Estimation of Blocking Probabilities in Cellular Networks with Dynamic Channel Assignment

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 and David Everitt †
Department of Electrical and Electronic Engineering
The University of Melbourne, Victoria 3010, Australia
Email: {l.andrew,d.everitt}@ee.mu.oz.au

November 15, 1999

Abstract

Blocking probabilities in cellular mobile communication networks using dynamic channel assignment are hard to compute for realistic sized systems. This computational difficulty is due to the structure of the state space, which imposes strong coupling constraints amongst components of the occupancy vector. In this paper we introduce three methods for efficient estimation of these blocking probabilities via simulation. First we focus on first- and second-generation cellular systems, for which we propose to use Markov Chain Monte Carlo (MCMC) methods. We show numerically that these methods dramatically outperform the currently used direct simulation approach. The second part of the paper deals with third-generation cellular systems, where blocking may be a rare event. We show that an importance sampling (IS) estimator with swapping rates is asymptotically efficient, but it requires a simulation time that grows super-linearly with problem size. Next we apply the change of measure to the standard clock simulation framework, which requires only linear computational effort in problem size.

1 Introduction

Efficient design of communications networks requires the ability to determine the quality of service provided by a particular network configuration. A common quality of service measure is the blocking probability, which is the probability that a new call will not be admitted

*Supported in part by NSERC-Canada grant # WFA0184198 and by the Australian Research Council while the author is on leave at the Department of Electrical and Electronic Engineering, University of Melbourne.
†Supported by the Australian Research Council (ARC).
to the network due to insufficient network resources. This paper will consider techniques for determining the blocking probability in cellular telephony systems with frequency reuse, including first generation systems such as the Advanced Mobile Phone System, AMPS (Lee, 1995), and second generation systems such as the Global System for Mobile Communication, GSM (Mouly and Pautet, 1992, Redl, Weber, and Oliphant, 1995).

In cellular networks, each mobile station communicates with a base station, which is connected to the wireline telephone network. The region around a given base station in which mobiles connect to that station is called a cell. Cells are distributed over the system coverage area, and cells are usually assumed, for convenience, to be hexagonal, such that the hexagonal cells tessellate the entire system coverage area. Each mobile station communicates with its base station using a specific frequency pair or frequency/timeslot pair known as a “channel”. To avoid interference, this channel cannot be used in nearby cells; however, it may be reused in cells a sufficient distance away such that interference caused by the reused channel is, in a worst-case scenario, below some specified threshold value.

In static assignment schemes, each cell is allocated a fixed subset of the available channels, and calls originating in a cell are connected only when there are free channels available from those channels preallocated to that cell. While ensuring that the constraints will be satisfied, static allocation may result in wasted resources: all the preallocated channels for one particular cell may be in use by concurrent calls, but adjacent cells may have free capacity that could be used to connect incoming calls.

Capacity can be improved by dynamic channel assignment (Cox and Reudink, 1972, Cox and Reudink, 1973), in which channels not currently in use in the nearby cells may be used. The capacity can be further improved by allowing existing calls in surrounding cells to be moved to different channels in order to make room for a new call. There are many possible algorithms for implementing dynamic channel assignment, and all have different blocking performance. One scheme which is used as a benchmark for the performance of specific algorithms is the maximum packing strategy of Everitt and Macfayden (1983). The maximum packing strategy assumes rearrangement of calls in progress to accommodate incoming calls wherever possible, while still satisfying the channel reuse constraints. The advantage of the maximum packing formulation is that it allows a closed product-form expression for the state probabilities, subject to the (difficult) determination of a normalizing constant. This product form solution applies to other more complex channel allocation strategies (Everitt, 1991, Everitt, 1994), and is similar to that obtained in other telecommunications networks, e.g., circuit-switched networks (Harvey and Hills, 1979), and wavelength-division multiplexed (WDM) optical networks (Yates, 1997). Furthermore, the product form solution is insensitive to holding time distribution (Everitt and Macfadyen, 1983) which means that a simulation methodology based on exponential holding time distributions can be used to generate numerical results for general holding time distributions.

In Section 2, the model for the channel occupancy process is introduced. While this model does not always reproduce the occupancy process under maximum packing, the corresponding blocking probability yields an accurate approximation to the actual one. A closed formula for the stationary probability of the process is given and two equivalent expressions are used for the blocking probability. Evaluating blocking probabilities using the closed formulas requires evaluation of a normalizing constant, and this problem is NP hard as the number of cells and/or channels available increase, which has led most researchers to use simulation instead. Section 3 introduces the general simulation framework. In Section 4, estimation of blocking probabilities
via direct simulation is presented, discussing the acceptance/rejection or “naïve” simulation method, a Metropolis-Hastings algorithm and the Gibbs Sampler. Section 5 deals with the case of cellular networks where the incoming call rate is smaller than the inverse mean holding time, which is relevant in third generation mobile systems (Prasad and Ojanperä, 1998, ETSI, 1997). Fast simulation is introduced in Section 5 and applied to clusters of cells to estimate blocking per cell under Importance Sampling. Using a modification of the queueing network framework of Chang, Heidelberger, and Shahabiddin (1995) it is shown that the ensuing estimator has uniformly bounded relative error. Next the Standard Clock technique of Vakili (1991) for efficient simulation is used to redefine the appropriate probability spaces and reformulate the change of measure for fast simulation. Section 6 draws some conclusions.

2 Blocking Probabilities

Consider a cellular network with $C$ channels that can be reused across the network, provided that reuse constraints are satisfied. For the purposes of this paper, we assume that the reuse constraints are satisfied provided that any channel is not simultaneously in use in any two adjacent cells. This is known as 1-cell buffering or 3-cell reuse (as a channel can be used in only one cell in every group of three closest adjacent cells in the network). The methodology of this paper is easily extended to cover other sets of reuse constraints.

The dynamic channel assignment of Everitt and Macfayden (1983) is called the maximum packing strategy and it prescribes rearrangement of calls in progress to accommodate incoming calls while still satisfying the channel reuse constraint. In this manner the number of calls carried by the network can be maximised.

Everitt and Macfayden (1983) consider a lower bound on the blocking probability of the maximum packing strategy, obtained by ignoring the identity of the individual channels in use at any moment and considering only the number of channels in use in each cell. Raymond (1991) shows that this bound, called the clique packing bound, is a very accurate approximation to the maximum packing blocking probability.

We will now describe maximum packing for the simple 7-cell cellular network of Figure 1. This network consists of $K = 7$ cells and $C$ channels. When discussing the 7-cell network, we will use the convention that $7 + 1 = 2$, so that adjacent cells are always subsets of cells labeled $(1,i,i+1)$ (these subsets are called “cliques”). Under the maximum packing strategy for this example an incoming call to cell $i$ can be connected whenever each clique of which cell $i$ is a member contains fewer calls than $C$.

For $i \neq 1$, clique packing corresponds to the total number of channels in use in each of the two cliques $(1, i, i + 1)$ and $(1, i - 1, i)$ covering cell $i$ being smaller than the total number of available channels $C$, that is:

\[ n_1 + n_i + \max\{n_{i+1} + n_{i-1}\} \leq C. \quad (1a) \]

For $i = 1$, clique packing corresponds to the total number of channels in use in each of the six cliques $(1, j, j + 1), j = 2 \cdots 7$ covering cell 1 being smaller than $C$, that is:

\[ n_1 + n_j + n_{j+1} \leq C, \quad j = 2 \cdots 7. \quad (1b) \]

We will now demonstrate for the 7-cell system that clique packing with a maximum of $C$ calls per clique can produce a channel allocation which requires at most $C$ channels. Let us
Figure 1: Simple Cellular Network Model

begin with one channel $C = 1$, and follow the decomposition approach of Tissainayagam and Everitt (1999). The single channel may be allocated to any of the following sets of cells \{1\}, \{2,4,6\}, \{3,5,7\}, \{2,5\}, \{3,6\}, \{4,7\}, and only these sets or subsets of these sets satisfy the clique packing conditions (1) by enumeration. Now for $C \geq 1$ we decompose the system into $C$ single channel systems, with each channel allocated to one of the sets of cells above. Since each of the $C$ channels individually satisfies the clique packing criteria, so must the $C$ channels jointly.

The continuous-time Markov process for this model is now described. For ease of notation, refer to the seven cell model, and to calculation of blocking probabilities at the outer cells $i$, $i = 2, \ldots, 7$ for which (1a) must be satisfied; the main ideas can be extended to more complex cellular structures, and to all cells within the structure.

Let $n(t) = (n_1(t), \ldots, n_7(t))$ be the state of the process at time $t$, where $n_i(t)$ represents the number of channels in use at cell $i$ at time $t$, and let $S$ be the state space consisting of all integer vectors $n = (n_1, \ldots, n_7) \in \mathbb{N}^K$ satisfying (1).

At each cell $i$ calls arrive following independent Poisson process with corresponding intensities $\lambda_i, i = 1, \ldots, 7$. Upon arrival of a call at cell $i, i \neq 1$ at time $t$, it is accepted if there is still at least one channel available, that is, if the current state satisfies:

$$n_1(t^-) + n_i(t^-) + \max\{n_{i+1}(t^-) + n_{i-1}(t^-)\} \leq C - 1$$

and then $n_i(t) = n_i(t^-) + 1$, all other components of the state remaining unchanged. We say that at this time the call is connected. If an incoming call to cell $i, i \neq 1$ finds that the current state satisfies (1a) with equality, then all channels are used and the call is blocked.

Remark: For a call arriving at cell $i = 1$, a similar set of inequalities must be satisfied – we must include inequalities for each of the six cliques covering cell 1 rather than just the two cliques which cover each of the outer cells $i, i \neq 1$.

Calls stay connected for a random length of time called the “holding” time, assumed to be exponentially distributed and independent of the rest of the process history. All holding times are identically distributed with mean $1/\mu$. When a call using a channel on cell $i$ terminates, the corresponding occupancy component is decreased by one unit.

The process can be seen as a truncated multidimensional birth and death process for each component. When the process is in state $n \in S$, the birth rate of component $i$ is $\lambda_i 1_{\{n \not\in \mathcal{B}_i\}}$. 

where \(1_{\{A\}}\) is the indicator function of event \(A\) and the set \(B_i\) is the set of blocking states for cell \(i\), that is:

\[
B_i = \{ n \in \mathcal{S} : n_1 + n_i + \max(n_{i+1} + n_{i-1}) = C \}, \quad i = 2, \ldots, 7
\]

\[
B_1 = \bigcup_{i=2}^7 B_i. \tag{2}
\]

The death rate at cell \(i\) in state \(n \in \mathcal{S}\) is \(n_i \mu\). The reflecting boundary couples the different components through (1), as shown in Figure 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:

\[
B = \lim_{t \to \infty} \frac{\sum_{i=1}^K Y_i(t)}{A(t)}
\]
where \( Y_i(t) \) is the total number of calls lost in cell \( i \) up to time \( t \) and \( A(t) \) is the total number of arrivals up to time \( t \). The above limit is w.p.1, and its existence is assured by uniqueness of the invariant measure. Several equivalent expressions can be used for this performance measure \( B \). The process \( A(\cdot) \) is a Poisson process with aggregate rate \( \lambda = \sum_{i=1}^{K} \lambda_i \). As shown in Ross (1983), for any renewal process with finite rate, \( A(t)/t \to \lambda \) w.p.1, so that

\[
\lim_{t \to \infty} \left( \frac{Y_i(t)}{A(t)} \right) = \left( \frac{\lambda_i}{\lambda} \right)
\]

Using \( Y_i(t)/t \leq 1 \) w.p.1, dominated convergence now yields

\[
\lim_{t \to \infty} \frac{Y_i(t)}{t} = \lim_{t \to \infty} \frac{E[Y_i(t)]}{t} = \lim_{t \to \infty} E\left[ \frac{\lambda_i}{t} \int_0^t 1_{\{n(s) \in B_i\}} ds \right]
\]

where we have used the fact that the expected number of lost calls on cell \( i \) is the expected number of arrivals while in a blocking state for cell \( i \), and expected value of the number of Poisson(\( \lambda_i \)) arrivals on a given time interval of length \( s \) is \( \lambda_i s \). For a finite state birth and death process the long term limits coincide with stationary averages, that is,

\[
\pi(n) = \lim_{t \to \infty} \frac{1}{t} \int_0^t 1_{\{n(s) = n\}} ds, \quad n \in \mathcal{S}.
\]

Therefore:

\[
B_i = \lim_{t \to \infty} \frac{1}{t} \int_0^t 1_{\{n(s) \in B_i\}} ds = \sum_{n \in \mathcal{S}} \pi(n) 1_{\{n \in B_i\}}.
\]

and

\[
B = \sum_{i=1}^{K} \left( \frac{\lambda_i}{\lambda} \right) B_i
\]

**Remark:** The later expression can alternatively be deduced using PASTA, as explained in Ross (1993): Poisson arrivals find the process in steady state.

The Renewal Theorem can be used to re-write (4) in terms of expectations within regenerative cycles. Consider a process \( \{n(t); t > 0\} \) on a probability space \((\Omega, \mathcal{F}, P)\) with natural filtration \( \mathcal{F}_t \) containing the information of the history of the process up to time \( t \). A state \( n_0 \in \mathcal{S} \) is called a *regenerative point* of the process if there is an a.s. finite stopping time \( T_1 \) defined as the first entry time to state \( n_0 \) and such that the distribution of the process \( \{n(t+T_1); t > 0\} \) is independent of \( \mathcal{F}_{T_1} \) and it is identical to the distribution of the process \( \{n(t); t > 0\} \) with \( n(0) = n_0 \).

Because the channel occupancy process \( n(t) \) described above is an irreducible Markovian process on a finite state space, all states are regenerative points and a unique stationary measure \( \pi \) exists. For such systems, the Renewal Theorem (see Ross, 1983) asserts that:

\[
B_i = \frac{E[X_i(T)]}{E(T)}, \quad X_i(t) = \int_0^t 1_{\{n(s) \in B_i\}} ds
\]

where \( T \) is the first cycle length.
When dealing with distributed systems with several state components, it is sometimes impractical to use regeneration points of the process. Not only are regeneration cycles usually extremely long, but local processors using only available local information may not necessarily know the state of the process everywhere. If estimation of blocking probabilities were to be made on-line, for example, large networks would require each cell to know the occupancy of every other cell in the network, which is not a realistic assumption. The concept of quasi-regeneration was introduced to calculate stationary averages for such systems, as explained in Gaivoronski and Messina (1996) and Chang, Heidelberger, and Shahabiddin (1995). Assume the process starts with the stationary distribution \( P_{n(0)} = \pi(n) \). Fix a cell \( i \neq 1 \) and let \( T^{(i)} \) be the time of the first arrival at cell \( i \) that finds all channels free \( (n_i = 0) \). Then consecutive transitions from the set \( \{n_i = 0\} \) to the set \( \{n_i = 1\} \) (called the quasi-regeneration points) constitute the starting and terminating points of what is called an “A-cycle.” Because the process is assumed stationary, and arrivals are Poisson, the state at the quasi-regeneration points possesses the stationary distribution conditioned on \( n_i = 0 \). Although consecutive A-cycles have the same distribution, they may not be independent. It can be shown using the theory of stationary processes in (Breiman, 1992) that

\[
B_i = \frac{E[X_i(T^{(i)})]}{E(T^{(i)})},
\]

where \( X_i(T^{(i)}) \) is the amount of time within an A-cycle that the process spends in \( B_i \).

### 3 Simulation Methods

Evaluating blocking probabilities using (4) is a difficult numerical problem. Everitt and Macfadyen (1983) propose a methodology that identifies geometrical structures called cliques within the network and characterizes the constraint sets in terms of the geometry of the cells. This is used to calculate the normalization factor \( G(C) \), where the sums have to respect (1). This method for calculating blocking probabilities is computationally prohibitive when the number of cells is large, which means that simulation is often superior.

To approximate the value of \( B \), simulation can be used to produce a sample of random variables \( Y_s, s = 1, \ldots, S \) whose sample average is consistent for \( B \), namely,

\[
E[\hat{Y}(S)] = \frac{1}{S} \sum_{s=1}^{S} E[Y_s] \rightarrow B
\]

as \( S \rightarrow \infty \). In some cases \( \hat{Y}(S) \) is unbiased, that is, \( E[\hat{Y}(S)] = B \). When consecutive samples come from independent replicas of a simulation, the variance of \( \hat{Y}(S) \) can easily be estimated. Under some conditions on the boundedness of this variance, Alexopoulos and Seila (1998) show that the Central Limit Theorem (CLT) holds:

\[
\frac{\hat{Y}(S) - B}{\text{Var}[\hat{Y}(S)]} \overset{d}{\rightarrow} N(0, 1)
\]

where \( \overset{d}{\rightarrow} \) represents convergence in distribution and \( N(0, 1) \) is the standard normal distribution. Confidence intervals (see Larson, 1973) can then be approximated using:

\[
\lim_{S \rightarrow \infty} P\left\{ B \notin \hat{Y}(S) \pm z_{1-\alpha/2} \sqrt{\text{Var}[\hat{Y}(S)]} \right\} \leq \alpha,
\]
where \( z_{1-\alpha/2} \) is the upper \((1 - \alpha/2)\) quantile of the normal distribution. In order to achieve a prespecified precision within a confidence level \( \alpha \), \( S \) must be large enough so that \( \text{Var}[\hat{Y}(S)] \) be within the specified limits. Some estimators have less variance than others, yet they may require extremely long simulation times.

**Definition 1** The efficiency of a consistent estimator \( \hat{Y}(S) \) is:

\[
\mathcal{E}(\hat{Y}(S)) = \frac{1}{\text{CPU}[\hat{Y}(S)] \text{Var}[\hat{Y}(S)]},
\]

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

In particular, if the simulation has produced a sample of i.i.d. random variables, then the efficiency is independent of \( S \): CPU times tend to grow linearly with the sample size and \( \text{Var}[\hat{Y}(S)] \) decreases as \( 1/S \).

The foregoing discussion is useful for estimating confidence intervals when a prespecified tolerance error is given. When estimating probabilities, however, absolute errors may often be misleading and one requires estimating the probabilities within a given relative error, in terms of a percentage of the (unknown) probability. In these cases, the sample size must grow in order for

\[
\frac{\sqrt{\text{Var}[\hat{Y}(S)]}}{B}
\]

to remain at the required level.

**Definition 2** The relative efficiency of a consistent estimator \( \hat{Y}(S) \) is:

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

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 \( \lim_{S \to \infty} \mathcal{E}_r(\hat{Y}(S)) \), if this limit exists.

When estimating efficiencies, we shall often use the estimated values of \( B \) and \( \text{Var}[\hat{Y}(S)] \) in the definition of \( \mathcal{E}_r \). If the estimators are known not to be consistent, i.e., \( \lim_{S \to \infty} \text{E}[\hat{Y}(S)] \neq B \), then the actual definitions of efficiency use the mean square errors instead of the variances. In this paper we deal with consistent or unbiased estimation, so the problem of estimating the MSE is not present.

Whenever each sample \( Y_s \) is unbiased, their mean \( Y(S) \) will also be unbiased. However, the variance of \( Y(S) \) may be difficult to estimate when the samples are highly correlated. We are interested in estimating blocking probabilities. The general simulation framework used is to simulate some underlying process \( \{X_k, k = 1, 2, \ldots\} \), such as a Markov Chain Monte Carlo (MCMC) process or the birth and death process of Section 2. Then it is possible to set \( Y_s \in \{0, 1\} \) to be an indicator of \( X_s \) being a blocking state. Some underlying processes introduce considerable correlation, so that from a point when \( Y_s = 0 \), it may take a long time \( t \) before an observation yields a non-zero value for \( Y_{s+t} \). In order to calculate a variance of \( Y(S) \)
in the presence of correlation, the batch means method of Alexopoulos and Seila (1998) can be used. It is based on regrouping $\beta$ consecutive samples to form a “batch mean” $\bar{Y}_s$, $s = 1, \ldots, S$ and simulating a total of $S$ batches. The idea is that if $\beta$ is large enough, then consecutive batch means will be approximately independent and the CLT is applied to these quantities. An extensive study on the choice of $\beta, S$ for accurate estimation of the confidence interval is provided in Alexopoulos and Seila (1998) and suggests that choosing $\beta \approx S$ may be optimal.

To alleviate the problem of high correlations of consecutive values, periodic sampling of the sequence $Y_s$ could — in principle — be used to reduce the variance of the estimation. Let $t$ denote the sampling period. Then if $\{Y_s, s = 1, 2, \ldots\}$ are unbiased, the average of the observations $\{Y_{k+t}, k = 1, 2, \ldots\}$ is also unbiased, yielding the estimator:

$$
\bar{Y}_s = \frac{1}{\beta} \sum_{k=1}^{\beta} Y_{s+kt} \\
\bar{Y}(S) = \frac{1}{S} \sum_{s=1}^{S} \bar{Y}_s \\
\hat{V}(S) = \frac{1}{S-1} \sum_{s=1}^{S} (\bar{Y}_s - \bar{Y}(S))^2,
$$

where $\hat{V}(S)$ is the estimated variance of a single batch mean, $\bar{Y}_s$, derived from $S$ batches.

For $S \geq 10$ the normal approximation gives the approximate confidence interval:

$$
CI: \bar{Y}(S) \pm z_{1-\alpha/2} \sqrt{\frac{\hat{V}(S)}{S}}.
$$

As $t$ grows, it may suffice to consider smaller values of $\beta$, as the samples themselves become less correlated.

In this paper we will compare several simulation methods for estimating $B$, in terms of their efficiencies.

4 Estimation of Stationary Probabilities

This section presents three methods for estimating blocking probabilities by means of (4). The methods focus on how to simulate the stationary distribution $\pi(n)$ in (3), using the Markov Chain Monte Carlo (MCMC) methodology.

4.1 Acceptance/Rejection Method

The method of acceptance/rejection explained in Ross (1997) and Bratley, Fox, and Schrage (1987) is used for generating random variables. In the case of truncated distributions, it is a natural method of generating a random variable $n$ with the stationary probability $\pi(n)$ of (3). The method follows the recursion:
Algorithm 1:

1. Repeat
   
   (a) Generate $M_1, \ldots, M_K$ independent variables with $M_i \sim \text{Poisson}(\rho_i)$.
   
   Until $N = (M_1, \ldots, M_K) \in \mathcal{S}$

2. $X_s = N$

3. $s \leftarrow s + 1$, go to 1.

The resulting random variables $\{X_s\}$ have distribution $\pi$ in (3). To see this, notice that $X_s$ is a variable having the conditional distribution that the value of $N$ is accepted, that is,

\[
\mathbb{P}[X_s = n] = \mathbb{P}[N = n | N \in \mathcal{S}] = \frac{\mathbb{P}[N = n]}{\mathbb{P}[N \in \mathcal{S}]} = \frac{1}{\mathbb{P}[N \in \mathcal{S}]} \prod_{i=1}^{K} \left( \frac{\binom{n_i}{n_i!}}{\rho_i} \right)
\]

which gives (3), since $G(C) = \mathbb{P}[N \in \mathcal{S}]$.

This method is used in Everitt and Macfayden (1983) and Yates (1997) to calculate (4) as follows. Random variables $\{X_s\}$ are generated according to Algorithm 1. Next, compute:

\[
Y_s = \sum_{i=1}^{K} \left( \frac{\lambda_i}{\lambda} \right) 1_{[X_s \in \mathcal{B}_i]}
\]

which identifies for which cells the state $X_s$ is in a blocking state and weights the probability accordingly. By definition, $\mathcal{B}_i \cap \mathcal{B}_j \neq \emptyset$ for $i \neq j$, and this overlapping means that some states may block several adjacent cells. By construction, $Y_s$ is an unbiased estimator of the blocking probability: $\mathbb{E}[Y_s] = B$. Clearly the samples $Y_s, s = 1, \ldots, S$ are independent, and the CLT can be used to build confidence intervals (9) for the estimation.

Generating $K$ independent Poisson random variables can be made very fast by means of pre-calculated tables. However, the acceptance probability $G(C)$ can be very low for moderately large networks. In previous studies we have used as many as two million iterations of the algorithm, on average, before obtaining a sample in the feasible set $\mathcal{S}$. To visualize why this happens, imagine the feasible region projected on $(n_i, n_{i+1}, n_{i-1})$ as shown in Figure 3. For each value of $n_1 \leq C$ a tetrahedron contained in the hypercube of side length $C$ symbolizes the feasible region on the projected coordinates. Clearly other coordinates will be also constrained by the values of $n_{i-1}$ and $n_{i+1}$. Even if a pre-calculated table is used to generate Poisson random variables truncated to the hypercube, that is, $M_i \sim \text{Poisson}(\rho_i) | C$, $M_i \leq C$ a.s., the volume of the feasible region can be considerably smaller than the whole space.

4.2 Metropolis-Hastings

The computational inefficiency of the acceptance/rejection method for generating $X_k$ is mainly due to the fact that most of the generated variables lie outside the region $\mathcal{S}$. An algorithm based on the Metropolis Hastings method is built here. It defines a Markov chain that evolves within the state space $\mathcal{S}$ and whose stationary probabilities are the probabilities $\pi$ given by (3).
For each state \( n \in S \) define a *neighborhood* \( N(n) \subset S \). The requirement of the neighborhoods is that for every \( m_0, m_f \in S \) there exists a finite sequence of states \( m_1, m_2, \ldots, m_p = m_f \) such that \( m_k \in N(m_{k-1}) \). In addition, if \( n \in N(m) \) then \( m \in N(n) \). The name “neighborhood” therefore does not necessarily mean that the elements within are adjacent to each other.

To generate the successive values of the process \( \{X_n\} \) use:

**Algorithm 2:**

1. Set \( n = X_k \),
2. Choose \( m \in N(n) \) uniformly,
3. Generate \( U \sim U[0,1] \) and define the next state:
   \[
   X_{k+1} = \begin{cases} 
   m & \text{if } U < \alpha(n,m) \\
   n & \text{otherwise}
   \end{cases}
   \]
4. \( k \leftarrow k + 1 \), go to 1.

where:
\[
\alpha(n,m) = \min \left( 1, \frac{\pi(m)}{\pi(n)} \frac{\|N(n)\|}{\|N(m)\|} \right), \quad m \in N(n)
\]
and \( \|N(n)\| \) is the number of elements in the neighborhood of \( n \).

Then the Markov chain \( \{X_k, k \geq 1\} \) has stationary probabilities \( \pi \) given by (3). To see this, first notice that the chain is irreducible by the requirement on the neighborhoods: all states communicate. Since it is a finite state chain, all the states are positive recurrent. Using the reversibility theorem of Ross (1993), the process is reversible with stationary probabilities \( \pi^* \) if and only if
\[
\pi^*(n)P_{n,m} = \pi^*(m)P_{m,n}
\]
where \( P_{n,m} = \text{P}\left\{X_{k+1} = m|X_k = n\right\}\) is the transition matrix. Proving (10) with \( \pi^* = \pi \) is not hard. Let \( m \in N(n) \) be any element of the neighborhood of \( n \). Then by construction

\[
P_{n,m} = \begin{cases} 
\frac{1}{\|N(n)\|} \alpha(n,m) & \text{if } m \neq n \\
\frac{1}{\|N(n)\|} \alpha(n,n) + \frac{1}{\|N(n)\|} \sum_{k \neq n} (1 - \alpha(n,k)) & \text{if } m = n.
\end{cases}
\]

If \( m \neq n \) in (10), assume w.l.o.g. that \( \alpha(n,m) < 1 \) and notice that this implies that \( \alpha(m,n) = 1 \). Then

\[
\pi(n)P_{n,m} = \pi(n) \frac{1}{\|N(n)\|} \times \frac{\pi(m)}{\|N(m)\|}
\]

and

\[
\pi(m)P_{m,n} = \pi(m) \frac{1}{\|N(m)\|}
\]

so that (10) is satisfied and the claim that the stationary probabilities are indeed \( \pi(n) \) is verified.

Using the same arguments as for the acceptance/rejection method, if

\[
Y_s = \sum_{i=1}^{K} \left( \frac{\lambda_i}{\lambda} \right) 1_{\{X_s \in B_i\}}
\]

then \( \lim_{s \to \infty} \text{E}[Y_s] = B \). Typically, one simulates the chain \( \{X_s\} \) for a “warm-up” initial stage of \( k_0 \) transitions: \( X_{k_0}, \ldots, X_{-1} \). The assumption is that from this time on, the distribution of the chain is nearly stationary. Once \( \{X_s, s \geq 0\} \) is assumed to be a stationary process, \( Y_s \) will be unbiased estimates of \( B \). Samples from the process \( \{X_s, s \geq 0\} \) can thus be used to estimate \( B \), using sample averages of the observations \( Y_s \). The high correlation between states in the Metropolis-Hastings algorithm (depending on how the neighborhood system is defined) makes the use of batch means essential.

Implementation of our Metropolis-Hastings algorithm still leaves the choice of the neighborhoods \( N(n), n \in S \), which significantly influences the efficiency of the resulting estimator. The choice of the sampling period \( t \) may also have some impact.

### 4.3 The Gibbs Sampler

The general form of Metropolis Hastings prescribes sampling from a general distribution \( q(M|X_k), \quad M = (M_1, \ldots, M_K) \) in step 2 of Algorithm 2. In that example, \( q(\cdot|X_k) \) is the uniform distribution in the neighborhood \( N(X_k) \). A sequential version of the algorithm can be implemented as explained in Gilks, Richardson, and Spiegelhalter (1996): when calculating \( X_{k+1} \) from a value \( X_k \), the updates are performed one component at a time. Let

\[
X_k^{j} = (X_{k+1}(1), \ldots, X_{k+1}(j-1), X_k(j+1), \ldots, X_k(K))
\]

be the current vector at the \( j \)-th stage of iteration \( k \), when component \( j = 1, \ldots, K \) is to be updated: all previous components \( i < j \) have their new value while all components with \( i > j \) are yet to be updated. The general description is to draw a candidate \( m_j \in \mathbb{N} \) from the conditional distribution \( q_j(\cdot|X_k^{j}) \), and set the acceptance probability to:

\[
\alpha(m_j, n_j, X_k^{j}) = \min \left( \frac{\pi(m_j|X_k^{j}) q_j(n_j|X_k^{j})}{\pi(n_j|X_k^{j}) q_j(m_j|X_k^{j})} \right),
\]
where \( n_j = X_k(j) \in \mathbb{N} \) is the current value of the \( j \)-th component to be updated. Once the \( K \) stages of the \( k \)-th iteration are finished, the resulting value \( X_k+1 \) is the next state of the Markov Chain.

When it is not difficult to generate samples from the conditional target distribution \( \pi(\cdot | X_k^{-j}) \), it possible to use that as the candidate distribution, that is,

\[
q_j(m|X_k^{-j}) = \pi(m|X_k^{-j}).
\]

Thus the acceptance probability becomes \( \alpha = 1 \), and the resulting algorithm is called the Gibbs sampler.

**Algorithm 3:**

1. For each \( j = 1, \ldots, K \) do:
   
   (a) Generate \( X_{k+1}(j) \sim \pi(\cdot | X_k^{-j}) \)

2. \( k \leftarrow k + 1 \), go to 1.

Due to the fixed order in which the updates takes place for the \( K \) stages of each iteration, the chain is no longer reversible, as explained in Brémaud (1999). In order to see that the ensuing process \( \{X_k\} \) also possesses stationary distribution \( \pi \), argue as follows. Irreducibility of the finite state chain ensures the existence of a unique ergodic measure for this example. Suppose now that \( X_k \sim \pi(\cdot) \) has the target distribution \( \pi \) given in (3). Then by construction of the vector \( X_{k+1} \) according to Algorithm 3, it follows that:

\[
P[X_{k+1} = m] = \sum_{n \in S} \pi(n) P[X_{k+1} = m | X_k = n] = \sum_{n \in S} \pi(n) \prod_{j=1}^{K} P[X_{k+1}(j) = m_j | X_k^{-j} = (m_1, \ldots, m_{j-1}, n_j+1, \ldots, n_K)]
\]

which can be shown by proving that the sum above is proportional to \( \hat{\beta}_k^m / m_! \). This result establishes that the probabilities (3) are stationary for the kernel of the Markov Chain \( \{X_k\} \).

Once again, if

\[
Y_s = \sum_{i=1}^{K} \left( \frac{\lambda_i}{\lambda} \right) 1_{\{X_s \in B_i\}}
\]

then \( \lim_{s \to \infty} E[Y_s] = B \). Once the chain \( \{X_s\} \) has been simulated for a “warm-up” stage, \( X_{-K_0}, \ldots, X_{-1} \), the distribution of the chain is nearly stationary. Sample averages of \( \{Y_s, s \geq 0\} \) can then be used to derive a confidence interval for \( B \), using (9).

### 4.4 Numerical Results

In order to evaluate the methods described above, numerical experiments were carried out to determine the relative efficiency using \( S = 1000 \) samples, \( \mathcal{E}_r(\hat{Y}(S)) \). In all cases, a batch size of \( \beta = 1000 \) was used, to facilitate comparison. These experiments were conducted on a 266 MHz Pentium II processor using the gnu C++ compiler under the Linux operating system.
Table 1: Average number of candidate states \( N \) rejected per accepted state.

<table>
<thead>
<tr>
<th>Load per cell, ( \rho_i )</th>
<th>7 cells</th>
<th>19 cells</th>
<th>37 cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0.05</td>
<td>0.19</td>
<td>0.5</td>
</tr>
<tr>
<td>13</td>
<td>0.17</td>
<td>0.7</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>0.42</td>
<td>2.4</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>1.1</td>
<td>11</td>
<td>167</td>
</tr>
<tr>
<td>16</td>
<td>4.2</td>
<td>69</td>
<td>7000</td>
</tr>
<tr>
<td>17</td>
<td>6.5</td>
<td>789</td>
<td>9.2e05</td>
</tr>
<tr>
<td>18</td>
<td>17</td>
<td>16660</td>
<td>5.3e08</td>
</tr>
</tbody>
</table>

In the simulations that follow, the arrival rate was the same over all of the cells, so that \( \rho_i \equiv \rho \) is constant. The clique limit was \( C = 50 \), typical of Advanced Mobile Phone System (AMPS) networks (Lee, 1995).

**Acceptance/rejection:** For efficiency, an early rejection approach was taken for the acceptance/rejection method. After each variable \( M_i \) was generated, a check was made that no clique constraints (1) were violated, and if so, the remaining \( M_j, j > i \), were not generated. Each clique was only checked once, when the occupancy of the last cell in that clique was generated. If the blocking is sufficiently high, it may be faster to check the occupancy of the clique every time a new cell in the clique is added to the state vector, but this will seldom help for realistic blocking rates (1% to 10%).

Table 1 shows the average number of candidate states \( N \) rejected by the acceptance/rejection method per accepted state. This clearly grows super-exponentially with the size of the grid, making the approach infeasible for realistic sized networks.

**Metropolis-Hastings:** The neighborhood function, \( N(n) \subset \mathcal{S} \), chosen for the Metropolis-Hastings algorithm corresponds to a single call arrival or departure:

\[
N(n) = \{ m \in \mathcal{S} : \| m - n \|_1 = 1 \},
\]

where \( \| \cdot \|_1 \) denotes the 1-norm. Because of the small changes between neighbors, the correlation between the blocking probability of successive states is high. However, the probabilities of successive states are also similar, which causes the chance that the update will be rejected at step 3 of Algorithm 2 to be low (around 0.1). This indicates that this neighborhood is slightly smaller than optimal, since the ideal trade-off in neighborhood size typically yields a rejection rate of 0.25 (Chib and Greenberg, 1995). Except for the tests to determine the impact of the sampling period, \( t \), a value of \( t = 7 \) was used. Because the stationary probabilities of the Metropolis-Hastings algorithm are known, albeit tedious to calculate, it is possible to avoid the need for a “warm-up” time by drawing the initial state from this distribution. This avoids the need to know how many initial iterations, \( k_0 \), to ignore. Except for the 37 cell network with load 18, in which calculating even a single sample by acceptance/rejection takes unacceptable time, this approach was used. For the 37 cell/load 18 case, a warm up time of \( k_0 = 10000 \) iterations was used, starting with \( X_{-k_0}(i) = 16, i = 1, \ldots, K \).

**Gibbs Sampler:** For this problem, the conditional distribution \( \pi(\cdot | X_k^{-j}) \) is a one-dimensional truncated Poisson random variable on \( \{0, \ldots, C_k(j)\} \), with:

\[
C_k(j) = C - n_1 - \max(n_{j-1}, n_{j+1}), \quad \text{with } (n_1, \ldots, n_{j-1}, n_{j+1}, \ldots, n_K) = X_k^{-j}, j \neq 1 \quad (11a)
\]
and
\[ C_k(1) = C - \max_i (n_i + n_{i+1}), \quad \text{with} \quad (n_2, \ldots, n_K) = X_k^{-1}. \] (11b)

Generating \( M \sim \pi(\cdot | X_k^{-j}) \) is not hard: at the start of the simulation, an array \( P \) is calculated with the cumulative probabilities of the Poisson distribution from 0 to \( C \), which is the largest possible value for all truncations:
\[
P_m(j) = \sum_{n=0}^{m} \frac{\rho_n^j}{n!} \times \left( \sum_{n=0}^{C} \frac{\rho_n^j}{n!} \right)^{-1}, \quad m = 1, \ldots, C.
\] (12)

If a Poisson random variable \( M \) truncated at \( c < C \) is to be generated, then the corresponding distribution satisfies \( P[M \leq m] = P_m(j) / P_c(j), m = 0, \ldots, c \) and can be calculated very efficiently by simple reading of the array \( P(j) \).

Step 1(a) of Algorithm 3 thus becomes

i) Set \( c = C_k(j) \) as in (11),

ii) Generate \( U \sim U[0,1] \) and set \( n_j = \min \{ m : P_m(j) \geq U P_c(j) \} \), using (12),

iii) Set \( X_{k+1}(j) = n_j \).

For initial state \( X_0 \) the empty state was chosen. Our experiments show fast convergence of the Markov Chain and the warm up period was always very short.

The relative efficiency of the acceptance/rejection, Metropolis-Hastings and Gibbs sampler methods on hexagonal grids of 7, 19 and 37 cells are shown in Figure 4 as a function of the load. As the blocking probability or size of the grid increases, the relative efficiency of the acceptance/rejection algorithm initially increases, but then degrades dramatically. The initial increase is due to the difficulty in estimating small blocking probabilities, which is the focus of Section 5. As the load decreases, the reduction in the number of observed blocking events degrades efficiency. The subsequence degradation is because the CPU time required is approximately inversely proportional to the probability that a state is feasible. For small blocking probability, this is approximately unity even for small networks, but as the blocking probability or network size increases, the probability decreases rapidly.

When the probability of a state being infeasible is negligible, the Gibbs sampler performs similarly to the acceptance/rejection method, since the only overhead required is determining the occupancy of the (six or fewer) cliques containing each newly generated cell. Consecutive samples are approximately independent since the truncation of the Poisson distribution to preclude infeasible states is negligible. As the blocking probability increases, the correlation between successive samples increases, but this effect is dominated by the variance reduction mentioned above.

Metropolis-Hastings avoids the cost of generating Poisson random variables, at the expense of keeping track of the occupancy of each clique for the calculation of \( ||N(n)|| \). The CPU time required by Metropolis-Hastings is thus less than the other two methods, but the substantial correlation between the blocking at successive states results in consistently lower relative efficiency than Gibbs sampling, and lower relative efficiency than acceptance/rejection when the probability of rejection is low. The performance could be improved greatly by careful selection of the neighborhood.
Figure 4: Relative efficiency vs. load for three algorithms and different grid sizes.
The number of iterations required by the acceptance/rejection method does not depend directly on the load, but rather on the probability that the candidate state $N$ is infeasible, and hence on the blocking probability. Figure 5 shows the relative efficiency of each estimator against the blocking probability, 7-, 19- and 37-cell grids. This figure also highlights the fact that the relative efficiency of the Metropolis-Hastings and Gibbs sampler methods are insensitive to the size of the network, while the performance of the acceptance/rejection method degrades rapidly as the network grows.

5 Fast Simulation

The first part of this paper dealt with systems where blocking probabilities are not very small. However, in third generation systems satisfying IMT-2000 requirements (Prasad and Ojanperä, 1998, ETSI, 1997), mobiles can connect to indoor “pico-cells” with very high data rates, con-
suming the entire allocated radio bandwidth. Thus only a single call can be accommodated in each clique, giving $C = 1$ and hence $\lambda_i < \mu$.

In addition to the problem that $G(C)$ may be very small and many samples must be discarded before one falls in the region $S$, the computational inefficiency of the acceptance/rejection method can be compounded by the problem of rare event estimation, explained in Asmussen and Rubinstein (1995) and Chang, Heidelberger, and Shahabiddin (1995). In the Markov Chain Monte Carlo (MCMC) methods seen before there is waste of simulated samples: many of the sample values that enter in the averaging are zero. The framework of rare event simulation is used to describe and qualify such “waste” and we now proceed to describe it briefly.

5.1 Preliminaries

Recall that the occupancy process $\{n(t); t \geq 0\}$ described in Section 2 is a continuous time Markov process on $(\Omega, \mathbb{P})$ referring to a truncated birth and death process with state-dependent reflecting boundaries. The arrival rate of calls into cell $i$ is $\lambda_i$, $i = 1, \ldots, K$, and the average holding time per call is $1/\mu$.

Fix any cell $i$ and consider expression (6) of the blocking probability using $A$-cycles. The start of an $A$-cycle corresponds to the first arrival at cell $i$ that finds all channels free, and $T^{(i)}$ is the duration of an $A$-cycle, which is adapted to the natural filtration of the process $\{n(t); t > 0\}$; that is, knowledge of the process history up to time $t$ is sufficient to determine if the $A$-cycle terminates at this epoch, and note that $T^{(i)}$ must necessarily coincide with an arrival to one of the cliques where cell $i$ belongs.

Define now $\tau_i$ to be the stopping time when blocking states are reached and $\mathcal{R}_i$ to be the event that the occupancy process enters the blocking set $\mathcal{B}_i$ before the cycle is over:

$$
\tau_i = \min\{k : S_k \leq T^{(i)} \text{ and } n(S_k) \in \mathcal{B}_i\}, \quad \text{and} \quad \mathcal{R}_i = \{S_{\tau_i} < T^{(i)}\}. \quad (13)
$$

Equation (6) requires estimation of the proportion of time that the process spends in blocking states. For all $\omega$ with $1_{\{\mathcal{R}_i\}}(\omega) = 0$, $X_i(T^{(i)})(\omega) = 0$, thus the numerator in (6) can be estimated with:

$$
\mathbb{E}[X_i(T^{(i)})] = \mathbb{E}\left(\mathbb{E}[X_i(T^{(i)})|\mathcal{R}_i]1_{\{\mathcal{R}_i\}}\right).
$$

Definition 3 Let $C$ be a parameter of the distribution of the process on $(\Omega, \{\mathcal{B}_k, k \geq 0\}, \mathbb{P})$ and let $P_C$ denote the corresponding distribution. The event $\mathcal{R}_i \subset \Omega$ is called a rare event if

$$
\lim_{C \to \infty} P_C(\mathcal{R}_i) = 0.
$$

Call $p(C) = P(\mathcal{R}_i)$, where $C$ is the maximum number of available channels in the cellular network problem. Then $\mathcal{R}_i$ is a rare event if we assume that $\lambda_i; i = 1, \ldots, K$ and $\mu$ are independent of $C$, which we shall do throughout the rest of the paper. If $p(C)$ is estimated via simulation using $1_{\{\mathcal{R}_i\}}$ for $S$ consecutive $A$-cycles, then the variance of the estimator is at least $p(C)(1 - p(C))/S$ because consecutive $A$-cycles introduce positive correlation on the Bernoulli sequence. The relative error in the estimation is bounded below by:

$$
\frac{\sqrt{p(C)(1 - p(C))/S}}{\hat{p}(C)} \approx \sqrt{\frac{1 - p(C)}{Sp(C)}} \to \infty
$$
as $C \to \infty$. Larger and larger sample sizes $S$ must be used to obtain the same accuracy in
the estimation for different parameter values. The efficiency in the estimation of rare events
can be improved using a change of measure approach via Importance Sampling (see Bratley,
among others). This improvement is due to two properties of the IS estimator: smaller required
sample sizes and smaller variances, as we shall discuss now.

The general framework for Importance Sampling will be stated in terms of the Discrete
Event formulation of the process, which is a natural description for simulation. The formulation
is not unique; two different ones will be considered in Sections 5.2 and 5.3.

Assume an underlying Markovian process $\{U_k\}$ with natural filtration $\mathcal{F}_k = \sigma(U_1, \ldots, U_k)$,
such that the occupancy process $\{n(t)\}$ can be imbedded into a discrete event process adapted
to the filtration $\{\mathcal{F}_k, k = 1, 2, \ldots\}$. Use the notation $\{n(k), k = 1, 2, \ldots\}$ for the imbedded
process (with the obvious abuse in notation).

**Example:** Discrete event simulation is commonly performed by generating, for each cell, the
inter-arrival times $A_k(i), k = 1, 2, \ldots$ according to a Poisson process with rate $\lambda_i$, and the
holding times $H_k(i), k = 1, 2, \ldots$ as independent exponential variables with mean $1/\mu$. At the
arrival epochs to cell $i$, given by the expression:

$$S_t(i) = \sum_{k=1}^{t} A_k(i),$$

the occupancy process $n_i$ increases by one, provided $n \notin B_i$. At the call termination epochs
$S_t(i) + H_k(i)$ the corresponding component decreases by one. Identify $U_k(i) = (A_k(i), H_k(i))$.
Then the imbedded process is adapted to the filtration $\mathcal{F}_k = \sigma(U_1(i), \ldots, U_k(i); i = 1, \ldots, K)$,
where $k(i)$ is the number of arrivals occurring at cell $i$ amongst the first $k$ total arrivals.

The general structure of the evolution of the Markovian process $\{U_k\}$ is assumed to satisfy:
$P[U_{k+1} \in A|\mathcal{F}_k] = P[U_{k+1} \in A|n(k)]$. In the example above, $U_{k+1}$ is actually independent of
the state $n(k)$; the more general formulation will prove useful for the Standard Clock technique
in Section 5.3. Let $f_n(u)$ be the conditional density of $U_{k+1}$ given the state $n(k) = n$ (the case
where $U_k$ has discrete components can be treated in a similar way). Then for any real valued
functional of the process $\Psi(U_1, U_2, \ldots)$ with finite expectation, and any finite integer $T$,

$$E[\Psi(U_1, \ldots, U_T)] = \int \Psi(u_1, \ldots, u_k) \left[ \prod_{k=0}^{T-1} f_n(k)(u_{k+1}) \right] \, du_1, \ldots, du_k, \quad n_0 : \text{initial state.}$$

Let now $f_n^*(\cdot)$ be another conditional density. Multiplying and dividing each term in the product
above by the new densities, we obtain:

$$E[\Psi(U_1, \ldots, U_T)] = \int \Psi(u_1, \ldots, u_k) \left[ \prod_{k=0}^{T-1} \frac{f_n(k)(u_{k+1})}{f_n^*(k)(u_{k+1})} \right] \left[ \prod_{k=0}^{T-1} f_n^*(k)(u_{k+1}) \right] \, du_1, \ldots, du_k$$

$$= E^*[\Psi(U_1, \ldots, U_T) L(U_1, \ldots, U_T)];$$

$$L(U_1, \ldots, U_T) = \prod_{k=0}^{T-1} \frac{f_n(k)(U_{k+1})}{f_n^*(k)(U_{k+1})} \quad (14)$$

19
where $E^*$ denotes the expectation w.r.t. a different probability: here the process $\{U_k\}$ evolves according to the transition densities $f^*$. When $L_k$ is interpreted as a statistics depending on the observations, it is called the “likelihood ratio”. The above expression is valid for general functionals only when the support of the old distributions is contained within that of the new distributions, i.e., if for every $x > 0$, $f_n(u) > 0$ implies $f^*_n(u) > 0$ for the corresponding densities. In such cases one says that the new measure $P^*$ is absolutely continuous w.r.t. $P$.

**Remark:** Let $L_k = L(U_1, \ldots, U_k)$, then the random process $\{L_k, k = 1, 2, \ldots\}$ is what is known as a (multiplicative) $P^*$-martingale adapted to the filtration, because it satisfies $E^*[L_{k+1}|\mathcal{F}_k] = L_k$. This martingale defines the appropriate Radon-Nykodim derivative for functionals of the process $\{U_k\}$, and the change of measure (14) is valid also when $T$ is a stopping time, as explained in detail in Asmussen and Nielsen (1995).

In the case of rare event estimation for the cell blocking probability, we will identify $T = \tau_i$ and $\Psi = 1_{\{R_i\}}$. In this case the change of measure approach can be specialized to the “importance” region and (14) can be used for densities such that $P^*(R_i) \gg p(i)$ with $L_{\tau_i} 1_{\{R_i\}} \leq 1$ w.p.1. It is in this sense that the estimation can be performed faster and with variance reduction. Indeed, the variance of the estimator without IS is $\text{Var}(1_{\{R_i\}}) = p(i)[1 - p(i)]$, but under the new measure,

$$\text{Var}^*(L_{\tau_i} 1_{\{R_i\}}) = E^*[L_{\tau_i}^2 1_{\{R_i\}}] - p^2(i) = E[L_{\tau_i} 1_{\{R_i\}}] - p^2(i) \leq p(i) - p^2(i),$$

because (14) can be applied to $\Psi = L_{\tau_i}$ and we have assumed that $L_{\tau_i} 1_{\{R_i\}} \leq 1$ w.p.1.

Ideally, we are not interested in variance reduction but in asymptotically efficient estimators, which is not always possible to build. It is shown in Chang, Heidelberger, and Shahabidin (1995) that $L_{\tau_i} 1_{\{R_i\}}$ is asymptotically efficient for $p(i)$ if there are constants $d_1, d_2, \alpha > 0$ such that

$$d_1 e^{-\alpha} \leq L_{\tau_i} 1_{\{R_i\}} \leq d_2 e^{-\alpha}. \quad (15)$$

In Section 5.2 we implement the change of measure that swaps arrival and service rates (which is commonly used in queueing systems) and show that it is asymptotically efficient, generalizing the above result.

### 5.2 Accelerating Cluster Processes

Fix a cell $i \neq 1$ to estimate $P[R_i]$. Suppose that we consider “accelerating” the input rate at all cells $j$ in the “cluster”:

$$C_i = \{1, i - 1, i, i + 1\}$$
which contains all the cells belonging to cliques where \( i \) belongs. In addition, suppose that the simulation is performed using holding time distributions with higher expectation. Under the new measure, blocking will certainly be more likely to occur, \( P^* (R_i) \approx 1 \) and the fraction of wasted samples will decrease. In this section we will determine an asymptotically efficient change of measure by determining the arrival rates and mean holding times for the change of measure.

Use uniformization of the process, as explained in Ross (1993): the aggregate arrival process at this cluster of surrounding cells is Poisson with intensity \( \lambda \equiv \lambda_1 + \lambda_i + \lambda_{i-1} + \lambda_{i+1} \). The original process is modeled now using this arrival process to generate the inter-arrival times \( A_k(i) \sim \exp(\lambda) \). Under this relabeling, \( S_k(i) \) is the epoch of the \( k \)-th local arrival at cluster \( C_i \). Next, at arrival epochs \( S_k(i) \), the cell to which the arrival to the cluster is assigned is chosen as \( I_k = j \) with probability \( \lambda_j / \lambda \). Holding times \( H_k(i) \sim \exp(\mu) \) are i.i.d. exponential, as before. These quantities correspond to the actual holding times of the \( k \)-th call to be connected within the cluster, regardless of the particular cell \( j \in C_i \) that it happens to belong. If the \( k \)-th arriving call to the cluster is blocked we set \( H_k(i) = 0 \) by convention (the call is not connected).

For this simulation framework, set \( U_k(i) = (A_k(i), H_k(i), D_k(i)) \) where \( D_k(i) \in C_i \) is the cell where the arrival occurs within the cluster. Let \( U_k(j) = (A_k(j), H_k(j)) \) be the inter-arrival and holding times for cells \( j \not\in C_i \). The natural filtration is given by:

\[
\mathcal{F}_k = \sigma(U_1(i), \ldots, U_k(i)); U_1(j), \ldots, U_k(j); j \not\in C_i); \quad k = 1, 2, \ldots
\]  

where exactly \( k(i) \) arrivals have occurred within the cluster \( C_i \) and \( k(j) \) arrivals at cell \( j \not\in C_i \) at the epoch of the \( k \)-th global arrival. The imbedded occupancy process \( \{n(k)\} \) can be evauated from the information in \( \mathcal{F}_k \) and is therefore adapted to the filtration (16), as required.

The new measure is defined through a process where the arrival rate at the cluster is \( \lambda^* \) and the mean holding time of calls within that cluster is \( \mu^* \), whilst preserving the shape of the distributions as being independent and exponential. Other cells retain their original distributions for arrivals and holding times.

Then the corresponding likelihood ratio, using (14) is:

\[
L_T = \left( \frac{\lambda}{\lambda^*} \times \frac{\mu}{\mu^*} \right)^T \exp \left[ - (\lambda_i - \lambda_i^*) S_T(i) - (\mu_i - \mu_i^*) \sum_{k=1}^{T} H_k(i) \right].
\]

A common change of measure in these models is to “swap” rates, which sets \( \lambda^* = \mu, \mu^* = \lambda \). The following argument holds only when \( \lambda < \mu \). Otherwise, single cell acceleration is preferred. Under this particular change of measure, if we stop the process at \( \tau_i \), we obtain:

\[
L_{\tau_i} = \exp \left[ (\mu - \lambda) \left( S_{\tau_i}(i) - \sum_{k=1}^{\tau_i} H_k(i) \right) \right].
\]  

We will now bound \( L_{\tau_i}(\omega) \) for \( \omega \in \Omega \) such that \( 1_{\{\tau_i\}}(\omega) = 1 \). The time \( S_{\tau_i} \) corresponds to the arrival epoch of the call that enters one of the blocking states. Because \( B_i \) is reached as soon as either \( n_1(\tau_i) + n_i(\tau_i) + n_{i-1}(\tau_i) = C \) or \( n_1(\tau_i) + n_i(\tau_i) + n_{i+1}(\tau_i) = C \), it follows that the arrival at time \( S_{\tau_i}(i) \) finds at least \( C \) on-going calls within the cluster. In Figure 6(a) the fourth arrival finds two channels occupied in the cluster, while the seventh finds four connected calls. If instead of connecting calls immediately, they were placed in an infinite buffer queue.
to wait for sequential termination, then the quantities $\sum_{k=1}^t H_k(i), t \leq \tau_i$ would be the call termination epochs. In Figure 6(b) we depict this situation, where the calls wait from arrival and then upon termination of the previous call, they are connected and keep the same holding time as in the original process in Figure 6(a). In the picture all calls have terminated within the time window shown, while in the queueing system one call is still connected and four are waiting for connection.

Figure 6: Pathwise Bound for the Likelihood Ratio

This implies stochastic domination of the process by a FCFS queueing: the $\tau_i$-th arrival in this queue would find at least $C$ waiting and connected calls, so that on the set where blocking of an arrival at $i$ occurs before the end of the $A$-cycle, that is $\{\omega : 1_{\{\tau_i\}}(\omega) = 1\},$

$$S_{\tau_i} - \sum_{k=1}^{\tau_i} H_k(i) \leq - \sum_{k=\tau_i - C}^{\tau_i} H_k(i) < 0, \quad P^* - a.s.$$ and the Likelihood ratio is bounded by 1, ensuring variance reduction, that is:

$$L_{\tau_i} 1_{\{\tau_i\}} < \exp\left\{-\left(\mu - \lambda\right) \sum_{k=\tau_i - C}^{\tau_i} H_k(i)\right\}.$$
Clearly this bound is not equivalent to (15), because the exponential random variables do not have a finite support away from zero (they are not uniformly bounded and away from zero). We now show that this change of measure is asymptotically efficient for \( p(C) \), that is, there is a constant \( b < \infty \) and an integer \( C_0 < \infty \) such that for every \( C \geq C_0 > 0 \):

\[
\frac{\text{Var}^*[L_{r_1}, 1_{\{R_i\}}]}{p(C)} \leq b.
\]

Since \( \text{Var}^*[L_{r_1}, 1_{\{R_i\}}] = E[L_{r_1}, 1_{\{R_i\}}] - p^2(C) \), we use conditioning to express \( E[L_{r_1}, 1_{\{R_i\}}] = E[L_{r_1}|R_i]p(C) \), which means that asymptotic efficiency is equivalent to showing:

\[
E[L_{r_1}|R_i] - p(C) \leq b.
\] (18)

First use \( p(C) = E^*[L_{r_1}, 1_{\{R_i\}}] \) to bound:

\[
L_{r_1} \geq \exp \left\{ - (\mu - \lambda) \sum_{k=1}^{\tau_i} H_k(i) \right\}.
\]

Wald’s equation yields \( E^* \left[ \sum_{k=1}^{\tau_i} H_k(i) \right] = E^*[\tau_i]/\lambda \), as explained in Ross (1983), because under \( P^* \) the random variables \( H_k(i) \) are exponential with rate \( \lambda \). Using the gambler’s ruin problem of Ross (1993) it is possible to bound \( E^*[\tau_i] \leq \rho C \) for large \( C \). Now with Jensen’s inequality, for large \( C \), \( p(C) \geq e^{-(\mu - \lambda) \rho C / \lambda} \). Showing (18) now requires a truncation argument. Recall that under the measure \( P \), \( H_k(i) \) are exponential with rate \( \mu \). Define the truncated random variables \( H'_k(i) = \max(\delta, H_k(i)) \) for small \( \delta > 0 \) and notice that for any \( C \), \( P \{ H_k(i) \neq H'_k(i), k = 1, \ldots, C \} = (1 - e^{-\delta})^C \to 0 \) as \( C \to \infty \). This implies that the truncated process will satisfy:

\[
P \{ \omega : L_{r_1}(\omega) \neq L'_{r_1}(\omega) \} \to 0.
\]

By construction, the truncated process satisfies the bound \( L'_{r_1} \leq e^{-(\mu - \lambda)\delta C} \) on the set \( R'_i \), so there is \( C_0 \) such that \( b = e^{-(\mu - \lambda)\delta C_0} - e^{-(\mu - \lambda)\rho C_0 / \lambda} \) will bound (18) for all \( C \geq C_0 \).

### 5.3 Standard Clock Acceleration

In the previous section we used the common queueing model approach for asymptotic efficiency. As will be explained in detail in the following subsection, incorporation of fast simulation and \( A \)-cycles requires simulating the process with cluster acceleration only until blocking is arrived at. Once the state enters \( B_i \) one computes the time that the state remains blocked with the current \( A \)-cycle, or \( E[X(T^{(i)})|R_i] \). Devetsikiotis and Townsend (1993) propose to use now another change of measure to empty the occupancy quickly for fast termination of the \( A \)-cycle. The usual problem for these systems is that the acceleration affects several holding times even after blocking has been detected and may slow down termination of the cycles. L’Ecuyer and Champoux (1996) seek to find a smaller stopping time in the hope of stopping the acceleration before blocking occurs. Instead we propose to do it as follows.

The simulation model is the Standard Clock technique of Vakili (1991), which in the exponential model is equivalent to uniformization of the process. When the total occupancy of the
current state is \( n \), an exponential random variable with intensity \( \Lambda_n \) is used to determine the
inter-event time, or the time for the next event. Here
\[
\Lambda_n = \sum_j \lambda_j + n\mu.
\]

Next, the event type is determined as a discrete random variable with distribution:
\[
D = \begin{cases} 
  a_j : \text{arrival at cell } j & \text{w.p. } \frac{\lambda_j}{\Lambda_n} \\
  s_j : \text{termination of call at cell } j & \text{w.p. } \frac{n_j\mu}{\Lambda_n}
\end{cases}
\]

where \( n_j \) is the number of calls at cell \( j \), so that \( n = \sum_j n_j \).

The appropriate filtration for this process is no longer (16) but:
\[
\delta_k = \sigma(T_1, \ldots, T_k; D_1, \ldots, D_k), \quad k = 1, 2, \ldots
\]
where \( T_k \) are the successive inter-event times and \( D_k \) the corresponding event types or “decisions”. The imbedded occupancy process is updated by setting
\[
n_j(k+1) = \begin{cases} 
  n_j(k) + 1 & \text{if } n(k) \not\in B_i \text{ and } D_{k+1} = a_j, \\
  n_j(k) - 1 & \text{if } D_{k+1} = s_j
\end{cases}
\]

Based on our results of the previous section, we conjecture now that the same change of measure is asymptotically optimal for estimating \( B \) (rather than \( P[R_i] \)). Notice that, although the two models describe the same process (in distribution), the partial histories \( \delta_k \) are not equivalent. Filtration (16) contains filtration (19) but clearly, \( H_k \) is not measurable w.r.t. \( \sigma(T_1, \ldots, T_k; D_1, \ldots, D_k) \).

The accelerated cluster process has arrival rate \( \mu \) at the cluster and holding times with intensity \( \lambda \) for calls within the cluster. That is, given \( \delta_k \), \( T_{k+1} \sim \exp(\Lambda^*_n(k)) \), for \( n(k) = n_1(k) + \ldots + n_K(k) \) the total occupancy of the process after event \( k \) has occurred. The new event rate is a \( \delta_k \)-measurable random variable:
\[
\Lambda^*_n(k) = \mu + \sum_{j \not\in C_i} \lambda_j + \lambda \sum_j n_j(k) + \mu \sum_j n_j(k),
\]
and the event types now follow:
\[
D_{k+1} = \begin{cases} 
  \text{arrival at cell } j \in C_i & \text{w.p. } \frac{\mu}{\Lambda_n} \\
  \text{arrival at cell } j \not\in C_i & \text{w.p. } \frac{\lambda_j}{\Lambda_n} \\
  \text{termination of call at cell } j \in C_i & \text{w.p. } \frac{n_j\lambda_i}{\Lambda_n} \\
  \text{termination of call at cell } j \not\in C_i & \text{w.p. } \frac{n_j\mu}{\Lambda_n}
\end{cases}
\]

Within an \( A \)-cycle, we perform this change of measure until the \( \tau_i \)-th event occurs, which is the first time that the state is in \( B_i \). Because the distributions of \( T_{k+1} \) and \( D_{k+1} \) are conditional
on $\mathcal{F}_k$, the change of measure defines a multiplicative martingale $D^*$ which is adapted to the filtration \( \{\mathcal{F}_k, k \geq 1\} \), and the corresponding Radon-Nykodim derivative is given by:

\[
L_{\tau_i} = \prod_{k=1}^{\tau_i-1} \left( \frac{\Lambda_n(k)'}{\Lambda_n(k)} \right) e^{(\Lambda_n(k)')-\Lambda_n(k)'}T_{k+1} \times \prod_{k=1}^{\tau_i-1} \left( \frac{\Lambda_n(k)'}{\Lambda_n(k)} \right) \left( \sum_{j \in C_i} \left( \frac{\lambda_j}{\mu} \right) 1_{\{D_{k+1}=s_j\}} + \left( \frac{\mu}{\lambda} \right) 1_{\{D_{k+1} \in S_i\}} \right)
\]

where \( a_j \) denotes the event that a call arrives at cell \( j \) and \( S_i = \cap_{j \in C_i} s_j \) is the set of event types which are termination of calls within the cluster. From their definitions, it follows that:

\[
\Lambda_n' - \Lambda_n = (\mu - \lambda) + (\lambda - \mu) \sum_{j \in C_i} n_j = -(\mu - \lambda)(n^{(i)} - 1)
\]

where \( n^{(i)} = \sum_{j \in C_i} n_j \) is the total occupancy of the cluster.

Simplifying the expression above,

\[
L_{\tau_i} = e^{- (\mu - \lambda) \sum_{k=1}^{\tau_i-1} (n^{(i)}(k) - 1) T_{k+1}} \left[ \prod_{j \in C_i} \left( \frac{\lambda_j}{\mu} \right) k_1(j) \right] \left( \frac{\mu}{\lambda} \right)^{k_2} \tag{20}
\]

where \( k_1(j) \) is the total number of arrivals to cell \( j \) up to event number \( \tau_i \) and \( k_2 \) is the corresponding number of call completions. Clearly \( \lambda_j < \lambda = \sum_{j \in C_i} \lambda_j \), and similarly, because there are \( C \) on-going calls within the cluster at the blocking time \( \tau_i \), then \( k_1 = \sum_{j \in C_i} k_1(j) = k_2 + C \). This yields the a.s. bound:

\[
\left( \frac{\mu}{\lambda} \right)^{k_2} \prod_{j \in C_i} \left( \frac{\lambda_j}{\mu} \right)^{k_1(j)} < \left( \frac{\lambda}{\mu} \right)^{C} < 1.
\]

Because the stopping time within an \( A \)-cycle considers only the transitions from the start of the \( A \)-cycle (an arrival at empty cell \( i \)) up until blocking, on the set \( \{\omega : \tau_i < T^{(i)} \} \) we have \( n^{(i)}(k) \geq 1, k = 1, \ldots, \tau_i \), which yields variance reduction of the Importance Sampling estimator with the Standard Clock (ISSC) \( L_{\tau_i} 1_{\{R_i\}} < 1, P^* - w.p.1 \).

### 5.4 Parallel Computation Using \( A \)-cycles

As explained in Section 5.1, (6) can be estimated using:

\[
E[X_i(T^{(i)})] = E \left( E[X_i(T^{(i)})|R_i] 1_{\{R_i\}} \right) = E^* \left( E[X_i(T^{(i)})|R_i] L_{\tau_i} 1_{\{R_i\}} \right). \tag{21}
\]

If we could calculate the conditional expectation analytically, it would naturally reduce the variance in the estimation and the simulation could stop once a blocking state is attained. Instead, \( E[X_i(T^{(i)})|R_i] \) can be calculated from the original measure once the blocking states are arrived at through the speed up of the cluster processes. The martingale process \( \{L_k\} \) is stopped at \( \tau_i \) and the subsequent inter-arrival and holding times are simulated according to the \emph{original} distributions. The reasons for stopping the change of measure are first, that under the new measure it will now take a long time to empty the cell (and finish the \( A \)-cycle), and second, that the bound (18) would not be satisfied. Devetsikoti and Townsend (1993) propose to accelerate the process \emph{back to emptying} the \( i \)-th cell, suggesting that further gain may yet be
obtained. Whether such scheme improves the efficiency of our estimators or not is the subject of further investigations.

Since $A$-cycles must start with the invariant distribution according to the original measure, the terminating states from an accelerated $A$-cycle should not be used to start a subsequent $A$-cycle. Chang, Heidelberger, and Shahabiddin (1995) and L’Ecuyer and Champoux (1996) use the splitting or “restart” technique, which yields the following algorithm.

**Algorithm 4:**

1. Warm up: Simulate $\{n(S_k), k = -k_0, \ldots, -1\}$ until the process is assumed to be in stationary state. Set $j = 1, M_i = 0, i \in \{1, \ldots, 7\}$.

2. As soon as the nominal process enters a state $n_i = 0$ for some $i \in \{1, \ldots, 7\}$, the current state (occupancy and residual times) is kept in memory, and is called $N(j)$. The cycle counter per cell is updated: $M_i \leftarrow M_i + 1$.

3. Start a parallel process from $N(j)$ using the change of measure for the cluster $C_i$ if $i \neq 1$, and accelerating only cell 1 arrival process and holding times in case $i = 1$. Once $\tau_i$ is arrived at, record $L_{\tau_i}1_{\{R_i\}}$.

4. If the value $L_{\tau_i}1_{\{R_i\}} > 0$ (a blocking state was reached) then continue the parallel computation generating now the original measure, until $n_i = 0$ is reached. Record $X_i(M_i)$, the total time spent in blocking states $B_i$ during the $A$-cycle.

5. Restart the nominal process from state $N(j)$ with the original measure. Set $j \leftarrow j + 1$ and go to 2.

The denominator in (6) can be estimated consistently with:

$$E[T^{(i)}] = \lim_{T \to \infty} \frac{T}{M_i(T)}$$

where $T$ is the total simulation time recorded from the nominal process and $M_i(T)$ is the final counter value for $A$-cycles for cell $i$. The numerator in (6) is obtained using:

$$E^* \{E[X(T^{(i)})|R_i]L_{\tau_i}1_{\{R_i\}}\} = \lim_{T \to \infty} \frac{1}{M_i(T)} \sum_{m=1}^{M_i(T)} X_i(m)L_{\tau_i(m)}$$

with

$$X_i(m) = \text{Time in } B_i \text{ during } m\text{-th } A\text{-cycle}.$$  

Since successive $A$-cycles for any cell are correlated, and the use of a single nominal process also correlates the estimates for different cells, the batch means method is used to estimate the variance of the estimator:

$$\hat{Y}(T) = \sum_{i=1}^{K} \left( \frac{\lambda_i}{\lambda} \right) \frac{1}{T} \sum_{m=1}^{M_i(T)} X_i(m)L_{\tau_i(m)}.$$
5.5 Discussion

The fast simulation framework above uses known methods for efficient estimation of the probabilities of rare events \( P(R_i) \). As mentioned in Chang, Heidelberger, and Shahabiddin (1995), when estimating blocking, the relative variance and CPU time of the estimation in (21) must be considered. Devetsikiotis and Townsend (1993) even suggest that effects in the initial acceleration may degrade performance after time \( \tau_i \), when the important event becomes emptying the state. Although the queueing framework was helpful in determining the bounds for the relative error of IS estimation, discrete event simulation using filtration (16) requires keeping an event list with a number of events which grows as \( K C \). Searching through this list to determine next events entails a computational effort of the order \( O(K C \log(K C)) \). In contrast, ISSC requires generating next events and then deciding which event types occur. Simulation time under the standard clock is of the order \( O(K C) \) and could compensate in relative efficiency with respect to the IS estimator. Our preliminary results indicate that this may well be the case.

When estimating \( B_i \), cell \( i = 1 \) is treated differently from the others. This is because in order for the change of measure to satisfy the bounds it was assumed that the aggregate arrival rate is bounded by \( \mu \). This would be very unrealistic, for cluster \( C_1 \) contains all cells in the network. Such underutilized models are probably not of much interest in practice. Therefore only the processes local to cell 1 are simulated under a change of measure. The corresponding estimator no longer satisfies the bound (18), although it can be shown that it has a reduced variance.

The treatment of non-exponential holding times for the IS estimator under the DES framework of Section 5.2 is similar to the change of measure approach in Vázquez-Abad and LeQuoc, 1999, where tilted exponentials are used to perform the change of measure, as suggested in Ross, 1997:

\[
    f^*(x) = \frac{e^{ax}f(x)}{\phi_X(a)}
\]

where \( \phi_X(a) = E[e^{aX}] \) is the moment generating function of \( X \) w.r.t. the original measure. The parameter \( a \) is chosen as a function of \( \mu, \lambda_i; i = 1, \ldots, K \) to optimize the estimator’s relative efficiency. For the ISSC estimator of Section 5.3 the standard clock can still be used, but the residual life times distributions must be taken into account, as explained in detail in Hu (1995).

6 Concluding Remarks

The MCMC methods proposed for blocking probability calculations, namely Metropolis Hastings and Gibbs sampler, clearly outperform the usual acceptance/rejection method and their relative efficiencies grow with problem size and with increasing load. However, all three methods may suffer when the load is relatively low, because of the waste in computational effort.

To overcome this problem, we implemented the main ideas of fast simulation using standard queueing-like results under the discrete event framework for simulation. Next we suggest using this change of measure for the standard clock framework for simulation, which entails significantly less computational effort.

Fast simulation can be shown to be asymptotically efficient under the condition that the aggregate cluster arrival rate \( \lambda \) be smaller than \( \mu \). However, this condition may not be satisfied for all clusters. A more realistic assumption is to suppose that \( \lambda < C \mu \). We are currently
studying a dynamic change of measure under the framework of Section 5.2 that swaps rates according to the current state: because the model is that of parallel servers, the effective service rate is $n_k \mu$ instead of $\mu$. A change of measure based on the current state is adapted to the filtration and must be studied using a Filtered Monte Carlo framework similar to the one in Section 5.3. A similar analysis than the one in Asmussen and Nielsen (1995) can be used to study such changes of measure.

Future third generation networks are expected to contain very large numbers of pico-cells, many of which will be lightly loaded for much of the time due to their small size. These are precisely the conditions under which the proposed techniques show the greatest improvement over the techniques currently in use for the design of cellular networks.

When dealing with cellular network lay-outs, it may be that some regions of the space satisfy the conditions for which the MCMC methods are best suited, while other regions may fall under the rare event framework. We believe that hybrid methods may well prove to be the most efficient ones in practice.

Acknowledgement

The authors gratefully thank Arnaud Doucet for suggesting the use of the Gibbs sampler.

References:


