APPROXIMATION ANALYSIS FOR OPEN TANDEM QUEUES WITH BLOCKING: EXPONENTIAL AND GENERAL SERVICE DISTRIBUTION

Stephen M. Pollock
John R. Birge
Jeffrey M. Alden

Department of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109

October, 1985

Technical Report 85-30

Approximation Analysis for Open Tandem Queues with Blocking: Exponential and General Service Distributions

Stephen M. Pollock, John R. Birge, and Jeffrey M. Alden

University of Michigan, Ann Arbor, Michigan

Calculating occupancy probabilities for open finite tandem queueing systems using Markov Process analysis is impractical for all but small systems due to the large number of states. We offer an iterative procedure for approximating the marginal occupancy probabilities for each queue of the system. The procedure is easy to implement, requires little memory, and is computationally fast. Two implementations are presented: one for exponential service distributions and one for general service distributions. Both implementations give better accuracy than other approximation methods.

Introduction

Systems of servers in tandem (series), where the output of one server is the input to the next one in line, can represent many processes of interest in manufacturing, computer systems, telecommunications, etc. This is particularly true when the service times are random variables. When the storage ("buffer") space between servers is finite, so that an upstream server can be "blocked" due to unavailability of space in the next buffer, the resulting system becomes notoriously hard to analyze. (For an excellent review, see Perros [1984].)

In particular, exact solutions (i.e., ones that provide the usual measures of queueing performance) are available only for either infinite intermediate buffers (Gordon and Newell [1967], Hordijk and Van Dijk [1981])—in which case solutions are obtained by using product-form representations of state probabilities—or small numbers of servers and buffer sizes (Asare [1978], Caseau and Pujolle [1979], Foster and Perros [1980], Gershwin [1983], Gordon and Newell [1967], Hillier and Boling [1967], Konheim and Reiser [1976], and Labetoulle and Pujolle [1977]). In these studies, service times at each server are assumed to be negative exponentially distributed random variables or, in the case of Gershwin [1983], a probability mixture of a discrete geometric random variable and a constant.

One reason for the lack of success in studying this important class of problems is that a comprehensive representation of the state-space of the system requires at least an enumeration (if not a computation!) of state variables that grows combinatorially with the number of servers. This unfortunate fact has led to a number of analyses based upon approximation methods. Most of these (Altiok [1982], Boxma and Konheim [1981], Hillier and Boling [1967], Suri and Diehl [1983], Takahashi et al. [1980]) collapse or aggregate system states into "super-states" or "marginal states" that in turn serve to characterize (to some degree of accuracy) the desired system performance measures.

We present here an approximation method that is similar to those presented in Perros and Altiok (forthcoming) and Takahashi et al. (1980) , but with simplifications (each server only considers the behavior of—or information from—its immediate neighbors), generalizability (it can be extended to non-exponential service times), and increased accuracy.

1.0 The M-server Tandem System: Exponential Service

The fundamental model we will study is identical to the model in Takahashi et al. [1980]. There are M servers in tandem (see Figure 1). Each server has a buffer (waiting room) for holding arrivals when it is busy. Each server has an exponentially distributed service time. Units are assumed to arrive only at the first queue as a Poisson process with a rate λ.

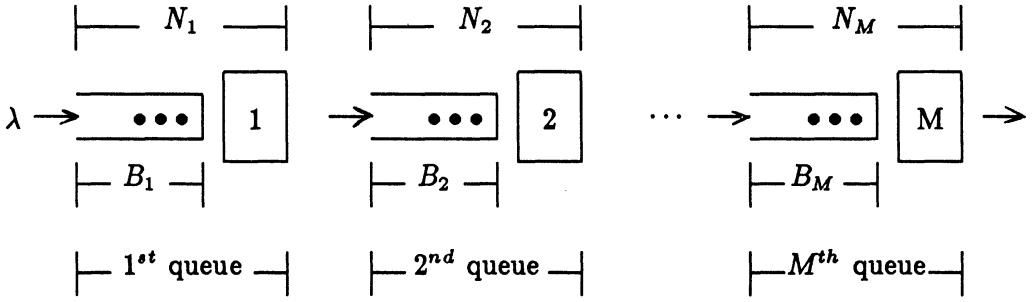

Figure 1. Tandem queueing system.

Let μ_i and B_i be the service rate and buffer capacity for the i^{th} server, where $i = 1, 2, \cdots, M$. The i^{th} queue is the combination of the i^{th} server and its buffer. Thus, if N_i is the capacity of the i^{th} queue (assumed finite for all but possibly the first queue), then

$N_i = B_i + 1$. Note that the queues are numbered from 1 to M starting from the input queue. Arrivals to any queue are served in a FIFO manner.

An arrival enters the first queue if it arrives at a time when the queue is not full; otherwise, the arrival is lost. When a unit completes service at the i^{th} queue, it proceeds to the $(i+1)^{st}$ queue if space is available. However, if the $(i+1)^{st}$ queue is full at that time, the unit must wait in the i^{th} server until a departure occurs from the $(i+1)^{st}$ queue. During this time the i^{th} server cannot serve other units that might be waiting in its buffer. In this case, the i^{th} server is blocked, and the $(i+1)^{st}$ queue is blocking. Thus, the first queue cannot be blocking, and the M^{th} server cannot be blocked.

A complete (steady-state) analysis of this system would produce all state occupancy probabilities, as well as related measures such as the expected number of units in each queue (or buffer), the waiting time distribution at each server, and the throughput of the system (since some arrivals can be turned away, the rate of units through the system is less than λ).

These measures are functions of the $2M+1$ parameters λ, μ_i and B_i $(i = 1, 2, \cdots, M)$. The system state is the vector

$$S \equiv (s_1, s_2, \ldots, s_M),$$

where $s_i = 0, 1, 2, \cdots, N_i$, is the number of units in the i^{th} queue and $s_i = N_i + 1$ indicates that the i^{th} queue is blocking the $(i-1)^{st}$ server. The total number of possible states is then $\Pi_{i=1}^{M}(N_i+2)$. For even a reasonably sized system, e.g., $M = 8, N_i = 4$ $(i = 1, 2, \cdots, 8)$, this number is over 1.7 x 10^6, and represents a considerable challenge for the computations associated with a straightforward Markov Process analysis.

1.1 The Approximation Method: General Approach

Our general approach (labelled SIMP for "simple iterative myopic procedure") is to focus on the measures that seem to be important (for example, individual queue state descriptions) and to seek an approximation that will produce these measures fairly accurately. We sacrifice the possibility of having good information about less important events (such as a specific subset of servers being simultaneously idle).

To do this, we analyze each queue separately, using information from only its nearest neighbors in order to approximate, by a simple model, what is in fact a very complicated and dependent state of affairs. In particular, we assume that the model for the i^{th} queue is a simple M/M/1/N queue with the following conditions:

a1) there are Poisson arrivals, with rate λ_i^*, as long as the i^{th} queue is not blocking (that is, as long as the number of units in the i^{th} queue is less than or equal to N_i). When the i^{th} queue is blocking ($s_i = N_i + 1$), there are no arrivals.

a2) The (random variable) time to complete service has two components: the actual service time (exponentially distributed with rate μ_i) plus a term due to the occasional and probabilistic delay caused by blocking downstream.

a3) the service completion time (having two components, see a2) is exponentially distributed, with rate μ_i^*.

These approximations are, of course, a distortion of what is actually happening in the system. Indeed, a1) is by itself a heroic assumption—the input to each queue but the first, being the (buffered) output of the preceding queue, is anything but Poisson; and a3) actually contradicts a2). Nonetheless, these assumptions allow the computation of approximate values for the i^{th} queue (steady-state) occupancy probabilities. We can then use these approximate values to evaluate measures of interest.

1.2 Definitions and Underlying Relationships

Let:

$S_i \equiv \{0, 1, 2, \cdots, N_i, N_i + 1\}$ be the state space for the i^{th} queue,

$s_i \equiv$ state of i^{th} queue ($s_i \in S_i$ is the number of units in the i^{th} queue, except when $s_i = N_i + 1$ in which case the i^{th} queue is blocking the $(i-1)^{st}$ server),

$\lambda_i^* \equiv$ arrival rate to queue i,

$\mu_i^* \equiv$ service rate of server i,

$f_i \equiv \Pr\{i^{th} \text{ queue is full}\} = \Pr\{s_i = N_i\} + \Pr\{s_i = N_i + 1\}$,

$b_i \equiv \Pr\{i^{th} \text{ queue is blocking}\} = \Pr\{s_i = N_i + 1\}$,

$E_i \equiv$ expected time between acceptances into queue i,

$T_i \equiv$ (random variable) time for server i to complete service and pass unit on to next queue, and

$E[T_i] \equiv$ the expected value of T_i.

These definitions, and the structure of the system, produce the following relationships:

1. The rate at which arrivals join the system is λ times Pr{queue 1 is not full} = $\lambda(1 - f_1)$. Thus,

$$E_1 = \frac{1}{\lambda(1 - f_1)}. \tag{1}$$

2. The expected time to complete service at server i is the unblocked expected service time $1/\mu_i$ plus the expected delay due to blocking: Pr{a served unit at server i will see queue $(i+1)$ full} times the expected time for server $(i+1)$ to complete service. The terms required for the computation are obtained from two additional (approximation) assumptions:

a4) A unit at server i at the instance service is finished sees queue $(i+1)$ in "steady-state".

a5) The expected sojourn time in the blocking state for a queue is equal to its expected service completion time.

Given these assumptions, the expected delay due to blocking is equal to server i's unconditional expected time to complete service. Thus,

$$E[T_i] = \frac{1}{\mu_i} + \frac{f_{i+1} - b_{i+1}}{1 - b_{i+1}} E[T_{i+1}], \quad i = 1, 2, \cdots, M, \tag{2}$$

where

$$E[T_{M+1}] \equiv 0.$$

Since the viewpoint is that of server i at completion of unblocked service, it is necessary to condition the $(i+1)^{st}$ queue probability on not being blocking, which produces the denominator of the second term. For the following discussion, we define the steady-state probability that queue $(i+1)$ blocks server i at the end of service as

$$\alpha_i = \frac{f_{i+1} - b_{i+1}}{1 - b_{i+1}}.$$

3. The expected time between acceptances at queue i is $1/\lambda_i^*$ plus $E[T_i]$ times Pr{an arrival sees queue i full}, since in this latter case the arrival must wait for server i to complete its current service before being accepted into the queue. We again use assumptions a4) and a5) to give

$$E_i = \frac{1}{\lambda_i^*} + \frac{f_i - b_i}{1 - b_i} E[T_i], \quad i = 2, 3, \cdots, M. \tag{3}$$

As before, the probability that queue i is full is conditioned on queue i not blocking queue $(i-1)$, since otherwise there could not be an arrival. Figure 2 shows a schematic representation of this process.

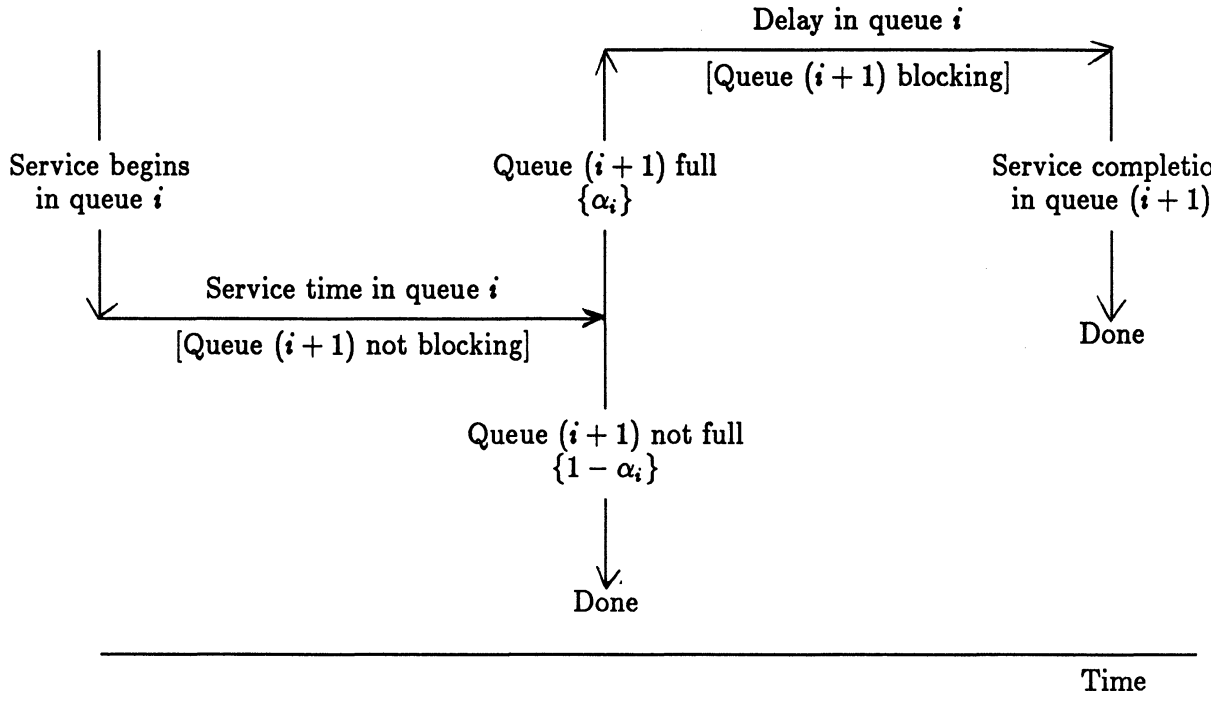

Figure 2. Service completion time in queue i.

4. By conservation of unit flow,

$$E_i = E_1, \quad i = 1, 2, \cdots, M. \tag{4}$$

5. The service completion rate is the reciprocal of the expected service completion time. Thus,

$$\mu_i^* = \frac{1}{E[T_i]}, \quad i = 1, 2, \cdots, M, \tag{5}$$

1.3 Finding the Occupancy Distributions: The M/M/1/N Model

Equations 1 through 4 together with standard M/M/1/N analysis suggest the following iterative procedure to find the arrival rate (λ_i^*), expected completion time ($E[T_i]$), and occupancy probability distribution ($\Pr\{s_i = n\}$) for each queue.

0. (Setup) Set the values for λ, μ_i , and N_i for $i = 1, 2, \cdots, M$.

1. (Initialization) Set $E[T_i] = 1/\mu_i$, $b_1 = 0$, $\lambda_1^* = \lambda$, and $\lambda_i^* = \lambda(1 - f_1)$ for $i = 2, 3, \cdots, M$ where f_1 is given by Equation 6*a*.

2. (Find full and blocking probabilities) With $\rho_i = \lambda_i^* E[T_i]$, set

$$f_1 = \frac{(1-\rho_1)\rho_1^{N_1}}{1-\rho_1^{N_1+1}}, \tag{6a}$$

$$f_i = \frac{(1-\rho_i)\rho_i^{N_i+1}}{1-\rho_i^{N_i+2}}, \quad i = 2, 3, \cdots, M, \quad \text{and} \tag{6b}$$

$$b_i = \rho_i f_i, \quad i = 2, 3, \cdots, M. \tag{6c}$$

3. (Calculate completion times) Use Equation 2 to solve for $E(T_i)$ in the order $i = M, M-1, \cdots, 1$.

4. (Update arrival rates) With $E_i = E_1 = 1/\lambda(1 - f_1)$ (from Equations 1 and 4), use Equation 3 to solve for λ_i^* in the order $i = 1, 2, \cdots, M$.

5. (Convergence check) If updated values of λ_i^* show little change (i.e., convergence), then go to Step 6, else go to Step 2.

6. (Calculate occupancy probabilities) Set

$$\Pr\{s_1 = n\} = \frac{(1-\rho_1)\rho_1^n}{1-\rho_1^{N_1+1}}, \quad n = 0, 1, \cdots, N_1, \quad \text{and}$$

$$\Pr\{s_i = n\} = \frac{(1-\rho_i)\rho_i^n}{1-\rho_i^{N_i+2}}, \quad i = 2, 3, \cdots, M \text{ and } n = 0, 1, \cdots, N_i + 1,$$

7. Stop.

Queue 1 is given a slightly different treatment in Steps 2 and 6 because the queue never blocks a previous queue.

To achieve convergence in the iterative procedure, it was sometimes necessary to introduce a damping factor to reduce excessive oscillation in the values of consecutive iterates.

1.5 Computational Results: M/M/1/N Model

The approximation method was tested on three sets of problems in the literature with exponential servers. In these problems, analytic solutions are often not available. In such cases, the literature has used the results of large simulations to replace analytical values. We use these results in our comparisions as previous authors have used them. Table 1 shows the throughput for a 3-queue system each with a buffer of size one as reported in Takahashi et al. [1980]. As can be seen, SIMP (using Equation 2) performs better (relative to the simulation results) than Takahashi's method, and almost as well as Hillier and Boling's (Hillier and Boling [1967]) method which is more complicated than SIMP, which requires only $O(M)$ elementary operations per iteration.

Table 2 shows a comparison with the method of Altiok and Perros (forthcoming) for 3-queue systems with different buffer sizes. Again, SIMP (using Equation 2) performs favorably as compared to the more complicated method.

Finally, Table 3 shows similarly good results for highly unbalanced systems of 3 and 5 queues.

2.0 Residual Service Time Considerations

The above approximation, although it yields suggestively good results, can be improved by relaxing one assumption. Consider relationship (2) and Assumption a5). This represents a correction to the expected service completion time in the event that queue i "sees" server $(i+1)$ blocking it. Assumption a4) allows us to compute the probability of this event as α_i. Strictly speaking, the (random variable) service completion time for server i is the sum of the unblocked service time (with pdf $\mu_i e^{-\mu_i t}$) and the (random variable) length of time that server $(i+1)$ will take to become unblocking, given server i completes service and finds server $(i+1)$ blocking. This second component is the (random variable) residual time for server $(i+1)$ to become unblocking. We now assume

a5′) the unblocked service of the unit in queue i ends at a random point in server $(i+1)$'s service completion period.

This assumption is less restrictive than a5) and it will allow the consideration of general service time distributions (Section 3).

The following relevant result is well known (see, for example Section 5.2 of Cooper [1981]): Let T be a recurrence time with cdf $F_T(t)$ and expectation $E[T]$, and let X be the (random variable) time from a random point in the recurrence interval to the recurrence event, then X (called the residual recurrence time) has the pdf

$$g_X(x) = \frac{1 - F_T(x)}{E[T]}. \tag{7a}$$

As a consequence, it can be shown that the moments of X are given by

$$E[X^n] = \frac{E[T^{n+1}]}{(n+1)E[T]}, \tag{7b}$$

and in particular,

$$E[X] = \frac{E[T^2]}{2E[T]}. \tag{8}$$

To use this result, we note that, if

$T_i \equiv$ (random variable) time for server i to complete service, and

$X_i \equiv$ (random variable) residual time for server i to complete service,

then T_i is the sum of the unblocked service time (with pdf $\mu_i e^{-\mu_i t}$) plus the residual completion time X_{i+1}, conditional on server $(i+1)$ blocking.

Thus, given a5'),

$$E[T_i] = \frac{1}{\mu_i} + \alpha_i E[X_{i+1}], \tag{9}$$

which, by Equation 8, gives

$$E[T_i] = \frac{1}{\mu_i} + \alpha_i \frac{E[T_{i+1}^2]}{2E[T_{i+1}]}, \quad i = M, M-1, \cdots, 1. \tag{10}$$

Under a5'), Equation 10 should be used, in place of Equation 2, to augment the (unblocked) average service time by a term that accounts for blocking. Similarly, $E[T_i]$ in Equation 3 should be replaced by $E[T_i^2]/2E[T_i]$. Note that Equation 10 requires that $E[T_{i+1}^2]$ be available. Indeed, the recursive use of Equation 10, starting with $i = M$, requires

the terms $E[T_i^n]$ for $n = 1, 2, \cdots, i$. The Appendix shows, by a simple transform argument, that these moments are given by:

$$E[T_i^n] = \frac{n!}{\mu_i^n} + \alpha_i \sum_{j=1}^{n} \binom{n}{j} \frac{(n-j)! E[T_{i+1}^{j+1}]}{(j+1)\mu_i^{n-j} E[T_{i+1}]}, \tag{11}$$

or written as a recursion in the index n,

$$E[T_i^n] = \frac{nE[T_i^{n-1}]}{\mu_i} + \frac{\alpha_i E[T_{i+1}^{n+1}]}{(n+1)E[T_{i+1}]}, \quad n = 1, 2, \cdots . \tag{12}$$

Thus, accounting for randomly interrupted residual service time requires replacing Equation 2 of Step 3 in the algorithm with Equation 12 (with $E[T_{M+1}^j] = 0$ for all $j > 1$) and replacing $E[T_i]$ with $E[T_i^2]/2E[T_i]$ in Equation 3 (used in Step 4). By using Equation 12, the complexity of SIMP now becomes $O(M^2)$ per iteration.

The results, using this modification are also given in Tables 1,2 and 3 under the heading "SIMP Eq. (12)". This modification slightly improves the accuracy of the model. Since, in these cases, we have exponential servers that are frequently unblocked the expected residual service time is appoximately equal to the expected service time and hence we do not expect much change due to this modification. However, the results derived in this section are used in Section 3.

3.0 General Service Time Distributions

The above discussion allows us to adapt the approximation method for general service time distributions and eliminate Assumption a3). We are aware of only one analysis in the literature not requiring exponential servers (Gershwin [1983]). Gershwin's approach, however, is restricted to a particular form of non-exponential service. Our method replaces the M/M/1/N analysis in Step 2 (finding b_i and f_i) and Step 6 (calculating occupancy distributions) with an M/G/1/N analysis. This generalization provides wider applicability at the expense of more computation and greater data requirements—as will be seen, it requires the Laplace transform of the service time distribution (and its derivatives) for each queue.

Traditional M/G/1/N analysis (see Section 5.9 of Cooper [1981]) examines the queue at departure epochs. Following this approach, define for each queue i the following (the subscript i is suppressed here for clarity):

$S \equiv$ (random variable) unblocked service time,

$g(t) \equiv$ service completion time probability distribution function,

$\lambda \equiv$ arrival rate,

$\mu \equiv$ service rate (inverse of the first moment of $g(.)$),

$P_n \equiv$ steady-state probability that a departure leaves n units behind,

$a_n \equiv$ probability of n arrivals during a service time, and

$\Pi_n \equiv$ steady-state probability of n units in the queue.

Since the number of units left by a departure is given by the number of arrivals during service and the number left by the previous departure,

$$P_{n+1} = a_0^{-1}(P_n - a_n P_0 - \sum_{j=1}^{n} a_{n-j+1} P_j), \quad n = 0, 1, \cdots, N-1, \tag{13}$$

where $\sum_{j=1}^{0} \equiv 0$. By defining $\delta_n = P_n / P_0$, δ_n can be calculated recursively from (13) after dividing by P_0. Using the δ_n, P_0 can obtained from

$$P_0 = \frac{1}{\sum_{n=0}^{N-1} \delta_n}, \tag{14}$$

and then

$$P_n = \delta_n P_0. \tag{15}$$

P_n is also the steady-state probability that an arrival sees n units in the queue (see Section 5.3 of Cooper [1981]), and is simply the steady-state probability of n units in the queue conditioned on the queue's not being full. This, combined with conservation of flow, gives

$$\Pi_n = \frac{P_n}{P_0 + \rho}, \quad n = 0, 1, \cdots, N-1, \text{ and} \tag{16a}$$

$$\Pi_N = 1 - \frac{1}{P_0 + \rho}, \tag{16b}$$

where $\rho = \lambda / \mu$.

The above equations provide a quick and easy method for obtaining the steady-state probabilities Π_n given the values of a_n. To find the values of a_n, let $f_i(t)$, $h_i(t)$, and $g_i(t)$ be the probability density functions of T_i, S_i, and X_i and $f_i^{*(n)}(s)$, $h_i^{*(n)}(s)$, and $g_i^{*(n)}(s)$ be the n^{th} derivatives of their Laplace transforms. By definition, for the i^{th} queue

$$a_n = \int_0^\infty \frac{(\lambda_i t)^n}{n!} e^{-\lambda_i t} f_i(t) dt,$$

or equivalently,

$$a_n = \frac{(-\lambda_i)^n}{n!} f_i^{*(n)}(\lambda_i). \tag{17}$$

In the Appendix, we show that the n^{th} derivative of the Laplace transform of the service completion time can be recursively obtained from

$$f_i^{*(n)}(s) = (1-\alpha_i) h_i^{*(n)}(s) + \alpha_i \sum_{j=1}^{n} \binom{n}{j} h_i^{*(n-j)}(s) g_{i+1}^{*(j)}(s). \tag{18}$$

The functions $h_i^{*(n)}(s)$ are given as data, but the $g_{i+1}^{*(j)}(s)$ must be evaluated. Taking the Laplace transform and then the j^{th} derivative of both sides of Equation 7*a* (with $X = X_{i+1}$) gives

$$g_{i+1}^{*(j)}(s) = \frac{1}{E[T_{i+1}]} \left(\frac{(-1)^j j!}{s^{j+1}} - \sum_{k=0}^{j} \binom{j}{k} \frac{(-1)^{j-k}(j-k)!}{s^{j-k+1}} f_{i+1}^{*(k)}(s) \right),$$

which is equivalent to the following recursion in the index j,

$$g_{i+1}^{*(j+1)}(s) = -\frac{1}{s} \left((j+1) g_{i+1}^{*(j)}(s) + \frac{f_{i+1}^{*(j+1)}(s)}{E[T_{i+1}]} \right), \tag{19a}$$

where

$$g_{i+1}^{*(0)}(s) = \frac{1 - f_{i+1}^{*}(s)}{s E[T_{i+1}]}. \tag{19b}$$

Since, in the last queue, $f_M^{*(j)}(s) = h_M^{*(j)}(s)$ for all $j \geq 0$, we can recursively solve for $g_{l+1}^{*(j+1)}(\lambda_i)$ and $f_l^{*(n)}(\lambda_i)$ using Equations 19 and 18, in the order $l = M-1, M-2, \cdots, i$. We then use the results in Equation 17.

We also need the first moment $E[T_i]$ to determine the utilization $\rho_i = \lambda_i E[T_i]$ used in Equation 16. The generalization of Equation 11 for general service distributions is (see the Appendix)

$$E[T_l^n] = E[S_l^n] + \alpha_l \sum_{j=1}^{n} \binom{n}{j} \frac{E[S_l^{n-j}] E[T_{l+1}^{j+1}]}{(j+1) E[T_{l+1}]}. \tag{20}$$

By setting $E[T_{M+1}^n] = 0$ for all $n > 0$ and noting that $E[S_l^n] = (-1)^n h_l^{*(n)}(0)$, Equation 20 recursively gives, in the order $l = M, M-1, \cdots, i$, all the moments necessary to calculate ρ_i.

3.1 Finding the Occupancy Distributions: The M/G/1/M Model

The iterative procedure in the case of general service distributions follows.

0. (Setup) Set the values for λ, N_i, and provide a routine to calculate $h_i^{*(j)}(s)$ $(i = 1, 2, \cdots, M)$.

1. (Initialization) Set $b_1 = 0$, $\lambda_1^* = \lambda$, $\lambda_i^* = \lambda(1 - f_i)$ $(i = 2, 3, \cdots, M)$ where f_1 is given by Equation 6*a*, $\bar{n}_1 = N_1 - 1$, and $\bar{n}_i = N_i$ $(i = 2, 3, \cdots, M)$. ($\bar{n}_i$ is the maximum number of arrivals during a service at queue i.)

2. (Calculate moments) With $E[T_M^n] = (-1)^n h_M^{*(n)}(0)$ $(n = 1, 2, \cdots, M)$ and $\alpha_i = (f_{i+1} - b_{i+1})/(1 - b_{i+1})$ use Equation 20 to calculate $E[T_i^n]$ $(n = 1, 2, \cdots, i)$ recursively in the order $i = M-1, M-2, \cdots, 1$.

3. (Find full and blocking probabilities) For each queue $i = 1, 2, \cdots, M$ let $\rho_i = \lambda_i^* E[T_i]$ and perform Steps 3a to 3d.

   3a. Set $f_M^{*(n)}(\lambda_i^*) = h_M^{*(n)}(\lambda_i^*)$ for $n = 0, 1, \cdots, \bar{n}_i$. If $i = M$ then go to Step 3d.

   3b. In the order $l = M-1, M-2, \cdots, i$ use Equations 19 and 18 to calculate $g_{l+1}^{*(n)}(\lambda_i^*)$ and $f_l^{*(n)}(\lambda_i^*)$ for $n = 0, 1, \cdots, \bar{n}_i$.

   3c. Calculate a_n $(n = 0, 1, \cdots, \bar{n}_i)$ from Equation 17.

   3d. Set $N = \bar{n}_i + 1$ and use Equations 13 to 16 to solve for Π_N and Π_{N-1}. If $i = 1$ then set $f_1 = \Pi_N$, else set $b_i = \Pi_N$ and $f_i = \Pi_N + \Pi_{N-1}$.

4. (Update arrival rates) With $E_i = E_1 = 1/\lambda(1 - f_1)$ (from Equations 1 and 4), use Equation 3 with $E[T_i]$ replaced by $E[T_i^2]/2E[T_i]$ to solve for λ_i^* in the order $i = 1, 2, \cdots, M$.

5. (Convergence check) If the updated values of λ_i^* show little change (i.e., convergence), then go to Step 6, else go to Step 2.

6. (Calculate occupancy probabilities) For each queue $i = 1, 2, \cdots, M$, repeat Steps 3a to

3d (giving Π_n) and set

$$\Pr\{s_i = n\} = \Pi_n, \quad n = 0, 1, \cdots, \bar{n}_i + 1.$$

7. Stop.

3.2 Computational Results: The M/G/1/N Model

The SIMP M/M/1/N and M/G/1/N Model results are compared with the only available non-exponential service results in the literature: Gershwin [1983]. The comparsion is summarized in Table 4. The service time at queue i in Gershwin's model is one period with probability $1 - p_i$ and n periods with probability $p_i r_i^n$, $n = 1, 2, \cdots$. The server of the first queue is never idle. We achieved this requirement by giving the first queue a buffer of 4 and making the arrival rate large compared to the average service time. In Table 4, $\underline{B}$ is the vector of buffer sizes, and $\underline{p}$ and $\underline{r}$ are the service time parameters (p_i, r_i) for each queue. Again, simulation values replace exact values when analytical results are not available. We note that the Gershwin model requires, amoung other things, the solution of a nonlinear equation in one unknown in each iteration whereas we have only $O(M^3 + \sum_{i=1}^{M} N_i^2)$ computations per iteration (from equation 20 and M/G/1/N analysis). Note also that the form of Gershwin's service distribution has a high variance, which leads to occasional, but very long blocking delays, a phenomenon the M/M/1/N model would miss. The lower throughput values for the M/G/1/N versus the M/M/1/N analysis indicate an improvement in this area. The results in Table 4 are encouraging given the generality of the SIMP model relative to the specific model of Gershwin and given the ease of SIMP's computations.

4.0 Conclusions

We have presented an extremely simple-minded, yet fast and apparently accurate approximation method for analyzing open tandem queues with blocking. The ability to incorporate non-exponential service times presents an opportunity to study the behavior of more realistic systems than those considered previously, without the need to develop special procedures for idiosyncratic service distributions.

The conclusions about accuracy, of course, rest upon the evidence obtained to date of the approach's ability to produce, within acceptable percent differences, performance measures of interest for previously analyzed systems. It remains to be seen whether we can guarantee error bounds, or provide explicit advice on parameter values for which the approximation is unequivocally recommended. However, due to the simplicity of the approximation, its straight-forward structure, and its ready use for general service distributions, it holds promise to be a useful tool in the study of systems of queues. In particular, we are currently engaged in extending the general approach to closed tandem queues, and more general open networks.

Appendix. Computation of Completion Time Moments

T_i, the completion time for the i^{th} server, is the weighted sum of two random variables:

$S_i \equiv$ (unblocked) service time for server i, and

$X_i \equiv$ residual completion time for a randomly encountered (busy) server i,

the latter weighted by $\alpha_i = \Pr\{\text{server } i \text{ will encounter a "block" from server } i+1\}$.

Let $f_i(t)$, $h_i(t)$, and $g_i(t)$ be the probability density functions, and $f_i^*(s)$, $h_i^*(s)$, and $g_i^*(s)$ be the Laplace transforms, for the random variables T_i, S_i, and X_i, respectively. Then

$$f_i^*(s) = (1-\alpha_i)h_i^*(s) + \alpha_i h_i^*(s)g_i^*(s). \qquad (A1)$$

From the moment-generating property of the Laplace transform, the n^{th} moment of T_i is found by taking the n^{th} derivative of $f_i^*(s)$ and evaluating it at $s = 0$. Taking the n^{th} derivative of both sides of $(A1)$ (where $f^{(n)}(s) \equiv \frac{d^n f(s)}{ds^n}$) gives

$$f_i^{*(n)}(s) = (1-\alpha_i)h_i^{*(n)}(s) + \alpha_i \sum_{j=1}^{n} \binom{n}{j} h_i^{*(n-j)}(s)g_{i+1}^{(j)}(s). \qquad (A2)$$

Since X_i is the residual time for the random variable T_i, we have from Equation 7*b*

$$g_{i+1}^{*(n)}(s)\Big|_{s=0} = E[X_{i+1}^n] = \frac{E[T_{i+1}^{n+1}]}{(n+1)E[T_{i+1}]}.$$

Thus, evaluating both sides of $(A2)$ at $s = 0$ gives

$$E[T_i^n] = (1-\alpha_i)E[S_i^n] + \alpha_i \sum_{j=0}^{n} \binom{n}{j} \frac{E[S_i^{n-j}]E[T_{i+1}^{j+1}]}{(j+1)E[T_{i+1}]}. \qquad (A3)$$

Since the $j = 0$ term in the sum is simply $E[S_i^n]$, Equation $A3$ can also be written as Equation 20. In the special case where the S_i are exponentially distributed with rate μ_i, then $E[S_i^n] = \frac{n!}{\mu_i^n}$ and $A3$ reduces to Equation 11.

TABLE 1

Throughput Comparisons (from Takahashi et al. 1980)

μ_1	μ_2	μ_3	Simul-ation	Hillier-Boling	Takahashi et al.	SIMP Eq. (2)	SIMP Eq. (12)	SIMP M/G/1
1.1	1.2	1.3	.709	.712	.645	.689	.691	.694
1.2	1.4	1.6	.762	.767	.706	.746	.748	.751
1.3	1.6	1.9	.798	.807	.755	.790	.790	.793
1.4	1.8	2.2	.828	.837	.793	.823	.824	.827
1.5	2.0	2.5	.855	.861	.824	.850	.850	.853
1.6	2.2	2.8	.877	.880	.849	.871	.872	.874
1.7	2.4	3.1	.889	.896	.870	.889	.889	.891
1.8	2.6	3.4	.902	.909	.887	.904	.904	.906
1.9	2.8	3.7	.915	.920	.901	.915	.915	.917
2.0	3.0	4.0	.929	.929	.913	.925	.926	.927
Max. Abs. Deviation			.000	.009	.064	.020	.018	.015
Ave. Abs. Deviation			.000	.005	.032	.007	.006	.005
Ave. Abs. % Deviation			.000	0.6	4.0	0.8	0.8	0.6

Arrivals to queue 1 constitute a Poisson process with rate 1; $B_i = 1$ except $B_1 = 2$.

TABLE 2

Comparisons with the Exact Solution and the Approximation of Altiok and Perros

Case	Measure	Exact Solution	Altiok and Perros	SIMP Eq. (2)	SIMP Eq. (12)	SIMP M/G/1/N
$(B_1,B_2,B_3)=(\infty,2,2)$	$P_1(0)$	.281	.299	.298	.299	.299
$(\mu_1,\mu_2,\mu_3)=(2,2,2)$	$P_2(0)$	.326	.331	.331	.331	.331
$\lambda=1.2$	$P_2(3)$	.262	.237	.239	.239	.237
	$P_3(0)$	.400	.400	.400	.400	.400
	$P_3(3)$	.182	.176	.176	.176	.176
$(B_1,B_2,B_3)=(\infty,1,1)$	$P_1(0)$	.166	.173	.167	.175	.175
$(\mu_1,\mu_2,\mu_3)=(2,2,2)$	$P_2(0)$	.267	.268	.268	.268	.268
$\lambda=1.2$	$P_2(2)$	.483	.474	.477	.476	.472
	$P_3(0)$	.400	.400	.400	.400	.400
	$P_3(2)$	.323	.323	.323	.323	.323
$(B_1,B_2,B_3)=(\infty,1,1)$	$P_1(0)$	.386	.402	.399	.402	.402
$(\mu_1,\mu_2,\mu_3)=(3,2.5,3.5)$	$P_2(0)$	.396	.396	.396	.396	.396
$\lambda=1.4$	$P_2(2)$	.338	.325	.328	.327	.325
	$P_3(0)$	.600	.600	.600	.600	.600
	$P_3(2)$	.149	.149	.149	.149	.149
$(B_1,B_2,B_3)=(1,1,1)$	$P_1(0)$	.118	.127	.131	.131	.128
$(\mu_1,\mu_2,\mu_3)=(2,2,2)$	$P_1(2)$	.574	.588	.591	.589	.588
$\lambda=3$	$P_2(0)$	.213	.239	.245	.240	.239
	$P_2(2)$	.524	.512	.507	.513	.512
	$P_3(0)$	.361	.382	.386	.383	.382
	$P_3(2)$	.356	.342	.338	.341	.343
$(B_1,B_2,B_3)=(1,1,1)$	$P_1(0)$	.209	.223	.224	.222	.222
$(\mu_1,\mu_2,\mu_3)=(3,4,2)$	$P_1(2)$	.451	.459	.456	.459	.459
$\lambda=3$	$P_2(0)$	.243	.274	.268	.276	.274
	$P_2(2)$	.503	.473	.476	.467	.474
	$P_3(0)$	.176	.188	.185	.189	.189
	$P_3(2)$	.608	.589	.593	.586	.587
Max. Abs. Deviation		.000	.031	.032	.036	.031
Ave. Abs. Deviation		.000	.011	.011	.012	.012
Ave. Abs. % Deviation		.000	4.0	3.9	4.2	4.0

TABLE 3

Throughput Comparisons with Simulation and the Approximation of Altiok and Perros

λ / $\underline{B}$ / $\underline{\mu}$	Simulation (Exact)	Altiok and Perros	SIMP Eq. (2)	SIMP Eq. (12)	SIMP M/G/1/N
1.5 (1,1,2) (2,3,1)	.916	.851	.939	.930	.928
2 (3,2,1) (4,3,2)	1.780	1.553	1.669	1.672	1.668
2 (1,1,2) (2,3,1)	.954	.951	.965	.959	.959
3 (4,3,2,1,1) (2,2,2,2,2)	1.277	1.196	1.248	1.262	1.265
3 (1,1,2,2,2) (3,3,2,2,2)	1.438	1.357	1.400	1.405	1.407
2.5 (1,2,3,2,1) (2,2,2,2,2)	1.295	1.113	1.259	1.263	1.263
3 (3,3,2,2,1) (2,3,4,3,2)	1.647	1.399	1.682	1.681	1.682
2 (1,2,3,2,1) (2,2,2,2,2)	(1.294)	1.344	1.207	1.208	1.208
2 (1,1,1,1,1) (2,2,2,2,2)	(1.131)	1.086	1.079	1.088	1.086
Max. Abs. Deviation	.000	.248	.111	.108	.112
Ave. Abs. Deviation	.000	.109	.046	.041	.041
Ave. Abs. % Deviation	.000	7.7	3.4	3.0	2.9

TABLE 4

Throughput Comparisons with the Approximation of Gershwin (1983)

$\frac{B}{p}$ r	Simulation (Exact)	Gershwin	SIMP Eq. (2)	SIMP Eq. (12)	SIMP M/G/1/N
(4,5,5) All .002 All .1	(.4545)	.4542	.5519	.5525	.4881
(4,5,5) All .002 All .05	(.8993)	.8993	.7980	.7969	.7591
(4,5,5) All .1 All .1	(.3109)	.3105	.4139	.4144	.3531
(4,10,10) All .1 All .1	(.3556)	.3563	.4469	.4466	.3939
(4,10,10) (.01,.013,.007) (.07,.1,.05)	(.7539)	.7546	.7864	.7866	.7512
(4,5,5,5) All .05 All .45	.8153	.8716	.7232	.7243	.7716
(4,10,10,10) All .1 All .1	.3364	.3352	.4393	.4395	.3786
(4,10,10,10,10) All .1 All .1	.3204	.3228	.4351	.4354	.3697
$B_1 = 4, B_i = 10 (i = 2, 3, \cdots, 12)$ (.01,.013,.007,.01,.013,.007, .01,.013,.007,.01,.013,.007) (.07,.1,.05,.07,.1,.05, .07,.1,.05,.07,.1,.05)	.5916	.6098	.7494	.7495	.6892
$B_1 = 4, B_i = 10 (i = 2, 3, \cdots, 20)$ All .1 All .1	.2767	.2924	.4243	.4243	.3498
(4,40,40,40) All .005 All .05	.8362	.8337	.8754	.8754	.8484
Max. Abs. Deviation	.0000	.0182	.1578	.1579	.1402
Ave. Abs. Deviation	.0000	.0040	.0986	.0983	.0523
Ave. Abs. % Deviation	.0000	.52	23.52	23.51	11.40

The service time at queue i is 1 period with probability $1 - p_i$ and n periods with probability $p_i r_i^n$.

References

1. Altiok, T., (1982) "Approximate analysis of exponential tandem queues with blocking", *European Journal of Operations Research* 11, 390-398.

2. Altiok, T., and Stidham, S., (1982) "A note on transfer lines with unreliable machines, random processing times and finite buffers", *AIIE Transactions* 14, 125-127.

3. Asare, B. K., (1978) "Queue networks with blocking", Ph.D. dissertation, Trinity College, Dublin, Ireland.

4. Boxma, O. J., and Konheim, A. G., (1981) "Approximate analysis of exponential queueing systems with blocking", *Acta Informatica* 15, 19-66.

5. Buzacott, J. A., (1967) "Markov chain analysis of automatic transfer lines with buffer stock", Ph.D. dissertation, Department of Engineering Production, University of Birmingham.

6. Caseau, P., and Pujolle, G., (1979) "Throughput capacity of a sequence of queues with blocking due to finite waiting room", *IEEE Transactions on Software Engineering* SE-5, 631-642.

7. Cooper, R. B., (1981) *Introduction to Queueing Theory.* 2^{nd} ed. North-Holland, New York.

8. de Souza e Silva, E. Lavenberg, S. S., and Munty, R.R., (1983), "A perspective on iterative methods for the approximate analysis of closed queueing networks", IBM Report RC 10141.

9. Foster, F. G., and Perros, H. G., (1980) "On the blocking process in queue networks", *European Journal of Operations Research* 5, 276-283.

10. Gershwin, S. B., (1983) "An efficient decomposition method for the approximate evaluation of tandem queues with finite storage and blocking", manuscript, Laboratory for Information and Decision Sciences, MIT.

11. Gordon, W. J., and Newell, G. F., (1967) "Cyclic queueing systems with restricted length queues", *Operations Research* 15, 266-278.

12. Hillier, F. S., and Boling, R., (1967) "Finite queues in series with exponential or Erlang service times—A numerical approach", *Operations Research* 15, 286-303.

13. Hordijk, A., and Van Dijk, N., (1981) "Networks of queues with blocking", *Performance* '81, F. J. Kylstra (Ed.,), North-Holland, New York.

14. Konheim, A. G., and Reiser, M., (1976) "A queueing model with finite waiting room and blocking", *Association for Computing Machinery* 23, 328-341.

15. Labetoulle, J., and Pujolle, G., (1977) "Modelling of packet switching communication networks with finite buffer size at each node", *Computer Performance*, Chandy and Reiser (Eds.), North-Holland 515-535.

16. Latouche, G., and Neuts, M. F., (1980) "Efficient algorithmic solutions to exponential tandem queues with blocking", *SIAM Journal of Algebraic Discrete Methods* 1, 93-106.

17. Marchal, W. G., (1984) "Numerical Performance of Approximate Queueing Formulas with Application to Flexible Manufacturing Systems", University of Toledo.

18. Perros, H. G., (1981) "A symmetrical exponential open queue network with blocking and feedback", *IEEE Transactions on Software Engineering* Vol. SE-7, 395-402.

19. Perros, H. G., (1984) "Queueing networks with blocking: A bibliography". Performance Evaluation Review, *ACM SIGMETRICS* 12, 8-12.

20. Perros, H. G., and Altiok, T., "Approximate analysis of open networks of queues with blocking: Tandem configurations", *IEEE Transactions on Software Engineering* (forthcoming).

21. Pinedo, M., and Wolf, R. W., (1982) "A Comparison between tandem queues with dependent and independent service times", *Operations Research* 30, 464-479.

23. Suri, R., and Diehl, G. W., (1983) "A variable buffer-size model and its use in analytic closed queueing networks with blocking", manuscript, Division of Applied Science, Harvard University.

23. Takahashi, Y., Miyahara, H., and Hasegawa, T., (1980) "An approximation method for open restricted queueing networks", *Operations Research* 28, 594-602.

24. Towsley, D., (1979) "Queueing network models with state-dependent routing", *Association for Computing Machinery* 27, 323-337.