Queueing Models and Stability of Message Flows in Distributed Simulators of Open Queueing Networks

Manish Gupta, Anurag Kumar, Rajeev Shorey
Electrical Communication Engineering
Indian Institute of Science, Bangalore
Bangalore, INDIA, 560 012
e-mail: manish, anurag, shorey@ece.iisc.ernet.in

Abstract

In this paper we study message flow processes in distributed simulators of open queueing networks. We develop and study queueing models for distributed simulators with maximum lookahead sequencing. We characterize the "external" arrival process, and the message feedback process in the simulator of a simple queueing network with feedback. We show that a certain "natural" modelling construct for the arrival process is exactly correct, whereas an "obvious" model for the feedback process is wrong; we then show how to develop the correct model. Our analysis throws light on the stability of distributed simulators of queueing networks with feedback. We show how the stability of such simulators depends on the parameters of the queueing network.

1 Introduction

In a *distributed discrete-event simulation* (DDES), each process can be viewed as receiving time-stamped messages over several sequential channels; i.e., on each channel the messages arrive in time-stamp order. The aggregate message stream arriving at the process is, however, not in time-stamp order. The purpose of the synchronization mechanism (optimistic [3] [2], conservative [7], or variations of these) is to produce a message stream that is in time-stamp order, for further processing by the event processor. Obviously, the complication in the sequencing procedure arises when a channel is empty, for then the sequencer does not know what to do with the messages in the non-empty channels. Suppose, however, the sequencer is told the time-stamp of the next message to arrive in all the empty channels. This is all the information the sequencer needs to sequence the messages that can be sequenced at this time (no additional information about future arrivals will help). Such information if available at each channel is called *maximum lookahead* [4], and is, in general, not practically obtainable. Maximum lookahead is, of course, an upper bound to any practical lookahead, and is hence expected to yield an upper bound on simulator performance. With this in mind, in this paper we study distributed simulators with maximum lookahead sequencing.

In this paper we are interested in message flow processes in distributed simulators of open queueing networks. Our objective is two fold. The simulation makes progress when event processors correctly process event messages; thus the rate of flow of correctly processed messages is indicative of the performance of the distributed simulator (e.g., speed-up). Further, the study of message flow processes yields an understanding of issues such as stability and boundedness of message queues. Note that it has been shown in [4] that in distributed simulation of feedforward networks the message queues are unstable. Our emphasis in this paper is on the second objective. The first objective is dealt with in a companion paper [5].

Our approach is via *queueing models of distributed simulators* with maximum lookahead. The "customers" in these queueing models are the event messages and the servers are the event processors. Obviously the throughput of correct messages in such a queueing model relates to the rate of progress of the simulation. Owing to the synchronization problem, these queueing models have features that do not appear in the usual queueing models in the literature. In this paper we develop and study such queueing models for distributed simulators of queueing networks with feedback. Maximum lookahead simulators are found to be the simplest to model. We show that a certain "natural" modelling construct for arrival processes in the simulator is, in fact, exactly correct, whereas an "obvious" model for the feedback

1087-4097/96 $5.00 © 1996 IEEE

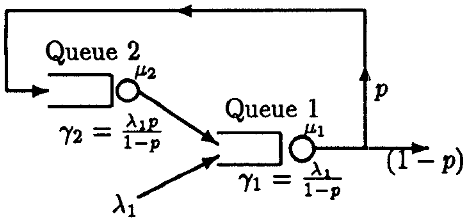
Queue 2
μ_2
$\gamma_2 = \frac{\lambda_1 p}{1-p}$
Queue 1
μ_1
p
$(1-p)$
$\gamma_1 = \frac{\lambda_1}{1-p}$
λ_1

Figure 1: A Queueing Network with Feedback

processes is wrong; we show how to develop the exact feedback model. Our analysis in this paper throws light on the stability of distributed simulators of queueing networks with feedback. We show how the performance of simulators of queueing networks with feedback depends on the parameters of the queueing network.

The paper is organized as follows. In Section 2, we consider a queueing model with two queues with feedback. We develop the queueing model of a simulator in which each queue is simulated by a process on separate processors. External arrivals into each queue are shown to result in arrivals of batch event messages into each process. In Section 3, we characterize these "external" batch arrival processes in the simulator of the queueing network. Section 4 is devoted to characterizing the message feedback process. We show the incorrectness of the Bernoulli feedback model, and demonstrate the exact feedback model for a limiting case of the original 2 queue model. Finally, in Section 5 we develop an exact model for the feedback process in a more general case. Analysis of this model suggests how the stability of the message queues depends on the parameters of the original queueing network. Section 6 has the conclusions. Appendix contains the details of some mathematical derivations.

2 Development of the Queueing Model of the Simulator

In this section, we consider a queueing network with two queues and feedback. We begin to develop the model of the distributed simulator of the queueing network. Some nonstandard modelling constructs arise, which we proceed to develop in subsequent sections.

As usual, we call the simulation processes running on the simulator processors as logical processes (LPs) and the queueing (sub)networks they simulate as physical processes (PPs). If the queueing network is mapped onto m logical processes (each on a separate processor) these are denoted $LP_1, LP_2, \ldots, LP_m$.

2.1 A Queueing Network with Feedback

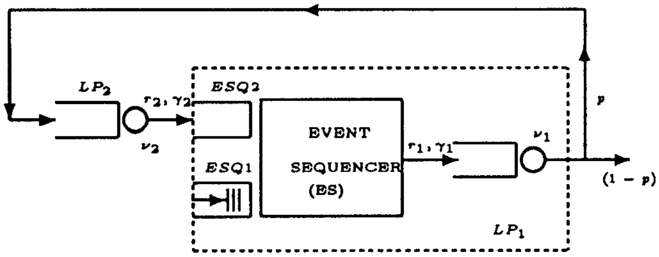

Figure 2: Simulator of the Queueing Network with Feedback

Consider a network with two queues and a feedback shown in Figure 1. The distributed simulator for this network is shown in Figure 2. The simulation of each queue is assigned to a separate LP. Denote by μ_i, $i = 1, 2$, the service (exponential) rate of queue i in the queueing network with feedback. λ_1 is the external arrival (Poisson) rate of customers to Queue 1 in the queueing network. $\nu_i, i \in \{1, 2\}$ is the service (exponential) rate of the event processor queue in LP_i. The event sequencer (ES) has two message queues ESQ1 and ESQ2. Queue ESQ2 is fed by LP_2. Note that since Queue 1 in the PP has external arrivals generated within LP_1, the corresponding event sequencer queue (ESQ1) will always have a backlog of messages.

Denote by γ_i the throughput of queue i ($i = 1, 2$) in the queueing network, and by r_i the throughput of LP_i ($i = 1, 2$) in the simulator. We assume that the queues of the queueing network (PP) are stable. Thus, $\gamma_2 = \frac{\lambda_1 p}{1-p} < \mu_2$, $\gamma_1 = \frac{\lambda_1}{1-p} < \mu_1$.

In this problem, owing to the feedback loop, the conservative event sequencer will deadlock; a deadlock occuring each time queues LP_2, ESQ2, and the event processor of LP_1 in the simulator are empty. With a maximum lookahead sequencer, however, it is clear that in this situation the lookahead in queue ESQ2 is ahead with probability (w.p.) 1, of the time-stamp of the first message in queue ESQ1, and hence this message can be allowed to enter and the simulator progresses. This can be argued as follows. Suppose that the first message in queue ESQ1 is fed back, then the maximum lookahead in queue ESQ2 will be equal to the sum of three terms: the time-stamp of the first message in queue ESQ1, a sample of an exponential random variable with rate μ_1, and, a sample of an exponential random variable with rate μ_2. Note that an exponentially distributed random variable (corresponding to the service time) is zero w.p. 0, thus the lookahead in queue ESQ2 is ahead (w.p. 1) of the time-stamp of the first message in queue ESQ1. If the first message in queue ESQ1 is not fed back, essentially the same arguments hold as above along with the fact that the time-stamps of messages in queue ESQ1

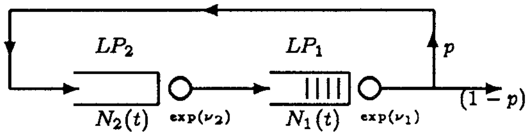

Figure 3: Equivalent Queueing Model of the Simulator in Figure 2

are monotonically increasing. *Thus the simulator with maximum lookahead sequencing does not deadlock.* Note also that *with maximum lookahead sequencing all event messages are correctly processed exactly once.*

We study the distributed simulator of Figure 2 with the maximum lookahead algorithm. Whenever a message leaves queue ESQ2 of the simulator, the timestamp of the next message to arrive at queue ESQ2 is provided to the sequencer (i.e., maximum lookahead operates), and all messages from queue ESQ1 with timestamp less than this lookahead time are moved to the event processor. Therefore, after each departure from queue ESQ2, a batch of messages arrive at the event processor in LP_1. This batch consists of the message that leaves queue ESQ2 plus some arrival messages from queue ESQ1. It is easy to show [6] that the mean batch size is $1 + \frac{\lambda_1}{\gamma_2} = \frac{1}{p}$. In the queueing model of Figure 1, if we have no feedback, the resulting network is a feedforward network. It can be proved [4] that in the distributed simulator of a feedforward network, if the timestamp processes are Poisson, then the batches departing from the event sequencer are geometrically distributed. No such claim can be immediately made about the batch size distribution in simulators of networks with feedback (as in Figure 1). In Section 3, we will investigate the distribution of the batch size in this model.

2.2 Queueing Model of the Distributed Simulator

A queueing model of the distributed simulator is shown in Figure 3. In this model no event sequencer is shown; instead, a departure from LP_2 brings a batch of messages to the event processor of LP_1. The mean batch size is given by $\frac{1}{p}$. As in the queueing model in the simulator, a fraction $(1-p)$ of the events processed by LP_1 depart from the system, and the rest are fed back. Note that this is not to claim that the *feedback process* is Bernoulli, i.e., independent and identically distributed (i.i.d.) feedbacks with probability (w.p.) p.

Our aim is to analyse the event message processes in this queueing model. We would like to obtain the simulator throughput, by which we mean the rate at which processed events *leave* the simulator after event processing at LP_1. We may be interested in the system sojourn time of customers entering Queue 1 in the queueing network model. Each departure from the simulator will yield a sample of sojourn time, and hence simulator throughput is a meaningful quantity.

Before steady-state throughput can be determined, however, stability of the message queues needs to be established. Analysis of the queueing model in Figure 3 can only proceed after we characterize the batch arrival process at LP_1 and the feedback process from LP_1 to LP_2. This we now proceed to do in Sections 3 and 4.

3 Characterization of the "External" Arrival Process in the Simulator

In the model in Figure 1, denote by ρ_i, $i = 1, 2$, the server utilization factor of queue i. Then, $\rho_1 = \frac{\lambda_1}{\mu_1(1-p)}$ and $\rho_2 = \frac{\lambda_1 p}{\mu_2(1-p)}$. For notational convenience $(1-p)$ is written as $\bar{p}$ in the analysis.

Now observe that the external arrival batches that arrive following a departure from LP_2 into LP_1 (see Figure 3) are just the batches of external arrivals between successive departures from Queue 2 to Queue 1 in Figure 1.

We prove that the batches of customers arriving between departures from Queue 2 to Queue 1 in Figure 1 are i.i.d. and geometrically distributed with mean equal to $\frac{1}{p}$.

3.1 Analysis for the Two Queue Model

Consider a departure from Queue 2 to Queue 1 (Figure 1) in the stationary regime. Let the number of customers in queue i, $i = 1, 2$, just *after* this departure be denoted by X_i. By the Arrival Theorem [8] we know that

$$P(X_1 = k_1, X_2 = k_2) = \rho_1^{k_1 - 1}(1 - \rho_1)\, \rho_2^{k_2}(1 - \rho_2),$$

$$k_1 \geq 1, k_2 \geq 0$$

Let

Y_i = the number of customers in queue i, $i = 1, 2$, just *after* the next departure from Queue 2 to Queue 1

A = the number of external arrivals to Queue 1 *until* the next departure from Queue 2 to Queue 1

We are interested in calculating the joint generating function

$$\tilde{b}(w_1, w_2, z) \quad := \quad E\left(w_1^{Y_1} w_2^{Y_2} z^A\right)$$

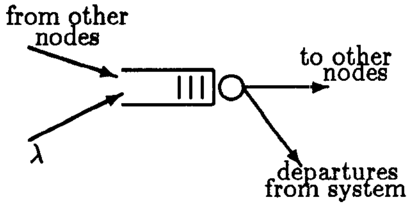

Figure 4: A Node in a Jackson Network

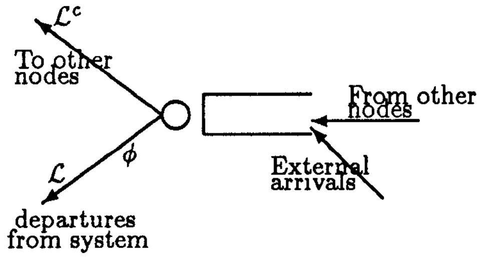

Figure 5: Reversal of the Node in Figure 4

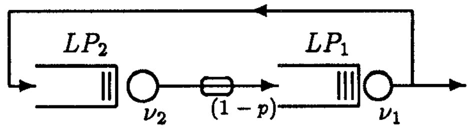

Figure 6: Queueing Model for the Simulator in Figure 2 showing Batch Arrivals into LP_1.

$$= \sum_{k_1=1}^{\infty}\sum_{k_2=0}^{\infty} P(X_1 = k_1, X_2 = k_2)\cdot$$

$$\cdot E\left(w_1^{Y_1} w_2^{Y_2} z^A | X_1 = k_1, X_2 = k_2\right)$$

It is easily shown that [6]

$$\tilde{b}(w_1, w_2, z) = \frac{p}{(1-\bar{p}z)}\tilde{g}(w_1, w_2) \qquad (1)$$

where, $\tilde{g}(w_1, w_2) := \sum_{k_1'=1}^{\infty}\sum_{k_2'=0}^{\infty} \rho_1^{k_1'-1}(1-\rho_1)\rho_2^{k_2'}(1-\rho_2)w_1^{k_1'}w_2^{k_2'}$.

Observe from Equation 1 that $\tilde{b}(w_1, w_2, z)$ factors into two generating functions, one in z and the other in (w_1, w_2). It follows that after a stationary departure (Y_1, Y_2) and A are independent. Further $P(A = k) = (1-p)^k p$ for $k \geq 0$, and $P(Y_1 = \ell_1, Y_2 = \ell_2) = \rho_1^{\ell_1 - 1}(1-\rho_1)\ \rho_2^{\ell_2}(1-\rho_2)$ for $\ell_1 \geq 1, \ell_2 \geq 0$, Since the number of external arrivals until the next departure after the one which left behind (Y_1, Y_2) depends only on (Y_1, Y_2), it follows that the numbers of external arrivals between successive departures from Queue 2 to Queue 1 form an i.i.d. sequence with a geometric distribution.

It follows that in the queueing model of the maximum lookahead simulator, shown in Figure 3, *each event arriving from LP_2 to LP_1 is immediately followed by a geometrically distributed number of external arrival events. The successive such batches are i.i.d..*

3.2 A Generalization Using Time Reversal Arguments

In this section, we extend the results of the last section. For a Jackson Network [8], we prove that the sequence of number of external arrivals at a node between successive arrivals from other nodes is a geometrically distributed i.i.d. process.

Consider a typical node of a stationary Jackson Network. This is shown in Figure 4. λ is the external arrival rate of customers to the node. Customers arrive at this node from various other nodes also. Now consider the reversal of the stationary Jackson Network. In the reversed network, a transition from state j to state i corresponds to the transition from state i to state j in the original network. The reversal of the node in the stationary Jackson Network in Figure 4 is shown in Figure 5. In the reversed network, the customers that leave the system correspond to the external arrivals in the original network. Therefore, we need to find the distribution of the customers that leave along link $\mathcal{L}$ in the reversed node.

We denote by ϕ, the probability of a customer leaving the node in the reversed network. Then, between two departures that join any of the other nodes (along $\mathcal{L}^c$ in Figure 5), the number of departures that leave on link $\mathcal{L}$ is geometrically distributed, and is equal to n with probability $\phi^n(1-\phi)$. Further the sequence of the number of such departures along $\mathcal{L}$, between successive departures along $\mathcal{L}^c$ is i.i.d. But the reversal of this process is precisely the number of external arrivals between successive arrivals from other nodes.

Thus, we have shown that at any node in a stationary Jackson Network, the number of external arrivals between two arrivals from other nodes is geometrically distributed, and the sequence of such numbers is i.i.d.

It follows that in the model of a distributed simulator of a Jackson Network, in the stationary regime, *the successive external arrival batches triggered by a departure from one queue to another are i.i.d and geometrically distributed.* Note that an *LP* simulates a single *Queue*.

4 Characterization of Message Feedback Process

From the analysis in Section 3, we conclude that the model for the maximum lookahead simulator in Figure 2

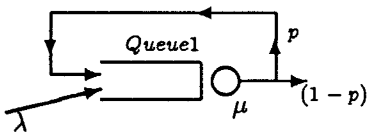

Figure 7: Limit of the Queueing Network in Figure 1 when $\mu_2 \rightarrow \infty$, and $\mu_1 = \mu$.

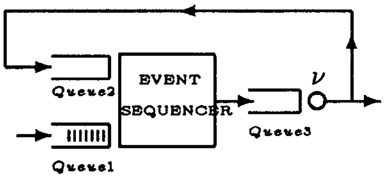

Figure 8: Limit of the Simulator in Figure 2 with $\nu_2 \rightarrow \infty$, and $\nu_1 = \nu$.

will have the form shown in Figure 6 where the $(1-p)$ next to the arrow entering LP_1 denotes that each departure message that moves from LP_2 to LP_1 actually brings in a batch of $k \geq 1$ customers with probability $(1-p)^{k-1}p$, and the successive such batches are i.i.d.

In Figure 7 is shown the limit of the queueing model of Figure 1 as $\mu_2 \rightarrow \infty$, and in Figure 8 is shown the limit of the simulator of Figure 2 as $\nu_2 \rightarrow \infty$. Note that we have replaced the symbols μ_1 by μ and ν_1 by ν. The queueing model for Figure 8 with maximum lookahead will have the form shown in Figure 9.

Due to the difficulty in characterizing the message feedback process from queue 1 to queue 2 in the model shown in Figure 6, we initially study the simpler model in Figure 9.

Observe that each message departure from the event processor corresponds to the departure of a customer from Queue 1 in the PP of Figure 7. If the customer is fed back in the original queueing network then the corresponding message is fed back in the simulator, otherwise the message corresponds to a customer leaving the system. Thus a fraction p of the message departures from the event processor are fed back (and subsequently carry with them i.i.d. geometrically distributed batches of external arrival messages into the event processor).

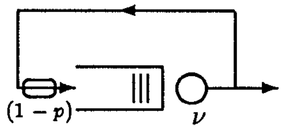

Figure 9: Partially Developed Model for the Simulator in Figure 8

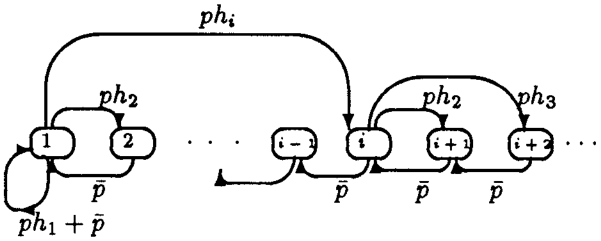

Figure 10: Transition Probability Diagram of the Message Queue Length Process in the Event Processor in Figure 8, using the Bernoulli Feedback Model

It remains to characterize the process by which messages are selected for feedback. We first show that the Bernoulli feedback model is wrong. We do this by assuming the Bernoulli feedback model and then analysing the resulting Markov chain of messages queued in the event processor; this Markov chain is found to be not positive recurrent. This is in disagreement with simulations, however, since we observe that, in simulations, message queue lengths remain stable. Also the Bernoulli feedback model predicts instability in cases where stability of the model is obvious(we shall demonstrate the source of the problem in section 4.2).

4.1 Incorrectness of the Bernoulli Feedback Model

With the Bernoulli feedback assumption, the i.i.d. geometric batch feedback model (Figure 9), and exponential event processing times, it is clear that $X(t)$, the number of messages in the event processor at time t, is a Continuous Time Markov Chain whose transitions occur at the epochs of a Poisson process of rate ν. Hence we can just as well study $\{X_n\}$ the queue length process embedded at these Poisson epochs. Further, nothing is lost in assuming $\nu = 1$. Letting, for $k \geq 1$, $h_k = (1-p)^{k-1}p$, the transition probability diagram of $\{X_k\}$ is depicted in Figure 10. For notational convenience $(1-p)$ is written as $\bar{p}$ in this figure and subsequent analysis.

Observe that when maximum lookahead synchronization is used, the simulation will never deadlock. Hence when just 1 message is present in the event processor and this message leaves, owing to the available maximum lookahead, the next arrival message can be generated and inserted into the event processor, thus avoiding deadlock. The Markov chain transition diagram in Figure 10 depicts this self transition from 1 to 1.

Thus, we obtain the transition probability matrix P, which has the well known structure of the embedded

Markov chain in an $M/G/1$ queue:

$$P = \begin{bmatrix} 1-p(1-h_1) & ph_2 & ph_3 & \cdots & ph_i & \cdots \\ \bar{p} & ph_1 & ph_2 & \cdots & ph_{i-1} & \cdots \\ 0 & \bar{p} & ph_1 & \cdots & ph_{i-2} & \cdots \end{bmatrix}$$

This is clearly an irreducible and aperiodic Markov chain. To study its positive recurrence, we attempt to find a solution for $\pi = \pi P$, $\sum_{i \in S} \pi_i = 1$. A little algebra shows that any solution to $\pi = \pi P$ will be of the form $\pi_i = p\pi_1$ for $i \geq 2$. Thus, $\sum_{i \in S} \pi_i = 1$ cannot be satisfied and the Markov chain is not positive recurrent. Therefore the message queue is unstable.

In addition, it can be shown by standard analysis that all the states of the Markov chain are recurrent. This implies that the Markov chain is null recurrent.

This analysis shows that the Bernoulli feedback model yields unstable message queues in the simulator of Figure 8. However, no instability is seen from a simulation study of the simulator.

We now refine the queueing model of the simulator in order to capture the stability of the event processor queue. This is done in the next subsection.

4.2 Refinement of the Queueing Model of the Simulator

We first demonstrate the source of the problem by studying the simulator in Figure 8 with infinite service rate in the PP of Figure 7, i.e., $\mu = \infty$ but finite service rate ν in the simulation. Consider now a feedback epoch τ in the PP. In the simulator (Figure 8) the corresponding feedback message is immediately followed by a geometrically distributed batch of messages corresponding to external arrivals *until the next feedback*. In the PP this (next) feedback must come from among the queued messages at the epoch τ^+ or the external arrival messages until the next feedback. Since the service time in the PP is infinite, however, each arrival always sees an empty PP queue, and hence the next feedback after τ must be due to the *last* arrival that arrives between the two feedbacks. Thus, in the simulator model, the feedback probability of the *last* arrival in the batch is 1 and for others it is zero, and not p for every message as asserted by the Bernoulli feedback model. Observe that the mean number of customers that will be fed back from each arriving batch (including the feedback that preceded the batch) is 1 in both models.

Then for $\mu = \infty$ we get the following CTMC model. When there are $n \geq 1$ messages in the event processor queue, messages depart after i.i.d. exponential(ν) service until 1 message is left. This message *must* be fed back after its service. There is thus always at least one message in the event processor queue. Again embedding at service completion epochs (which form a Poisson process of rate ν) we get a Markov chain with the transition probability diagram shown in Figure 11; h_k, $k \geq 1$ is as defined in Section 4.1. In this Markov chain the mean time to return to state 1 is $\sum_{i=1}^{\infty} ih_i < \infty$. We conclude that the simulator is stable. Note that we have the exact model for the simulator when $\mu = \infty$ and event processor service times are i.i.d. exponential(ν).

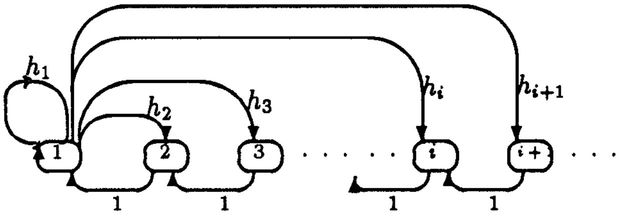

Figure 11: Transition Probability Diagram of the Message Queue Length Process in the Event Processor in Figure 8, with $\mu = \infty$, using the Correct Feedback Model

We conclude that the feedback processes in the distributed simulator model need to be modelled carefully, otherwise erroneous conclusions arise. Further for the simple queueing model and its distributed simulator studied above, ***the stability of the simulator message queues is independent of ν***, the processor speed. Also observe that, whereas in a serial simulation of the model of Figure 7 there will be at most 2 pending events, ***the number of unprocessed messages in the event processor in the distributed simulator is unbounded***, as is clear from the Markov chain model (Figure 11).

5 Queueing Network with Two Queues and Feedback: Revisited

A little thought shows that in the simple model of Figure 7 and the model of its maximum lookahead simulator in Figure 8, there is a close correspondence between customer queues in the PP and message queues in the LP. Hence stability of the PP implying stability of the LP is hardly surprising. A similar result is not at all obvious, in general, for the PP in Figure 1 and its simulator model of Figure 6. If ν_2 is, for example, very large (say ∞) then the message queue in LP_1 of the simulator model in Figure 6 can grow to very large values, and will have little correspondence to the queue lengths in Queue 1 of the PP. We therefore turn now to the more difficult problem of analysing the feedback process in

Figure 6 and a study of the stability of the consequent queueing model.

We consider the situation in which μ_1 and μ_2 in Figure 1 are finite, but in Figure 6 (the simulator model) $\nu_2 = \infty$, while $\nu_1 < \infty$. In this situation each feedback from LP_1 (Figure 6) instantaneously enters LP_1 again, bringing with it a batch of arrivals. Thus the LP model is as in Figure 9, except that feedback events have their time-stamps incremented owing to the nonzero service time in Queue 2 of the PP. We change our point of view slightly by associating with each departure from LP_2 to LP_1 those arrivals that immediately precede this departure (i.e., this departure was the lookahead event, based on which the batch of arrivals was "taken in"). Observe that, with maximum lookahead, LP_1 will never become empty, so we consider an arrival from LP_2 and its associated arrival batch as *one customer* with exponential service time distribution with rate $\nu_1 p$; this is an exactly correct model which follows from the results in Section 3.

Now let $\{X_n, n \geq 0\}$ be the queue length process in the event processor in LP_1, embedded at batch service completion epochs, i.e., at Poisson epochs with rate $\nu_1 p$. Each batch of events, upon being serviced, yields events that get fed back into LP_1. The feedback events correspond to those customers who come in the batch in the PP but get fed back to Queue 2 after service at Queue 1. Let this sequence of feedback batches in the PP be $\{F_k\}$; for a stationary and ergodic PP, the sequence $\{F_k\}$ is also stationary and ergodic. Then clearly (recall $\nu_2 = \infty$), for $X_0 \geq 1$, $X_{n+1} = \max(X_n - 1 + F_n, 1)$. It is well known ([1], [8]) that $\{X_n\}$ converges in distribution to a proper random variable if, with probability 1, $lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} F_n = \ E(F) < 1$. In Appendix, it is established that $E(F) < 1$ if $\rho_1 < 1$ and $\rho_2 < 1$ in the PP.

Now let $\tau_1, \tau_2, ...$ be the epochs of service completions at LP_1 with $\nu_2 < \infty$. Observe that for any $\nu_2 \in (0, \infty]$, the n^{th} service completion at LP_1 corresponds to the same event in the PP. If $(X_{\tau_n}^{(2)}, X_{\tau_n}^{(1)})$ is the state of the system with $\nu_2 < \infty$ at the epochs τ_n then it can be shown by induction on n that $X_{\tau_n}^{(1)} + X_{\tau_n}^{(2)} = X_n$. Intuitively, the only way the customers can leave the system (with or without $\nu_2 < \infty$) is at a service completion at LP_1. Also the only way the customers enter LP_2 is when there is a service completion at LP_1, i.e., at service completion at LP_2 (with $\nu_2 < \infty$) the total number in the system is invariant.

For other transitions between τ_n and τ_{n+1} (these correspond to jumps from LP_2 to LP_1) we have the number in the system equal to X_n. Thus, letting $n = 0, 1, ...$ index all the jumps in the uniformized chain, if $(X_n^{(2)}, X_n^{(1)})$ is the state of the system ($\nu_2 < \infty$) then $\{X_n^{(2)} + X_n^{(1)}\}$ also converges in distribution to the proper random variable.

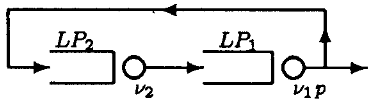

Figure 12: Queueing Model for the Simulator in Figure 2, obtained from Figure 6, by Dilating the Service Time at LP_1.

Thus we find that the simulator is stable if the original queueing model is stable. The processing rates in the simulator do not play a role in the stability of the simulator. Of course, the rate of simulation progress is more if the simulator processes are faster.

6 Conclusions

In this paper we have developed models for message flows in distributed simulators of queueing networks with maximum lookahead sequencing, with a view towards stability analysis of event processor queues.

We characterized the "external" arrival processes in the simulator of an open Jackson Network, and proved that the batches of customers arriving between departures from one queue to another are i.i.d. and geometrically distributed.

We demonstrated the difficulty in characterizing the message feedback processes in the simulator model. We showed that the Bernoulli feedback model is wrong, and developed the feedback process model for a two queue example.

Analysis of the resulting models showed that *the stability of message queues in the simulator depends on the stability of the queueing model being simulated*; thus *faster processors in the distributed simulators do not yield a stable simulation of an unstable queueing model.* This is quite in contrast with serial simulation; observe that serial simulation of an unstable M/M/1 queue requires an event list with at most 2 pending events.

This paper is just an initial effort on what is obviously a hard problem. Further work should help to show how the throughput of the simulator depends on parameters of the queueing model and the processing speeds in the simulator.

Appendix: Proof that $E(F) < 1$

Let Γ denote a customer (recall that each such customer models a batch of events) that has just finished service at LP_1 in the simulator model of Figure 12 (obtained from Figure 6 by dilating the service time at LP_1). To determine $E(F)$, i.e., the expected number of customers that are fed back when Γ finishes service, we look at the original queueing network (in Figure 1). Recall that Γ corresponds to a customer C that jumps from Queue 2 to Queue 1 in the PP (the queueing network). When the customer C is in the midst of jumping from Queue 2 to Queue 1 in the PP, it sees some number n_2 in Queue 2 and some number n_1 in Queue 1 in the PP. Observe that all the external arrivals that are implicitly associated with Γ in the simulator model have already entered Queue 1 of the PP before the epoch at which C jumps from Queue 2 to Queue 1 Let the number of these external arrivals associated with C be given by A. If $n_1 \geq A$, it means that all the A arrivals are candidates for feedback to Queue 2 in the PP. In this situation, the expected number of feedbacks that are put in LP_2, upon the service completion of Γ in LP_1, is Ap. (Note that since $\nu_2 = \infty$, all the feedbacks to LP_2 arrive instantaneously to LP_1). Now consider the case when $n_1 < A$; without further analysis, we can only infer that the expected number of arrivals in A that are fed back to Queue 2 is at least $n_1 p$. We show below that the expected number of feedbacks to Queue 2 is $n_1 p + \min(A - n_1, n_2)p_1$, where $p_1 := \left(\frac{\mu_2}{\mu_2+\lambda}\right)$. Note that C can be fed back with probability p.

When $n_1 < A$ then consider the queueing system in Figure 1 at the instant when C makes a transition from Queue 2 to Queue 1 in "forward" time. Now we look at "reversed" time. In the reversal of the system, C jumps from Queue 1 to Queue 2 and A is the number of external departures from Queue 1 that follow C but precede the next jump from Queue 1 to Queue 2. Further F is the number of these departures that resulted from customers that came to Queue 1 from Queue 2. When $n_1 < A$ then we know that $E(F) \geq n_1 p$. Given that Queue 2 is nonempty, the probability that an arrival to Queue 1 came from Queue 2 is $p_1 = \left(\frac{\mu_2}{\mu_2+\lambda}\right)$. Obviously, no more than $\min(A - n_1, n_2)$ can come from Queue 2 in the reversed network. It can be easily shown that the expected additional number that came from Queue 2 is $(\min(A - n_1, n_2))p_1$.

Thus we can write $E(F)$ as follows, ($\bar{p} := 1 - p$)

$$\begin{aligned} E(F) \quad = \quad & \sum_{n_1, n_2} \pi(n_1, n_2) \left\{ \sum_{k=0}^{n_1} \bar{p}^k pkp \right. \\ & + \sum_{k=n_1+1}^{n_1+n_2} (n_1 p + (k - n_1)p_1)\bar{p}^k p \\ & \left. + \sum_{k=n_1+n_2+1}^{\infty} (n_1 p + n_2 p_1)\bar{p}^k p \right\} + p \end{aligned}$$

where, for $\rho_1 < 1$ and $\rho_2 < 1$, by Arrival theorem [8], $\pi(n_1, n_2) = (1 - \rho_1)\rho_1^{n_1}(1 - \rho_2)\rho_2^{n_2}$, and $p_1 := \left(\frac{\mu_2}{\mu_2+\lambda}\right)$.

The explanation of the above expression is the following. We first condition on what C sees in Queue 1 and Queue 2 in the PP when it is making a jump from Queue 2 to Queue 1. Then we condition on the number A in the arrival batch that precedes it. The first term in the curly brackets in the expression above corresponds to the case when $n_1 \geq A$; the second term corresponds to the case when $n_1 < A$ but $n_2 \geq (A - n_1)$; and the third term corresponds to the case when $n_1 < A$, $n_2 < A - n_1$. The last term (p) in the expression for $E(F)$ corresponds to the fact that the customer C corresponding to the batch Γ that has just finished service at LP_1 can also be fed back with probability p.

Finally, note that $E(F) - 1$ simplifies to

$$\frac{(1 - \rho_1)\bar{p}}{(1 - \bar{p}\rho_2)(1 - \bar{p}\rho_1)} \{-1 + \rho_2 (1 - p_1\bar{p} - p\bar{p}_1)\}$$

which is < 0 when $\rho_1 < 1$ and $\rho_2 < 1$.

References

[1] Borovkov, A. A., *Stochastic Processes in Queueing Theory*, Springer-Verlag, 1976.

[2] Fujimoto, R.M., "Parallel discrete event simulation", *Commun. ACM 33*, 10 (October 1990), 30-53.

[3] Jefferson, D.R., "Virtual time", *ACM Trans. Prog. Lang. and Syst.*, 7, 3 (July 1985), 404-425.

[4] Kumar, A. and Shorey, R., "Stability of Event Synchronisation in Distributed Discrete Event Simulation", In *Proceedings of 8th Workshop on Parallel and Distributed Simulation (PADS '94)*, July 6-8, 1994, Edinburgh, Scotland, U.K.

[5] Shorey, R., Kumar, A., Manish, G., *"Throughput Bounds and Optimal Mappings for Distributed Simulation of Queueing Networks"*. Submitted.

[6] Shorey, R., *"Modelling and Analysis of Event Message Flows in Distributed Discrete Event Simulators of Queueing Networks"*, PhD Thesis, Submitted Jan. 1996, Deptt. of ECE, Indian Institute of Science, Bangalore, India.

[7] Misra, J., "Distributed discrete event simulation", *ACM Comput. Surv. 18*, 1 (March 1986), 39-65.

[8] Walrand, J., *An Introduction to Queueing Networks*, Prentice Hall, 1988.

[9] Wolff, R. W., *Stochastic Modelling and the Theory of Queues*, Prentice Hall, 1989.