Comput. Opns Res. Vol. 15, No. 4, pp. 357–367, 1988
Printed in Great Britain.

0305-0548/88 $3.00 + 0.00

THE TRANSIENT BEHAVIOR OF THE $M/E_k/2$ QUEUE AND STEADY-STATE SIMULATION*

JOSEPH R. MURRAY[1,†] and W. DAVID KELTON[2,‡]

[1]Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109 and [2]Department of Management Sciences, The University of Minnesota, 271 19th Avenue South, Minneapolis, MN 55455, U.S.A.

(*Received April* 1987; *revised November* 1987)

Scope and Purpose—The purpose of this paper is to study the time dependent behavior of some performance measures of a particular queueing system. The system consists of a single queue feeding two servers who work in parallel. Customers leave the system after having been served by one of the servers. The time dependent behavior is of inherent interest in Queueing Theory. The paper also discusses the implications of these results for Simulation Methodology.

Abstract—The probabilistic structure for the transient $M/E_k/2$ queue is derived in discrete time, where E_k denotes a k-Erlang distribution. This queue has a two-dimensional state-space. Expressions for the expected delay in queue are formulated in terms of transition probabilities. Results are numerically evaluated for a few cases. The convergence behavior is similar to that seen in previous work on queues with one-dimensional state spaces. The implications for initialization of steady-state simulations are discussed.

1. INTRODUCTION

A review of the literature in queueing theory reveals an abundance of results for steady-state conditions and relatively few results for the transient phase of a queueing system. One reason for this is the complexity and intractability of the mathematics involved in solving the transient problem, and it is not uncommon to see results stated in terms of transforms which are very difficult, if not impossible, to invert. Results, when left as transforms, seem somewhat less satisfactory in terms of ease of interpretation. Analytical solutions for the transient characteristics of queueing models are useful for studying the finite-time properties of systems that are accurately represented by such models. Even if exact analytical transient results are not known, it would be useful to know, in some fashion, the rate and manner (e.g. monotonic or oscillatory) of convergence of the system to steady-state.

Analytical transient results are also valuable in the evaluation of alternative start-up strategies for simulations aimed at estimating steady-state parameters. Kelton and Law [1] and Kelton [2] present transient results for $M/M/s$, $M/E_k/1$, and $E_k/M/1$ queues and use them as benchmarks to evaluate alternative initialization methods for simulation of similar systems. Enlarging the range of benchmark models to include systems with multivariate state spaces and multiple servers with non-exponential service times served as the main motivation for this paper.

Another simulation-related area where transient results can be applied is the external control variates technique for variance reduction. When examining the transient behavior of a complex, analytically intractable system in a terminating simulation, a simpler system with known transient behavior is simulated alongside the system of interest. The results from the two systems, assuming the use of common random numbers, would be expected to be correlated, leading to a variance reduction. The larger the class of systems for which transient results are known analytically, the greater the similarity possible, leading to stronger variance reductions.

Known transient results can be classified according to whether or not the time measure is continuous (real time) or discrete (indexing by customer number). Continuous time results for

*This research was supported in part by a contract from Electronic Data Systems Decision Technologies Division to the University of Michigan.

†Joseph R. Murray is a Ph.D. candidate in the Industrial and Operations Engineering Department at the University of Michigan. His research interests include Simulation Methodology and Analysis and Manufacturing Systems Modeling and Analysis.

‡David Kelton is Associate Professor in the Department of Management Sciences in the Carlson School of Management at the University of Minnesota, in Minneapolis. His research interests are in Simulation, Applied Statistics and Quality Control.

various queues can be found in Saaty [3], Kleinrock [4], Odoni and Roth [5], Grassman [6], Pegden and Rosenshine [7], etc. Continuous time results describe the behavior of system performance measures, e.g. the number of customers in the system, at every point in time. Discrete time results, on the other hand, focus on the state of the system at certain transition points, e.g. at the point of arrival of the nth customer, or at the point of departure of the nth customer.

The treatment of queues in discrete time is especially relevant from the standpoint of simulation. Standard measures of steady-state performance of general $GI/G/s$ queueing systems include the expected delay in queue (excluding service time), denoted by d, the expected wait in system (delay in queue plus service time), w, the expected number of customers in the queue, Q, and the number of customers in the system, L. Estimates for all of these quantities can be made directly during a simulation, but Carson and Law [8] have shown that it is preferable (in terms of achieving a reduction in variance) to estimate w, Q, and L indirectly from a direct estimate of d using the conservation equations: $w = d + E(S)$, $Q = \lambda d$ and $L = \lambda[d + E(S)]$ where $E(S)$ is the expected service time and λ is the arrival rate. For this reason, most simulation application and methodological research has focused on the delay-in-queue process, which clearly is in discrete time. Hence, analytical results for queueing systems in discrete time can be more easily related to simulation methodology. Morisaku [9], Kelton and Law [1], Kelton [2], Moore [10], Heathcote and Wiener [11], Stanford *et al.* [12] and Bhat and Sahin [13] present discrete-time results for various queueing systems.

In this paper, we present discrete-time transient results for an $M/E_k/2$ queue, where E_k denotes the k-Erlang distribution. The state variable for this system is expressed as a tuple. The method of analysis used here admits arbitrary (deterministic) initial states for the queue; this allows a numerical evaluation of the effect of alternative initial states on the nature of convergence to steady-state, a general question of interest in simulation aimed at estimating steady-state parameters.

In Section 2 we derive analytical results for the $M/E_k/2$ system. In Section 3 we present numerical results and discuss their implications for initialization of simulations. Section 4 contains some conclusions, and the Appendix contains proofs of the results presented in Section 2.

2. THE $M/E_k/2$ SYSTEM

Customer arrivals to the system are assumed to be Poisson with rate λ. A single queue feeds two servers who work in parallel. The service time distribution at each server is identical and is k-Erlang with mean $1/\mu$, independent of the arrival process. The analysis assumes that an arriving customer uses server 1 if server 1 is idle, server 2 if server 2 is idle and server 1 is not, and waits in queue for the first available server otherwise.

Each complete service period is modeled as k consecutive independent exponential stages, each at rate $k\mu$. The traffic intensity is $\rho = \lambda/(2\mu)$. Let T_n, $n = 1, 2, 3 \ldots$ be a random variable that represents the time of arrival of the nth customer to the system. Let A_t be the number of service stages yet to be completed at server 1 at each point of continuous time t, $t \geqslant 0$, and let B_t be the total number of service stages present in the system at time t. (Server 1 is idle at time t if and only if $A_t = 0$.) The pair $\mathbf{X}(t) = (A_t, B_t)$ is sufficient to describe the state of the system at time t, since other quantities, such as the number of customers in queue at time t or the number of service stages yet to be completed at server 2 at time t, are functions of $\mathbf{X}(t)$. Note that $0 \leqslant A_t \leqslant k$ and $A_t \leqslant B_t$. The process $\mathbf{X}(t)$ renews at each point of continuous time t (i.e. the evolution of (A_s, B_s) for $s \geqslant t$ is a function only of (A_t, B_t), and is independent of all that happened in $[0, t)$), because both the interarrival time and the service stage distributions are exponential. $\mathbf{X}(t)$ is, in fact, a continuous time Markov chain. It is easily seen that the T_n's are stopping times for the process $\mathbf{X}(t)$, and $\mathbf{X}(t)$ therefore renews at each time T_n. In other words, $\mathbf{X}_n \equiv \mathbf{X}(T_n) = (A_{T_n}, B_{T_n})$ is a Markov chain. Similarly, the process $\mathbf{X}(t)$ also renews at each (random) epoch in time when a service stage is completed.

In the next Section, the transition probabilities for $\mathbf{X}_n$ are presented. In Section 2.2 various quantities of interest are derived in terms of these transition probabilities.

2.1. *Transition probabilities for* $\mathbf{X}_n$

Let $\mathbf{x}_0 = (a_0, b_0)$ be the system state at time 0. The probabilities of interest are

$$P_{x_0}(\mathbf{x}_n; n) = P(\mathbf{X}_n = \mathbf{x}_n | \mathbf{X}_0 = \mathbf{x}_0) \qquad n = 1, 2, 3, \ldots$$

Note that the range of values $\mathbf{x}_n$ can take is determined by $\mathbf{x}_0$ and n, and includes the k stages of the nth arriving customer.

The following Propositions, the proofs for which are given in the Appendix, are sufficient to compute the $P_{x_0}(\mathbf{x}_n; n)$'s. For convenience, let $\alpha_i = \lambda/(\lambda + ik\mu)$ and $\beta_i = 1 - \alpha_i$, for $i = 1, 2$. α_1 is the probability that, given only one server is busy, a customer arrives before the next service stage completion. Similarly, α_2 is the probability that, given both servers are busy, a customer arrives before the next service stage completion. β_1 and β_2 are the probabilities of the complementary events. [Note: $\mathbf{x}_0 = (a_0, b_0)$ and $\mathbf{x}_n = (a_n, b_n)$ for $n = 1, 2, 3, \ldots$ below.]

Proposition 1: For $a_0 = b_0 = 0$,

$$P_{(a_0,b_0)}[(k,k); 1] = 1.$$

Proposition 2: For $a_0 = 0$ and $1 \leqslant b_0 \leqslant k$,

(a) if $a_1 = b_1 = k$, then

$$P_{(a_0,b_0)}[(a_1,b_1); 1] = \beta_1^{b_0}.$$

(b) if $a_1 = k$ and $k < b_1 \leqslant b_0 + k$, then

$$P_{(a_0,b_0)}[(a_1,b_1); 1] = \alpha_1 \beta_1^{b_0 - b_1 + k}.$$

This proposition represents the initial condition where server 1 is idle, server 2 is busy and there is no queue. In (a), the system empties before the first arrival; in (b), the first arrival occurs before the system empties.

Proposition 3: For $a_0 = b_0$ and $1 \leqslant b_0 \leqslant k$,

(a) if $a_1 = b_1 = k$, then

$$P_{(a_0,b_0)}[(a_1,b_1); 1] = \beta_1^{b_0}.$$

(b) if $1 \leqslant a_1 \leqslant a_0$ and $b_1 = a_1 + k$, then

$$P_{(a_0,b_0)}[(a_1,b_1); 1] = \alpha_1 \beta_1^{a_0 - a_1}.$$

This proposition represents the initial condition where server 1 is busy, server 2 is idle and there is no queue. In (a), the system empties before the first arrival; in (b), the first arrival occurs before the system empties.

Proposition 4: For $1 \leqslant a_0 < b_0 \leqslant k + 1$,

(a) if $a_1 = b_1 = k$, then

$$P_{(a_0,b_0)}[(a_1,b_1); 1] = \beta_1^{a_0} + \beta_1^{b_0 - a_0} - 1 + \alpha_2 \sum_{n_1=0}^{a_0-1} \sum_{n_2=0}^{b_0-a_0-1} (\beta_2/2)^{n_1+n_2} \binom{n_1 + n_2}{n_1}$$

(b) if $1 \leqslant a_1 \leqslant a_0$ and $b_1 = a_1 + k$, then

$$P_{(a_0,b_0)}[(a_1,b_1); 1] = \alpha_1 \beta_1^{a_0 - a_1} - \alpha_2 (\beta_2/2)^{a_0 - a_1} \sum_{n_1=0}^{b_0-a_0-1} (\beta_2/2)^{n_1} \binom{n_1 + a_0 - a_1}{n_1}$$

(c) if $k = a_1 < b_1$ and $1 \leqslant (b_1 - a_1) \leqslant (b_0 - a_0)$, then

$$P_{(a_0,b_0)}[(a_1,b_1); 1] = \alpha_1 \beta_1^{b_0 - a_0 - b_1 + a_1} - \alpha_2 (\beta_2/2)^{b_0 - a_0 - b_1 + a_1} \sum_{n_1=0}^{a_0-1} (\beta_2/2)^{n_1} \binom{n_1 + b_0 - a_0 - b_1 + a_1}{n_1}$$

(d) if $1 \leqslant a_1 \leqslant a_0$, $a_1 + k < b_1 \leqslant b_0 + k$ and $(a_0 - a_1) \leqslant (b_0 - b_1 + k)$, then

$$P_{(a_0,b_0)}[(a_1, b_1); 1] = \alpha_2(\beta_2/2)^{b_0 - b_1 + k} \binom{b_0 - b_1 + k}{a_0 - a_1}.$$

This proposition represents the initial condition where server 1 and server 2 are busy and there are at most $k + 1$ service stages in the system. In (a), the system empties before the first arrival; in (b), the first arrival finds server 1 busy and server 2 idle; in (c), the first arrival finds server 1 idle and server 2 busy; and in (d), the first arrival finds both servers busy.

Proposition 5: For $1 \leqslant a_0 \leqslant k$ and $b_0 > k + 1$,

(a) if $a_1 + k < b_1 \leqslant b_0 + k$ and a_1 as stated below, then

$$P_{(a_0,b_0)}[(a_1, b_1); 1] = \alpha_2(\beta_2/2)^{b_0 - b_1 + k} \sum_{n_1 = 0}^{n'} \binom{b_0 - b_1 + k}{n_1 k + c}$$

where

$1 \leqslant a_1 \leqslant k$ if $b_1 \leqslant b_0$,

$a_1 \in \{m_i, i = 0, 1, 2, \ldots, i'\}$ if $b_1 > b_0$ where

$i' = b_0 - b_1 + k$,

$m_i = a_0 - i$ if $a_0 > i$ and

$m_i = a_0 - i + k$ if $a_0 \leqslant i$,

$c = a_0 - a_1$ if $a_0 \geqslant a_1$,

$c = a_0 - a_1 + k$ if $a_0 < a_1$, and

$n' = \lfloor (b_0 - b_1 + k - c)/k \rfloor$. ($\lfloor x \rfloor$ denotes the greatest integer $\leqslant x$.)

(b) if $k \leqslant b_1 \leqslant a_1 + k$ and a_1 as stated below, then

$$P_{(a_0,b_0)}[(a_1, b_1); 1] = (\beta_2/2)^{b_0 - k - 1} \sum_{a_1 = 1}^{k} \sum_{n_1 = 0}^{n'} \binom{b_0 - k - 1}{n_1 k + c} P_{(a_1, k+1)}[(a_1, b_1); 1]$$

where

$1 \leqslant a_1 \leqslant k$ if $b_0 > a_0 + k$,

$a_1 \in \{0, 1, 2, \ldots, a_0, k\}$ if $b_0 \leqslant a_0 + k$,

$P_{(a_1,k+1)}[(a_1, b_1); 1]$ is found from Proposition 4,

$c = a_0 - a_1$ if $a_0 \geqslant a_1$,

$c = a_0 - a_1 + k$ if $a_0 < a_1$, and

$n' = \lfloor (b_0 - k - 1 - c)/k \rfloor$.

This proposition represents the initial condition where server 1 and server 2 are busy and there are at least $(k + 2)$ service stages in the system. In (a), the first arrival joins the queue. In (b), the first arrival immediately enters service.

Proposition 6: For $n \geqslant 2$,

$$P_{(a_0,b_0)}[(a_n, b_n); n] = \sum_{j=k}^{b_0 + (n-1)k} \sum_{i=1}^{k} P_{(a_0,b_0)}[(i, j); n - 1] P_{(i,j)}[(a_n, b_n); 1].$$

This follows directly from the fact that the process $\mathbf{X}(t)$ renews at T_n for all n. The quantities $P_{(i,j)}[(a, b); 1]$ can be found using Propositions 1 through 5.

2.2. Applications

If the probability mass function $P_{\mathbf{x}_0}(\mathbf{x}_n; n)$ of $\mathbf{X}_n$ is known, then formulae for several standard

measures of queueing performance can be developed easily. Some examples of these measures are: the expected total number of stages present in the system just after T_n, the expected number of customers present in the system just after T_n, the expected number of customers in queue just after T_n and the expected delay in queue for the nth customer. Only the last performance measure is considered in detail below. If D_n denotes the delay in queue of the nth arrival, then

$$E_{\mathbf{x}_0}(D_n) = \sum_{\mathbf{x}_n} E(D_n|\mathbf{X}_n = \mathbf{x}_n) P_{\mathbf{x}_0}(\mathbf{x}_n; n).$$

The quantity of interest, therefore, is $E(D_n|\mathbf{X}_n = \mathbf{x}_n)$. Clearly, $E(D_n|(a_n, b_n)) = 0$ if $b_n \leqslant k+1$ or if $(b_n - a_n) \leqslant k$, since both these conditions imply the nth arrival directly enters service.

Suppose the nth arrival does have to wait in queue. Then let $\pi(a, b)$ be the probability that there are exactly a service stages remaining at server 1 and exactly $(b-a)$ service stages remaining at server 2 just after the nth arrival finally enters service. Since the customer just entered service, either a or $(b-a)$ or both will be equal to k. Let EZ_m = Expected time for m service stage completions given both servers remain busy throughout the period required for these m stage completions. Then $EZ_m = m/(2k\mu)$, since the rate of service stage completions is $2k\mu$ when both servers are busy.

Conditioning on the total number of remaining service stages at server 1 and server 2 when the nth customer just enters service

$$\begin{aligned} E(D_n|(a_n, b_n)) &= \sum_{b=k+1}^{2k-1} EZ_{b_n - b}[\pi(k, b) + \pi(b-k, b)] + EZ_{b_n - 2k}\pi(k, 2k) \\ &= \sum_{b=k+1}^{2k-1} \frac{(b_n - b)}{2k\mu}[\pi(k, b) + \pi(b-k, b)] + \frac{(b_n - 2k)}{2k\mu}\pi(k, 2k). \end{aligned}$$

The formulae for $\pi(a, b)$ are given and derived in the Appendix. Finally

$$\begin{aligned} E_{\mathbf{x}_0}(D_n) = \sum_{\mathbf{x}_n:(b_n - a_n) > k} P_{\mathbf{x}_0}(\mathbf{x}_n; n) \Bigg\{ &\sum_{b=k+1}^{2k-1} [\pi(k, b) + \pi(b-k, b)] \frac{(b_n - b)}{2k\mu} \\ &+ \pi(k, 2k) \frac{(b_n - 2k)}{2k\mu} \Bigg\}. \end{aligned}$$

An expression for the cumulative distribution function of D_n in terms of the $\pi(a, b)$'s can be obtained in a similar manner.

We empirically confirmed our results by simulating an $M/E_2/2$ queue, initially empty and idle, with $\rho = 0.7$ and $\lambda = 1.0$. The delay in queue of the first 10 customers was computed. The 95% confidence intervals obtained from simulation and the values obtained from our results are shown in Table 1. Table 2 contains the same information for an $M/E_4/2$ queue, initially empty and idle, with $\rho = 0.7$ and $\lambda = 1.0$.

Table 1. Expected delay in queue for an $M/E_2/2$ queue, initially empty and idle, with $\lambda = 1.0$ and $\rho = 0.7$ (Note: 10,000 replications were used to obtain the simulation results)

Customer No.	Our results	Simulation results	
		Mean	H.L. – 95% c.i.
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.1863	0.1859	0.0081
4	0.3027	0.3105	0.0111
5	0.3932	0.3917	0.0131
6	0.4663	0.4571	0.0148
7	0.5269	0.5158	0.0162
8	0.5780	0.5573	0.0173
9	0.6218	0.6042	0.0184
10	0.6596	0.6507	0.0196

Table 2. Expected delay in queue for an $M/E_4/2$ queue, initially empty and idle, with $\lambda = 1.0$ and $\rho = 0.7$ (Note: 10,000 replications were used to obtain the simulation results)

Customer No.	Our results	Simulation results	
		Mean	H.L. – 95% c.i.
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.1920	0.2021	0.0075
4	0.2824	0.2859	0.0094
5	0.3628	0.3708	0.0114
6	0.4245	0.4345	0.0128
7	0.4756	0.4795	0.0140
8	0.5182	0.5225	0.0151
9	0.5543	0.5526	0.0160
10	0.5852	0.5832	0.0167

3. IMPLICATIONS FOR INITIALIZATION OF SIMULATIONS

In general, the goal of steady-state simulation is to estimate properties of the steady-state distribution. A judicious choice of the initial state can result in a reduction in the time required to reach steady-state. Clearly, the best choice for the initial state would be the steady-state value(s) but lack of knowledge precludes this choice.

Kelton [2] and Kelton and Law [1] present results for systems with a one-dimensional state-space (i.e. $\mathbf{X}_n$ is a scalar), and show that it is better to choose an initial state other than the popular empty-and-idle state to promote convergence to steady-state. The results in this paper extend the work to include particular two-dimensional state space systems, which require a bivariate initial state. The assumption of Erlang service time distributions lends realism to the model, since it appears that for many processes an Erlang-shaped histogram arises from the service time data.

Figure 1 is a plot of $E_{\mathbf{x}_0}(D_n)$ as a function of n, for an $M/E_2/2$ system with traffic intensity $\rho = 0.7$ and $\lambda = 1.0$, and $\mathbf{x}_0 = \{(0, 0), (2, 4), (2, 6), (2, 8), (2, 10), (2, 14), (2, 20)\}$. Kelton and Law [1], Kelton [2] and Bhat and Sahin [13] took advantage of the special structure of the transition matrix for systems with a one-dimensional state-space and efficiently computed results for large values of n. We could not find a similar efficient computational algorithm for our results. Obtaining the numerical results, therefore, was computationally very expensive*. The value for the expected steady-state delay in queue for this system (dashed line) was found from tables in Hillier and Yu [14].

The convergence of $E_{\mathbf{x}_0}(D_n)$ to steady-state is highly dependent on $\mathbf{x}_0$ and is nonmonotonic in some cases. Similar behavior was observed using discrete time analysis by Kelton [2], Kelton and Law [1] and Stanford *et al.* [12], and using continuous time analysis by Grassman [6] and Odoni and Roth [5]. Odoni and Roth identified four types of behavior: (i) monotonic convergence from below, the function being concave in time, (ii) initial decrease in the function, followed by a monotonic increase to the steady state value, (iii) monotonic convergence from above, the function being convex in time, and, (iv) monotonic convergence from above, the function being convex in time, but with a linear decrease initially. No claim was made that these types of behavior were exhaustive, and the basis for the characterization was empirical observations. The four types of behavior were observed for: (i) an empty and idle or near-empty initial state, (ii) initial state near steady-state value, (iii) initial state $>$ steady-state value, and (iv) initial state $\gg$ steady-state value, respectively, as illustrated in Fig. 1.

Figure 2 is a plot of $E_{\mathbf{x}_0}(D_n)$ as a function of n, for an $M/E_2/2$ system with $\lambda = 1.0$, $\rho = (\lambda/2\mu) = 0.9$ and queue capacity = 8. Customers who arrive when the queue is full are assumed to balk. The state-space of this system is finite because the queue capacity is finite. One-step transition probabilities for this system are given by the Propositions in Section 2.1 with one exception. The exception occurs when (a_0, b_0) is such that the queue is full. In this case, using the fact that the arrival and service stage processes renew after a balked arrival, we have the following relationships:

if $a_0 \neq 1$, then

$$P_{(a_0,b_0)}[(a_1, b_1); 1] = 0.5 \times P_{(a_0-1,b_0-1)}[(a_1, b_1); 1] + 0.5 \times P_{(a_0,b_0-1)}[(a_1, b_1); 1]$$

*We were limited to maximum values of $n \approx 30$. A program written in FORTRAN 77 took ≈ 200 min of CPU time to evaluate the delays of 30 customers on a Harris 800 computer and the VOS operating system.

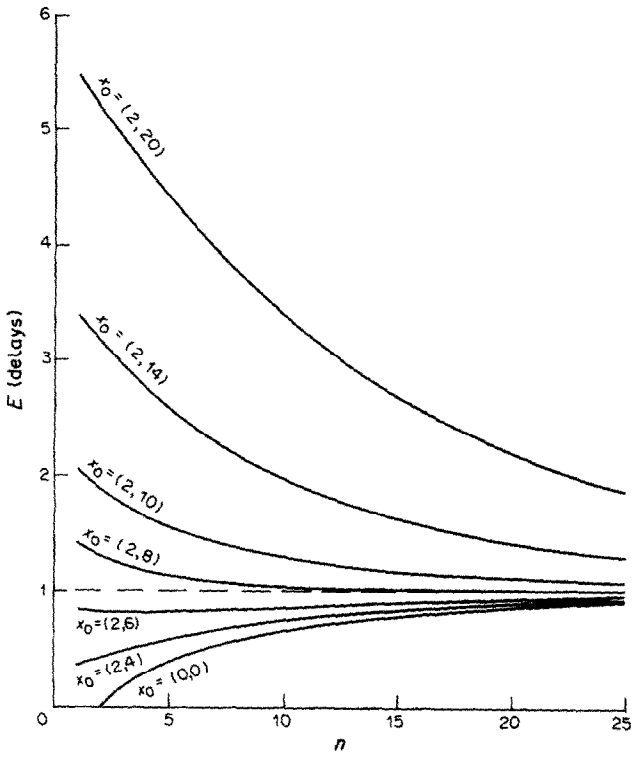

Fig. 1. $E_{x_0}(D_n)$ for an $M/E_2/2$ queue with $\lambda = 1.0$ and $\rho = 0.7$.

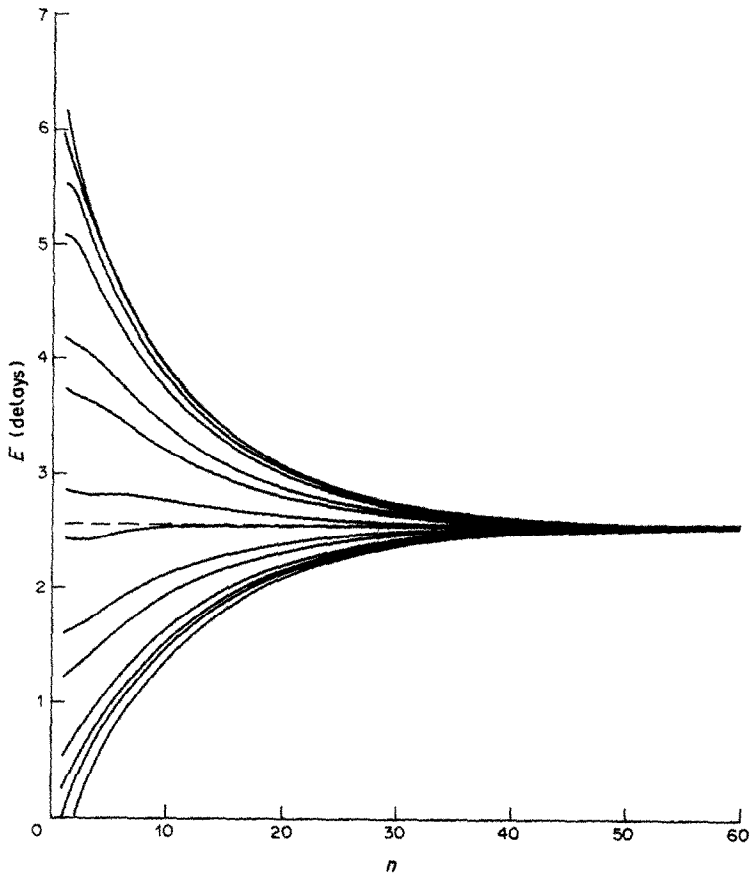

Fig. 2. $E_{x_0}(D_n)$ for an $M/E_2/2$ queue with $\lambda = 1.0$ and $\rho = 0.9$ and queue capacity = 8.

and if $a_0 = 1$, then

$$P_{(a_0,b_0)}[(a_1, b_1); 1] = 0.5 \times P_{(k,b_0-1)}[(a_1, b_1); 1] + 0.5 \times P_{(a_0,b_0-1)}[(a_{,1}, b_1); 1].$$

The first term on the right hand side accounts for the case when server 1 completes a service stage before server 2; the second term accounts for the complementary case. The probabilities on the right hand side can either be evaluated using the propositions of Section 2.1 or expressed recursively as above, depending on the subscript of P. The expected delay in queue for arriving customers is calculated as before. Curves for only 14 of the 41 possible initial states are shown in order to keep the figure uncluttered. We note that the curves starting with high values were for overcongested initial states, and those starting with low values were for undercongested initial states. The steady-state expected delay in queue was obtained from a straight forward Markov chain analysis. The transient behavior for this finite capacity queue exhibits patterns similar to those discussed above.

It is clear, as in Kelton [2] and Kelton and Law [1] that the time for the expected delays to fall within a specified tolerance zone around the steady-state value is greatly influenced by the initial state. It is therefore advisable to investigate alternative initializations for simulation of systems such as these, in order to reduce bias and to shorten the length of the non-productive warm-up periods. A method similar to the one discussed in Kelton [14] that uses a series of preliminary runs, can be employed to find reasonable values for initialization of actual production runs.

4. CONCLUSIONS

Results for the transition probabilities for the transient, discrete time $M/E_k/2$ system have been presented in this paper. Further results using these transition probabilities have also been derived. It is shown that the results are in close agreement with previously published results, which were solely for single dimensional state-space systems.

A question that may be raised is: is it more efficient (with respect to computer time) to simulate than to analytically compute the D_n's? In the case of an uncapacitated queue, it is clear that the analytical computation time grows rapidly with n, and simulation will be more efficient for large n. The actual value of n where simulation starts to get more efficient depends on k, ρ, and the precision required in the solution.

For capacitated queues with reasonably small values of queue capacity, we expect that our analytical computations will be more efficient. For example, with queue capacity equal to 8, $k = 2$ and $\rho = 0.9$ simulation took ≈ 250 times longer than the analytical computations to obtain solutions that had 99% half-lengths with 1% relative precision. When the queue capacity was raised to 20, simulation took ≈ 6 times longer than the analytical computations.

Steady-state simulation of models like the one investigated is greatly influenced by the choice of the initial state, thus warranting some experimentation to identify good starting conditions. Good starting conditions would result in a quicker approach to steady state and consequent reduction in computer run-time for the simulations.

The same method of analysis can be used for the $M/E_k/s$ system for $s > 2$, but the benefit of such an analysis is questionable, unless some method for reducing the computation time is found. Other multi-dimensional state-space models, e.g. a system with two servers in tandem can possibly be studied in a similar manner, thereby providing more analytical test models for multivariate initialization heuristics for simulations.

Acknowledgements—We would like to thank Electronic Data Systems Decision Technology Division for their support, and in particular express our appreciation to Ziv Barlach, Sam MacMillan and Michael Moore.

REFERENCES

1. W. D. Kelton and A. M. Law, The transient behavior of the $M/M/s$ queue, with implications for steady-state simulation. *Opns Res.* **33**, 378–395 (1985).
2. W. D. Kelton, Transient Exponential-Erlang queues and steady state simulation. *CACM* **28**, 741–749 (1985).
3. T. L. Saaty, *Elements of Queueing Theory with Applications*. McGraw-Hill, New York (1961).
4. L. Kleinrock, *Queueing Systems*, Vol. 1 *Theory*. Wiley, New York (1975).

5. A. R. Odoni and E. Roth, An empirical investigation of the transient behavior of stationary queueing systems. *Opns Res.* **31**, 432–455 (1983).
6. W. K. Grassman, Transient and steady-state results for two parallel queues. *Omega* **8**, 105–112 (1980).
7. C. D. Pegden and M. Rosenshine, Some new results for the $M/M/1$ queue. *Mgmt Sci.* **28**, 821–828 (1982).
8. J. S. Carson and A. M. Law, Conservation equations and variance reduction in queueing simulations. *Opns Res.* **28**, 535–546 (1980).
9. T. Morisaku, Ph.D. dissertation, Univ. of Southern Calif., Los Angeles, Calif. (1976).
10. S. C. Moore, Approximating the behavior of nonstationary single-server queues. *Opns Res.* **23**, 1011–1032 (1975).
11. C. R. Heathcote and P. Winer, An approximation for the moments of wating times. *Opns Res.* **17**, 175–186 (1969).
12. D. A. Stanford, B. Pagurek and C. M. Woodside, Optimal prediction of times and queue lengths in the $GI/M/1$ queue. *Opns Res.* **31**, 322–337 (1983).
13. U. N. Bhat and I. Sahin, Transient behavior of queueing systems. $M/D/1$, $M/E_k/1$, $D/M/1$ and $E_k/M/1$. Tech. Memo. 135, Dept. of Opns Res., Case Western Reserve Univ. (1969).
14. F. S. Hillier and O. S. Yu, *Queueing Tables and Graphs*. North-Holland, Amsterdam (1981).
15. W. D. Kelton and A. M. Law, A new approach for dealing with the startup problem in discrete event simulation. *Naval Res. Logist. Q.* **30**, 641–658 (1983).
16. M. H. Rothkopf and S. S. Oren, A closure approximation for the non-stationary $M/M/s$ queue. *Mgmt Sci.* **25**, 522–534 (1979).

APPENDIX

Proposition 1 is trivially true.

Proof of Proposition 2:

(a) Let T_1 denote the exponential arrival time of the first customer. Let S_n, $n = 1, 2, \ldots$ be random variables that denote the times of the nth service stage completion. $P_{(0,b_0)}[(k, k); 1]$ is the probability that there will be b_0 service stage completions before the first arrival and the first arrival uses server 1. Since $b_0 \geqslant 1$, there is a service stage in progress at time 0. Because this service process is memoryless, it renews at time 0. The probability that the completion of the service stage in progress is before the first arrival is $k\mu/(\lambda + k\mu)$, since the two competing processes have independent exponential distributions. At the instant of completion of the first service stage, i.e. at S_1, the arrival process renews, and therefore $P(S_2 < T_1 | S_1 < T_1) = k\mu/(\lambda + k\mu)$. Continuing in this fashion

$$P(S_1 < T_1, S_2 < T_1, \ldots, S_{b_0} < T_1) = P(S_1 < T_1).P(S_2 < T_1 | S_1 < T_1).P(S_3 < T_1 | S_1 < T_1, S_2 < T_1)\ldots$$
$$= \left(\frac{k\mu}{\lambda + k\mu}\right)^{b_0}.$$

(b) $P_{(0,b_0)}[(k, b_1); 1]$ is the probability that there will be exactly $b_0 - (b_1 - k)$ service stage completions before the first arrival. At the instant of the $[b_0 - (b_1 - k)]$th service stage completion, the arrival process renews, and the probability that the arrival will be before the next service stage completion is $\lambda/(\lambda + k\mu)$. As in (a), the probability that there will be $b_0 - (b_1 - k)$ service stage completions before the first arrival is $[k\mu/(\lambda + k\mu)]^{b_0 - (b_1 - k)}$. Due to the independence of events defined on nonoverlapping time intervals, the result follows immediately.

Proof of Proposition 3:

(a) Same as the proof of Proposition 2(a).

(b) The number of service stage completion before the first arrival is $(a_0 - a_1)$, and the result follows by using the same arguments as in the proof of Proposition 2(b).

Proof of Proposition 4:

The initial conditions for this proposition imply that the queue is empty initially.

(a) $P_{(a_0,b_0)}[(k, k); 1]$ is the probability that the first arriving customer finds the system empty and uses server 1. Let random variables C_1 and C_2 be defined as follows: C_i = no. of service stage completions from server i in $[0, T_1]$. Note that as long as server i is busy, C_i is a Poisson counting process with rate $k\mu$. Further, C_1 and C_2 are independent. Initially, there are a_0 service stages remaining at server 1, and since $b_0 \leqslant k + 1$, there are $b_0 - a_0$ service stages remaining at server 2. If the first arriving customer finds the system empty and idle, then $C_1 = a_0$ and $C_2 = b_0 - a_0$. Conditioning on T_1, the arrival time of the first customer, we get

$$P(C_1 = a_0, C_2 = b_0 - a_0) = E_{T_1}[P(C_1 = a_0, C_2 = b_0 - a_0 | T_1)]$$
$$= \int_0^\infty P(C_1 = a_0, C_2 = b_0 - a_0 | x < T_1 \leqslant x + dx)\lambda e^{-\lambda x} dx.$$

Since C_1 and C_2 are independent processes

$$P(C_1 = a_0, C_2 = b_0 - a_0 | T_1) = P(C_1 = a_0 | T_1)P(C_2 = b_0 - a_0 | T_1).$$

Since C_1 cannot exceed a_0,

$$P(C_1 = a_0 | x < T_1 \leqslant x + dx) = 1 - P(C_1 < a_0 | x < T_1 \leqslant x + dx)$$
$$= 1 - \sum_{n_1 = 0}^{a_0 - 1} \exp(-k\mu x)(k\mu x)^{n_1}/n_1!$$

Similarly

$$P(C_2 = b_0 - a_0 | x < T_1 \leqslant x + dx) = 1 - P(C_2 < b_0 - a_0 | x < T_1 \leqslant x + dx)$$
$$= 1 - \sum_{n_2=0}^{b_0-a_0-1} \exp(-k\mu x)(k\mu x)^{n_2}/n_2!$$

Finally

$$P(C_1 = a_0, C_2 = b_0 - a_0) = \int_0^\infty \left[1 - \sum_{n_1=0}^{a_0-1} e^{-k\mu x}(k\mu x)^{n_1}/n_1!\right]$$
$$\times \left[1 - \sum_{n_2=0}^{b_1-a_0-1} e^{-k\mu x}(k\mu x)^{n_2}/n_2!\right] \lambda e^{-\lambda x} dx.$$

The result follows after integration and simplification.

(b) The result is obtained by using $C_1 = a_0 - a_1$ and $C_2 = b_0 - a_0$ in the previous proof.

(c) The result is obtained by using $C_1 = a_0$ and $C_2 = b_0 - a_0 - b_1 + a_1$ in the previous proof.

(d) The result is obtained by using $C_1 = a_0 - a_1$ and $C_2 = (b_0 - a_0) - (b_1 - a_1 - k)$ in the previous proof.

Proof of Proposition 5:

The initial conditions for this proposition imply that both servers are busy and there are at least $k + 2$ service stages in the system.

(a) In this case, both servers remain busy throughout the first interarrival time, $[0, T_1]$. The number of service stage completions from the system in $[0, T_1]$ is $(b_0 - b_1 + k)$. We term a service stage completion from server 1 as a success of type 1 and one from server 2 as a success of type 2. The arrival of a new customer to the system is termed a failure. Since both servers remain busy throughout, the probability of a success is β_2 and the probability of a failure is α_2.

By the independence and memorylessness of the service stage processes and the arrival process, the probability that there will be j successes and then a failure is $\alpha_2\beta_2^j$. The probability that a given success was either of type 1 or type 2 is 1/2. Given that there were j successes, the probability that i were of type 1 is $\binom{j}{i}(1/2)^j$. Therefore, the probability that there will be i successes of type 1 and $(j - i)$ successes of type 2 and then a failure is $\alpha_2\beta_2^j\binom{j}{i}(1/2)^j$. In order that $\mathbf{X}_1 = (a_1, b_1)$, we require that the number of service stage completions from server 1 in $[0, T_1]$ be either c or $c + k$ or ... or $c + n'k$, where c and n' are as defined in the proposition statement. The result is immediate.

(b) Since $b_0 > k + 1$ and $b_1 \leqslant a_1 + k$, there has to be a time $T_1 \in (0, T_1)$ when the total remaining service stages in the system just became $(k + 1)$. Let this intermediate state be $\mathbf{X}_1 = \mathbf{X}(T_1) = (a_1, k + 1)$. Using the same reasoning as in the proof of the previous part, we see that the probability of reaching state $(a_1, k + 1)$ before the first arrival is

$$P_{a_1} = \beta_2^{b_0-(k+1)} \sum_{n_1=0}^{n'} \binom{b_0 - (k+1)}{n_1 k + c} (1/2)^{b_0-(k+1)}$$

where c and n' are as defined in the proposition statement. Since T_1 is a stopping time for $\mathbf{X}(t)$, the process $\mathbf{X}(t)$ renews at T_1. We can, therefore, use Proposition 4 to determine the probability of a transition from $(a_1, k + 1)$ to (a_1, b_1). Due to the independence of events defined on non-overlapping time intervals, we can take the product of P_{a_1} and $P_{(a_1,k+1)}[(a_1, b_1); 1]$ to obtain $P_{(a_0,b_0)}[(a_1, b_1); 1]$ conditioned on a_1. Summing over all possible a_1's gives the result.

Derivation of $\pi(i,j)$'s

We extend the definition of $\pi(i, j)$ as follows: $\pi(i,j)$ = probability that there are exactly i and $(j - i)$ service stages remaining at server 1 and server 2 respectively, conditioned on:

(i) the nth arrival had to wait in queue on arrival, and the system state just after his arrival was (a_n, b_n),

(ii) there have been exactly $(b_n - j)$ service stage completions since the nth arrival, and

(iii) the nth arrival had not yet entered service after the completion of $[b_n - (j + 1)]$ service stages since his arrival. (If the nth arrival has just entered service, then either i or $(j - i)$ or both will equal k, and the definition reduces to that stated in Section 2.2.)

In the arguments below, it is assumed that the nth arrival had to wait in queue, and *service stage completions, unless otherwise qualified, mean service stage completions since the arrival of the nth customer.*

Case 1: $b_n > 2k$

After $(b_n - 2k)$ service stage completions, the nth arrival is either first in queue or has just entered service. Since both servers are busy during these $(n_n - 2k)$ service stage completions, the probability that any of these completions is from a particular server is 1/2.

Let

$$c = (a_n - i) \qquad \text{if} \quad a_n \geqslant i,$$
$$c = (a_n - i + k) \qquad \text{if} \quad a_n < i,$$

and

$$n' = \lfloor (b_n - 2k - c)/k \rfloor.$$

Then, using the binomial distribution formula

$$\pi(i, 2k) = \sum_{n=0}^{n'} \binom{b_n - 2k}{nk + c} (1/2)^{b_n - 2k} \qquad \text{for} \quad 1 \leqslant i \leqslant k.$$

It is easy to see that the following recursive relationships hold:

For $k+1 \leqslant j \leqslant 2k-1$,

$$\pi(i,j) = 0.5\pi(i,j+1) + 0.5\pi(i+1,j+1) \qquad \text{for } 1 \leqslant i \leqslant (j-k-1)$$

$$\pi(j-k,j) = 0.5\pi(j-k,j+1)$$

$$\pi(k,j) = 0.5\pi(1,j+1).$$

Using the formula for $\pi(i, 2k)$ and the recursive formulae above, we can calculate any $\pi(i,j)$ of interest.

Case 2: $b_n \leqslant 2k$

In this case, the nth customer is the first in queue on arrival. Therefore, $\pi(i,j) = 0$ for $b_n < j \leqslant 2k$. Set $\pi(a_n, b_n) = 1$ and $\pi(i, b_n) = 0$ for $i \neq a_n$. Then $\pi(i,j)$ for $(k+1) \leqslant j \leqslant (b_n - 1)$ can be calculated recursively, as in Case 1.