

# Filtered Gibbs Sampler for Estimating Blocking Probabilities in WDM Optical Networks

Felisa J. Vázquez-Abad \*

Department of Computer Science and Operations Research  
University of Montreal, Montreal, Canada H3C 3J7  
Email: vazquez@iro.umontreal.ca

Lachlan Andrew †

Department of Electrical and Electronic Engineering  
The University of Melbourne, Victoria 3010, Australia  
Email: l.andrew@ee.mu.oz.au

## ABSTRACT

Blocking probabilities in Wavelength Division Multiplex optical networks are hard to compute for realistic sized systems, even though analytical formulas for the distribution exist (under maximal packing of wavelengths). This computational difficulty is mainly due to the structure of the state space, which imposes strong coupling constraints amongst components of the occupancy vector. Direct simulation is often used to estimate the blocking probabilities. This paper presents a new method based on the Gibbs sampler, for which the estimation is “localised” (distributed computation) and carried out conditioning on an appropriate filtration. The filtered estimators benefit from the variance reduction of conditional expectations as well as a reduction in computational effort due to the local estimation. Our simulation results suggest that that these methods dramatically outperform the currently used computational methods. For some systems, other techniques may prove impossible to implement whilst our method is not subject to the curse of dimensionality.

**Keywords:** Blocking probabilities, dynamically reconfigurable optical networks, performance evaluation of loss networks, Markov chain Monte Carlo, filtered Monte Carlo.

## 1 INTRODUCTION

Wavelength Division Multiplexing (WDM) networks are a promising and practical means of exploiting the enormous capacity of optical fibre communication systems, as discussed by Ramaswami (1993) and Borella et al. (1997). This paper proposes a new simulation technique, the Filtered Sequential Gibbs Sampler, and applies it to the analysis of a loss network model for circuit switched WDM networks, in which connections that are established between two nodes are held for a substantial time. The primary performance measure is the *blocking probability*, the probability that a caller will be unable to place a call from a particular origin to destination because there are insufficient unused wavelengths on the route to carry the call. A maximum blocking probability is often imposed by regulatory authorities or commercial pressure, but over-dimensioning a network is very costly. Thus it is important to know accurately and quickly the blocking probability of a candidate network design.

It is possible to distribute the available wavelengths statically among the origin/destination (O/D) pairs so that no two O/D pairs whose paths share a link can ever use the same

wavelength. Alternatively, wavelengths can be allocated dynamically as calls arrive and depart, as assumed here. A useful lower bound on the overall blocking probability of these networks is obtained using the *maximum packing strategy* of Everitt and Macfayden (1983). The approach, originally proposed for cellular telephony, assumes that a call can be admitted if several linear constraints on the number of calls between each O/D pair are satisfied. An advantage of the maximum packing formulation is that it allows a closed product-form expression for the stationary probabilities, subject to the (difficult) determination of a normalising constant. Everitt (1991) and Everitt (1994) claim that this product form solution applies to other more complex wavelength allocation strategies.

Section 2 summarises the known results for the blocking probability given by the maximum packing bound. Evaluating these blocking probabilities requires evaluation of a normalising constant, and this problem grows exponentially in the number of routes and/or wavelengths (see Nelson, 1993), which has led most researchers to use simulation instead.

Two simulation approaches seem natural: generation of the random variables with the truncated probabilities that appear in the closed form solution, and simulation of the original process. In this paper we deal with the first approach. Direct simulation of the target distribution can be done via a truncation of a multivariate Poisson vector, but as explained in Section 3, this approach may become impractical. Section 4 introduces a Markov chain Monte Carlo (MCMC) method, namely the Gibbs sampler, to solve this problem. To our knowledge, our studies are the first to implement MCMC methods for performance evaluation in loss networks. In Section 4 the estimation of the Gibbs sampler is performed locally (distributed computation) and a filtering technique is used to improve the efficiency. Our proposed method is a filtered Monte Carlo simulation applied to the MCMC with local estimation that benefits from a variance reduction as well as ease of computation. Section 5 presents the numerical results which indicate how our method outperforms the currently used methods for direct simulation.

## 2 NETWORK MODEL

Consider a WDM network with  $\Lambda$  wavelengths, in which each node can transmit and receive on any wavelength. Each wavelength carries only one call on a given link. The *wavelength continuity constraint* requires that at nodes without “wavelength converters”, the same wavelength be used on the incoming link as the outgoing link. This model covers WDM networks with no, partial or full wavelength conversion, and also traditional TDM networks (with time slots replacing wavelengths). Only one route is used per O/D pair (fixed routing).

\*Supported in part by NSERC-Canada grant # WFA0184198 and by the Australian Research Council while on leave at the University of Melbourne.

†Supported by the Australian Research Council (ARC).

## 2.1 Maximum and Clique Packing

Maximum packing admits an incoming call on a given route if a rearrangement of wavelengths of ongoing connections can be found to free a sequence of wavelengths that satisfies the continuity constraint. The call is then connected and held for a random time.

Let  $R$  be the number of routes in the network. Let  $n_i$  denote the number of calls using the  $i$ th route. Define a graph  $\mathcal{G} = (V, E)$  where each vertex represents a route, and there is an edge between two vertices if the corresponding routes share a link. Allocating wavelengths to routes then corresponds to “colouring”  $\mathcal{G}$  so that vertex  $i$  is allocated  $n_i$  colours, and no two adjacent vertices are allocated the same colour. This is called an  $n$ -colouring, where  $n = (n_1, \dots, n_R)$ . For a WDM network under the wavelength continuity constraint, the *maximum packing* strategy will admit a new call if there is an  $n$ -colouring of  $\mathcal{G}$  using at most  $\Lambda$  colours, where  $n$  is the occupancy vector after admission of the call, and  $\Lambda$  is the number of wavelengths per optic fibre. Networks with partial conversion require a reinterpretation of a “route”.

In general, the necessary and sufficient conditions for admission can be quite complex (see Kind, Niessen, and Mathar, 1998), and so a simple set of necessary conditions, known as the *clique packing* conditions, is commonly used instead. To simplify notation, this paper describes the proposed method applied to clique packing, rather than maximal packing.

**Definition 1** A clique of a graph  $\mathcal{G}$  is a subgraph which is completely connected, that is, there is an edge between each pair of vertices. A maximal clique is one which is not a subgraph of any larger clique of  $\mathcal{G}$ .

For any maximal clique  $c$  of  $\mathcal{G}$ , let  $n_c$  denote the total number of calls on the routes in  $c$ . Maximum packing will only admit a call on route  $i$  when  $n_c \leq \Lambda - 1$  for all cliques  $c$  containing route  $i$ . These necessary conditions are the clique packing conditions. Raymond (1991) notes that clique packing is not equivalent to maximum packing. Nonetheless, it results in approximately equal blocking performance.

Let  $\mathcal{S}$  be the set of all permissible network occupancies. For clique packing,  $\mathcal{S}$  is given by

$$\mathcal{S} = \left\{ n \in \mathbb{N}^R : \sum_{j \in C_l} n_j \leq \Lambda, \quad \forall l = 1, \dots, L \right\} \quad (1)$$

where  $C_l$  is the  $l$ th clique and  $L$  is the number of cliques in the network. In general,  $\mathcal{S}$  is such that, if  $x_i$  are specified for all  $i \neq j$ , then there is maximum occupancy  $\Lambda_j(x)$  for route  $j$ . That is, an occupancy vector  $n = (x_1, \dots, x_{j-1}, n_j, x_{j+1}, \dots, x_R)$  is in  $\mathcal{S}$  if and only if

$$0 \leq n_j \leq \Lambda_j(x). \quad (2)$$

## 2.2 The Closed Form Solution

The network dynamics can be modeled as a continuous-time stochastic process. Let  $n(t) = (n_1(t), \dots, n_R(t))$  be the state of the process at time  $t$ , where  $n_i(t)$  represents the number of wavelengths in use on route  $i$  at time  $t$ , and let the state space be  $\mathcal{S}$  described above.

Calls arrive at each route  $i$  following independent Poisson processes with corresponding intensities  $\lambda_i$ ,  $i = 1, \dots, R$ .

Upon arrival of a call for route  $i$  at time  $t$ , it is accepted if there is still at least one wavelength available, that is, if the current state satisfies:

$$\max_{l: i \in C_l} \sum_{j \in C_l} n_j(t^-) < \Lambda.$$

The call is said to be connected, and  $n_i(t) = n_i(t^-) + 1$ , while  $n_j(t) = n_j(t^-)$  for  $j \neq i$ . If an incoming call to route  $i$  finds all wavelengths in use at one of the cliques  $C_l \ni i$ , then the call is *blocked* (lost) and  $n(t) = n(t^-)$ . Calls stay connected for a random “holding” time, assumed independent of the rest of the process history. All holding times are identically distributed with mean  $1/\mu$ . When a call on route  $i$  terminates, the occupancy component  $n_i(t)$  is decreased by one.

**Definition 2** The performance measure of interest is called the blocking probability, and it is defined as the long term probability that an incoming arrival is lost:

$$B = \lim_{t \rightarrow \infty} \frac{\sum_{i=1}^R Y_i(t)}{A(t)},$$

where  $Y_i(t)$  is the total number of calls lost on route  $i$  up to time  $t$  and  $A(t)$  is the total number of arrivals up to time  $t$ .

From the dynamics of the process, it follows that the set of blocking states for route  $i$  is given by:

$$\mathcal{B}_i = \left\{ n \in \mathcal{S} : \max_{l: i \in C_l} \sum_{j \in C_l} n_j = \Lambda \right\}, \quad i = 1, \dots, R, \quad (3)$$

so that incoming calls to route  $i$  are blocked if the current state is in the set  $\mathcal{B}_i$ . The blocking probability can be expressed in terms of the blocking probabilities per route:

$$B = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) B_i, \quad (4)$$

$$B_i = \sum_{n \in \mathcal{S}} \pi(n) \mathbf{1}_{\{n \in \mathcal{B}_i\}}, \quad (5)$$

where  $\pi(\cdot)$  are the stationary probabilities of the process  $\{n(t)\}$  and  $\lambda = \sum_{i=1}^R \lambda_i$ . Let  $\rho_i = \lambda_i/\mu$  be the “offered traffic” to route  $i$ , then it can be shown (see Vázquez-Abad, Andrew, and Everitt, 1999) that:

$$\pi(n) = \frac{1}{G} \prod_{i=1}^R \left( \frac{\rho_i^{n_i}}{n_i!} \right), \quad n \in \mathcal{S} \quad (6)$$

$$G = \sum_{n \in \mathcal{S}} \prod_{i=1}^R \left( \frac{\rho_i^{n_i}}{n_i!} \right),$$

where  $G$  is called the normalising constant.

## 3 PERFORMANCE ESTIMATION

Evaluating blocking probabilities using (5)–(6) directly is a difficult numerical problem for realistic sized networks. A typical WDM backbone network may have over 20 nodes and 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 (6) over all states  $n \in \mathcal{B}_i$ . The

number of routes is  $R = n^2/2 + o(n^2)$ , and for densely connected networks, the number of states is  $\mathcal{O}(\Lambda^R)$ . Thus computing  $G$  directly takes of the order of  $\Lambda^{n^2/2}$  multiplications. For a modest network of 10 nodes with eight wavelengths, this requires around  $8^{45} \approx 10^{40}$  multiplications, or  $10^{21}$  years on a 1 Tflops computer. These times are clearly impractical, which leaves simulation as the only option. Two approaches seem natural for simulation as a numerical method for approximating the required probabilities: generation of the random variables with the truncated probabilities that appear in the closed form solution, and simulation of the original process. In this paper we deal with the first approach.

### 3.1 Relative Efficiency

Suppose that a random variable  $X$  can be generated under the distribution  $\pi$  of (6) (denoted  $X \sim \pi$ ). Then  $S$  consecutive replications can be used to form the sample average estimator, say  $\hat{Y}(S)$ . While simulation is a viable numerical tool, it yields a random estimator as an approximation and it is essential to estimate also the approximation error. Typically, asymptotic normality of  $\hat{Y}(S)$  can be established, in which case the *precision error* at an approximate confidence level  $\alpha$  is  $\epsilon_S = z_{1-\alpha/2} \sqrt{\text{Var}(\hat{Y}(S))}$  ( $z_q$  is the  $q$ th quantile of the standard normal distribution) and it can be shown to be of the order  $\mathcal{O}(S^{-1/2})$  for the estimators considered in this paper.

Design of many networks, such as those carrying multimedia traffic, relies on estimates of blocking probabilities under different loads as well as different model parameters. When estimating probabilities, it is often desirable to achieve a certain prespecified relative precision, e.g.,  $\epsilon_S = 0.01B$ . A fixed relative precision can be obtained faster by either decreasing the CPU time required to generate  $X$  or by using an estimator of  $B$  with reduced variance. The appropriate performance measure for the simulation methods that we shall be describing here is the *relative efficiency*, which measures the trade-off between longer simulations and smaller relative error, and is defined by:

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

where  $\text{CPU}[\hat{Y}(S)]$  denotes the average CPU time of the simulation that produces the  $S$  samples. The *asymptotic relative efficiency* is defined as  $\lim_{S \rightarrow \infty} \mathcal{E}_r(\hat{Y}(S))$ , if this limit exists.

### 3.2 Direct Simulation

The method of acceptance/rejection is a natural method for generating a random variable  $X \sim \pi$  with the stationary probability  $\pi(n)$  of (6). In the sequel, use subscripts to denote iterations and parenthesis for individual components, so  $X = (X(1), \dots, X(R)) \in \mathbb{N}^R$ . For each  $k = 1, 2, \dots$ , generate a vector  $M_k = (M_k(1), \dots, M_k(R))$  of independent Poisson random variables with respective means  $\rho_i, i = 1, \dots, R$ , and let  $k^*$  be the first index such that  $M_{k^*} \in \mathcal{S}$ . Then  $X = M_{k^*}$  has the desired distribution  $\pi$  in (6), as shown by Vázquez-Abad, Andrew, and Everitt (1999). This method is used in Everitt and Macfayden (1983) and Yates (1997) to calculate (4) as follows. Random variables  $\{X_s, s = 1, \dots, S\}$  are generated with independent and identical replications of

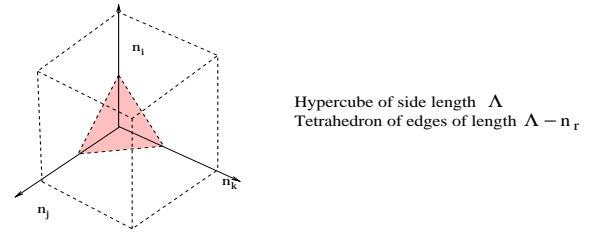


Figure 1: Projection of the feasibility region  $\mathcal{S}$ .

$M_{k^*}$ . Next, compute:

$$Y_s = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{X_s \in \mathcal{B}_i\}},$$

which identifies for which routes the state  $X_s$  is a blocking state and weights the probability accordingly. Notice that states may block several routes, since there exist  $i \neq j$  such that  $\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset$ . By construction,  $Y_s$  is an unbiased estimator of the blocking probability:  $\mathbb{E}[Y_s] = B$ . Using independence, the sample mean converges in the order of  $\mathcal{O}(S^{-1/2})$ , where  $S$  is the simulation length.

Generating  $R$  independent Poisson random variables can be made very fast by means of pre-calculated tables and efficient search methods. However, the acceptance probability,  $G$ , can be very low for moderately large networks, and  $\mathbb{E}[k^*] = G^{-1}$ . In previous studies we have used millions of iterations of the algorithm, on average, before obtaining a sample in the feasible set  $\mathcal{S}$ . To visualise why this happens, consider routes  $i, j$  and  $k$  in a clique, and imagine the feasible region projected onto  $(n_i, n_j, n_k)$  as shown in Figure 1. If the remaining routes in the clique carry a total of  $n_r$  connections, then the projection of the feasible region for this constraint is a tetrahedron contained in the hypercube of side length  $\Lambda - n_r$ . Clearly other cliques will impose further constraints on  $n_i, n_j$  and  $n_k$ . The volume of the feasible region can be considerably smaller than the whole space, resulting in many rejected samples. Moreover, the probability distribution for heavily loaded networks will give more weight to infeasible samples:  $G$  increases with  $\rho_i, i = 1, \dots, R$ .

The relative efficiency of the method decreases as the dimension  $R$ , or the load  $\rho_i$  per route increases, since more of the generated samples must be rejected. The main contribution of this paper is to introduce a simulation method that (a) avoids rejection of samples, thus improving the computational effort, and (b) uses conditioning for variance reduction. Both properties will result in an increase of the relative efficiency, as we shall show.

## 4 MONTE CARLO CHAIN SIMULATION

This section presents the construction of a Markov chain  $\{X_k, k = 1, 2, \dots\}$  with state space  $\mathcal{S}$  that is ergodic and whose limit probabilities are uniquely determined and given exactly by  $\pi$  in (6), that is:

$$\forall n \in \mathcal{S} \quad \lim_{k \rightarrow \infty} \mathbb{P}(X_k = n) = \pi(n). \quad (7)$$

Such simulation methods are known under the generic name of Markov chain Monte Carlo (MCMC) methods. Instead of using the continuous-time physical process of call connections and route/wavelength assignments, a different process is

simulated to estimate the blocking probability  $B$ . This artificial (or “surrogate”) process is defined in discrete time as a Markov chain and it no longer has the interpretation of being an “occupancy process”: calls do not arrive, connect or get rejected, and there is no concept of wavelength assignment or holding times for the Markov chain. The Gibbs sampler is a special case of the MCMC method called the sequential Metropolis Hastings algorithm and is described in Chib and Greenberg (1995), Gilks, Richardson, and Spiegelhalter (1996), Ross (1997) and Brémaud (1999), among others.

#### 4.1 The Standard Gibbs Sampler

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

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

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

$$\pi(y | x^{-j}) = \mathbb{P}[X(j) = y | X^{-j} = x^{-j}] = \frac{\pi(x_y)}{\sum_{x(j)=0}^{\Lambda_j(x)} \pi(x)},$$

where  $x_y(i) = x(i)$  for  $i \neq j$  and  $x_y(j) = y$ , and  $\Lambda_j(x)$  is the state dependent bound such that all states in the sum lie in  $\mathcal{S}$ .

**Definition 3** *The Randomised Gibbs Sampler is a Markov chain  $\{\xi_k\}$ , where  $\xi_{k+1}$  satisfies:*

$$\xi_{k+1}(i) \begin{cases} = \xi_k(i) & \text{if } i \neq \sigma_k, \\ \sim \pi(\cdot | \xi_{k+1}^{-\sigma_k}) & \text{otherwise,} \end{cases}$$

where  $\sigma_k \sim U\{1, \dots, R\}$  is chosen uniformly amongst the  $R$  components, independently of the history of the process.

Denote by  $q_s$  a distribution on  $\{1, 2, \dots, R\}$ . Let the Markov chain  $\{\xi_k\}$  have transition probabilities:

$$\mathbb{P}[\xi_{k+1} = y | \xi_k = x] = \begin{cases} q_s \pi(y(s) | x^{-s}) & \text{if } y(i) = x(i), i \neq s \\ 0 & \text{otherwise.} \end{cases}$$

It can be shown by a reversibility argument (see Ross, 1997 and Brémaud, 1999, Chapter 7) that the stationary probabilities of the process  $\{\xi_k\}$  are precisely  $\pi(n)$ ,  $n \in \mathcal{S}$ , as required in (7). In this paper  $q_s$  represents the uniform distribution. The conditional distribution of the randomly chosen component is very simple to generate when  $\pi$  has a product form. In particular, for (6) it is a one dimensional Poisson distribution truncated to (2). For each  $1 \leq j \leq R$ , let

$$P_j(m) = \sum_{n=0}^m \frac{\rho_j^n}{n!} \quad m = 1, \dots, \Lambda. \quad (8)$$

At every step  $k$ , let  $j = \sigma_k$  and calculate  $\Lambda_j(\xi_k)$ . For clique packing,

$$\Lambda_j(\xi_k) = \Lambda - \max_{i:j \in C_i} \sum_{c \in C_i} \xi_k(c) \mathbf{1}_{\{j \neq c\}}. \quad (9)$$

Then the required conditional probability is a Poisson distribution with parameter  $\rho_j$  truncated at  $\Lambda_j(\xi_k) \leq \Lambda$ , i.e.,

the corresponding distribution satisfies  $\mathbb{P}[\xi_{k+1}(j) \leq m] = P_j(m)/P_j(\Lambda_j(\xi_k))$ ,  $m = 0, \dots, \Lambda_j(\xi_k)$  and can be generated very efficiently by simple reading of the array  $P_j$ . As a result, no samples are rejected and the chain lies completely within the state space  $\mathcal{S}$ . Referring back to Figure 1, the Gibbs sampler updates one component at a time, drawing the random number from the one-dimensional truncated Poisson distribution within the feasible region only.

Consider the random variables:

$$Y_k = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\xi_k \in \mathcal{B}_i\}}. \quad (10)$$

Using dominated convergence,  $Y_k \rightarrow B$  w.p. 1 and the sample averages of  $Y_k$  will converge to  $B$ . Brémaud (1999) has an analysis of the convergence rates (Chapter 7) of the Gibbs sampler. For this example the chain  $\{\xi_k\}$  is geometrically ergodic and sample averages converge to  $B$  also in the order of  $\mathcal{O}(S^{-1/2})$ , with  $S$  samples.

There is no need for the order of the updates to be random. By selecting the components cyclically, the following algorithm avoids the generation of the random variable  $\sigma_k$  for each new sample generated.

**Definition 4** *Consider a Markov Chain  $\zeta_{k+1}$  satisfying:*

$$\zeta_{k+1}(i) \begin{cases} = \zeta_k(i) & \text{if } i \neq \sigma_k = k \bmod(R) + 1, \\ \sim \pi(\cdot | \zeta_{k+1}^{-\sigma_k}) & \text{otherwise.} \end{cases} \quad (11)$$

For any  $r \in \{1, \dots, R\}$ , define the Periodic Gibbs Sampler by  $\zeta_k^{(r)} \equiv \zeta_{r+Rk}$ , which is a periodic sample of the chain  $\{\zeta_k\}$ .

Due to the fixed order in which the updates take place for the  $R$  stages of each iteration, the chain is no longer reversible, as explained by Brémaud (1999). Instead of using the balance equations for reversible chains, one proves directly that  $\pi$  is a stationary probability for the process  $\{\zeta_k^{(r)}\}$ , as explained by Vázquez-Abad, Andrew, and Everitt (1999): one shows that  $\pi(m) = \sum_{n \in \mathcal{S}} \pi(n) \mathbb{P}[\zeta_{k+1}^{(r)} = m | \zeta_k^{(r)} = n]$ . Irreducibility of the finite state chain ensures the existence of a unique ergodic measure for this example, so that if  $\pi \mathbb{P} = \pi$ , where  $\mathbb{P}$  is the transition matrix of the chain, then (7) will also be true. By a similar argument as before, using the CLT for ergodic Markov Chains, the estimators:

$$Y_k^{(r)} = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k^{(r)} \in \mathcal{B}_i\}} \quad (12)$$

are consistent and their sample average converges with rate  $\mathcal{O}(S^{-1/2})$ , where  $S$  is the number of simulations.

#### 4.2 The Gibbs Sampler with Local Estimation

The Gibbs sampler as described in the previous section is the current state of the art. Because the periodic sampler is consistent for every  $r$ , so is the *averaged estimator*:

$$\frac{1}{R} \sum_{r=1}^R \mathbb{E}[Y_k^{(r)}] \rightarrow B \quad \text{as } k \rightarrow \infty.$$

**Proposition 1** Consider the chains  $\{\xi_k\}$  and  $\{\zeta_k\}$  in stationary state, and  $f: \mathcal{S} \rightarrow \mathbb{R}$  a function with finite stationary variance  $v^2$ . The ratio between the efficiency of the randomised estimator and the averaged estimator is:

$$\frac{\mathcal{E}_r[\frac{1}{R} \sum_{k=1}^R f(\zeta_k)]}{\mathcal{E}_r[f(\xi_1)]} = \frac{v^2}{v^2 + \frac{1}{R} \sum_{i \neq j=1}^R \text{Cov}[f(\zeta_i), f(\zeta_j)]}.$$

**Proof:** The variables  $\zeta_1, \dots, \zeta_R$  have a common distribution  $\pi$ . Let  $\sigma \sim U\{1, \dots, R\}$  be a uniformly distributed index, independent of  $\{\zeta_i\}$ . Then, conditioning on  $\sigma$ :

$$\begin{aligned} \text{Var}[f(\zeta_\sigma)] &= \frac{1}{R} \sum_{i=1}^R \text{Var}[f(\zeta_i)|\sigma = i] + \text{Var}[E[f(\zeta_i)|\sigma = i]] \\ &= \frac{1}{R} \sum_{i=1}^R \text{Var}[f(\zeta_i)] \equiv v^2, \end{aligned}$$

where we have used that  $E[f(\zeta_i)|\sigma = i]$  is independent of  $i$ , from the assumption that all  $\zeta_i$  have a common distribution, and therefore their variance (w.r. to  $\sigma$ ) is zero. Then the efficiency of the randomised estimator is proportional to  $1/v^2$ .

The averaged estimator, on the other hand, satisfies:

$$\text{Var} \left[ \frac{1}{R} \sum_{i=1}^R f(\zeta_i) \right] = \frac{1}{R} \left[ v^2 + \frac{1}{R} \sum_{i \neq j=1}^R \text{Cov}[f(\zeta_i), f(\zeta_j)] \right].$$

Because the averaged estimator requires computing  $R$  evaluations of the function  $f$  while the randomised only requires one, the efficiency of the averaged one is proportional to

$$\frac{1}{v^2 + \frac{1}{R} \sum_{i \neq j=1}^R \text{Cov}[f(\zeta_i), f(\zeta_j)]}.$$

The proof is finished by noticing that  $\xi_1 \stackrel{\mathcal{L}}{=} \zeta_\sigma$ , because the distribution of  $\xi_0$  is the same as the distribution of all  $\zeta_i$ , and component  $\sigma$  is the one being updated.  $\triangleleft$

**Corollary 1** Assume that the stationary covariance of a function  $f(\zeta_k)$  within periods is negative (positive), that is:  $\text{Cov}(f(\zeta_i), f(\zeta_j)) < (>) 0$ ,  $i \neq j$ ;  $i, j \in \{1, \dots, R\}$ , where  $\zeta_1 \sim \pi$  has the stationary distribution. Then the average estimator  $(1/R) \sum_{k=1}^R f(\zeta_k)$  has a better (worse) efficiency than the randomised Gibbs sampler  $f(\xi_1)$ , given  $\zeta_0 = \xi_0 \sim \pi$ .

The averaged version of the estimation is equivalent to what we call the ‘‘sequential’’ Gibbs sampler, which follows the chain  $\{\zeta_k\}$  and sets the estimation in terms of

$$Y_k = \sum_{i=1}^R \left( \frac{\lambda_i}{\lambda} \right) \mathbf{1}_{\{\zeta_k \in \mathcal{B}_i\}}. \quad (13)$$

This estimator may present the problem of high correlations, as is apparent from Proposition 1. Most of the terms in the sums for (10) or (13) will not be affected by changes in the component  $\sigma_k$ : if changes in component  $\sigma_k$  do not affect the

state of the cliques that contain route  $i$ , then  $P[\mathbf{1}_{\{\zeta_{k+1} \in \mathcal{B}_i\}} = \mathbf{1}_{\{\zeta_k \in \mathcal{B}_i\}}] = 1$ . In our simulations, there are  $R = 300$  routes and in many iterations the component being updated does not greatly influence the statistic  $Y_{k+1}$  given its previous value  $Y_k$ . Although we cannot give a formal proof of this fact, this simple observation leads to the conjecture that the statistics  $\{Y_k\}$  are highly correlated, and this computation is of little use, hindering efficiency.

For any  $i$ , the estimators  $Y_{k,i} = \mathbf{1}_{\{\zeta_{k+1} \in \mathcal{B}_i\}}$  are consistent for  $B_i$ . Indeed, since  $\{\zeta_k\}$  has stationary probability  $\pi$ , then  $\lim_{k \rightarrow \infty} E[Y_{k,i}] = \pi(\mathcal{B}_i) = B_i$ . To alleviate the problem of inefficient calculations in the sums, we propose to choose only one term  $i(k)$  to be updated at iteration  $k$ . Judicious choice of the sequence of updates for the sequential Gibbs sampler can in fact introduce a degree of negative correlation into the estimates. Consider two consecutive updates with  $\sigma_k$  and  $\sigma_{k+1}$  in a common clique. If  $\zeta_k(\sigma_k)$  is large, then  $\Lambda_{\sigma_k}(\zeta_k) \geq \zeta_k(\sigma_k)$  will be large. Thus  $E[Y_{k,\sigma_k}|\zeta_k]$  will be small, since it decreases monotonically with  $\Lambda_{\sigma_k}(X_k)$ . Moreover,  $\zeta_{k+1}(\sigma_k) \leq \Lambda_{\sigma_k}(\zeta_k)$  is allowed to be large. If indeed  $\zeta_{k+1}(\sigma_k)$  is large, then  $\Lambda_{\sigma_{k+1}}(\zeta_{k+1}) \leq \Lambda - \zeta_{k+1}(\sigma_k)$  will be small (since  $\sigma_{k+1}$  and  $\sigma_k$  are in a common clique), causing  $E[Y_{k+1,\sigma_{k+1}}|\zeta_{k+1}]$  to be large. Thus if consecutive updates are often in the same clique, then a small  $Y_{k,\sigma_k}$  will often be followed by a large  $Y_{k+1,\sigma_{k+1}}$ . This effect arises due to the truncation of the state space, and should thus become more apparent as the blocking increases.

Using consistency and ergodicity of the sequential Gibbs sampler, it follows that for each  $i = 1, \dots, R$ ,

$$\lim_{S \rightarrow \infty} \frac{1}{S(i)} \sum_{k=1}^S Y_{k,i} \mathbf{1}_{\{\sigma_k=i\}} \rightarrow B_i$$

a.s., where  $S(i) = \sum \mathbf{1}_{\{\sigma_k=i\}}$  counts the number of iterations where  $\sigma_k = i$ , and the convergence is of the order  $\mathcal{O}(S^{1/2})$ . We call these the *local estimates*.

### 4.3 Filtered Gibbs Sampler

Consider a Markov Chain  $\{X_k\}$  and an estimator of the form:

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

which is a sample average (other functional forms can also be considered in the general setting). The method known as *Filtered Monte Carlo* is based on conditioning at each stage, as explained by Ross (1997), obtaining:

$$\bar{X}'_S = \frac{1}{S} \sum_{k=1}^S E[f(X_k)|X_{k-1}].$$

For a single random variable it is always true that  $\text{Var}(X) = \text{Var}(E[X|Z]) + E(\text{Var}[X|Z])$ , and therefore conditioning entails a variance reduction. For example, if  $X$  is a Bernoulli variable, it always takes the extreme values 0 and 1, while its conditional expectation is a probability  $P(X = 1|Z) \in [0, 1]$  and thus has less variability. However, it is not always the case for Markov chains that  $\text{Var}[\bar{X}'_S] \leq \text{Var}[\bar{X}_S]$ , due to the correlation structure. Fortunately, when estimating probabilities we can always ensure variance reduction.

**Proposition 2** Let  $\{X_k\}$  be a stationary Markov chain with state space  $\mathcal{S} \subset \mathbb{N}^R$  and  $f(x) = \mathbf{1}_{\{x \in A\}}$  the indicator function of a set  $A \in \mathcal{S}$ . Then  $\text{Var}[\bar{X}'_S] \leq \text{Var}[\bar{X}_S]$ .

**Proof :** The variance of the sample average is:

$$\text{Var}[\bar{X}_S] = \frac{1}{S} \text{Var}[f(X_1)] + \frac{1}{S^2} \sum_{k \neq j=1}^S \text{Cov}[f(X_k), f(X_j)].$$

The first term contains the variance of the Bernoulli variable  $\text{Var}[f(X_1)] = \text{P}(A)(1 - \text{P}(A))$ , which is no smaller than the variance of the conditional expectation  $\text{Var}\{\text{E}[f(X_1)|X_0]\}$  as usual. Call  $\kappa_k = \text{P}(X_{k+1} \in A|X_k)$ , then:

$$\text{Var}[\bar{X}'_S] = \frac{1}{S} \text{Var}[\kappa_1] + \frac{1}{S^2} \sum_{k \neq j=1}^S \text{Cov}[\kappa_k, \kappa_j].$$

For any two adjacent terms,

$$\begin{aligned} \text{Cov}[f(X_k), f(X_{k+1})] &= \text{E}[f(X_k)f(X_{k+1})] - \text{P}(A)^2 \\ &= \text{E}[\text{E}[f(X_k)f(X_{k+1})|X_k, X_{k-1}]] - \text{P}(A)^2 \\ &= \text{E}[\text{E}[f(X_k)\kappa_k|X_{k-1}]] - \text{P}(A)^2. \end{aligned}$$

Use now the fact that  $f(X_k) = \mathbf{1}_{\{X_k \in A\}}$  to establish that the inner conditional expectation has value  $\kappa_k$  on the set where  $X_k \in A$  (conditioned on  $X_{k-1}$ ) and zero otherwise, so that

$$\begin{aligned} \text{Cov}[f(X_k), f(X_{k+1})] &= \text{E}[\text{E}[\mathbf{1}_{\{X_k \in A\}}|X_{k-1}]\kappa_k] - \text{P}(A)^2 \\ &= \text{E}[\kappa_{k-1} \kappa_k] - \text{P}(A)^2 \\ &= \text{Cov}[\kappa_{k-1}, \kappa_k]. \end{aligned}$$

Using a similar argument for other terms, it can be shown that  $\text{Cov}[f(X_k), f(X_j)] \geq \text{Cov}[\kappa_k, \kappa_j]$ , completing the proof.  $\triangleleft$

Applying this method to the Gibbs samplers requires evaluation of the conditional probabilities:

$$\begin{aligned} \text{P}[X_{k+1} \in \mathcal{B}_j|X_k] &= \frac{P_j(\Lambda_j(X_k)) - P_j(\Lambda_j(X_k) - 1)}{P_j(\Lambda_i(X_k))} \\ &\equiv g(\Lambda_j(X_k); \rho_j) \end{aligned}$$

where  $P_j(\cdot)$  are given in (8) and  $\Lambda_j(X_k)$  is given in (9). Therefore conditioning not only reduces the variance, but when it is feasible to pre-compute  $g(\cdot; \cdot)$ , calculation of the probabilities is as simple as reading a table. This is the case when network traffic is uniform, or the load on each route is selected from a small set.

Our proposed method combines the filtering with the distribution of the estimation via the local estimates as follows.

**Definition 5** The Filtered Sequential Gibbs Sampler is constructed from the chain  $\{X_k\}$  with transitions governed by:

$$X_{k+1}(i) \begin{cases} = X_k(i) & \text{if } i \neq \sigma_k = k \bmod(R) + 1, \\ \sim \pi(\cdot|X_{k+1}^{-\sigma_k}) & \text{otherwise.} \end{cases}$$

by evaluating the sample average:

$$\hat{Y}(S) = \frac{R}{S} \sum_{k=1}^S \left( \frac{\lambda_{\sigma_k}}{\lambda} \right) \text{P}[X_{k+1} \in \mathcal{B}_{\sigma_k}|X_k]. \quad (14)$$

Each of the periodic Gibbs samplers embedded in the computation of (14) is dedicated to estimating  $\mathcal{B}_{\sigma_k}$ . Since  $S(\sigma_k)/S \rightarrow 1/R$  as  $S \rightarrow \infty$ , the Filtered Gibbs sampler is strongly consistent for the blocking probability  $B$ . Yet the distribution of the estimation considerably improves the computational effort, as the results of the following section show.

Filtering can be applied analogously to the randomised version of the Gibbs sampler, which is done in the results of the following section.

## 5 NUMERICAL RESULTS

The network topology used was a  $5 \times 5$  mesh-torus topology with clique packing. Two networks were considered: one with  $\Lambda = 8$  wavelengths per link, and one with  $\Lambda = 32$  wavelengths per link. The performance of the estimators was compared for a range of loads. The loads were selected to yield blocking probabilities in the range  $10^{-6}$  to 0.1, which is typical of current networks.

The method of batch means (see Alexopoulos and Seila, 1998) was used to estimate the variance of the estimators. This method is based on regrouping  $\beta$  consecutive samples to form a ‘‘batch mean’’  $\bar{Y}_s, s = 1, \dots, S$  and simulating a total of  $S$  batches:

$$\begin{aligned} \bar{Y}_s &= \frac{1}{\beta} \sum_{k=1}^{\beta} Y_{(s-1)\beta+k}, & \hat{Y}(S) &= \frac{1}{S} \sum_{s=1}^S \bar{Y}_s, \\ \text{and } \hat{V} &= \widehat{\text{Var}}[\hat{Y}(S)] = \frac{1}{S(S-1)} \sum_{s=1}^S (\bar{Y}_s - \hat{Y}(S))^2. \end{aligned}$$

The experiments used  $S = 10000$  samples, with a batch size of  $\beta = 10000$ . Because of its low computational efficiency, the A/R algorithm used  $S = 1000$  and  $\beta = 100$  (with no reduction in resulting efficiency). The early rejection approach of Vázquez-Abad, Andrew, and Everitt (1999) was used for A/R. These experiments were conducted on a 266 MHz Pentium II processor using the gnu C++ compiler under the Linux operating system. In the simulations that follow, the arrival rate was the same over all of the routes, so that  $\rho_i \equiv \rho$  is constant.

Table 1 shows the estimator of the blocking probability, its standard deviation and the CPU time required for the A/R method, which shows the expected deterioration as load and number of wavelengths increase.

Figure 2 shows the relative efficiency for direct simulation of  $\pi$  using A/R, and the variants of the Gibbs sampler described in Section 4.1. The computation of the Gibbs algorithms is dominated by the evaluation of the state blocking probability, and is thus similar in all cases. Since the random and sequential variants increase the correlation between consecutive states, they increase the variance, resulting in reduced efficiency. While the periodic Gibbs sampler performs comparably with A/R when the probability of rejection is low, it outperforms A/R substantially as the size of the state space and/or the load grows, as expected.

The estimated blocking probabilities and their estimated standard deviations are shown in Table 2 for the the periodic (P) and filtered sequential (F) Gibbs samplers. At first sight, it seems from the table that our proposed method does not perform better than the periodic sampler, exhibiting always greater variance. However, the periodic estimator is

$\rho$	$\hat{Y}(1000)$	$\sqrt{\hat{V}}$	time (s)
$\Lambda = 8$ wavelengths			
0.05	2.93333e-06	1.19608e-06	32.3
0.1	0.0003878	1.41897e-05	32.4
0.15	0.00427497	4.77054e-05	32.8
0.2	0.0181383	9.8596e-05	35.7
0.25	0.0456161	0.000160489	46.3
0.3	0.0836662	0.000203071	83.6
0.35	0.128659	0.000248271	253
$\Lambda = 32$ wavelengths			
1.0	0.0001242	7.89775e-06	32.5
1.2	0.00199897	3.12384e-05	33.2
1.4	0.0122448	7.6496e-05	40.7
1.6	0.0383563	0.000138135	97.7
1.8	0.0779384	0.00019375	1290
2	0.124802	0.000237888	273000

Table 1: Estimated blocking probabilities, its standard deviation, and the required CPU time for direct simulation.

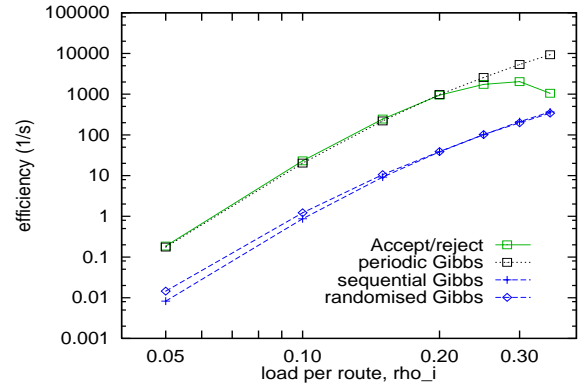
$\rho$	$\hat{Y}_P(10000)$	$\sqrt{\hat{V}_P}$	$\hat{Y}_F(10000)$	$\sqrt{\hat{V}_F}$
$\Lambda = 8$ wavelengths				
0.05	3.198e-06	4.018e-08	3.324e-06	2.721e-07
0.1	0.0003803	4.409e-07	0.0003764	3.203e-06
0.15	0.004307	1.509e-06	0.004297	1.252e-05
0.2	0.01836	3.047e-06	0.01832	2.762e-05
0.25	0.04607	4.733e-06	0.04605	4.290e-05
0.3	0.08466	6.025e-06	0.08476	5.414e-05
0.35	0.128772	6.923e-06	0.128901	6.095e-05
$\Lambda = 32$ wavelengths				
0.8	2.057e-06	3.186e-08	2.075e-06	1.126e-07
1.0	0.0001258	2.522e-07	0.0001249	1.083e-06
1.2	0.002027	1.014e-06	0.002023	5.291e-06
1.4	0.01240	2.508e-06	0.01240	1.462e-05
1.6	0.03836	4.193e-06	0.03834	2.517e-05
1.8	0.07771	5.663e-06	0.07769	3.131e-05
2	0.1233	6.651e-06	0.1233	3.369e-05

Table 2: Estimators and standard deviation for periodic,  $\hat{Y}_P$ , and filtered,  $\hat{Y}_F$ , sequential Gibbs samplers.

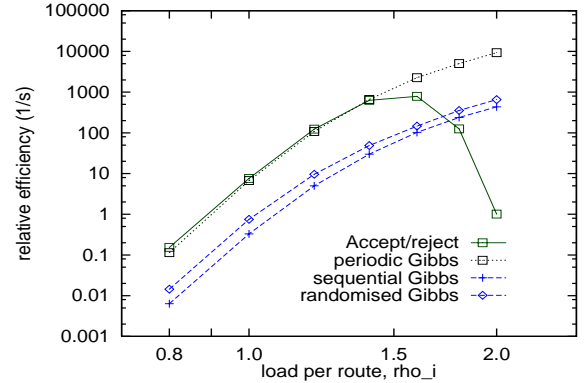
$R = 300$  times slower. It requires  $R$  transitions of the Markov chain (11) per sample, compared to one for the sequential sampler, and there  $R$  terms in the sum (12), compared to one for local estimation.

Figure 3 shows the corresponding efficiencies when the Gibbs samplers are implemented via their filtered versions with local estimation. These versions are substantially more efficient than the standard Gibbs sampler. The filtered sequential Gibbs algorithm is at least an order of magnitude better than direct simulation. Some of the improvement comes from the greater speed of local estimation, but filtering reduces the variance by an additional factor of two compared to the unfiltered local variant.

In teletraffic engineering, a network is typically designed to satisfy a maximum blocking probability dependent on human factors, which does not vary as the network grows. Thus the efficiency for a given blocking probability (Figure 4) is an important performance measure for an algorithm. The ratio of the relative efficiencies is the inverse ratio of the CPU times required to achieve a fixed relative precision. As seen from the plot, for  $B = 0.0004$ ,  $\Lambda = 8$  (moderate network sizes and loads) our method requires 4.2 times less CPU time to com-



(a) 8 wavelengths per link



(b) 32 wavelengths per link

Figure 2: Efficiency against offered load for direct simulation, and random, periodic and sequential Gibbs samplers applied to a  $5 \times 5$  mesh-torus.

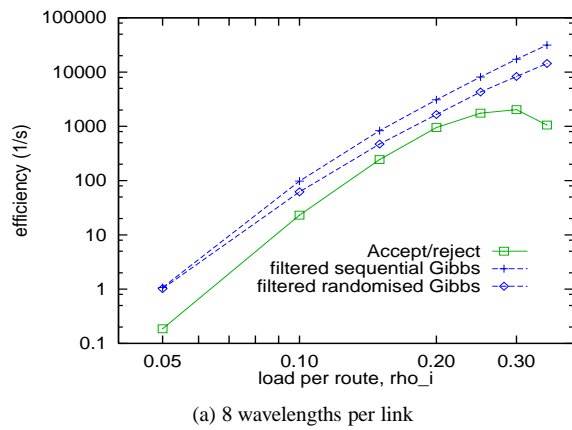
pute the approximation at the same level of relative error. For the more realistic model with  $\Lambda = 12$  at  $B = 0.12$ , the gain factor is 93,931: if our method requires 1 minute, the usual A/R would need 65.2 days to complete the simulation.

Not only does our proposed method clearly outperform A/R, but its performance continues to increase for increasing load at lower blocking rates as  $\Lambda$  increases, dramatically overcoming the curse of dimensionality.

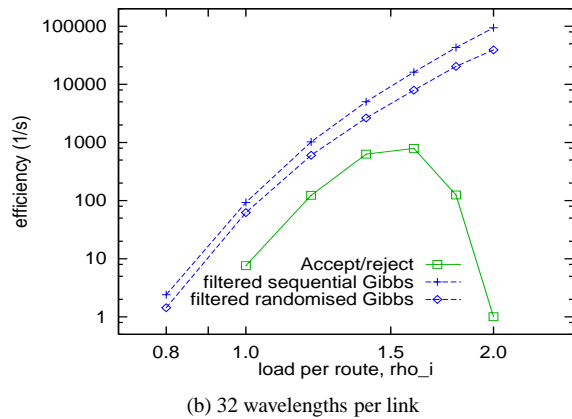
## 6 CONCLUDING REMARKS

The MCMC method proposed for blocking probability calculations, which we call the filtered Gibbs sampler, not only outperforms the usual acceptance/rejection method, but its relative efficiency actually grows with problem size and with increasing load. Most of the efficiency improvement for blocking probabilities is a consequence of the combination of filtering and localisation of the estimation. For some systems, other techniques may prove impossible to implement whilst our method overcomes the curse of dimensionality.

As mentioned before, any distribution  $q_s$  can be used for the randomised Gibbs sampler. Moreover, in the localised estimation, it is not necessary to use  $i(k) = \sigma_k$ . Future work involves adaptively establishing which component  $\sigma_k$  should be used for the *estimation* (not only for the updates) in order to minimise the overall variance: those indices  $i$  for which  $B_i$  is small could be allocated more samples. This method,



(a) 8 wavelengths per link



(b) 32 wavelengths per link

Figure 3: Efficiency against offered load for direct simulation and filtered Gibbs samplers applied to a  $5 \times 5$  mesh-torus.

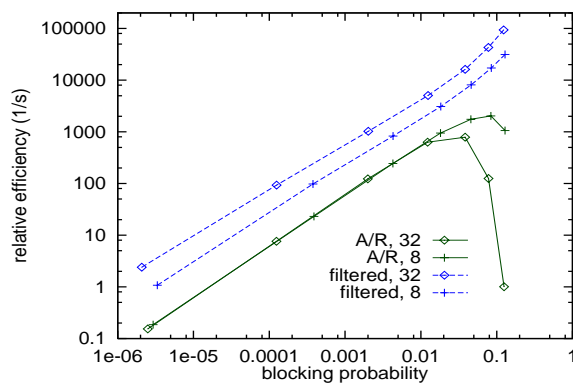


Figure 4: Efficiency against blocking probability for direct simulation and filtered sequential Gibbs samplers with  $\Lambda = 8$  and  $\Lambda = 32$  wavelengths.

known as stratification, could in principle yield even better performance of the Gibbs sampler and is the subject of future research.

## AUTHOR BIOGRAPHIES

**Felisa J. Vázquez-Abad** is Professor at the Department of Computer Science and Operations Research at the University of Montreal. She is also Fellow at the Department of Electrical and Electronic Engineering at the University of Melbourne. She received her B.Sc. degree in Physics in 1983 and her M.Sc. degree in Statistics and OR in 1984 from Uni-

versidad Nacional Autónoma de México, and her Ph.D. degree in Applied Mathematics in 1989 from Brown University. Her research interests include stochastic control and simulation with applications in telecommunications, manufacturing, transportation, insurance and finance. She is a member of SIAM, INFORMS and IEEE. She is Web Editor of the INFORMS College on Simulation and Associate Editor of the IEEE CSS.

**Lachlan L. H. Andrew** is a research fellow in the Department of Electrical and Electronic Engineering at the University of Melbourne, Australia. He received his B.Sc. in computer science in 1992, B.E. in electrical engineering in 1993 and Ph.D. in engineering in 1996, all from the University of Melbourne. His research interests include performance analysis and resource allocation in optical and wireless communication networks. He is a member of the IEEE and associate member of the IEE.

## References:

- Alexopoulos, C., and A. Seila. 1998. Output data analysis. chapter 7. In *Handbook of Simulation*, ed. J. Banks, 225–272. John Wiley & Sons.
- Borella, M. S., J. P. Jue., D. Banerjee., B. Ramamurthy., and B. Mukherjee. 1997. Components for WDM lightwave networks. *Proc. IEEE*, 85(8):1274–1307.
- Brémaud, P. 1999. *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*. Texts in Applied Mathematics, 31, New York: Springer.
- Chib, S., and E. Greenberg. 1995. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335.
- Everitt, D. 1991. Product form solutions in cellular mobile communication systems. In *Teletraffic and Datatraffic in a Period of Change*, ed. A. Jensen and V. Iversen, 483–488. North-Holland.
- Everitt, D. 1994. Traffic engineering of the radio interface for cellular mobile networks. *Proc. IEEE*, 82(9):1371–1382.
- Everitt, D., and N. Macfayden. 1983. Analysis of multicellular mobile radiotelephone systems with loss. *Br. Telecom Technol. J.*, 1(2):37–45.
- Gilks, W., S. Richardson., and D. Spiegelhalter. 1996. *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.
- Kind, J., T. Niessen., and R. Mathar. 1998. Theory of maximum packing and related channel assignment strategies for cellular radio networks. *Math. Meth. Op. Res.*, 48(1):1–16.
- Nelson, R. D. 1993. The mathematics of product form queuing networks. *Computing Surveys*, 25(3):339–369.
- Ramaswami, R. 1993. Multiwavelength networks for computer communications. *IEEE Commun. Mag.*, 31(2):78–88.
- Raymond, P.-A. 1991. Performance analysis of cellular networks. *IEEE Trans. Commun.*, 39(12):1787–1793.
- Ross, S. 1997. *Simulation*. Second ed. Boston: Academic Press.
- Vázquez-Abad, F., L. Andrew., and D. Everitt. 1999. Estimation of blocking probabilities in cellular networks with dynamic channel assignment. (submitted).
- Yates, J. 1997. Performance analysis of dynamically-reconfigurable wavelength division multiplexed networks. PhD thesis, University of Melbourne, Australia.