

Using Gibbs Sampler in Simulating Multiservice Loss Systems

Pasi Lassila, Jorma Virtamo***

Laboratory of Telecommunications Technology, Helsinki University of Technology

P.O.Box 3000, FIN-02015 HUT, Finland, Tel: +358-9-451 2439,
+358-9-451 4783**, e-mail: {Pasi.Lassila, Jorma.Virtamo}@hut.fi*

Abstract

In this article we consider the problem of calculating the blocking probabilities of calls in a multiservice network by using simulation. Traditional simulation methods become computationally intensive as the state space grows. We develop a method that alleviates this problem. The method is based on using the so called Gibbs sampler to generate a Markov chain with the desired stationary distribution. In particular, by making an additional ‘virtual’ step from each state and calculating the expected contribution from this step analytically, we are able to collect information from a subset of the state space for each generated sample. This leads to a smaller variance of the estimate for a given computational effort.

Keywords

loss systems, simulation, Gibbs sampler, variance reduction

1 INTRODUCTION

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 modelled 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. We are interested in calculating the blocking probabilities for each call.

The steady state distribution of the system has the 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. In a network of realistic size the state space very rapidly becomes astronomical.

In such a situation we have two alternatives: to use analytical approximations or to simulate the problem to a desired level of accuracy. In this paper we will be dealing with an efficient method for doing the simulation. Traditionally the simulation approaches have focused on either Monte Carlo (MC) summation techniques or the Markov chain simulation techniques. MC summation has been extensively studied by e.g. Ross (Ross 1995, chap. 6). Markov chain simulation methods include the regenerative method (Crane *et al.* 1975, Crane *et al.* 1977), which has been lately used in the context of rare event simulation in loss networks by Heegaard (Heegaard 1997).

The problem with the aforementioned methods is that they become computationally intensive as the state space grows. The reason is that they collect information about the state space on a state-per-state basis. What is actually needed is a method that can collect more information with each sample than that represented by the sample itself.

In this article, we will present a method that is based on a family of simulation methods called Markov Chain Monte Carlo (MCMC) methods using the Gibbs sampler (Tierney 1994). This method enables us to exploit the product form solution of the system, and we are able to calculate part of the problem analytically while the simulation is being carried out. In practice, this means that with each generated sample we can collect information of not just the current sample state, but a large number of other surrounding states.

The rest of the paper is organized as follows. Chapter 2 introduces the basic stochastic model of the problem. Chapter 3 describes the Gibbs sampling method for loss networks. Chapter 4 contains the main results of this paper with methods for improving the performance of the Gibbs sampler. Numerical results are presented in chapter 5 and the conclusions in chapter 6.

2 MODEL DESCRIPTION

Consider a network consisting of J links, indexed with $j = 1, \dots, J$, each having a capacity of C_j resource units. The network supports K classes of calls. Associated with a class- k call, $k = 1, \dots, K$, is an offered load ρ_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 $\mathbf{b}_j = (b_{j,1}, \dots, b_{j,K})$ denote the required bandwidths of different classes 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 $\mathbf{x} = (x_1, \dots, x_K)$, where element x_k is the number of class- k calls in progress.

The set of allowed states \mathcal{S} can be described as $\mathcal{S} = \{\mathbf{x} : \mathbf{b}_j \cdot \mathbf{x} \leq C_j, \forall j\}$.

$j = 1, \dots, J\}$, where the scalar product is defined as $\mathbf{b}_j \cdot \mathbf{x} = \sum_k b_{j,k}x_k$. This system has the well known product form stationary distribution

$$\pi(\mathbf{x}) = \frac{1}{G} \prod_{k=1}^K \frac{\rho_k^{x_k}}{x_k!} = \frac{1}{G} \prod_{k=1}^K f_k(x_k) = \frac{f(\mathbf{x})}{G}, \quad \text{with } G = \sum_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}), \quad (1)$$

where $f_k(x_k) = \rho_k^{x_k}/x_k!$, and $f(\mathbf{x})$ denotes the unnormalized state probability. G is the normalization constant.

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

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

where \mathbf{e}_k is a K -component vector with 1 in the k th component and zeros elsewhere. The blocking probability of a class- k call, B_k , is then

$$B_k = \sum_{\mathbf{x} \in \mathcal{B}^k} \pi(\mathbf{x}) = \sum_{\mathbf{x} \in \mathcal{S}} \pi(\mathbf{x}) 1_{\mathbf{x} \in \mathcal{B}^k}. \quad (2)$$

In the remainder of this paper we will be dealing with an efficient simulation method for calculating the blocking probabilities.

3 GIBBS SAMPLING FOR LOSS SYSTEMS

Our problem is now of the following type. We want to evaluate some quantity H defined as the sum of a function $h(\cdot)$ over the allowed state space \mathcal{S} ,

$$H = \sum_{\mathbf{x} \in \mathcal{S}} h(\mathbf{x}). \quad (3)$$

In general, the MC method solves the problem by generating identically distributed samples $\mathbf{X}_n \in \mathcal{S}$ from some distribution $p(\mathbf{x}) = \Pr[\mathbf{X}_n = \mathbf{x}]$ such that $p(\mathbf{x}) \neq 0, \forall \mathbf{x} \in \mathcal{S}$. With respect to this distribution H can be written as an expectation

$$H = \sum_{\mathbf{x} \in \mathcal{S}} \frac{h(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) = E_p [h(\mathbf{X})/p(\mathbf{X})].$$

The estimator for H when N samples have been drawn is

$$\hat{H} = \frac{1}{N} \sum_{n=1}^N \frac{h(\mathbf{X}_n)}{p(\mathbf{X}_n)}. \quad (4)$$

Estimator (4) has the correct expectation when each \mathbf{X}_n has the distribution p , irrespective of whether the \mathbf{X}_n are independent or not. Positive correlation between the samples, however, would make the estimator less efficient from the point of view of its variance.

In our case we are interested in calculating the blocking probabilities as given by eq. (2) with $h(\mathbf{x}) = \pi(\mathbf{x})\mathbf{1}_{\mathbf{x} \in \mathcal{B}^k}$. Then a natural choice is to let $p(\mathbf{x}) = \pi(\mathbf{x})$, and we get the estimator

$$\hat{B}_k = \frac{1}{N} \sum_{n=1}^N \mathbf{1}_{\mathbf{X}_n \in \mathcal{B}^k}. \quad (5)$$

One approach for generating the samples is by Markov chain simulation. This relies on the fact that, assuming the holding times are exponentially distributed, our system itself is defined by a Markov chain, e.g. the embedded discrete time Markov chain (jump chain) associated with the arrival and departure epochs. The points in the full jump chain, when weighted with the expected life time of each state, have the stationary distribution π and, as noted above, can be used as samples in the summation of eq. (5) despite the fact that they are not independent.

In MCMC methods the idea is the same – to simulate some Markov chain for constructing the distribution π . The question is only: are there other Markov chains that have the same steady state distribution π ? The answer is yes and, in fact, many of them (Tierney 1994). The Gibbs sampler introduced later in this chapter is just one of them, but its properties allow us to exploit the product form of the steady state distribution to gain significant simulation efficiency increases, as will be discussed in chapter 4.

3.1 General Theory

Let $\mathbf{X} = (X_1, \dots, X_K) \in \mathcal{S}$ denote the vector random variable with the distribution $\pi(\mathbf{x})$ as in (1). We are now interested in ways of constructing a Markov chain \mathbf{X}_n^* having the invariant distribution π . One way is to use transition probabilities based on conditioning, as defined in the following theorem (taken with slight modification from (Tierney 1994)).

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

$$\Pr [\mathbf{X}_{n+1}^* = \mathbf{y} \mid \mathbf{X}_n^* = \mathbf{x}] = \Pr [\mathbf{X} = \mathbf{y} \mid \mathbf{X} \in \mathcal{A}(\mathbf{x})] \quad (6)$$

has the invariant distribution π .

Proof:

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

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

$$\Pr [\mathbf{X}_{n+1}^* = \mathbf{y}] = \sum_i \Pr [\mathbf{X} = \mathbf{y} \mid \mathbf{X} \in \mathcal{A}_i] \Pr [\mathbf{X} \in \mathcal{A}_i] = \Pr [\mathbf{X} = \mathbf{y}] = \pi(\mathbf{y}) \quad \square$$

Let $P^{(1)}$ denote the transition probability matrix with the components given by eq. (6). The Markov chain generated by this transition matrix is not irreducible, because there are no transitions between different sets. However, by defining several partitions $1, \dots, M$ we can construct an irreducible Markov chain \mathbf{X}_n^* . Let $P^{(m)}$, $m = 1, \dots, M$, 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 $P = P^{(1)} \dots P^{(M)}$ will be irreducible. Since each $P^{(m)}$ has the invariant distribution π also the compound matrix P will have the invariant distribution π , and because \mathbf{X}_n^* is now irreducible, π is also its unique stationary distribution.

3.2 Gibbs Sampler and its Application to Loss Networks

In our case we have a product form solution π 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.

We define K partitions and denote the particular set in partition k to which a state \mathbf{x} belongs by $\mathcal{A}^k(\mathbf{x})$. This set consists of the states

$$\mathcal{A}^k(\mathbf{x}) = \{\mathbf{y} = (x_1, \dots, x_{k-1}, l, x_{k+1}, \dots, x_K) : l \in \mathcal{I}, \mathbf{b}_j \cdot \mathbf{y} \leq C_j, j = 1, \dots, J\},$$

where \mathcal{I} is the set of non-negative integers. For the sequel, we denote by $L^k(\mathbf{x})$ the largest value of the component k (variable l) in the above set $\mathcal{A}^k(\mathbf{x})$.

To illustrate the concept consider the simple network of Fig. 1, with the state space depicted in the same figure. In this case we have two traffic classes,

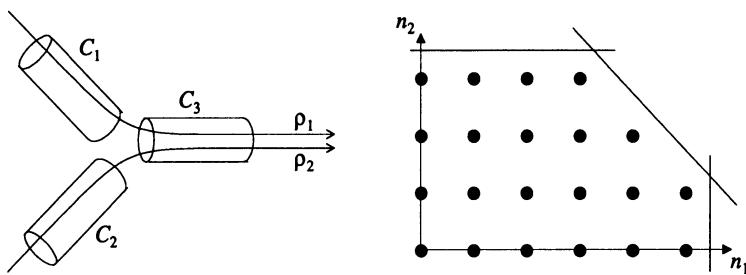


Figure 1 Example network and its state space.

$K = 2$, and we use two different partitions with the ‘rows’ corresponding to partition 1 and the ‘columns’ to partition 2 (see Fig. 2).

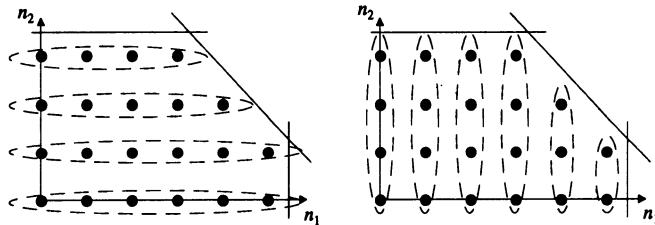


Figure 2 The state space partitions 1 and 2 (left and right).

Associated with each partition, there is a transition matrix $P^{(k)}$ with the transition probabilities (6). Then we construct a compound transition matrix $P = P^{(1)} \dots P^{(K)}$. The corresponding Markov chain \mathbf{X}_n^* is irreducible since it is possible to move from any state \mathbf{x} in the coordinate convex state space S to any other state \mathbf{y} with at most K transitions.

The simulation of the Markov chain \mathbf{X}_n^* consists of making transitions with the transition matrices $P^{(k)}$ in cyclical order, see Fig. 3. In transitions generated with $P^{(k)}$, the state remains in one of the sets $\mathcal{A}^k(\mathbf{x})$, i.e. only the component x_k changes. Starting from the state \mathbf{X}_n^* the value of x_k of the next state is obtained by drawing it from the distribution $f_k(x_k)/G_k(L^k(\mathbf{X}_n^*))$ in the range $0, \dots, L^k(\mathbf{X}_n^*)$, where the normalizing function $G_k(\cdot)$ is defined by

$$G_k(L) = \sum_{l=0}^L f_k(l) .$$

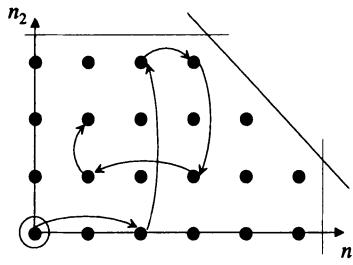


Figure 3 Gibbs sampler example.

Along the generated path, we collect information about the number of visits to the blocking states of class- k calls in order to form the estimator (5), i.e.

$$\hat{B}_k = \frac{1}{N} \sum_{n=1}^N \mathbf{1}_{\mathbf{x}_n^* \in \mathcal{B}^k} .$$

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 MC summation 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, much less correlated than the samples from the embedded Markov chain.

3.3 Uniform Sampling with Gibbs Sampler

This idea of partitioning the state space can also be used to generate a Markov chain with a uniform distribution over the complete allowed state space \mathcal{S} . In general, if we want to use uniformly distributed samples from the state space we will have $p(\mathbf{x}) = 1/S$ in estimator (4), where S denotes the size of the state space \mathcal{S} . The uniform distribution is trivially of the product form and the Gibbs sampler is applicable. Now, starting from the state \mathbf{X}_n^* the transitions generated with $P^{(k)}$ are obtained by drawing x_k of the next state

from the uniform distribution in the range $0, \dots, L^k(\mathbf{X}_n^*)$. The estimator from (4) for the blocking probability of a class- k call is then

$$\hat{B}_k = \frac{S}{N} \sum_{n=1}^N \pi(\mathbf{X}_n^*) 1_{\mathbf{X}_n^* \in \mathcal{B}^k} = \frac{S}{NG} \sum_{n=1}^N f(\mathbf{X}_n^*) 1_{\mathbf{X}_n^* \in \mathcal{B}^k}. \quad (7)$$

G is unknown in (7), but similarly from (1) and (4) we can estimate G by

$$\hat{G} = \frac{S}{N} \sum_{n=1}^N f(\mathbf{X}_n^*). \quad (8)$$

Using this in (7) we get another estimator

$$\hat{B}_k = \frac{\sum_{n=1}^N f(\mathbf{X}_n^*) 1_{\mathbf{X}_n^* \in \mathcal{B}^k}}{\sum_{n=1}^N f(\mathbf{X}_n^*)} = \frac{\hat{G}_k^B}{\hat{G}}, \quad (9)$$

where \hat{G}_k^B is the estimator for $G_k^B = \sum_{\mathbf{x} \in \mathcal{B}^k} f(\mathbf{x})$.

The uniform sampling may not be a very effective way to do the sampling, because it often ‘wastes’ time on sampling every part of the state space with equal probability, when some parts of the state space are actually more important than the others. In particular, in our case when the distribution $\pi(\mathbf{x})$ is concentrated in a small part of the state space \mathcal{S} , the uniform sampling does not necessarily produce very good results.

4 IMPROVED GIBBS SAMPLING METHOD

The method as described in the previous chapter does not yet give any significant improvement over the known techniques. We can, however, improve the efficiency of the method considerably by utilizing prior knowledge about the conditional distributions of the sets.

4.1 Improved Poisson Sampling

The idea is simple: At each step of the chain, starting from the current state \mathbf{X}_n^* , we make a ‘virtual’ transition with transition matrix $P^{(k)}$ to the state \mathbf{Y}_n^* . Since in stationary state \mathbf{X}_n^* has the distribution π and $P^{(k)}$ has the invariance

property, the state \mathbf{Y}_n^* after the virtual transition also has the distribution π . We can then use \mathbf{Y}_n^* as a sample point instead of \mathbf{X}_n^* and get the estimator

$$\hat{B}_k = \frac{1}{N} \sum_{n=1}^N 1_{\mathbf{Y}_n^* \in \mathcal{B}^k} . \quad (10)$$

Now it is possible to take into account the effect of the extra step analytically by calculating the expectation of (10) conditioned on the \mathbf{X}_n^* (see Appendix for a formal explanation on this step). This is easy to do since

$$E [1_{\mathbf{Y}_n^* \in \mathcal{B}^k} | \mathbf{X}_n^*] = \frac{f_k(L_n^k)}{G_k(L_n^k)} = B(L_n^k, \rho_k) ,$$

where $L_n^k = L^k(\mathbf{X}_n^*)$ is the largest value of x_k in the set $\mathcal{A}^k(\mathbf{x})$ and $B(n, \rho)$ is the well known Erlang loss function (B-formula). Then our estimator becomes

$$\hat{B}_k = \frac{1}{N} \sum_{n=1}^N \frac{f_k(L_n^k)}{G_k(L_n^k)} = \frac{1}{N} \sum_{n=1}^N B(L_n^k, \rho_k) . \quad (11)$$

In effect, by this method we have included transitions to all states in $\mathcal{A}^k(\mathbf{x})$. In particular, a contribution from a blocking state is obtained for every point \mathbf{X}_n^* in the Markov chain. It is easy to see that the variance of estimator (11) is smaller than that of (5). They are both unbiased estimators with the same expectation. In case of (5), the sample values are either 0 or 1, while in (11) the samples belong to the range $(0, 1]$.

Note that the function $f_k(\cdot)/G_k(\cdot) = B(\cdot, \rho_k)$ can be precomputed for all k and all required values of the argument. It is also worth noting that our improvement method does not require the use of the Gibbs sampler to generate the samples \mathbf{X}_n^* . Actually, they can be generated by any means provided that the \mathbf{X}_n^* have the distribution π . Thus using traditional MC summation techniques to generate the \mathbf{X}_n^* is also possible.

4.2 Improved Uniform Sampling

The improvements described in the previous chapter can also be used when the state space is sampled with a uniform distribution. Again, let \mathbf{Y}_n^* denote the state after making a virtual transition from the state \mathbf{X}_n^* having the stationary distribution and consider using \mathbf{Y}_n^* for the estimator

$$\hat{B}_k = \frac{1}{GN} \sum_{n=1}^N f(\mathbf{Y}_n^*) 1_{\mathbf{Y}_n^* \in \mathcal{B}^k} .$$

Calculating the expectation of this extra virtual step conditioned on \mathbf{X}_n^* gives

$$\mathbb{E} [f(\mathbf{Y}_n^*) \mathbf{1}_{\mathbf{Y}_n^* \in \mathcal{B}^k} \mid \mathbf{X}_n^*] = \frac{1}{L_n^k} f^{(k)}(\mathbf{X}_n^*) f_k(L_n^k) ,$$

where $f^{(k)}(\mathbf{X}) = \prod_{l \neq k} f_l(X_l)$, and, as before, $L_n^k = L^k(\mathbf{X}_n^*)$ is the largest value of x_k in the set $\mathcal{A}^k(\mathbf{x})$. The estimator then becomes

$$\hat{B}_k = \frac{1}{GN} \sum_{n=1}^N \frac{1}{L_n^k} f^{(k)}(\mathbf{X}_n^*) f_k(L_n^k) , \quad (12)$$

where the unknown G can similarly be estimated using the extra ‘virtual’ step,

$$\hat{G} = \frac{1}{N} \sum_{n=1}^N \frac{1}{L_n^k} f^{(k)}(\mathbf{X}_n^*) \sum_{l=0}^{L_n^k} f_k(l) = \frac{1}{N} \sum_{n=1}^N \frac{1}{L_n^k} f^{(k)}(\mathbf{X}_n^*) G_k(L_n^k) . \quad (13)$$

Combining (12) and (13) gives the following ratio estimator for the blocking probabilities

$$\hat{B}_k = \frac{\sum_{n=1}^N f^{(k)}(\mathbf{X}_n^*) f_k(L_n^k) / L_n^k}{\sum_{n=1}^N f^{(k)}(\mathbf{X}_n^*) G_k(L_n^k) / L_n^k} . \quad (14)$$

Again, the functions $f_k(\cdot)$ and $G_k(\cdot)$ can be separately precalculated and stored into arrays before the actual simulation run.

5 NUMERICAL RESULTS

Our numerical example consists of the star network with 12 traffic classes studied by Ross (Ross 1995, p. 240) for the moderate traffic case. We compare our results with the results obtained using the importance sampling heuristics in (Ross 1995, chap. 6). Table 1 shows the results we obtained from using the improved Poisson sampling and uniform sampling (MCMC*) and the results of Ross (Ross 1995, p. 243). However, Ross obtained his results by generating 100 000 i.i.d. MC samples and we used 50 independent runs of simulations with 50 000 cycles (each roughly corresponding to one MC sample). In order to make the results comparable, we have scaled our results by multiplying the confidence intervals by $\sqrt{25}$. As expected, the uniform sampling method does not give particularly good results, but the Poisson sampling method gives confidence intervals almost one third of those of Ross’ indicating clear variance reduction.

Table 1 Blocking probabilities (%) with confidence intervals

Class	Ross	MCMC* (Poisson)	MCMC* (Uniform)
1	0.336 ± 0.024	0.349 ± 0.008	0.347 ± 0.067
2	0.283 ± 0.022	0.299 ± 0.008	0.303 ± 0.077
3	0.275 ± 0.022	0.291 ± 0.008	0.283 ± 0.502
4	0.072 ± 0.009	0.069 ± 0.003	0.070 ± 0.015
5	0.064 ± 0.009	0.061 ± 0.003	0.064 ± 0.017
6	0.011 ± 0.003	0.011 ± 0.001	0.011 ± 0.003
7	2.270 ± 0.070	2.291 ± 0.025	2.256 ± 0.330
8	1.935 ± 0.065	1.965 ± 0.022	1.973 ± 0.332
9	1.885 ± 0.065	1.911 ± 0.026	1.919 ± 0.320
10	0.488 ± 0.025	0.493 ± 0.009	0.502 ± 0.092
11	0.438 ± 0.024	0.433 ± 0.009	0.452 ± 0.087
12	0.073 ± 0.008	0.081 ± 0.003	0.084 ± 0.016

We also made tests on the rare event example of Heegaard (Heegaard 1997), but noticed a tendency to underestimate the probabilities. The reason is that when performing rare event simulation the Markov chain of the Gibbs sampler is confined to move only within a very small part of the whole state space. Then it does not necessarily sample the most important blocking states for all traffic classes. Therefore, when dealing with rare event simulation our improved method needs to be combined with importance sampling methods to shift the probability mass closer to the state space boundaries.

6 CONCLUSIONS

In this article we have presented an efficient simulation method for calculating the blocking probabilities for calls in a multiservice loss system. The method is based on the use of the Gibbs sampler with an appropriate partition of the state space. We are then able to exploit the product form solution by calculating analytically the effect of using as samples not the current state \mathbf{X}_n^* of the simulation of the Markov chain, but states after a ‘virtual’ step from the current state generated with the transition matrix $P^{(k)}$. It is then possible to calculate analytically the probability of this new sample hitting the blocking state of a class- k call. Thus for each state \mathbf{X}_n^* in the chain we get a contribution from the blocking state for class- k call leading to clear reduction in the variance over the traditional MC summation techniques as shown by our numerical results. Also, it was shown that the improved method does not lead to excessive computational complexity.

ACKNOWLEDGEMENT

The authors wish to thank Samuli Aalto for many useful discussions during the course of the work.

REFERENCES

- Crane, M.A. and Iglehart, D.L. (1975) Simulating Stable Stochastic Systems: III. Regenerative Processes and Discrete Event Simulations. *Operations Research*, vol. 23, no. 1, 33–45.
- Crane, M.A. and Lemoine, A.J. (1977) An Introduction to the Regenerative Method for Simulation Analysis. Springer-Verlag, Berlin.
- Heegaard P.E. (1997) Efficient Simulation of Network Performance by Importance Sampling. *Teletraffic Contributions for the Information Age. Proceedings of ITC-15*, vol. 2a, Elsevier, Netherlands, 623–632.
- Ross K.W. (1995) Multiservice Loss Models for Broadband Telecommunication Networks. Springer-Verlag, Berlin.
- Tierney L. (1994) Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, vol. 22, No. 4, 1701–1762.

APPENDIX

Consider the sum

$$H = \sum_{\mathbf{x} \in S} h(\mathbf{x})p(\mathbf{x}) = E[h(\mathbf{X})] ,$$

where \mathbf{X} is a random variable with the distribution $p(\mathbf{x})$, and the corresponding estimator

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

where \mathbf{X}_n are samples of the random variable \mathbf{X} .

Let P be any transition matrix with invariant distribution $p(\mathbf{x})$. Then \mathbf{Y} which is obtained from \mathbf{X} with this matrix also has the distribution $p(\mathbf{x})$. H may then be written as

$$H = E[h(\mathbf{Y})] = E[E[h(\mathbf{Y}) | \mathbf{X}]] .$$

Assume now that $E[h(\mathbf{Y}) | \mathbf{X}]$ can be calculated analytically. Then a new estimator for H can be written

$$\hat{H} = \frac{1}{N} \sum_{n=1}^N E[h(\mathbf{Y}) | \mathbf{X} = \mathbf{X}_n] ,$$

which for each sample \mathbf{X}_n includes all the points reachable from \mathbf{X}_n with the transition matrix P .