DTIC FILE COPY

ARO 25839.34-M

AD-A227 879

EXPERIMENTS WITH INITIAL TRANSIENT DELETION FOR PARALLEL, REPLICATED STEADY-STATE SIMULATIONS

by

Peter W. Glynn
Philip Heidelberger

TECHNICAL REPORT No. 59

May 1990

Prepared under the Auspices
of
U.S. Army Research Contract
DAAL-03-88-K-0063

DTIC
ELECTE
OCT 24 1990
S E D

Approved for public release: distribution unlimited.

Reproduction in whole or in part is permitted for any purpose of the United States government.

DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

90 10 24 005

Experiments with Initial Transient Deletion for Parallel, Replicated Steady-State Simulations

Peter W. Glynn*

Department of Operations Research
Stanford University
Stanford, California 94305

Philip Heidelberger

IBM Research Division
T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, New York 10598

Abstract: A simple and effective way to exploit parallel processors in discrete event simulations is to run multiple independent replications, in parallel, on multiple processors and to average the results at the end of the runs. We call this the method of parallel replications. This paper is concerned with using the method of parallel replications for estimating steady-state performance measures. We report on the results of queueing network simulation experiments that compare the statistical properties of several possible estimators that can be formed using this method. The theoretical asymptotic properties of these estimators were determined in Glynn and Heidelberger (1989a and 1989b). Both the theory and the experimental results reported here strongly indicate that a nonstandard (in the context of steady-state simulation), yet easy to apply, estimation procedure is required on highly parallel machines. This nonstandard estimator is a ratio estimator. The experiments also show that use of the ratio estimator is advantageous even on machines with only a moderate degree of parallelism.

Keywords: initial transient deletion, parallel simulation, steady-state, ratio estimation, discrete-event simulation

*Research supported by the IBM Corporation under SUR-SST Contract 12480042 and by the U.S. Army Research Office under contract number DAAL-03-88-K-0063.

A-1

1. Introduction

A simple and effective way to exploit parallel processors for computationally intensive discrete event simulations is to run multiple independent replications, in parallel, on multiple processors and to average the results at the end of the runs. We call this the method of parallel replications. This paper is concerned with using the method of parallel replications for estimating steady-state performance measures. In particular, we report on the results of queueing network simulation experiments that compare the statistical properties of several possible estimators that can be formed using this method. The theoretical asymptotic properties of these estimators were determined in Glynn and Heidelberger (1989a and 1989b). Both the theory and the experimental results reported here strongly indicate that a nonstandard (in the context of steady-state simulation), yet easy to apply, estimation procedure is required on highly parallel machines. This nonstandard estimator takes the form of a ratio estimator. The experiments also show that use of the ratio estimator is advantageous even on machines with only a moderate degree of parallelism.

We remark that an alternative approach to parallel processing of simulations is distributed simulation, in which multiple processors cooperate together to generate a single realization of the stochastic process being simulated. For an excellent introduction to distributed simulation and a thorough bibliography on this topic, see Fujimoto (1989). A theoretical comparison of the statistical efficiencies of parallel replications and distributed simulation for estimating steady-state parameters may be found in Heidelberger (1986).

Intuitively, when using the method of parallel replications on a large number of processors, one expects to get highly accurate estimates after only a relatively short amount of time. However, there are some potentially serious statistical problems inherent in this approach, and careful estimation procedures must be applied in order to obtain estimates with the proper (or desired) statistical properties. These problems basically arise because any bias effects are magnified on highly parallel machines, i.e., because of the bias, one obtains highly accurate estimates of the wrong quantity.

In the context of estimating transient performance measures (or steady-state performance measures in regenerative simulations), these problems have been identified and addressed in Heidelberger (1988) and Glynn and Heidelberger (1990). These papers show that nonstandard estimators are required on highly parallel machines. Other issues related to parallel replications for estimating transient quantities are described in Bhavsar and Isaac (1987).

For estimating steady-state performance measures, the traditional approaches (on a single processor) are to use either the method of batch means, or independent replications with initial transient deletion (see, e.g., Law (1977), Law and Carson (1979), Law and Kelton (1982), or

Bratley, Fox and Schrage (1987)). When using replications, it is generally advised to use only "a few long" replications (say 10 to 20) with deletion to reduce susceptibility to the effects of initialization bias.

With the prospect of parallelism as motivation, Glynn and Heidelberger (1989a and 1989b) have addressed, from a theoretical point of view, how one should control the number of replications (processors), the length of each replication, and the length of the initial transient deletion interval in order to obtain valid central limit theorems for steady-state parameters. Such central limit theorems can then be used as the basis for confidence interval formation. These papers, which extend the single processor results of Glynn (1987 and 1990), show that valid confidence intervals can be obtained even for a very large number of processors P (relative to the replication length) provided the deletion interval grows appropriately *and* the proper (nonstandard) ratio estimator is used.

On the other hand, if each processor is run for a prespecified amount of computer time c, then it was shown that initial transient deletion does not, in fact, remove the dominant term in the bias expansion (i.e., the term of order $1/c$) of the traditional (standard) independent replications estimator, $\alpha_T(P, c)$. In this case, the amount of simulated time generated by each processor is a random variable (rv) and thus the traditional estimator becomes a ratio estimator. The bias expansion of this estimator reveals two sources of bias of order $1/c$:

1. "Initialization" bias, i.e., bias essentially due to the simulation not being started in steady-state conditions.
2. "Ratio" bias, i.e., bias due to the fact that the denominator of the ratio estimator is a rv.

When done appropriately, initial transient deletion effectively removes the initialization bias. However, initial transient deletion does not remove the ratio bias. The nonstandard estimator, $\alpha_R(P, c)$, corresponds to the classical ratio estimator which is typically used in sample surveys (see, e.g., Cochran (1963)) and regenerative simulations (see, e.g., Crane and Iglehart (1975) or Iglehart (1975)). The initialization bias (of order $1/c$) in $\alpha_R(P, c)$ is the same as the initialization bias in $\alpha_T(P, c)$, but the ratio bias in $\alpha_R(P, c)$ is P times smaller than the ratio bias in $\alpha_T(P, c)$. Thus initial transient deletion effectively removes all bias of order $1/c$ from $\alpha_R(P, c)$.

The net effect of this analysis is that, when using the ratio estimator $\alpha_R(P, c)$, valid confidence intervals for steady-state parameters can be formed when very highly parallel machines (large P) are run for a relatively short amount of time (small c/P). In this situation, valid confidence intervals are not obtained when estimating the steady-state parameter by the traditional estimator $\alpha_T(P, c)$. Using $\alpha_T(P, c)$, valid confidence intervals are only obtained when c/P is very large, i.e., when the length of each replication is large with respect to the number of processors.

We emphasize that $\alpha_T(P, c)$ and $\alpha_R(P, c)$ both make use of exactly the same underlying data: they merely average these data differently.

The purpose of this paper is to demonstrate, experimentally, that this dramatic difference between the theoretical asymptotic behaviors of these two estimators is exhibited in sample sizes that are not unreasonable in practice. In simulations of simple queueing systems, we show that noticeable effects (increased bias and decreased confidence interval coverage) are present on as few as 32 to 64 processors. Severe effects are observed on 128 or more processors. Thus, from both a theoretical and practical viewpoint, the traditional estimator, $\alpha_T(P, c)$, should be avoided on even moderately sized parallel processors.

The rest of the paper is organized as follows. In Section 2, we summarize the relevant theoretical results from Glynn and Heidelberger (1989a and 1989b) and Glynn (1987 and 1990). In Section 3, we describe the queueing models that we used for experimentation. In Section 4, we describe the design of the experiments, including point and interval estimation procedures. The results of the simulation experiments are presented in Section 5. Finally, Section 6 contains a summary of our findings, a discussion of their relevance to traditional steady-state estimation on single processor systems, and an indication of related future research topics.

2. Summary of Theoretical Results

The results that we quote from Glynn and Heidelberger (1989a and 1989b) and Glynn (1987 and 1990) were derived under reasonable, yet fairly technical assumptions. These basically involve assumptions concerning the existence of central limit theorems and their associated uniform integrability (i.e., moment convergence in the central limit theorem), an exponential convergence rate to the steady-state distribution, and certain boundedness conditions. Since a precise statement of these conditions would be rather tedious (and not particularly illuminating for the present purposes), we will make the simplifying assumption that the process being simulated is an irreducible, finite state space, continuous time Markov Chain (CTMC) with state space denoted by $\mathbf{E}$. Such processes automatically satisfy all of the necessary assumptions.

We let $\{X(s), s \geq 0\}$ denote the CTMC. The parameter s denotes simulated time so that $X(s)$ is the state of the process at simulated time s. There then exists a rv X such that $X(s) \Rightarrow X$ where $\Rightarrow$ denotes convergence in distribution. We call X the steady-state distribution and we shall be interested in estimating quantities of the form $\alpha = \mathrm{E}[f(X)]$ for some function f.

There are P processors. Simultaneously an independent simulation of the CTMC is started on each processor. We let $X_i(s)$ denote the state of the process at simulated time s on processor i, $i = 1, \ldots, P$. Let $T_i(c)$ denote the simulation time on processor i after c units of computer time. (The discussion in this paper also holds if c is measured in units of "wall clock" time, or for that

matter, any other way of measuring time.) Let $C_i(s)$ denote the amount of computer time required on processor i to obtain s units of simulated time.

There are a variety of ways to set the run length. We will consider two reasonable and practical approaches. In the first approach, a fixed amount of simulated time, say t_P, is generated on each processor. In this case, the completion time of the simulation experiment is $C(t_P) = \max\{C_1(t_P), \ldots, C_P(t_P)\}$, which is a rv. Since we can view $\{C_i(s), s \geq 0\}$ as a cumulative process, it is reasonable to assume that each $C_i(t_P)$ obeys a central limit theorem, i.e. there exist finite positive constants λ and σ_1 such that

$$\frac{C_i(t_P) - t_P/\lambda}{\sqrt{t_P}} \Rightarrow \sigma_1 N(0,1) \tag{2.1}$$

where $N(0,1)$ denotes a normally distributed rv with mean zero and variance one. The parameter λ^{-1} is the long run rate at which computer time is expended per unit of simulated time. Alternatively, λ is the long run rate at which simulated time is generated per unit of computer time. Since the completion time is the maximum of iid (independent and identically distributed) rvs that are approximately normally distributed, the expected completion time is approximately equal to $(t_P/\lambda) + \sigma_1\sqrt{2t_P \ln(P)}$ provided t_P and $P \to \infty$ appropriately. In this expression, (t_P/λ) is the expected completion time of an individual processor and $\sigma_1\sqrt{2t_P \ln(P)}$ is the additional time until the last processor finishes. The factor $\sqrt{2\ln(P)}$ arises as the maximum of P iid N(0,1) rvs, which then gets multiplied by the standard deviation of an individual completion time, $\sigma_1\sqrt{t_P}$. Notice that if $\sigma_1 = 0$, then $C_i(t_P) \equiv t_P/\lambda$, i.e., computer time is deterministically proportional to simulated time and there is no completion time penalty. Since the holding time in a state (in simulated time units) is a rv and since the amount of work (computer time) to generate a transition may depend on the state of the system (e.g., the time to put an event on the future event list typically grows with the length of the list), we view such proportionality as the exception, rather than the rule. Thus, in general, the completion time penalty grows as $\sqrt{2t_P \ln(P)}$ which is clearly undesirable.

In the second approach, we stop each simulation at exactly the same computer time, c. In this approach, the completion time of the experiment is deterministic, but the amount of simulated time generated on each processor, $T_i(c)$, is now a rv. Note that $T_i(c) = \sup\{ t \geq 0 : C_i(t) \leq c\}$, so that $\{T_i(c), c \geq 0\}$ is the inverse process of $\{C_i(t), t \geq 0\}$. We will assume that $C_i(t)$ can be represented as an integral, i.e.,

$$C_i(t) = \int_0^t \chi(X_i(s))\, ds \qquad \text{where } 0 < \chi(j) < \infty \text{ for all } j \in \mathbf{E}. \tag{2.2}$$

It is well known that such CTMCs satisfy a bivariate central limit theorem:

$$\sqrt{c}\left(\frac{\int_0^{T_i(c)} f(X_i(s))\, ds}{T_i(c)} - \alpha \;,\; \frac{T_i(c)}{c} - \lambda \right) \Rightarrow \mathbf{N(0,A)} \tag{2.3}$$

where **N(0,A)** denotes a bivariate normally distributed random vector with means zero and covariance matrix **A**. In fact, a slightly stronger version of this central limit theorem is valid (and required), namely a functional central limit theorem version of Equation 2.3. See Billingsley (1968) for a discussion of functional central limit theorems. In practice, this is not a restriction.

We next assume that all the data generated in the first $\kappa(c)$ units of computer time are deleted for the purpose of reducing initialization bias. We assume that $\kappa(c)$ is a deterministic quantity. Define $\gamma_i(c) \equiv T_i(\kappa(c))$ to be the (random) simulated time at which processor i begins collecting data for steady-state estimation and let

$$Y_i(c) \;\equiv\; \int_{\gamma_i(c)}^{T_i(c)} f(X_i(s))\, ds \;. \tag{2.4}$$

The length of the interval in which it is assumed that the process is "in steady-state" is then $\tau_i(c) \equiv T_i(c) - \gamma_i(c)$. The traditional steady-state simulation estimator of α (assuming we were simulating on a single processor and associating the index i with an independent replication) is

$$\alpha_T(P, c) \;\equiv\; \frac{1}{P} \sum_{i=1}^{P} \frac{Y_i(c)}{\tau_i(c)} \;. \tag{2.5}$$

The above estimator is employed in most simulation packages and languages when steady-state estimation is performed using the method of independent replications with initial transient deletion. However, the ratio form of $\alpha_T(P, c)$ immediately suggests an alternative estimator, which is more suitable for ratio estimation:

$$\alpha_R(P, c) \;\equiv\; \frac{\sum_{i=1}^{P} Y_i(c)}{\sum_{i=1}^{P} \tau_i(c)} \;=\; \frac{\overline{Y}(c)}{\overline{\tau}(c)} \tag{2.6}$$

where $\overline{Y}(c) = (1/P) \sum_{i=1}^{P} Y_i(c)$ and $\overline{\tau}(c) = (1/P) \sum_{i=1}^{P} \tau_i(c)$. Other simulation contexts in which such ratio estimators have been considered are regenerative simulation (identify $\tau_i(c)$ and $Y_i(c)$ as the

length of the i-th regenerative cycle and an integral over the i-th cycle, respectively) and the method of batch means (identify $\tau_i(c)$ and $Y_i(c)$ as the length of the i-th batch and an integral over the i-th batch, respectively - see Fox and Glynn (1987)).

We begin by stating bias expansions for $\alpha_T(P, c)$ and $\alpha_R(P, c)$ when $c = c_P$ and we let P and $c_P \to \infty$ together. We first consider the case when no initial transient deletion is performed, i.e., $\kappa(c_P) = \gamma_i(c_P) = 0$. The conditions stated above are sufficient to guarantee the following asymptotic bias expansions:

$$\begin{aligned} E[\alpha_T(P, c_P)] &= \alpha + \frac{b_T}{c_P} + o(1/c_P) \\ E[\alpha_R(P, c_P)] &= \alpha + \frac{b_R}{c_P} + o(1/c_P) \end{aligned} \tag{2.7}$$

where

$$\begin{aligned} b_T &= a - \frac{A_{12}}{\lambda} \\ b_R &= a \,. \end{aligned} \tag{2.8}$$

The expansion for $\alpha_T(P, c)$ was derived in Glynn (1990) while the expansion for $\alpha_R(P, c)$ was derived in Glynn and Heidelberger (1989b). The precise form of the constant a is given in these papers. Roughly speaking, we can think of a as "initialization" bias, i.e., bias because

$$E[\int_0^{T_i(c)} (f(X_i(s)) - \alpha)\, ds] \neq 0 \,. \tag{2.9}$$

The traditional estimator contains an extra bias term, $-A_{12}/\lambda$, which can be thought of as ratio bias, i.e., bias because the denominator of the ratio is a rv.

To see why the bias expansions of $\alpha_T(P, c)$ and $\alpha_R(P, c)$ differ, we give the following brief heuristic arguments (which are made rigorous in the above mentioned papers). Notice that both $E[\alpha_T(P, c_P)]$ and $E[\alpha_R(P, c_P)]$ can be written as $E[A(c_P)/B(c_P)]$. Now let $\varepsilon(c_P) = (B(c_P) - E[B(c_P)])/E[B(c_P)]$ and write

$$\frac{A(c_P)}{B(c_P)} = \frac{A(c_P)}{E[B(c_P)]\,(1 + \varepsilon(c_P))} \approx \frac{A(c_P)\,(1 - \varepsilon(c_P) + \varepsilon(c_P)^2 \ldots)}{E[B(c_P)]} \tag{2.10}$$

Taking expectations of Equation 2.10 yields

$$E\left[\frac{A(c_P)}{B(c_P)}\right] = \frac{E[A(c_P)]}{E[B(c_P)]} - \frac{Cov[A(c_P),B(c_P)]}{E[B(c_P)]^2} + \ldots \tag{2.11}$$

For both $\alpha_T(P, c_P)$ and $\alpha_R(P, c_P)$, the initialization bias term a arises from the fact that $E[A(c_P)]/E[B(c_P)] = E[Y_i(c_P)]/E[\tau_i(c_P)] \neq \alpha$. The ratio bias arises from the covariance term in Equation 2.11. For $\alpha_T(P, c_P)$, $E[B(c_P)] = E[\tau_i(c_P)] \approx \lambda\, c_P$ and $Cov[A(c_P),B(c_P)] = Cov[Y_i(c_P), \tau_i(c_P)] \approx c_P\, \lambda\, A_{12}$ by the central limit theorem in Equation 2.3 (and its uniform integrability). Thus the ratio bias for $\alpha_T(P, c_P)$ is $-A_{12}/(\lambda c_P)$ as stated. For $\alpha_R(P, c_P)$, the ratio bias is reduced by a factor of P since

$$Cov[A(c_P),B(c_P)] = Cov[\overline{Y}(c_P), \overline{\tau}(c_P)] = \frac{Cov[Y_i(c_P), \tau_i(c_P)]}{P} \approx \frac{c_P\, \lambda\, A_{12}}{P}. \tag{2.12}$$

Combining Equations 2.11, 2.12 and the expression for $E[B(c_P)]$ shows that the ratio bias of $\alpha_R(P, c_P)$ is $O(1/Pc_P)(= o(1/c_P)$ as $P \to \infty)$.

The effect of the bias expansions of Equation 2.7 is that, without deletion, $\alpha_T(P, c_P)$ and $\alpha_R(P, c_P)$ obey the following central limit theorems:

Theorem 1

Let $\{X_i(s), s \geq 0\}$ be iid samples of an irreducible, finite state space CTMC satisfying Equations 2.2 and 2.3. Define $\sigma^2 = A_{11}$ and let $\kappa(c_P) = \gamma_i(c_P) = 0$. As $P \to \infty$,

1. If $P/c_P \to \infty$, $c_P \to \infty$ and $b_T \neq 0$, then $\sqrt{Pc_P}\,|\alpha_T(P, c_P) - \alpha| \Rightarrow \infty$
2. If $P/c_P \to m$ $(0 < m < \infty)$ and $b_T \neq 0$, then $\sqrt{Pc_P}\,(\alpha_T(P, c_P) - \alpha) \Rightarrow N(0, \sigma^2) + b_T m^{1/2}$.
3. If $P/c_P \to 0$, then $\sqrt{Pc_P}\,(\alpha_T(P, c_P) - \alpha) \Rightarrow N(0, \sigma^2)$

Theorem 2

Theorem 1 is also valid for $\alpha_R(P, c_P)$ with b_R replacing b_T.

Theorems 1 and 2 imply that, without deletion, one must let $P/c_P \to 0$ in order to obtain valid confidence intervals for α, i.e., the length of each replication must be large with respect to the number of replications (processors).

We next consider the case of asymptotically negligible deletion, i.e., $\kappa_P(c_P) \to \infty$ but $\kappa_P(c_P)/c_P \to 0$. In this case, it is shown in Glynn and Heidelberger (1989b) that

$$\begin{aligned} E[\alpha_T(P, c_P)] &= \alpha + \frac{d_T}{c_P} + o(1/c_P) \\ E[\alpha_R(P, c_P)] &= \alpha + o(1/c_P) \end{aligned} \tag{2.13}$$

where

$$d_T = -\frac{A_{12}}{\lambda} . \tag{2.14}$$

Equations 2.13 and 2.14 imply that, for $\alpha_T(P, c_P)$, initial transient deletion is effective in removing initialization bias, but does not remove ratio bias (unless $A_{12} = 0$ in which case simulated time and computer time are deterministically proportional). The effect of this bias expansion on the central limit theorem for $\alpha_T(P, c_P)$ is that valid confidence intervals will, again, only be obtained if $P/c_P \to 0$. On the other hand, initial transient deletion removes all sources of bias of order $1/c_P$ from the bias expansion of $\alpha_R(P, c_P)$. This will permit a valid central limit theorem for $\alpha_R(P, c_P)$ even if $P/c_P \to \infty$ provided the length of the deletion interval does not grow too slowly.

Theorem 3

Let $\{X_i(s), s \geq 0\}$ be iid samples of an irreducible, finite state space CTMC satisfying Equations 2.2 and 2.3. Assume $\kappa_P(c_P)/c_P \to 0$. As $P \to \infty$,

1. If $P/c_P \to \infty$, $c_P \to \infty$ and $d_T \neq 0$, then $\sqrt{Pc_P}\,|\alpha_T(P, c_P) - \alpha| \Rightarrow \infty$
2. If $P/c_P \to m$ $(0 < m < \infty)$ and $d_T \neq 0$, then $\sqrt{Pc_P}\,(\alpha_T(P, c_P) - \alpha) \Rightarrow N(0, \sigma^2) + d_T m^{1/2}$.
3. If $P/c_P \to 0$, then $\sqrt{Pc_P}\,(\alpha_T(P, c_P) - \alpha) \Rightarrow N(0, \sigma^2)$

Theorem 4

Let $\{X_i(s), s \geq 0\}$ be iid samples of an irreducible, finite state space CTMC satisfying Equations 2.2 and 2.3. Assume $\kappa_P(c_P)/c_P \to 0$. As $P \to \infty$, if either

1. $P/c_P \to \infty$ and $\kappa_P(c_P)/\ln(P) \to \infty$, or
2. $P/c_P \to m$ $(0 < m < \infty)$ and $\kappa_P(c_P) \to \infty$, or
3. $P/c_P \to 0$

then

$$\sqrt{Pc_P}\,(\alpha_R(P, c_P) - \alpha) \Rightarrow N(0, \sigma^2) . \tag{2.15}$$

The $\ln(P)$ term in part (1) of Theorem 4 arises because finite state space CTMCs converge exponentially fast to their steady-state distribution. As indicated earlier, the bias expansions and Theorems 1 - 4 are valid under more general conditions. Basically, one needs a functional version of the central limit theorem in Equation 2.3, uniform integrability of second moments in this joint central limit theorem, exponential convergence to steady-state, and some sort of regularity conditions on $C_i(t)$ and $\{X_i(s), s \geq 0\}$. In Glynn and Heidelberger (1989b), it was assumed that

$C_i(t) = \int_0^t \chi_i(s)ds$ where $\chi_i(s)$ is bounded and that $\{X_i(s), s \geq 0\}$ is a bounded regenerative process. The regenerative assumption is not really as restrictive as it might seem since the estimation procedures do not make use of the regenerative structure. It is mainly used as a proof device, and in addition, many stochastic processes possess a (hidden) regenerative structure (see, e.g., Glynn (1989)). We further believe the result to be true for more general cumulative processes $\{C_i(t), t \geq 0\}$ where, e.g., $C_i(t)$ is discontinuous.

3. Queueing Models Used for Experimentation

In this section, we describe four queueing models that we used for determining, experimentally, the behavior of $\alpha_T(P, c)$ and $\alpha_R(P, c)$. These represent simplified versions of models (with analytically tractable solutions) that often arise in simulations of computer or communications systems. We ran experiments on the waiting time process in an M/M/1 queue and on three CTMCs: the queue length processes in an M/M/1 queue with feedback, a open Jackson network and a closed product form network (see, e.g., Kleinrock (1975)).

For the M/M/1 waiting time simulations, we let ϕ be the arrival rate, μ be the service rate, and $\rho = \phi/\mu$ be the traffic intensity. Let W_n be the waiting time of the n-th customer. For $\rho < 1$, $W_n \Rightarrow W$. The performance measure of interest is $\alpha = \mathrm{E}[W] = \rho/[\mu(1-\rho)]$.

For the M/M/1 queue with feedback, we let ϕ denote the arrival rate, μ the service rate and p the feedback probability. The expected number of visits a customer makes to the queue is $1/(1-p)$ and the traffic intensity is $\rho = \phi/[\mu(1-p)]$. We let $Q(s)$ denote the queue length at (simulated) time s, including the customer in service. Then $Q(s) \Rightarrow Q$ as $s \to \infty$ provided $\rho < 1.0$. The output performance measure of interest is the steady-state mean queue length, $\alpha = \mathrm{E}[Q] = \rho/(1-\rho)$. We set $\phi = 1$, $\mu = 20$, and $p = 0.9$, so that $\rho = 0.50$ and $\alpha = 1.0$. We ran experiments with two sets of initial conditions: $Q(0) = 0$ and $Q(0) = 5$.

A diagram of the open Jackson network is shown in Figure 1. This network is sometimes called an open central server model (see Buzen (1973)) with server 0 representing a CPU (central processing unit), and servers 1 to 4 representing I/O devices. There is a single type of job. Jobs arrive to the network (at the CPU) according to a Poisson process with rate ϕ. All servers operate using the FCFS service discipline and the service times of jobs at server i are iid exponentially distributed rvs with mean s_i. When a job leaves the CPU, it goes to I/O device i with probability p_i $(1 \leq i \leq 4)$, and when a job leaves an I/O device, it goes to the CPU with probability p_0 and exits the system with probability $(1 - p_0)$. Let $Q_i(s)$ denote the queue length at server i at time s (including the customer in service) and let ρ_i denote the traffic intensity at server i. Provided $\rho_i < 1$, then $(Q_0(s), \ldots, Q_4(s)) \Rightarrow (Q_0, \ldots, Q_4)$ as $s \to \infty$. Under the above assumptions, the steady-state distribution of $(Q_0, \ldots, Q_4)$ has a product form, and in particular $\mathrm{E}[Q_i] = \rho_i/(1-\rho_i)$. The

output performance measure of interest is $\alpha = E[Q_0]$. We set $\phi = 1.0$, $p_0 = 0.75$, $p_i = 0.25$ for $i \geq 1$, $s_0 = 0.1875$, and $s_i = 0.50$ for $i \geq 1$. With these parameters, $\rho_0 = 0.75$, $\rho_i = 0.50$ for $i \geq 1$, $\alpha = E[Q_0] = 3.00$, and $E[Q_i] = 1.0$ for $i \geq 1$. This model was simulated with initial conditions $Q_i(0) = 0$ for all i.

The closed, product form queueing network model is shown in Figure 2. This model is sometimes called a closed central server model. Again server 0 represents a CPU and servers 1 to 4 represent I/O devices. There are a fixed number of jobs N circulating in the network. As in the open model, the service discipline is FCFS at all servers and we assume iid exponentially distributed service times with mean s_i at server i. When a job leaves the CPU, it goes to I/O device i with probability p_i and when a job leaves an I/O device it goes back to the CPU. Let $Q_i(s)$ denote the queue length at server i at time s (including the customer in service) and let ρ_i denote the steady-state utilization of server i. Then $(Q_0(s), \ldots, Q_4(s)) \Rightarrow (Q_0, \ldots, Q_4)$ as $s \to \infty$. The performance measure of interest is $\alpha = E[Q_0]$. We set $N = 10$, $p_i = 0.3$ for $1 \leq i \leq 3$, $p_4 = 0.1$, $s_0 = 1.0$, $s_i = 2.0$ for $1 \leq i \leq 3$, and $s_4 = 11.0$. With these parameters, $\rho_0 = 0.82$, $\rho_i = 0.49$ for $1 \leq i \leq 3$, $\rho_4 = 0.90$, $\alpha = E[Q_0] = 3.06$, $E[Q_i] = 0.92$ for $1 \leq i \leq 3$, and $Q_4 = 4.17$. This model was simulated with initial conditions $Q_0(0) = 6$ and $Q_i(0) = 1$ for $i \geq 1$.

Because we did not have convenient access to a very highly parallel machine, all experiments were run on a single processor. The effect of running parallel replications on multiple processors with a computer time stopping constraint was simulated as follows. For the queue length processes, we assumed that each event (external arrival or service completion) took one unit of computer time to process. Thus $C_i(t)$ is the number of events completed at simulated time t (on replication i) and $T_i(c)$ is the amount of simulated time generated after processing c events.

For the M/M/1 waiting time simulations, we let $C_i(t)$ be the arrival time of customer number t and $T_i(c)$ be the number of customers that arrive in the interval $(0,c)$. In this example, the integrals in Equations 2.3 and 2.4 get replaced by sums. Using these definitions for $C_i(t)$ and $T_i(c)$ allows the ratio bias term d_T to be calculated analytically, as follows. Let $\overline{W}_n = \sum_{k=0}^{n-1} W_k/n$ be the average waiting time of the first n customers and let $\overline{A}_n = \sum_{k=1}^{n} A_k/n$ be the average interarrival time of the first n customers. Note that $\overline{A}_n = C_i(n)/n$. Since $T_i(n)/n \to \phi$ (the arrival rate), in this example we have $\phi = \lambda$. We also have

$$\sqrt{n}\,(\,\overline{W}_n - \alpha\,,\ \overline{A}_n - \lambda^{-1}\,) \Rightarrow \mathbf{N(0,B)} \tag{3.1}$$

for some covariance matrix $\mathbf{B}$. By Theorem 5 of Glynn and Heidelberger (1989b), $\mathbf{B}$ and $\mathbf{A}$ (the covariance matrix in Equation 2.3) are related by $A_{11} = B_{11}/\lambda$, $A_{12} = -\lambda B_{12}$, and $A_{22} = \lambda^3 B_{22}$. Thus by Equation 14, $d_T = B_{12}\,(= -A_{12}/\lambda)$. All the terms of $\mathbf{B}$ can be explicitly calculated (we

set $\mu = 1$): $B_{11} = \rho[2 + 5\rho - 4\rho^2 + \rho^3]/(1-\rho)^4$ (see, e.g., Blomqvist (1967) or, more recently, Whitt (1989)), $B_{22} = \mathrm{Var}[A_k] = 1/\lambda^2$, and, by using regenerative process theory (see, e.g., Crane and Iglehart (1975))

$$B_{12} = \frac{\mathrm{Cov}[\,(Y_i - \alpha N_i)\,,(t_i - \lambda^{-1}N_i)\,]}{\mathrm{E}[N_i]} \tag{3.2}$$

where Y_i is the sum of the waiting times in the i-th regenerative cycle, t_i is the sum of the interarrival times in the i-th cycle and N_i is the number of arrivals in the i-th cycle. Now $\mathrm{E}[N_i] = 1/(1-\rho)$, while the covariance term in Equation 3.2 can be calculated from results contained in Lavenberg, Moeller and Sauer (1977) (the research report version of Lavenberg, Moeller and Sauer (1979)). (This covariance term arises in the context of control variables for simulation variance reduction.) Specifically, for M/G/1 queues $\mathrm{Cov}[(Y_i - \alpha N_i)\,,(t_i - N_i/\lambda)] = -\,b_2/[2(1-\rho)^3]$ where b_2 is the second moment of the service times. Since, for M/M/1 (with $\mu = 1$) $b_2 = 2$, Equation 3.2 reduces to $d_T = B_{12} = -\,1/(1-\rho)^2$.

The effect of ratio bias on confidence interval coverage can now be calculated analytically. Let $\Phi(x) \equiv \mathrm{P}\{N(0,1) \le x\}$ and define $z_{\delta/2}$ by $\Phi(z_{\delta/2}) = 1 - \delta/2$. From part 2 of Theorem 3, if $P/c_P = m$, and $\beta = d_T\sqrt{m\,/\,A_{11}}$, then

$$\begin{aligned} \mathrm{P}\{\sqrt{Pc_P}\,|\alpha_T(P,c_P) - \alpha|/\sqrt{A_{11}} \le z_{\delta/2}\} &\approx \mathrm{P}\{|N(0,1) + \beta| \le z_{\delta/2}\} \\ &= \Phi(z_{\delta/2} - \beta) - \Phi(-z_{\delta/2} - \beta)\,. \end{aligned} \tag{3.3}$$

Thus for any given ρ, P, and c_P, the actual coverage of presumed $100 \times (1-\delta)\%$ confidence intervals can be predicted. In Section 5, we will compare the predicted coverage with the actual coverage observed in simulation experiments. Note that by using the heavy traffic approximation $B_{11} \approx 4\rho/(1-\rho)^4$ (see Whitt (1989)) we obtain $\beta \approx -0.5\sqrt{m}$. This approximation also works well for moderate values of ρ. Thus for given P and c_P, we expect the loss in coverage due to ratio bias to be approximately independent of the traffic intensity (provided ρ is not too small and P and c_P are large enough that the central limit theorem is valid). This behavior will be observed in Section 5.

4. Design of the Simulation Experiments

In this section we describe how the simulation experiments were performed. As mentioned earlier, the effect of running parallel replications on multiple processors with a computer time stopping constraint was simulated on a single processor. For the various models, and different values of P, c and $\kappa(c)$, we were interested in estimating the mean, variance and confidence

interval coverage of $\alpha_T(P, c)$ and $\alpha_R(P, c)$. We built a simple queueing network simulator suitable for these purposes. (We used the combined generator described in L'Ecuyer (1988) as a source of random numbers.) To estimate these quantities for given values of P, c, and $\kappa(c)$, M "super replications" were performed where each super replication consisted of P replications, each of length c and having truncation interval $\kappa(c)$. Thus for super replication j $(1 \le j \le M)$ samples $\alpha_T(P,c,j)$ and $\alpha_R(P,c,j)$ of $\alpha_T(P,c)$ and $\alpha_R(P,c)$, respectively, were obtained (according to Equations 2.5 and 2.6). $E[\alpha_T(P,c)]$ and $E[\alpha_R(P,c)]$ were estimated by $\bar{\alpha}_T(P,c) \equiv \sum_{i=1}^{M} \alpha_T(P,c,j)/M$ and $\bar{\alpha}_R(P,c) \equiv \sum_{i=1}^{M} \alpha_R(P,c,j)/M$, respectively. The sample standard deviations, $S_T(P,c)$ and $S_R(P,c)$ of $\bar{\alpha}_T(P,c)$ and $\bar{\alpha}_R(P,c)$, respectively, were computed in the usual way, e.g. $S_T^2(P,c) = \sum_{i=1}^{M} (\alpha_T(P,c,j) - \bar{\alpha}_T(P,c))^2/[M(M-1)]$.

On each super replication we also obtained asymptotic standard deviation estimates $\hat{\sigma}_T(P,c,j)$ and $\hat{\sigma}_R(P,c,j)$ for $\alpha_T(P, c)$ and $\alpha_R(P, c)$, respectively. These were estimated as follows. Let $Y_i(c,j)$ and $\tau_i(c,j)$ be the samples of $Y_i(c)$ and $\tau_i(c)$ obtained on the j-th super replication. Then

$$\hat{\sigma}_T^2(P,c,j) \equiv \frac{\sum_{i=1}^{P} \left(\frac{Y_i(c,j)}{\tau_i(c,j)} - \alpha_T(P,c,j) \right)^2}{P-1} . \tag{4.1}$$

Computation of $\hat{\sigma}_R^2(P,c,j)$ is analogous to variance estimation in regenerative simulation:

$$\hat{\sigma}_R^2(P,c,j) \equiv \frac{\frac{1}{P} \cdot \sum_{i=1}^{P} \left(Y_i(c,j)^2 - 2\,\alpha_R(P,c,j)\, Y_i(c,j)\, \tau_i(c,j) + \alpha_R(P,c,j)^2\, \tau_i(c,j)^2 \right)}{\left(\frac{1}{P} \sum_{i=1}^{P} \tau_i(c,j) \right)^2} . \tag{4.2}$$

From these point and variance estimates, presumed $100 \times (1-\delta)\%$ confidence intervals for α can be formed as follows. Using the traditional estimator the confidence interval is $\alpha_T(P,c,j) \pm t_{\delta/2}(P-1)\hat{\sigma}_T(P,c,j)/\sqrt{P}$ where $t_{\delta/2}(P-1)$ is defined by $1 - \delta/2 = P\{T_{P-1} \le t_{\delta/2}(P-1)\}$ and T_{P-1} has a Student's t distribution with $(P-1)$ degrees of freedom. Using the classical ratio estimator the confidence interval is $\alpha_R(P,c,j) \pm z_{\delta/2}\hat{\sigma}_R(P,c,j)/\sqrt{P}$. For a given estimator, we define its coverage to be the fraction of these confidence intervals that actually contain α. If valid confidence intervals are being formed for α, then, by definition the coverage should converge to $(1\text{-}\delta)$ as $M \to \infty$. In all cases we set $\delta = 0.1$ corresponding to 90% confidence intervals.

The simulator was organized in such a way that statistics could be collected for multiple values of P, c and $\kappa(c)$ from the same set of runs. Thus the data generated for a particular model are correlated. We took values of P to be powers of two, ranging from $P = 8$ to $P = 512$ for the CTMCs and $P = 128$ to $P = 1024$ for the M/M/1 waiting time simulations. Each super replication for P processors also comprised 2 super replications for $P/2$ processors, 4 super replications for $P/4$ processors, etc. We used 200 super replications for the largest values of P in each case. Thus, e.g., 12,800 super replications of the CTMCs were obtained for $P = 8$. These sample sizes were generally large enough so that very accurate point estimates were obtained.

5. Experimental Results

The first set of experiments are for the M/M/1 waiting times. The purpose of these experiments is to compare the analytic results of Section 3 to actual simulation results. To isolate just the effect of the ratio bias, these simulations were started in the steady-state distribution. We simulated until (simulated) time $c = 1{,}000$. By deleting customers arriving before times $\kappa(c) = 100$, 250 and 500, we obtained runs of effective lengths $c - \kappa(c) = 900$, 750 and 500, respectively. We simulated at $\rho = 0.50$ and $\rho = 0.75$.

The results of these experiments are listed in Table 1. Table 1 lists the predicted coverages for $\alpha_T(P, c)$ as calculated by Equation 3.3 (using the effective run length for c in that equation), as well as the actual coverages for $\alpha_T(P, c)$ and $\alpha_R(P, c)$ observed in the simulations. Table 1 indicates generally excellent agreement between the predictions and the experiments. Notice that, for given P and $\kappa(c)$, the predicted and actual coverage for $\alpha_T(P, c)$ is quite insensitive to the value of ρ, as explained in Section 3. In addition, for fixed $\kappa(c)$, as P increases the coverage for $\alpha_T(P, c)$ decreases. This is in agreement with part 2 of Theorem 3 and is explained by the fact that as P increases, increasingly accurate estimates of (the biased) $E[\alpha_T(P, c)]$ are obtained. This loss in coverage is greatest for the largest value of $\kappa(c)$ since that corresponds to the smallest effective run length. On the other hand, the coverage for $\alpha_R(P, c)$ stays close to its nominal value of 0.90.

Figures 3 to 5 plot results from simulations of the M/M/1 queue with feedback. Figure 3 plots $\bar{\alpha}_T(P,c)$ and $\bar{\alpha}_R(P,c)$ as a function of $\kappa(c)$ for $c = 1000$ events, $P = 512$, and two different initial queue lengths. Actually, when $Q(0) = 0$, $\alpha_T(P, c)$ appears almost unbiased without truncation ($\kappa(c) = 0$), but $\bar{\alpha}_T(P,c)$ increases above the steady state value of $\alpha = 1.0$ as $\kappa(c)$ increases. In this case, the initialization bias and ratio bias are of opposite signs and, in effect, approximately cancel each other out when $\kappa(c) = 0$. When $Q(0) = 5$, $\bar{\alpha}_T(P,c)$ decreases as $\kappa(c)$ increases, but, again, does not come close to α. For $\kappa(c) = 500$ the values of $\bar{\alpha}_T(P,c)$ are very nearly the same for both $Q(0) = 0$ and $Q(0) = 5$, but are about 8% above the steady-state value. On the other hand, $\bar{\alpha}_R(P,c)$ approaches α as $\kappa(c)$ increases for both $Q(0) = 0$ and $Q(0) = 5$. These point estimates are

very accurate. For example, when $Q(0) = 0$ and $\kappa(c) = 500$, $\bar{\alpha}_T(512,c) = 1.084$, $S_T(512,c) = 0.002$, $\bar{\alpha}_R(512,c) = 1.002$ and $S_R(512,c) = 0.002$.

Figure 4 plots the coverages for these estimators (without deletion) as a function of P. Because, by coincidence, $E[\alpha_T(P, c)] \approx \alpha$ when $Q(0) = 0$ and $\kappa(c) = 0$, the coverage for $\alpha_T(P, c)$ remains at or near the nominal value of 0.90. However, the coverage for $\alpha_T(P, c)$ decreases (to zero) as P increases when $Q(0) = 5$ because of the stronger initialization bias. Similarly, because of initialization bias, the coverage for $\alpha_R(P, c)$ is seriously degraded for both $Q(0) = 0$ and $Q(0) = 5$.

Figure 5 shows the coverages when $\kappa(c) = 250$. With this value of $\kappa(c)$, the initialization bias is essentially eliminated, although ratio bias is still present: for example, when $Q(0) = 5$, $\bar{\alpha}_R(512,c) = 1.004$ compared to $\alpha = 1.0$ while $\bar{\alpha}_T(512,c) = 1.060$. Because of the ratio bias, the coverage for $\alpha_T(P, c)$ decreases from around 0.90 to less than 0.20 as P increases from 8 to 512 for both initial conditions. Significant coverage loss begins to be observed in the range from $P = 32$ to $P = 64$. On the other hand, the coverage for $\alpha_R(P, c)$ starts out slightly below 0.90 for $P = 8$ and then rapidly approaches 0.90 as P increases. The low coverage when $P = 8$ is due both to a less robust variance estimate as well as to the use of a normal multiplier, rather than a t-multiplier, in the confidence interval. For example, when $Q(0) = 0$ and a t-multiplier with 7 degrees of freedom is used instead of the normal multiplier, the coverage for $\alpha_R(8,P)$ increases from 0.820 to 0.864.

Figures 6 and 7 display results of simulating the open central server model. This network was simulated for $c = 2500$ events. Figure 6 plots $\bar{\alpha}_T(P,c)$ and $\bar{\alpha}_R(P,c)$ as a function of $\kappa(c)$ for $P = 8$, 64 and 512. (Because of the organization of the simulator's data collection facilities, $\bar{\alpha}_T(P,c)$ is independent of P.) Initialization bias is essentially eliminated by $\kappa(c) = 1000$, but significant ratio bias is still evident in $\bar{\alpha}_T(P,c)$. Note also that there are only slight differences between $\bar{\alpha}_R(8,c)$, $\bar{\alpha}_R(64,c)$ and $\bar{\alpha}_R(512,c)$. Because of the initialization bias, without deletion, the coverage for both $\alpha_T(P, c)$ and $\alpha_R(P, c)$ are well below the 0.90 level: the coverage for $\alpha_T(64,c)$ is 0.627 while the coverage for $\alpha_R(64,c)$ is 0.457. Note that when $\kappa(c) = 250$, $\alpha_T(P, c)$ is, by chance, almost unbiased. Thus, with this amount of deletion, the coverage for $\alpha_T(P, c)$ will be (approximately) correct, but for the wrong reason. For example, $\alpha_T(512,c)$ has coverage 0.91 while $\alpha_R(512,c)$ has coverage of only 0.60.

Figure 7 plots the coverages for $\alpha_T(P, c)$ and $\alpha_R(P, c)$ as a function of P when initialization bias is essentially removed ($\kappa(c) = 1000$). Again, the coverage for $\alpha_T(P, c)$ decreases as P increases, while the coverage for $\alpha_R(P, c)$ increases to and then remains at or near the nominal 0.90 level.

Figure 8 displays a similar pattern for simulations of the closed central server model. This figure plots coverage results when $c = 2000$ and $\kappa(c) = 500$. With these parameters, initialization

bias is removed but ratio bias is still present. The steady-state value being estimated is $\alpha = 3.057$, and $\bar{\alpha}_R(512,c) = 3.056$ ($S_R(512,c) = 0.002$) while $\bar{\alpha}_T(512,c) = 3.111$ ($S_T(512,c) = 0.002$).

As has been indicated several times above, for given values of P, c and $\kappa(c)$, the values of $S_T(P,c)$ and $S_R(P,c)$ have been very nearly the same. This has been observed throughout our experiments. This is explained by the fact that, even with ratio bias still present, $\alpha_T(P, c)$ and $\alpha_R(P, c)$ both obey central limit theorems with the same asymptotic standard deviation (see Theorems 3 and 4).

6. Summary and Conclusions

This paper has considered the problem of estimating steady-state parameters on multiple processors via the method of parallel replications. While the method is conceptually straightforward to apply, statistical considerations point to the need for using an alternative steady-state estimation procedure. This need arises because the traditional estimator, $\alpha_T(P, c)$, contains two sources of bias having the same order of magnitude: initialization bias and ratio bias. While appropriate deletion of an initial portion of each simulation effectively removes initialization bias, it does not affect the ratio bias. When using a large number processors, this residual ratio bias results in a biased estimate and corresponding loss in confidence interval coverage.

The alternative estimator, $\alpha_R(P, c)$, corresponds to the classical ratio estimator that is commonly used in regenerative simulation. Its ratio bias is order P times smaller than its initialization bias. Thus appropriate deletion is effective in removing the major source of bias. The net effect is that by using $\alpha_R(P, c)$ rather than $\alpha_T(P, c)$ allows one to either:

1. use many more processors for a given amount of computing time per processor, or
2. make shorter runs for a given number of processors.

This paper examined these issues empirically via simulations of a variety of queueing systems. Our experiments confirm the theoretical results, and indicate that the ratio bias can become a problem even on moderately sized parallel processors with, say, 32 to 64 processors.

The results of this paper also have some applicability to the standard single processor method of independent replications. In this method, the replication length is often determined by either the total number of events, a simulated time limit, a computer time limit, or the number of events of a particular type such as the number of departures from a queue. (Sometimes a combination of these limits is used.) When estimating many parameters in a queueing network, there will always be some parameters that are estimated on a different time scale than that used to determine the replication length. Thus the denominator of some parameter estimates will be random, resulting in ratio bias. For example, if simulated time is used to control the replication

length, then response time estimates will have a random denominator (the number of customers departing from the queue). On the other hand, if an event count is used to control the replication length, then queue length estimates will have a random denominator (the simulated time). Thus ratio bias could be a concern, even on a single processor. However, there is usually little motivation to run a very large number of short replications on a single processor, since either batch means or a running few long replications will be less sensitive to initialization bias. Never-the-less, the issue of ratio bias should be kept in mind. In fact, for a small number of replications, the ratio form of $\alpha_R(P, c)$ suggests the use of jackknifing (see Miller (1974)) for both (ratio) bias reduction and for robust variance estimation. However, the properties and validity of jackknifing in this situation have not yet been established, and remain as open problems for research.

In addition, if the replication length is determined within a sequential procedure (see, e.g., Law and Kelton (1982)), then the denominator of the estimates will typically be random resulting in possible ratio bias. This will also be true if the length of the truncation interval is determined by statistical tests of the simulation output (see, e.g., Schruben (1982)). The effect of ratio bias in these situations also has yet to be analyzed.

Acknowledgement

The work of Peter Glynn was supported by the IBM Corporation under SUR-SST Contract 12480042 and by the U.S. Army under contract number DAAL-03-88-K-0063.

References

1. Bhavsar, B.C. and Isaac, J.R. (1987). Design and Analysis of Parallel Monte Carlo Algorithms. *SIAM Journal on Scientific and Statistical Computing* **8**, s73-s95.
2. Billingsley, P. (1968). *Convergence of Probability Measures*. Wiley, New York.
3. Blomqvist, N. (1967) The Covariance Function of the M/G/1 Queueing System. *Skand. AktuarTidskr.* **8**, 157-174.
4. Bratley, P., Fox, B.L. and Schrage, L.E. (1987). *A Guide to Simulation*, Second Edition. Springer-Verlag, New York.
5. Buzen, J.P. (1973) Computational Algorithms for Closed Queueing Networks with Exponential Servers. *Commun. ACM* **16**, 527-531.
6. Cochran, W.G. (1963). *Sampling Techniques*, Second Edition. Wiley, New York.
7. Crane, M.A. and Iglehart, D.L. (1975). Simulating Stable Stochastic Systems, III: Regenerative Processes and Discrete Event Simulations. *Operations Research* **23**, 33-45.
8. Fox, B.L. and Glynn, P.W. (1987). Estimating Time Averages via Randomly Spaced Observations. *SIAM J. Appl. Math.* **47**, 186-200.
9. Fujimoto, R.M. (1989). Parallel Discrete Event Simulation. *1989 Winter Simulation Conference Proceedings*. E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.). IEEE Press, 19-28.
10. Glynn, P.W. (1987). Limit Theorems for the Method of Replication. *Stochastic Models* **4**, 344-350.
11. Glynn, P.W. (1989). A GSMP Formalism for Discrete Event Systems. *Proceedings of the IEEE* **77**, 14-23.
12. Glynn, P.W. (1990). A Low Bias Steady-State Estimator for Equilibrium Processes. Technical Report, Department of Industrial Engineering, University of Wisconsin, Madison, Wisconsin.
13. Glynn, P.W. and Heidelberger, P. (1989a). Analysis of Initial Transient Deletion for Replicated Steady-State Simulations. IBM Research Report RC 15259. Yorktown Heights, New York.
14. Glynn, P.W. and Heidelberger, P. (1989b). Analysis of Initial Transient Deletion for Parallel Steady-State Simulations. IBM Research Report RC 15260. Yorktown Heights, New York.
15. Glynn, P.W. and Heidelberger, P. (1990). Analysis of Parallel, Replicated Simulations Under a Completion Time Constraint. IBM Research Report RC 15466. Yorktown Heights, New York.
16. Heidelberger, P. (1986). Statistical Analysis of Parallel Simulations. *1986 Winter Simulation Conference Proceedings*. J. Wilson and J. Henriksen (eds.). IEEE Press, 290-295.
17. Heidelberger, P. (1988). Discrete Event Simulations and Parallel Processing: Statistical Properties. *SIAM Journal on Scientific and Statistical Computing* **9**, 1114-1132.
18. Iglehart, D.L. (1975). Simulating Stable Stochastic Systems, V: Comparison of Ratio Estimators. *Naval Res. Logist. Quart.* **22**, 553-565.
19. Kleinrock, L. (1975). *Queueing Systems, Volume I, Theory*. Wiley, New York.
20. Lavenberg, S.S., Moeller, T.L. and Sauer, C.H. (1977). Concomitant Control Variables Applied to the Regenerative Simulation of Queueing Systems. IBM Research Report RC 6413. Yorktown Heights, New York.
21. Lavenberg, S.S., Moeller, T.L. and Sauer, C.H. (1979). Concomitant Control Variables Applied to the Regenerative Simulation of Queuing Systems. *Operations Research* **27**, 134-160.
22. Law, A.M. (1977). Confidence Intervals in Discrete Event Simulation: A Comparison of Replication and Batch Means. *Naval Res. Logist. Quart.* **24**, 667-678.
23. Law, A.M. and Carson, J.S. (1979). A Sequential Procedure for Determining the Length of Steady-State Simulations. *Operations Research*. **27**, 1011-1025.
24. Law, A.M. and Kelton, W.D. (1982). Confidence Intervals for Steady-State Simulations II: A Survey of Sequential Procedures. *Management Science*. **28**, 550-562.

25. L'Ecuyer, P. (1988). Efficient and Portable Combined Random Number Generators. *Commun. ACM* **31**, 742-749 and 774.

26. Miller, R.G. (1974). The Jackknife - A Review. *Biometrika* **61**, 1-15.

27. Schruben, L.W. (1982). Detecting Initialization Bias in Simulation Output. *Operations Research* **30**, 569-590.

28. Whitt, W. (1989). Planning Queueing Simulations. *Management Science* **35**, 1341-1366.

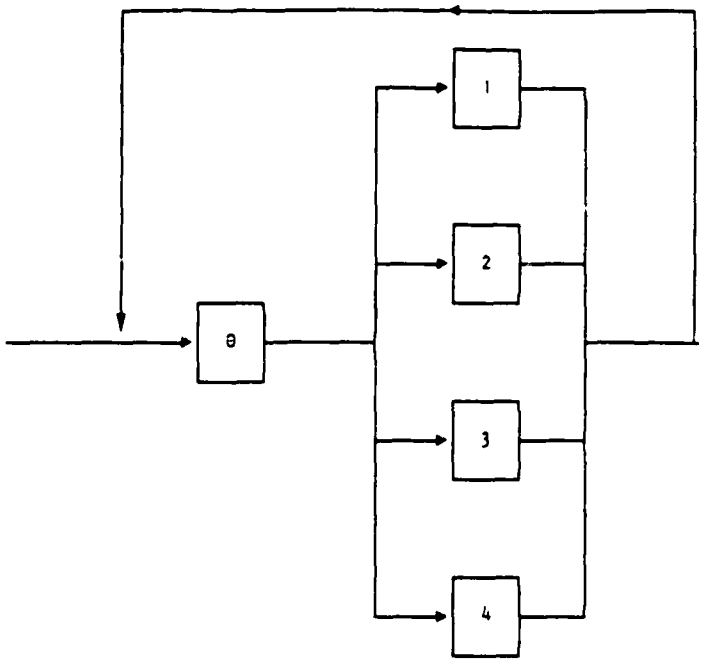

Figure 1
Open Central Server Model

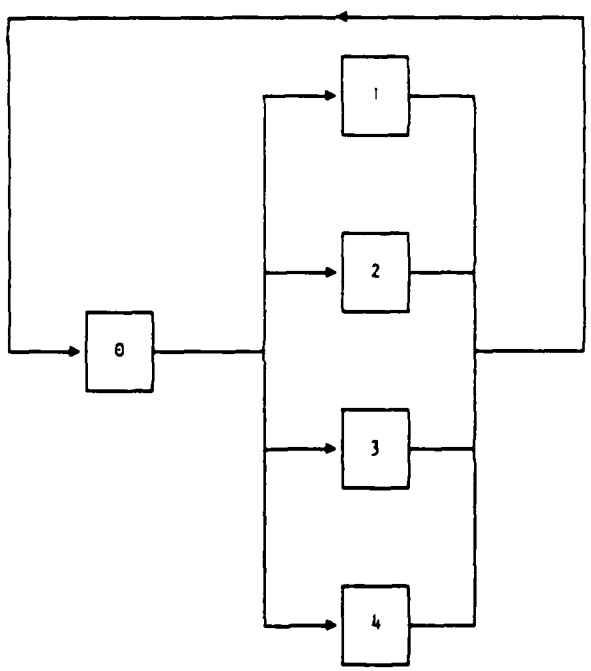

Figure 2
Closed Central Server Model

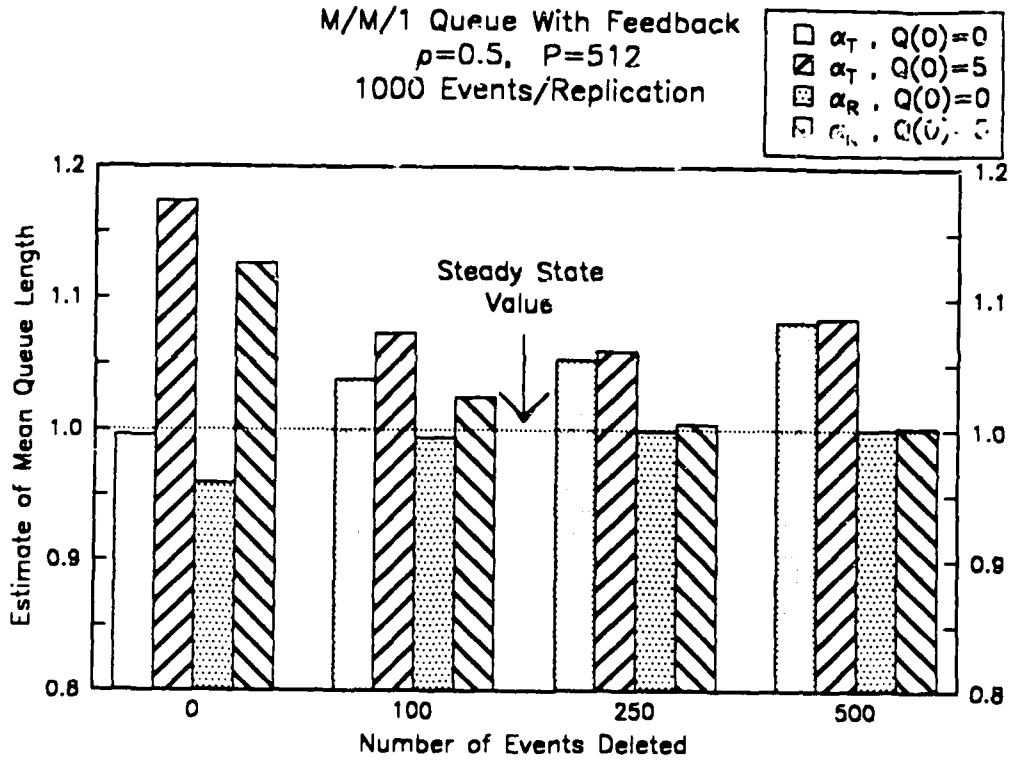

Figure 3: Estimated Mean Queue Length in M/M/1 Queue With Feedback

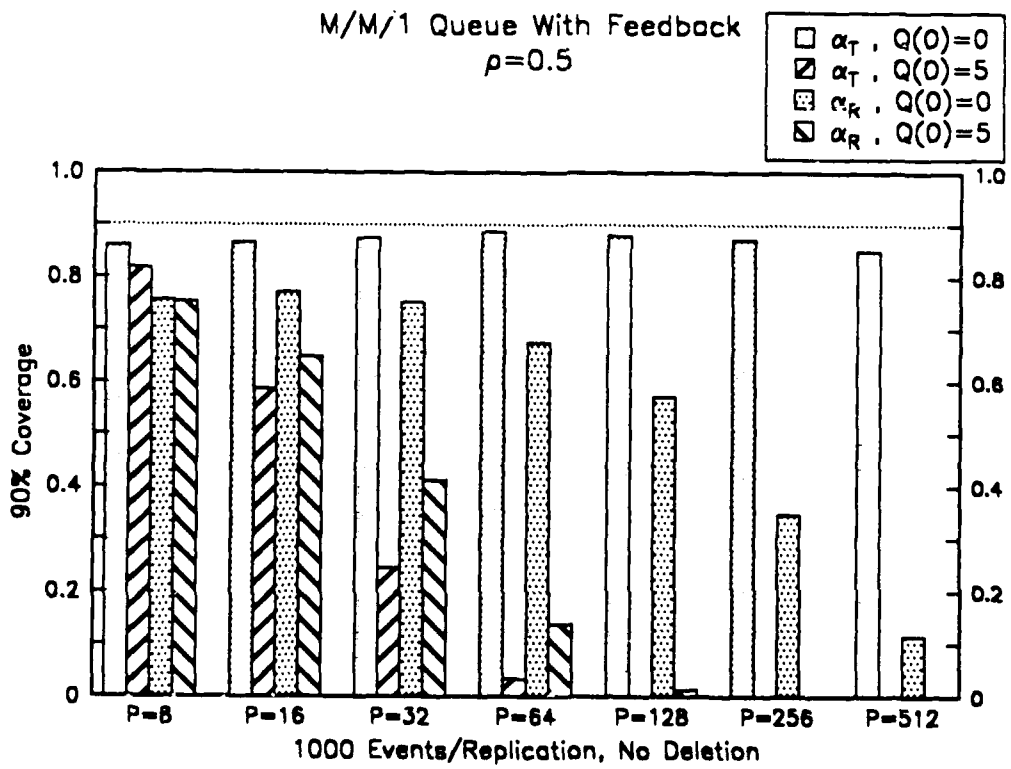

Figure 4: Estimated 90% Coverage in M/M/1 Queue With Feedback (Without Deletion)

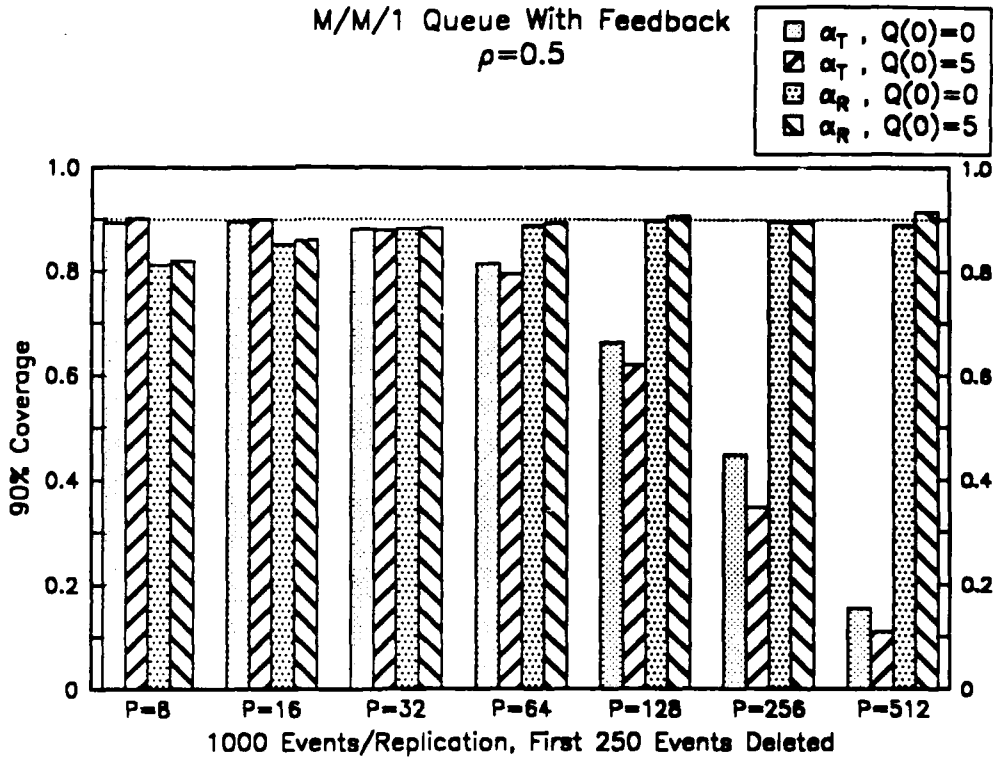

Figure 5: Estimated 90% Coverage in M/M/1 Queue With Feedback (With Deletion)

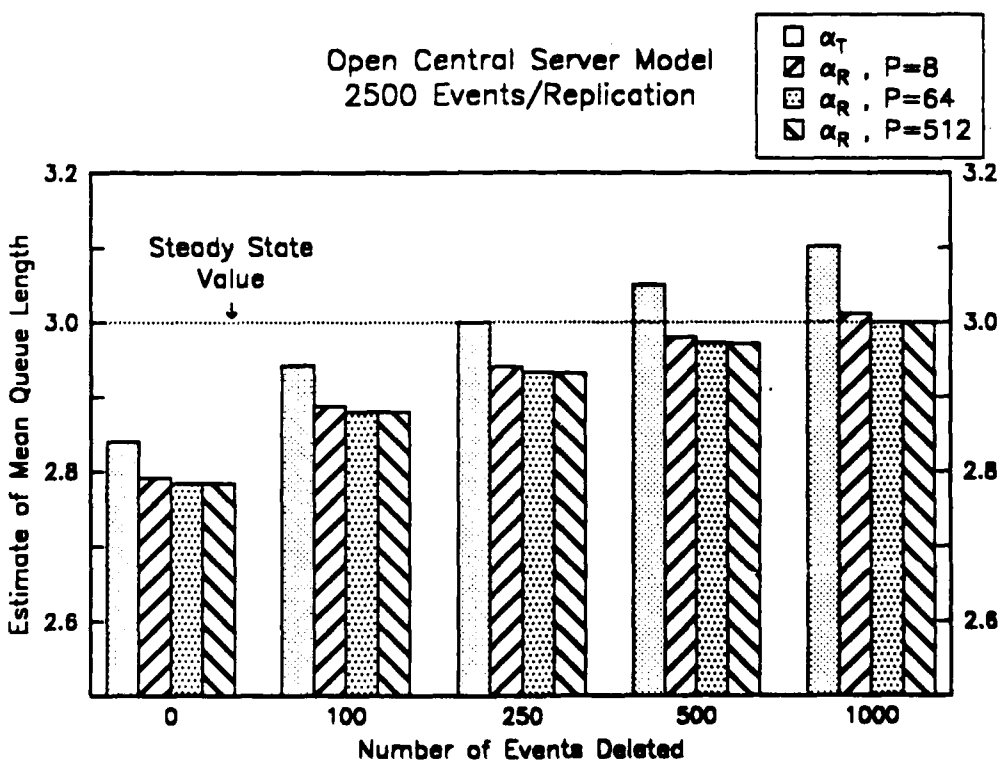

Figure 6: Estimated Mean Queue Length in Open Central Server Model

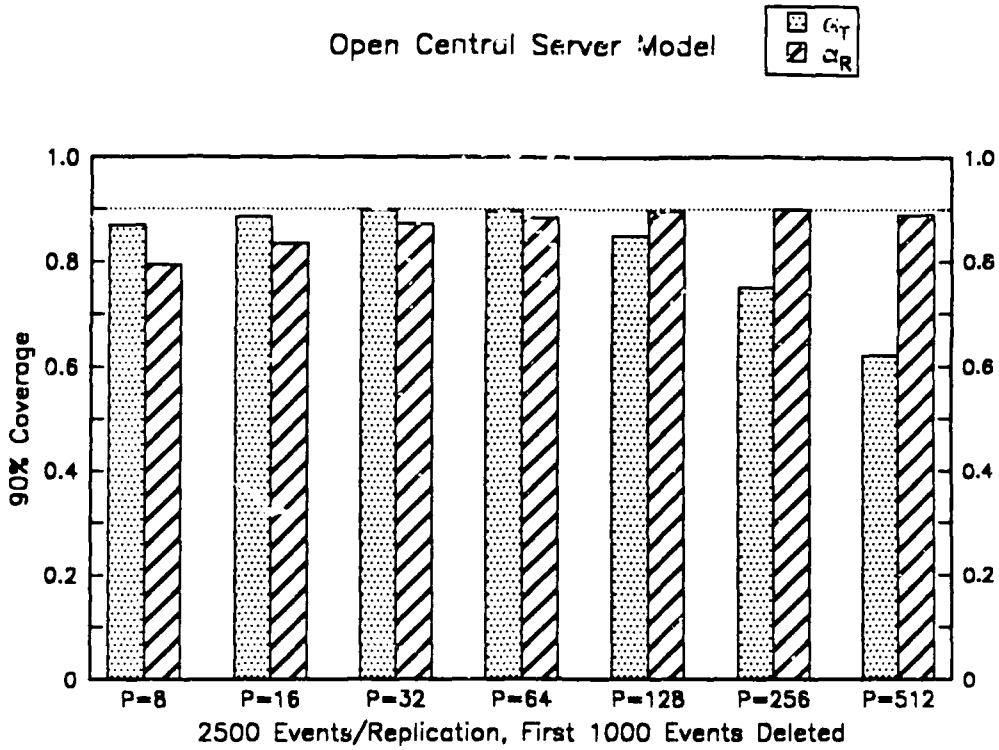

Figure 7: Estimated 90% Coverage in Open Central Server Model

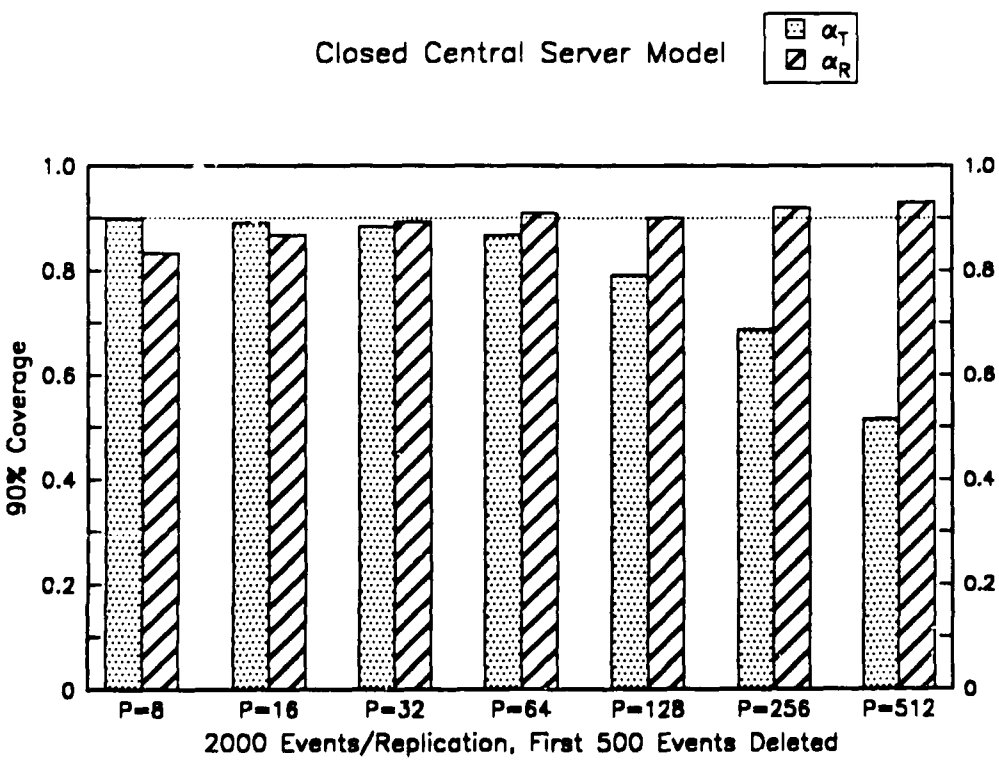

Figure 8: Estimated 90% Coverage in Closed Central Server Model

Table 1

Predicted and Actual 90% Confidence Interval Coverages
in M/M/1 Queue Waiting Time Simulations with $c = 1000$

$c - \kappa(c)$	P	$\rho = 0.5$ Predicted $\alpha_{\mathcal{T}}(P, c)$	$\rho = 0.5$ Actual $\alpha_{\mathcal{T}}(P, c)$	$\rho = 0.5$ Actual $\alpha_R(P, c)$	$\rho = 0.75$ Predicted $\alpha_{\mathcal{T}}(P, c)$	$\rho = 0.75$ Actual $\alpha_{\mathcal{T}}(P, c)$	$\rho = 0.75$ Actual $\alpha_R(P, c)$
500	128	0.888	0.874	0.894	0.889	0.871	0.898
	256	0.876	0.856	0.892	0.878	0.862	0.908
	512	0.852	0.848	0.909	0.856	0.875	0.912
	1024	0.806	0.795	0.930	0.813	0.850	0.920
750	128	0.892	0.888	0.903	0.893	0.885	0.898
	256	0.884	0.876	0.894	0.885	0.882	0.902
	512	0.868	0.870	0.900	0.871	0.870	0.912
	1024	0.837	0.845	0.915	0.842	0.865	0.950
900	128	0.893	0.886	0.900	0.894	0.882	0.898
	256	0.887	0.881	0.880	0.888	0.876	0.896
	512	0.874	0.880	0.900	0.876	0.865	0.912
	1024	0.847	0.855	0.930	0.851	0.870	0.920

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

MASTER COPY - FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Technical Report No. 59	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARO 25839.34-MA
6a. NAME OF PERFORMING ORGANIZATION Dept. of Operations Research 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) Stanford, CA 94305-4022	7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING / SPONSORING ORGANIZATION U. S. Army Research Office 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAL03-88-K-0063
8c. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
Experiments with initial transient deletion for parallel, replicated steady-state simulations

12. PERSONAL AUTHOR(S)
Peter W. Glynn and Philip Heidelberger

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM ____ TO ____	May 1990	24

16. SUPPLEMENTARY NOTATION
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
initial transient deletion; parallel simulation; steady-state; ratio estimation; discrete-event simulation

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Abstract: A simple and effective way to exploit parallel processors in discrete event simulations is to run multiple independent replications, in parallel, on multiple processors and to average the results at the end of the runs. We call this the method of parallel replications. This paper is concerned with using the method of parallel replications for estimating steady-state performance measures. We report on the results of queueing network simulation experiments that compare the statistical properties of several possible estimators that can be formed using this method. The theoretical asymptotic properties of these estimators were determined in Glynn and Heidelberger (1989a and 1989b). Both the theory and the experimental results reported here strongly indicate that a nonstandard (in the context of steady-state simulation), yet easy to apply, estimation procedure is required on highly parallel machines. This nonstandard estimator is a ratio estimator. The experiments also show that use of the ratio estimator is advantageous even on machines with only a moderate degree of parallelism.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED