To be published in *Proceedings of the 29th Simulation symposium, New Orleans, Louisiana, April 1996.*

Computation of the Asymptotic Bias and Variance for Simulation of Markov Reward Models

Aad P. A. van Moorsel, Latha A. Kant, and William H. Sanders

Center for Reliable and High-Performance Computing
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main St., Urbana, IL 61801
(moorsel, lkant, whs)@crhc.uiuc.edu

Abstract

The asymptotic bias and variance are important determinants of the quality of a simulation run. In particular, the asymptotic bias can be used to approximate the bias introduced by starting the collection of a measure in a particular state distribution, and the asymptotic variance can be used to compute the simulation time required to obtain a statistically significant estimate of a measure. While both of these measures can be computed analytically for simple models and measures, e.g., the average buffer occupancy of an M/G/1 queue, practical computational methods have not been developed for general model classes. Such results would be useful since they would provide insight into the simulation time required for particular systems and measures and the bias introduced by a particular initial state distribution. In this paper, we discuss the numerical computation of the asymptotic bias and variance of measures derived from continuous-time Markov reward models. In particular, we show how both measures together can be efficiently computed by solving two systems of linear equations. As a consequence of this formulation, we are able to numerically compute the asymptotic bias and variance of measures defined on very large and irregular Markov reward models. To illustrate this point, we apply the developed algorithm to queues with complex traffic behavior, different service time distributions, and several alternative scheduling disciplines that may be typically encountered in nodes in high-speed communication networks.

1 Introduction

Steady-state discrete-event simulation is frequently used for quantitative evaluation of computer and communication systems. It provides an estimate of the performance and/or dependability (e.g., availability) of a system whose accuracy and precision [1] depends on the "initial transient" period and "observation period" of the simulation [8]. In planning which experiments to carry out in the time span available, one is interested in the time (initial transient + observation period) individual simulation experiments will require to reach a desired accuracy and precision. If this information is available, a decision can be made whether experiments are feasible, whether it is economically justified to carry them out, and what accuracy and precision can be obtained in the allocated time.

In steady-state discrete-event simulation, the *initial transient* can be defined as the period of time which must be discarded (before beginning to collect observations) to reach a desired accuracy in the simulation. More precisely, the accuracy of a simulation is determined by the "bias" of the measure – the difference between the expected outcome of the simulation and the actual result. Bias is introduced by the state distribution at the end of the initial transient period. When the observation period is long (as is usually required to obtain acceptable precision), the *asymptotic bias* – the difference between the expected outcome of the simulation and actual result, as the observation period goes to infinity – is a good measure of accuracy.

Likewise, the *observation period* of a simulation can be defined as the period of time that observations must be collected to reach the desired precision. The precision, in turn, is specified by a confidence level z and a desired width of the z% confidence interval. The observation period to achieve a desired confidence interval with a specified level is determined by the "asymptotic variance" of the measure defined on the model. Said informally, the *asymptotic variance* of the measure quantifies the variability one can expect in the possible outcomes of the simulation. Thus both the asymptotic bias and variance are important measures of the quality of results that can be obtained from a simulation.

Determining the asymptotic bias and/or the asymptotic variance has received considerable attention (e.g., [2, 4, 5, 14, 19]). In general, two approaches have been taken. The first approach has been to derive explicit, closed-form expressions of the asymptotic bias and variance of specific measures and models. Explicit expressions have as their main advantage that they are usually insightful and easy to compute. However, they only apply for the specific model and measure they were derived for, and derivations are only available for

relatively simple cases. In many cases, the derivations depend on assumptions that make the resulting expressions approximate. For instance, most of the results in [19] are derived for the average buffer occupancy in single queue models, and are based on heavy-traffic approximations which require knowledge of the first two moments of inter-arrival and service-time distributions (which are not always readily available).

The second approach has been to derive algorithmic expressions of the asymptotic bias and variance for larger classes of measures and models. In particular, Hazen and Pritsker [5], Grassmann [4] and Glynn [2] have developed expressions for the asymptotic bias and/or variance of measures expressed in terms of continuous-time Markov reward models. These methods have the advantage that they apply to a large class of Markov reward models, allowing considerable freedom in the model and measure considered. However, the expressions in [2, 4, 5] require matrix inversions, which makes them computationally unattractive. This requirement severely limits, in practice, the largeness and complexity of the models that can be considered, since the sparseness of a matrix is usually lost during its inversion.

This paper presents an efficient algorithm for computing the asymptotic bias and variance which is not limited to measures and models with a special structure, as are the closed-form expressions that have been developed, nor requires matrix inversions, as do the expressions developed in [2, 4, 5]. The method is based on the matrix expressions derived in [2, 4, 5] and is thus applicable to the general class of Markov reward models. However, the algorithm requires only the single solution of two systems of simultaneous linear equations. Iterative methods can be used to solve these equations (e.g., successive over-relaxation [3]), and hence models with hundreds of thousands of states can be solved on a modern workstation. Note that the algorithm requires as much effort as solving the system of equations to obtain an exact answer for the measures, and hence should not be viewed as a practical step in solving for measures in specific models. Instead, it is a method by which to gain insight into the effect particular types of measures and models have on the quality of the results that can be obtained via simulation. Apart from being interesting in their own right, these insights can be very important in a practical sense, helping to plan simulations of systems whose state spaces are too large to permit analytic solution, or that have non-exponential delays.

The algorithm has also been implemented as a solver in the *UltraSAN* [16] modeling package. *UltraSAN* permits the automatic generation of large Markov reward models from a compact, higher-level representation known as stochastic activity networks [10, 11]. Complex architectures, work loads, and measures can be represented as SANs, avoiding the need to build the possibly extremely large Markov process representations by hand. The tool implementation thus dramatically increases the number of practical models and measures that can be solved. An example application is also presented. The model under investigation is a single queue with finite buffer, multiple sources of non-homogeneous and bursty workload, priority queueing, and Erlang service discipline. The measures considered in the example are the average queue length and the probability the buffer is full. More generally, we discuss generic insights valid for broad classes of models as well as investigate the influence of particular system details on the initial transient and the observation period. No results have been established for such a model in the literature, but they are easily obtainable using our approach.

2 Preliminaries

Before outlining our approach, it is helpful to introduce the problem in a more precise manner than presented in the introduction. We restrict our attention to models and measures that can be specified in the following class of Markov reward models. Let $X = \{X(t) : t \geq 0\}$ denote an ergodic continuous-time Markov chain (CTMC) with finite state space S. Let r_i denote the *reward rate* associated with state $i \in S$. The reward is collected over every time interval the system is in state i. This reward structure is identical to that considered in [2, 5], and is a specific (but natural) case of the so-called yield rate in [6] (specific, since the rewards are independent of the successor state of i). The CTMC together with the reward structure form a Markov reward model.

Given this class of Markov reward models, one can apply simulation to obtain the steady-state reward $\mathbf{E}R$, defined as

$$\mathbf{E}R = \lim_{t \to \infty} \mathbf{E}R(t),$$

where $R(t) = r_{X(t)}$, i.e., $R(t)$ is the rate of reward at instant t. Since we restrict our attention to finite ergodic Markov reward models, $\mathbf{E}R$ is always well-defined.

The standard estimator $\hat{R}(t)$ for $\mathbf{E}R$ is

$$\hat{R}(t) = \frac{1}{t} \int_0^t R(s)\ ds, \tag{1}$$

where t denotes the interval over which the simulation is carried out. The bias $\mathbf{B}$ is the difference between the expected outcome of the simulation and the actual result, i.e.,

$$\mathbf{B} = \mathbf{E}\hat{R}(t) - \mathbf{E}R. \tag{2}$$

Hence, the estimator $\hat{R}(t)$ is unbiased when $\mathbf{E}R(s) = \mathbf{E}R$ for all $s \in [0, t]$, since then

$$\mathbf{E}\hat{R}(t) = \frac{1}{t} \int_0^t \mathbf{E}R(s)\ ds = \mathbf{E}R. \tag{3}$$

However, $R(s), s \in [0, t]$, is dependent on the initial distribution, and therefore the estimator (1) will usually be biased.

Since $\hat{R}(t)$ converges to $\mathbf{E}R$ for increasing t [2], it can be expected that the introduced bias decreases if

observations are discarded for the initial transient period. In particular, let t_d be the time point such that all observations in $[0, t_d]$ are discarded. Then the bias $\mathbf{B}_d$ becomes

$$\mathbf{B}_d = \frac{1}{t - t_d} \int_{t_d}^{t} \mathbf{E}R(s) - \mathbf{E}R \; ds.$$

The time interval $[0, t_d]$ is called the *initial transient*. To overcome writing t_d in the expressions that follow, we assume that the clock is "reset" to 0 at instant t_d, and that the "initial" distribution equals $\underline{\pi}(t_d)$. Then, if the simulation is carried out over t time units after time point t_d, the estimator $\hat{R}(t)$ after deletion of the initial transient is as in (3), and the bias $\mathbf{B}$ is as in (2). Furthermore, if the observation period is long enough, bias at the end of the observation period will be extremely small. Therefore, the bias can be approximated by the *asymptotic bias*, defined as

$$\mathbf{B} = \int_{0}^{\infty} \mathbf{E}R(s) - \mathbf{E}R \; ds.$$

It is important to realize that the influence of the (asymptotic) bias depends on the length of the observation period, since the estimator (1) contains a division by the length of the observation period. It is therefore of interest to determine how long the simulation should be to outweigh the asymptotic bias. To determine this, we assume that the observation period is long enough such that the asymptotic bias (which represents the bias over an infinite "observation period") is a close approximation of the actual bias over the finite observation period. If the observation period has length s, and if the asymptotic bias should be such that $\mathbf{B}/s < \epsilon$, then it follows that s should obey the following inequality ("$\doteq$" denoting "is defined as"):

$$s \geq s_{abs} \doteq \frac{|\mathbf{B}|}{\epsilon}.$$

Similarly, if the bias is considered relative to the average, we get for some chosen ϵ, that:

$$s \geq s_{rel} \doteq \frac{|\mathbf{B}|}{\epsilon \mathbf{E}R}. \tag{4}$$

The precision of the simulation can be determined using the asymptotic variance and constructing a confidence interval about the estimate. This can be done using the central limit theorem which specifies that under mild conditions (e.g., [5, 19] contain references to the precise conditions),

$$\sqrt{t}(\hat{R}(t) - \mathbf{E}R) \Rightarrow N(0, \mathbf{AV}), \text{ as } t \to \infty.$$

In this equation, the *asymptotic variance* $\mathbf{AV}$ is given by [19]

$$\mathbf{AV} = Var(\sqrt{t}\hat{R}(t)) = 2 \int_{0}^{\infty} \mathbf{Cov}(R(0), R(s)) \; ds.$$

The confidence interval about the estimate of the measure then is given by

$$[\hat{R}(t) - \frac{\sqrt{\mathbf{AV}}\gamma(z)}{\sqrt{t}}, \; \hat{R}(t) + \frac{\sqrt{\mathbf{AV}}\gamma(z)}{\sqrt{t}}], \tag{5}$$

where $\gamma(z)$ is the point on the standard normal distribution $N(0,1)$ such that for the predefined percentage point z,

$$\mathbf{Pr}\{N(0,1) > \gamma(z)\} = \mathbf{Pr}\{\sqrt{t}\frac{|\hat{R}(t) - \mathbf{E}R|}{\sqrt{\mathbf{AV}}} > \gamma(z)\} =$$

$$\mathbf{Pr}\{|\hat{R}(t) - \mathbf{E}R| > \frac{\sqrt{\mathbf{AV}}\gamma(z)}{\sqrt{t}}\} = z.$$

In words, this implies that the probability that the actual measure $\mathbf{E}R$ lies outside the confidence interval of (5) is less than $z\%$. From the confidence interval, one can directly derive the observation period t, necessary to achieve a desired accuracy, as follows. Assume one wants the half width of the confidence interval to be less than or equal to some pre-defined value $\epsilon > 0$, t then must be such that

$$t \geq t_{abs} \doteq \frac{\mathbf{AV}\gamma(z)^2}{\epsilon^2}.$$

If a relative accuracy ϵ is desired, then the half width of the confidence interval must be within $\epsilon \mathbf{E}R$, thus leading to

$$t \geq t_{rel} \doteq \frac{\mathbf{AV}\gamma(z)^2}{(\mathbf{E}R)^2\epsilon^2}. \tag{6}$$

Note that the observation periods t_{abs} and t_{rel} are expressed in time units on the scale of the model.

3 Computation of Asymptotic Bias and Asymptotic Variance

In this section, we derive an algorithmic method to compute both the asymptotic bias and variance for Markov reward models. The method obtains the asymptotic bias and variance simultaneously, requiring only the solution of two systems of linear equations, which can be done efficiently using an iterative method such as successive over-relaxation.

Let the CTMC X be defined by the generator matrix $\mathbf{Q}$ and let $\mathbf{P}(t)$ be the transient transition probability matrix with entries $\mathbf{P}_{i,j}(t) = \mathbf{Pr}\{X_t = j | X_0 = i\}$. The row vector $\underline{\pi}$ denotes the steady-state distribution of the CTMC X. For notational convenience, we let $\mathbf{\Pi}$ denote the diagonal matrix with $\mathbf{\Pi}_{i,i} = \pi_i, i \in S$, and $\mathbf{\Pi}^r$ the matrix with all rows containing the vector $\underline{\pi}$, i.e., $\mathbf{\Pi}^r_{i,j} = \pi_j, i, j \in S$. For a CTMC, the *fundamental matrix* $\mathbf{Z}$ is defined by Keilson [7] as

$$\mathbf{Z} = \int_{0}^{\infty} (\mathbf{P}(t) - \mathbf{\Pi}^r) dt. \tag{7}$$

This integral is well-defined for finite ergodic Markov chains because $\mathbf{P}(t)$ tends to $\mathbf{\Pi}^r$ exponentially fast when $t \to \infty$. It can be shown [2] that

$$\mathbf{Z} + \mathbf{\Pi}^r = (\mathbf{\Pi}^r - \mathbf{Q})^{-1} \doteq \mathbf{F}.$$

The asymptotic bias and asymptotic variance are then given by [2]

$$\mathbf{B} = \underline{\pi}(0)\mathbf{Z}\underline{r}', \tag{8}$$

and

$$\mathbf{AV} = 2\underline{r}\mathbf{\Pi}\mathbf{Z}\underline{r}', \tag{9}$$

$\underline{r}$ and $\underline{r}'$ being the reward row and column vector, respectively. However, equations (8) and (9) involve the matrix inverse $\mathbf{Z} = \mathbf{F} - \mathbf{\Pi}^r$ and are therefore often not tractable for practical models. The following section provides the translation from equations (8) and (9) to a system of linear vector equations $\mathbf{Q}\underline{x} = \underline{y}$ which can be solved for models with large state spaces by iterative methods.

3.1 Linear Equations

The derivation of the linear equations follows from several different matrix equalities. First, it is evident that $\mathbf{\Pi}^r\mathbf{Q} = \mathbf{0}$ and $\mathbf{\Pi}^r\mathbf{\Pi}^r = \mathbf{\Pi}^r$, and hence that $(\mathbf{\Pi}^r - \mathbf{I})(\mathbf{\Pi}^r - \mathbf{Q}) = \mathbf{Q}$. Furthermore, the equality $\mathbf{QZ} = \mathbf{Q}(\mathbf{F} - \mathbf{\Pi}^r) = \mathbf{QF} = (\mathbf{\Pi}^r - \mathbf{I})(\mathbf{\Pi}^r - \mathbf{Q})(\mathbf{\Pi}^r - \mathbf{Q})^{-1} = \mathbf{\Pi}^r - \mathbf{I}$ leads to

$$\mathbf{QZ}\underline{r}' = (\mathbf{\Pi}^r - \mathbf{I})\underline{r}'. \tag{10}$$

Defining the vectors

$$\underline{x} = \mathbf{Z}\underline{r}' \text{ and } \underline{y} = (\mathbf{\Pi}^r - \mathbf{I})\underline{r}', \tag{11}$$

Equation (10) can be written as the system of linear equations $\mathbf{Q}\underline{x} = \underline{y}$. From Equations (8) and (9) it then follows that $\mathbf{B} = \underline{\pi}(0) \times x$ and $\mathbf{AV} = 2\underline{r}\mathbf{\Pi}\underline{x}$, where $\underline{a} \times \underline{b}$ denotes the inner product $\sum_{i \in S} \underline{a}_i \underline{b}_i$. However, the system (11) does not possess a unique solution because of the rank deficiency of $\mathbf{Q}$. In particular, observe that since $\mathbf{Q}\underline{e} = \underline{0}$, if $\underline{\tilde{x}}$ is such that $\mathbf{Q}\underline{\tilde{x}} = \underline{y}$, then also $\mathbf{Q}(\underline{\tilde{x}} + c\underline{e}) = \underline{y}$, for any $c \in \Re$ ($\underline{e}$ is a vector with all elements equal to 1). Therefore a normalization condition on $\underline{x}$ will be enforced to establish a unique solution.

To determine the normalization equation, observe that from the definition of $\mathbf{Z}$ in (7) it follows that $\underline{\pi}\mathbf{Z} = \underline{0}$. Hence, $\underline{\pi} \times \underline{x} = 0$, and the system of linear equations to solve becomes

$$\mathbf{Q}\underline{x} = (\mathbf{\Pi}^r - \mathbf{I})\underline{r} \text{ and } \underline{\pi} \times \underline{x} = 0. \tag{12}$$

It remains to be shown that the normalization condition $\underline{\pi} \times \underline{x} = 0$ establishes uniqueness of the solution of the system of linear equations (12). We do this by contradiction. In particular, assume that both $\underline{x}$ and $\underline{x}', \underline{x}' \neq \underline{x}$, are solutions of (12). Then $\underline{\pi} \times \underline{x} = \underline{\pi} \times \underline{x}' = 0$, and, furthermore, there exists a $c \in \Re$ such that $\underline{x} = \underline{x}' + c\underline{e}$. Combining these facts it follows that $c\underline{\pi} \times \underline{e} = 0$, and, hence, that $c = 0$, and $\underline{x}' = \underline{x}$. So $\underline{x} = \underline{x}'$ is the unique solution of the system (12).

3.2 Iterative Solution Algorithm

To efficiently solve the system of linear equations just derived, we develop a scheme that is similar to that of solving the steady-state distribution $\underline{\pi}$ from the system $\underline{\pi}\mathbf{Q} = \underline{0}$, with the normalizing equation $\sum_{i \in S} \pi_i = 1$. The preferred method of solving these equations is to first solve for any solution of $\underline{\pi}\mathbf{Q} = \underline{0}$, then normalize (as opposed to including the normalization in the system of linear equations and solving for the unique steady-state distribution at once) [17]. We can do the same for the system of linear equations in (12). In particular, observe that since $\mathbf{Q}\underline{e} = \underline{0}$, if $\underline{\tilde{x}}$ is such that $\mathbf{Q}\underline{\tilde{x}} = \underline{y}$ (but not necessarily $\underline{\pi} \times \underline{\tilde{x}} = 0$), then $\mathbf{Q}(\underline{\tilde{x}} - c\underline{e}) = \underline{y}$, for any c. If we then take c such that $c = \underline{\pi} \times \underline{\tilde{x}}$, we obtain that $\underline{\pi} \times (\underline{\tilde{x}} - c\underline{e}) = 0$, and thus $\underline{x} = \underline{\tilde{x}} - c\underline{e}$ is the unique solution of (12).

Following this argument, we obtain the following algorithm for computing the asymptotic bias and variance for a given set of reward variables:

Algorithm 1 Algorithm for computing B and AV

The asymptotic bias and variance can be computed via the following algorithm:

```
Compute π from πQ = 0;
Do for all r
{
    y = [Π^r − I]r;
    Solve for x by solving Qx = y;
    c = π × x;
    x = x − ce;
    B = π(0) × x;
    AV = 2rΠx;
}
```

B and **AV** can thus be obtained, for any set of reward variables, by a single solution of two systems of linear equations. Both of these solutions can be done iteratively, using successive over-relaxation (e.g., [3]). Iterative methods will preserve the sparsity of a model and hence permit the solution of models with hundreds of thousands of states on a typical workstation. An example of such a solution will be given in the next sections, which study the observation period for measures defined on a complex queueing system.

4 Tool Implementation and Application

The solution algorithm developed in the previous section has been implemented in the *UltraSAN* modeling package [16]. We will discuss the tool implementation in this section and demonstrate its use by deriving the observation period for different varieties of a model of a high-speed communication network node. The introduced models have over one hundred thousand states but can be conveniently specified and solved in *UltraSAN*.

4.1 Tool Implementation

UltraSAN is a software package for the evaluation of the performance and/or dependability of systems represented as stochastic activity networks (SANs). From the high-level model specification, *UltraSAN* automatically derives the underlying Markov reward model. Several solution methods for Markov reward models have been incorporated in *UltraSAN* to analytically derive both transient and steady-state measures that are of interest. The tool is organized such that all solvers are independent modules which use as input the files containing the previously generated generator matrix and reward vectors. The solver for deriving the asymptotic bias and variance can thus be easily added and benefit from the model specification and construction facilities of *UltraSAN*.

UltraSAN takes as input a Markov process and set of reward structures (generated automatically from a SAN-level description) and set of time points. It computes, using the algorithm presented in the last section, the asymptotic variance and, for each time point, the asymptotic bias assuming that the time point is the end of the initial transient period for the simulation. In this way, one can easily see the effect of changing the initial transient period on the bias introduced into the estimator.

To obtain the desired results, the steady-state distribution $\underline{\pi}$ is first obtained from $\underline{\pi}\mathbf{Q} = \underline{0}$ and $\sum_{i \in S} \pi_i = 1$ using successive over-relaxation (SOR). Then, for each reward variable, $\underline{y} = [\mathbf{\Pi}^r - \mathbf{I}]\underline{r}$ is computed using the steady-state distribution $\underline{\pi}$, and $\mathbf{Q}\underline{x} = \underline{y}$ and $\underline{\pi} \times \underline{x} = 0$ is solved using SOR. The asymptotic variance for each variable is then obtained using the equation given in Algorithm 1. To obtain the asymptotic bias for different initial transients, the solver then solves for the transient state occupancy probability at each specified time point (using uniformization, given a specified initial state) and applies the appropriate equation in Algorithm 1. The effect of different initial states on the asymptotic bias can be studied by specifying different initial markings for the SAN. From these results, it can be seen which initial state is optimal with respect to the bias, and how long the initial transient should be to diminish the introduced bias.

4.2 Example Application

As an example of the computation of the asymptotic bias and variance, we consider a model of a node of a communication network. In our discussion, we will look at three architectures of the system: the *delay architecture* for a node dedicated to delay-sensitive applications such as voice, the *loss architecture* for loss-sensitive applications such as data transfer, and the *hybrid architecture* for the combination of the two.

Figure 1 presents the stochastic activity network of the hybrid architecture. The models for the delay and loss architectures are straightforward simplifications of this model and, hence, will not be discussed. The dynamics of a SAN model are specified in a similar way as that of stochastic Petri nets (SPNs), i.e., the distribution of tokens over places determines the state of the process, while activities (similar to "transitions," in SPNs) and their associated time delays determine the way the process can change state. In the following description, we will thus only elaborate on the SAN-specific features of the model. For more detailed information regarding SANs, see [10, 11, 16].

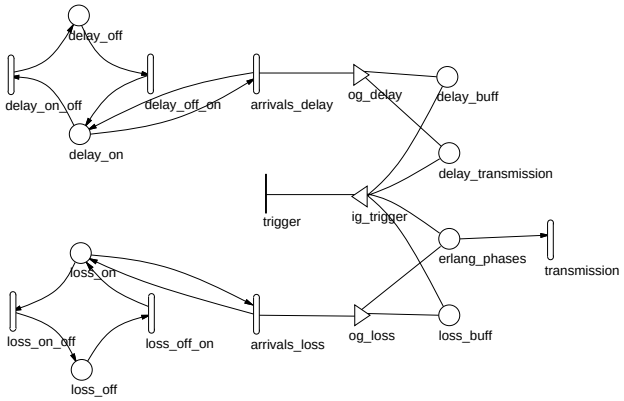

Figure 1: Stochastic activity network for the hybrid architecture.

The hybrid architecture considers delay and loss-sensitive traffic, both of which are modeled by interrupted Poisson processes. Arriving jobs for both traffic sources are queued in dedicated buffers (*delay_buff* and *loss_buff*, respectively), which have finite capacities. Furthermore, the places *delay_transmission* and *erlang_phases* determine the type of job that is currently being transmitted (delay or loss sensitive), and the number of phases of an Erlang distribution the transmission will take. The way the marking of the network is changed when an activity completes is determined by a construct in SANs known as "gates." Gates, depicted as triangles, make it possible to specify rules for the transitions that are different and typically more complicated than those in standard stochastic Petri nets. More formally, the functions in gates can be any computable function, on the marking of the connected places.

The gates in our model have the following functionality. To stop arriving jobs from entering the buffer when it is full, the output gate *og_delay* prohibits a token moving to the place *delay_buff*, whenever the number of delay-sensitive jobs equals the buffer size. More complicated functionality that can be modeled with a gate definition can be found in the input gate *ig_trigger*, whose definition is given in Table 1. It determines the scheduling discipline of the modeled queue. In particular, its "predicate" (see Table 1) specifies that the instantaneous activity *trigger* is enabled when the transmission of the previous job is finished (the place *erlang_phases* is empty) and when there is at least one new job present in one of the buffers. Similarly, its "function" (executed when activity *trigger* completes) specifies the job that is to be transmitted next. Specifically, whenever the number of delay-sensitive jobs in the place *delay_buff* is greater than or equal to the num-

Gate	*Definition*

ig_trigger	Predicate
	((*MARK*(*erlang_phases*) == 0) &&
	((*MARK*(*delay_buff*) > 0) ||
	(*MARK*(*loss_buff*) > 0)))
	Function
	MARK(*erlang_phases*) =
	GLOBAL_S(*STAGES*);
	/* *Schedule the job* */
	if (*MARK*(*delay_buff*) >= *MARK*(*loss_buff*)) {
	MARK(*delay_transmission*) = 1;
	MARK(*delay_buff*) −−; }
	else {
	MARK(*loss_buff*) −−;
	MARK(*delay_transmission*) = 0; }

Table 1: Input gate definition for *ig_trigger* in the SAN model.

ber of loss-sensitive jobs in *loss_buff*, a delay-sensitive job is given service.

Reward variables are also specified at the SAN level; for more information see [15]. In particular, to derive the average buffer occupancy of delay-sensitive traffic, a reward equal to the number of tokens in *delay_buff* is defined (adding one if the job currently transmitted is a delay-sensitive job, i.e., if $MARK(delay\_transmission) == 1$). Similarly, to compute the fraction of time the buffer for loss-sensitive jobs is full, a reward of 1 is defined whenever the buffer reaches capacity.

5 Results

Given the SAN model and reward structure, we derive numerically the asymptotic bias and the observation period needed to achieve some desired accuracy and precision. We will discuss the asymptotic variance and the length of the observation period first, and then discuss issues related to the asymptotic bias and the initial transient period.

5.1 Observation Period

For the delay architecture, we present results for the observation period when estimating the average buffer occupancy. While known analytic approximations assume infinite buffers, we are able to investigate the length of the observation period when the buffer is finite and provide exact results. For the loss architecture, we compute the observation period for the fraction of time the buffer is full. We see that for this measure the observation period dramatically increases when the blocking probability becomes small, a widely investigated phenomenon known as the rare event problem (e.g., [12, 13]). For the hybrid architecture, we investigate the influence on the observation period of mixed traffic sources and priority scheduling. We will see that, because the observation period is influenced by both the asymptotic variance and the magnitude of the estimate, a prediction of the necessary simulation time often cannot be based on simple rules. The desired insight requires a more exact computation, as is possible with our approach.

Variable	Symbol	Expression
Erlang phases	k	–
Service rate per phase	$k\mu$	–
Arrival rate	λ	–
On-off rate	α	–
Off-on rate	β	–
Number of sources	N	–
Buffer size	B	–
Burst length	l	$\frac{1}{\alpha}$
Fraction of sources on	–	$\frac{N\beta}{\alpha+\beta}$
Activity fraction	A	$\frac{\beta}{\alpha+\beta}$
Load	ρ	$\frac{N\lambda}{\mu}\frac{\beta}{\alpha+\beta}$

Table 2: Parameters in the SAN model

Throughout the section, we report the observation period required to achieve a confidence interval with a half width of 10% of the expected value at a 90% level, (i.e., we present t_{rel} in (6) with $z = 1.65$ and $\epsilon = 0.1$). For other levels of accuracy, the observation period can be directly derived by appropriate scaling. We also take $\mu = 1$ throughout the section, which implies that the observation period is expressed in time units corresponding to the average transmission time of jobs. The parameters used in creating variants of the three models are given in Table 2. Specific values for each of these parameters will be given for each experiment.

Average buffer occupancy in the delay architecture We first derive the observation period to estimate the average buffer occupancy in the delay architecture. For this architecture, we are able to investigate the influence of the finite buffer capacity on the observation period. Previously known analytic approximation approaches for analyzing more complicated queueing models only apply to models with infinite buffers.

In the experiment, the parameter values in the model are based on those typical for delay-sensitive traffic such as voice applications (e.g., [9]). We consider the case with a load of $\rho = 0.8$, with activity fraction $A = 0.4$, burst length $l = 5$, arrival rate $\lambda = 0.2$, and exponentially distributed transmission time with rate $\mu = 1$. If one wishes the blocking probability to be less than 10^{-5} with these parameter values and varies the number of sources $N = 1, 5$ and 10, buffer sizes of $B = 180, 70$ and 60 are required, respectively.

Figure 2 presents the observation period for varying load ρ (we varied ρ by varying λ, keeping the burst length and the activity fraction fixed) and for $N = 1, 5$ and 10 traffic sources. Furthermore, the result for an M/M/1/60 queue has been computed. The capacity of 60 was chosen for this queue, since this is what was

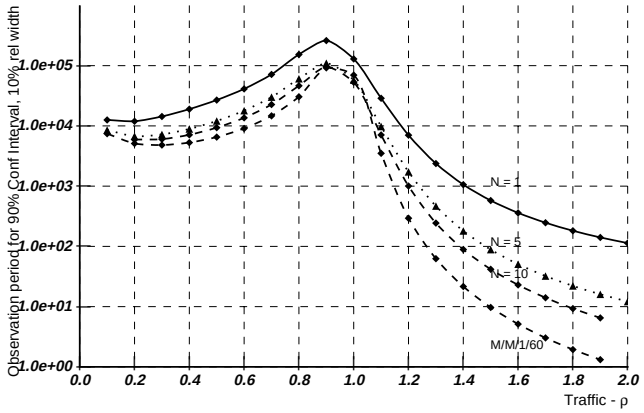

Figure 2: Observation period for the average buffer occupancy in the delay architecture.

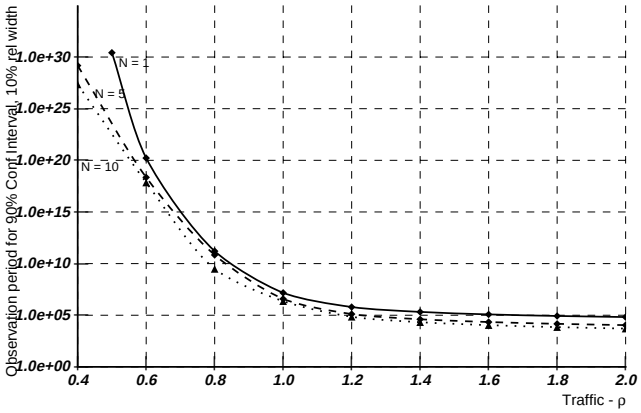

Figure 3: Observation period for the fraction of time the buffer is full in the loss architecture.

Erlang phases	$N = 1$	$N = 5$	$N = 10$
$k = 1$	1.53e+05	6.16e+04	4.78e+04
$k = 5$	1.39e+05	4.82e+04	3.32e+04
$k = 10$	1.38e+05	4.63e+04	3.13e+04

Table 3: Observation period for different number of Erlang phases in transmission delay.

used for $N = 10$ IPP sources. The curves for different numbers of sources in Figure 2 exhibit similar trends: an increasing length of the observation period for ρ roughly between 0.2 to 0.9, while it decreases for $\rho > 0.9$.

Similarly, we can investigate how the transmission time distribution influences the observation period. To do this, we increase the number of Erlang phases in the transmission time distribution from $k = 1$ to $k = 5$ and $k = 10$. Each of the k phases has rate $k\mu$, for $k = 1, 5$ and 10, thus leaving the load ρ of the system unaltered. In Table 3, we see that the observation period decreases when the number of phases k increases, a fact that was to be expected. Note, however, the relative small influence on the observation period compared to the results in Figure 2. In words, it says that for the considered model, changing the exponential service time distribution to a more deterministic one influences the observation period less than changing the exponential inter-arrival time distribution into a more bursty one.

Probability of full buffer in the loss architecture For the loss architecture, we are interested in the fraction of time the buffer is full, as this is closely related to the loss probability of jobs. In the presented results, we observe that the observation period increases excessively when the load of the system decreases. We will see that this is caused by the well-known "rare-event problem" (e.g., [12, 13]).

The parameter values used for the loss architecture are as follows. Compared to the delay architecture, the loss probability should be lower, while the traffic pattern can be expected to be more bursty (e.g., in case of file transfer). Following results in [9], we require that the blocking probability be less than 10^{-7}. For a load of $\rho = 0.8$, activity fraction $A = 0.2$, burst length $l = 10$, and arrival rate $\lambda = 0.4$, buffer sizes $B = 1600, 380$ and 220 are required, for number of sources $N = 1, 5$, and 10, respectively.

Figure 3 clearly shows the rare event problem when the load decreases (we varied ρ by varying λ). This problem occurs because $t_{rel} \doteq \frac{\sigma^2_\infty \gamma(z)}{(\mathbf{E}R)^2 \epsilon^2}$ apparently increases in an unbounded fashion when the denominator becomes small. So, in contrast to the observation period for the average buffer occupancy, the trends in the curve are not determined by the variability in the model but by the magnitude of the measure (see [18] for a more detailed discussion). In our discussion of the results in the hybrid architecture, we will discuss these two effects, and the interaction between them, more closely.

In retrospect, we note that the rare event problem also appeared in the case of the delay architecture. Namely, when $\rho \downarrow 0$, the average buffer occupancy becomes small and the observation period increases (see Figure 2) due to the magnitude of the measure, not its absolute asymptotic variance.

Hybrid architecture For the hybrid architecture, the combination of the two traffic streams and the priority scheduling results in a much more intricate and larger model than for the dedicated architectures. As a consequence, the prediction by heuristic methods of the observation period is not possible. This example shows how increasing variability in the model and the decreasing magnitude of the measure often result in opposite consequences for the length of the observation period. In addition, it illustrates the potential of the method for determining the observation period in a large and complicated example whose Markov reward model has more than 150,000 states.

The blocking probabilities for each type of traffic

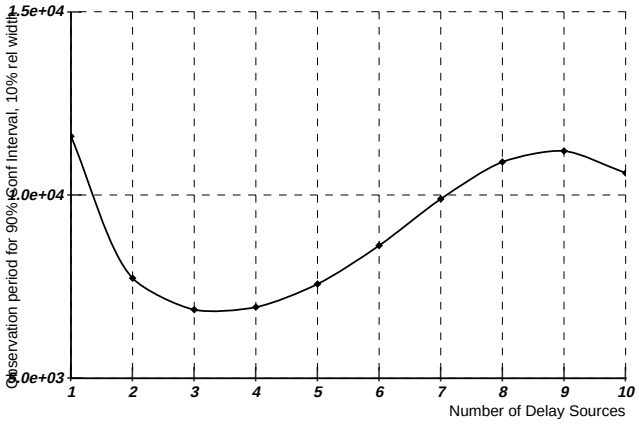

Figure 4: Observation period versus number of delay sources in the hybrid architecture.

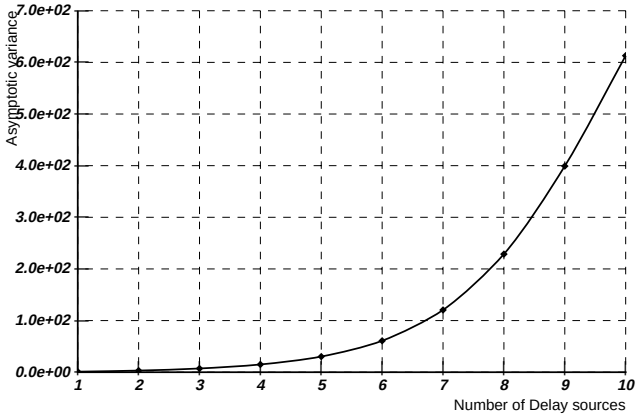

Figure 5: Asymptotic variance versus number of delay sources in the hybrid architecture.

and the model parameter values are as given previously, except that we consider a total load of $\rho = 0.8$, equally divided over 5 delay-sensitive and 5 loss-sensitive traffic sources. Given these system parameter values and required blocking probabilities, we consider buffer sizes of $B = 15$ and $B = 150$ for delay-sensitive and loss-sensitive traffic, respectively. This leads to a Markov reward model with 163,345 states. In the results presented, we discuss the average buffer occupancy for the delay-sensitive traffic for a changing number of delay-sensitive sources. We keep the total number of sources equal to 10, while the total load of the system always equals $\rho = 0.8$.

The result for the observation period is given in Figure 4. Since the observation period, given by $t_{rel} = \frac{\mathbf{AV}\gamma(z)^2}{(\mathbf{E}R)^2\epsilon^2}$, involves both the asymptotic variance and the average buffer occupancy, we give the asymptotic variance separately in Figure 5 and the average buffer occupancy in Figure 6.

Figure 5 shows that the asymptotic variance increases with the number of delay sources. Apparently, the fluctuation in the buffer occupancy of the delay architecture is increasing in the number of delay sources. In other words, despite its bursty character, the influence of the intervening loss-sensitive traffic is not so large that it disturbs the relation between the variability in the buffer occupancy and the load of delay-sensitive traffic. Figure 6 shows a similar relation between the value of the measure and the increase of the number of delay-sensitive sources. Combining the curves in Figures 5 and 6 determines Figure 4. From this a non-monotonic shape arises because the asymptotic variance and the magnitude of the measure influence the observation period differently.

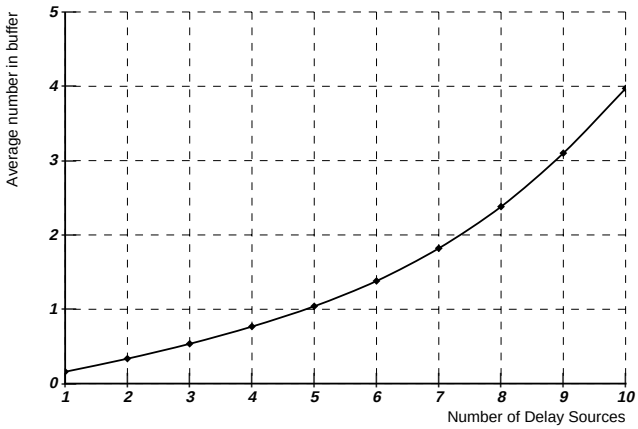

Figure 6: Expected buffer occupancy versus number of delay sources in the hybrid architecture.

We thus see that for complicated models the behavior of the observation period is hard to predict, because it depends both on the asymptotic variance and the measure of interest. Typically, only exact computation such as made possible by the method presented in this paper, can reveal whether measurement of a system or simulation of a model is feasible.

5.2 Initial Bias

We now investigate which initial state and what length of the initial transient are best suited for a discrete-event simulation run of the model. The results in this section are for the delay architecture, for a load of $\rho = 0.8$ and with other parameters as in the previous section. To investigate transient effects, we vary the number of jobs initially in the buffer.

Average buffer occupancy in the delay architecture Figure 7 gives the asymptotic bias for the average buffer occupancy in the delay architecture. Each curve in Figure 7 corresponds to the asymptotic bias for a particular number of jobs initially in the buffer, and given the initial state each curve shows results for a range of initial transient periods. For all curves, all sources are initially idle.

We see from Figure 7 that bias is minimized if the initial buffer occupancy is relatively close to the average buffer occupancy of 5.12. In fact, the curves for an initial buffer occupancy of a little above the average

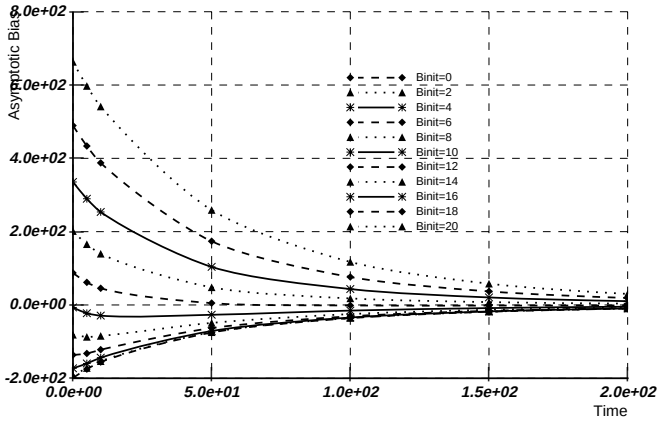

Figure 7: Asymptotic bias for different number of jobs and for all sources idle.

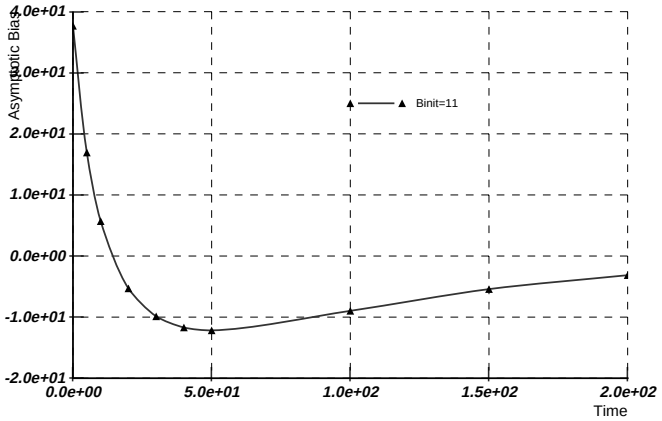

Figure 8: Asymptotic bias for 11 jobs initially in the buffer.

show the least bias. A possible explanation for this is that with all sources idle and a service rate higher than the arrival intensity, the buffer will initially be drained fast. To make up for this, an initial buffer occupancy higher than the long-run average is preferred.

Not all curves in Figure 7 go monotonically to zero, in particular not the curves for 8 and 10 jobs. To investigate this more closely, Figure 8 shows the asymptotic bias when 11 jobs are initially present in the queue. For small values of t, the asymptotic bias decreases rapidly, since all sources are idle. As a consequence, there is a value, around $t = 20$, for which the estimator becomes unbiased. Starting the simulation with the initial distribution corresponding to the state distribution at $t = 20$ would remove all bias and is therefore recommended from this point of view. Besides bias, it is, however, also important to consider which method is used to compute confidence intervals. If the batch means method is used, one might prefer to take a value for t such that the results for individual batches are assured to be close to steady-state behavior (since that is an assumption in the batch-means method). On the other hand, if independent replicates are used, $t = 20$ forms the optimal choice for the initial transient period.

We now discuss how to determine if the initial transient is sufficient and optimal. The problem can be stated as follows: is the cost of extending the initial transient outweighed by the obtained bias reduction? We will discuss two criteria, both with their own merits.

The first method is to minimize the sum of the length of the initial transient and the time needed for the remaining bias to become sufficiently small. It turns out that a criterion based on the transient expected reward can be found that optimizes the length of the initial transient. To obtain this result, let t_1 and $t_2 > t_1$ be two possible values for the initial transient. Assume that we want the observation period so long that the resulting bias is less than $\epsilon \mathbf{E}R$. It then follows from (4) that the observation periods should equal $s_1 = |\mathbf{B}(t_1)|/\epsilon\mathbf{E}R$ and $s_2 = |\mathbf{B}(t_2)|/\epsilon\mathbf{E}R$, respectively (we assume in what follows that t_1 is large enough to assure that bias decreases monotonically, i.e., $s_2 < s_1$).

It is worthwhile increasing the length of the initial transient period from t_1 to t_2 if

$$t_2 + s_2 < t_1 + s_1 \Rightarrow t_2 - t_1 < \frac{|\mathbf{B}(t_1)| - |\mathbf{B}(t_2)|}{\epsilon\mathbf{E}R}$$

$$\Rightarrow \frac{|\mathbf{B}(t_2) - \mathbf{B}(t_1)|}{t_2 - t_1} > \epsilon\mathbf{E}R. \qquad (13)$$

We will now use the following result, which can be derived directly from the definition of asymptotic bias:

$$\frac{d\mathbf{B}(t)}{dt} = -(\mathbf{E}R(t) - \mathbf{E}R). \qquad (14)$$

Hence, if we let $t_2 \to t_1$, criterion (5.2) becomes

$$|\frac{d\mathbf{B}(t)}{dt}| > \epsilon\mathbf{E}R \Rightarrow |\mathbf{E}R(t) - \mathbf{E}R| > \epsilon\mathbf{E}R. \qquad (15)$$

In other words, the optimal initial transient period is when $|\mathbf{E}R(t) - \mathbf{E}R| = \epsilon\mathbf{E}R$. As an example, for the case represented in Figure 8, if $\epsilon = 0.001$, the optimal value of t is about $t = 400$.

The above criterion is useful if the time it takes for the bias to diminish is larger than the observation period needed to obtain the desired width of the confidence interval. However, often the length of the observation period needed to reach the desired precision is so high that the bias is diminished, largely independent of the length of the initial transient. In that case, if a simulation with an observation period of length s is carried out, the resulting deviation from the true expected reward is given by $\mathbf{B}/s$. Again, the question arises, is it worthwhile to spend simulation time on a longer initial transient and decrease the bias?

A similar criterion as above can be derived if we decide to increase the initial transient by k time units if it decreases the bias by $\epsilon\mathbf{E}R$. In other words, we are willing to increase the initial transient from t_1 to t_2 if

$$|\frac{\mathbf{B}(t_1)}{s} - \frac{\mathbf{B}(t_2)}{s}| > \epsilon\mathbf{E}R\frac{t_2 - t_1}{k}.$$

Similar to above, letting $t_2 \to t_1$, the optimal value of t can be shown to obey

$$|\mathbf{E}R(t) - \mathbf{E}R| = \epsilon \mathbf{E}R\frac{s}{k}. \quad (16)$$

For the example of Figure 8, the length of the observation period is $s = 5 \times 10^{-4}$, as given in Figure 2. If we are willing to spend $k = 1000$ time units of simulation provided the bias decreases by at least 0.001 $\mathbf{E}R$, then the optimal length of the initial transient period is about $t = 250$, while, if we are willing to spend $k = 10000$ time units, the optimal initial transient is $t = 450$.

6 Conclusion

This paper presents an efficient algorithm for the computation of the asymptotic bias and variance of measures defined on Markov reward models. The asymptotic bias and variance are important for establishing the initial transient and the observation period in steady-state discrete-event simulation. Previously known methods to evaluate the asymptotic bias and variance are restricted to either specific classes of models and measures, such as simple queueing models in [19], or are only applicable to relatively small Markov reward models, because their formulation involves a matrix inverse, e.g., [5]. The proposed solution algorithm applies to a very general class of Markov reward models and does not require matrix inversions for its solution. It thus has wide applicability and, because of its implementation as a solver in *UltraSAN*, can be used on models with hundreds of thousands of states.

To demonstrate the applicability of the method, we analyzed the observation period for two measures defined on a queueing model with different classes of customers, arriving according to interrupted Poisson processes, and with a priority scheduling discipline. The measures considered were the average buffer occupancy and the fraction of time the buffer is full, neither of which have known analytical solutions. The results we obtained for the model illustrated both the difficulty of heuristically predicting the observation period for intricate models and the ease of computing the exact observation period, using the proposed algorithm. Furthermore, it was discussed how to determine the optimal length of the initial transient period using the computed results.

References

[1] W. G. Cochran, *Sampling Techniques*, John Wiley, New York, third edition, 1977.

[2] P. W. Glynn, "Some asymptotic formulas for Markov chains with applications to simulation," *J. Statist. Comput. Simul*, vol. 19, pp. 97–112, 1984.

[3] G. H. Golub and C. F. van Loan, *Matrix Computations*, John Hopkins, Baltimore, Maryland, 1983.

[4] W. K. Grassmann, "Initial bias and estimation error in discrete event simulation," in *Winter Simulation Conference*, M. Highland, Chao, editor, pp. –377–384, 1982.

[5] G. B. Hazen and A. A. B. Pritsker, "Formulas for the variance of the sample mean in finite state Markov processes," *J. Statist. Comput. Simul.*, vol. 12, pp. 25–40, 1980.

[6] R. A. Howard, *Dynamic Probabilistic Systems*, volume I and II of *Series in Decision and Control*, Wiley, 1971.

[7] J. Keilson, *Markov Chain Models - Rarity and Exponentiality*, Springer-Verlag, New York, 1979.

[8] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, New York, second edition, 1991.

[9] S. Q. Li, "A new performance measurement for voice transmission in burst and packet switching," *IEEE Transactions on Communications*, vol. 35, no. 10, pp. 1083–1094, October 1987.

[10] J. F. Meyer, A. Movaghar, and W. H. Sanders, "Stochastic activity networks: Structure, behavior and applications," in *International Conference on Timed Petri Nets*, pp. 106–115, Torino, Italy, July 1985.

[11] A. Movaghar and J. F. Meyer, "Performability modeling with stochastic activity networks," in *Real-Time Systems Symposium*, Austin, TX, December 1984.

[12] V. F. Nicola, M. K. Nakayama, P. Heidelberger, and A. Goyal, "Fast simulation of dependability models with general failure, repair and maintenance processes," in *20th International Symposium on Fault-Tolerant Computing*, pp. 491–498, 1990.

[13] S. Parekh and J. Walrand, "A quick simulation method for excessive backlogs in networks of queues," *IEEE Transactions on Automation Control*, vol. 34, no. 1, pp. 54–66, 1989.

[14] J. F. Reynolds, "The covariance structure of queues and related processes - a survey of recent work," *Advances in Applied Probability*, vol. 7, pp. 383–415, 1975.

[15] W. H. Sanders and J. F. Meyer, "A unified approach for specifying measures of performance, dependability, and performability," in *Dependable Computing for Critical Applications*, A. Avizienis and J. Laprie, editors, volume 4 of *Dependable Computing and Fault-Tolerant Systems*, pp. 215–238, Springer Verlag, Wien, 1991.

[16] W. H. Sanders, W. D. Obal, M. A. Qureshi, and F. K. Widjanarko, "The *UltraSAN* modeling environment," *Performance Evaluation*, vol. 24, no. 1, pp. 89–115, 1995.

[17] W. J. Stewart and A. Goyal, "Matrix methods in large dependability models," RC 11485, IBM research, 1985.

[18] A. P. A. van Moorsel, *Performability Evaluation Concepts and Techniques*, PhD thesis, University of Twente, The Netherlands, 1993.

[19] W. Whitt, "Planning queueing simulations," *Management Science*, vol. 35, no. 11, pp. 1341–1366, 1989.