{E}\left[\sum_{n=\tau_B+1}^{\tau-1} 1_{X_n\in A} \mid B\right] \Pr\left[B\right].
\end{aligned}$$

From this it can be seen that the estimation corresponds to the estimation of two quantities: first estimating the probability of reaching the set A and then the additional expectation for the average number of times the process returns to the set A once the process is started from the set A before returning to the regeneration state of which the latter is not a rare event problem.

Iterative Methods

Now we return to the question of how to choose the IS measure for the simulation. In practice it is not possible to try to minimize $\mathrm{E}[(\hat{\gamma}_N(Q))^2]$ over the class of all possible measures Q. A feasible approach is then to limit the search to a parametric family parameterized by some $\theta \in \Theta$ and try to find the θ^* that minimizes $\mathrm{E}[(\hat{\gamma}_N(\theta))^2]$ for estimator (5.4) or (5.5). In most cases closed form formulas for $\mathrm{E}[(\hat{\gamma}_N(\theta))^2]$ are not available, so one can then try to estimate $\frac{\partial}{\partial\theta}\mathrm{E}[(\hat{\gamma}_N(\theta))^2]$ by simulation and use the stochastic optimization techniques described in e.g. [Gly90] to perform the search for the minimum.

This approach is very practical, since it requires very little analytical knowledge of the system to be simulated as opposed to the asymptotically optimal IS methods we will review later. It has been used successfully by Devetsikiokis and Townsend [Dev93a] for estimating cell losses in the context of DTMC chain simulation of queuing systems and in [Dev93b] for estimating bit error rates in communication systems. The idea is that during the search for optimal θ^* only relatively short simulations need to be done to estimate $\mathrm{E}[(\hat{\gamma}_N(\theta))^2]$. Once convergence has been achieved the actual simulation is performed with the parameter θ^*. In [Dev93c] the authors also use this method to find the value of θ under which the probability of the rare event is maximized.

A related approach has been proposed by Stadler and Roy [Sta93] (see references therein for more articles on this approach). They consider the i.i.d. case in the context of bit error estimation in communication systems. The idea is based on the following: During a simulation those samples that hit the rare set A are distributed according to the optimal IS measure and can be used to estimate its properties. Then we can run several (short) simulation runs and during each run the probability of interest $\hat{\gamma}_N(\cdot)$ and the properties of the optimal IS measure are estimated. The actual IS measure (in the i.i.d. case a density) used for generating the samples is then modified in such a way that its properties match the estimated properties of the optimal IS measure, and the modified IS measure is used in the next simulation run. In this iterative manner the IS measure becomes more like the optimal IS measure in terms of the properties that one is estimating from it as the number of simulation runs increases, and, hence, also the accuracy of $\hat{\gamma}_N(\cdot)$ increases. In practice, the chosen property that is to be estimated from the optimal IS measure can be, for example, the mean of the optimal IS measure, i.e. the conditional expectation $\mathrm{E}[X \mid X \in A]$, and/or the conditional variance of X in the set A.

Asymptotically Optimal Importance Sampling

Since the unique optimal IS measure is impractical, many authors have tried other weaker criteria. Earlier we discussed that the relative error of the estimates is unbounded when the estimated probability goes to zero. More formally, let us consider a sequence of rare event problems indexed by a parameter ϵ such that the probability of interest $\gamma(\epsilon) \to 0$ as $\epsilon \to 0$. Then one criterion for the efficiency of the IS measure Q is that the relative error is bounded under the measure Q:

$$\lim_{\epsilon \to 0} RE(\hat{\gamma}_N(Q)) = \lim_{\epsilon \to 0} \frac{\sigma(\hat{\gamma}_N(Q))}{\gamma(\epsilon)} \to d < \infty,$$

where d is some constant. This implies that the number of samples required to reach a fixed confidence level (say 95%) is also bounded as $\epsilon \to 0$. This criterion has been used in the context of simulating reliability models, see e.g. [Sha94] for proofs of certain IS measures having this property.

In queuing systems the standard terminology is to call an asymptotically efficient IS measure *asymptotically optimal*. The conditions of asymptotic optimality are very much rooted in the study of exponentially rare events for which large deviations theory can be applied effectively. For such models it can sometimes be shown that

$$\lim_{\epsilon \to 0} \epsilon \, \log(\gamma(\epsilon)) \to c$$

for some constant c. If

$$\lim_{\epsilon \to 0} \epsilon \, \log(\mathrm{E}\left[\hat{\gamma}_N(Q)^2\right]) \to 2c,$$

then the IS measure Q is said to be asymptotically optimal, i.e. for an exponentially rare event the rate of decrease of the second moment of the estimator is twice that of the first moment. Thus also the number of samples to reach a fixed relative error increases more slowly than any exponential, although it may not be bounded. Note that for exponentially rare events this is a weaker form of optimality than having bounded relative error.

Early references in asymptotic optimality include Siegmund [Sie76] and Cottrell et. al. [Cot83]. In both articles the authors consider the parametric family of so called *exponentially shifted* IS measures and they are able to derive the optimal form of the IS measure exhibiting the asymptotic optimality criteria as discussed above. In particular, Siegmund was the first to relate large deviation results to asymptotic optimality in simulations. He considers the r.v. $\hat{\gamma}_n = 1/n \sum_{k=1}^{n} f(X_k)$, where X_k are i.i.d. with density $p(x)$. Then, provided certain conditions are satisfied, the probability $P_n = \Pr[\hat{\gamma}_n \in A]$ vanishes at an exponential rate. An exponentially shifted distribution $q(x)$ is specified by

$$q(x) = e^{\theta f(x) - \log \lambda(\theta)} p(x),$$

where $\lambda(\theta)$ is the so called moment generating function

$$\lambda(\theta) = \mathrm{E}\left[e^{\theta f(X_k)}\right].$$

Siegmund shows that the particular shift determined by the parameter θ obtained from the minimization of the so called large deviation rate function, also asymptotically minimizes the estimator variance in the class of exponential shifts.

Cottrell et. al. [Cot83] consider the problem of determining most likely paths for discrete time Markov processes. Their approach deals directly with the sample paths of the process and uses "Wentzell-Freidlin"-type large deviation results. They first determine the optimal path, i.e. in some sense the most likely path, leading to some interesting set of states when the process is started from some fixed state as a solution to a dynamic programming problem. Then they derive the form of the change of measure under which the probability of the optimal path is maximized and the form turns out to be an exponentially shifted distribution at each state of the path where the shift parameter is, again, determined by minimization of the large deviation rate function. Finally they prove that in the family of exponential shifts this choice of the shift parameter minimizes the asymptotic estimator variance.

Some of the most powerful asymptotic optimality results have been obtained by Bucklew and Sadowsky in a series of papers. Let us first consider the following asymptotic optimality results presented by Bucklew in [Buc90a, chap. 8]. There he takes X_i to be an irreducible Markov chain under P on a finite state space with initial distribution ν and transition probabilities $p(\cdot,\cdot)$, and considers the event

$$A_n = \left\{ x_1, \ldots, x_n : \frac{1}{n} \sum_{i=1}^{n} f(x_i) \geq 0 \right\},$$

where $\mathrm{E}[f(x)] < 0$ taken with respect to the stationary distribution of the chain. The probability $\Pr[A_n]$ is estimated by

$$\hat{\gamma}_p = \frac{1}{k} \sum_{j=1}^{k} 1_{X^j \in A_n},$$

where $X^j = (X_1^j, \ldots, X_n^j)$ is the j^{th} independent sample of the first n steps of the chain X_i. Then let us limit ourselves to the case where the IS measure Q is a Markov chain Y_i on the same state space with initial distribution η and transition probabilities $q(\cdot,\cdot)$. The corresponding IS estimator is given by

$$\hat{\gamma}_q = \frac{1}{k} \sum_{j=1}^{k} 1_{X^j \in A_n} \frac{\nu(Y_1^j)}{\eta(Y_1^j)} \prod_{i=1}^{n-1} \frac{p(Y_i^j, Y_{i+1}^j)}{q(Y_i^j, Y_{i+1}^j)}.$$

Then by the large deviation principle

$$\frac{1}{n} \log \Pr\left[A_n\right]^2 \to 2 \log \lambda(\theta_0),$$

where $\lambda(\theta_0)$ denotes now the largest eigenvalue of the operator

$$T_\theta g(x) = \sum_y e^{\theta f(y)} g(y) p(x,y)$$

and $\lambda'(\theta_0) = 0$ defines θ_0. Note that the $T_\theta g(x)$ operator is a generalization of the moment generating function defined earlier for the i.i.d. case. Recall that the variance of the IS estimator is $\mathrm{Var}[\hat{\gamma}_q] = \mathrm{E}[\hat{\gamma}_q^2] - \Pr[A_n]^2$. Bucklew then shows that q is asymptotically optimal, i.e.

$$\lim_{n\to\infty} \frac{1}{n} \log\left(\mathrm{Var}\left[\hat{\gamma}_q\right]\right) = 2\log\lambda(\theta_0),$$

if

$$\lim_{n\to\infty} \frac{1}{n} \log\left(\mathrm{E}\left[\hat{\gamma}_q^2\right]\right) = 2\log\lambda(\theta_0);$$

otherwise

$$\lim_{n\to\infty} \frac{1}{n} \log\left(\mathrm{E}\left[\hat{\gamma}_q^2\right]\right) = \lim_{n\to\infty} \frac{1}{n} \log\left(\mathrm{Var}\left[\hat{\gamma}_q\right]\right) > 2\log\lambda(\theta_0).$$

He then proceeds to show that the optimal choice for q is, in fact, unique and is given by

$$q(x,y) = e^{\theta_0 f(y)} \frac{r(y)}{r(x)\lambda(\theta_0)} p(x,y), \tag{5.7}$$

where $r(\cdot)$ is the right eigen-vector associated with $\lambda(\theta_0)$. Note that this is a non-parametric result which states that among the class of all finite state space irreducible Markov chains the optimized exponentially shifted chain is the only asymptotically optimal IS measure. In Bucklew et. al. [Buc90b] this result is extended to a more general state space and more general sets A_n. They also show that for a fixed relative error the number of samples required grows as $O(\sqrt{n})$ when $n \to \infty$ when the IS measure is (5.7); otherwise the number of samples required grows exponentially.

In [Sad90] Sadowsky and Bucklew consider the following multidimensional case. Let $\{Y_n, n = 1, 2, \ldots\}$ be a sequence of d-dimensional random vectors. The goal is to estimate the probability

$$\gamma_n = \Pr\left[Y_n \in E\right] = \int 1_{y\in E}\, F_n(dy) \tag{5.8}$$

where E is a multidimensional set and $F_n(\cdot)$ is the true distribution of the sequence Y_n. Then the IS estimator for (5.8) is

$$\hat{\gamma}_n^{(L)} = \frac{1}{L} \sum_{l=1}^{L} \frac{dF_n}{dF_n^*}(Y_n^{(l)})\, 1_{Y_n^{(l)} \in E}, \tag{5.9}$$

where $Y_n^{(l)}, l = 1, \ldots, L$, are L independent realizations of the sequence Y_n generated from the IS distribution $F_n^*(\cdot)$.

For the sequence Y_n we assume that the following holds. We define for each $\theta \in \mathcal{R}^d$ and each $n < \infty$

$$\mu_n(\theta) = \frac{1}{n} \log\left(\mathrm{E}\left[e^{n\theta\cdot Y_n}\right]\right),$$

where the center dot denotes the normal dot product. Then we assume that the asymptotic log-moment generating function,

$$\mu(\theta) = \lim_{n\to\infty} \mu_n(\theta),$$

always exists for all $\theta \in \mathcal{R}^d$. This is the case, for example when $Y_n = 1/n \sum_{k=1}^{n} Z_k$, where Z_k are bounded and i.i.d. or Gaussian with covariance matrix proportional to $1/n$. Also, we define the large deviation rate function

$$I(y) = \theta_0 y - \mu(\theta_0),$$

where θ_0 is obtained as a solution from $\nabla\mu(\theta_0) = y$. From this, one obtains the so called Cramér transform of the set E as

$$I(E) = \inf_{y\in E} I(y). \tag{5.10}$$

An important concept in large deviation theory is the so called *dominating point*. In earlier results we have reviewed, it has not been explicitly stated, since we have been dealing with one dimensional random variables and in such cases the sets of practical interest in most cases have a dominating point. However, in a multidimensional case this is not so obvious anymore. This can be illustrated in the following way. Let us assume there exists a unique solution ν as the minimizing value of y in (5.10) and let θ_0 be the solution to $\nabla\mu(\theta_0) = \nu$. Then let us define the half space $H(\nu) = \{y : \theta_0 \cdot (y - \nu) \geq 0\}$, i.e. the part of space lying "above" the hyperplane obtained by setting a tangent to the rate function level set $\{y : I(y) = I(\nu)\}$ at point ν. If the set E is wholly contained within $H(\nu)$, i.e. $E \subset H(\nu)$, then the point ν is the dominating point of the set E. This is illustrated in Fig. 5.1. A sufficient condition, but not a necessary one, for the existence of a dominating point is e.g. that E is convex and that the rate function $I(\cdot) < \infty$ at least somewhere in E.

However, a set does not always have a dominating point as illustrated in Fig. 5.2. In the figure we can see that to cover the set E with half spaces, two points $V = \{\nu_1, \nu_2\}$ are required. Still in the figure the set E has a unique *minimum rate point*, since $I(\nu_1) < I(\nu_2)$. In the article all such "important" points are still called minimum rate points.

Sadowsky and Bucklew then proceed to show that the second moment of the estimator (5.9), i.e. $\mathrm{E}[(\hat{\gamma}_n^{(L)})^2]$, is uniquely minimized by the dominating point exponentially shifted distribution,

$$F_n^*(dy) = e^{n[\theta_0 \cdot y - \mu_n(\theta_0)]} F_n(dy), \tag{5.11}$$

if ν is the dominating point of the set E (θ_0 is again the solution to the equation $\nabla\mu(\theta_0) = \nu$). Hence, (5.11) is also asymptotically optimal when ν is a dominating point.

The main result of the paper, however, concerns the case when there is no dominating point, i.e. the case illustrated in Fig. 5.2. To present the result we first define the essential domain of the rate function as $I = \{t \in \mathcal{R}^d : I(y) < \infty\}$. Also, we denote by $V = \{\nu_1, \ldots, \nu_m\} \subset I$ the set of minimum rate points of the set E as described earlier. To define the set V precisely we require a) that $\nu_i \in \partial E$, i.e. each point ν_i lies on the boundary of the set E, b)

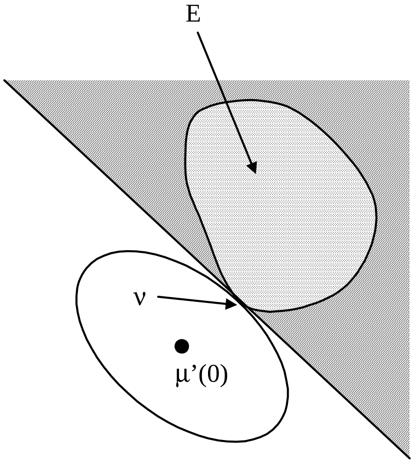

Figure 5.1: The set E with a dominating point ν.

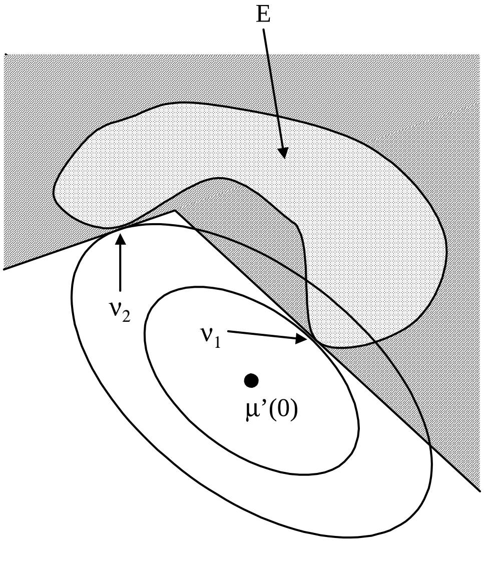

Figure 5.2: The set E with two minimum rate points ν_1 and ν_2.

that $\bar{E} \subset \bigcup_{i=1}^{m} H(\nu_i)$, where $\bar{E}$ denotes the closure of the set E and $H(\nu_i)$ is a half space lying "above" the hyperplane obtained by setting a tangent to the rate function level set $\{y : I(y) = I(\nu_i)\}$ at point ν_i, and c) that $I(\nu_i) \geq I(E)$. Essentially this means that the set V must have enough points so that the set E can be covered with the half spaces $H(\nu_i)$. Then the authors show that the unique asymptotically optimal shifted distribution is given by

$$F_n^*(dy) = \left[\sum_{i=1}^{m} e^{n[\theta_i \cdot y - \mu_n(\theta_i)]} p_i \right] F_n(dy),$$

where $(p_1, \ldots, p_m)$ is a probability vector and θ_i is again the solution to the equation $\nabla\mu(\theta_i) = \nu_i$, stating that the asymptotically optimal IS distribution has a composite form. Based on this result we will later present a similar composite IS distribution for simulating the blocking probabilities in loss systems.

The previous works show that the dominating point shifted distribution asymptotically minimizes the variance of the estimate. In [Sad93] Sadowsky continues the development of the above discussed paper [Sad90] and shows that the optimized shifted distribution asymptotically minimizes all error moments $\mathrm{E}[(\hat{\gamma}_n^{(L)} - \gamma)^k], k \geq 2$, thus implying that the number of samples required to estimate any error moment grows as $O(\sqrt{n})$ when $n \to \infty$. Any other IS distribution causes this sampling cost to grow exponentially as $n \to \infty$. This has a very important practical implication as regards to e.g. when to stop the simulation. Consider the stopping rule for simulation, where the simulation is stopped after the *estimated* variance has dropped below some threshold requiring stability of e.g. the fourth moment of $\hat{\gamma}_n^{(L)}$. If the IS distribution is not asymptotically optimal the estimated variances as a function of the number samples that have been drawn is unstable resulting typically in serious underestimation of the true variance. Furthermore, the behavior of the estimated variance is inconsistent as the number of samples increases in the sense that the estimated variance usually has sudden "jumps" where the estimate's value is suddenly affected by some very infrequent sample which has a huge weight. When the asymptotically optimal IS distribution is used, this problem is avoided.

In [Sad91] Sadowsky studies the $GI/GI/m$ queue. Specifically, he considers the probability of the event

$$A_n = \{\text{queue length exceeds } n \text{ in a busy period}\}.$$

The paper first provides a rigorous proof of the heuristic arguments given in [Par89] for the case of the $GI/GI/1$ queue. He shows that the unique asymptotically optimal IS measure exponentially shifts both the interarrival- and service time distributions by a parameter θ_0 satisfying

$$\lambda_A(\theta_0)\lambda_B(\theta_0) = 1,$$

where $\lambda_A(\cdot)$ and $\lambda_B(\cdot)$ denote the moment generating functions of the interarrival- and service times. Then the results are extended to the multiserver case. Chang et. al. have extended this idea to queues with complex arrival processes, see e.g. [Cha95] and the references therein.

A word of caution should be made at this point regarding the usability of the above mentioned results, which is mentioned e.g. in [Buc90a, chap. 8]. The asymptotic optimality applies only for rare event problems for which a large deviation principle applies and one must remember that this is not the case for all rare event problems. In such cases the exponential shifting may not be optimal or even near optimal.

Importance Sampling Methods for Queuing Networks

Parekh and Walrand were the first to suggest exponential shifting for queuing networks in a highly influential paper [Par89]. They first consider the $M/M/1$ queue and the estimation of the probability of the event, P_n, that the queue reaches level n before becoming empty again. They show by using similar methods as Cottrell et. al. [Cot83] that for this case the optimal exponentially shifted IS distribution is obtained by interchanging the arrival intensity λ and service rate μ of the queue. However, when this approach is applied to a queuing network it fails due to discontinuities of the transition rates at the boundaries of the state space.

Then the authors study a simple tandem queue network where arrivals are Poisson (with rate λ), the service rates are exponential in both queues (with rates μ_1, μ_2) and customers enter the system at queue 1 and leave the system only through queue 2. The probability of interest, P_n, is that of the number of customers in the network reaching n before emptying again given that the system starts empty. They give a large deviation approximation for P_n leading to a constrained optimization problem. Then it is shown numerically that if queue 1 is the bottleneck, then setting $\lambda = \mu_1$ and $\mu_1 = \lambda$ is the optimal solution and if queue 2 is the bottleneck, then $\lambda = \mu_2$ and $\mu_2 = \lambda$. This generalizes the interchange of the parameters of the $M/M/1$ queue to a tandem network. Similar results are obtained for the cases with two queues in parallel and the tandem case where customers leave the system with some probability after finishing service in queue 1. In [Gla95] Glasserman and Kou consider the tandem case in more detail and they show that the IS measure is not asymptotically optimal when the value of μ_1 is close to the value of μ_2. However, when their values are very different from each other, the heuristics give an asymptotically optimal IS measure in the sense as discussed earlier. Frater et. al. continued the ideas of [Par89] and extended them to general Jackson networks with routing in [Fra91]. They show that the resulting constrained optimization problem has a unique solution and that it can be solved analytically. In [Fra94] Frater and Anderson show how to solve the corresponding heuristic optimization problem for a tandem network of $GI/GI/1$ queues.

Since provably asymptotically optimal IS measures for general queuing networks are very difficult to obtain due to the complexity of the problem, Hsieh proposes in a recently published Ph.D. thesis [Hsi97] an alternative. The thesis considers generalized Jackson networks and estimating the probability that the total network backlog exceeds n. The method, called SEEKPATH, is adaptive in nature and begins by estimating a good shifting parameter for the exponentially shifted IS measure for a small value of n corresponding to a "scaling down" of the original rare event. After the shift parameter has been estimated the actual system is simulated. In the thesis it is showed for the case of the $GI/M/1$ queue that the SEEKPATH-method behaves approximately as the optimal exponentially shifted IS distribution. The actual performance of the method has been tested through extensive numerical studies, where the method is compared to other methods from the literature, e.g. the heuristics suggested by Parekh and Walrand [Par89].

In [Ros93] Ross and Wang consider a closed queuing network in an asymptotic region where the population size $N \to \infty$ while the load of the network decreases correspondingly such that each queue in the network still remains stable, i.e. the network satisfies so called normal usage conditions. The authors consider applying Monte Carlo summation techniques for estimating the normalization constant and utilization of the network. First the authors derive an IS distribution for sampling the normalization constant of the network that has bounded relative error, as defined earlier in this chapter, having the form of a multivariate independent exponential distribution, i.e. the components of the distribution are independent. Next they consider estimating the utilization of the network under normal usage conditions and conclude that the IS distribution with bounded relative error is a correlated multivariate normal distribution, from which it is difficult to sample from. However, the authors are able to develop an approximation for the asymptotically optimal solution, which is easy to sample from. Finally the authors consider the network under critical usage, i.e. where the load of the queues is close to 1. In this case also the asymptotically optimal IS distribution turns out to be difficult to sample from.

5.2.2 Importance Sampling Methods for Loss Systems

The literature on importance sampling methods specifically on loss networks is not very extensive. Furthermore, the methods that have been applied are all applications of the ideas already presented earlier in the previous section.

In [Ros95, chap. 6] Ross derives a heuristic IS distribution, which is also an exponentially shifted version of the original sampling distribution. However, the rationale behind the determination of the shift parameter is quite different from earlier methods. When using the static MC, the simulation consists of estimating two independent probabilities (see eq. (2.3): the probability of hitting the blocking states and the probability of being in the allowed state space. Ross proposes heuristics based on an observation that when using the static MC method and the same IS distribution for both probabilities, then with the optimal, i.e. minimum variance, IS distribution half of the samples will fall within the allowed state space and half of the samples will fall within the blocking states. Based on this observation Ross has presented heuristics for selecting the shift parameter attempting to increase the likelihood of the blocking states, while, at the same, trying to limit the likelihood of generating misses from the allowed state space. The problem, however, with these heuristics is that the resulting shift is too conservative as the blocking probabilities become smaller.

In [Man97] Mandjes considers the same problem and using the static Monte Carlo method for simulating the blocking probabilities. He notes first that it is not advantageous in the rare event context to estimate the two probabilities at the same time using the same sampling distribution. Instead, it is only the probability of hitting the blocking states that requires importance sampling. Mandjes' method is motivated by large deviation results for rare events of sums of independent Poisson variables and he proposes the use of an importance sampling distribution which shifts the mean of the sampling distribution to match the most probable blocking state. Essentially the method assumes that to estimate the blocking probability it is enough to identify the most probable blocking state, which also identifies the link where the blocking will occur, and to shift the sampling distribution such that its probability mass will be concentrated around the most probable blocking state. In the language of [Sad90], the method of Mandjes only identifies the unique minimum rate point of the set of the blocking states and, as discussed earlier, Sadowsky has shown that especially in multidimensional settings this is not necessarily enough to satisfy the conditions of asymptotical optimality for the IS distribution. We will return to this later when we present our own composite IS distribution for the same problem.

In [Hee97] Heegaard presents a regenerative Markov chain method, which uses the ideas of dynamic importance sampling and adaptive importance sampling. The basic idea is to use importance sampling to drive the process to the set of the blocking states and then turn off the importance sampling to speed up the regeneration. The method is adaptive in the sense that during the IS phase at each state along the generated path the jump distributions are changed according to the current state of the chain. In practice this is accomplished by first randomly sampling one of the link constraints to be the "target" of the path and then using the heuristics of [Par89] to select the shifting parameters for the jump distributions. The paper also gives heuristics for constructing the "target" distribution at each state.

5.2.3 The RESTART Method

As we have noted above, the effectiveness of importance sampling depends critically on the ability to find the right IS distribution. In [Vil91] M. Villén–Altamirano and J. Villén–Altamirano have presented an alternative method for rare event simulation that appears to require very little analytical analysis for its applicability. Their method, called RESTART, is based on a known method called splitting used e.g. in particle transmission simulation, see e.g. [Ham67, chap. 8.2]. The method can also be taken as a generalization of the conditional expectation method where also the conditional expectation is estimated through simulation.

Let us again consider the simulation of some process X_n describing e.g. the queue length process of a stable queue. Consider, again, the probability of a rare event A, $\Pr[A]$, that starting from the origin the process reaches the level N before returning to the origin. Let C be another less rare event such that $C \supset A$, for example that the queue length exceeds some intermediate threshold. Then we have the conditional probability that $\Pr[A] = \Pr[A \mid C]\Pr[C]$. Using simulation it is easy to estimate the probability $\Pr[C]$ since it is estimated from the whole simulation data and it is less rare. However, the conditional probability $\Pr[A \mid C]$ is estimated only from the portion of the simulation where the event C occurred. Since the occurrence of event C is still relatively infrequent (although much more frequent than the occurrence of event A), the conditional probability $\Pr[A \mid C]$ does not get estimated very well. In RESTART the idea is to increase the accuracy of the estimate for the conditional probability $\Pr[A \mid C]$ by making several independent replications of those parts of the process where the event C occurred, i.e. the process evolution is split into several paths when the threshold has been reached. In [Vil94] this method is extended to the case with several intermediate thresholds. Also, some guidelines for choosing the thresholds are given.

In [Gla96] Glasserman et. al. address the problem of choosing the number of subpaths to generate when a path splits. They assume certain characteristics from the dynamics of the process between the thresholds and they model the movement of the process from one threshold to another as a branching process with different levels of generality. They show that appropriately choosing the degree of splitting at each threshold is critical to the effectiveness of the method. In fact, they show that with the right amount of splitting the method is asymptotically optimal in the sense as described earlier. The choice balances two competing concerns: excessive splitting creates an explosive computational burden, and insufficient splitting eliminates the advantage over straightforward simulation. Loosely speaking, the result can be interpreted such that when a path splits, the number of subpaths should be chosen such that on average one subpath makes it to the next threshold. This keeps the expected number of paths alive at each threshold roughly constant. In [Gla97] Glasserman et. al. address the significance of choosing the threshold levels appropriately. Essentially they show that to achieve asymptotical optimality the thresholds should be chosen in some sense consistent with the most likely path to the rare set implying again a necessity to understand the large deviation asymptotics of the process to be simulated.

In a recent paper [Har97] Haraszti and Townsend present a new related method called *direct probability redistribution* (DPR). Their method is related to RESTART in the sense that in DPR the state space is also divided into progressively rarer subsets but, unlike in RESTART, the sets do not have to be nested. In DPR also the basic idea is to increase the accuracy of the conditional probabilities by making several retrials for those paths that enter a specific subset. In the paper the theory is developed for the case of simulating DTMCs.

5.3 Composite Importance Sampling Distribution for Loss Systems

In this section we present the derivation of an efficient IS distribution for estimating the blocking probabilities in a multiservice loss system, which will be published in [Las99]. Recall that based on the identity (2.3) we can have the following MC estimator

$$\hat{B}_k = \frac{1/N \ \sum_{n=1}^{N} 1_{\tilde{\mathbf{X}}_n \in \mathcal{B}^k}}{1/N \ \sum_{n=1}^{N} 1_{\tilde{\mathbf{X}}_n \in \mathcal{S}}}, \tag{5.12}$$

where $\tilde{\mathbf{X}}_n$ are independent samples of $\tilde{\mathbf{X}}$ defined in the larger state space $\tilde{\mathcal{S}}$.

Let $P^* : p^*(\mathbf{x}) = \Pr[\tilde{\mathbf{X}} = \mathbf{x}] > 0$ for $\tilde{\mathbf{X}} \in \tilde{\mathcal{S}}$ denote the importance sampling distribution. With respect to this distribution the expectation in the numerator of (2.3) becomes

$$\mathrm{E}\left[1_{\tilde{\mathbf{X}}\in\mathcal{B}^k}\right] = \mathrm{E}_{p^*}\left[1_{\tilde{\mathbf{X}}\in\mathcal{B}^k} w(\tilde{\mathbf{X}})\right],$$

where $w(\mathbf{x}) = p(\mathbf{x})/p^*(\mathbf{x})$ is the likelihood ratio. The same holds for the denominator as well, and we get the estimator

$$\hat{B}_k = \frac{\sum_{n=1}^N w(\tilde{\mathbf{X}}_n) 1_{\tilde{\mathbf{X}}_n\in\mathcal{B}^k}}{\sum_{n=1}^N w(\tilde{\mathbf{X}}_n) 1_{\tilde{\mathbf{X}}_n\in\mathcal{S}}}. \tag{5.13}$$

As has been noted earlier by Mandjes in [Man97] that in the case of the multiservice loss system it is only the estimation of the numerator that can be made more efficient by using IS. Hence, we will next focus on efficiently estimating the numerator of (5.13), i.e. $1/N \sum_{n=1}^N w(\tilde{\mathbf{X}}_n) 1_{\tilde{\mathbf{X}}_n\in\mathcal{B}^k}$.

5.3.1 Efficient IS distribution

Let us consider a general problem of estimating the probability of the event $\mathcal{B}$

$$\beta = \Pr[\mathbf{X} \in \mathcal{B}] = \mathrm{E}[1_{\mathbf{X}\in\mathcal{B}}],$$

where $\mathbf{X} \in \mathcal{S}$ has some distribution $p(\mathbf{x})$ and $\mathcal{B} \subseteq \mathcal{S}$. Now we wish to get insight into how the IS distribution should be chosen. By denoting with $p^*(\mathbf{x})$ the IS distribution, we can again express the expectation as

$$\beta = \mathrm{E}_{p^*}[w(\mathbf{X})], \tag{5.14}$$

where now $w(\mathbf{X}) = [p(\mathbf{X})/p^*(\mathbf{X})]\, 1_{X\in\mathcal{B}}$. Note that here we also include the indicator $1_{X\in\mathcal{B}}$ in the definition of $w(\cdot)$ as opposed to how it has been defined earlier. Also, let $\beta^* = \mathrm{E}_{p^*}[1_{\mathbf{X}\in\mathcal{B}}]$ be the probability of the event $\mathcal{B}$ with respect to the distribution $p^*(\mathbf{x})$. By conditioning on the value of the random variable $I = 1_{\mathbf{X}\in\mathcal{B}}$ we can express (5.14) as

$$\beta = \mathrm{E}_{p^*}[w(\mathbf{X})] = \mathrm{E}_{p^*}[\mathrm{E}_{p^*}[w(\mathbf{X})|I]] = \beta^* \mathrm{E}_{p^*}[w(\mathbf{X})|I=1].$$

Thus we have $\mathrm{E}_{p^*}[w(\mathbf{X})|I=1] = \beta/\beta^*$. The variance of $w(\mathbf{X})$ under P^* is then

$$\begin{aligned}\mathrm{Var}_{p^*}[w(\mathbf{x})] &= \mathrm{Var}_{p^*}[\mathrm{E}_{p^*}[w(\mathbf{X})|I]] + \mathrm{E}_{p^*}[\mathrm{Var}_{p^*}[w(\mathbf{X})|I]] \\ &= \mathrm{E}_{p^*}\left[\mathrm{E}_{p^*}[w(\mathbf{X})|I]^2\right] - \mathrm{E}_{p^*}[\mathrm{E}_{p^*}[w(\mathbf{X})|I]]^2 + \beta^*\sigma^{*2} \\ &= \frac{\beta^2}{\beta^*} - \beta^2 + \beta^*\sigma^{*2}, \end{aligned} \tag{5.15}$$

where $\sigma^{*2} = \mathrm{Var}_{p^*}[w(\mathbf{X})|I=1]$ is the variance of $w(\mathbf{X})$ in $\mathcal{B}$ under P^*. From this formulation we are able to get the desired insight into the effect of the IS distribution.

When no shifting is used and $p^*(\mathbf{x}) = p(\mathbf{x})$ then also $\beta^* = \beta$ and $\sigma^{*2} = 0$. In this case

$$\mathrm{Var}_p\left[w(\mathbf{X})\right] = \beta(1-\beta) \approx \beta.$$

By increasing the probability β^* of $\mathcal{B}$ under $p^*(\mathbf{x})$, the first and most important term in (5.15) is reduced. Ideally, if one can make $\beta^* = 1$, the first and second term completely cancel. If the probability of $\mathcal{B}$ can be increased uniformly, i.e. with $w(\mathbf{x})$ constant in $\mathcal{B}$, then $\sigma^{*2} = 0$ and the estimator would have a zero variance. The ideal IS distribution implies knowledge of the quantity to be estimated, and cannot be easily constructed. An efficient IS distribution, however, tries to approximate it as closely as possible. In practice one is limited to a family of shifted distributions, and one has to compromise between the two factors. It is important to increase the probability of $\mathcal{B}$ but at the same time it is important to keep $w(\mathbf{x})$ as constant as possible in $\mathcal{B}$ in order to minimize σ^{*2}.

Guided by this insight, we now develop an efficient IS distribution for simulating the blocking probabilities in the multiservice loss system. The basic idea is to derive a distribution for traffic class k which will make all the blocking states associated with the active link capacity constraints more probable. For this we first need to identify the most probable blocking states on the links which traffic class k uses in the network. This is illustrated in Fig. 5.3, which shows an example with two traffic classes. Now we could choose one of these points and shift the original distribution such that the main mass of the distribution is centered around that point. In this way we can increase β^*. However, if we use only one point then the distribution of $w(\mathbf{x})$ in $\mathcal{B}$ will be uneven giving rise to a large σ^{*2}. Therefore, it is more advantageous to use a composite distribution which is a weighted combination of the individual shifted distributions.

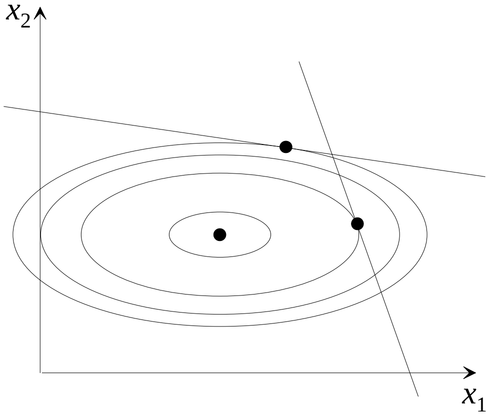

Figure 5.3: Most probable blocking states in a two traffic class example with two link constraints.

This approach is supported by the results of [Sad90], as was discussed earlier, where it was shown that so called asymptotically optimal shifted distribution for $\mathcal{B}$ indeed is of this composite form for problems satisfying a large deviation principle. For loss systems this is the case e.g. in the limit when the offered loads tend to zero. Then a failure to include the most probable blocking states on all the links involved, corresponding to the minimum rate points in [Sad90], means that the sampling distribution can be asymptotically very inefficient. This suggests that the composite distribution leads to an efficient sampling distribution even in the non-asymptotic regime. The asymptotic theory, however, leaves the weights in the composite distribution open. We will here fix them by the heuristics of maximal uniformity of $w(\mathbf{x})$ in $\mathcal{B}$.

The most probable blocking states

The most probable blocking state on link j is found by maximizing (2.1) on a given hyperplane representing the capacity constraint of the link,

$$\max_{\mathbf{x}} \prod_{k=1}^{K} \frac{\rho_k^{x_k}}{x_k!} = \max_{\mathbf{x}} \prod_{k=1}^{K} f(x_k, \rho_k)$$

$$\text{subject to} \quad \mathbf{x} \cdot \mathbf{b}_j = C_j,$$

where the components of $\mathbf{x}$ are real valued, i.e. not restricted to integers. This problem has been considered elsewhere in the literature (e.g. [Kel86] and [Man97]) and usually it has been solved by using the Stirling's approximation for the factorial term and then solving the constrained optimization problem. However, the solution can easily be found numerically even without using the Stirling's approximation by using standard techniques. To this end, consider the following equivalent optimization problem

$$\begin{aligned} &\max_{\mathbf{x}} \sum_{k=1}^{K} \log f(x_k, \rho_k) \\ &\text{subject to} \quad \mathbf{x} \cdot \mathbf{b}_j = C_j. \end{aligned} \tag{5.16}$$

Introducing the Lagrange multiplier θ, we are lead to the unconstrained maximization of the following objective function

$$\max_{\mathbf{x}} \sum_{k=1}^{K} [\log f(x_k, \rho_k) - \theta x_k b_{j,k}] - \theta C_j. \tag{5.17}$$

Maximization of (5.17) with respect to the x_k, leads to the solution

$$x_k = h^{-1}(\theta b_{j,k}, \rho_k), \quad k = 1, \ldots, K, \tag{5.18}$$

where $h^{-1}(\cdot, \rho)$ denotes the inverse function of $h(\cdot, \rho)$,

$$h(x_k, \rho_k) = \frac{\partial}{\partial x_k} \log f(x_k, \rho_k).$$

It is easy to check that $h(\cdot,\cdot)$ is a monotonously decreasing function of x_k and hence the inverse function is also well defined. The Lagrange multiplier θ is now determined by requiring that the solution must satisfy the constraint of (5.16), i.e.

$$\sum_{k=1}^{K} b_{j,k} h^{-1}(\theta b_{j,k}, \rho_k) = C_j. \tag{5.19}$$

Hence, the solution $\mathbf{x}_j^*$ to (5.16) is obtained by first solving (5.19) for θ and then evaluating (5.18) for $k = 1, \ldots, K$. This solution represents the "most likely" blocking state on the link j. Note, that the solution $\mathbf{x}_j^*$ may actually lie outside the allowed state space $\mathcal{S}$. Despite this, we can still use such a point as a shifting center.

Determining the weights

We restrict ourselves to consider IS distributions belonging to the family of exponentially shifted distributions. In the case of the Poisson distribution the shifted distribution is also a Poisson distribution but with a different load parameter $\boldsymbol{\gamma}$, instead of $\boldsymbol{\rho}$. Now the main mass of the sampling distribution can be moved to the point $\mathbf{x}_j^*$ by selecting $\boldsymbol{\gamma}_j = \mathbf{x}_j^*$. Let us denote by $p_j^*(\mathbf{x})$ the resulting shifted distribution, defined in the state space $\tilde{\mathcal{S}}$. In fact, if $\tilde{\mathcal{S}}$ is the whole space $\tilde{\mathbf{X}} \geq 0$, the shifting would correspond to moving the mean of the distribution to $\mathbf{x}_j^*$. This is again illustrated in Fig. 5.4, which shows the sampling distributions of a two traffic class example with two link constraints.

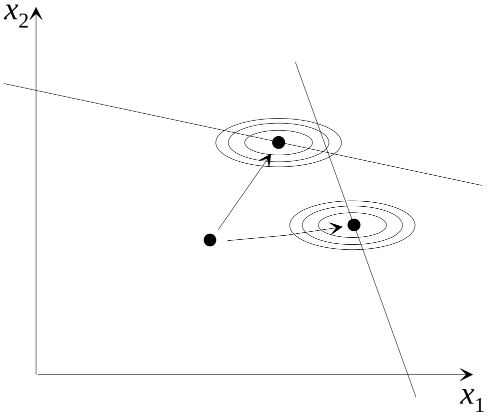

Figure 5.4: The shifted distributions in a two traffic example with two link constraints.

A traffic class uses only a subset of all the links in the network. Thus, when estimating the blocking probability of traffic class k, we only need to sample on those links, which the traffic class k actually uses and the complete sampling distribution for traffic class k is then a weighted combination of those distributions. For this let us denote by R_k the set of links the traffic class k uses. Formally we have that

$$R_k = \{j \in 1, \ldots, J \mid b_{j,k} > 0\}.$$

Also, let J_k denote the number of links in the set R_k. Then the composite sampling distribution for traffic class k is expressed as

$$p^*(\mathbf{x}) = \sum_{i \in R_k} P_i \, p_i^*(\mathbf{x}), \tag{5.20}$$

where the P_i form a discrete probability distribution. In particular P_i is the probability of using the distribution $p_i^*(\cdot)$ for generating the sample. It remains to find a method to obtain these weights. Again, the used heuristics are based on the goal of keeping the likelihood ratio as constant as possible in $\mathcal{B}^k$. With J_k parameters P_i available, we cannot make $w(\mathbf{x})$ constant everywhere in $\mathcal{B}^k$. Instead, we choose to require $w(\mathbf{x})$ to be constant, η, at the most probable blocking states $\mathbf{x}_j^*$. This requirement leads to a set of linear equations

$$\sum_{i \in R_k} P_i \, p_i^*(\mathbf{x}_j^*) = \eta \, p(\mathbf{x}_j^*), \quad \forall j \in R_k, \tag{5.21}$$

where the constant η is chosen such that $\sum_i P_i = 1$.

Unfortunately, there is no guarantee that the solution always satisfies $P_i \geq 0, \forall i \in R_k$. If negative values appear, (5.21) may be replaced by a suitable minimization problem with the constraint, $P_i \geq 0, \forall i \in R_k$. This case, however, is left for future research.

5.3.2 Numerical examples

Here we consider some numerical examples in order to illustrate the efficiency of the composite IS distribution in Monte Carlo simulation of the blocking probabilities. First we consider a simple 2 traffic class network with 3 links. The exact parameters of the network are: $C_j = [100, 120, 170], \mathbf{b}_1 = [2, 0], \mathbf{b}_2 = [0, 3]$ and $\mathbf{b}_3 = [2, 3]$ (i.e. the network topology is the same as e.g. in Fig. 2.1). We consider the blocking probability of traffic class 1 with two different loads such that the blocking probabilities are of the order 10^{-2} (Case 1 in the Table 5.1) and 10^{-4} (Case 2 in the Table 5.1). The exact used offered loads were $\rho = [35, 22]$ (Case 1) and $\rho = [27, 18]$ (Case 2). We compare the composite method against results obtained with the standard MC (MC in the table) and the methods proposed by Mandjes (Single shift in the table) in [Man97], and Ross in [Ros95, chap. 6], which both correspond to the use of a single shifted IS distribution. For this, we estimated the standard deviation under P^* of the observed variable $w(\tilde{\mathbf{X}}) \, 1_{\tilde{\mathbf{X}} \in \mathcal{B}^k}$ in the estimator (5.13). We used 100 000 samples for Case 1 and 10 000 000 samples for Case 2. To verify the accuracy of the result we also calculated the exact result for the standard deviation by brute force summation. This is given in parenthesis next to the estimated result. The results show that with the composite distribution we are able to reduce the variance of the samples considerably. For example, in Case 2 the deviation of the sample with the composite method is almost 14 times smaller than with the standard Monte Carlo method.

Next we experiment with the numerical example studied by Heegaard [Hee97], where he uses an adaptive importance sampling scheme in a Markov chain simulation setting for

Case	Composite	Single shift	Ross	MC
1	0.0434 (0.0442)	0.0582 (0.0569)	0.0686 (0.0685)	0.0988 (0.0999)
2	0.00057 (0.00080)	0.0015 (0.0030)	0.0030 (0.0030)	0.0110 (0.0110)

Table 5.1: The standard deviation of the observed variable.

extremely low blocking probabilities. The example corresponds to a network where for most of the classes the blocking is not dominated by any single link. The network has 11 links and 10 traffic classes. However, the link sizes are small enough to permit the calculation of the exact blocking probabilities B_k by brute force summation or by the use of a convolution algorithm, see e.g. [Ive87]. We selected as examples three traffic classes (2, 4 and 6) to illustrate the differences in accuracy when using just one shifted distribution and the composite distribution. In this case the single shifted IS distribution is obtained with the method of Mandjes. We do not consider the method by Ross, because it suffers from the shift being too conservative and e.g. in the examples here the method is not able to produce any estimate even after 1 000 000 samples. To compare the results we compute the relative error of the estimate, given by $(\hat{B}_k - B_k)/B_k$, and the 95% confidence interval as estimated from the simulation. For each example we give the results for $N = 10\ 000$ or $N = 100\ 000$ samples in the simulation.

class	B_k	N	Single shift	Composite distribution
2	$0.587 \cdot 10^{-9}$	10 000	-0.329 ± 0.020	-0.014 ± 0.032
2	$0.587 \cdot 10^{-9}$	100 000	-0.317 ± 0.026	-0.015 ± 0.020
6	$0.244 \cdot 10^{-9}$	10 000	-0.184 ± 0.013	-0.086 ± 0.051
6	$0.244 \cdot 10^{-9}$	100 000	-0.111 ± 0.054	0.025 ± 0.089
4	$0.186 \cdot 10^{-9}$	10 000	-0.051 ± 0.031	-0.043 ± 0.032
4	$0.186 \cdot 10^{-9}$	100 000	0.016 ± 0.039	-0.019 ± 0.027

Table 5.2: The relative error for the estimates and the estimated 95% confidence intervals.

From the results we can clearly see the better accuracy of using the composite distribution, see especially the results of traffic class 2. However, for traffic class 4 it could be seen from the results of the likelihood maximization problem that there is basically only a single link where the main contribution to the blocking probability comes. Then, as the results in the table show, it is sufficient to use only a single shifted distribution.

For the cases covered here Heegaard obtained results with a relative accuracy of approximately $10\% - 20\%$, but the estimated confidence intervals were wide enough to include also the correct values. However, to obtain the results the simulation required 15 replicas of 300 000 regenerative cycles starting from an empty system and ending there.

The results in Table 5.2 also illustrate a serious problem which often appears when using IS for simulation of very rare events: the variances are heavily under estimated. As mentioned earlier, this problem has been studied by Sadowsky in [Sad93]. There he showed that the asymptotically optimal sampling distribution is also asymptotically optimal for estimating all moments of the estimator. For example the estimation of the sample variance requires stability of the fourth moment of the estimator. Instability of the fourth moment results typically in under estimation of the true variance.

An intuitive explanation for this is the following. When using just a single shifted distribution the problem is that the likelihood ratio can have a huge value in some points in the state space, but under the shifted distribution these points have a very small probability, and during the simulation we may never observe these significant points. Hence, the estimates for the mean and, in particular, for the variance, are not accurate. The composite method is not totally immune to this either, since by comparing the reduction in the variance of the estimator when increasing the number of samples from 10 000 to 100 000, the variance is not reduced correspondingly by the factor 10. In fact, from the results of traffic class 6, we can see that although the relative error diminished when increasing the number of the samples the estimated variance actually became larger.

5.4 Conditional Expectation Method in Loss Systems

In this section we first give a general description of a variance reduction method for Monte Carlo simulations published in [Las98b], which is based on the conditional expectation method as described earlier in section 5.1.4 in this thesis and e.g. in [Law91] or [Rub98, p. 97]. Then we show how it can be applied in the context of the multiservice loss system.

5.4.1 The General Formulation

Let us consider a general problem of estimating the expectation

$$H = \mathrm{E}\left[h(\mathbf{X})\right] \tag{5.22}$$

of some function $h(\cdot)$ of a vector random variable $\mathbf{X} \in \mathcal{S}$ with some state space $\mathcal{S}$ and having a distribution P. The Monte Carlo method consists of drawing N independent samples $\mathbf{X}_n,\ n = 1, \ldots, N$, from the distribution P yielding an unbiased estimate

$$\hat{H} = \frac{1}{N} \sum_{n=1}^{N} h(\mathbf{X}_n). \tag{5.23}$$

Now the following elementary identity holds always

$$H = \mathrm{E}\left[h(\mathbf{X})\right] = \mathrm{E}\left[\mathrm{E}\left[h(\mathbf{X}) \mid g(\mathbf{X})\right]\right],$$

where $g(\cdot)$ is another function. Assume that the conditional expectation

$$\eta(\mathbf{x}) = \mathrm{E}\left[h(\mathbf{X}) \mid g(\mathbf{X}) = \mathbf{x}\right]$$

can be calculated analytically. Then the expectation of $h(\mathbf{X})$ becomes

$$H = \mathrm{E}\left[h(\mathbf{X})\right] = \mathrm{E}\left[\eta(g(\mathbf{X}))\right],$$

Correspondingly, we get a new Monte Carlo estimator for H,

$$\hat{H} = \frac{1}{N}\sum_{n=1}^{N}\eta(g(\mathbf{X}_n)). \tag{5.24}$$

More specifically, we consider the case where the state space $\mathcal{S}$ has a partitioning into sets $\mathcal{A}_i,\ i = 1, \dots, I$. A state $\mathbf{X}$ belongs to one and only one of the sets $\mathcal{A}_i$. Let us again denote the unique index of this set by $\iota(\mathbf{X})$. We use this discrete valued function as the function $g(\cdot)$ in the above formulae. So, finally, our estimator is

$$\hat{H} = \frac{1}{N}\sum_{n=1}^{N}\eta(\iota(\mathbf{X}_n)) = \frac{1}{N}\sum_{i=1}^{I}\eta(i)N_i, \tag{5.25}$$

where N_i is the count of the samples having $\iota(\mathbf{X}_n) = i$ or, equivalently, $\mathbf{X}_n \in \mathcal{A}_i$, and $\eta(i) = \mathrm{E}[h(\mathbf{X}) \mid \mathbf{X} \in \mathcal{A}_i]$. In fact, the latter form could have been written directly from

$$\mathrm{E}\left[h(\mathbf{X})\right] = \sum_i \mathrm{E}\left[h(\mathbf{X}) \mid \mathbf{X} \in \mathcal{A}_i\right]\, \mathrm{Pr}\left[\mathbf{X} \in \mathcal{A}_i\right] = \sum_i \eta(i)\mathrm{Pr}\left[\mathbf{X} \in \mathcal{A}_i\right].$$

Since $\eta(i)$ represents the conditional expectation of $h(\mathbf{X})$ over the set $\mathcal{A}_i$ it is intuitively obvious that the variance of estimator (5.25) is smaller than that of (5.23). That this indeed is the case can be seen by calculating the variances of the estimators by conditioning on the value of $\iota(X)$. In the case of estimator (5.23) the variance (multiplied by N) is

$$N\,\mathrm{Var}\left[\hat{H}\right] = \mathrm{Var}\left[h(X)\right] = \mathrm{E}\left[\mathrm{Var}\left[h(X) \mid \iota(X)\right]\right] + \mathrm{Var}\left[\mathrm{E}\left[h(X) \mid \iota(X)\right]\right], \tag{5.26}$$

whereas for estimator (5.25) the same quantity is

$$\begin{aligned} N\,\mathrm{Var}\left[\hat{H}\right] = \mathrm{Var}\left[\eta(\iota(X))\right] &= \mathrm{E}\left[\mathrm{Var}\left[\eta(\iota(X)) \mid \iota(X)\right]\right] + \mathrm{Var}\left[\mathrm{E}\left[\eta(\iota(X)) \mid \iota(X)\right]\right] \\ &= \mathrm{Var}\left[\mathrm{E}\left[h(X) \mid \iota(X)\right]\right], \end{aligned}$$

which shows that sampling the values of the exact conditional expectations eliminates the internal variance of $h(X)$ within each set $\mathcal{A}_i$, i.e. the first term in (5.26).

The method described above is simple. However, it is very useful in cases where one is able to define a partition of the state space $\mathcal{S}$ such that the conditional expectations $\eta(i) = \mathrm{E}[h(\mathbf{X}) \mid \mathbf{X} \in \mathcal{A}_i]$ can be calculated analytically for all i. This requirement is nicely fulfilled by systems with product form state probabilities such as the multiservice loss system.

5.4.2 Application to Loss Systems

Specifically, in the case of the multiservice loss system, we use the same K partitions as with the Gibbs sampler, i.e. partition k consists of columns in the direction k, and $\mathcal{A}_i^k$ denotes the i^{th} k-column in partition k. Now, the blocking probability, given by (2.2), is an expectation of the considered type with $h(\mathbf{X}) = 1_{\mathbf{X}\in\mathcal{B}^k}$. Because of the product form of the state probabilities (2.1) the conditional expectation $\eta^k(i) = \mathrm{E}[1_{\mathbf{X}\in\mathcal{B}^k} \mid \mathbf{X} \in \mathcal{A}_i^k]$ can be calculated easily. The probability distribution constrained to column i in partition k (set $\mathcal{A}_i^k$) is a truncated Poisson distribution and the conditional blocking probability is given by the Erlang loss function (B formula), $\mathrm{erl}(L_i^k, \rho_k)$, where L_i^k denotes the length of the k-column i (set $\mathcal{A}_i^k$). This leads to the estimator

$$\hat{B}_k = \frac{1}{N}\sum_{n=1}^{N} \mathrm{erl}(L^k(\mathbf{X}_n), \rho_k), \tag{5.27}$$

where, for clarity, we have written directly $L^k(\mathbf{X})$ (instead of $L^k_{\iota(\mathbf{X})}$) for the length of the k-column to which the state $\mathbf{X}$ belongs.

This rather obvious decomposition does not seem to have been exploited in the context of simulation of loss systems in spite of its significant advantages. Note that in the standard Monte Carlo simulation, a sample point $\mathbf{X}_n$ gives a contribution to the blocking probability only when it hits the set $\mathcal{B}^k$. In contrast, in the proposed method, for each sample point we collect the conditional expectation over the whole column, always containing a blocking state at the end of the column. Further, note that there is no penalty for this advantage as the values of $\mathrm{erl}(L, \rho)$ can be easily precomputed and stored into an array for all the values of L and ρ needed.

5.4.3 Alternative Application to Loss Networks

In the previous application, the samples $\mathbf{X}_n$ have to be generated in the state space $\mathcal{S}$ from the distribution (2.1). However, as we noted at the end of Chapter 2, eq. (2.3), we can define the blocking probability by considering a random vector $\tilde{\mathbf{X}}$ in a larger state space $\tilde{\mathcal{S}}$. Monte Carlo method can be applied both for the numerator and the denominator leading to the estimator

$$\hat{B}_k = \frac{\sum_{n=1}^{N} 1_{\tilde{\mathbf{X}}_n\in\mathcal{B}^k}}{\sum_{n=1}^{N} 1_{\tilde{\mathbf{X}}_n\in\mathcal{S}}}. \tag{5.28}$$

If the same samples $\tilde{\mathbf{X}}_n$ are used in both the numerator and the denominator this estimator, in effect, reduces to the estimator $\hat{B}_k = 1/N_S \sum_n 1_{\mathbf{X}_n\in\mathcal{B}^k}$, where the samples $\mathbf{X}_n$ are obtained from $\tilde{\mathbf{X}}_n$ by including only those N_S samples which fall within $\mathcal{S}$, i.e. the $\mathbf{X}_n$ are generated by the rejection method.

Something new, however, is obtained when we notice that both the numerator and the denominator of (2.3) can be estimated with the conditional expectation method. To this end we define K partitions of space $\tilde{\mathcal{S}}$ into sets $\tilde{\mathcal{A}}_i^k$ where the sets in the k^{th} partition consists of k-columns in the space $\tilde{\mathcal{S}}$, as illustrated in Fig. 4. Further, we define the conditional

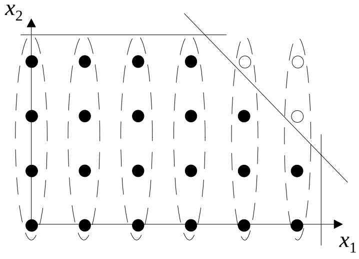

Figure 5.5: New partitioning of the state space.

expectations

$$\begin{aligned}\eta^k(i) &= \mathrm{E}\left[1_{\tilde{\mathbf{X}}\in\mathcal{B}^k} \mid \tilde{\mathbf{X}} \in \tilde{\mathcal{A}}_i^k\right],\\ \vartheta^k(i) &= \mathrm{E}\left[1_{\tilde{\mathbf{X}}\in\mathcal{S}} \mid \tilde{\mathbf{X}} \in \tilde{\mathcal{A}}_i^k\right].\end{aligned}$$

Note that the set $\mathcal{B}^k$ is still formed by the endpoints of the k-columns $\mathcal{A}_i^k$, not by those of the $\tilde{\mathcal{A}}_i^k$. Both of the conditional expectations can be calculated analytically,

$$\eta^k(i) = \begin{cases} 0, & \tilde{\mathcal{A}}_i^k \cap \mathcal{S} = \varnothing,\\ f(L_i^k, \rho_k)/g(N_{\max}^k, \rho_k), & \tilde{\mathcal{A}}_i^k \cap \mathcal{S} \neq \varnothing,\end{cases}$$

$$\vartheta^k(i) = \begin{cases} 0, & \tilde{\mathcal{A}}_i^k \cap \mathcal{S} = \varnothing,\\ g(L_i^k, \rho_k)/g(N_{\max}^k, \rho_k), & \tilde{\mathcal{A}}_i^k \cap \mathcal{S} \neq \varnothing,\end{cases}$$

where, again, L_i^k is the length of the column $\mathcal{A}_i^k$ ($\subseteq \tilde{\mathcal{A}}_i^k$), and, as before, $g(L,\rho) = \sum_{l=0}^{L} \rho^l/l!$.

With the aid of these conditional expectations the blocking probability (2.3) can be written as

$$B_k = \frac{\mathrm{E}[\eta^k(\iota^k(\tilde{\mathbf{X}}))]}{\mathrm{E}[\vartheta^k(\iota^k(\tilde{\mathbf{X}}))]},$$

where $\iota^k(\tilde{\mathbf{X}})$ now denotes the unique index i of set $\tilde{\mathcal{A}}_i^k$ to which $\tilde{\mathbf{X}}$ belongs. The corresponding Monte Carlo estimator becomes

$$\hat{B}_k = \frac{\sum_{n=1}^{N} \eta^k(\iota^k(\tilde{\mathbf{X}}_n))}{\sum_{n=1}^{N} \vartheta^k(\iota^k(\tilde{\mathbf{X}}_n))} = \frac{\sum_{n=1}^{N} f(L(\tilde{\mathbf{X}}_n), \rho_k) 1_{\tilde{\mathbf{X}}_n^{(k)}\in\mathcal{S}}}{\sum_{n=1}^{N} g(L(\tilde{\mathbf{X}}_n), \rho_k) 1_{\tilde{\mathbf{X}}_n^{(k)}\in\mathcal{S}}}, \tag{5.29}$$

where $L^k(\tilde{\mathbf{X}})$ denotes the length of the column $\mathcal{A}_i^k$ to which $\tilde{\mathbf{X}}$ belongs, $\tilde{\mathbf{X}}_n^{(k)}$ is the K-vector obtained from $\tilde{\mathbf{X}}_n$ by setting its k^{th} component to 0 (the set $\tilde{\mathcal{A}}_i^k$ to which $\tilde{\mathbf{X}}_n$ belongs has a nonempty intersection with $\mathcal{S}$ if and only if $\tilde{\mathbf{X}}_n^{(k)} \in \mathcal{S}$), and where we have utilized the fact that $g(N_{\max}^k, \rho_k)$ is constant for traffic class k and cancels out.

In this formulation even a sample $\tilde{\mathbf{X}}_n$ falling outside $\mathcal{S}$ can give a contribution to the numerator and denominator of (5.29).

5.4.4 Numerical results

Here we compare through numerical examples the efficiency of the different methods presented in this paper. As an example we use the same four link star network as studied by Ross in [Ros95, chap 6] with moderate and light traffic loads (cases 1 and 2 in the tables). Blocking probabilities and the 95% confidence intervals are given for two typical traffic classes (out of 12). We also experiment with a larger network (case 3) where the link capacities have been increased roughly by a factor of 20 and the traffic intensities have been increased correspondingly.

In Table 1 we compare the efficiency of the conditional expectation method (CE method in the tables), with samples generated in $\mathcal{S}$ and $\tilde{\mathcal{S}}$, against those obtained with standard Monte Carlo simulation. From the results one can see that by using the conditional expectation method a significant variance reduction is obtained and that the reduction factor increases as the system size increases. Note that if the standard deviation is reduced by a factor of e.g. 7 as in the case 3 for traffic class 2, this corresponds to a reduction by a factor of 49 in the required number of samples.

Table 5.3: Blocking probabilities (%) with confidence intervals

Case	Class	Standard MC	CE method in $\mathcal{S}$	CE method in $\tilde{\mathcal{S}}$
1	2	0.295 ± 0.034	0.287 ± 0.010	0.301 ± 0.010
1	8	1.960 ± 0.090	1.945 ± 0.042	1.976 ± 0.031
2	2	0.052 ± 0.014	0.043 ± 0.006	0.046 ± 0.004
2	8	0.360 ± 0.040	0.343 ± 0.011	0.350 ± 0.012
3	2	0.112 ± 0.021	0.116 ± 0.004	0.114 ± 0.003
3	8	0.600 ± 0.049	0.596 ± 0.006	0.595 ± 0.008

Chapter 6

Conclusions and Future Research

Over the past decade, simulation as a method for obtaining estimates of performance measures has become increasingly popular due to the enormous increase in the computing power of modern computers. In this thesis we have considered the problem of using simulation to efficiently estimate the blocking probabilities of a multiservice loss system.

The model for the loss system and blocking probabilities was defined in chapter 2. There we also considered the exact calculation of blocking probabilities and gave an explicit formulation for the so called convolution method which can, in cases where the number of traffic classes is greater than the number of links in the network, reduce the computational effort over the direct brute force summation approach. Then the applicability of using the loss system to model the call scale behavior of ATM networks was considered. It was concluded that the loss system can be used for this purpose, but the shape of the allowed state space can be potentially difficult to obtain depending on the way how statistical multiplexing effects are taken into consideration.

In chapter 3 we reviewed the literature available on analytical approximations for loss systems. In chapter 4 we presented several different direct simulation methods for simulating the loss system of which the Gibbs sampler appears to be a novel application in this context. Then a number of numerical/analytical studies were made regarding the variance properties of the methods in comparison with their computational effort. Based on our results it appears that when using a DTMC simulation technique the most efficient way in terms of variance and simulation effort is the weighted samples method. However, at the same time it seems that the differences become smaller as the traffic intensities and system size are increased. When comparing the Markov chain methods, the Gibbs sampler and the rejection sampling method, the rejection sampling method gives the most efficient samples but at the highest computational effort per sample. The Gibbs sampler is next in terms of both variance and effort. The DTMC methods (the subchain methods and the weighted samples) have the lowest computational cost per sample but the worst variance performance. Based on the studies it is not, however, possible to determine uniquely the best alternative for simulating the system. Additionally, we developed an analytical method to assess the bias and the deviation of the regenerative estimator for the multiservice loss system as a function of the number of simulated cycles given any choice for the regeneration state.

The downside of the simulation method is that to reach some level of accuracy may require very long simulation times. In chapter 5 we first reviewed the literature on variance reduction techniques, with which simulation times can be reduced. Special attention was given to surveying the literature on rare event simulation. Based on the survey many of the techniques presented there could be applied to loss systems as well, e.g. the gradient estimation techniques and the queuing network heuristics of [Fra91].

Then we presented our application of some large deviation results for multidimensional random variables to loss systems. Previous research on IS distributions for simulating loss systems has considered the use of only one exponentially twisted distribution. We derived an IS distribution which has the form of a composite distribution. The distribution is a weighted combination of several exponentially twisted distributions, each of which corresponds to a distribution for effectively sampling the blocking states on a single link. In [Sad90] the composite form has, in fact, been shown to be asymptotically optimal. However, the asymptotic theory leaves open the choice of the weights. For this, we also presented heuristics, which try to minimize the variance of the samples within the set of the blocking states. This is done by choosing the weights such that the likelihood ratio in the most probable blocking states is a constant. However, the heuristics cannot guarantee that the weights would always be positive. This can be avoided by defining a suitable optimization problem for determining the weights, but we have left the development of this for further study. The numerical results confirm the accuracy of the proposed method.

Another contribution of the thesis is the conditional expectation method, which gives significant variance reduction, and is easy to apply in systems having a product form. The method is based on partitioning the state space into sets such that within each set the conditional expectation of the estimated function can be calculated analytically, which in effect eliminates the internal variance within the set from the estimator. We presented two versions of the method: one where the sets are constrained within the allowed state space and another where we define a larger sampling state space and use the conditional expectation method to collect the contribution of even those samples, which fall outside the allowed state space. Another important property of the method is that it does not incur any extra computational effort, since the information collected for each sample can be precomputed prior to the simulation and stored into arrays without increasing the memory requirements excessively.

All in all, based on the studies made in this thesis, the simulation problem still remains a difficult one mainly due to the dimensionality problem. The conditional expectation method presented here alleviates the problem by reducing the dimensionality essentially by one giving substantial variance reduction. The IS approach, on the other hand, gives huge variance reduction when the blocking probabilities are very small. However, this is not really the case when considering blocking events in realistic networks but even in such cases the proposed IS heuristics still give considerable variance reduction. Thus, it can be said that with the methods presented in this thesis we are able to study notably larger systems than would be possible when using straight forward methods, although we cannot claim that the problem of dimensionality inherent in the system is solved entirely with the proposed techniques. Hence, the problem still leaves room for research especially in the area of attempting to combine different variance reduction methods e.g. the conditional expectations method and IS. More theoretical open issues include the development of the theory of asymptotically optimal sampling distributions for loss systems when the scaling of the system is done such that both the offered loads and the capacities of the links are scaled, as is done when developing analytical approximations for loss systems.

Bibliography

[Ble66] N. Bleistein, “Uniform Asymptotic Expansions of Integrals with Stationary Point Near Algebraic Singularity”, Communications of Pure Applied Mathematics, vol. 19, 1966.

[Bro98] S. P. Brooks, “Markov Chain Monte Carlo and its Application”, The Statistician, vol. 47, 1998.

[Buc90a] J. A. Bucklew, “Large Deviation Techniques in Decision, Simulation and Estimation”, John Wiley & Sons, New York, 1990.

[Buc90b] J. A. Bucklew, P. Ney, J. S. Sadowsky, “Monte Carlo Simulation and Large Deviations Theory for Uniformly Recurrent Markov Chains”, Journal of Applied Probability, vol. 27, pp. 44–59, 1990.

[Cha95] C. S. Chang, P. Heidelberger, P. Shahabuddin, “Fast Simulation of Packet Loss Rates in a Shared Buffer Communications Switch”, ACM Transactions on Modeling and Computer Simulation, vol. 5, no. 4, 1995.

[Cho95a] G. L. Choudhury, K. K. Leung, W. Whitt, “An Algorithm to Compute Blocking Probabilities in Multi-Rate Multi-Class Multi-Resource Loss Models”, Advances in Applied Probability, vol. 27, 1995.

[Cho95b] G. L. Choudhury, K. K. Leung, W. Whitt, “An Inversion Algorithm to Compute Blocking Probabilities in Loss Networks with State-Dependent Rates”, IEEE/ACM Transactions on Networking, vol. 3, no. 5, 1995.

[Chu93] S. Chung, K. W. Ross, “Reduced Load Approximations for Multirate Loss Networks”, IEEE Transactions on Communications, vol. 41, no. 8, Aug. 1993.

[Cot83] M. Cottrell, J. Fort, G. Malgouyres, “Large Deviations and Rare Events in the Study of Stochastic Algorithms”, IEEE Transactions on Automatic Control, vol. 28, no. 9, Sept. 1983.

[Cra75] M. A. Crane, D. L. Iglehart, “Simulating Stable Stochastic Systems: III. Regenerative Processes and Discrete Event Simulations”, Operations Research vol. 23, no. 1, 1975.

[Cra77] M. A. Crane, A. J. Lemoine, “An Introduction to the Regenerative Method for Simulation Analysis”, Springer-Verlag, 1977.

[Dep95] M. de Prycker,“Asynchronous Transfer Mode, Solution for Broadband ISDN”, Prentice-Hall, 1995.

[Dev93a] M. Devetsikiotis, J. K. Townsend, “Statistical Optimization of Dynamic Importance Sampling Parameters for Efficient Simulation of Communication Networks”, IEEE Transactions on Networking, vol. 1, no. 3, June 1993.

[Dev93b] M. Devetsikiotis, J. K. Townsend, “An Algorithmic Approach to the Optimization of Importance Sampling Parameters in Digital Communication System Simulation”, IEEE Transactions on Communications, vol. 41, no. 10, October 1993.

[Dev93c] M. Devetsikiotis, W. A. Al-Qaq, J. A. Freebersyser, J. K. Townsend, “Stochastic Gradient Techniques for the Efficient Simulation of High Speed Networks Using Importance Sampling”, in proceedings of IEEE GLOBECOM ’93, vol. 2, November 1993.

[Dow96] D. G. Down, J. T. Virtamo, “Blocking Probabilities in Multirate Circuit Switched Networks: Capturing Dependent Behaviour”, 13^{th} Nordic Teletraffic Seminar, Aug. 20-22, 1996.

[Dzi87] Z. Dziong, J. W. Roberts, “Congestion Probabilities in a Circuit Switched Integrated Services Network”, Performance Evaluation, vol. 7, 1987.

[Eva91] S. P. Evans,“Optimal Bandwidth Management and Capacity Provision in a Broadband Network Using Virtual Paths”, Performance Evaluation, vol. 13, 1991.

[Fox96] B. L. Fox, P. W. Glynn, “Discrete Time Conversion for Simulating Semi-Markov Processes”, Operations Research Letters, vol. 5, 1986.

[Fra91] M. R. Frater, T. M. Lennon, B. D. O. Anderson, “Optimally Efficient Estimation of the Statistics of Rare Events in Queueing Networks”, IEEE Transactions on Automatic Control, vol. 36, no. 12, Dec. 1991.

[Fra94] M. R. Frater, B. D. O. Anderson, “Fast Simulation of Buffer Overflows in Tandem Networks of $GI/GI/1$ Queues”, Annals of Operations Research, vol. 49, 1994, pp. 207–220.

[Gaz93] P. W. Gazdicki, I. Lambadaris, R. R. Mazumdar, “Blocking Probabilities for Large Multirate Erlang Loss Systems”, Advances in Applied Probability, vol. 25, 1993.

[Gel90] A. E. Gelfand, A. F. M. Smith, “Sampling Based Approaches to Calculating Marginal Densities”, Journal of the American Statistical Association, vol. 85, 1990.

[Gem84] S. Geman, D. Geman, “Stochastic Relaxation, Gibbs distributions and the Bayesian Restoration of Images”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, 1984.

[Gla95] P. Glasserman, S. Kou, "Analysis of an Importance Sampling Estimator for Tandem Queues", ACM Transactions on Modeling and Computer Simulation, vol. 5, no. 1, Jan. 1995.

[Gla96] P. Glasserman, P. Heidelberger, P. Shahabuddin, t. Zajic, "Multilevel Splitting for Estimating Rare Event Probabilities", IBM Research Report RC 20478 (to appear in Operations Research), Yorktown Heights, New York, 1996, available from http://www.research.ibm.com/people/b/berger/papers.html (link verified November 30^{th}, 1998).

[Gla97] P. Glasserman, P. Heidelberger, P. Shahabuddin, T. Zajic, "A Large Deviations Perspective on the Efficiency of Multilevel Splitting", IBM Research Report RC 20691 (to appear in IEEE Transactions on Automatic Control), Yorktown Heights, New York, 1997, available from http://www.research.ibm.com/people/b/berger/papers.html (link verified November 30^{th}, 1998).

[Gly89] P. W. Glynn, D. L. Iglehart, "Importance Sampling for Stochastic Simulations", Management Science, vol. 35, no. 11, Nov. 1989.

[Gly90] P. W. Glynn, "Likelihood Ratio Gradient Estimation for Stochastic Systems", Communications of the ACM, vol. 33, no. 10, 1990.

[Gly92] P. W. Glynn, W. Whitt, "The Asymptotic Efficiency of Simulation Estimators", Operations Research, vol. 40, 1992, pp. 505–520.

[Gly93] P. W. Glynn, "Some Topics in Regenerative Steady State Simulation", Technical Document, Stanford University, 1993, available from http://www-or.stanford.edu/papers/glynn5.ps (link verified November 30^{th}, 1998).

[Gly94] P. W. Glynn, "Importance Sampling for Markov Chains: Asymptotics for the Variance", Stochastic Models, vol. 10, 1994, pp. 701–717.

[Goy92] A. Goyal, P. Shahabuddin, P. Heidelberger, V. F. Nicola, P. W. Glynn, "A Unified Framework for Simulating Markovian Models of Highly Dependable Systems", IEEE Transactions on Computers, vol. 41, no. 1, Jan. 1992.

[Ham67] J. M. Hammersley, D. C. Handscomb, "Monte Carlo Methods", Methuen's Monographs on Applied Probability and Statistics, 1967.

[Har97] Z. Haraszti, J. K. Townsend, "The Theory of Direct Probability Redistribution and its Application to Rare Event Simulation", North Carolina State University CACC Technical Report TR-97/16, 1997, available from http://www.ece.ncsu.edu/cacc/tech_reports/year/y1997.html (link verified November 30^{th}, 1998).

[Has70] W. K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and their Applications", Biometrika, vol. 57, 1970

[Hee97] P. E. Heegaard, “Efficient Simulation of Network Performance by Importance Sampling”, Teletraffic Contributions for the Information Age, Proceedings of ITC-15, vol. 2a, Elsevier, Netherlands.

[Hei95] P. Heidelberger, “Fast Simulation of Rare Events in Queueing and Reliability Models”, ACM Transactions on Modeling and Computer Simulation, vol. 5, no. 1, Jan. 1995.

[Hen97] S. G. Henderson, “Variance Reduction via an Approximating Markov Process”, Ph.D. thesis, Department of Engineering-Economic Systems and Operations Research, Stanford University, 1997, available from http://www-or.stanford.edu/ simlib/theses.html (link verified December 14th, 1998).

[Hsi97] M. Hsieh, “Adaptive Importance Sampling for Rare Event Simulation of Queuing Networks”, Ph.D. thesis, Department of Engineering-Economic Systems and Operations Research, Stanford University, 1997, available from http://www-or.stanford.edu/ simlib/theses.html (link verified December 14th, 1998).

[Hui88] J. Y. Hui, “Resource Allocation for Broadband Networks”, IEEE Journal on Selected Areas in Communications, vol. 6, no. 9, Dec. 1988.

[Hun89] P. J. Hunt, F. P. Kelly, “On Critically Loaded Loss Networks”, Advances in Applied Probability, vol. 21, 1989.

[Ive87] V. B. Iversen, “A Simple Convolution Algorithm for the Exact Evaluation of Multiservice Loss Systems with Heterogeneous Traffic Flows and Access Control”, in proceedings of 7th Nordic Teletraffic Seminar (NTS-7), Lund, August 25-27, 1987.

[Kau81] J. S. Kaufman, “Blocking in a Shared Resource Environment”, IEEE Transactions on Communications, vol. 29, no. 10, Oct. 1981.

[Kel86] F. P. Kelly, “Blocking Probabilities in Large Circuit Switched Networks”, Advances in Applied Probability, vol. 18, 1986.

[Kel91a] F. P. Kelly, “Effective Bandwidths at Multi-class Queues”, Queing Systems, no. 9, 1991.

[Kel91b] F. P. Kelly, “Loss Networks”, The Annals of Applied Probability, no. 1, 1991.

[Lab92] J. P. Labourdette, G. W. Hart, “Blocking Probabilities in Multitraffic Loss Systems: Insensitivity, Asymptotic Behavior and Approximations”, IEEE Transactions on Communications, vol. 40, no. 8, Aug. 1992.

[Las98a] Lassila, P. E., Virtamo J. T., “Using Gibbs Sampler in Simulating Multiservice Loss Systems”, in the proceedings of Performance of Information and Communication Systems ’98 (PICS ’98), Lund, May 25.-28., 1998.

[Las98b] Lassila, P. E., Virtamo J. T., "Variance Reduction in Monte Carlo Simulation of Product Form Systems", IEE Electronics Letters, vol. 34, no. 12, June 1998, p. 1204-1205.

[Las99] Lassila, P. E., Virtamo J. T., "Efficient Importance Sampling for Monte Carlo Simulation of Loss Systems", to appear in the proceedings of the 16^{th} International Teletraffic Congress (ITC–16), Edinburgh, England, June 7–11, 1999.

[Law91] A. M. Law, W. D. Kelton, "Simulation Modeling and Analysis", McGraw-Hill, 1991.

[Lou94] G. Louth, M. Mitzenmacher, F. Kelly, "Computational complexity of loss networks", Theoretical Computer Science 125, 1994.

[Man97] M. Mandjes, "Fast Simulation of Blocking Probabilities in Loss Networks", European Journal of Operations Research, Vol. 101, 1997, pp. 393-405.

[Met53] N. Metropolis, A. W. Rosenbluth, A. H. Teller, E. Teller, "Equations of State Calculations by Fast Computing Machines", Journal of Chemical Physics, vol. 21, 1953.

[Mitr82] I. Mitrani, "Simulation Techniques for Discrete Event Systems", Cambridge Computer Science Texts 14, Cambridge University Press, 1982.

[Mit94] D. Mitra, J. A. Morrison, "Erlang Capacity and Uniform Approximations for Shared Unbuffered Resources", IEEE/ACM Transactions on Networking, vol. 2, no. 6, Dec. 1994.

[Mit95] D. Mitra, J. A. Morrison and K. G. Ramakrishnan, "Unified Approach to Multirate ATM Network Design and Optimization", in proceedings of 9th ITC Specialist Seminar, Leidschendam, The Netherlands, 1995.

[Onv97] R. O. Onvural, R. Cherukuri, "Signaling in ATM Networks", Artech House, 1997.

[Par89] S. Parekh, J. Walrand, "A Quick Simulation Method for Excessive Backlogs in Networks of Queues", IEEE Transactions on Automatic Control, vol. 34, no. 1, Jan. 1989.

[Par93] A. K. Parekh, R. G. Gallager, "A Generalizaed Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single Node Case", IEEE/ACM Transactions on Networking, vol. 1, no. 3, June 1993.

[Rei91] M. I. Reiman, "A Critically Loaded Multiclass Erlang Loss System", Queuing Systems, vol. 9, 1991.

[Rob81] J. W. Roberts, "A Service System with Heterogenous User Requirements - Application to Multi-Services Telecommunications Systems", Performance of Data Communication Systems, North-Holland Publishing Company, 1981.

[Rob96] J. Roberts, U. Mocci, J. Virtamo (eds.), “Broadband Network Teletraffic – Final Report of Action COST 242, Springer, Lecture Notes in Computer Science, 1996.

[Ros93] K. W. Ross, J. Wang, “Asymptotically Optimal Importance Sampling for Product Form Queueing Networks”, ACM Transactions on Modeling and Computer Simulation, vol. 3, no. 3, July 1993.

[Ros95] K. W. Ross, “Multiservice Loss Models for Broadband Telecommunication Networks”, Springer-Verlag, 1995.

[Rub98] R. Y. Rubinstein, B. Melamed, “Modern Simulation and Modeling”, Wiley Series in Probability and Statistics, 1998.

[Sad90] J. S. Sadowsky, J. A. Bucklew, “On Large Deviations Theory and Asymptotically Efficient Monte Carlo Estimation”, IEEE Transactions on Information Theory, vol. 36, no. 3, May 1990.

[Sad91] J. S. Sadowsky, “Large Deviations Theory and Efficient Simulation of Excessive Backlogs in a $GI/GI/m$ Queue”, IEEE Transactions on Automatic Control, vol. 36, no. 12, Dec. 1991.

[Sad93] J. S. Sadowsky, “On the Optimality and Stability of Exponential Twisting in Monte Carlo Estimation”, IEEE Transactions on Information Theory, vol. 39, 1993, pp. 119–128.

[Sch95] A. Schwartz, A. Weiss, “Large Deviations for Performance Analysis”, Chapman-Hall, London, 1995.

[Sha94] P. Shahabuddin, “Importance Sampling for the Simulation of Highly Reliable Markovian Systems”, Management Science, vol. 40, no. 3, 1994.

[Sie76] D. Siegmund, “Importance Sampling in the Monte Carlo Study of Sequential Tests”, Annals of Statistics, vol. 4, 1976, pp. 673–684.

[Sim92] A. Simonian, “Analyse asymptotique des taux de blocage pour un traffic multidébit”, Annales de Télécommunication, vol. 47, 1992.

[Sim97] A. Simonian, J. W. Roberts, F. Théberge, R. Mazumdar, “Asymptotic Estimates for Blocking Probabilities”, Advances in Applied Probability, vol. 29, 1997.

[Sri96] G. Srinivas, “Estimating Probabilities of Rare Events in Regenerative Systems via Importance Sampling”, Master’s Thesis, Department of Electrical Engineering, Indian Institute of Science, Bangalore, 1996.

[Sta93] J. S. Stadler, S. Roy, “Adaptive Importance Sampling”, IEEE Journal on Selected Areas in Communications, vol. 11, no. 3, April 1993.

[Tie94] L. Tierney, “Markov Chains for Exploring Posterior Distributions”, The Annals of Statistics, vol. 22, No. 4, 1994.

[Vil91] M. Villén–Altamirano, J. Villén–Altamirano, “RESTART: A Method for Accelerating Rare Event Simulations”, in proceedings of 13^{th} International Teletraffic Congress (ITC’13): Queueing, Performance and Control in ATM, 1991.

[Vil94] M. Villén–Altamirano, J. Villén–Altamirano, “Enhancement of the Accelerated Simulation Method RESTART by Considering Multiple Thresholds”, in proceedings of 14^{th} International Teletraffic Congress (ITC’14), vol. 1a, 1994.

[Vir88] J. T. Virtamo,“Reciprocity of Blocking Probabilities in Multiservice Loss Systems”, IEEE Transactions in Communications, vol. 36, no. 10, 1988.

[Wro97] J. Wroklawski, “The Use of RSVP with IETF Integrated Services”, IETF Network Working Group, RFC2210.