

Instability and Performance Limits of Distributed Simulators of Feedforward Queueing Networks

RAJEEV SHOREY and ANURAG KUMAR

Indian Institute of Science

and

KIRAN M. REGE

Lucent Technologies: Bell Labs

In this article we study the performance of distributed simulation of open feedforward queueing networks, by analyzing queueing models of message flows in distributed discrete event simulators. We view each logical process in a distributed simulation as comprising a message sequencer with associated message queues, followed by an event processor. We introduce the idealized, but analytically useful, concept of *maximum lookahead*. We show that, with quite general stochastic assumptions for message arrival and time-stamp processes, the message queues are *unstable* for conservative sequencing, and for conservative sequencing with maximum lookahead and hence for optimistic resequencing, and for any resequencing algorithm that does not employ interprocessor *flow control*. Finally, we provide formulas for the throughput of distributed simulators of feedforward queueing networks.

Categories and Subject Descriptors: C.4 [Computer Systems Organization]: Performance of Systems; I.6.1 [Simulation and Modeling]: Simulation Theory; I.6.8 [Simulation and Modeling]: Types of Simulation—*discrete event*

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: Distributed discrete event simulation, event synchronization, stability of queueing systems

This work was done while R. Shorey was a Ph.D. candidate in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. An earlier version of this article appeared in the *ACM/IEEE/SCS 8th Workshop on Parallel and Distributed Simulation (PADS'94)* (July 6–8, 1994) 65–72. This article is an expanded version of the PADS'94 paper.

Authors' addresses: R. Shorey, Silicon Automation Systems Pvt. Ltd., 3008, 12th B Main, 8th Cross, HAL 2nd Stage, Indiranagar, Bangalore - 560 008, India; email: (shorey@sas.soft.net); A. Kumar, Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore - 560 012, India; email: (anurag@ece.iisc.ernet.in); K. M. Rege, Lucent Technologies: Bell Labs, Room 3M 308, Crawfords Corner Road, Holmdel, NJ 07733.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 1997 ACM 1049-3301/97/0400-0210 \$03.50

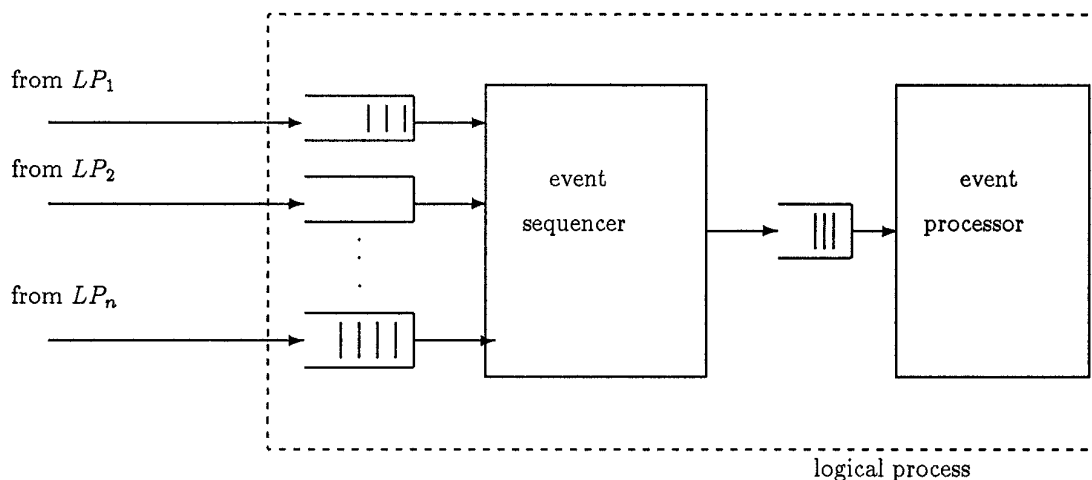


Fig. 1. Schematic view of a logical process.

1. INTRODUCTION

In a distributed discrete event simulation (DDES), the simulation model is partitioned into several logical processes (LPs) which are assigned to the various processing elements. The time evolution of the simulation at the various logical processes is synchronized by means of time-stamped messages that flow between the logical processes. We are concerned with the situation in which the messages are not just for synchronization, but also carry “work” which when done modifies the state of the receiving logical process. A typical example is the distributed simulation of a queueing network model, in which one or more queues is assigned to each logical process, and the messages indicate the motion of customers between the queues in the various logical processes. We assume that the simulation makes correct progress *if each logical process processes the incoming events, from all other logical processes, in time-stamp order*. It is possible for a conservatively synchronized parallel simulation to execute events out of time-stamp order, and still be correct (e.g., Gaujal et al. [1993]). Methods for doing so inevitably involve exploiting model-specific information that is not applicable in the general case. Likewise, it is possible for an optimistically synchronized parallel simulation that uses lazy cancellation [Reiher et al. 1990] or lazy re-evaluation to get the correct result without effectively committing events in time-stamp order. Its ability to do so is also problem dependent. The context in which our results apply is that of a *general purpose* conservatively synchronized parallel simulator where the only information available for sequencing is message time-stamps, or an optimistically synchronized parallel simulator that uses aggressive cancellation.

Each logical process may be viewed as comprising an input queue for each channel over which it can receive messages from another logical process (i.e., LP_1, LP_2, \dots, LP_n); (see Figure 1). Since the messages must

be processed in time-stamp order, the event processor must be preceded by an event sequencer. The messages must emerge from the sequencer in time-stamp order.

It is the event sequencer that is at the core of much of the research in distributed discrete event simulation. Event sequencing algorithms fall into one of two classes: *conservative* or *optimistic*. A conservative event sequencer allows a message to pass through only if it is sure that no event with a lower time-stamp can arrive in the (real-time) future [Chandy and Misra 1981; Misra 1986]. An optimistic event sequencer, on the other hand, occasionally lets messages pass through without being sure that no lower time-stamped event can arrive in the future. If a lower time-stamped event does arrive, corrective action is taken resulting in a *roll-back* of the simulation [Jefferson 1985; Fujimoto 1990].

Considerable work has been done on performance models of distributed simulation with the objective of obtaining estimates or bounds on simulation speedup with respect to centralized simulation.¹ In these models, speedup is defined as the ratio of the real-time rate of advancement of correctly simulated virtual time in a distributed simulation and a centralized simulation. These analyses are usually made under considerably simplifying assumptions, and generally yield bounds on the expected speedup. In particular, little attention seems to have been paid to formally studying the behavior of the interprocessor message queues in distributed discrete event simulators. The simulation progresses by processing these messages. Following Wagner and Lazowska [1989], we view a simulator as a queueing network in which the customers are these interprocessor messages; the throughput of this network would correspond to the progress rate of the simulation. Furthermore, viewing the problem in this way may lead to useful insights into issues such as the allocation of logical processes to processors.

In this article we study stochastic models for distributed simulators of feedforward stochastic queueing networks. We first study a particular class of stochastic models for message and time-stamp arrivals at an event sequencer in a logical process. We consider event sequencers that rely only on time-stamp information in event messages. We show that for this class of models, and for conservative sequencing, the message queues that precede the sequencer are essentially unstable. Next, we show that even with maximum lookahead (i.e., prescient knowledge of the time-stamp of the next message yet to arrive on the channel with the empty message queue) these queues are still unstable. It follows that the resequencing problem is fundamentally unstable (even for optimistic sequencing), and some form of interprocessor “flow control” is necessary in order to make the message queues stable (without message loss). Using mainly simulation results and heuristic reasoning Shanker and Patuwo [1993] have antici-

¹See Mitra and Mitrani [1984], Nicol [1988], Wagner and Lazowska [1989], Kleinrock [1990], Felderman and Kleinrock [1991], and Nicol [1993].

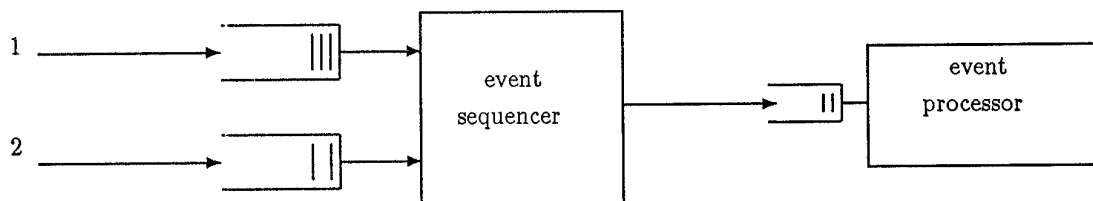


Fig. 2. A logical process with two input message streams.

pated some of the results we report here. Our results are based on a complete analysis of a formal stochastic model.

Our notion of maximum lookahead is based on the notion of correctness of the simulation previously set down; that is, the simulation is correct by ensuring that every logical process processes the events in time-stamp order. This notion of correctness reproduces sample paths of a queueing network simulation, no matter which sequencing algorithm is used. Note that simulating two customers out of order at a queue will at least give them different samples of service time (with probability one), although it may not alter the average delay estimate. In particular, this implies that our analysis does not permit lazy cancellation [Gafni 1988] in optimistic simulation.

We obtain some generalizations of the instability results to time-stamped message arrival processes with certain ergodicity properties. We characterize the departure process from the event sequencer and obtain a stability condition for the event processor (see Figure 1). Finally, we provide formulas for the throughput of distributed simulators of feedforward queueing networks. These formulas involve parameters of the queueing model, and the service rates of the processors in the distributed simulator, and hence demonstrate, for example, the performance of a particular mapping of a problem onto a simulator.

The article is organized as follows. In Section 2, we study the instability of queues associated with the event sequencer. In Section 3 we present some complements and extensions of the results of Section 2. Section 4 contains the conclusions and directions for future work.

2. INSTABILITY OF SEQUENCING

Two time-stamped message streams arrive at a logical process (see Figure 2). Within each stream the messages are in time-stamp order. The messages must be processed in overall time-stamp order by the event processor.

We first assume that the two message arrival streams form independent Poisson processes, and that the successive time-stamps in each stream are independent sequences of Poisson epochs independent of the message arrival process. With these stochastic assumptions, we prove that the message queues are unstable for both conservative sequencing and sequencing with maximum lookahead.

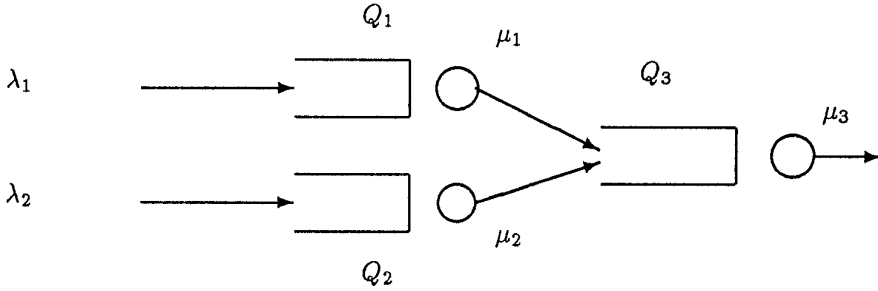


Fig. 3. A queueing network.

We then show some generalizations of these instability results to point processes with certain ergodicity properties.

2.1 Poisson Arrivals and Exponential Time-Stamp Increments

2.1.1 Conservative Sequencing. In this section we assume that the sequencer uses the conservative sequencing algorithm.

Arriving messages queue up in their respective queues, at the sequencer, in their order of arrival. If both queues are nonempty, then the sequencer takes the head-of-the-line (HOL) message with the smaller time-stamp and forwards it to the processor. We assume that the service time for doing this work is negligible and take it to be zero.² If either of the queues is empty then the sequencer does not know the time-stamp order of the HOL message in the nonempty queue and does not forward any message to the processor. It follows that at most one of the message queues at the sequencer is ever nonempty and, in fact, exactly one of the queues is always nonempty.

We study the process of the number of unsequenced messages, embedded at the arrival epochs in the superposition of the two message arrival streams. Let X_n denote the number of unsequenced messages just after the n th message arrival. If there are k messages in queue 1, then $X_n = +k$, whereas if there are k messages in queue 2, then $X_n = -k$. In this model for conservative sequencing, $X_n \neq 0$ for all n .

We assume that the two message arrival streams form Poisson processes with rates ν_1 and ν_2 , respectively, and that the successive time-stamps in each stream are Poisson epochs with rates, λ_1 and λ_2 , respectively.

Remark. Before we proceed with the analysis of this model, we note here that this model is not vacuous, but is obtained as the model of a logical process in a distributed simulator of an open feedforward queueing network. Consider the queueing model in Figure 3 with external Poisson

²This assumption does not affect the stability result. For the purpose of simulator performance analysis, the event synchronization time can be added to the event processing time in the event processor.

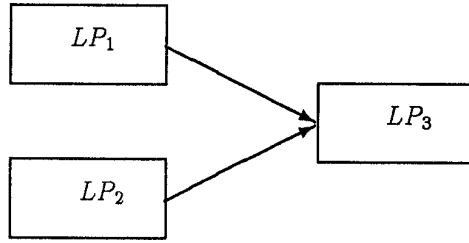


Fig. 4. Distributed simulator for the model in Figure 3.

arrivals and exponential service times, $\lambda_1 < \mu_1$, $\lambda_2 < \mu_2$, $\lambda_1 + \lambda_2 < \mu_3$, and the queues Q_1 and Q_2 being stationary.

The model in Figure 3 is mapped onto the distributed simulator in Figure 4 in the obvious way. LP_1 and LP_2 simulate the work in the system [Kleinrock 1975] in Q_1 and Q_2 , and thus are driven by the two arrival processes. LP_i ($i \in \{1, 2\}$) progresses by generating an interarrival time with distribution exponential (λ_i), updating the work in the system process for Q_i and generating a departure event corresponding to the arrival. Since the queues are stationary, the departure processes in the queueing model are Poisson with rates λ_1 and λ_2 , respectively. If it takes LP_i an exponentially distributed amount of time with mean ν_i^{-1} to do the work corresponding to each arrival, then we get a model for LP_3 that is exactly the same as previously described.

Define

$$\frac{\nu_1}{\nu_1 + \nu_2} := \alpha, \quad \frac{\lambda_1}{\lambda_1 + \lambda_2} := \sigma.$$

If the rates ν_1 , ν_2 and λ_1 , λ_2 are strictly greater than zero, then $0 < \alpha < 1$, and $0 < \sigma < 1$.

THEOREM 1. $\{X_n, n \geq 0\}$ is a Markov Chain on $\{\dots, -3, -2, -1\} \cup \{1, 2, 3, 4, \dots\}$ with transition probabilities:

for $i \geq 1$

$$\begin{aligned} p_{i,i+1} &= \alpha \\ p_{i,i-j} &= (1 - \alpha)\sigma^j(1 - \sigma) \quad \text{for } 0 \leq j \leq i - 1 \\ p_{i,-1} &= (1 - \alpha)\sigma^i \end{aligned}$$

and

$$\begin{aligned} p_{-i,-(i+1)} &= (1 - \alpha) \\ p_{-i,-(i-j)} &= \alpha(1 - \sigma)^j\sigma \quad \text{for } 0 \leq j \leq i - 1. \\ p_{-i,1} &= \alpha(1 - \sigma)^i \end{aligned}$$

PROOF. The result is intuitively clear from the memoryless properties of the Poisson process and the exponential distribution. We present, however, a careful proof in the Appendix. \square

THEOREM 2. (i) For all $\nu_1, \nu_2, \lambda_1, \lambda_2$, except those for which $\nu_1/\lambda_1 = \nu_2/\lambda_2$, the Markov chain $\{X_n\}$ is transient. (ii) For $\nu_1/\lambda_1 = \nu_2/\lambda_2$, $\{X_n\}$ is null recurrent.

PROOF. These conclusions follow from standard Markov chain results. The detailed analysis is given in the Appendix. \square

It follows that for all instances of the problem the message queues are unstable. In particular, we conclude from Theorem 2 that if $\nu_1/\lambda_1 < \nu_2/\lambda_2$ then the queue of messages received from LP_2 will grow without bound. Observe that ν_i/λ_i has the interpretation of “rate of virtual time arrival per unit real-time”; hence the result is intuitive. In practice, of course, the downstream LP must flow control the upstream LP to prevent unbounded message queues from being formed.

2.1.2 Sequencing with Maximum Lookahead. An optimistic sequencer works as in the case of conservative sequencing whenever both the message queues are nonempty. When a queue is empty, however, the processor is allowed to process messages in the nonempty queue. Messages whose time-stamps precede that of the next message to arrive in the empty queue will get processed correctly. The rest will have to be reprocessed. Thus at any time there are messages that cannot be processed correctly even if the sequencer had maximum lookahead, that is, (somehow) knew the time-stamp of the next message to arrive in the empty queue. Since we are limiting ourselves to sequencers that use only time-stamp information for sequencing, these are the messages that optimistic sequencing (in fact, any sequencing algorithm) cannot process correctly until the next message in the empty queue is received. We show that the number of these messages forms a transient or null recurrent Markov chain under the same assumptions and conditions as before. In optimistic sequencing, all such messages will be either in the input message queues, or in the queue of processed but uncommitted messages.

Note that in conservative sequencing with lookahead, the best that lookahead can do is to let the sequencer know the time-stamp of the next message to arrive at the empty message queue. Hence our term *maximum lookahead*. Maximum lookahead is an unachievable algorithm, but its analysis should yield fundamental limits on the performance of any sequencing algorithm.

Let $\{X_n, n \geq 0\}$ denote the number of unsequenced messages just after n th arrival, when the sequencer has maximum lookahead, with the same stochastic assumptions and notation as before. Observe that now X_n can be 0. Again we find that the message queues are unstable.

THEOREM 3. $\{X_n, n \geq 0\}$ is a Markov chain on $\{\dots, -3, -2, -1, 0, 1, 2, 3, 4, \dots\}$ with transition probabilities:

for $i \geq 1$

$$p_{i,i+1} = \alpha$$

$$p_{i,i-j} = (1 - \alpha)\sigma^j(1 - \sigma) \quad 0 \leq j \leq i - 1$$

$$p_{i,0} = (1 - \alpha)\sigma^i$$

$$p_{-i,-(i+1)} = 1 - \alpha$$

$$p_{-i,-(i-j)} = \alpha(1 - \sigma)^j\sigma \quad 0 \leq j \leq i - 1$$

$$p_{-i,0} = \alpha(1 - \sigma)^i$$

$$p_{0,1} = \alpha(1 - \sigma)$$

$$p_{0,-1} = (1 - \alpha)\sigma$$

$$p_{00} = 1 - (p_{0,1} + p_{0,-1}).$$

PROOF. Similar to Theorem 2. \square

THEOREM 4. (i) $\{X_n, n \geq 0\}$ is transient except when $\nu_1/\lambda_1 = \nu_2/\lambda_2$; (ii) $\{X_n, n \geq 0\}$ is null recurrent when $\nu_1/\lambda_1 = \nu_2/\lambda_2$.

PROOF. Exactly the same as for Theorem 2. \square

It follows that the resequencing problem is fundamentally unstable, and no sequencing algorithm, which does not exercise some form of interprocessor flow control, will yield stable message queues. In practical implementations, finite message buffers and communication flow control in the operating system automatically impose flow control between the LPs. In addition, many authors have proposed and studied flow controls between LPs that further limit the asynchrony among the various processes. For example, a buffer level based backpressure control can be applied by downstream LPs, or various LPs can be prevented from getting too far apart in virtual time by means of a mechanism such as time windows [Sokol et al. 1991; Nicol et al. 1989] or bounded lag [Lubachevsky 1989]. What we have shown here is that the “open-loop” system is unstable.

2.2 Generalizations of the Stochastic Assumptions

We now provide instability results for more general message arrival and time-stamp increment processes.

2.2.1 Unequal Rates of Virtual Time Advance. Denote by $N_i(t)$, $i = 1, 2$, the message arrival counting process at input stream i of the event

sequencer. Assume that $N_i(t)$ has an arrival rate ν_i , that is, with probability one (w.p. 1),

$$\lim_{t \rightarrow \infty} \frac{N_i(t)}{t} = \nu_i > 0.$$

Denote by $V_n^{(i)}$, $i = 1, 2$, the time-stamp of the n th arrival in stream i . Assume that $V_n^{(i)}$ has an ‘‘average time-stamp increment’’; that is,

$$\lim_{n \rightarrow \infty} \frac{V_n^{(i)}}{n} = \lambda_i^{-1} > 0 \quad w.p. 1.$$

It is clear that the assumptions on $N_i(t)$ and $V_n^{(i)}$ hold for the stochastic model in Section 2.1.

THEOREM 5. *If $\nu_1/\lambda_1 \neq \nu_2/\lambda_2$ then*

$$|V_{N_1(t)}^{(1)} - V_{N_2(t)}^{(2)}| \rightarrow \infty \quad w.p. 1.$$

PROOF.

$$\frac{V_{N_1(t)}^{(1)} - V_{N_2(t)}^{(2)}}{t} = \frac{N_1(t)}{t} \frac{V_{N_1(t)}^{(1)}}{N_1(t)} - \frac{N_2(t)}{t} \frac{V_{N_2(t)}^{(2)}}{N_2(t)}.$$

Letting $t \rightarrow \infty$, and noting that $\nu_i > 0$ implies $N_i(t) \rightarrow \infty$, the preceding expression converges to

$$\frac{\nu_1}{\lambda_1} - \frac{\nu_2}{\lambda_2} \quad w.p. 1.$$

Now

$$\frac{\nu_1}{\lambda_1} - \frac{\nu_2}{\lambda_2} > 0 \Rightarrow (V_{N_1(t)}^{(1)} - V_{N_2(t)}^{(2)}) \rightarrow +\infty \quad w.p. 1, \quad (1)$$

whereas

$$\frac{\nu_1}{\lambda_1} - \frac{\nu_2}{\lambda_2} < 0 \Rightarrow (V_{N_1(t)}^{(1)} - V_{N_2(t)}^{(2)}) \rightarrow -\infty \quad w.p. 1. \quad (2)$$

Hence the result follows. \square

Observe that if $V_{N_1(t)}^{(1)} - V_{N_2(t)}^{(2)} > 0$ then, for conservative sequencing, this difference is the amount by which the time-stamp of the last message in input queue 1 exceeds the virtual time at the event sequencer. If $\nu_1/\lambda_1 > \nu_2/\lambda_2$, then this difference increases without bound.

2.2.2 Equal Rates of Virtual Time Advance. It is not surprising that when $\nu_1/\lambda_1 \neq \nu_2/\lambda_2$, the queues at the event sequencer are unstable, for if the real-time rate of virtual time advance of stream 1 is greater than that of stream 2, it is intuitively clear that queue 1 will be unstable.

The more interesting case is when the real-time rates of virtual time advance in the two input streams are equal, that is, when $\nu_1/\lambda_1 = \nu_2/\lambda_2$. With the earlier assumption that the two message arrival streams form a Poisson process with rates ν_1 and ν_2 , respectively, and that the successive time-stamps in each stream are Poisson epochs with rates λ_1 and λ_2 , respectively, we showed that the Markov Chain of the number of unsequenced messages just after the n th message arrival is null recurrent when $\nu_1/\lambda_1 = \nu_2/\lambda_2$.

Motivated by the previous result, we prove a form of instability of the message queues when the arrival processes are very general and the time-stamp arrival rates are balanced. We, however, take a slightly different approach in this section. Instead of carrying a time-stamp, each message carries a sequence number, which is unique across both streams, and within a stream the messages arrive in sequence number order. Each sequence number (1, 2, 3, . . .) is assigned to exactly one message of one of the two streams. Obviously, time-stamps imply such a sequence numbering, but sequence numbers provide more information to the sequencer than time-stamps alone. In fact, sequencing of such sequence-numbered streams is equivalent to sequencing time-stamped streams with maximum lookahead.

In the remaining part of this section, we use the following notation. As before, let $N_1(t)$ and $N_2(t)$ be two point processes representing the inputs to the two event sequencer queues. Define $A(n)$ to be the assignment process that assigns global sequence numbers to the points of N_1 and N_2 , that is,

$$A(n) = \begin{cases} 1 & \text{if sequence number } n \text{ is assigned to } N_1, \\ 2 & \text{otherwise.} \end{cases}$$

The sequencer forwards messages in N_1 and N_2 to the event processor in the proper global order determined by their sequence numbers. The forwarding is assumed to be instantaneous.

For $i = 1, 2$, let $E_j^{(i)}$ denote the sequence number assigned to the j th message in the process $N_i(t)$. Thus the sequence number of the most recent message (up to time t) of $N_i(t)$ will be $E_{N_i(t)}^{(i)}$, since $N_i(t)$ is assumed to have generated $N_i(t)$ messages in the interval $[0, t]$.

Let, for $i = 1, 2$, $F^{(i)}(t)$ denote the number of messages in $N_i(t)$ that have been forwarded to the event processor up to time t . Thus the number of messages in $N_i(t)$ that are waiting in the sequencer queue at time t equals $N_i(t) - F^{(i)}(t)$.

With sequence numbers, a message ξ generated by $N_1(t)$ or $N_2(t)$ can be forwarded to the event processor if all messages with sequence numbers up to 1 less than the sequence number of ξ have been forwarded. Such is not the case with time-stamps since if the other queue is empty one cannot be sure if the next arrival into that queue will carry a larger time-stamp. The

following sequencing strategy essentially models with sequence numbers the behavior of the conservative strategy using time-stamps.

According to this strategy, if $E_{N_1}^{(1)}(t) > E_{N_2}^{(2)}(t)$, then $F^{(1)}(t) = N_1(t')$ where $t' = \max\{\tau: E_{N_1}^{(1)}(\tau) < E_{N_2}^{(2)}(t)\}$ and $F^{(2)}(t) = N_2(t)$. Similarly, if $E_{N_2}^{(2)}(t) > E_{N_1}^{(1)}(t)$, then $F^{(1)}(t) = N_1(t)$ and $F^{(2)}(t) = N_2(t')$, where $t' = \max\{\tau: E_{N_2}^{(2)}(\tau) < E_{N_1}^{(1)}(t)\}$. In other words, if at time t , the sequence number of the most recent arrival into Q_1 is greater than that of Q_2 , then at time t the sequencer will have forwarded all arrivals into Q_2 up to t whereas, as far as arrivals into Q_1 are concerned, the sequencer will have forwarded only those whose sequence numbers are less than that of the most recent arrival into Q_2 . It is easy to see that this strategy mimics the behavior of the conservative strategy based on time-stamps.

Let $Q^{(i)}(t)$ denote the number of unsequenced messages waiting in queue Q_i at time t . Under the conservative strategy then,

$$\begin{aligned} Q^{(i)}(t) &= N_i(t) - F^{(i)}(t) \\ &= \max\{N_i(t) - N_i(\tau), 0\}, \end{aligned}$$

where $\tau = \max\{\tau' : E_{N_1}^{(1)}(\tau') < E_{N_2}^{(2)}(t)\}$.

Define $S(t) := Q^{(1)}(t) + Q^{(2)}(t)$ to be the number of events awaiting synchronization. Assume that the counting process $N_1(t)$ and $N_2(t)$ both have a “rate”; that is,

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{N_1(t)}{t} &= \nu_1, \\ \lim_{t \rightarrow \infty} \frac{N_2(t)}{t} &= \nu_2. \end{aligned}$$

THEOREM 6. $N_1(t)$ and $N_2(t)$ are general point processes with rates ν_1 and ν_2 . $A(n)$ is a Bernoulli process independent of $N_1(t)$ and $N_2(t)$; that is,

$$A(n) = \begin{cases} 1 & w.p. \ \alpha \\ 2 & w.p. \ 1 - \alpha, \end{cases}$$

and the $A(n)$, $n \geq 1$, are i.i.d. If $\nu_1/\alpha = \nu_2/(1 - \alpha)$, then $E[S^2(t)] \rightarrow \infty$ as $t \rightarrow \infty$; that is, the second moment (and hence the variance) of the number of events awaiting synchronization increases without bound.

Remark. Observe that $1/\alpha$ (resp., $1/(1 - \alpha)$) is the mean increment of the sequence number between successive messages in stream 1 (resp., stream 2). Hence ν_1/α (resp., $\nu_2/(1 - \alpha)$) is the rate of the sequence number increment in stream 1 (resp., stream 2) per unit “wall-clock” time. Also note that Poisson time-stamp processes (as in Section 2.1) yield a Bernoulli sequence number allocation with $\alpha = \lambda_1/(\lambda_1 + \lambda_2)$.

PROOF. Let $P[n, m; l, k]$ denote the joint probability that the l th arrival into Q_1 has sequence number n and the k th arrival into Q_2 has sequence number m . Then it can be shown that:

$$P[n, m; l, k] = \begin{cases} \alpha^l (1 - \alpha)^{n-l} \binom{n-m-1}{n-l-k} \binom{m-1}{k-1} & \text{for } n > m \\ \alpha^{m-k} (1 - \alpha)^k \binom{m-n-1}{m-l-k} \binom{n-1}{l-1} & \text{for } m > n. \end{cases}$$

Consider the quantity:

$$E \left[Q^{(1)}(t)(Q^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} Q^{(2)}(t)(Q^{(2)}(t) + 1) \right].$$

We can write:

$$\begin{aligned} & E \left[Q^{(1)}(t)(Q^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} Q^{(2)}(t)(Q^{(2)}(t) + 1) \middle| N_1(t) = l, N_2(t) = k \right] \\ &= \sum_{n=l+k}^{\infty} \sum_{m=k}^{l+k-1} P[n, m; l, k] (l - (m - k))(l - (m - k) + 1) \\ &\quad + \frac{\alpha^2}{(1 - \alpha)^2} \sum_{m=l+k}^{\infty} \sum_{n=l}^{l+k-1} P[n, m; l, k] (k - (n - l))(k - (n - l) + 1) \\ &= \sum_{n=l+k}^{\infty} \sum_{m=k}^{l+k-1} (l + k - m)(l + k - m + 1) \alpha^l (1 - \alpha)^{n-l} \binom{n-m-1}{n-l-k} \binom{m-1}{k-1} \\ &\quad + \sum_{m=l+k}^{\infty} \sum_{n=l}^{l+k-1} (l + k - n)(l + k - n + 1) \alpha^{m-k} (1 - \alpha)^k \binom{m-n-1}{m-l-k} \\ &\quad \cdot \binom{n-1}{l-1} \frac{\alpha^2}{(1 - \alpha)^2}. \end{aligned}$$

It can be shown that the preceding expression reduces to (details of the proof are given in the Appendix):

$$l(l+1) - \frac{2\alpha lk}{1-\alpha} + \frac{\alpha^2 k(k+1)}{(1-\alpha)^2} = \left(l - \frac{\alpha k}{1-\alpha} \right)^2 + l + \frac{k\alpha^2}{(1-\alpha)^2}.$$

Thus

$$E \left[Q^{(1)}(t)(Q^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} Q^{(2)}(t)(Q^{(2)}(t) + 1) | N_1(t) = l, N_2(t) = k \right] \\ = \left(l - \frac{\alpha k}{1 - \alpha} \right)^2 + l + \frac{k\alpha^2}{(1 - \alpha)^2}.$$

Therefore

$$E \left[Q^{(1)}(t)(Q^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} Q^{(2)}(t)(Q^{(2)}(t) + 1) \right] \\ = E \left[\left(N_1(t) - \frac{\alpha N_2(t)}{1 - \alpha} \right)^2 \right] + E[N_1(t)] + \frac{\alpha^2 E[N_2(t)]}{(1 - \alpha)^2}.$$

Therefore, even if $\nu_1/\alpha = \nu_2/(1 - \alpha)$,

$$E \left[Q^{(1)}(t)(Q^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} Q^{(2)}(t)(Q^{(2)}(t) + 1) \right] \rightarrow \infty \quad \text{as } t \rightarrow \infty.$$

Now

$$Q^{(1)^2}(t) + Q^{(1)}(t) + \frac{\alpha^2}{(1 - \alpha)^2} [Q^{(2)^2}(t) + Q^{(2)}(t)] \leq \\ \begin{cases} \frac{\alpha^2}{(1 - \alpha)^2} [Q^{(1)^2}(t) + Q^{(1)}(t) + Q^{(2)^2}(t) + Q^{(2)}(t)] & \text{if } \frac{\alpha}{1 - \alpha} \geq 1, \\ Q^{(1)^2}(t) + Q^{(1)}(t) + Q^{(2)^2}(t) + Q^{(2)}(t) & \text{if } \frac{\alpha}{1 - \alpha} < 1. \end{cases}$$

Therefore

$$Q^{(1)^2}(t) + Q^{(1)}(t) + \frac{\alpha^2}{(1 - \alpha)^2} [Q^{(2)^2}(t) + Q^{(2)}(t)] \\ \leq \max \left(1, \frac{\alpha^2}{(1 - \alpha)^2} \right) [Q^{(1)^2}(t) + Q^{(1)}(t) + Q^{(2)^2}(t) + Q^{(2)}(t)] \\ \leq \max \left(1, \frac{\alpha^2}{(1 - \alpha)^2} \right) [S^2(t) + S(t)],$$

where $S(t)$ is the number of events awaiting synchronization. Thus, as $t \rightarrow \infty$, $E[S^2(t) + S(t)] \rightarrow \infty$ as long as $N_1(t)$ and $N_2(t)$ increase without bound as $t \rightarrow \infty$.

Since $S(t) \geq 1$ for all t ,

$$E[S^2(t) + S(t)] \leq 2E[S^2(t)].$$

Thus $E[S^2(t)] \rightarrow \infty$ as $t \rightarrow \infty$. \square

The preceding result shows that the second moment of $S(t)$, the number of events awaiting synchronization, increases without bound as $t \rightarrow \infty$ under very weak assumptions about the arrival process $N_1(t)$ and $N_2(t)$ (as long as $A(n)$ is Bernoulli). This means that either the mean of the number of events awaiting synchronization increases without bound, or, if the mean is bounded, then the variance grows without bound.

3. COMPLEMENTS AND EXTENSIONS

In this section we present some complements and extensions of the results in Section 2.

3.1 Departure Process from the Sequencer

Consider two time-stamped message streams (with assumptions as in Section 2.1) arriving at a sequencer; let the parameters of the streams be denoted by (ν_i, λ_i) , $i \in \{1, 2\}$. We assume that $\nu_1/\lambda_1 \neq \nu_2/\lambda_2$. If the messages departing from the sequencer are fed to an event processor or another event sequencer, then it is necessary to investigate the characteristics of the departure process. Suppose $\nu_1/\lambda_1 < \nu_2/\lambda_2$, then the queue of messages from stream 2 increases without bound. It is easily seen that a batch of sequenced messages departs whenever a message arrives from stream 1 (one from queue 1 and the rest from queue 2, which can be assumed to be infinite). The batch size is geometrically distributed (with at least one message in each batch) with mean $(1 + \lambda_2/\lambda_1)$. Hence the message departure rate is $\nu_1(1 + \lambda_2/\lambda_1)$. Furthermore, for both conservative sequencing and maximum lookahead, it can be shown that the epochs of message departures forms a renewal process (of rate $\nu_1(1 + \lambda_2/\lambda_1)$), and the time-stamp sequence of the departing messages is an independent Poisson process with rate $(\lambda_1 + \lambda_2)$.

This result has an immediate consequence for the stability of the event processor that follows the event sequencer. If the event service times at the processor are i.i.d. with mean $1/\nu$, it follows that the event processor queue is stable if $\nu_1(1 + \lambda_2/\lambda_1) < \nu$. The corresponding result holds for $\nu_1/\lambda_1 > \nu_2/\lambda_2$.

We have also shown in Shorey [1996] that flow controlled throughput (e.g., with buffer limit flow control [Shorey 1996] or Moving Time Windows [Sokol et al. 1991]) is bounded above by this unstable throughput of the event sequencer calculated in the preceding. Thus flow control does not help to speed up the simulation; the open-loop throughput provides a fundamental bound.

Finally, observe that if the physical process represents a stationary open feedforward Jackson-type queueing network, then the time-stamp process at the output of the event processor will be Poisson [Wolff 1989].

3.2 More than 2 Message Streams

Consider n time-stamped message streams with renewal arrival epochs, and the time-stamps forming Poisson processes. Let (ν_i, λ_i) , $1 \leq i \leq n$, denote the parameters of the stream; we assume here that $\nu_i/\lambda_i \neq \nu_j/\lambda_j$, $i \neq j$. Let $j^* = \min_{1 \leq i \leq n} \nu_i/\lambda_i$. These n streams are offered to a sequencer. If we view the sequencing as being done on pairs of streams, and then on pairs of the resulting departure processes, and so on, it easily follows that

- (i) Queues i , $i \neq j^*$, $1 \leq i \leq n$, are unstable.
- (ii) The departure process from the sequencer comprises geometrically distributed batches of messages (with at least one message in each batch) departing at the arrival epochs of stream j^* . The mean batch size is $1 + 1/\lambda_{j^*} \sum_{i=1, i \neq j^*}^n \lambda_i = 1/\lambda_{j^*} \sum_{i=1}^n \lambda_i$
- (iii) The message departure epochs form a renewal process of rate $\nu_{j^*}/\lambda_{j^*} \sum_{i=1}^n \lambda_i$, and the time-stamp process is Poisson with rate $\sum_{i=1}^n \lambda_i$. We denote such a stream by $(\nu_{j^*}, \lambda_{j^*}, \sum_{i=1}^n \lambda_i)$; observe that in this notation, a Poisson message stream of rate ν , with Poisson time stamps of rate λ is denoted by (ν, λ, λ) .

3.3 Feedforward Queueing Networks

We restrict our discussion to simulators of feedforward queueing networks (FQNs) with no “split routing” in the queueing network; that is, all customers departing a queue enter exactly one downstream queue.

With this restriction on the routing, the topology of the FQN is just a tree, with each node of the tree representing a synchronization/queueing station. The root of the tree represents the penultimate queue; customers enter at the leaves and flow out of the root. The tree is m levels deep if m is the maximum number of queues that any customer traverses. A queue is at level i , $1 \leq i \leq m$ if a customer leaving it has $m - i$ queues left to traverse; the level of the root is m . Each queue may receive an external Poisson arrival process. Figure 5 shows an example of such an FQN with $m = 4$.

We also assume that each queue in the queueing network model is simulated by an LP on a separate processor; thus the message flow between LPs has the same topology as the customer flow in the queueing network. The LPs representing the queues on leaves of the tree are like sources of messages. Recalling the notation introduced in Section 3.2, let the message streams flowing out of the LPs at stage i , $1 \leq i \leq m$, be denoted by $(\nu_1^{(i)}, \lambda_1^{(i)}, \hat{\lambda}_1^{(i)})$, \dots , $(\nu_{n_i}^{(i)}, \lambda_{n_i}^{(i)}, \hat{\lambda}_{n_i}^{(i)})$, where $\hat{\lambda}_j^{(i)} \geq \lambda_j^{(i)}$. An LP at level i that represents a leaf node will have a flow out of it of the form (ν, λ, λ) . If an LP is not representing a leaf queue and has an external Poisson arrival process of rate λ , this external arrival process can be viewed as a message stream with parameter $(\infty, \lambda, \lambda)$.

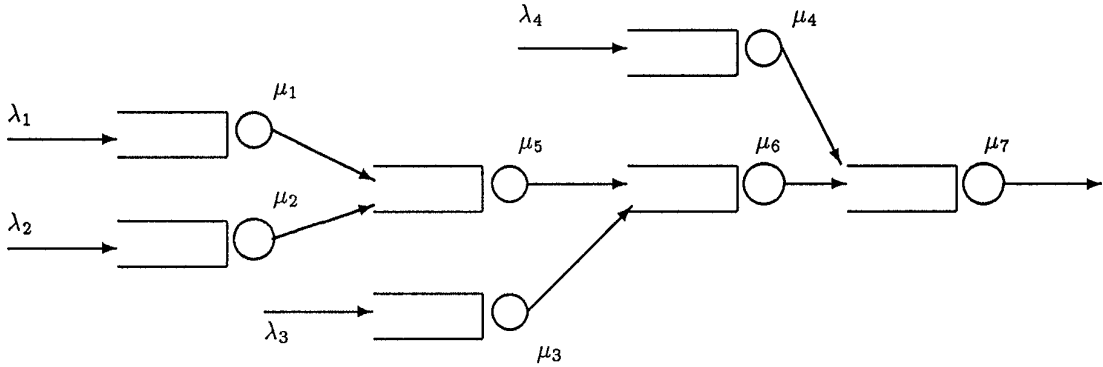


Fig. 5. A feedforward queueing network.

Denote by $\nu_j, 1 \leq j \leq L$, the processor service rates of the LPs at the leaf nodes, and by $\lambda_j, 1 \leq j \leq L$, the external arrival rates at the corresponding queues. Let $\hat{\lambda}$ denote the total external arrival rate in the queueing model. As in Section 3.2, we assume that $\nu_i/\lambda_i \neq \nu_j/\lambda_j, i \neq j$.

THEOREM 7. *If the processors of the LPs at the nonleaf nodes have an infinite service rate, then the departure process of the simulator is $(\nu_{j^*}, \lambda_{j^*}, \lambda)$ where $j^* = \arg \min_{1 \leq j \leq L} (\nu_j/\lambda_j)$.*

PROOF. Easy observation from the results of Section 3. The restriction of infinite service rate is required here as we have proved the previous results only for renewal message arrival processes. □

Thus the throughput of the simulator of the FQN is given by:

$$\left(\min_{1 \leq j \leq L} \frac{\nu_j}{\lambda_j} \right) (\hat{\lambda})$$

This formula clearly shows the effect of mapping the model onto the simulator processors; for example, the worst performance will be obtained if the leaf queue with the largest λ_j is simulated by the slowest processor (i.e., the least ν_j).

Example. Consider the FQN shown in Figure 5. The simulator for the FQN is shown in Figure 6.

Let $\mu_i, i = 1, 2, \dots, 7$ denote the service rates of the queues of the FQN. We assume that these queues are stable; that is, $\lambda_i < \mu_i, i = 1, 2, 3, 4, \sum_{i=1}^2 \lambda_i < \mu_5, \sum_{i=1}^3 \lambda_i < \mu_6, \sum_{i=1}^4 \lambda_i < \mu_7$. With reference to the previous description, the leaf queues in the FQN (i.e., queues 1, 2, 3, and 4) are simulated by LPs with event processing rates $\nu_i, 1 \leq i \leq 4$. The remaining event processors have service rate ν . Hence we obtain the simulation model shown in Figure 6.

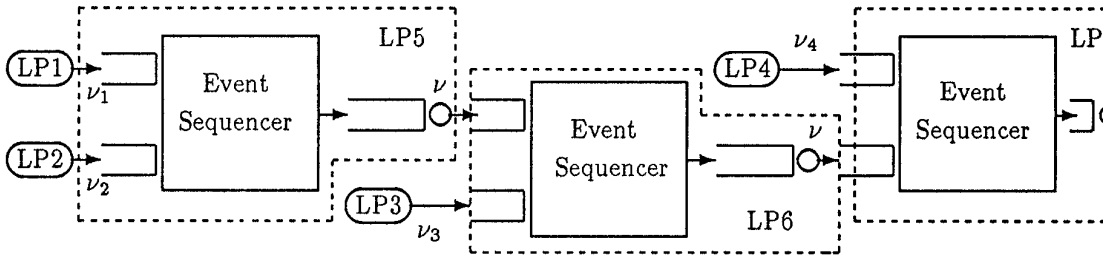


Fig. 6. Feedforward queueing network simulator.

Table I shows the throughput of the simulator obtained from analysis and simulation as a function of the parameters of the PP and the LP. In the example, we keep $\nu = 10$, and $\mu_i = 1$ for all $i = 1, 2, 3, 4$.

It can be seen that the throughput obtained from analysis (γ_{Anal}) matches very well with that obtained from simulation (γ_{Sim}), the difference being due to the fact that γ_{Anal} is a long-run average, whereas γ_{Sim} is a finite sample average.

4. CONCLUSION

The instability results obtained in this article are not surprising if viewed in the light of similar results obtained for other queueing models. It is easy to see that event synchronization is similar to the assembly problem arising in manufacturing systems. If the parts to be assembled come from independent streams, it was shown by Harrison [1973] and Latouche [1981] that under fairly general conditions the queues of parts to be assembled are unstable. The assumption of independent part streams may not always be appropriate in the manufacturing context, as the part streams usually originate from a common order stream. No such parent stream can be argued in the context of distributed simulation of open queueing networks. Hence if all logical processors are permitted to proceed at their own rates, then message buffers will overflow. Such simulations must be stabilized by some form of interprocessor flow control. For example, a buffer level based backpressure control can be applied by downstream LPs, or various LPs can be prevented from getting too far apart in virtual time by means of a mechanism such as time windows [Sokol et al. 1991] or bounded lag [Lubachevsky 1989].

Although such mechanisms will serve to stabilize buffers, our approach of modeling and analyzing the message flow processes in the simulator has pointed towards certain fundamental limits of efficiency of distributed simulation imposed by the synchronization mechanism. In subsequent work [Shorey 1996], we have shown for the simple models considered in the article that flow-controlled throughput is bounded above by the open-loop throughput.

It is clear that the rate of departure of processed messages from the simulator corresponds to the rate of progress of the simulation. We have obtained formulas or bounds for the throughput of simulators of feedfor-

Table I. Throughput of the Feedforward Queueing Network Simulator (analysis and simulation).

ν_1	ν_2	ν_3	ν_4	λ_1	λ_2	λ_3	λ_4	μ_5	μ_6	μ_7	γ <i>Anal</i>	γ <i>Sim</i>
0.2	0.3	0.25	0.25	0.25	0.25	0.25	0.25	0.6	0.8	1.2	0.8	0.79494
0.3	0.1	0.4	0.2	0.25	0.25	0.25	0.25	0.6	0.8	1.2	0.4	0.39641
0.1	0.3	0.2	0.4	0.5	0.2	0.1	0.2	0.8	0.9	1.2	0.2	0.19837

ward networks. These formulas involve parameters of the simulator (processor speeds) and the model being simulated, and hence clearly demonstrate the performance impact of various ways of mapping the simulation model onto the processors.

In subsequent work (see, e.g., Shorey [1996] and Gupta et al. [1996]) we have attempted to develop more detailed formal models for message flows in distributed simulators of queueing networks, and to study the stability and performance of these models; in particular, we have explored the stability of simulators of queueing networks with feedbacks. We expect that this approach will yield useful insights into the performance limits of distributed simulators and how the performance could be optimized.

APPENDIX A. Proof of Theorems

PROOF OF THEOREM 1. Let τ_n denote the “virtual” time up to which synchronization is complete just after the n th arrival epoch. Note that τ_n is the time-stamp of the last message allowed to pass through at the n th arrival. Time-stamps of queued messages and messages yet to arrive are viewed relative to τ_n , as increments beyond τ_n .

The result follows from the following Lemma.

LEMMA. Let $X_n = i$, and the time-stamps of the queued messages relative to τ_n be $S_1, S_1 + S_2, \dots, S_1 + S_2 + \dots + S_i$; here S_1 is the amount by which the time-stamp of the first queued message exceeds τ_n . Since time-stamp increments are exponentially distributed, owing to the memoryless property of exponential distribution, residual time S_1 is also exponentially distributed. $\{S_1, S_2 \dots\}$ are i.i.d., $\text{Exp}(\lambda_1)$. Let T denote the time-stamp of the message arriving at the $(n + 1)$ st arrival epoch relative to τ_n . Let $\{T_1, T_1 + T_2, \dots\}$ denote the time-stamps, relative to τ_{n+1} , of the messages left in queue after the $(n + 1)$ st arrival.

Then

$$\begin{aligned}
 &P(X_{n+1} = j, T_1 > t_1, T_2 > t_2, \dots, T_j > t_j | X_n = i) \\
 &= \begin{cases} \alpha \prod_{k=1}^j e^{-\lambda_1 t_k} & j = i + 1 \quad (i) \\ (1 - \alpha) \sigma^{i-j} (1 - \sigma) \prod_{k=1}^j e^{-\lambda_1 t_k} & 1 \leq j \leq i \quad (ii). \\ (1 - \alpha) \sigma^j e^{-\lambda_2 t_1} & j = -1 \quad (iii) \end{cases}
 \end{aligned}$$

PROOF.

- (i) $j = i + 1$ if the $(n + 1)$ st arrival is from stream 1 (probability α). In that case $\tau_{n+1} = \tau_n$, and $T_1 = S_1, T_2 = S_2, \dots, T_i = S_i, T_{i+1} \sim \text{Exp}(\lambda_1)$ and is independent of the others; (here \sim is to be read “is distributed as”).
- (ii) Let $\ell = i - j$ for $1 \leq j \leq i$. In this case $\tau_{n+1} = T + \tau_n, T_1 = \sum_{k=1}^{\ell+1} S_k - T, T_2 = S_{\ell+2}, \dots, T_j = S_{\ell+j} = S_i$, and $T \sim \text{Exp}(\lambda_2)$

$$\begin{aligned} & P(X_{n+1} = j, T_1 > t_1, \dots, T_j > t_j | X_n = i) \\ &= (1 - \alpha) P(G < T < G + S_{\ell+1}, G + S_{\ell+1} - T > t_1, \\ & \quad S_{\ell+2} > t_2, \dots, S_{\ell+j} > t_j), \end{aligned}$$

where $G := \sum_{k=1}^{\ell} S_k$, and which, letting $g(\cdot)$ be the probability density of G ,

$$\begin{aligned} &= (1 - \alpha) \int_0^{\infty} g(u) du \int_u^{\infty} \lambda_2 e^{-\lambda_2 t} dt e^{-\lambda_1(t_1+t-u)} e^{-\lambda_1 t_2} \dots e^{-\lambda_1 t_j} \\ &= (1 - \alpha) \prod_{k=1}^j e^{-\lambda_1 t_k} \int_0^{\infty} g(u) du \int_u^{\infty} \lambda_2 e^{-\lambda_2 t} dt e^{-\lambda_1(t-u)} \end{aligned}$$

and, letting $t - u = v$,

$$\begin{aligned} &= (1 - \alpha) \prod_{k=1}^j e^{-\lambda_1 t_k} \int_0^{\infty} g(u) e^{-\lambda_2 u} du \int_0^{\infty} \lambda_2 e^{-(\lambda_1 + \lambda_2)v} dv \\ &= (1 - \alpha) \left(\frac{\lambda_1}{\lambda_1 + \lambda_2} \right)^{\ell} \left(\frac{\lambda_2}{\lambda_1 + \lambda_2} \right)^j \prod_{k=1}^j e^{-\lambda_1 t_k} \\ &= (1 - \alpha) \sigma^{i-j} (1 - \sigma) \prod_{k=1}^j e^{-\lambda_1 t_k}. \end{aligned}$$

(iii)

$$P(X_{n+1} = -1, T_1 > t_1 | X_n = i) = (1 - \alpha) P\left(\sum_{k=1}^i S_k < T, T - \sum_{k=1}^i S_k > t_1 \right),$$

letting $G = \sum_{k=1}^i S_k$, and $g(\cdot)$ be the probability density of G ,

$$\begin{aligned} &= (1 - \alpha) \int_0^\infty g(u) du e^{-\lambda_2(t_1+u)} \\ &= (1 - \alpha) \sigma^j e^{-\lambda_2 t_1}. \quad \square \end{aligned}$$

Thus after each arrival epoch, the time-stamps of the queued messages are successive epochs of a Poisson process. Returning to the proof of Theorem 1, let $\tau_0 = 0$, $X_0 = i_0 (\geq 1)$, and let the time-stamps of these queued messages have the same distribution as the first i_0 epochs of a Poisson process of rate λ_1 .

$$P_{\mathcal{P}}(X_{n+1} = j | X_0 = i_0, \dots, X_{n-1} = i_{n-1}, X_n = i_n),$$

where the subscript \mathcal{P} denotes that the initial time stamps form a segment of a Poisson process. Now writing this out,

$$= \frac{P_{\mathcal{P}}(X_1 = i_1, \dots, X_n = i_n, X_{n+1} = j | X_0 = i_0)}{P_{\mathcal{P}}(X_1 = i_1, \dots, X_n = i_n | X_0 = i_0)}.$$

Consider the numerator

$$\begin{aligned} &P_{\mathcal{P}}(X_1 = i_1, \dots, X_n = i_n, X_{n+1} = j | X_0 = i_0) \\ &= \int_{(\mathcal{R}^+)^{i_1}} P_{\mathcal{P}}(X_1 = i_1, S_1^{(1)} \in ds_1, S_2^{(1)} \in ds_2, \dots, S_{i_1}^{(1)} \in ds_{i_1} | X_0 = i_0) \\ &\cdot P(X_2 = i_2, X_3 = i_3, \dots, X_{n+1} = j | X_0 = i_0, X_1 = i_1, S_{i_1}^{(1)} = s_1, \dots, S_{i_1}^{(1)} = s_{i_1}), \end{aligned}$$

where \mathcal{R}^+ is the nonnegative real line, and $\{S_k^{(1)}\}$ are time-stamps of messages queued just after the first arrival, in relation to τ_1 . Now note that since the arrival epochs form a Poisson process, and the time-stamps of the yet to arrive messages also form a Poisson process, independent of the past, the conditioning on $X_0 = i_0$ in the second term under the integral can be dropped, and applying the preceding lemma we get

$$= P_{\mathcal{P}}(X_1 = i_1 | X_0 = i_0) P_{\mathcal{P}}(X_2 = i_2, X_3 = i_3, \dots, X_{n+1} = j | X_1 = i_1).$$

Proceeding this way in the numerator and denominator we will get

$$P_{\mathcal{P}}(X_{n+1} = j | X_0 = i_0, \dots, X_{n-1} = i_{n-1}, X_n = i_n) = P_{\mathcal{P}}(X_{n+1} = j | X_n = i_n),$$

where the transition probabilities are obtained from the Lemma. \square

PROOF OF THEOREM 2. (i) Let Q be the transition probability matrix restricted to the set of states $\{1, 2, 3, \dots\}$. We show that whenever $\lambda_1 \nu_2 \neq$

$\nu_1\lambda_2$, there exists a bounded, nonnegative, nonzero solution to (see Cinlar [1975])

$$Q\underline{y} = \underline{y},$$

that is, (y_1, y_2, \dots) such that, for $i \geq 1$,

$$y_i = \alpha y_{i+1} + \sum_{j=0}^{i-1} (1-\alpha)\sigma^j(1-\sigma)y_{i-j}.$$

Multiplying by z^i , for $0 < z < 1$, and summing from 1 to ∞ ,

$$\sum_{i=1}^{\infty} z^i y_i = \sum_{i=1}^{\infty} \alpha z^i y_{i+1} + (1-\alpha)(1-\sigma) \sum_{i=1}^{\infty} \sum_{j=0}^{i-1} z^i \sigma^j y_{i-j}.$$

Hence, defining $\tilde{y}(z) = \sum_{i=1}^{\infty} z^i y_i$,

$$\tilde{y}(z) = \frac{\alpha}{z} (\tilde{y}(z) - z y_1) + \frac{(1-\alpha)(1-\sigma)}{(1-\sigma z)} \tilde{y}(z),$$

from which, on simplification, we get

$$\tilde{y}(z) = \frac{z(1-\sigma z)}{(1-z)(1-(\sigma/\alpha)z)} y_1.$$

Case (i) $\alpha \neq \sigma$ (i.e., $\nu_1/(\nu_1 + \nu_2) \neq \lambda_1/(\lambda_1 + \lambda_2)$, i.e., $\nu_1\lambda_2 \neq \lambda_1\nu_2$). Using partial fraction expansion

$$\begin{aligned} \tilde{y}(z) &= \frac{\alpha}{\sigma} z \frac{y_1}{(\alpha/\sigma) - 1} \left(\frac{1-\sigma}{1-z} - \frac{1-\alpha}{(\alpha/\sigma) - z} \right) \\ &= \sum_{j=1}^{\infty} z^j \left(\frac{(1-\sigma) - (1-\alpha)(\sigma/\alpha)^j}{1-\sigma/\alpha} \right) y_1, \end{aligned}$$

hence, by definition of $\tilde{y}(z)$, the solutions to $Q\underline{y} = \underline{y}$ for $\sigma \neq \alpha$ are of the form, for $j \geq 1$,

$$y_j = \left(\frac{(1-\sigma) - (1-\alpha)(\sigma/\alpha)^j}{1-\sigma/\alpha} \right) y_1.$$

It is clear that there is a bounded nonzero solution between 0 and 1 if $\sigma < \alpha$ and none if $\sigma > \alpha$. Thus for $\sigma < \alpha$ there are states in $\{1, 2, 3, \dots\}$ from which there is a positive probability of never leaving this set. Hence $\{X_n\}$ would be transient. It is similarly clear that for $(1-\sigma) < (1-\alpha)$, that is,

$\sigma > \alpha$, there are states in $\{\dots, -3, -2, -1\}$ from which there is a positive probability of never leaving $\{\dots, -3, -2, -1\}$. Thus for $\sigma \neq \alpha$, $\{X_n\}$ is transient.

Case (ii) $\sigma = \alpha$.

$$\begin{aligned} \tilde{y}(z) &= z \frac{1 - \sigma z}{(1 - z)^2} y_1 \\ &= \sum_{i=1}^{\infty} z^i (i(1 - \sigma) + \sigma) y_1. \end{aligned}$$

Recalling that $\sigma < 1$, there is no bounded solution to $Q_{\underline{y}} = \underline{y}$; hence $\{X_n\}$ is recurrent for $\sigma = \alpha$.

(ii) From (i) we know that for $\lambda_1/\lambda_2 = \nu_1/\nu_2$ (i.e., $\alpha = \sigma$) $\{X_n\}$ is recurrent. We show now that for $\alpha = \sigma$, $\{X_n\}$ is not positive recurrent, and hence is null.

Consider the Markov chain $\{X'_n\}$ on the state space $\{0, 1, 2, 3, \dots\}$ with the transition probabilities $p'_{i,j}$ given by (recall that $p_{i,j}$ are transition probabilities for $\{X_n\}$):

$$p'_{i,j} = p_{i,j} \quad \text{for } i \geq 1, j \geq 1$$

$$p'_{i,0} = p_{i,-1} \quad \text{for } i \geq 1$$

$$p'_{0,1} = p_{-1,1} = 1 - p'_{0,0}.$$

Observe that $\{X_n\}$ positive recurrent $\Rightarrow \{X'_n\}$ is positive recurrent. We show that for $\alpha = \sigma$, $\{X'_n\}$ is not positive recurrent. To do this we use a result due to Kaplan [1979] (see also Szpankowski [1990]). For $i \geq 1$,

$$\begin{aligned} E(X'_{k+1} - X'_k | X'_k = i) &= \alpha - \sum_{j=0}^{i-1} j(1 - \alpha)\sigma^j(1 - \sigma) - i(1 - \alpha)\sigma^i \\ &= \alpha - \left(\frac{1 - \alpha}{1 - \sigma} \right) \sigma \{1 - \sigma^i\}. \end{aligned}$$

Hence for $\alpha = \sigma := a$ and $i \geq 1$,

$$E(X'_{k+1} - X'_k | X'_k = i) = a^{i+1} > 0 \quad \text{for } a > 0.$$

Also, directly,

$$E(X'_{k+1} - X'_k | X'_k = 0) = a(1 - a) > 0,$$

for $0 < a < 1$. Furthermore, for $i \geq 1$, and $z \in (0, 1]$,

$$\sum_{j=0}^{\infty} p'_{ij}(z^i - z^j) \geq \sum_{0 \leq j < i} p'_{ij}(-(i-j))(1-z),$$

where we use the inequality

$$z^i - z^j \geq -(i-j)^+(1-z),$$

for $z \in [0, 1]$ (see Szpankowski [1985]). Hence for $i \geq 1$, $z \in (0, 1]$ (see preceding mean drift calculations),

$$\sum_{j=0}^{\infty} p'_{ij}(z^i - z^j) \geq -a(1-\alpha^i)(1-z) \geq -a(1-z).$$

Hence the required conditions in Kaplan [1979] are satisfied and $\{X'_k\}$ is not positive recurrent, implying that $\{X_k\}$ is not positive recurrent. It follows that $\{X_k\}$ is null recurrent for $\sigma = \alpha$. \square

PROOF OF THEOREM 6. We give here the details of the proof of Theorem 6. We prove that

$$\begin{aligned} E \left[Q^{(1)}(t)(Q^{(1)}(t) + 1) + \frac{\alpha^2}{(1-\alpha)^2} Q^{(2)}(t)(Q^{(2)}(t) + 1) | N_1(t) = \ell, N_2(t) = k \right] \\ = \ell(\ell + 1) - \frac{2\alpha k \ell}{1-\alpha} + \frac{\alpha^2 k(k+1)}{(1-\alpha)^2}. \end{aligned}$$

Consider $P[n, m; \ell, k] = P[E_\ell^{(1)} = n \text{ and } E_k^{(2)} = m]$.

If $n > m$, then $k \leq m \leq \ell + k - 1$, and $n \geq \ell + k$

If $m > n$, then $\ell \leq n \leq \ell + k - 1$, and $m \geq \ell + k$.

Obviously,

$$\sum_{n=\ell+k}^{\infty} \sum_{m=k}^{\ell+k-1} P[n, m; \ell, -k] + \sum_{m=\ell+k}^{\infty} \sum_{n=\ell}^{\ell+k-1} P[n, m; \ell, k] = 1.$$

Thus

$$\begin{aligned} & \sum_{n=\ell+k}^{\infty} \sum_{m=k}^{\ell+k-1} \alpha^{\ell} (1-\alpha)^{n-\ell} \binom{m-1}{k-1} \binom{n-m-1}{n-\ell-k} \\ & + \sum_{m=\ell+k}^{\infty} \sum_{n=\ell}^{\ell+k-1} \alpha^{m-k} (1-\alpha)^k \binom{n-1}{\ell-1} \binom{m-n-1}{m-\ell-k} = 1. \end{aligned}$$

In the first term substitute $\nu = n - \ell - k$ and $\mu = m - k$; in the second term substitute $\nu = m - \ell - k$ and $\mu = n - \ell$. Then the preceding expression reduces to

$$\begin{aligned} & \sum_{\nu=0}^{\infty} \sum_{\mu=0}^{\ell-1} \alpha^{\ell} (1-\alpha)^{\nu+k} \binom{\mu+k-1}{k-1} \binom{\nu+\ell-\mu-1}{\nu} \\ & + \sum_{\nu=0}^{\infty} \sum_{\mu=0}^{k-1} \alpha^{\nu+\ell} (1-\alpha)^k \binom{\mu+\ell-1}{\ell-1} \binom{\nu+k-\mu-1}{\nu} = 1; \end{aligned}$$

that is,

$$\begin{aligned} & \alpha^{\ell} (1-\alpha)^k \sum_{\nu=0}^{\infty} \left\{ (1-\alpha)^{\nu} \sum_{\mu=0}^{\ell-1} \binom{\mu+k-1}{k-1} \binom{\nu+\ell-\mu-1}{\nu} \right\} \\ & + \alpha^{\ell} (1-\alpha)^k \sum_{\nu=0}^{\infty} \left\{ \alpha^{\nu} \sum_{\mu=0}^{k-1} \binom{\mu+\ell-1}{\ell-1} \binom{\nu+k-\mu-1}{\nu} \right\} = 1. \quad (3) \end{aligned}$$

It can be shown that for $0 < x < 1$, and $n = 0, 1, 2, \dots$,

$$\sum_{\nu=0}^{\infty} x^{\nu} \binom{\nu+n}{n} = (1-x)^{-(n+1)}.$$

Therefore the preceding expression can be written as

$$1 = \alpha^{\ell} (1-\alpha)^k \left\{ \sum_{\mu=0}^{\ell-1} \alpha^{-(\ell-\mu)} \binom{\mu+k-1}{k-1} + \sum_{\mu=0}^{k-1} (1-\alpha)^{-(k-\mu)} \binom{\mu+\ell-1}{\ell-1} \right\};$$

that is,

$$1 = (1 - \alpha)^k \sum_{\mu=0}^{\ell-1} \alpha^\mu \binom{\mu + k - 1}{k - 1} + \alpha^\ell \sum_{\mu=0}^{k-1} (1 - \alpha)^\mu \binom{\mu + \ell - 1}{\ell - 1}.$$

Note that this expression holds for $\ell, k = 1, 2, \dots$.

We write

$$S_{\ell,k} = (1 - \alpha)^k \sum_{\mu=0}^{\ell-1} \alpha^\mu \binom{\mu + k - 1}{k - 1} + \alpha^\ell \sum_{\mu=0}^{k-1} (1 - \alpha)^\mu \binom{\mu + \ell - 1}{\ell - 1} = 1. \quad (4)$$

We now show that

$$\begin{aligned} E \left[\mathbf{Q}^{(1)}(t)(\mathbf{Q}^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} \mathbf{Q}^{(2)}(t)(\mathbf{Q}^{(2)}(t) + 1) \mid N_1(t) = \ell, N_2(t) = k \right] \\ = \ell(\ell + 1) - \frac{2\alpha}{(1 - \alpha)} \ell k + \frac{\alpha^2}{(1 - \alpha)^2} k(k + 1). \end{aligned}$$

In particular, we have

$$\begin{aligned} E \left[\mathbf{Q}^{(1)}(t)(\mathbf{Q}^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} \mathbf{Q}^{(2)}(t)(\mathbf{Q}^{(2)}(t) + 1) \mid N_1(t) = \ell, N_2(t) = k \right] \\ = \sum_{n=\ell+k}^{\infty} \sum_{m=k}^{\ell+k-1} (\ell + k - m)(\ell + k - m + 1) \alpha^\ell (1 - \alpha)^{n-\ell} \binom{m-1}{k-1} \\ \cdot \binom{n-m-1}{n-\ell-k} \\ + \sum_{m=\ell+k}^{\infty} \sum_{n=\ell}^{\ell+k-1} (\ell + k - n)(\ell + k - n + 1) \alpha^{m-k} (1 - \alpha)^k \binom{n-1}{\ell-1} \\ \cdot \binom{m-n-1}{m-\ell-k} \frac{\alpha^2}{(1 - \alpha)^2} \end{aligned}$$

$$\begin{aligned}
 &= \alpha^\ell (1 - \alpha)^k \sum_{\nu=0}^{\infty} \left\{ (1 - \alpha)^\nu \sum_{\mu=0}^{\ell-1} (\ell - \mu)(\ell - \mu + 1) \binom{\mu + k - 1}{k - 1} \binom{\nu + \ell - \mu - 1}{\nu} \right\} \\
 &\quad + \alpha^\ell (1 - \alpha)^k \sum_{\nu=0}^{\infty} \left\{ \alpha^\nu \sum_{\mu=0}^{k-1} (k - \mu)(k - \mu + 1) \binom{\mu + \ell - 1}{\ell - 1} \binom{\nu + k - \mu - 1}{\nu} \right\} \\
 &= (1 - \alpha)^k \sum_{\mu=0}^{\ell-1} (\ell - \mu)(\ell - \mu + 1) \alpha^\mu \binom{\mu + k - 1}{k - 1} \\
 &\quad + \alpha^{\ell+2} \sum_{\mu=0}^{k-1} (k - \mu)(k - \mu + 1) (1 - \alpha)^{\mu-2} \binom{\mu + \ell - 1}{\ell - 1} \\
 &= (1 - \alpha)^k \sum_{\mu=0}^{\ell+1} (\ell - \mu)(\ell - \mu + 1) \alpha^\mu \binom{\mu + k - 1}{k - 1} \\
 &\quad + \alpha^{\ell+2} \sum_{\mu=0}^{k+1} (k - \mu)(k - \mu + 1) (1 - \alpha)^{\mu-2} \binom{\mu + \ell - 1}{\ell - 1}. \tag{5}
 \end{aligned}$$

Also,

$$\begin{aligned}
 \ell(\ell + 1) &= \ell(\ell + 1) \cdot S_{\ell+2,k} \\
 &= (1 - \alpha)^k \sum_{\mu=0}^{\ell+1} \alpha^\mu \binom{\mu + k - 1}{k - 1} \ell(\ell + 1) + \alpha^{\ell+2} \sum_{\mu=0}^{k-1} (1 - \alpha)^\mu \\
 &\quad \cdot \binom{\mu + \ell - 1}{\ell + 1} \ell(\ell + 1) \\
 &= (1 - \alpha)^k \sum_{\mu=0}^{\ell+1} \alpha^\mu \binom{\mu + k - 1}{k - 1} \ell(\ell + 1) \\
 &\quad + \alpha^{\ell+2} \sum_{\mu=0}^{k+1} (1 - \alpha)^{\mu-2} \binom{\mu + \ell - 1}{\ell - 1} \mu(\mu - 1) \tag{6}
 \end{aligned}$$

$$\frac{\alpha^2 k(k + 1)}{(1 - \alpha)^2} = \frac{\alpha^2 k(k + 1)}{(1 - \alpha)^2} \cdot S_{\ell,k+2}$$

$$\begin{aligned}
&= \frac{\alpha^2}{(1-\alpha)^2} \left[(1-\alpha)^{k+2} \sum_{\mu=0}^{\ell-1} \alpha^\mu \binom{\mu+k+1}{k+1} k(k+1) \right] \\
&\quad + \frac{\alpha^2}{(1-\alpha)^2} \left[\alpha^\ell \sum_{\mu=0}^{k+1} (1-\alpha)^\mu \binom{\mu+\ell-1}{\ell-1} k(k+1) \right] \\
&= (1-\alpha)^k \sum_{\mu=0}^{\ell+1} \alpha^\mu \binom{\mu+k-1}{k-1} \mu(\mu-1) \\
&\quad + \alpha^{\ell+2} \sum_{\mu=0}^{k+1} (1-\alpha)^{\mu-2} \binom{\mu+\ell-1}{\ell-1} k(k+1) \tag{7}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\alpha \ell k}{(1-\alpha)} &= \frac{\alpha \ell k}{1-\alpha} \cdot S_{\ell+1, k+1} \\
&= \frac{\alpha}{1-\alpha} \left[(1-\alpha)^{k+1} \sum_{\mu=0}^{\ell} \alpha^\mu \binom{\mu+k}{k} \ell k + \alpha^{\ell+1} \sum_{\mu=0}^k (1-\alpha)^\mu \right. \\
&\quad \left. \cdot \binom{\mu+\ell}{\ell} \ell k \right] \\
&= (1-\alpha)^k \sum_{\mu=0}^{\ell} \alpha^{\mu+1} \binom{\mu+k}{k-1} \ell \mu + \alpha^{\ell+2} \sum_{\mu=0}^k (1-\alpha)^{\mu-1} \binom{\mu+\ell}{\ell-1} k \mu \\
&= (1-\alpha)^k \sum_{\mu=0}^{\ell+1} \alpha^\mu \binom{\mu+k-1}{k-1} \ell \mu \\
&\quad + \alpha^{\ell+2} \sum_{\mu=0}^{k+1} (1-\alpha)^{\mu-2} \binom{\mu+\ell-1}{\ell-1} k \mu. \tag{8}
\end{aligned}$$

Taking Equations (6) + (7) - 2.(8), we have

$$\ell(\ell+1) + \frac{k(k+1)\alpha^2}{(1-\alpha)^2} - \frac{2\ell k \alpha}{1-\alpha}$$

$$\begin{aligned}
&= (1 - \alpha)^k \sum_{\mu=0}^{\ell+1} \alpha^\mu \binom{\mu + k - 1}{k - 1} (\ell - \mu)(\ell - \mu + 1) \\
&\quad + \alpha^{\ell+2} \sum_{\mu=0}^{k+1} (1 - \alpha)^{\mu-2} \binom{\mu + \ell - 1}{\ell - 1} (k - \mu)(k - \mu + 1) \\
&= E \left[Q^{(1)}(t)(Q^{(1)}(t) + 1) + \frac{\alpha^2}{(1 - \alpha)^2} Q^{(2)}(t)(Q^{(2)}(t) \right. \\
&\quad \left. + 1) | N_1(t) = \ell, N_2(t) = k \right]. \quad \square
\end{aligned}$$

ACKNOWLEDGMENTS

The authors would like to thank Philip Heidelberger for a useful discussion, and the anonymous referees for providing comments that have improved the presentation of the article.

REFERENCES

- CHANDY, K. M. AND MISRA, J. 1981. Asynchronous distributed simulation via a sequence of parallel computations. *J. ACM* 24, 11 (Nov.), 198–205.
- CINLAR, E. 1975. *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ.
- FELDERMAN, R. E. AND KLEINROCK, L. 1991. Bounds and approximations for self-initiating distributed simulation without look-ahead. *ACM Trans. Model. Comput. Simul.* 1, 4 (Oct.), 386–406.
- FUJIMOTO, R. M. 1990. Parallel discrete event simulation. *Commun. ACM* 33, 10 (Oct.), 30–53.
- GAFNI, A. 1988. Rollback mechanisms for optimistic distributed simulation systems. In *Proceedings of SCS Multiconference on Distributed Simulation* (July), 61–67.
- GAUJAL, B., GREENBERG, A., AND NICOL, D. 1993. A sweep algorithm for massively parallel simulation of circuit-switched networks. *J. Parallel Distrib. Comput.* 18, 4 (Aug.), 484–500.
- GUPTA, M., KUMAR, A., AND SHOREY, R. 1996. Queueing models and stability of message flows in distributed simulators of open queueing networks. In *Proceedings of ACM/IEEE/SCS 10th Workshop on Parallel and Distributed Simulation (PADS'96)* (Philadelphia, PA, May 1996).
- HARRISON, J. M. 1973. Assembly-like queues. *J. Appl. Prob.* 10, 354–367.
- JEFFERSON, D. R. 1985. Virtual time. *ACM Trans. Program. Lang. Syst.* 7, 3 (July), 404–425.
- KAPLAN, M. 1979. A sufficient condition for nonergodicity of a markov chain. *IEEE Trans. Inf. Theor.* IT-25, 4, 470–471.
- KLEINROCK, L. 1975. *Queueing Systems, Volume I*. Wiley, New York.
- KLEINROCK, L. 1990. On distributed systems performance. *Comput. Netw. ISDN Syst.* 20, 209–215.
- LATOUCHE, G. 1981. Queues with paired customers. *J. Appl. Prob.* 18, 684–696.
- LUBACHEVSKY, B. D. 1989. Efficient distributed event-driven simulations of multiple-loop networks. *Commun. ACM* 32, 1 (Jan.), 111–123.
- MISRA, J. 1986. Distributed discrete event simulation. *ACM Comput. Surv.* 18, 1 (March), 39–65.

- MITRA, D. AND MITRANI, I. 1984. Analysis and optimum performance of two message-passing parallel processors synchronised by rollback. In *Proceedings of Performance '84*, Elsevier Science, North Holland, Amsterdam, 35–50.
- NICOL, D. M. 1988. High performance parallelized discrete event simulation of stochastic queueing networks. In *Proceedings of 1988 Winter Simulation Conference* (Dec.), 306–314.
- NICOL, D. M. 1993. The cost of conservative synchronization in parallel discrete event simulations. *J. ACM* 40, 2 (April), 304–333.
- NICOL, D. M., MICHAEL, C., AND INOUE, P. 1989. Efficient aggregation of multiple lps in distributed memory parallel simulations. In *Proceedings of the 1989 Winter Simulation Conference* (Washington DC, Dec.), 680–685.
- REIHER, P., FUJIMOTO, R., BELLENOT, S., AND JEFFERSON, D. 1990. Cancellation strategies in optimistic execution systems. In *Proceedings of the 1990 Conference on Parallel and Distributed Simulation* (Jan.), 112–121.
- SHANKER, M. S. AND PATUWO, B. E. 1993. The effect of synchronization requirements on the performance of distributed simulations. In *Proceedings of the 7th Workshop on Parallel and Distributed Simulation (PADS'93)*, 151–154.
- SHOREY, R. 1996. Modelling and analysis of event message flows in distributed discrete event simulators of queueing networks. Ph.D. Thesis, Dept. of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India.
- SOKOL, L. M., WEISSMAN, J. B., AND MUTCHLER, P. A. 1991. Mtw: An empirical performance study. In *Proceedings of the 1991 Winter Simulation Conference*, 557–563.
- SZPANKOWSKI, W. 1985. Some conditions for non-ergodicity of Markov chains. *J. Appl. Prob.* 22, 138–147.
- SZPANKOWSKI, W. 1990. Towards computable stability criteria for some multidimensional stochastic processes. In *Stochastic Analysis of Computer and Communication Systems*, Elsevier, Amsterdam, The Netherlands, 131–172.
- WAGNER, D. B. AND LAZOWSKA, E. D. 1989. Parallel simulation of queueing networks: Limitations and potentials. *Perf. Eval. Rev.* 17, 1 (May), 146–155.
- WOLFF, R. W. 1989. *Stochastic Modelling and the Theory of Queues*. Prentice-Hall, Englewood Cliffs, NJ.

Received March 1995; revised November 1996; accepted January 1997