TRANSIENT EXPONENTIAL - ERLANG QUEUES AND STEADY-STATE SIMULATION

by

W. David Kelton

Department of Industrial and

Operations Engineering

Technical Report 84-7

April 1984

Transient Exponential-Erlang Queues and Steady-State Simulation

W. David Kelton
Department of Industrial and Operations Engineering
The University of Michigan

ABSTRACT

The transient probabilistic structure of arbitrarily initialized $M/E_m/1$ and $E_m/M/1$ queues is derived in discrete time. Computational algorithms for obtaining the required probabilities are provided, and their application in calculating a variety of system performance measures is illustrated. The results are used to investigate the question of initializing simulations of systems such as these in order to promote rapid convergence to steady state, if that is the object of the simulation. These results are consistent with earlier studies for transient queueing systems, such as the M/M/s, but allow greater flexibility in specification of interarrival or service-time models inherent in the Erlang distributions.

1. INTRODUCTION

Analytical results for transient characteristics of queueing models are not as widely available as are steady-state results, but are directly useful for studying the finite-time properties of systems accurately represented by such models. There are several additional indirect reasons for having exact transient results in simulation applications and methodological research:

- To serve as a controlling system in the external control variates technique for variance reduction (see, for example, Gaver and Shedler [4]). If a terminating simulation is to be done for a system resembling a simpler system with known transient behavior, this second system's output from a simulation would be expected to be strongly correlated with that of the system of interest (assuming the use of common random numbers for both simulations), leading to large variance reductions. The wider the class of analytically tractable models from which to choose, the greater the similarity possible, leading to better variance reductions.

- To serve as benchmark models on which to test techniques for controlling startup bias in steady-state simulations (see Gafarian, Ancker and Morisaku [3], Schruben [18], or Kelton and Law [9]).

- To serve as benchmark models for studying alternative methods for initializing simulations aimed at estimating steady-state parameters (see Wilson and Pritsker [21] or Kelton and Law [10]).

This final reason served as the main motivation for the present paper, and is treated in more detail later.

Available transient results for queueing models may be classified according to whether the time measure is continuous (real time) or discrete

(indexing by customer number). Continuous-time results for M/M/1 and M/M/s queues appear in Morse [13], Saaty [17], Rothkopf and Oren [16], Clark [2], van Doorn [19], Whitt [20], Halfin and Whitt [6], Pegden and Rosenshine [15], Grassmann [5], and Odoni and Roth [14]; see also references in these papers. Whereas continuous-time analysis is useful for studying questions such as the queue length at a particular time or the experience of a customer who might arrive at a certain point, discrete-time results are more relevant if we are interested in the experience of, say, the nth arrival to a system or the state just after the nth arrival. Such is the case in many simulations, where one typically focuses on estimating properties of customers' delays in queue, other continuous-time parameters (e.g., mean queue length) being estimable indirectly from delay statistics via conservation equations (see Carson and Law [1]). Papers dealing with discrete-time transients of queueing systems include Heathcote and Winer [7], Morisaku [12], and Kelton and Law [10].

In this paper we extend the body of discrete-time transient results to include $M/E_m/1$ and $E_m/M/1$ queues, where E_m denotes an m-Erlang distribution. Also, our results permit arbitrary initial states of the system in terms of the number of Erlang stages present; this allows a numerical evaluation of the effect of alternative initial conditions on the nature of convergence to steady state, a general question of interest in simulation aimed at estimating steady-state parameters.

In Sections 2 and 3 the analytical results for the two classes of models are derived, with specific algorithms for computational implementation and application. Section 4 reports on numerical evaluation of these results to address questions of initialization to promote rapid convergence to steady state in simulation experiments. Some conclusions are drawn in Section 5, and the Appendix contains proofs of the results in Sections 2 and 3.

2. THE $M/E_m/1$ QUEUE

Let λ be the exponential arrival rate, μ be the m-Erlang service rate (where a complete service time is composed of m consecutive exponential stages each at rate $m\mu$), and let $\rho = \lambda/\mu$; it is not necessary to assume that $\rho < 1$ for any of the results of this paper, so that the rate of explosion of these queues could be studied in the case that no steady state exists. For $n \geq 1$, let t_n be the time of arrival of the nth customer to the system. The "method of stages" analysis of this system proceeds by using as a state variable the number of exponential service stages (rather than customers) present in the system, i.e., if there are c customers present in the system (including the one in service, if any, so $c \geq 0$) and the customer in service is in the dth of his m service stages, the system state would be $cm - d + 1$.

2.1 Mass Functions

The embedded discrete-time process used is defined (following Morisaku [12]) as

X_n = the number of service stages present in the system at time t_n, including the m stages arriving at time t_n,

for $n \geq 1$. Letting k be the number of stages in the system at time 0 ($k \geq 0$) and noting that the range on X_n is then $\{m, m+1, \ldots, k+nm\}$, the probability mass function of X_n, conditional on k, is

$$P_k(n,i) = P(X_n = i \mid X_0 = k),$$

where X_0 is the initial number of stages. The first arrival occurs at time t_1, which is exponentially distributed at rate λ; thus, $t_1 > 0$ and the first arrival finds at most k service stages already present. The following three propositions (proved in the Appendix) are sufficient for computation of the $P_k(n,i)$'s.

Proposition 1. For $n \geq 1$,

$$P_k(n,k+nm) = \begin{cases} [\rho/(\rho + m)]^n & \text{if } k \geq 1 \\ [\rho/(\rho + m)]^{n-1} & \text{if } k = 0 \end{cases}.$$

Proposition 1 represents a boundary contition i.e., that X_n is at its maximum. The following proposition establishes another boundary condition concerning the mass function of the system state just after the first arrival.

Proposition 2. For $k \geq 1$ and $m + 1 \leq i \leq k + m - 1$,

$$P_k(1,i) = [(m/(\rho + m)]^{k-i+m}[\rho/(\rho + m)].$$

Note that $k = 0$ is excluded in Proposition 2, but in this case $X_1 = m$ almost surely. The following result is the main recursion, and is true regardless of whether k is zero or positive.

Proposition 3. For $n \geq 2$ and $m + 1 \leq i \leq k + nm - 1$,

$$P_k(n,i) = [\rho/(\rho + m)] \sum_{j=\max(i-m,m)}^{k+(n-1)m} [m/(\rho + m)]^{j-i+m} P_k(n-1,j).$$

Note that the case $i = k + nm$, omitted in Proposition 3, is covered by Proposition 1, and that for the case $i = m$, we obtain

$$P_k(n,m) = 1 - \sum_{i=m+1}^{k+nm} P_k(n,i).$$

2.2 Computational Algorithm

The formulas in Propositions 1 - 3 above can be used directly to compute the mass function of X_n, but it is possible to manipulate them into the following more efficient algorithm. The procedure must be invoked in the order $n = 1$, 2, 3, ..., and takes as input the values of n, m, ρ, k, and $P_k(n-1,i)$ for $m \leq k + (n-1)m$ (unless $n = 1$); the output is $P_k(n,i)$ for $m \leq i \leq k + nm$.

```
procedure M/E_m/1 [n, m, ρ, k, P_k(n-1,i); P_k(n,i)]
  if n = 1 then
    if k = 0 then
      P_k(1,m) ← 1; return
    else
      h ← m/(ρ + m); P_k(1,k+m) ← ρ/(ρ + m); s ← P_k(1,k+m)
      for i ← k + m - 1 to m + 1 by -1 do
        P_k(1,i) ← hP_k(1,i+1); s ← s + P_k(1,i)
      end do
      P_k(1,m) ← 1 - s; return
    end if
  else
    a ← ρ/(ρ + m); b ← 1 - a; P_k(n,k+nm) ← a^(n-1); s ← 0
    if k > 0 then P_k(n,k+nm) ← aP_k(n,k+nm)
    for i ← k + nm - 1 to 2m by -1 do
      P_k(n,i) ← aP_k(n-1,i-m) + bP_k(n,i+1); s ← s + P_k(n,i)
    end do
    for i ← 2m -1 to m + 1 by -1 do
      P_k(n,i) ← bP_k(n,i+1); s ← s + P_k(n,i)
    end do
    P_k(n,m) ← 1 - s; return
  end if
end procedure M/E_m/1
```

The principal storage requirements of this algorithm are two vectors of maximal length $k + (n^* - 1)m + 1$, where n^* is the largest value of n for which the mass function of X_n is desired; the first vector holds $P_k(n-1,i)$ and the second holds $P_k(n,i)$. After the nth invocation, the previous mass function

$P_k(n-1,i)$ is replaced by the newly computed $P_k(n,i)$ for input into the (n+1)st invocation. Translation of the algorithm into any structured language should be immediate; the computations in Section 4 were carried out using VS Fortran, a subset of Fortran 77. It was found that double precision (64-bit) was necessary to avoid buildup of roundoff in the recursive computations by the time large values of n (e.g., 500) were reached.

2.3 Applications

Given the mass function $P_k(n,i)$ of X_n, it is possible to develop simple formulas for several standard measures of queueing performance. Immediately, the expectation and cumulative distribution function of X_n are given as

$$E_k(X_n) = \sum_{i=m}^{k+nm} iP_k(n,i)$$

and

$$P_k(X_n \leq x) = \sum_{i=m}^{\lfloor x \rfloor} P_k(n,i),$$

where $\lfloor \cdot \rfloor$ denotes the greatest integer function, E_k and P_k respectively denote the conditional expectation and probability measure conditioned on the event $X_0 = k$, x is any real number, and an empty sum is defined as zero.

More easily interpreted than X_n (the units of which is service stages) is Y_n, the number of customers present in the system just after time t_n. The range on Y_n is thus all integers between 1 and $p'(k,m,n) = \lfloor k/m \rfloor + n + 1$ inclusively, and the relation between Y_n and X_n is given by

$$Y_n = p \text{ if and only if } pm \leq X_n \leq (p+1)m - 1, \text{ if } 1 \leq p \leq p'(k,m,n) - 1$$

and

$$Y_n = p'(k,m,n) \text{ if and only if } p'(k,m,n)m \leq X_n \leq k + nm.$$

Thus, letting $Q_k(n,p)$ denote the mass function of Y_n, we get

$$Q_k(n,p) = \begin{cases} \sum_{i=pm}^{(p+1)m-1} P_k(n,i) & \text{if } 1 \le p \le p'(k,m,n) - 1 \\ \sum_{i=pm}^{k+nm} P_k(n,i) & \text{if } p = p'(k,m,n) \end{cases}$$

The expectation, for example, of the number of customers (not service stages) in system just after t_n is thus

$$E_k(Y_n) = \sum_{p=1}^{p'(k,m,n)} pQ_k(n,p)$$

and the cumulative distribution, variance, etc. of Y_n could be found similarly. Further, if Z_n denotes the number of customers in queue just after t_n, then

$$Z_n = \begin{cases} Y_n - 1 & \text{if } Y_n \ge 1 \\ 0 & \text{if } Y_n = 0 \end{cases}$$

from which the mass function, expectation, etc. of Z_n can be found using the probabilities $Q_n(n,p)$.

As a final application that is of most interest to simulation, let D_n be the delay in queue (excluding service time) of the nth arriving customer. If $X_n = m$, then customer n arrives to find the system empty, so $D_n = 0$. However, if $X_n = i > m$, then at least one service stage remains at the time of the nth arrival, so customer n is delayed in queue for the remainder of the in-progress service stage, plus $i - m - 1$ additional complete service stages. By exponentiality of service stages, the remainder of the in-progress stage also is exponential, so that D_n is the sum of $i - m$ independent exponential service stages, each at rate $m\mu$, i.e., D_n is an $(i-m)$-Erlang random variable with mean $(i-m)/(m\mu)$. The cumulative distribution function of D_n is then obtained by conditioning on X_n,

$$P_k(D_n \leq x) = \sum_{i=m}^{k+nm} P_k(D_n \leq x \mid X_n = i)\, P_k(n,i)$$

$$= P_k(n,m) + \sum_{i=m+1}^{k+nm} G_{i-m}(x;m\mu)\, P_k(n,i),$$

where $G_q(x;\eta)$ is the q-Erlang cumulative distribution function with mean q/η,

$$G_q(x;\eta) = 1 - e^{-\eta x} \sum_{j=0}^{q-1} (\eta x)^j/j!$$

for any $x \geq 0$. Similarly, the expected delay in queue of the nth customer is

$$E_k(D_n) = [1/(m\mu)] \sum_{i=m+1}^{k+nm} (i-m)\, P_k(n,i).$$

Section 4 discusses results of evaluating $E_k(D_n)$ over a range of system parameters. Finally, the distribution of the total system wait of the nth customer, $W_n = D_n + S_n$, where S_n is an independent m-Erlang service time, can be found from the distribution of D_n.

3. THE $E_m/M/1$ QUEUE

As before, let λ and μ denote the arrival and service rates, and define $\rho = \lambda/\mu$. Here, however, service times are simply exponential, and we think of an arrival as occurring in m consecutive independent exponential stages, each at rate $m\lambda$, with exactly one customer in some stage of arrival at all times; see, for example, Kleinrock [11]. As soon as an arriving customer finishes the mth stage of arrival (and thus physically arrives to the system), another customer begins the first stage of his arrival. The state of the system is the number of exponential arrival stages present, counting m for each customer physically present. Thus, if c customers are physically present (including the customer in service, if any) and the customer currently in the arrival process is in the dth stage of arrival ($1 \leq d \leq m$), the system state would be $cm + d - 1$.

3.1 Mass Functions

Let t_j be the time of the jth arrival stage completion, for $j \geq 1$, and let X_j be the number of arrival stages present just after time t_j. If there are k ($k \geq 0$) arrival stages present at time 0, then the time of the nth ($n \geq 1$) customer arrival (physical) to the system is $t_{nm-k+m\lfloor k/m \rfloor}$. Note that at each t_j the system state rises by 1, and at the time of each service completion the state falls by m. For $j \geq 1$ and $k \geq 0$, let

$$P_k(j,i) = P(X_j = i \mid X_0 = k), \tag{1}$$

where X_0 is the number of arrival stages present at time 0. Again, t_1 is not zero, but is exponential with mean $1/(m\lambda)$. To determine the range of X_j, first note that it is maximally $j + k$, an attainable bound in the event that no departures occur in $[0, t_j]$. At the other extreme, the minimal possible value for X_j occurs if there are no customers in the system just after time t_j other than the one completing an arrival stage at that time. If $j + k$ is divisible by m, then a customer physically arrives at time t_j, so the minimal X_j is m; otherwise, the minimal X_j is in $\{1, 2, \ldots, m-1\}$. In either case, the minimal X_j is attained if the maximal number of departures in $[0, t_j]$ occurs, which is $\lfloor (j + k - 1)/m \rfloor$. Finally, since each customer departure drops the system state by m, it is not possible for X_j to take on all integral values between its minimum and $j + k$. Thus, the general range on X_j is

$$\{j + k - fm : 0 \leq f \leq \lfloor (j + k - 1)/m \rfloor\}, \tag{2}$$

the range of i over which the mass function in (1) must be computed. Here, f represents the number of service completions occurring in $[0, t_j]$.

The following three results, proved in the Appendix, are sufficient for calculating the mass function in (1) over the range in (2):

Proposition 4. For $j \geq 1$,

$$P_k(j,j+k) = \begin{cases} 1 & \text{if } k < m \text{ and } j \leq m-k \\ [m\rho/(m\rho+1)]^{j+k-m} & \text{if } k < m \text{ and } j > m-k \\ [m\rho/(m\rho+1)]^{j} & \text{if } k \geq m \end{cases}.$$

Proposition 5. For $k \geq m$ and $0 < f < \lfloor k/m \rfloor$,

$$P_k(1,k+1-fm) = m\rho/(m\rho+1)^{f+1}.$$

Proposition 6. For $j \geq 2$ and $1 \leq f \leq \lfloor (j+k-1)/m \rfloor - 1$,

$$P_k(j,j+k-fm) = [m\rho/(m\rho+1)] \sum_{g=0}^{f} [1/(m\rho+1)]^{f-g} P_k(j-1,j-1+k-gm).$$

Note that the analogue to Proposition 5 for the case $k < m$ is covered by the first two branches of Proposition 4, since in this case the range of f in the set (2) for $j = 1$ is simply $f = 0$; thus $P_k(1,k+1-fm) = P_k(1,1+k)$, which is in Proposition 4 for $j = 1$. Similarly, if f were 0 in Proposition 5, then Proposition 4 would instead apply. Finally, as a result of Proposition 5 we obtain

$$P_k(1,k+1-\lfloor k/m \rfloor m) = 1 - \sum_{f=0}^{\lfloor k/m \rfloor - 1} P_k(1,k+1-fm),$$

completing calculation of the mass function of X_1. In Proposition 6, the cases $j = 1$ or $f = 0$ are covered by Proposition 4, and we obtain

$$P_k(j,j+k-\lfloor (j+k-1)/m \rfloor m) = 1 - \sum_{f=0}^{\lfloor (j+k-1)/m \rfloor - 1} P_k(j,j+k-fm).$$

3.2 Computational Algorithm

As for the $M/E_m/1$ model, we can derive from Propositions 4 - 6 a more efficient recursive algorithm for computing the values for the $P_k(j,i)$ mass function for i in the set (2). The following procedure is entered in the

order $j = 1, 2, \ldots,$ takes as input j, m, ρ, k, and the values in $P_k(j-1,i)$ (unless $j = 1$) and returns the values $P_k(j,i)$ for i in (2).

```
procedure E_m/M/1 [j, m, ρ, k, P_k(j-1,i); P_k(j,i)]
  a ← mρ/(mρ + 1); b ← 1 - a
  if j = 1 then
    if k < m then
      P_k(1,k+1) ← 1; return
    else
      P_k(1,k+1) ← a; s ← a
      for f ← 1 to ⌊k/m⌋ - 1 do
        P_k(1,k+1-fm) ← bP_k(1,k+1-(f-1)m); s ← s + P_k(1,k+1-fm)
      end do
      P_k(1,k+1-⌊k/m⌋ m) ← 1 - s; return
    end if
  else
    if k < m then
      if j ≤ m - k then
        P_k(j,k+j) ← 1
      else
        P_k(j,k+j) ← a^(j+k-m)
      end if
    else
      P_k(j,k+j) ← a^j
    end if
    s ← P_k(j,k+j)
    if ⌊(j + k - 1)/m⌋ = 1 then go to LAST
    if k ≥ m or j ≥ m - k + 1 then
```

```
      P_k(j,k+j-m) ← bP_k(j,j+k) + aP_k(j-1,j-1+k-m)
    else
      P_k(j,k+j-m) ← a[b + P_k(j-1,j-1+k-m)]
    end if
    s ← s + P_k(j,k+j-m)
    for f ← 2 to ⌊(j + k - 1)/m⌋ - 1 do
      P_k(j,j+k-fm) ← bP_k(j,j+k-(f-1)m) + aP_k(j-1,j-1+k-fm)
      s ← s + P_k(j,j+k-fm)
    end do
LAST:
    P_k(j,j+k-⌊(j+k-1)/m⌋ m) ← 1 - s
  end if
  return
end procedure E_m/M/1
```

As before, two vectors of storage are required, one for the previous j and one for the current j. Also, the length of the vector for storing $P_k(j,i)$ is not $j + k$ (the maximal i), but is $\lfloor (j + k - 1)/m \rfloor + 1$, the number of values in (2).

3.3 Applications

Given $P_k(j,i)$, several measures of performance of the transient $E_m/M/1$ queue are possible which parallel those for the $M/E_m/1$ described in Section 2.3. Again letting D_n be the delay in queue of the nth (physically) arriving customer, recall that $t_{j(n)}$ is the time the nth customer physically joins the system, where $j(n) = nm - k + m\lfloor k/m \rfloor$; thus, we are concerned only with the probabilities $P_k(j(n),k+j(n)-fm) = P_k(j(n),(n+\lfloor k/m \rfloor -f)m)$, for $0 \leq f \leq n + \lfloor k/m \rfloor - 1$. If $f = n + \lfloor k/m \rfloor - 1$, then the arrival at $t_{j(n)}$ finds the system empty, so $D_n = 0$. If $f < n + \lfloor k/m \rfloor - 1$, then this arrival finds

$n + \lfloor k/m \rfloor - f - 1$ other customers in the system, each of which still requires an independent exponential service with mean $1/\mu$, so D_n is in this case an $(n-\lfloor k/m \rfloor -f-1)$-Erlang random variable with mean $(n + \lfloor k/m \rfloor - f - 1)/\mu$. Thus, given $x \geq 0$, the cumulative distribution function of D_n is

$$P_k(D_n \leq x) = \sum_{f=0}^{n+\lfloor k/m \rfloor -2} G_{n+\lfloor k/m \rfloor -f-1}(x;\mu)\ P_k(j(n),k+j(n)-fm) + P_k(j(n),nm)$$

and the expected delay in queue is

$$E_k(D_n) = (1/\mu) \sum_{f=0}^{n+\lfloor k/m \rfloor -2} (n+\lfloor k/m \rfloor -f-1)\ P_k(j(n),(n+\lfloor k/m \rfloor -f)m),$$

where an empty sum is taken to be zero. Again, properties of the total system wait W_n of the nth customer may be derived from these expressions.

As a second application, let Y_n be the number of customers present in the system just after $t_{j(n)}$. Thus, $Y_n = X_j/m$, and is a discrete random variable on $\{1, 2, \ldots, n+\lfloor k/m \rfloor\}$ with mass function $P_k(Y_n=i) = P_k(j(n),im)$, enabling computation of its cumulative distribution, moments, etc.

4. EVALUATION OF ALTERNATIVE INITIALIZATIONS FOR SIMULATION

The principal motivation for this study was to examine the effect of alternative simulation initialization policies on the nature of convergence of simulation output to steady state, for models similar to those analyzed above. In a steady-state simulation, the goal is to estimate some property of the steady-state distribution (assuming it exists) of the output stochastic process, often its expectation. One of the most difficult problems facing an analyst in this case is choosing initial conditions which are in some sense representative of steady state. Of course, it is impossible to do this exactly since knowledge of the steady state distribution would be required. Results such as those obtained above can be used to address the question of

prudent choice of initial condidtions.

In [10] we carried out such an investigation for M/M/s queues, and found that for that system, it may be wise to initialize in something other than the popular empty-and-idle state to reduce the length of time required for near-steady-state conditions to be attained. One limitation of that study was that all interarrival time and service time distributions were assumed to be exponential. The results of this paper, while limited to single-server systems, allow a much richer class of distributions to be used, the Erlang, providing more realism; an Erlang assumption for service-time distributions is especially attractive, since it appears that for many processes an Erlang-shaped histogram arises from service time data. For [10] as well as this paper, the critical point is that analytical transient results were obtained allowing an arbitrary initial state specification, so we can evaluate the resulting functions for various choices of initialization and observe convergence behavior.

Values for $E_k(D_n)$ were computed for both the $M/E_m/1$ and $E_m/M/1$ systems across a range of parameter values: ρ = 0.5, 0.8, and 0.9 (always setting λ = 1), and m = 2, 4, 8, and 9; k was chosen as described below. Figures 1 and 2 show plots of $E_k(D_n)$, as functions of n, for various initialization schemes, for m = 4 and ρ = 0.9. The number of customers (not stages) physically present in the system initially is c, shown next to the corresponding curves. For simplicity, we assumed for the $M/E_m/1$ model in Figure 1 that the customer in service (if any) was just beginning his first service stage at time 0; similarly, for the $E_m/M/1$ model of Figure 2, the customer in the arrival process at time zero was assumed to be at the start of his first stage of arrival. In both figures, the dashed line is at the expected steady-state delay in queue, found from Kleinrock [11] for Figure 1 and from tables in

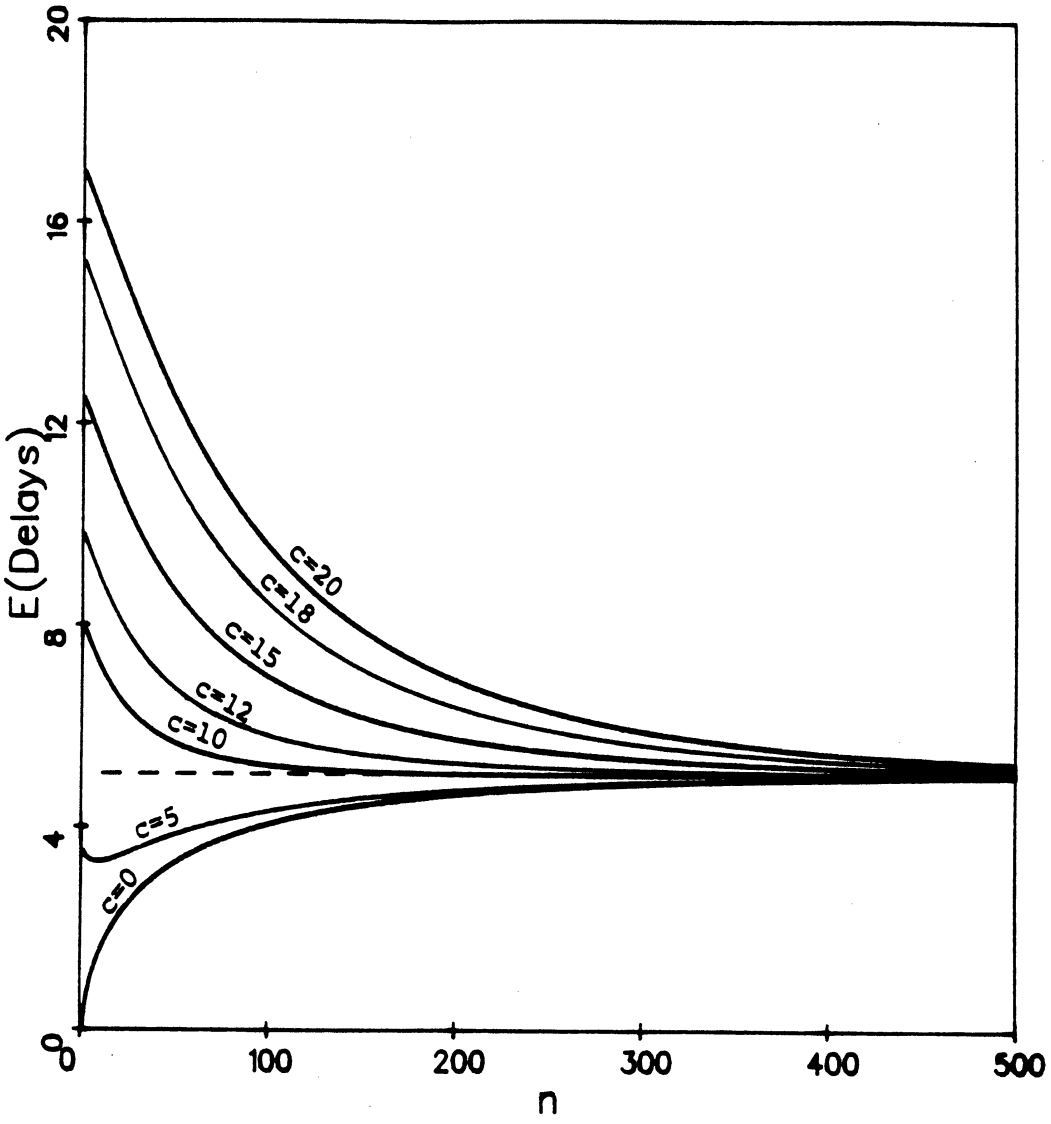

Figure 1. $E_k(D_n)$ for the $M/E_4/1$ Queue with $\rho = 0.9$ and $k = 4c$.

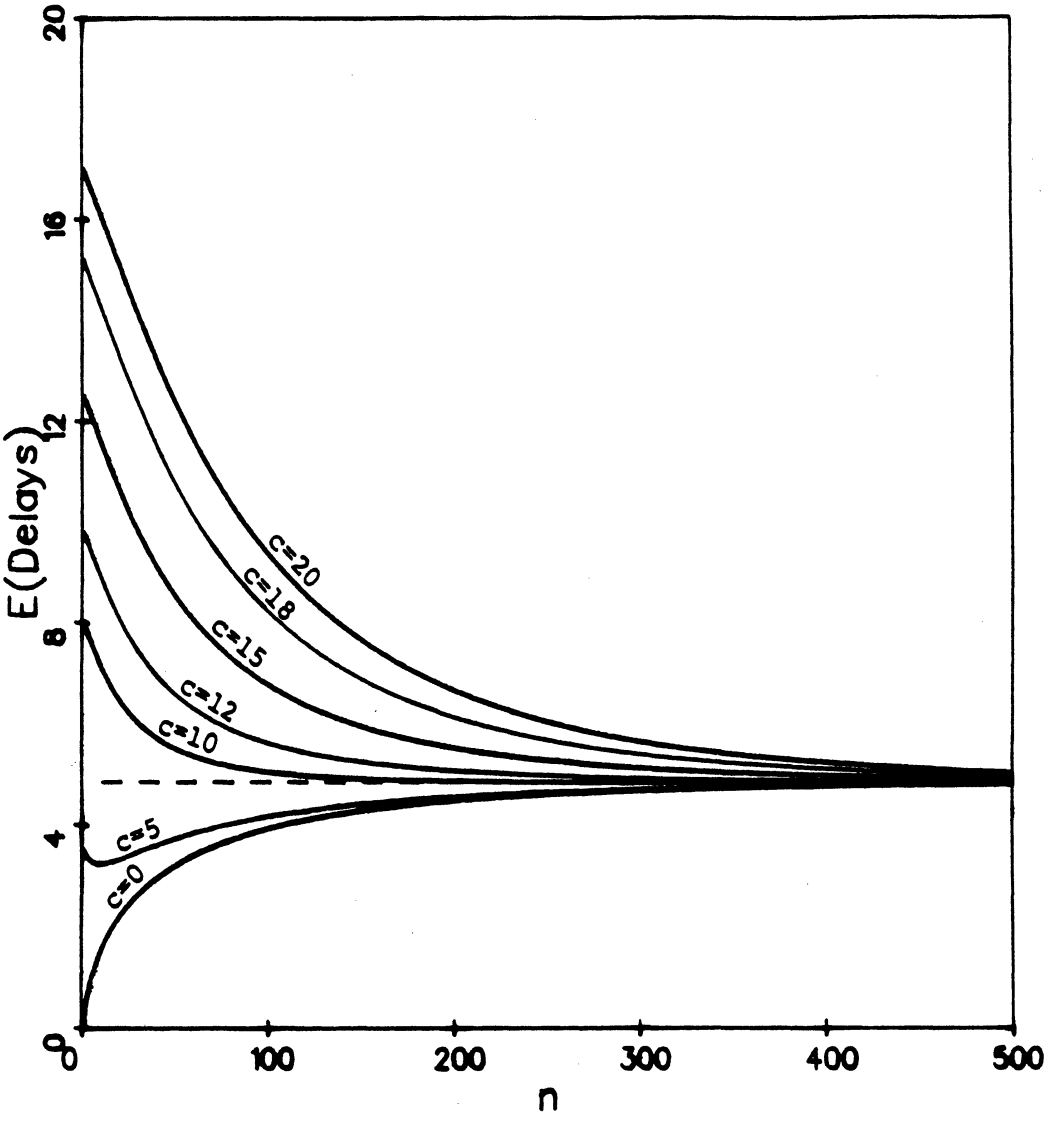

Figure 2. $E_k(D_n)$ for the $E_4/M/1$ Queue with $\rho = 0.9$ and $k = 4c$.

Hillier and Yu [8] for Figure 2.

In both cases, $E_k(D_n)$ converges to the steady-state expected delay monotonically from below if c = 0, but for other choices of c the approach may be nonmonotonic (e.g. c = 5 in both figures). Such behavior has been noticed for the discrete-time M/M/s queues in [10], as well as in continuous-time transient results by Grassmann [5] and Odoni and Roth [14]. As pointed out in these last two papers, the initial decrease in the curves (even though they begin below the steady-state mean) is attributable to the fact that c > 0 implies that the server is initially busy (which is not necessarily the case in steady state), increasing the probability of a downward state transition with respect to its steady-state value. Plots of other cases exhibited similar behavior; sometimes $E_k(D_n)$ would begin above the dashed line, decrease through it, then turn and converge from below.

As in [10], it is clear that nonempty initialization can greatly reduce the time for the expected delays to fall within a specified band about the steady-state value. Thus, in simulating systems such as these, it would be advisable to investigate alternative initializations, especially if a large number of replications are to be made for purposes of statistical analysis, in order to reduce bias and shorten the length of nonproductive warmup periods. Although optimality studies such as those in [10] could be carried out, it is anticipated that the results and recommendations would be similar. In particular, empty and idle (c = 0) initialization may not be a good idea unless ρ is quite small.

5. CONCLUSIONS

In this paper we have derived new discrete-time transient results for two classes of single-server queueing systems that admit a rich family of

interarrival or service time distributions; explicit algorithms for computation of the required probability mass functions have also been provided. Given these probabilities, several system performance measures were derived and one of these, the expected delay in queue, was numerically evaluated. The results of this evaluation were used to investigate the choice of initial conditions in simulations of models such as these aimed at estimating steady-state characteristics. The warmup period may be greatly affected by this choice, indicating that some initial experimentation with a given simulation model may prove fruitful in identifying good starting conditions.

APPENDIX

To prove Propositions 1 - 3, let A denote an exponential interarrival time random variable and let S be an m-Erlang service time; thus, $E(A) = 1/\lambda$ and $S = S_1 + \cdots + S_m$ where the S_i's are independent exponential with common mean $1/(m\mu)$. Then $P(A < S_i) = \lambda/(\lambda + m\mu) = \rho/(\rho + m)$.

Proof of Proposition 1: $P_k(n,k+nm) = P_k(X_n = k+nm)$, and since $X_n \leq k + nm$, this is the probability of no service stage completions in $[0,t_n]$, a period of n consecutive interarrivals. If $k \geq 1$ then a service stage is in progress at time 0, at which time it renews (due to exponential memorylessness), and will again renew at each arrival. Thus, $P_k(n,k+nm)$ is the probability of n interarrivals during a single service stage, so is

$$P(A_1 < S_1, A_2 < S_1, \ldots, A_n < S_1) = [P(A_i < S_1)]^n$$
$$= [\rho/(\rho + m)]^n,$$

where A_i is the ith interarrival. On the other hand, if $k = 0$ then no departure in $[0,t_1]$ is possible, so $X_n = k + nm$ is equivalent to interarrivals $A_2, \ldots, A_n$ occurring during a single service stage, having probability

$[\rho/(\rho + m)]^{n-1}$.

Proof of Proposition 2: The event here is that exactly i service stages are present just after the first customer arrival. This is equivalent to there being k - i + m service stage completions in $[0,t_1]$ and the arrival must occur before the next service stage completion. Thus, the event is

$$\{S_1 < A, \ldots, S_{k-i+m} < A, S_{k-i+m+1} > A\},$$

since the interarrival time A renews at the time of each service stage completion. Thus, the required probability is $[m/(\rho + m)]^{k-i+m}[\rho/(\rho + m)]$.

Proof of Proposition 3: The event is that $X_n = i$, and we will condition on $X_n = j$, in which case j must be at least i - m if X_n is to equal i. Further, since $m \leq X_{n-1} \leq k + (n - 1)m$, the range of j is

$$\max\{i-m,m\} \leq j \leq k + (n - 1)m.$$

Thus,

$$P_k(n,i) = \sum_{j=i-m}^{k+(n-1)m} P_k(X_n = i \mid X_{n-1} = j)\, P_k(n-1,j).$$

To compute the conditional probability, note that if $X_{n-1} = j$, then $X_n = i$ exactly if j - i + m service stage completions occur in the space of the single interarrival $t_n - t_{n-1}$ and the nth arrival occurs before an additional service stage completion. Thus,

$$P_k(X_n = i \mid X_{n-1} = j) = [m/(\rho + m)]^{j-i+m}[\rho/(\rho + m)],$$

the desired result after simplification.

To prove Propositions 4 - 6, it is convenient to let A_i and S denote an interarrival stage completion and a service time, respectively; an interarrival time is thus $A = A_1 + \cdots + A_m$. With this notation, $P(A_i < S) = m\lambda/(m\lambda + \mu) = m\rho/(m\rho + 1)$.

Proof of Proposition 4:

Case 1: $k < m$. Since $k < m$, no customers are initially present (physically) in the system, so the initial departure rate is 0. The required probability is that of the event of no departures in $[0,t_j]$, a period of j consecutive exponential interarrival stages. Since the first arrival occurs with the (m-k)th interarrival stage completion, $j \leq m - k$ implies that at time t_j no customer has arrived, so X_j must be equal to $j + k$. If $j > m - k$, then the required probability is that of no departures during j interarrival stages, the final $j - (m - k)$ of which are during a period when the server is busy, which is

$$[P(A_i < S)]^{j-(m-k)} = [m\rho/(m\rho + 1)]^{j+k-m}.$$

Case 2: $k \geq m$. Here, the server is initially busy and, as in Case 1, we want the probability of no departures in the span of j interarrival stages; this is

$$[P(A_i < S)]^{j} = [m\rho/(m\rho + 1)]^{j}.$$

Proof of Proposition 5: $X_1 = k + 1 - fm$ exactly if there were f departures in $[0,t_1]$, occurring if f service completions take place during one interarrival stage but the next event is the interarrival stage completion. Thus,

$$\begin{aligned} P_k(1,k+1-fm) &= [P(A_i > S)]^f P(A_i < S) \\ &= [1/(m\rho + 1)]^f [m\rho/(m\rho + 1)], \end{aligned}$$

as desired.

Proof of Proposition 6: Conditioning on X_{j-1},

$$P_k(j,j+k-fm) = \sum_{g=0}^{\lfloor (j+k-2)/m \rfloor} P_k(X_j=j+k-fm \mid X_{j-1}=j-1+k-gm)\, P_k(j-1,j-1-k-gm). \quad (A1)$$

Case 1: $j + k - 1$ is divisible by m. Then $X_j \geq m$, so departures in $[t_{j-1},t_j]$ are possible regardless. If the number of such departures is $f - g$, then

$0 \le f - g \le \lfloor (j - 1 + k - gm)/m \rfloor$, implying that $g \le f$ as well as $g \le \lfloor (j + k - 2)/m \rfloor$ on the range of the summation in (A1). However, since $f \le \lfloor (j + k - 1)/m \rfloor - 1$ by assumption, it is easy to see that $g \le f$ is the binding upper bound on g, so the range on g in (A1) can be reduced to $\{0, 1, \ldots, f\}$. Finally, the probability of exactly $f - g$ departures in $[t_{j-1}, t_j]$ is

$$[1/(m\rho + 1)]^{f-g}[m\rho/(m\rho + 1)], \tag{A2}$$

being $f - g$ departures followed by an interarrival stage completion.

Case 2: $j + k - 1$ is not divisible by m. Thus, m must be at least 2. Then the minimal X_{j-1} is at most $m - 1$, creating the possibility of a zero exit rate; this is true for the $g = \lfloor (j + k - 2)/m \rfloor$ term in the sum in (A1). In this case, X_j must be $j + k - gm$, so

$$P_k(X_j = j+k-fm \mid X_{j-1} = j-1+k-\lfloor (j+k-2)/m \rfloor m) = \begin{cases} 1 \text{ if } f = \lfloor (j+k-2)/m \rfloor \\ 0 \text{ otherwise} \end{cases}.$$

However, if $f = \lfloor (j + k - 2)/m \rfloor$, then due to the upper bound on f in the proposition statement, we must have

$$f = \lfloor (j + k - 2)/m \rfloor \le \lfloor (j + k - 1)/m \rfloor - 1. \tag{A3}$$

To show that (A3) cannot hold, note that since $j + k - 1$ is not divisible by m, there are integers h and r with $1 \le r < m$ such that $j + k - 1 = hm + r$. Thus,

$$\begin{aligned} \lfloor (j + k - 2)/m \rfloor &= h + \lfloor (r - 1)/m \rfloor \\ &= h, \end{aligned}$$

since $1 \le r < m$, and

$$\begin{aligned} \lfloor (j + k - 1)/m \rfloor - 1 &= h + \lfloor (r - m)/m \rfloor \\ &= h + \lfloor r/m \rfloor - 1 \\ &= h - 1, \end{aligned}$$

since $r < m$. Thus (A3) is contradicted, so that we must have $f <$

$\lfloor (j + k - 2)/m \rfloor$, for f in the range considered by the proposition. Therefore, the $g = \lfloor (j + k - 2)/m \rfloor$ term in the sum in (A1) drops out, and the range on g can be restricted to $0 \leq g \leq \lfloor (j + k - 2)/m \rfloor - 1$. For such g, $X_{j-1} \geq m$, i.e., there is at least one customer physically in the system after t_{j-1}. In this case, $X_j = j + k - fm$ if and only if $f - g$ departures occur in $[t_{j-1}, t_j]$. Also, $g \leq \min\{f, \lfloor (j+k-2)/m \rfloor - 1\}$, and this minimum is easily seen to be f. Thus, the range on g in the sum in (A1) is $\{0,1,\ldots,f\}$, and $P_k(X_j = j+k-fm \mid X_{j-1} = j-1+k-gm)$ is given by (A2), completing the proof.

REFERENCES

1. Carson, J.S. and Law, A.M. Conservation Equations and Variance Reduction in Queueing Simulations. Operations Research 28 (1980), 535-546.

2. Clark, G.M. Use of Polya Distributions in Approximate Solutions to Nonstationary M/M/s Queues. Commun. ACM 24 (1981), 206-217.

3. Gafarian, A.V., Ancker, C.J., Jr., and Morisaku, T. Evaluation of Commonly Used Rules for Detecting 'Steady State' in Computer Simulation. Naval Res. Logist. Quart. 25 (1978), 511-529.

4. Gaver, D.P. and Shedler, G.S. Control Variable Methods in the Simulation of a Multiprogrammed Computer System. Naval Res. Logist. Quart. 18 (1971), 435-450.

5. Grassman, W.K. Transient and Steady State Results for Two Parallel Queues. Omega 8 (1980), 105-112.

6. Halfin, S. and Whitt, W. Heavy-Traffic Limits for Queues with Many Exponential Servers. Operations Res. 29 (1981), 567-588.

7. Heathcote, C.R. and Winer, P. An Approximation for the Moments of Waiting Times. Operations Res. 17 (1969), 175-186.

8. Hillier, F.S. and Yu, O.S. Queueing Tables and Graphs. North Holland, New York, 1981.

9. Kelton, W.D. and Law, A.M. A New Approach for Dealing with the Startup Problem in Discrete Event Simulation. Naval Res. Logist. Quart. 30 (1983), 641-658.

10. Kelton, W.D. and Law, A.M. The Transient Behavior of the M/M/s Queue, with Implications for Steady-State Simulation. Operations Res., forthcoming.

11. Kleinrock, L. Queueing Systems, Vol. 1: Theory. Wiley, New York, 1975.

12. Morisaku, T. Techniques for Data Truncation in Digital Computer Simulation. Ph.D. Dissertation, Department of Industrial and Systems Engineering, University of Southern California, 1976.

13. Morse, P.M. Stochastic Properties of Waiting Lines. J. Operations Res. Soc. Amer. 3 (1955), 255-261.

14. Odoni, A.R. and Roth, E. An Empirical Investigation of the Transient Behavior of Stationary Queueing Systems. Operations Res. 31 (1983), 432-455.

15. Pegden, C.D. and Rosenshine, M. Some New Results for the M/M/1 Queue. Management Sci. 28 (1982), 821-828.

16. Rothkopf, M.H. and Oren, S.S. A Closure Approximation for the Nonstationary M/M/s Queue. Management Sci. 25 (1979), 522-534.

17. Saaty, T.L. Time-Dependent Solution of the Many-Server Poisson Queue. Operations Res. 8 (1960), 755-772.

18. Schruben, L.W. Detecting Initialization Bias in Simulation Output. Operations Res. 30 (1982), 569-590.

19. van Doorn, E. Stochastic Monotonicity and Queueing Applications in Birth-Death Processes. Springer-Verlag, New York, 1981.

20. Whitt, W. Comparing Counting Processes and Queues. Adv. Appl. Prob. 13 (1981), 207-220.

21. Wilson, J.R. and Pritsker, A.A.B. Evaluation of Startup Policies in Simulation Experiments. Simulation 31 (1978), 79-89.