Probability in the Engineering and Informational Sciences, 6, 1992, 323–348. Printed in the U.S.A.

MONTE CARLO SUMMATION APPLIED TO PRODUCT-FORM LOSS NETWORKS

KEITH W. ROSS AND JIE WANG
Department of Systems
University of Pennsylvania
Philadelphia, Pennsylvania 19104

Loss networks with direct routing have a product-form solution for their equilibrium probabilities. The product-form solution typically involves a normalization constant calling for a multidimensional summation over an astronomical number of states. We propose the application of Monte Carlo summation in order to determine the normalization constant, the blocking probabilities, and the revenue sensitivities. We show that if the proper sampling technique is employed, then the computational effort of Monte Carlo summation is independent of link capacities. We also discuss the application of importance sampling, antithetic variates, and indirect estimation via Little's formula. The method is illustrated with a four-leaf star network supporting multirate traffic.

1. INTRODUCTION

A loss network is a stochastic network that blocks, rather than queues, a connection request when the available resources are insufficient for immediate service. Loss networks can be used to model telephone networks and broadband telecommunication networks supporting multirate connections such as voice, video, and facsimile.

Loss networks with direct routing give rise to a product-form solution for their equilibrium probabilities. The appeal of the product-form solution is that

This research supported partially by NSF grant DDM-9000481 and partially by AT&T Bell Laboratories.

 0269-9648/92 $5.00 + .00

it enables one to circumvent solving the global balance equations associated with the underlying Markov process. But for problems of practical interest, a summation must still be performed over an astronomical number of states in order to calculate the associated normalization constant.

Several efficient combinatorial algorithms for calculating the normalization constant and blocking probabilities have been developed. Kaufman [7] and Roberts [15] independently developed a very efficient algorithm to obtain the blocking probabilities for a single-link network with multirate traffic. Ross and Tsang [16,19] developed efficient algorithms for multirate tree and hierarchial tree networks. However, these combinatorial algorithms require loss networks with specific topologies. It appears that efficient combinatorial algorithms for general topologies are difficult, if not impossible, to develop.

McKenna, Mitra, and Ramakrishnan [10,11,13,14] have pioneered the use of asymptotic expansions for calculating the normalization constant in product-form queueing networks. This technique can rapidly solve problems that are completely off limits to the combinatorial algorithms. However, except for a very particular topology [12], little progress has been reported for asymptotic expansions for loss networks.

In this paper, we pursue a third school of thought, namely, applying Monte Carlo summation to the problem of evaluating the normalization constant. Unlike the combinatorial methods, the Monte Carlo summation method has computational requirements that grow polynomially in the problem size.

Harvey and Hills [3] have considered a related Monte Carlo method—rejection sampling combined with conditional Monte Carlo—for solving a class of loss networks. The computational effort required by their method is not very encouraging; furthermore, conditional Monte Carlo does not offer a significant reduction of variance for networks with more than a small number of links. Our study differs from theirs in its concern with the application of *sampling techniques* and *variance reduction techniques* to large stochastic networks. We show that a major reduction in the computational effort is possible with the proper sampling technique, and that variance reduction techniques, other than conditional Monte Carlo, can significantly reduce confidence intervals. Our study also differs from that of Harvey and Hills [3] in that we are interested in estimating revenue sensitivity as well as blocking probabilities.

In Section 2 we discuss the application on Monte Carlo summation and importance sampling to product-form summands. Special attention is given to ratio estimation since many performance measures in stochastic networks can be expressed as the ratio of two normalization constants. In Section 3, we apply the Monte Carlo summation technique to loss networks. Significant reduction in variance with the proper choice of importance sampling function is shown to be possible for test networks. Optimal importance sampling and indirect estimation via Little's formula are also explored. In Section 4, we study Monte Carlo summation techniques for estimating revenue sensitivity and associated confidence intervals.

2. OVERVIEW OF THE MONTE CARLO SUMMATION METHOD

Let Ω denote the state space of the underlying stochastic process. Each element in Ω is a K-dimensional vector $\mathbf{n} = (n_1, \ldots, n_K)$. The normalization constant, g, for product-form stochastic networks most commonly takes the form

$$g := \sum_{\mathbf{n}\in\Omega} \prod_{k=1}^{K} q_k(n_k), \tag{1}$$

where $q_k(\cdot)$, $k = 1, \ldots, K$, are known functions. Let

$$q(\mathbf{n}) := 1(\mathbf{n} \in \Omega) \prod_{k=1}^{K} q_k(n_k),$$

where $1(\cdot)$ is the indicator function. We can rewrite Eq. (1) as follows:

$$g = \sum_{n_1=0}^{N_1} \cdots \sum_{n_K=0}^{N_K} q(\mathbf{n}),$$

where $N_k := \max\{n_k : \mathbf{n} \in \Omega\}$. Thus calculating g involves a multidimensional summation. But it is now the belief of many researchers that, in the absence of special structure, multidimensional integration (or summation) is best solved by Monte Carlo methods [6]. Specifically, let $\mathbf{V}^i = (V_1^i, \ldots, V_K^i)$, $i = 1,2,\ldots$, be a sequence of i.i.d. random vectors, where each $\mathbf{V}^i$ takes values in $\Lambda := \{0, \ldots, N_1\} \times \cdots \times \{0, \ldots, N_K\}$. Let $p(\mathbf{n}) := P(\mathbf{V}^i = \mathbf{n})$, $\mathbf{n} \in \Lambda$, which is a sampling distribution to be specified in order to obtain the maximum efficiency from the Monte Carlo method, and let

$$Z^i := \frac{q(\mathbf{V}^i)}{p(\mathbf{V}^i)}.$$

Then

$$\bar{Z}_n := \frac{1}{n} \sum_{i=1}^{n} Z^i$$

is an unbiased estimator for g (i.e., $E[\bar{Z}_n] = g$). Moreover, we have from the Central Limit Theorem that for large n

$$P\left(|\bar{Z}_n - g| \le \frac{c(\alpha)\sigma_n(Z)}{\sqrt{n}} \right) = 1 - \frac{\alpha}{2}, \tag{2}$$

where $c(\alpha)$ is the critical value of the standard normal distribution and $\sigma_n^2(Z)$ is the sample variance of Z^i, $i = 1, \ldots, n$; i.e.,

$$\sigma_n^2(Z) = \frac{1}{n-1} \sum_{i=1}^{n} (Z^i - \bar{Z}_n)^2.$$

Note that for any fixed n, $\bar{Z}_n$ is an estimate of g whose accuracy can be assessed by the confidence interval $\bar{Z}_n \pm c(\alpha)\sigma_n(Z)/\sqrt{N}$ induced by Eq. (2). As

the samples are being drawn, the sample variance can be calculated and the confidence intervals can be given explicitly. Furthermore, if greater accuracy is desired, more samples can be drawn, thereby decreasing the width of the confidence interval. Note that this method is particularly well suited for optimization because only rough estimates are needed for performance measures and gradients when the current solution is not close to the optimal. Moreover, we shall see that the gradients of the performance measures can be obtained with little additional effort (which is the main feature of perturbation analysis [4]).

From Eq. (2) we observe that the effectiveness of the Monte Carlo summation method largely depends on

1. the effort required to generate $\mathbf{V}^i$ from the distribution $p(\mathbf{n})$, $\mathbf{n} \in \Lambda$,
2. the effort required to evaluate the "integrand" $q(\cdot)/p(\cdot)$ during the sampling procedure, and
3. σ^2, the variance of Z^i.

If $V_1^i, \ldots, V_K^i$ are independent (i.e., $p(\mathbf{n}) = p_1(n_1) \cdots p_K(n_K)$), then $\mathbf{V}^i$ can be generated in a total of $O(K)$ time with the alias algorithm (e.g., see Bratley, Fox, and Schrage [1]). Note that this effort is independent of N_k, $k = 1, \ldots, K$, the maximum values of the stochastic process. This means that the method has potential to handle loss networks with large link capacities.

It has been repeatedly observed in the Monte Carlo integration literature that the variance σ^2 can often be significantly reduced by choosing the appropriate sampling distribution $p(\mathbf{n})$, $\mathbf{n} \in \Lambda$. In particular, it is desirable to sample more frequently the points $\mathbf{n}$ at which $q(\mathbf{n})$ is "important," which is typically done by considering functions $p(\cdot)$ that are similar to $q(\cdot)$. Ideally, one would like $q(\cdot)/p(\cdot)$ to be nearly constant; however, there exists a tradeoff between this similarity and the effort required to sample from $p(\cdot)$.

2.1. Comparison with Discrete Event Simulation

Traditional discrete event simulation generates a sequence of random states that mimic the dynamics of the stochastic system. The output observations are generally correlated so that many independent replications are needed in order to generate confidence intervals. Furthermore, it has the disadvantage of being sensitive to the effects of the initial transient and wasteful of data if the transient portion is discarded from the beginning of each replication. It appears that no effective method has been developed for estimating revenue sensitivity for loss networks (by perturbation analysis or other means). Finally, variance reduction techniques for discrete event simulation of stochastic networks have rarely been successful in real applications [9].

The Monte Carlo summation method also generates a sequence of random states but, in contrast to discrete event simulation, it exploits the product-form solution. Independent outputs are generated so that only one run is necessary to obtain confidence intervals; furthermore, the run does not have a transient.

With Monte Carlo summation, revenue sensitivity can be obtained with little additional effort beyond what is required to obtain estimates of performance measures. Monte Carlo summation also offers enormous potential for variance reduction.

Of course, one major advantage of discrete event simulation is that it is not restricted to product-form stochastic networks.

2.2. Ratio Estimators

The estimator $\bar{Z}_n$ is useful for calculating the normalization constant for a product-form network. However, many performance measures of interest are given by nonlinear functions of normalization constants; in particular, acceptance probabilities for loss networks take the form of a ratio:

$$\phi := \frac{\sum_{\mathbf{n}\in\Lambda} f(\mathbf{n})q(\mathbf{n})}{\sum_{\mathbf{n}\in\Lambda} q(\mathbf{n})}, \tag{3}$$

where $f(\cdot)$ is a known function. A natural estimate for ϕ therefore is

$$\Phi_n := \frac{\sum_{i=1}^{n} Y^i}{\sum_{i=1}^{n} Z^i}, \tag{4}$$

where $Y^i := f(\mathbf{V}^i)q(\mathbf{V}^i)/p(\mathbf{V}^i)$ and $Z^i := q(\mathbf{V}^i)/p(\mathbf{V}^i)$.

Although Φ_n converges (almost surely) to ϕ, Φ_n has the undesirable property of being biased. However, this bias diminishes as n becomes large. Moreover, the ratio estimator Φ_n can be made free of bias to order $1/n$ with a simple modification that requires an insignificant amount of additional CPU time (see Fishman [2, pp. 55–59]). We should also stress that the confidence interval for ϕ can again be constructed as the sampling proceeds (see Fishman [2, pp. 59–61]): it is obtained on line from the sample mean, variance, and covariance of Y^i and Z^i as follows. Let $\bar{Y}_n$ and $\sigma_n^2(Y)$ be the sample mean and variance associated with Y_i, $i = 1, \ldots, n$ (analogous to $\bar{Z}_n$ and $\sigma_n^2(Z)$). Further, let

$$\sigma_n^2(Y, Z) = \frac{1}{n-1}\sum_{i=1}^{n}(Y^i - \bar{Y}_n)(Z^i - \bar{Z}_n)$$

be the sample covariance associated with the two sets of random variables. Then the $(1-\alpha)100\%$ confidence interval for Φ_n is

$$\left(\frac{\bar{Y}_n\bar{Z}_n - \frac{c^2(\alpha)}{n}\sigma_n^2(Y,Z) - r_n}{\bar{Z}_n^2 - \frac{c^2(\alpha)}{n}\sigma_n^2(Z)}, \frac{\bar{Y}_n\bar{Z}_n - \frac{c^2(\alpha)}{n}\sigma_n^2(Y,Z) + r_n}{\bar{Z}_n^2 - \frac{c^2(\alpha)}{n}\sigma_n^2(Z)}\right),$$

where $c(\alpha)$ is the critical value of the standard normal distribution and r_n is given by

$$r_n = \sqrt{\left[\bar{Y}_n\bar{Z}_n - \frac{c^2(\alpha)}{n}\sigma_n^2(Y,Z)\right]^2 - \left[\bar{Z}_n^2 - \frac{c^2(\alpha)}{n}\sigma_n^2(Z)\right]\left[\bar{Y}_n^2 - \frac{c^2(\alpha)}{n}\sigma_n^2(Y)\right]}.$$

Note that the width of the confidence interval is $O(1/\sqrt{n})$.

We will see in Section 4 that estimates for revenue sensitivity do not take the form of a ratio but are, instead, given by more complicated nonlinear expressions. In Section 4, we derive confidence intervals for revenue sensitivity.

2.3. Rejection Techniques

Harvey and Hills [3] have considered a rejection technique for estimating blocking probability for loss networks. We now discuss this method in a more general context (i.e., arbitrary product-form networks) and show that it is a special case of the ratio estimate (3).

Let $\mathbf{X}^i$, $i = 1,2,\ldots$, be i.i.d. random vectors with distribution

$$P(\mathbf{X}^i = \mathbf{n}) = \frac{q(\mathbf{n})}{g}, \qquad \mathbf{n} \in \Omega.$$

Then, we see from ratio estimate (3) that

$$\Psi_n = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{X}^i) \tag{5}$$

is an unbiased estimator for ϕ. But to construct this estimator, we must have an efficient means of generating $\mathbf{X}^i$ with the required distribution. To this end, let $U_1^i,\ldots,U_K^i$ be independent random variables with distributions

$$P(U_k^i = n) = \frac{q_k(n)}{\sum_{l=0}^{N_k} q_k(l)}, \qquad k = 1,\ldots,K,$$

and let $\mathbf{U}^i = (U_1^i,\ldots,U_K^i)$. If we let $T_1, T_2,\ldots$ be the sequence of times that $\mathbf{U}^i \in \Omega$, then $\mathbf{X}^i = \mathbf{U}^{T_i}$ has the desired distribution. Note that $\mathbf{U}^i$ is rejected whenever $\mathbf{U}^i \notin \Omega$. The method studied by Harvey and Hills [3] is this rejection method specialized to product-form loss networks. Clearly, the effectiveness of the method deteriorates as $P(\mathbf{U}^i \notin \Omega)$ increases.

It is important to note that the rejection estimator Ψ_n is essentially equivalent to ratio estimator Φ_n when samples $\mathbf{V}^i$ are generated according to

$$p(\mathbf{n}) = \prod_{k=1}^{K} \frac{q_k(n_k)}{\sum_{l=0}^{N_k} q_k(l)}.$$

In this case we can set $\mathbf{V}^i = \mathbf{U}^i$ so that

$$\Psi_n = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{X}^i)$$

$$= \frac{\sum_{i=1}^{T_n} f(\mathbf{U}^i)(\mathbf{U}^i \in \Omega)}{\sum_{i=1}^{T_n} 1(\mathbf{U}^i \in \Omega)}$$

$$= \Phi_{T_n}.$$

Note that the estimators Ψ_{T_n} and Φ_n both require the generation of the same number of U_i's.

3. PRODUCT-FORM LOSS NETWORKS

Consider a loss network with links $j = 1, \ldots, J$, where link j has C_j circuits. Suppose that the network supports K classes of connections, where each class is distinguished by its route (i.e., a subset of the J links), its bandwidth requirement (the number of circuits required on each link), and the arrival and service rates. Connection requests are assumed to arrive according to a Poisson process with class-dependent rate λ_k. Let A_{jk} be the number of circuits required by a class-k connection on link j. (In ordinary "single-rate" telephone networks, A_{jk} is equal to 0 or 1.) When a class-k connection request arrives, it is set up in the network if the number of busy circuits on link j is $\leq C_j - A_{jk}$ for all $j = 1, \ldots, J$; otherwise, it is blocked and assumed lost. The holding-time distribution for a class-k connection has an arbitrary distribution; denote $1/\mu_k$ for its mean. Also denote $\rho_k := \lambda_k/\mu_k$ for the offered load of class-k connections.

Denote $\mathbf{A}$ for the $J \times K$ dimensional matrix with elements A_{jk}; denote $\mathbf{A}_{\cdot k}$ and $\mathbf{A}_{j\cdot}$ for the k th column and the j th row, respectively, of the matrix $\mathbf{A}$. Let $\mathbf{C} = (C_1, \ldots, C_J)$ be the vector of link capacities. The state of the system is defined by the vector $\mathbf{n} = (n_1, \ldots, n_K)$ where n_k represents the number of class-k connections currently in the system. The set of all possible states is $\Omega := \Omega(\mathbf{C})$, where

$$\Omega(\mathbf{C}) := \{\mathbf{n} : \mathbf{An} \leq \mathbf{C}\}.$$

Also let $\Omega_k := \Omega(\mathbf{C} - \mathbf{A}_{\cdot k})$ and $\Omega_{kl} := \Omega(\mathbf{C} - \mathbf{A}_{\cdot k} - \mathbf{A}_{\cdot l})$.

Denote $\pi(\mathbf{n})$ for the equilibrium probability of being in state $\mathbf{n}$. It is well known (e.g., see Kelly [8]) that

$$\pi(\mathbf{n}) = \frac{\prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}}{g(\mathbf{C})}, \qquad \mathbf{n} \in \Omega,$$

route. The routes, bandwidth requirements, and traffic intensities are specified for the 12 classes in Table 1.

There are over a trillion states in this example, and none of the known combinatorial or asymptotic techniques apply to this network. Table 2 illustrates the performance of the estimate $\bar{Z}_n$ for the normalization constant in light, moderate, and heavy traffic. For each case, we have 95% confidence intervals with $n = 100{,}000$ based on (i) $\boldsymbol{\gamma} = \boldsymbol{\rho}$, (ii) $\boldsymbol{\gamma} \neq \boldsymbol{\rho}$, and (iii) $\boldsymbol{\gamma} = \boldsymbol{\rho}$ with antithetic variates (see Bratley et al. [1] for a description of antithetic variates). For the case $\boldsymbol{\gamma} \neq \boldsymbol{\rho}$, which we henceforth refer to as the importance sampling method, the importance sampling parameters γ_k, $k = 1, \ldots, K$, are pseudo-optimal in the sense that a large variety of $\boldsymbol{\gamma}$'s were tried and those reported are those that gave the best performance in the sense of reducing the width of the confidence intervals (uniformly over k). In Section 3.4, we shall give a heuristic for choosing good importance sampling parameters when calculating the confidence intervals for blocking probabilities.

The "improvement factor" given in the various tables is defined as the width of the confidence interval for the case $\boldsymbol{\gamma} = \boldsymbol{\rho}$ divided by the width of the confidence interval for the variance reduction technique (i.e., $\boldsymbol{\gamma} \neq \boldsymbol{\rho}$ or antithetic variates in Table 2). We see from Table 2 that importance sampling and antithetic variates do not provide significant variance reduction in light and moderate traffic. However, both techniques give reduction for heavy traffic. Note that the importance sampling parameters are smaller than the corresponding offered loads (for both moderate and heavy traffic cases), causing the samples $\mathbf{v}^i$ to fall more frequently in Ω.

Table 1. Network Data

Class	Route	Bandwidth Requirement	Offered Load			
			Light	Moderate	Heavy	Super Heavy
1	1,2	1	9.0	10.0	15.0	16.0
2	1,3	1	9.0	10.0	15.0	16.0
3	1,4	1	9.0	10.0	15.0	16.0
4	2,3	1	9.0	10.0	15.0	16.0
5	2,4	1	9.0	10.0	15.0	16.0
6	2,5	1	9.0	10.0	15.0	16.0
7	1,2	5	1.6	2.0	3.0	5.0
8	1,3	5	1.6	2.0	3.0	5.0
9	1,4	5	1.6	2.0	3.0	5.0
10	2,3	5	1.6	2.0	3.0	5.0
11	2,4	5	1.6	2.0	3.0	5.0
12	3,4	5	1.6	2.0	3.0	5.0

TABLE 2. Confidence Intervals for Normalization Constants

Traffic	$\gamma = \rho$	Importance Sampling		Antithetic Variates	
	Confidence Intervals	Confidence Intervals	Improvement Factors	Confidence Intervals	Improvement Factors
Light[a]	(41,682, 41,707)	(41,682, 41,707)	1.00	(41,682, 41,707)	1.00
Moderate[b]	(18,179, 18,212)	(18,179, 18,211)	1.03	(18,181, 18,213)	1.03
Heavy[c]	(33,088, 33,578)	(33,022, 33,421)	1.23	(33,162, 33,494)	1.48

[a]Confidence interval endpoints should be multiplied by 10^{23}. For importance sampling, $\gamma = \rho$ gave the best results.
[b]Confidence interval endpoints should be multiplied by 10^{27}. For importance sampling, the following parameters were used: $\gamma_1 = \ldots = \gamma_6 = 9.99$, $\gamma_7 = \ldots = \gamma_{12} = 1.985$.
[c]Confidence interval endpoints should be multiplied by 10^{42}. For importance sampling, the following parameters were used: $\gamma_1 = \ldots = \gamma_6 = 14.7$, $\gamma_7 = \ldots = \gamma_{12} = 2.7$.

3.2. Estimating Blocking Probabilities

Now consider the problem of estimating acceptance probabilities for class-k connections via the ratio estimate (4) with the importance sampling function (6):

$$\Phi_n = \frac{\sum_{i=1}^{n} Y^i}{\sum_{i=1}^{n} Z^i} = \frac{\sum_{i=1}^{n} \alpha^i 1(\mathbf{V}^i \in \Omega_k)}{\sum_{i=1}^{n} \alpha^i 1(\mathbf{V}^i \in \Omega)}. \tag{8}$$

Tables 3–5 illustrate the performance of the estimator for the network of Figure 1 with $n = 100{,}000$, again in light, moderate, and heavy traffic. (The results are given in terms of blocking probabilities.) For each case, we have estimates based on (i) $\boldsymbol{\gamma} = \boldsymbol{\rho}$ and (ii) $\boldsymbol{\gamma} \neq \boldsymbol{\rho}$. In Tables 3–5, the importance sampling parameters for $\boldsymbol{\gamma} \neq \boldsymbol{\rho}$ were determined from the heuristic given in Section 3.4. Each traffic condition (light, moderate, heavy) required 60–70 seconds of CPU time on a Sun 4/280 in order to estimate *all* blocking probabilities. For light traffic, the conditions under which telephone networks typically operate, importance

Table 3. Confidence Intervals for Percent Blocking for Light Traffic

Class	$\gamma = \rho$	Importance Sampling[a]	Improvement Factors
1	(.040,.069)	(.041,.052)	2.64
2	(.038,.066)	(.037,.047)	2.40
3	(.037,.065)	(.036,.046)	2.80
4	(0,.008)	(.004,.008)	2.00
5	(0,.006)	(.003,.007)	1.50
6	(0,.003)	(.000,.001)	3.00
7	(.35,.43)	(.34,.38)	2.00
8	(.32,.40)	(.30,.34)	2.00
9	(.32,.39)	(.30,.33)	2.33
10	(.032,.058)	(.049,.060)	2.36
11	(.024,.048)	(.044,.054)	2.40
12	(.003,.015)	(.005,.007)	6.00

[a]For importance sampling, the following parameters were used: $\gamma_1 = \ldots = \gamma_6 = 9.585$, $\gamma_7 = \ldots = \gamma_{12} = 2.192$.

TABLE 4. Confidence Intervals for Percent Blocking for Moderate Traffic

Class	$\gamma = \rho$	Importance Sampling[a]	Improvement Factors
1	(.298,.370)	(.312,.359)	1.53
2	(.261,.329)	(.261,.305)	1.55
3	(.253,.320)	(.253,.297)	1.52
4	(.044,.070)	(.063,.081)	1.44
5	(.031,.064)	(.055,.072)	1.94
6	(.005,.018)	(.008,.013)	2.60
7	(2.20,2.39)	(2.20,2.34)	1.36
8	(1.87,2.05)	(1.87,2.00)	1.39
9	(1.81,1.98)	(1.82,1.95)	1.31
10	(.467,.557)	(.463,.513)	1.80
11	(.403,.486)	(.414,.462)	1.73
12	(.072,.110)	(.065,.080)	2.53

[a]For importance sampling, the following parameters were used: $\gamma_1 = \ldots = \gamma_6 = 10.500$, $\gamma_7 = \ldots = \gamma_{12} = 2.553$.

sampling gives improvement factors of 1.5–6.0, with most of them being greater than 2.0. This is significant because the procedure would otherwise have to run about four times longer to achieve an improvement factor of 2. (The confidence interval width is proportional to $n^{-1/2}$.) For moderate traffic, the improvement

TABLE 5. Confidence Intervals for Percent Blocking for Heavy Traffic

Class	$\gamma = \rho$
1	(5.465,5.910)
2	(4.421,4.825)
3	(3.939,4.321)
4	(2.310,2.608)
5	(1.835,2.102)
6	(0.772,0.949)
7	(27.94,28.80)
8	(23.18,23.99)
9	(21.13,21.93)
10	(13.14,13.80)
11	(10.88,11.49)
12	(4.631,5.043)

factors range from 1.31 to 2.60, which is still significant. But in contrast to the results for the normalization constant, importance sampling does not give a significant reduction in the confidence interval width for blocking probabilities in heavy traffic. (In fact, the heuristic of Section 3.4 gives $\boldsymbol{\gamma} = \boldsymbol{\rho}$.) Also note that, for light and moderate traffic, the importance sampling parameters are now larger than the corresponding offered loads, causing the samples to fall near the boundary of Ω more frequently.

Computational testing for antithetic variates was also carried out: Little or no variance reduction was observed for all three traffic conditions.

Let us now determine the computational effort required by one iteration of the Monte Carlo summation algorithm corresponding to the estimator Φ_n. We do this for a star network (as in Fig. 1) with J leaves and one central node. We also suppose that there is one class for each pair of leaves and that each class requires one circuit per link. Thus, there are $J(J-1)/2$ classes, and each class employs one circuit in each of the two links along its route. For this network, the effort required to generate $\mathbf{V}^i$ is $O(J^2)$. The effort required to determine $\xi_j = \Sigma_k A_{jk} V_k^i$ for $j = 1, \ldots, J$ is $O(J^2)$. The effort required to determine if $\xi_j \leq C_j$ for all $j = 1, \ldots, J$ is $O(J)$. Thus, the effort required to determine if $\mathbf{V} \in \Omega$ is $O(J^2)$. If $\mathbf{V} \in \Omega$, we must then proceed to determine if $\mathbf{V} \in \Omega_k$ for each class $k = 1, \ldots, K$. This requires an additional $O(J^2)$ time, since for each class k we compare ξ_j with $C_j - A_{jk}$ for two j's and there are $O(J^2)$ classes. Thus, the overall effort required to update Φ_n for all classes is $O(J^2) = O(K)$. Note that the effort is independent of the capacity of the links. When all the links have the same capacity C, the memory requirement of the algorithm is $O(CJ^2)$.

We performed some computational testing for a large star network, as described in the preceding paragraph, with 20 links and 100 circuits on each link. We therefore had 190 classes. Once again, 100,000 iterations were performed in each run. We set $\rho_k = 3.8$ for each class k, which corresponds to light traffic conditions (i.e., small blocking probabilities). Note that the network is symmetric. For the case of $\gamma_k = 3.8$ for all k (i.e., $\boldsymbol{\gamma} = \boldsymbol{\rho}$), we obtained the following confidence intervals for *percent* blocking for the first three classes: (.064,.100), (.046,.078), and (.050,.0.82). For the case $\gamma_k = 3.96$ (which is specified by our heuristic in Section 3.4), we obtained (.045,.070), (.046,.076), and (.042,.067), which corresponds to improvement factors of 1.44, 1.06, and 1.28, respectively. The CPU time required was 887 seconds for $\gamma = 3.8$ and 998 seconds for $\gamma = 3.96$. Note that these CPU times are more or less consistent with the CPU times for the 12-class network and what is predicted by the complexity analysis.

3.3. Optimal Importance Sampling

In the empirical work described above, we used importance sampling functions of the form of function (6). This function was chosen because the corresponding samples and estimates can be easily constructed. We now derive the optimal

importance sampling function $p(\mathbf{n})$, $\mathbf{n} \in \Lambda$, for ratio estimators of the form of function (4). Although it is difficult to implement this optimal scheme in practice, the analysis sheds insight on the proper choice of sampling parameters.

Recall the definition of the ratio estimator Φ_n given by function (4). To simplify the discussion, let $f(\mathbf{n}) = 1(\mathbf{n} \in \Omega_k)$ so that $\phi = 1 - \beta_k$, i.e., ϕ is the acceptance probability of class-k connections. Let $\text{var}_p(\Phi_n)$ denote the variance of Φ_n with respect to the probability measure p. We are interested in minimizing $\text{var}_p(\Phi_n)$ for large n. To this end, let

$$H(p) := \lim_{n\to\infty} n \,\text{var}_p(\Phi_n)$$

be the asymptotic cost corresponding to p. We shall say that p^* is *optimal* if it minimizes $H(p)$.

THEOREM 1: *The optimal importance sampling function is given by*

$$p^*(\mathbf{n}) = \begin{cases} \dfrac{q(\mathbf{n})}{2\phi g} & \mathbf{n} \in \Omega_k \\ \dfrac{q(\mathbf{n})}{2(1-\phi)g} & \mathbf{n} \in \Omega - \Omega_k \\ 0 & \mathbf{n} \in \Lambda - \Omega, \end{cases}$$

where $\Omega_k := \Omega(\mathbf{C} - \mathbf{A}_{\cdot k})$. *The asymptotic cost for the optimal sampling function is*

$$H(p^*) = 4\phi^2(1-\phi)^2.$$

The estimator corresponding to p^* *is*

$$\Phi_n = \frac{\phi \sum_{i=1}^{n} 1(\mathbf{V}^i \in \Omega_k)}{\phi \sum_{i=1}^{n} 1(\mathbf{V}^i \in \Omega_k) + (1-\phi) \sum_{i=1}^{n} 1(\mathbf{V}^i \in \Omega - \Omega_k)}.$$

PROOF: From Fishman [2, p. 59], we have

$$H(p) = \frac{\phi^2}{g_k^2 g^2} E_p[(gY^i - g_k Z^i)^2]$$

$$= \frac{1}{g^4} \sum_{\mathbf{n}\in\Lambda} \frac{c(\mathbf{n})}{p(\mathbf{n})},$$

where $g_k := g(\mathbf{C} - \mathbf{A}_{\cdot k})$ and $c(\mathbf{n}) := q^2(\mathbf{n})[g1(\mathbf{n} \in \Omega_k) - g_k 1(\mathbf{n} \in \Omega)]^2$. Consider minimizing the preceding expression for $H(p)$ subject to the constraints $\sum_{\mathbf{n}\in\Lambda} p(\mathbf{n}) = 1$, $p(\mathbf{n}) \geq 0$, $\mathbf{n} \in \Lambda$. This is a standard resource allocation problem, whose solution is given by

$$p^*(\mathbf{n}) = \frac{\sqrt{c(\mathbf{n})}}{\sum_{\mathbf{n}\in\Lambda} \sqrt{c(\mathbf{n})}}$$

$$= \begin{cases} \dfrac{q(\mathbf{n})|g1(\mathbf{n} \in \Omega_k) - g_k|}{\sum_{\mathbf{n}\in\Omega} q(\mathbf{n})|g1(\mathbf{n}\in\Omega_k) - g_k|} & \mathbf{n} \in \Omega \\ 0 & \mathbf{n} \in \Lambda - \Omega. \end{cases}$$

A straightforward calculation gives

$$\sum_{\mathbf{n}\in\Omega} q(\mathbf{n})|g1(\mathbf{n} \in \Omega_k) - g_k| = 2g_k(g - g_k).$$

The desired results directly follow. ■

It is of interest to note that the optimal importance sampling function, p^*, satisfies

$$\sum_{\mathbf{n}\in\Omega_k} p^*(\mathbf{n}) = \sum_{\mathbf{n}\in\Omega-\Omega_k} p^*(\mathbf{n}) = 1/2,$$

$$\sum_{\mathbf{n}\in\Lambda-\Omega} p^*(\mathbf{n}) = 0;$$

i.e., ideally half of the samples fall in Ω_k, half of the samples fall in $\Omega - \Omega_k$, and no samples fall in $\Lambda - \Omega$. We also observe that a sample $\mathbf{V}$ can be generated from $p^*(\mathbf{n})$ according to the following procedure. First, flip an unbiased coin. If the result is heads, generate (independent) $\mathbf{U}$'s, as defined in Section 2.3, and set $\mathbf{V} = \mathbf{U}$ the first time $\mathbf{U} \in \Omega_k$. If the result is tails, generate $\mathbf{U}$'s and set $\mathbf{V} = \mathbf{U}$ the first time $\mathbf{U} \in \Omega - \Omega_k$. Although this rejection procedure is straightforward, the scheme cannot be implemented in practice because the estimator Φ_n given in Theorem 1 requires knowledge of ϕ, which is what we are trying to estimate in the first place.

Now let us compare p^* with $\tilde{p}$, where

$$\tilde{p}(\mathbf{n}) = \begin{cases} \dfrac{q(\mathbf{n})}{g} & \mathbf{n} \in \Omega \\ 0 & \mathbf{n} \in \Lambda - \Omega. \end{cases}$$

Note that $\tilde{p}$ corresponds to the pure rejection technique discussed in Section 2.3, which is equivalent to using the importance sampling procedure of function (6) with $\boldsymbol{\rho} = \boldsymbol{\gamma}$ (see discussion at end of Section 2.3). If $\phi > \frac{1}{2}$ (i.e., less than 50% blocking for class-k connections), then

$$p^*(\mathbf{n}) < \tilde{p}(\mathbf{n}), \qquad \mathbf{n} \in \Omega_k,$$

$$p^*(\mathbf{n}) > \tilde{p}(\mathbf{n}), \qquad \mathbf{n} \in \Omega - \Omega_k.$$

This result seems to indicate that, in light and moderate traffic, a good sampling function p would encourage the samples to fall further from the origin than those samples generated from $\tilde{p}$. This has been found to be true in our empirical studies.

3.4. A Heuristic for Choosing Sampling Parameters

Returning to the sampling function of the form (6), we now develop a heuristic for choosing the sampling parameters $\gamma_1, \ldots, \gamma_K$. Based on the observations of Section 3.3, the sampling procedure should attempt to satisfy the following two criteria when estimating blocking probabilities:

1. Only a small fraction of the samples fall in $\Lambda - \Omega$.
2. A significant fraction of the samples should fall near the boundary of Ω.

If we set all of the γ_k's close to zero, then the vast majority of the samples will fall in Ω_k, in which case the first criterion will be satisfied but not the second. On the other hand, if the γ_k's are large, the second criterion may be satisfied but not the first. Therefore, the two criteria are conflicting and it is necessary to compromise.

But how can we determine, a priori, where the samples are going to fall relative to the boundary of Ω? If $\gamma_k < N_k$ (as is typically the case) then $E[V_k] \approx \gamma_k$. Thus if

$$\sum_{k=1}^{K} A_{jk}\gamma_k > C_j$$

for at least one link, then one expects the majority of the samples to fall outside of Ω. In this case, we would want to decrease the values of those γ_k's such that $A_{jk} > 0$. On the other hand, if

$$\sum_{k=1}^{K} A_{jk}\gamma_k \ll C_j,$$

then very few samples will fall near the boundary of Ω. In this case, we would want to increase the γ_k's for those connections k such that $A_{jk} > 0$.

Based on the preceding principles and on observations made from trial runs, we have developed the following heuristic for choosing the sampling parameters $\gamma_1, \ldots, \gamma_K$. First we calculate

$$\delta := \max_{1 \le j \le J} \frac{\sum_{k=1}^{K} A_{jk}\rho_k}{C_j},$$

and

$$b_k := \max_{1 \le j \le J} A_{jk}, \qquad k = 1, \ldots, K.$$

Table 6. Improvement Factors for Indirect Estimation

Class	Heavy	Super Heavy
1	0.95	1.54
2	0.85	1.39
3	0.81	1.23
4	0.62	1.08
5	0.55	0.98
6	0.36	0.85
7	1.03	2.00
8	0.93	1.75
9	0.90	1.72
10	0.70	1.67
11	0.63	1.49
12	0.40	0.43

4. REVENUE SENSITIVITY

Now suppose that a class-k connection, once admitted into the network, generates revenue at a rate of r_k units per second. The average revenue accrued by the network is

$$w = \sum_{k=1}^{K} r_k \rho_k (1 - \beta_k).$$

Manipulating the product-form solution (e.g., see Hunt [5]) gives the following expression for revenue sensitivity:

$$\frac{\partial w}{\partial \rho_l} = \frac{1}{\rho_l} \sum_{k=1}^{K} r_k \left[\frac{h_{kl}}{g(\mathbf{C})} - \frac{h_k h_l}{g^2(\mathbf{C})} \right], \tag{11}$$

where

$$h_k := \sum_{\mathbf{n} \in \Omega} n_k \prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!}, \qquad k = 1,2,\ldots,K,$$

$$h_{kl} := \sum_{\mathbf{n} \in \Omega} n_k n_l \prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!} \qquad k,l = 1,2,\ldots,K.$$

In this section, we are interested in developing point estimates and associated confidence intervals for $\partial w/\partial \rho_l$. As for the case of estimating blocking probabilities, this can be done both "directly" and "indirectly."

4.1. Revenue Sensitivity via Direct Estimation

It is easily shown that

$$h_k = \rho_k g(\mathbf{C} - \mathbf{A}_{\cdot k})$$

and

$$h_{kl} = \begin{cases} \rho_k \rho_l g(\mathbf{C} - \mathbf{A}_{\cdot k} - \mathbf{A}_{\cdot l}) & k \neq l \\ \rho_k^2 g(\mathbf{C} - \mathbf{A}_{\cdot k} - \mathbf{A}_{\cdot k}) + \rho_k g(\mathbf{C} - \mathbf{A}_{\cdot k}) & k = l. \end{cases}$$

Therefore, we can write Eq. (11) as

$$\begin{aligned} \frac{\partial w}{\partial \rho_l} &= \frac{1}{\rho_l}\left[\frac{\sum_{k=1}^{K} r_k \rho_k \rho_l g(\mathbf{C} - \mathbf{A}_{\cdot k} - \mathbf{A}_{\cdot l}) + r_l \rho_l g(\mathbf{C} - \mathbf{A}_{\cdot l})}{g(\mathbf{C})} - \frac{\rho_l g(\mathbf{C} - \mathbf{A}_{\cdot l}) \sum_{k=1}^{K} r_k \rho_k g(\mathbf{C} - \mathbf{A}_{\cdot k})}{g(\mathbf{C})^2}\right] \\ &= \frac{\sum_{\mathbf{n}\in\Lambda}\left(\prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!}\right)\sum_{k=1}^{K} r_k \rho_k 1(\mathbf{n} \in \Omega_{kl}) + \sum_{\mathbf{n}\in\Omega_l}\left(\prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!}\right) r_l}{\sum_{\mathbf{n}\in\Omega}\prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!}} \\ &\quad - \frac{\left[\sum_{\mathbf{n}\in\Omega_l}\prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!}\right]\sum_{\mathbf{n}\in\Lambda}\left(\prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!}\right)\sum_{k=1}^{K} r_k \rho_k 1(\mathbf{n} \in \Omega_k)\right]}{\left[\sum_{\mathbf{n}\in\Omega}\prod_{m=1}^{K} \frac{\rho_m^{n_m}}{n_m!}\right]^2}. \end{aligned} \tag{12}$$

Therefore, a point estimator for $\partial w/\partial p_l$ is

$$\begin{aligned} S_n &= \frac{\sum_{i=1}^{n} \alpha^i \sum_{k=1}^{K} r_k \rho_k 1(\mathbf{V}^i \in \Omega_{kl}) + \sum_{i=1}^{n} \alpha^i r_l 1(\mathbf{V}^i \in \Omega_l)}{\sum_{i=1}^{n} \alpha^i 1(\mathbf{V}^i \in \Omega)} \\ &\quad - \frac{\left[\sum_{i=1}^{n} \alpha^i 1(\mathbf{V}^i \in \Omega_l)\right]\left[\sum_{i=1}^{n} \alpha^i \sum_{k=1}^{K} r_k \rho_k 1(\mathbf{V}^i \in \Omega_k)\right]}{\left[\sum_{i=1}^{n} \alpha^i 1(V^i \in \Omega)\right]^2}, \end{aligned}$$

where $\gamma_1, \ldots, \gamma_K$ are importance sampling parameters that appear in function (6). This estimator is consistent (i.e., it converges to $\partial w/\partial \rho_l$ almost surely).

Observe that, at each iteration, the following quantities have to be calculated in order to construct the estimator:

We can therefore construct a confidence interval for $\partial w/\partial \rho_k$ as follows. We generate the sequences $\{N_l^i, i = 1,2,\dots\}$ and $\{D_l^i, i = 1,2,\dots\}$ and keep track of their sample means, sample variances, and sample covariance. We then invoke the formulas for confidence intervals for ratio estimation, given in Section 2.3, with Y and Z replaced by N_l and D_l, respectively. Note that two $\mathbf{V}$'s must now be generated in order to construct N_l^i and D_l^i for a single fixed i.

The preceding formulas correspond to indirect estimation. Analogous formulas can be developed from Eq. (12) for direct estimation.

Tables 7 and 8 present computational results for revenue sensitivity with direct and indirect estimation, respectively. In both cases $n = 100{,}000$ and the importance sampling parameter γ_k is set to ρ_k, $k = 1,\dots,12$. For light and moderate traffic, the confidence intervals associated with direct estimation are substantially smaller. For heavy traffic, however, indirect estimation gives remarkably smaller confidence intervals. Recall that it was necessary to go to *super-heavy* traffic in order to get this effect when estimating *blocking probabilities* in Section 3.1.

5. CONCLUSION

We have shown that Monte Carlo summation can be a viable technique for estimating blocking probabilities and revenue sensitivities for product-form loss networks. In addition, importance sampling and indirect estimation can offer significant variance reduction for a wide range of networks and traffic conditions. Finally, we have studied the computational effort and accuracy for two techniques for estimating revenue sensitivity.

TABLE 7. Confidence Intervals for Revenue Sensitivity with Direct Estimation

Class	r_k	Light	Moderate	Heavy
1	1.0	(0.98,1.00)	(0.87,0.93)	(−0.01,0.80)
2	1.2	(1.18,1.20)	(1.09,1.15)	(0.52,1.24)
3	1.4	(1.38,1.40)	(1.30,1.35)	(0.60,1.26)
4	1.6	(1.59,1.60)	(1.55,1.58)	(1.08,1.71)
5	1.8	(1.79,1.80)	(1.75,1.78)	(1.22,1.77)
6	2.0	(2.00,2.00)	(1.98,2.00)	(1.73,2.15)
7	3.0	(2.86,2.98)	(2.32,2.42)	(−0.48,0.58)
8	3.6	(3.48,3.51)	(3.03,3.12)	(0.88,1.85)
9	4.2	(4.08,4.11)	(3.65,3.73)	(1.45,2.37)
10	4.8	(4.77,4.78)	(4.57,4.62)	(1.90,2.83)
11	5.4	(5.37,5.38)	(5.19,5.24)	(2.80,3.65)
12	6.0	(6.00,6.00)	(5.94,5.97)	(4.45,5.11)

Table 8. Confidence Intervals for Revenue Sensitivity with Indirect Estimation

Class	r_k	Light	Moderate	Heavy
1	1.0	(0.95,1.06)	(0.88,0.99)	(−0.20,0.38)
2	1.2	(1.20,1.31)	(1.13,1.24)	(0.64,0.84)
3	1.4	(1.34,1.45)	(1.27,1.39)	(0.85,1.05)
4	1.6	(1.53,1.64)	(1.50,1.61)	(1.02,1.23)
5	1.8	(1.72,1.84)	(1.69,1.81)	(1.28,1.48)
6	2.0	(1.99,2.10)	(1.97,2.09)	(1.71,1.93)
7	3.0	(2.71,2.97)	(2.19,2.43)	(−0.58,0.25)
8	3.6	(3.40,3.66)	(3.00,3.24)	(0.54,0.89)
9	4.2	(3.94,4.20)	(3.54,3.79)	(1.32,1.68)
10	4.8	(4.69,4.95)	(4.48,4.73)	(1.96,2.35)
11	5.4	(5.37,5.38)	(5.19,5.24)	(2.80,3.65)
12	6.0	(5.81,6.08)	(5.75,6.02)	(4.27,4.73)

Some recent work on the application of Monte Carlo summation to multichain queueing networks has been carried out by Ross and Wang [18]. A more in-depth study of this subject is currently in progress [17].

Acknowledgment

We would like to thank the referee for the many valuable suggestions.

References

1. Bratley, P., Fox, B.L., & Schrage, L.E. (1987). *A guide to simulation*. New York: Springer-Verlag.
2. Fishman, G.S. (1978). *Principles in discrete event simulation*. New York: John Wiley.
3. Harvey, C. & Hills, C.R. (1979). Determining grades of service in a network. In *9th International Teletraffic Conference*.
4. Ho, Y.C. (1987). Performance evaluation and perturbation analysis of discrete event dynamic system, perspective and open problems. *IEEE Transactions on Automatic Control* 32: 563–572.
5. Hunt, P.J. (1989). Implied costs in loss networks. *Advances in Applied Probability* 21: 661–680.
6. Kalos, M. & Whitlock, P.A. (1986). *Monte Carlo methods*, Vol. 1: *Basics*. New York: John Wiley.
7. Kaufman, J.S. (1981). Blocking in a shared resource environment. *IEEE Transactions on Communications*, COM-29(10): 1474–1481.
8. Kelly, F. (1986). Blocking probabilities in large circuit-switched networks. *Advances in Applied Probability* 18: 473–505.
9. Lavenberg, S.S. & Welch, P.D. (1981). A perspective on the use of control variables to increase the efficiency of Monte Carlo simulation. *Management Science* 27: 322–335.
10. McKenna, J. & Mitra, D. (1982). Integral representations and asymptotic expansions for closed Markovian queueing networks: Normal usage. *Bell Systems Technical Journal* 61: 661–683.

11. McKenna, J., Mitra, D., & Ramakrishnan, K.G. (1981). A class of closed Markovian queueing networks: Integral representations, asymptotic expansions, and generalizations. *Bell Systems Technical Journal* 60: 599–641.
12. Mitra, D. (1987). Asymptotic analysis and computational methods for a class of simple, circuit-switched networks with blocking. *Advances in Applied Probability* 19: 219–239.
13. Mitra, D. & McKenna, J. (1986). Asymptotic expansions for closed Markovian queueing networks with state dependent service rates. *Journal for the Association of Computing Machinery* 33: 568–592.
14. Ramakrishnan, K.G. & Mitra, D. (1982). An overview of PANACEA, a software package for analyzing Markovian queueing networks. *Bell Systems Technical Journal* 61: 568–592.
15. Roberts, J.W. (1981). A service system with heterogeneous user requirements. *Performance of data communications systems and their applications*. Amsterdam: Elsevier (North-Holland), pp. 423–431.
16. Ross, K.W. & Tsang, D. (1990). Teletraffic engineering for product-form circuit-switched networks. *Advances in Applied Probability* 22: 657–675.
17. Ross, K.W. & Wang, J. (1991). Monte Carlo summation applied to multichain queueing networks. Technical Report, University of Pennsylvania, Philadelphia.
18. Ross, K.W. & Wang, J. (1990). Solving product form stochastic networks with Monte Carlo summation. In *Proceedings of the 1990 Winter Simulation Conference*, New Orleans.
19. Tsang, D. & Ross, K.W. (1990). Algorithms for determining exact blocking probabilities in tree networks. *IEEE Transactions on Communications* 38: 1266–1271.