Les Cahiers du GERAD ISSN: 0711–2440

Filtered and Setwise Gibbs Samplers for Teletraffic Analysis

Lachlan Andrew
Felisa J. Vázquez-Abad

G–2003–53

August 2003

Filtered and Setwise Gibbs Samplers for Teletraffic Analysis

Lachlan Andrew

The ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN)
Department of Electrical and Electronic Engineering
The University of Melbourne, Victoria 3010, Australia
l.andrew@ee.mu.oz.au

Felisa J. Vázquez-Abad[*]

GERAD and Department of Computer Science and Operations Research
University of Montreal, Montreal, Canada H3C 3J7
and Dept. Elec. and Elec. Engg., University of Melbourne, Australia
{vazquez@iro.umontreal.ca;fva@ee.mu.oz.au}

August, 2003

Les Cahiers du GERAD

G–2003–53

[*]This work was supported in part by NSERC-Canada grant # WFA0184198 and by the Australian Research Council (ARC).

Abstract

The Gibbs sampler is a very simple yet efficient method for the performance evaluation of product form loss networks. This paper introduces the setwise Gibbs sampler as a flexible technique for analysing closed BCMP networks, which model telecommunication networks using window flow control. The efficiency of another variant, the filtered Gibbs sampler (FGS), is also investigated. It is shown that the FGS is considerably more efficient than the standard Gibbs sampler. It is also shown that traditional estimates of the accuracy of FGS can be excessively optimistic, and a more conservative estimator is presented.

Keywords: Product form; Queueing networks; Gibbs Sampler; Markov chain Monte Carlo.

Résumé

L'échantilloneur de Gibbs est une méthode simple et efficace pour évaluer la performance des réseaux de perte avec des probabilités limites sous la forme de produits. Cet article introduit l'echantilloneur de Gibbs aux ensembles, une technique flexible pour l'analyse des réseaux BCMP, qui modélisent des réseaux de télécommunications avec un contrôle de flôt par fenêtres. L'efficacité d'une variation de l'echantilloneur de Gibbs apellée "échantilloneur de Gibbs filtré" (FGS) est aussi étudiée. Nous montrons que le FGS est remarquablement plus efficace que l'echantilloneur de Gibbs standard. En plus, nous démontrons aussi que les estimations traditionnelles de la précision pour FGS peuvent être excessivement optimiste, et un estimateur plus conservatif est présenté.

1 Introduction

Product form stationary distributions arise in many models for telecommunications systems. Truncated multi-class $M/G/\infty$ queues model traditional circuit switched networks with fixed routing, cellular networks with frequency-reuse constraints [5,12,24], packet networks with fixed routing using effective bandwidth admission control [4,15] or with marking-based admission control [16], or intelligent networks in which connections require a particular set of services for the duration of the call [14]. Closed BCMP networks [3,8] model packet switched networks with sliding window or token-based flow control [25,29]. Many other applications are listed in [22].

The importance of product form networks has led to many techniques for their analysis [26]. Many important performance measures may be calculated from the normalising constant, denoted G in equation (3). These measures include the blocking probability of circuit switched networks, and mean queue lengths and throughputs of packet switched networks. The normalising constant, G, may be calculated by convolutional methods [7,10], numerical inversion of generating functions [9] or by Monte Carlo integration [5,27].

This paper investigates the performance of Markov chain Monte Carlo simulation as an alternative means of estimating blocking probabilities in product form networks [20,30] (see also [21]). In addition to blocking analysis, these algorithms can generate actual samples from the state distribution, which can be used, say, for starting simulations to calculate time-dependent measures, as is done in [11].

The Gibbs sampler traverses the state space by modifying one component of the state vector at a time. As such it is not directly applicable to closed queueing networks, in which the sum of the components is fixed. The traditional solution is to remove one component from the state vector; updates to a single component of the reduced state vector implicitly update the omitted component as well. Section 3 presents a more flexible approach, the setwise Gibbs sampler (SGS) in which arbitrary subsets of components are updated simultaneously. Necessary and sufficient conditions on the choice of subsets are given for the process to converge to the correct distribution.

The filtered Gibbs sampler (FGS) [2, 30] is an alternative, complementary enhancement to the standard Gibbs sampler, for which no thorough performance analysis has yet appeared. It is evaluated numerically in Sections 4.2 and 4.3, and an expression is derived for the maximum benefit relative to the standard Gibbs sampler under the assumption of low network load. Section 4.4 then shows that confidence intervals based on standard estimates of variance can be significantly too small, and proposes a more conservative variance estimator.

2 Network Model

Consider the general BCMP model for a queueing network, introduced in [3]. There are N service stations that may have single or multiple servers (as described below), and R classes of customers (that may possess different service requirements). A customer of class r that ends service at station i is routed to station j and given class s with probability $p_{(i,r),(j,s)}$ independently of the history of the process. Arrivals to service station i of class r customers from outside the network follow independent Poisson processes with a rate that may depend on the current occupancy of the network. The network may be as complicated as having some classes with no external arrivals, so their behaviour is that of a closed network, while other classes sharing the network resources may have external arrivals and departures. The general model therefore considers the possibility that the routing matrix

$$P = \{p_{(i,r),(j,s)}\} \tag{1}$$

is not irreducible, but consists of m irreducible transition kernels. In this paper, transitions will not occur between classes, that is, P consists of R submatrices, each of them irreducible, corresponding to the subspaces per class. This is the model for closed multiple-chain networks. Each subspace $S_k, k = 1, \dots, m$ corresponds to either a closed or an open subsystem. For each subsystem, the *effective arrival rate* is the solution, e, of the linear equations:

$$e_{j,s} = \lambda_{j,s} + \sum_{(i,r)\in S_k} e_{i,r} p_{(i,r),(j,s)}, \tag{2}$$

where $\lambda_{i,r}$ is the external arrival rate. If the subsystem S_k is closed, then $\lambda_{i,r} = 0$ and the above linear system is only defined up to a multiplicative constant. In that case one sets $\sum_{(i,r)\in S_k} e_{i,r} = 1$ and the factors are interpreted as the *relative number of visits to state* (i,r). The complete set of indices is denoted $S = \cup_k S_k$. Service stations can be of different types. Denote by G_i the service distribution of station i. The *occupancy vector* will be denoted by $n = (n_{i,r}; i = 1, \dots, N; r = 1, \dots, R)$ indicating how many customers of class r are in station i. The aggregate occupancy of station i is $n_i = \sum_{r=1}^{R} n_{i,r}$.

Service stations must be of one of the following types:

Type 1: First-come-first-served (FCFS), $G_i \sim \exp(\mu_i(n_i))$ for all customer classes (station may have one or several servers)

Type 2: Processor sharing, $G_{i,r}$ arbitrary, single server

Type 3: Infinite number of parallel servers, $G_{i,r}$ arbitrary

Type 4: Last-come-first-served (LCFS), $G_{i,r}$ arbitrary, single server.

Remark: In the original paper [3], only queues with service times whose distributions have rational Laplace transforms were considered. However, modern treatments, such as [8], prove the result for general distributions, using the continuity arguments of [31].

Denote by $1/\mu_{i,r}$ the mean service time of class r at service station i, and let $\rho_{i,r} = e_{i,r}/\mu_{i,r}$ be the *utilization factor* of the server/class pair (i,r). For single class networks, the second subscript will be dropped for clarity.

Theorem 1 (BCMP) *[3] Let $y_i = (n_{i,1}, \ldots, n_{i,R})$ denote the occupancy vector at station i. Then the stationary distribution of the network occupancy has the product form:*

$$\pi(y_1, \ldots, y_N) = \frac{1}{G}\, d(S) \prod_{i=1}^{N} g_i(y_i), \tag{3}$$

where:

- *if i is of type 1, then $g_i(y_i) = n_i! \left(\frac{1}{\mu_i}\right)^{n_i} \prod_{r=1}^{R} \frac{e_{i,r}^{n_{i,r}}}{n_{i,r}!}$,*
- *if i is of type 2 or 4, then $g_i(y_i) = n_i! \prod_{r=1}^{R} \frac{\rho_{i,r}^{n_{i,r}}}{n_{i,r}!}$,*
- *if i is of type 3, then $g_i(y_i) = \prod_{r=1}^{R} \frac{\rho_{i,r}^{n_{i,r}}}{n_{i,r}!}$,*

$d(S)$ is a function of the external arrival rates such that $d(S) = 1$ when the whole network is closed, and G is the normalising constant*, chosen so as to make $\sum_{n \in \mathcal{S}} \pi(n) = 1$, where $\mathcal{S}$ denotes the state space of the occupancy vector.*

Note that $g_i(y_i)$ can be written as

$$g_i(y_i) = h_i(n_i) \prod_{r=1}^{R} \frac{\rho_{i,r}^{n_{i,r}}}{n_{i,r}!}, \tag{4}$$

where $h_i(n) = 1$ if station i is of type 3 (which we will call IS — infinite server station), and $h_i(n) = n!$ otherwise.

For a single class closed network, a considerable simplification follows: let T_1 be the subset of all stations that are of type 1, 2 or 4, and T_2 be the set of the (remaining) stations which are of type 3, then:

$$\pi(n) = \frac{1}{G} \prod_{i \in T_1} \rho_i^{n_i} \prod_{i \in T_2} \left(\frac{\rho_i^{n_i}}{n_i!}\right). \tag{5}$$

2.1 Circuit switched networks

In circuit switched networks, the N service stations model distinct routes through the network and n_i is the number of calls currently using route i. If the network can support a particular combination of calls, then it can also support any subset of those calls. Thus for any feasible occupation vector $n = (n_1, \ldots, n_N) \in \mathcal{S}$, we have $\{n' : n'_i \leq n_i\} \subseteq \mathcal{S}$, where '$\leq$' is taken componentwise.

The feasible region, $\mathcal{S}$, is often of the form

$$\mathcal{S} = \{n \in \mathbb{N}^N : An \leq \mathbf{C}\} \tag{6}$$

(but [14,17] give exceptions). Here $A = [a_{ji}] \in \{0,1\}^{L\times N}$ (or more generally $\mathbb{N}^{L\times N}$) specifies the number of channels required by route i on link j, and $\mathbf{C} = (C_i) \in \mathbb{N}^L$ is a vector of the number of channels available on each link.

Because the model corresponds to a single class open network of type 3 servers, the form of the marginal densities $g_i(n_i)$ in (3) is

$$g_i(n_i) = \left(\frac{\rho_i^{n_i}}{n_i!}\right).$$

Let B be the network blocking probability. A feasible state, n, is a blocking state for route i if one more call on route i would lead to an infeasible state. The set of blocking states for route i, $i = 1, \ldots, N$, is

$$\mathcal{B}_i = \{n \in \mathcal{S} : \exists j, a_{ji} + (An)_j > C_j\}\,. \tag{7}$$

Let $B_i = \mathsf{P}(n \in \mathcal{B}_i)$ be the blocking probability of route i. Writing $\lambda = \sum_{i=1}^N \lambda_i$ for the total arrival rate gives

$$B = \sum_{i=1}^{N} \left(\frac{\lambda_i}{\lambda}\right) B_i. \tag{8}$$

2.2 Window flow control

Closed BCMP networks in which users cannot change class can model a packet switched communication network with window flow control in the following sense [25]. Each connection on the communication network is a class. Customers in the queueing network can represent either packets in transit in the communication network or acknowledgements in transit. They can also represent packets received but not acknowledged, or packets within the current transmit window which have not yet been transmitted. (With greedy sources and fast receivers, the latter two cases are not encountered.) The number of customers of each class is equal to the size of the window, which is assumed constant. Store-and-forward switches are represented as FCFS nodes, and transmission delays can be modelled by IS nodes with constant service times. The routing of customers through the queueing network is the same as that of packets through the communication network, and in this paper will be assumed to be deterministic.

For these networks,

$$\mathcal{S} = \left\{ n : \sum_{i=1}^{N} n_{i,r} = C_r \text{ for all } i \right\},$$

where C_r is the constant number of customers on route r, which is equal to the window size for the corresponding connection.

Measures of interest in packet networks include overflow probabilities (the probabilities that the buffers exceed a certain threshold), mean queue lengths and throughputs. In general the performance of the network will be of the form

$$B = \sum_{i=1}^{N} w_i \, B_i,$$

for some weight factors w_i and local performance functions $B_i = \mathsf{E}[b_i(n_i)]$. The sample performance b_i is a local function of the occupancy of station i, and the expectation is with respect to π. This is clearly the case for the three criteria mentioned above, with throughputs calculated by applying Little's law to an estimate of the idle time of each queue.

3 Markov Chain Monte Carlo Simulation

Evaluating blocking probabilities using (3) and (8) directly is a difficult numerical problem for realistic sized networks. Moreover, in many cases, it is not sufficient to know the blocking probability, and it is desirable to sample from the distribution itself (see for example [11]). In [30] a WDM network was studied. A typical WDM backbone network may have over $m = 20$ nodes and $C = 32$ or more wavelengths. The simplest approach is to calculate the normalising factor G, where the sums are over the space $\mathcal{S}$, and then explicitly sum (3) over all states $n \in \mathcal{B}_i$. The number of routes is $R = m^2/2 + o(m^2)$, and for densely connected networks, the number of states is $\mathcal{O}(C^R)$. Thus computing G directly takes of the order of $C^{m^2/2}$ multiplications. For a modest network of $m = 10$ nodes with $C = 8$ wavelengths, this requires around $8^{45} \approx 10^{40}$ multiplications, taking 10^{21} years on a 1 Tflops computer.

Monte Carlo techniques, such as the FGS, bridge the gap between exact algorithms [7,9,10] and approximations [18,22,23]. They allow a quantifiable tradeoff between computational time and accuracy, while being conceptually simple.

This section presents the construction of a "surrogate" Markov chain $\{X_k : k = 1, 2, \ldots\}$ with state space $\mathcal{S}$ whose steady state probabilities are given exactly by π in (3). That is,

$$\forall \; n \in \mathcal{S} \quad \lim_{k\to\infty} \mathsf{P}(X_k = n) = \pi(n). \tag{9}$$

Such methods are called Markov chain Monte Carlo (MCMC) methods (see [6]). Then B can be estimated from S samples as $\hat{Y}(S) = (1/S)\sum_{i=1}^{S} y(x_i)$ for any function $y(\cdot)$ with $\mathsf{E}[y(X)] = B$.

A fixed relative square error, measured by the quantity $\mathsf{Var}[\hat{Y}(S)]/B^2$, can be obtained faster by either decreasing the CPU time required to evaluate $y(X)$ or by using an estimator of B with reduced variance. This tradeoff is quantified by the *relative efficiency* defined by

$$\mathcal{E}_r(\hat{Y}) = \lim_{S\to\infty} \frac{B^2}{\text{CPU}[\hat{Y}(S)]\mathsf{Var}[\hat{Y}(S)]},$$

where $\text{CPU}[\hat{Y}(S)]$ denotes the average CPU time of the simulation that produces the S samples.

Note that it is not necessary for the S replications to be independent. However, if there is significant correlation between them, then $\mathsf{Var}[\hat{Y}(S)]$ may be very much larger than $\mathsf{Var}[\hat{Y}(1)]/S$, which would have resulted from independent samples. Thus, in addition to having the desired steady state distribution, a good surrogate process will have a lower correlation between successive states than the simple arrival/departure process. This can reduce the variance of the final estimate of the blocking probability by orders of magnitude.

One good MCMC method is the Gibbs sampler [6,13,28]. After describing the standard Gibbs sampler, this section presents two enhancements: the *setwise* Gibbs sample, which extends the range of networks which can be analysed, and *filtered* Gibbs sampler, which improves the efficiency of the estimator.

3.1 The Standard Gibbs Sampler

The Gibbs sampler applies to multi-dimensional state spaces. The key principle is that each transition in the surrogate Markov chain updates only one component, but the transition probabilities are proportional to the (known) stationary conditional probabilities for that component given the current values of all other components. This is clearly ideally suited to product form distributions, where these conditional probabilities have a very simple form. It is the ability to make large changes to each component, reducing the correlation between samples generated by a Gibbs sampler, which leads to greater efficiency than direct simulation of the arrival and departure of calls.

In the following, the algorithms for generating state X_{k+1} from X_k require the following notation. For $X \in \mathbb{N}^N$, define:

$$X^{-j} = (X(1), \dots, X(j-1), X(j+1), \dots, X(N)),$$

which is a vector in $\mathbb{N}^{N-1}$, missing component j. Given any $x \in \mathcal{S}$ and an index $1 \le j \le N$, the notation $\pi(\cdot|x^{-j})$ is used for the conditional probability of the jth component given all the others:

$$\begin{aligned} \pi(y|x^{-j}) &= \mathsf{P}(X(j) = y | X^{-j} = x^{-j}) \\ &= \frac{\pi(x_j(y))}{\sum_{x(j)=0}^{C_j(x)} \pi(x)}, \end{aligned}$$

where $x_j(y)$ denotes the vector x with the scalar y replacing x_j, and $C_j(x)$ is the state dependent bound such that all states in the sum in the denominator lie in $\mathcal{S}$.

A *Gibbs Update* is a rule for generating X_{k+1} from X_k, of the form:

1. Select a coordinate $\sigma_k \in \{1, \dots, N\}$, independent of X_k.
2. Set $X_{k+1}(\sigma_k) \sim \pi(\cdot|X_{k+1}^{-\sigma_k})$ and leave all other components unchanged.

For example, if σ_k are i.i.d. random variables then $\{X_k\}$ forms a Markov chain, while if $\sigma_k = k (\text{mod } R)$, then $\{(X_k, \sigma_k)\}$ forms a Markov chain, as does every Nth sample, $\{X_{Nk}\}$. The key property of Gibbs updates is that if X_k is distributed according to π (denoted

$X_k \sim \pi$) then $X_{k+1} \sim \pi$. In other words, the target probability is stationary for the Gibbs sampler.

For the model of the circuit-switched network, $\pi(\cdot|X_{k+1}^{-\sigma_k})$ is a one dimensional Poisson distribution truncated by (6). For each $1 \leq j \leq N$, let

$$P_j(m) = \sum_{n=0}^{m} \frac{\rho_j^n}{n!} \quad m = 1, 2, \ldots. \tag{10}$$

Let $Z_i(X) = C_i - \sum_{c \in L_i} a_{ic} X(c)$ be the number of free channels on link i in state X, where $L_i = \{j : a_{ij} \neq 0\}$ is the set of all routes using the ith link. At every step k, let $j = \sigma_k$ and let

$$C_j(X_k) = \min_{i:j \in L_i} (Z_i(X_k)/a_{ij} + X_k(j)) \tag{11}$$

be the maximum allowable number of connections using route j given X_k^{-j}. Then the required conditional probability satisfies $\mathsf{P}(X_{k+1}(j) \leq m) = P_j(m)/P_j(C_j(X_k)), m = 0, \ldots, C_j(X_k)$.

Since, as $k \to \infty$, $X_k \sim \pi$, it is possible to estimate B_i by $(1/S) \sum_{k=1}^{S} \mathbf{1}_{\{X_k \in \mathcal{B}_i\}}$, where $\mathbf{1}_{\{A\}} = 1$ if A is true, 0 otherwise. However since updates to component j only change $\mathbf{1}_{\{X_k \in \mathcal{B}_i\}}$ when i and j share a link, evaluating this sum involves significant unnecessary computation at each step k for all links l that do not share a link with the current updated route. Having evaluated $C_j(X_k)$ and X_{k+1}, it is easy to calculate $\mathbf{1}_{\{X_{k+1} \in \mathcal{B}_j\}} = \mathbf{1}_{\{X_k(j) = C_j(X_k)\}}$ for the component j which is updated at iteration k. Thus B_i can be estimated by

$$Y_i(S) = \frac{1}{S(i)} \sum_{k=1}^{S} y_i(X_k) \mathbf{1}_{\{\sigma_k = i\}} \tag{12}$$

where $y_i(X_k) = \mathbf{1}_{\{X_{k+1} \in \mathcal{B}_i\}}$, and $S(i) = \sum \mathbf{1}_{\{\sigma_k = i\}}$ counts the number of iterations where $\sigma_k = i$. These *local estimates* converge to B_i at rate $\mathcal{O}(S^{-1/2})$ as S increases.

3.2 Setwise Gibbs Sampler

For a closed network, it is impossible to update one coordinate at a time: if only one occupation number, n_σ, is to be updated, the requirement that the number of customers of each class in the network remains constant means that the next state must equal the previous state. The next state still satisfies $X_{k+1} \sim \pi$, since $X_k \sim \pi$, but the process is no longer ergodic. We now present the *setwise Gibbs sampler*, which restores ergodicity.

Let $l = \{(l_1, r), \ldots, (l_j, r)\}$ denote a set of components in S, corresponding to the same class of customer, r. Consider a Gibbs-style update of these j components. In particular, when $j = 2$ the occupancy constraint now becomes simply that the *sum* $A = n_{l_1,r} + n_{l_2,r}$ remain constant. Notice that for any set $l \subset S$ and for $x, y \in \mathcal{S}$, the statement $y_i = x_i$ for all $i \notin l$ implies

$$\frac{\pi(y)}{\pi(x)} = \frac{\pi_l(y_l \,|\, x_i, i \notin l)}{\pi_l(x_l \,|\, x_i, i \notin l)}, \tag{13}$$

where π_l is the conditional distribution defined only on the set of coordinates l.

The general scheme for a *Setwise Gibbs Update* is a set of sets, $\mathcal{L}$, and a rule for generating X_{k+1} from X_k, of the form:

1. Select a coordinate set $l(\sigma_k) \in \mathcal{L}$, independent of X_k.
2. Set $(X_{k+1}(i); i \in l(\sigma_k)) \sim \pi_l(\cdot | X_{k+1}^{-l(\sigma_k)})$ and leave all other components unchanged.

The selection of $l(\sigma_k)$ may be deterministic or random, but each set, l, is assumed to be selected an infinite number of times as $k \to \infty$.

We will now focus only on the case where the sets $l \in \mathcal{L}$ are pairs of coordinates. Rather than prohibiting an update, the occupancy constraint now helps by substituting the generation of a two-dimensional random variable with that of a one-dimensional one. This gives updates of the form $n_{i,r} = M$, $n_{j,r} = A - M$. For a closed class of a BCMP network,

$$\mathsf{P}(M = m) \propto \frac{h_i(n'_i + m)}{m!} \frac{h_j(n'_j + A - m)}{(A - m)!} \left(\frac{\rho_{i,r}}{\rho_{j,r}}\right)^m , \qquad (14)$$

where $n'_i = \sum_{s \neq r} n_{i,s}$. Note in particular that if nodes i and j are both single-class nodes ($n'_i \equiv 0$) of a type other than type 3 (IS), then the distribution of m becomes a truncated geometric distribution. If one of the two nodes is instead of type 3 (IS), then a truncated Poisson distribution results. Both of these special cases allow m to be generated efficiently using pre-computed lookup tables.

These updates are closely related to the true network dynamics; instead of customers moving from one queue to the next one at a time, groups of customers move in batches between queues on their route which need not be consecutive. This extra flexibility reduces the correlation between successive estimates of the quantity of interest (such as queue length or link utilisation), which improves the efficiency of the estimation. Define the surrogate routing matrix, P', of a set-wise Gibbs sampler as $P' = \{p'_{(i,r),(j,s)}\}$, where $p'_{(i,r),(j,s)} = 1$ if $(i, r), (j, s) \in \mathcal{L}$ and zero otherwise.

Theorem 2 *Consider a set-wise Gibbs sampler whose surrogate routing matrix, P', has the same reducibility structure as the true routing matrix, $P = \{p_{(i,r),(j,s)}\}$. Further, let the restriction of P to any class, r, be irreducible. Then the SGS converges to its equilibrium distribution.*

This will be proved with the help of the following lemma.

Lemma 1 *Consider $n \in \mathcal{S}$, and a partition, $\{F, V\}$, of the components of n, with the cardinalities $|F| \geq 0$ and $|V| \geq 2$. Assume further that there is a subset of components $\mathcal{V} \supseteq V$, such that the restriction of the surrogate routing matrix, P', to components $\mathcal{V}$ is irreducible. Then for any component $j \in V$, and target value $t_j \in \{0, \ldots, \bar{C}_F\}$ with $\bar{C}_F = C - \sum_{k \in F} n_k$, it is possible under the randomized setwise Gibbs sampler to attain a state, m, where the occupancy of all components $f \in F$ will be unchanged, $m_f = n_f$, and where component j is the target value $m_j = t_j$.*

Proof: Call the components in V "variable" and those in F "fixed". Consider an arbitrary $i_1 \in V$. If $\{i_1, j\} \subset l \in \mathcal{L}$ and $n_{i_1} + n_j \geq \bar{C}$ then the Gibbs sampler can reach the target state in one step, by choosing this set and changing the occupancy so that station j reaches exactly t_j customers in that route. Next consider the case that no such i_1 exists, but that $n_j > t_j$. Notice that the target occupancy $t_j \leq \bar{C}_F$. We now argue that it is possible to transfer $t_j - n_j$ customers from the variable components into component j in a finite number of steps. Because of the irreducibility hypothesis, all the components in $\mathcal{V}$ communicate, which means that there exists a sequence, (i_k), of components such that:

- the sequence starts with a component $i_1 \in V$
- the segment ends with component j
- all components in V are in the sequence
- consecutive components, i_k, i_{k+1}, satisfy $\{i_k, i_{k+1}\} \subset l \in \mathcal{L}$.

With positive probability, this sequence will be chosen for the Gibbs updates. Again with positive probability, the Gibbs update will transfer all the customers at i_1 (or $t_j - n_j$, if it is less) to i_2 without changing any other components. Customers may similarly be transferred all the way to j, collecting customers from variable nodes along the way, up to a maximum of $t_j - n_j$. Notice that the sequence may contain fixed components, $i_k \in F$. For any subsequence (i_{k-1}, i_k, i_{k+1}) with $i_k \in F$, there is a positive probability that exactly the same number of customers will be transferred into i_k in the first step as is transferred out on the second step. Thus customers can be "tunnelled" through the fixed components. Eventually, $t_j - n_j$ customers in the variable components can, with positive probability, be transferred to component j, with no net change in the fixed components.

Finally, if $n_j < t_j$ but there does not exist an $i_1 \in V$ with $\{i_1, j\} \subset l \in L$, then the reverse sequence can be used to transfer $n_j - t_j$ customers to any one of the other variable components. This establishes the lemma.

□

Proof of theorem 2: Let $P(l) = (p_{m,n}(l))$ be the transition kernel in $\mathcal{S}$ when all, but only, those coordinates in set l are updated according to the Gibbs sampling strategy:

$$p_{m,n}(l) = \pi_l(n_l \,|\, m_i, i \notin l)\mathbf{1}_{\{n_i = m_i; i \notin l\}},$$

It is easily shown [13, sec. 5.15, 16] that the target distribution π is stationary for $P(l)$, that is, $\pi P(l) = \pi$. Indeed, for any $n \in \mathcal{S}$, from (13)

$$\begin{aligned}
\sum_{m \in \mathcal{S}} \pi(m) p_{m,n}(l) &= \sum_{m \in \mathcal{S}} \pi(m) \pi_l(n_l \,|\, m_i, i \notin l)\mathbf{1}_{\{m_i = n_i; i \notin l\}}, \\
&= \sum_{m \in \mathcal{S}} \pi(m) \left(\frac{\pi(n)}{\pi(m)}\right) \pi_l(m_l \,|\, x_i, i \notin l)\mathbf{1}_{\{m_i = n_i; i \notin l\}} \\
&= \pi(n) \sum_{m \in \mathcal{S}} \pi_l(m_l \,|\, m_i, i \notin l)\mathbf{1}_{\{m_i = n_i; i \notin l\}},
\end{aligned}$$

For any $n \in \mathcal{S}$, $\sum_{m\in\mathcal{S}} \pi_l(m_l \,|\, m_i, i \notin l)\mathbf{1}_{\{m_i=n_i; i\notin l\}} = 1$ because the conditional probability satisfies the law of total probability on the set of coordinates l. So $\pi(n) = \sum_{m\in\mathcal{S}} \pi(m)p_{m,n}(l)$, as required.

Because π is stationary under $P(l)$ for all l, it suffices now to ensure that the successive iterations of a Gibbs sampler will produce an *ergodic* chain, that is, one for which all states are reachable, so that the limit distribution will be the target one: $\lim_{k\to\infty} \mathsf{P}(X_k = n) = \pi(n)$ for all $n \in \mathcal{S}$. For this, it is sufficient to show that, from any state, n, which is recurrent under true process, any state, m, reachable in one step under true process is reachable under the SGS.

The (two) components in which m and n differ must be in the same irreducible block of the routing matrix, P. Without loss of generality, label the components in the maximal such irreducible block as $1, \ldots, N_r$. By hypothesis, the restriction of P' to components $1, \ldots, N_r$ is also irreducible.

We show now that starting from n there is a path of the randomized setwise Gibbs sampler that has positive probability and reaches m in finite time. That is, we will show how to perform a series of positive probability Gibbs updates that will change the occupancy from n_i to m_i for all $i = 1, \ldots, N_r$.

First, from Lemma 1 it is always possible to construct an update with positive probability that reaches a state with the target occupancy for the last station m_{N_r}, by changing one or several of the other occupancies. Next, reach a state where this occupancy remains constant and station $N_r - 1$ reaches the target value m_{N_r-1}. By continuing in this fashion it is straightforward that m is reachable from n using the setwise Gibbs sampler, by Lemma 1. Thus reachability of the whole state space follows from the randomisation in the updates: the Gibbs sampler will choose the next route to update at random, and then chooses one set $l \in L_r$ for the update, also at random. □

Corollary 1 *Lemma 1 and Theorem 2 also hold for SGS with sequential updates.*

Proof: With non-zero probability, intervening updates have no impact. □

As with the analysis of the circuit switched model, it is possible to estimate B_i consistently using the fact that the chain satisfies (9), so that

$$B_i = \lim_{S\to\infty} \frac{1}{S} \sum_{k=1}^{S} b_i[X_k(i)].$$

Clearly, for the mean queue length where $b_i(n_i) = n_i$, if coordinate i is not updated at iteration k then $X_k(i) = X_{k+1}(i)$ and it contributes nothing to the estimate to add this sample: on the contrary, it increases computational effort. Use instead the localised estimation:

$$Y_i(S) = \frac{1}{S} \sum_{k=1}^{S} b_i[X_k(i)]\mathbf{1}_{\{i\in l(\sigma_k)\}}.$$

Note that a single update yields estimators for all elements of $l(\sigma_k)$.

If the variance of the occupancy of either component (i, r) or (j, r) is very small, then the state will usually not change significantly when $l(\sigma_k) = \{(i, r), (j, r)\}$. In BCMP networks, this typically occurs when the expected occupancy is low. Since the resulting correlation in performance estimates will reduce the efficiency, it is advisable to group components together with others of similar expected occupancy.

Each component can be grouped with arbitrarily many other components. In the extreme case, one component could be selected and grouped with every other component of the same class, giving updates a star topology. This can be viewed as converting the closed network into an open network with one fewer dimension, and then using the standard Gibbs sampler, as mentioned in Section 1. However, this only provides one estimate per update, unlike SGS. More importantly, if the selected component has very little variance, then estimates can be very highly correlated, as noted in the previous paragraph. Finally, this may break the symmetry between different components, which would increase the implementation effort required.

3.3 Performance of SGS

The setwise Gibbs sampler will be demonstrated by investigating the impact of delay on the utilisation of a window flow control network. To understand this model, consider the case illustrated in Figure 1 to the left, where two possible connections are depicted, one sending packets from node 1 to node 3 via node 2, and the other one only from node 1 to 2. Consider, for example, the class representing the first connection (going to node 3). Packets are processed at node 1 at a certain service rate called the "transmission rate", after which they are send to node 2 along the link, which takes a fixed amount of time known as the "propagation delay" δ. Next, they receive service at node 2 and are routed towards node 3, where they arrive after δ units of time. Once serviced at node 3 they are released, and an acknowledgement is sent back along the same route in opposite direction, following transmission and propagation delays until they arrive back at the originating node of the connection (this path is shown in dashed arrows in the figure to the left).

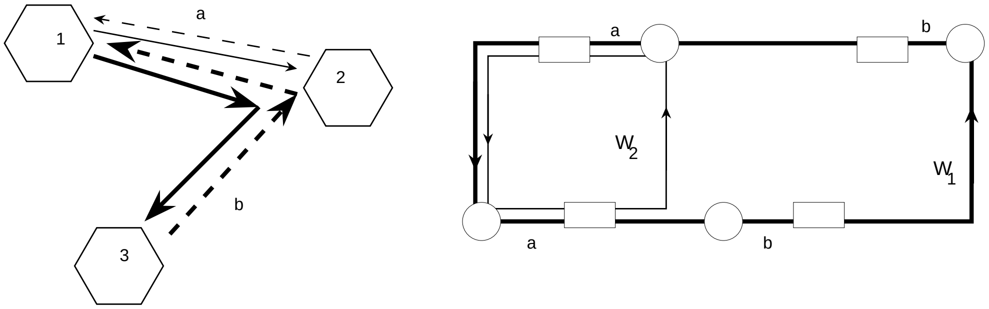


Figure 1: Two connections for three store-and-forward nodes. Left: the system. Right: the closed network model.

A simplified model of the TCP protocol for flow control establishes a window size W_i, $i = 1, 2$ for each connection, and works as follows. Packets at node 1 are sent while there are less than W_1 packets sent for which no acknowledgment has yet been received. As soon as there are W_1 unacknowledged packets the source is stopped until an acknowledgement is received. While the connection is active, there are always packets to be sent, and it is straigtforward to see that after an initial transient, the number of packets plus the number of acknowledgements within the system at any given time is always exactly W_1. This is why a connection can be modelled as a closed sub-network. The closed network model associated with the example is depicted to the right of Figure 1. Each link, i, in the model corresponds to either a store and forward node (FCFS) with iid service times following distribution G_i, or a propagation delay (IS) where the server is of type 3 with deterministic service times δ. The two classes of customers in the network correspond to the two connections of the example.

The data rate of any given connection is the inverse of the maximal service rate along the (closed) path, which corresponds to the maximum transmission rate. The window size for a connection is set at four times the number of hops in the path. Using a fixed window size, W_i, models a situation where the window uses the maximum buffer space allowed by the receiver, and cannot increase as the propagation delay grows.

The standard ARPA2 topology, with 21 nodes and 26 links, was used in our experiments. For this network the transmission rate of all FCFS stations is assumed constant and it is expressed in units of $1/\delta$. Studying the impact of delay in network utilization is equivalent to studying the proportion of idle time as a function of the transmission rate (in units of reciprocal propagation time). Application of the SGS matches similar queues in the pairs $l \in \mathcal{L}$, that is, consecutive updates consider pairs of IS-IS or FCFS-FCFS queues in the network. For each path, at one IS-FCFS pair and one FCFS-IS are also included in $\mathcal{L}$ to ensure the required irreducibility of the surrogate routing matrix, P'. Because these network consists of a mixture of FCFS and IS nodes, these results, shown in Figure 2, could not be generated by, for example, Buzen's algorithm [7].

4 Filtered Gibbs Sampler

Consider a Markov chain $\{X_k\}$ and an estimator

$$\bar{B}_S = \frac{1}{S}\sum_{k=1}^{S} b(X_k),$$

for a sample performance b. The method of *filtered Monte Carlo* is based on conditioning at each stage [28]:

$$\bar{B}'_S = \frac{1}{S}\sum_{k=1}^{S} \mathsf{E}[b(X_{k+1})|X_k].$$

This is closely related to "inverse convolution" [19].

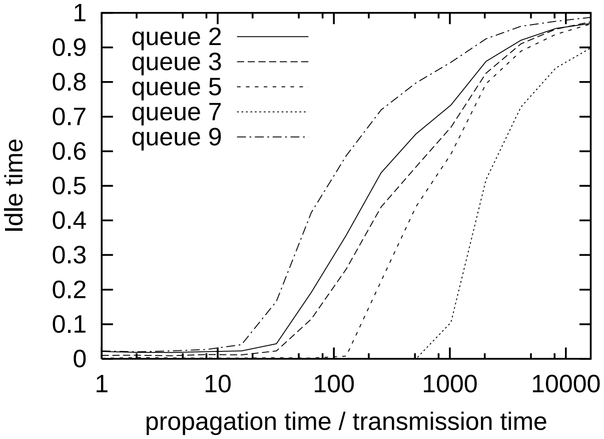


Figure 2: Fraction of time queues are idle in ARPA2 network as a function of data rate, expressed as mean propagation time normalised by transmission time.

The *Filtered Gibbs Sampler* (FGS) combines the filtering with the distribution of the estimation via the local estimates as follows.

Consider the chain $\{X_k\}$ with Gibbs updates using the set of components l_k, with a sequential assignment of period p that updates every coordinate at least once in p iterations. The FGS estimator is of the form:

$$\hat{Y}(S) = \frac{1}{S} \sum_{k=1}^{S} \left(\frac{\nu(i)}{p} \right) y_{\sigma_k,F}(X_k)\, \mathbf{1}_{\{i \in l_k\}}, \tag{15}$$

where $y_{i,F}(x) = \mathsf{E}(b_i[X_{k+1}(i)] \mid X_k, l_k)$ and $\nu(i) > 0$ is the number of times that coordinate i is updated in one cycle. Each of the periodic Gibbs samplers embedded in the computation of (15) is dedicated to estimating $B_i \mathbf{1}_{\{i \in l_k\}}$. Since $S(i)/S \to \nu(i)/p$ as $S \to \infty$, it follows that under the FGS, $\hat{Y}(S) \to B$ [30].

Applying this method to the circuit switched network requires evaluation of the conditional probabilities:

$$\begin{aligned} \mathsf{P}(X_{k+1} \in \mathcal{B}_j | X_k) &= \frac{P_j(C_j(X_k)) - P_j(C_j(X_k) - 1))}{P_j(C_j(X_k))} \\ &\equiv g(C_j(X_k); \rho_j) \end{aligned} \tag{16}$$

where $P_j(\cdot)$ are given in (10) and $C_j(X_k)$ is given in (11). When it is feasible to precompute $g(\cdot;\cdot)$, calculation of the probabilities is as simple as reading a table. This is the case when there is a small number of distinct loads, ρ_j, in the network.

$$\hat{Y}(S) = \frac{R}{S} \sum_{k=1}^{S} \left(\frac{\lambda_{\sigma_k}}{\lambda} \right) y_{\sigma_k,F}(X_k), \tag{17}$$

where $y_{i,F}(x) = g(C_i(x); \rho_i) = \mathsf{P}(X_{k+1} \in \mathcal{B}_i | X_k = x)$.

Note that this is not restricted to estimating blocking probabilities. With a suitable choice of function g, other performance statistics may be estimated, such as mean queue size.

Unlike most exact techniques whose complexity is $\mathcal{O}(C)$, the complexity per iteration of the FGS is $\mathcal{O}(1)$ as the capacity per link increases, assuming the time to generate a single random number is independent of C. However, its primary strength is that it is $\mathcal{O}(R \max_i |L_i|)$ as the number of nodes and links increases. The complexity of all known exact methods is exponential in the number of links.

4.1 Test networks

The FGS was tested on the following network topologies:

(a) Mesh-torus: a rectangular grid with each node connected to four neighbours, wrapping at the edges. Components of the state vector n are the numbers of current calls on a route. In the experiments, the load on all routes was equal. Static shortest path routing ensured a constant number of routes used each link.

(b) Cellular: Spatial reuse constraints in cellular networks with dynamic channel assignment produce “cliques” of cells with a maximum aggregate number of calls [12]. These cliques are analogous to links, while cells correspond to routes. The networks considered here employ a hexagonal grid of cells, and cliques consist of groups of three mutually adjacent cells.

4.2 Correlation

For a single random variable, $\mathsf{Var}[Y] = \mathsf{Var}[\mathsf{E}[Y|Z]] + \mathsf{E}[\mathsf{Var}[Y|Z]]$, and conditioning always entails a variance reduction. However, it is not always the case for Markov chains that $\mathsf{Var}[\hat{Y}'_S] \leq \mathsf{Var}[\hat{Y}_S]$, due to the correlation structure [28]. Explicitly,

$$\begin{aligned} \mathsf{Var}[\hat{Y}_i(S)] &= \frac{1}{S}\mathsf{Var}[y_i(X_1)] \\ &\quad + \frac{2}{S^2}\sum_{j=1}^{S-1}\sum_{k=1}^{S-k}\mathsf{Cov}[y_i(X_j), y_i(X_{j+k})], \end{aligned}$$

and an increase in the second term may exceed the decrease in the first term.

The variance $\mathsf{Var}[\hat{Y}_i(S)]$ can be estimated using batch means (grouping runs of K samples to obtain approximately independent estimates [1]). The impact of the correlation can be quantified by the ratio of $\mathsf{Var}[\hat{Y}_i(S)]$ to $\mathsf{Var}[y_i(X_1)]/S$, the variance estimated by treating individual samples as independent.

Figure 3 shows the results of using batches of size $K = 3 \times 10^6$ (10000 for each of the 300 routes) in a 5×5 mesh-torus, for both the FGS and the standard Gibbs sampler. (Note that these only show the impact of correlation, and do not compare the actual variances of FGS and the standard Gibbs sampler.) These results show that the covariance term

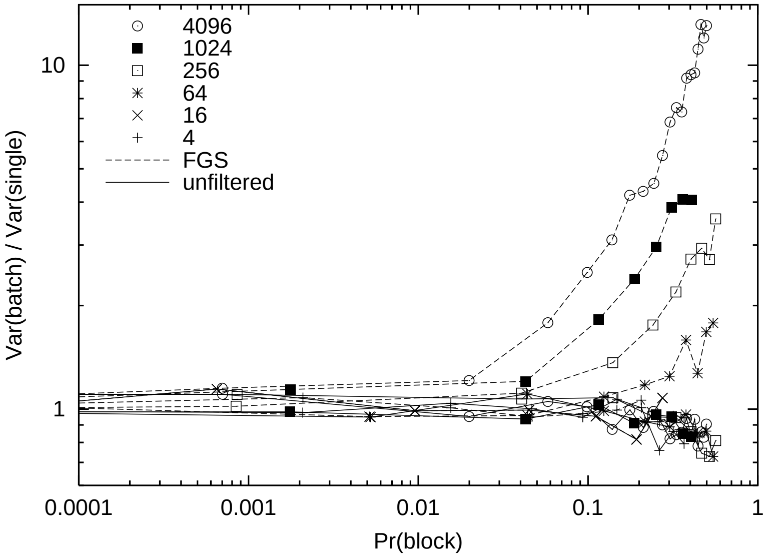


Figure 3: Ratio of variance of a 4-hop route estimated by batch means, $K = 3 \times 10^6$, divided by $\mathsf{Var}[y_i(X_1)]/S$. 5×5 mesh-torus network, 4 to 4096 channels per link.

has minimal impact except when blocking is very high. This justifies ignoring its effect in arguing that filtering should reduce the variance of the estimated blocking probability. However, when blocking is high, the variance of the final blocking estimator using FGS is up to an order of magnitude higher than would be predicted by treating samples as independent. Since this does not occur without filtering, the benefit due to filtering would be overestimated in the case of high blocking if batch means were not used. This effect is greatest for networks with many channels per link, as they have a higher occupancy per channel for a given blocking probability, due to increased trunking efficiency.

Figure 3 suggests that, for high blocking, the true variance of the standard Gibbs sampler is actually less than would be predicted by treating samples as independent. This indicates a negative correlation between samples, but the reason for this is unclear.

4.3 Improvement due to filtering

Numerical results show that filtering causes negligible increase in efficiency for closed BCMP networks. However the gains can be quite significant for $M/G/\infty$ networks. This will now be quantified.

Consider a single link of C channels, used by N routes of load ρ each, and assume that the blocking probability, B, is low. As was demonstrated in Section 4.2, for low blocking the variance of FGS is dominated by the variance of each update, rather than the covariance introduced by the Markov structure. Let

$$D_A = \sum_{j=0}^{C} A^j/j!$$

and note that for $B << 1$ (small A or large C), $D_A \approx e^A$. Denote the Erlang loss function by

$$E_k(\rho) = \frac{\rho^k/k!}{\sum_{j=0}^{k} \rho^j/j!}.$$

The blocking probability of the link is $B = E_C(N\rho)$, and the variance of the Gibbs sampler estimator is $B - B^2$.

For low B, the occupancy of the N routes is well approximated by independent Poisson variables. Each FGS update will see the link filled with the aggregate of the $N-1$ other routes, which is Poisson with rate $(N-1)\rho$. Thus with probability

$$\frac{((N-1)\rho)^{C-j}/(C-j)!}{D_{(N-1)\rho}},$$

the FGS estimate is $E_j(\rho)$. Thus

$$\mathsf{Var}[FGS] + B^2 \quad = \quad \sum_{j=0}^{C} \frac{((N-1)\rho)^{C-j}/(C-j)!}{D_{(N-1)\rho}} \left(\frac{\rho^j/j!}{\sum_{k=0}^{j} \rho^k/k!} \right)^2,$$

and

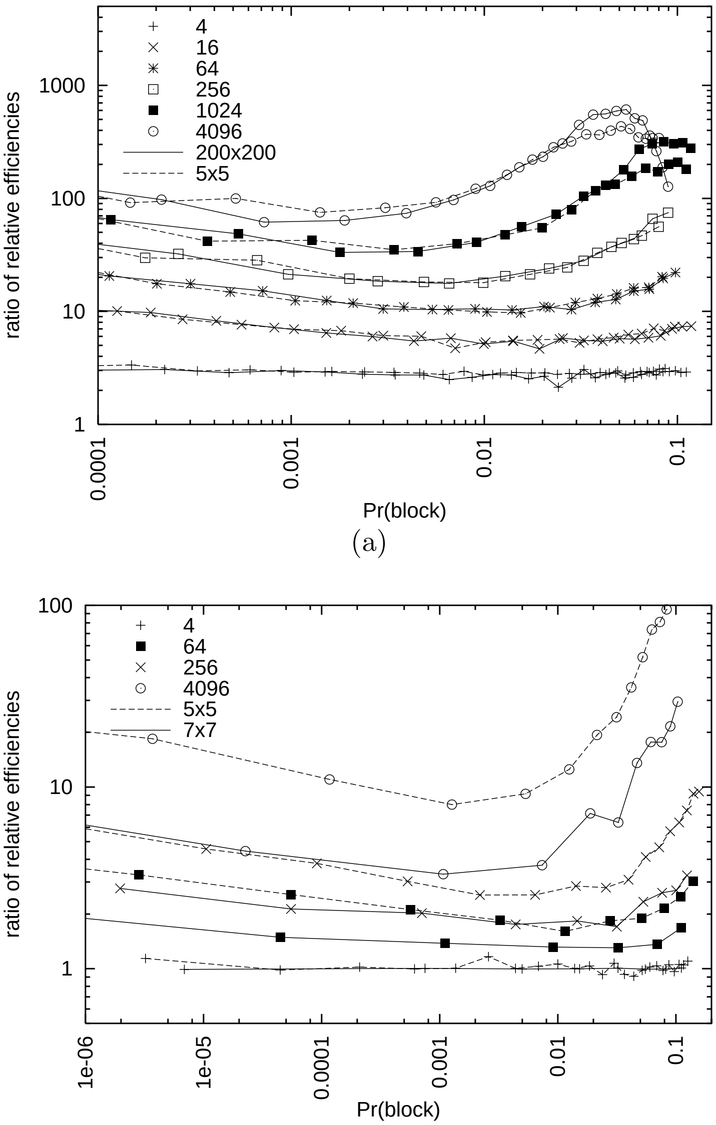
$$\frac{\mathsf{Var}[FGS] + B^2}{\mathsf{Var}[GS] + B^2} = \frac{D_{N\rho}}{D_{(N-1)\rho}} \sum_{j=0}^{C} \frac{C! \quad (N-1)^{C-j}}{(C-j)!j! \quad N^C} \frac{\rho^j/j!}{(\sum_{k=0}^{j} \rho^k/k!)^2}$$

$$= e^{\rho} \left(\frac{N-1}{N} \right)^C \sum_{j=0}^{C} \binom{C}{j} \left(\frac{1}{N-1} \right)^j \frac{E_j(\rho)}{\sum\limits_{k=0}^{j} \rho^k/k!} \tag{18}$$

$$\to \left(\frac{N-1}{N} \right)^C \text{ as } \rho \to 0. \tag{19}$$

This analysis extends easily to unequal loads.

Figure 4 shows the increase in the relative efficiency of the FGS compared to a standard Gibbs sampler for 5×5 and 200×200 cellular networks ($N = 3$) and 5×5 and 7×7 mesh-torus networks ($N = 15, 42$). The results are very similar for both cellular networks, while the results differ for the two mesh-torus networks. This is because cellular networks have $N = 3$ cells per clique, while the values of N differ greatly for the mesh-tori.



(a)

(b)

Figure 4: Ratio of efficiency of filtered to standard Gibbs sampler with 4 to 4096 channels per link for (a) 5 × 5 and 200 × 200 cellular (b) 5 × 5 and 7 × 7 mesh-torus networks.

As indicated by (18), the gain in relative efficiency due to conditioning increases as the capacity of the of the links increases. It is a minimum in the range of blocking probabilities which are of greatest interest, around 10^{-2} to 10^{-3}. However, even in this range the gains are substantial for networks with many channels per link.

4.4 Confidence Intervals

In contrast to the Bernoulli outcomes of standard Monte Carlo, FGS produces samples from an unknown and highly skewed distribution. This makes it possible to underestimate the variance of the estimator by orders of magnitude if insufficient samples are taken. Figure 5 shows the estimate of blocking after each iteration, and also the value ("traditional upper") which is usually used as the upper limit of a confidence interval, i.e., the estimated mean plus twice the estimated standard deviation. After a small number of samples, this "2σ" upper limit is below the true value for much more than 2.5% of the time (which it would be in the Gaussian case), and is ineffective as a confidence bound.

To see why this occurs, consider the terms in

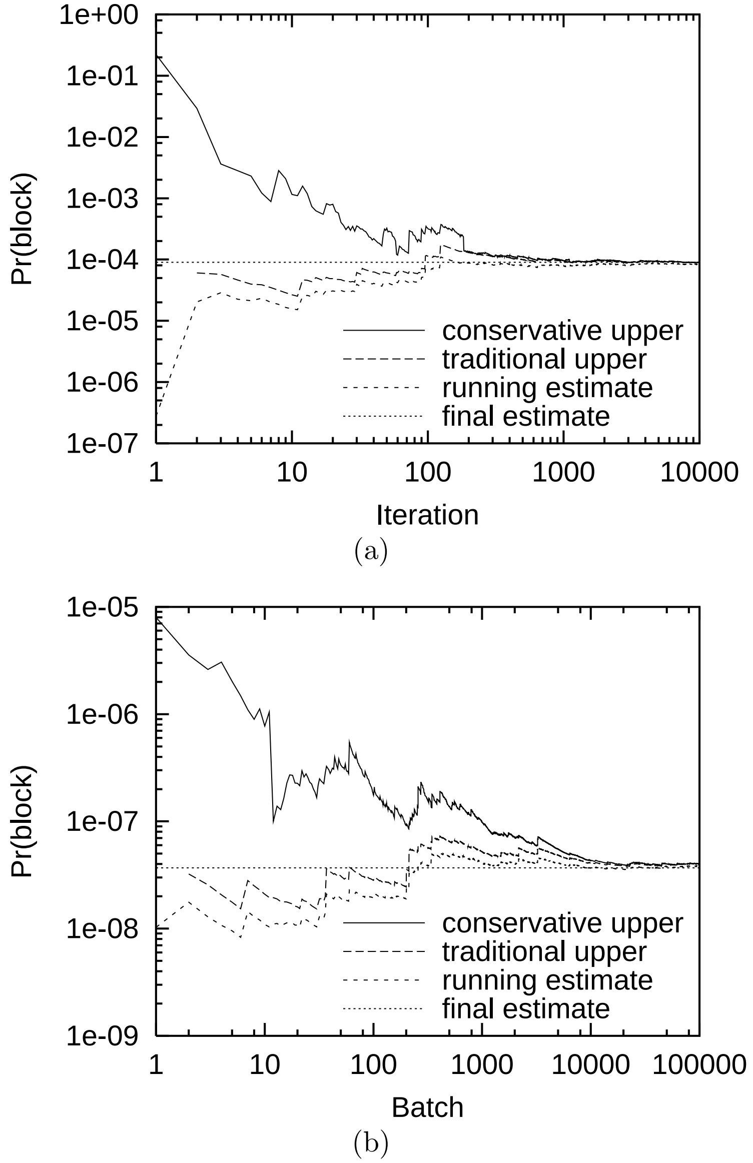
$$B_i = \sum_{j=0}^{C} g(j;\rho_i)\mathsf{P}(C_i(X) = j), \tag{20}$$

with $g(j;\rho_i)$ defined by (16). Without filtering, $y_j(X) = 1$ if $C_i(X) = x_i$ and 0 otherwise. If at least one non-zero sample is generated then the variance estimator will, with high probability, be of the correct order of magnitude. If all samples are 0, it is clear that the sample variance (zero) is not a true indication of the error. However, this is not the case for highly skewed continuous distributions. There are many non-zero terms in (20) which have a high probability, but make very little contribution to the sum due to small values of $g(j;\rho_i)$. Thus if the sample size is too small, the sample mean and variance can be very much smaller than the ensemble values, without any tell-tale zeros to indicate their unreliability. For the FGS to be of practical value, it is necessary to be able to detect when an estimate is statistically unreliable.

For a better indication of the accuracy of the result, consider the individual terms ("partial expectations") of (20). Figure 6 plots these terms against the cumulative probability for a 37-cell cellular network with 64 channels and 12 Erlangs per cell. (As $\mathsf{P}(C_i < j)$ is monotonic in j, the horizontal axis is simply a non-linear scale for j.)

Since $g(j;\rho_i)$ is known, it suffices to estimate $\mathsf{P}(C_i(X) = j)$, or those for which $g(j;\rho_i)\mathsf{P}(C_i(X) = j)$ is a significant fraction of B. Because these terms decay rapidly for $j < \operatorname{argmax}_j(g(j;\rho_i)\mathsf{P}(j))$, as seen in Figure 6, it is possible to determine by inspection when all "significant" terms have been estimated with sufficient confidence.

To quantify this, assume that the sample contains enough points to capture the peak of the probability distribution, which requires orders of magnitude less data than capturing the peak of the partial expectation. (Note the different scales in Figure 6.) Let m be the smallest value such that $\mathsf{P}(C_i = m)$ can be reliably estimated from the sample, and for



(a)

(b)

Figure 5: Estimated upper bounds on B: $\hat{B} + 2\sigma$ and the conservative estimator of (21) for a 3×3 mesh with 64 channels per link. (a) 13 Erlangs per route, simple variance (b) 10 Erlangs per route, batches of 100

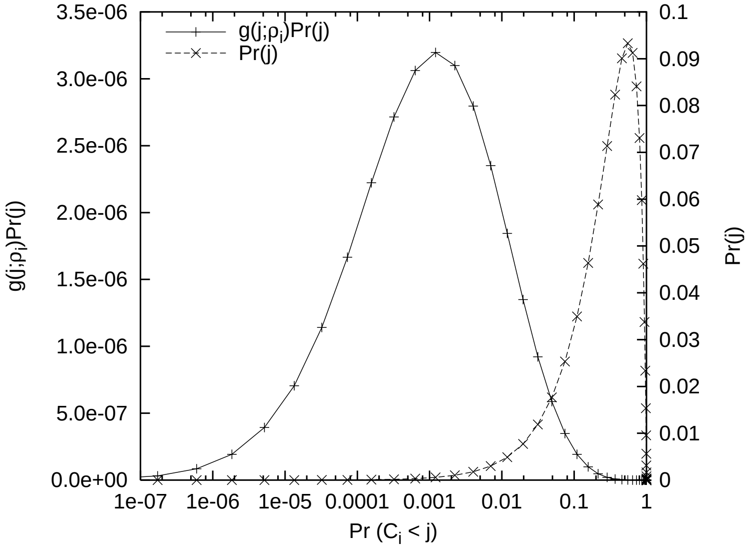


Figure 6: Comparison of partial expectations and state probabilities.

$j \geq m$, let p_j be the sample estimate of $\mathsf{P}(C_i = j)$. For $j < m$, conservatively approximate the tail as $\mathsf{P}(C_i = j) \approx p_j \equiv p_m \Delta^{j-m}$, where Δ is fitted to the sample data. In this paper,

$$\Delta = \sqrt[h]{\frac{\sum_{j=0}^{h-1} p_{m+j}}{\sum_{j=0}^{h-1} p_{m+h+j}}},$$

where m is the smallest value of $C_i(X)$ observed more than once in the simulation, and h is such that $m + 2h$ is the fourth smallest such value.

Ignoring correlations (Section 4.2), the variances of the estimates p_j based on S samples, and $\mathsf{Var}[\hat{Y}_i(S)]$ can then be approximated by

$$\begin{aligned} \hat{V}_i(j) &= p_j(1 - p_j)/S, \\ \hat{V}_i &= \sum_{j=0}^{C} (g(j;\rho_i))^2 \hat{V}_i(j). \end{aligned} \tag{21}$$

The curve "conservative" in Figure 5 plots $\hat{B} + 2\sqrt{\hat{V}}$. It is clearly overly conservative for very small sample sizes, since p_j, $j < m$, are very conservative. However, if the sample is large enough for B to be suitably accurate, then the bound becomes usably tight.

5 Concluding Remarks

The Gibbs sampler has been extended to the broad class of BCMP queueing networks, including both closed queueing networks and truncated $M/G/\infty$ networks as important special cases. For $M/G/\infty$ networks, the filtered Gibbs sampler not only outperforms the usual Gibbs sampler, but its relative efficiency actually grows with problem size and with increasing load.

The key limitations of the FGS are its relatively poor performance when the load per channel is low, which is typically the case in models of networks using window flow control.

References

[1] C. Alexopoulos and A. Seila, *Output data analysis*, in: J. Banks, editor, *Handbook of Simulation*, John Wiley and Sons, New York, NY, 1998 pp. 225–272.

[2] L. L. H. Andrew and F. J. Vázquez-Abad, *Filtered Gibbs sampler for estimating blocking in product form networks*, in: *Proc. IASTED Wireless and Optical Communications*, Banff, Canada, July, 2002, pp. 527–532.

[3] F. Baskett, M. Chandy, R. Muntz and J. Palacios, *Open, closed, and mixed networks of queues with different classes of customers*, J. ACM 22 (1975), 248–260.

[4] A. W. Berger and W. Whitt, *Effective bandwidths with priorities*, IEEE/ACM Trans. Networking 6 (1998), 447–460.

[5] R. J. Boucherie and M. Mandjes, *Estimation of performance measures for product form cellular mobile communications networks*, Telecommunication Systems 10 (1998), 321–354.

[6] P. Brémaud, "Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues," Texts in Applied Mathematics, 31, Springer, New York, NY, 1999.

[7] J. P. Buzen, *Computational algorithms for closed queueing networks with exponential servers*, Comm. ACM 16 (1973), 527–531.

[8] X. Chao, M. Mayazawa and M. Pinedo, "Queueing Networks: Customers, Signals and Product Form Solutions," John Wiley and Sons, Chichester, UK, 1999.

[9] G. L. Choudhury, K. K. Leung and W. Whitt, *An inversion algorithm to compute blocking probabilities in loss networks with state-dependent rates*, IEEE/ACM Trans. Networking 3 (1995), 585–601.

[10] J. L. Coleman, W. Henderson and P. G. Taylor, *A convolution algorithm for calculating exact equilibrium distributions in resource allocation problems with moderate user interference*, IEEE Trans. Commun. 42 (1994), 1106–1111.

[11] A. E. Conway and D. E. O'Brien, *Hybrid analysis of response time distributions in queueing networks*, IEEE Trans. Commun. 41 (1993), 1091–1101.

[12] D. Everitt and N. W. Macfadyen, *Analysis of multicellular mobile radiotelephone systems with loss*, Br. Telecom Technol. J. 1 (1983), 37–45.

[13] G. S. Fishman, "Monte Carlo: Concepts, Algorithms, and Applications," Springer-Verlag, New York, NY, 1996.

[14] S. Jordan, *A continuous state space model of multiple service, multiple resource communication networks*, IEEE Trans. Commun. 43 (1995), 477–484.

[15] F. P. Kelly, *Effective bandwidths at multi-class queues*, Queue. Syst. 9 (1991), 5–16.

[16] F. P. Kelly, P. B. Key and S. Zachary, *Distributed admission control*, IEEE J. Select. Areas Commun. 18 (2000), 2617–2628.

[17] J. Kind, T. Niessen and R. Mathar, *Theory of maximum packing and related channel assignment strategies for cellular radio networks*, Math. Meth. Op. Res. 48 (1998), 1–16.

[18] C. Knessl and C. Tier, *Asymptotic approximations and bottleneck analysis in product form queueing networks with large populations*, Perf. Eval. 33 (1998), 219–248.

[19] P. Lassila and J. Virtamo, *Nearly optimal importance sampling for Monte Carlo simulation of loss systems*, ACM Trans. Model. Comput. Simul. 10 (2000), 326–347.

[20] P. E. Lassila and J. T. Virtamo, *Efficient Monte Carlo simulation of product form systems*, in: *Proc. Nordic Teletraffic Seminar (NTS) 14* (1998), pp. 355–366.

[21] P. E. Lassila and J. T. Virtamo, *Variance reduction in Monte Carlo simulation of product form systems*, Electron. Lett. 34 (1998), 1204–1205.

[22] D. Mitra and J. A. Morrison, *Erlang capacity and uniform approximations for shared unbuffered resources*, IEEE/ACM Trans. Networking 2 (1994), 558–570.

[23] D. Mitra, J. A. Morrison and K. G. Ramakrishnan, *Optimization and design of network routing using refined asymptotic approximations*, Perf. Eval. 36-37 (1999), 267–288.

[24] D. L. Pallant and P. G. Taylor, *Modeling handovers in cellular mobile networks with dynamic channel allocation*, Operations Research 43 (1995), 33–42.

[25] M. Reiser, *A queueing network analysis of computer communication networks with window flow control*, IEEE Trans. Commun. 27 (1979), 1199–1209.

[26] K. W. Ross, "Multiservice loss models for broadband telecommunication networks," Springer-Verlag, Berlin, Germany, 1995.

[27] K. W. Ross, D. H. K. Tsang and J. Wang, *Monte Carlo summation and integration applied to multiclass queuing networks*, J. ACM 41 (1994), 1110–1135.

[28] S. M. Ross, "Simulation," Academic Press, Boston, MA, 1997, 2nd edition.

[29] F. Vázquez-Abad and L. G. Mason, *Decentralized adaptive isarithmic flow control for high speed data networks*, Operations Research 47 (1999), 928–942.

[30] F. J. Vázquez-Abad and L. Andrew, *Filtered Gibbs sampler for estimating blocking probabilities in WDM optical networks*, in: D. Landeghem, editor, *Proc. 14th European Simulation Multiconference* (May, 2000), pp. 548–555.

[31] W. Whitt, *Continuity of generalized semi-markov processes*, Mathematics of Operations Research 5 (1980), 494–501.