THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 59, No. 9 November 1980
Printed in U.S.A.

Approximations for Customer-Viewed Delays in Multiprogrammed, Transaction-Oriented Computer Systems

By A. A. FREDERICKS

(Manuscript received April 22, 1980)

A common method of insuring the efficient use of resources in a computer system is to limit the degree of multiprogramming, i.e., the number of jobs allowed in the system at the same time. Many systems vary the multiprogramming level dynamically, while others fix the level a priori. This paper analyzes a transaction-oriented computer system with a fixed level of multiprogramming. We are primarily concerned with the performance of such systems as viewed by the customers submitting the transactions. While the common technique of approximating such systems with a state-dependent server model is often adequate, in some situations it can lead to large errors. We introduce the concept of the mean forward recurrence time of the output process under saturation and we show that errors that occur in using a state-dependent server model are directly related to this mean forward recurrence time. We give an improved approximation technique using this quantity and show that its accuracy depends on the dominant time constant (relaxation time) of the computer system. The variability in the output process of the computer system that is implied by a large mean forward recurrence time (relative to the mean interdeparture time) can have broad implications on the analysis and design of computer systems, particularly with a fixed degree of multiprogramming. Thus, even though estimation of this mean forward recurrence time may be difficult (although readily measured) in complex systems, a knowledge of this quantity may be necessary to achieve a reasonable degree of confidence in performance models.

I. INTRODUCTION

The analysis of a multiprogramming computer system is greatly complicated by the requirement of an external queue to limit the level of multiprogramming (see Fig. 1a), and one is often led to approximation techniques. One common method for finding customer-experienced delays (such as the mean access time or mean response time) is to solve for the mean output rate from a closed (generally Markovian) model of the computer system with a fixed number of jobs resident equal to $n = 1, 2, \cdots, M$ (M is the maximum allowable multiprogramming level—see Fig. 1b) and then approximate the entire system by a (Markovian) state-dependent server process (see Fig. 1c) which can be analyzed for the quantities of interest.

The accuracy of this approximation technique has been studied by Avi-Itzhak and Heyman[1] for a certain class of models, and some specific error bounds are given there. Konheim and Reiser also consider briefly the accuracy question in Ref. 2, where they give an exact solution for a two-resource computer system. The conditions given to insure the accuracy of the state-dependent server approximation are essentially the same as those needed for decomposibility in the sense of Courtois[3,4] as applied to the special class of systems considered in Refs. 1 and 2. While the state-dependent server model is often appli-

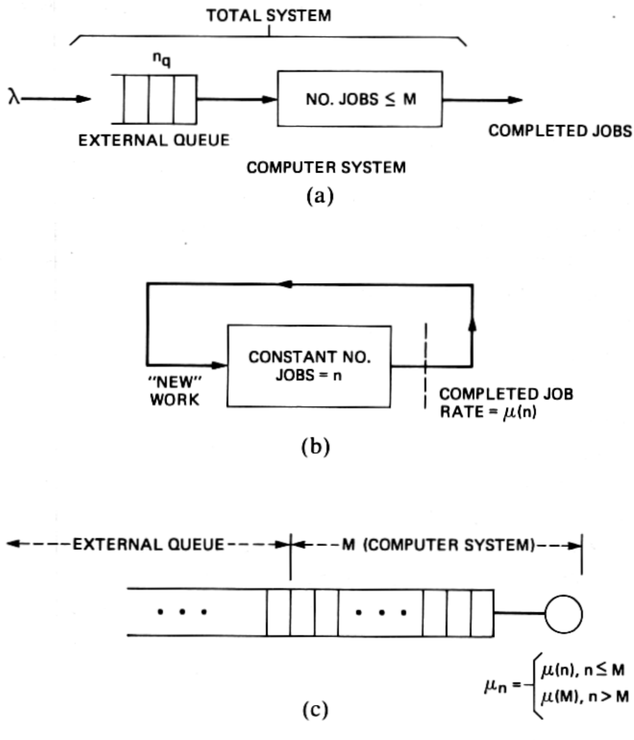


Fig. 1—Modeling a multiprogrammed computer system. (a) Computer system with fixed multiprogramming level. (b) Closed (Markovian) network. (c) (Markovian) state-dependent server model.

cable, the magnitude of the errors that can result[1,2] is disconcerting since, as we shall see, the verification of necessary conditions for the applicability of this method to a complex system can be extremely difficult.

In this paper, we look in detail at the underlying cause of these errors and give an improved characterization of the applicability of the state-dependent server model. We introduce the concept of the mean forward recurrence time, R_M, of the output process of a multiprogrammed computer system under saturation, and show that the accuracy of the state-dependent approximation technique is explicitly related to this quantity. This leads us to a new method of approximating the behavior of such systems that captures the essence of performance as viewed by the customer and that can be applied under much weaker conditions than are needed to insure the accuracy of the state-dependent model. Our results reduce to those for the state-dependent server model where the latter is applicable.

The application of the method is illustrated by considering the system studied in Ref. 2. The accuracy of the method is studied in detail, including quantitative consideration for its applicability. A summary of some key results is included in the next section, but before proceeding, we note that the concepts discussed here have more general implications for the analysis and design of a variety of computer systems. We discuss this point briefly in the last section.

II. SUMMARY OF KEY RESULTS

In Section IV, the following approximation for the mean waiting time $\bar{w}_q$* in the external queue as seen by a (Poisson) arrival is developed:

$$\bar{w}_q \approx \bar{w}_q(A) = \frac{R_M P_M}{1 - \lambda\, S(M)}, \tag{1}$$

where

R_M = mean forward recurrence time of the interdeparture process under saturation (M in the computer system), i.e., the time from an arbitrary time point (Poisson arrival) until the next departure.

P_M = probability the system is saturated

$S(M)$ = mean interdeparture time under saturation.

This approximation for $\bar{w}_q$ is "derived" under the assumption that the system is "always" in equilibrium. In general, its applicability depends

* Throughout this paper, the notation $\bar{z}$ denotes the expected value of the random variable z.

on the rate at which the system approaches equilibrium and the magnitude of the disturbances from equilibrium. However, it is much more accurate than the state-dependent server model which, in addition to the equilibrium condition noted above, essentially assumes that $R_M = S(M)$. Indeed, we show that

$$\frac{\bar{w}_q(\text{exact})}{\bar{w}_q(\text{state-dependent})} = \frac{\bar{n}_q(\text{exact})}{\bar{n}_q(\text{state-dependent})} \approx \frac{R_M}{S(M)},$$

where $\bar{n}_q$ is the mean number in the external queue; the equality follows from Little's law.

This is clearly shown on Table II where, in addition, we see that the magnitude of the error is such that the exact value can differ from that predicted by the state-dependent server model by a factor of 10. Also shown on Table II, for comparison, is the same ratio for our approximation. We see generally excellent agreement, particularly as compared to the state-dependent server model. Moreover, a sufficient condition given for the applicability of our approximation is that the dominant time constant ($T(2)$ on the table) of the closed network be small compared to the interarrival time, λ^{-1}. The correlation of the accuracy of our approximation with $T(2)$ is apparent.

It could be argued that, for the parameter values that led to the large errors, the system clearly violated the conditions given in Refs. 1 to 4 for the applicability of the state-dependent server model. However, as we note, for more complex systems the violation of these conditions could be difficult to detect. Hence, the approach taken here may not only provide an improved approximation method, but may also be a necessary approach to insure the adequacy of performance prediction models. These points are discussed further in Section VI.

III. APPLICABILITY OF THE STATE-DEPENDENT SERVER MODEL

We begin by considering the following simple model of a computer system (see Fig. 2a). There are two system resources, e.g. a processor and a disk. Jobs that enter the system consist of a (random) number of alternating requests for processor and disk. More explicitly, after an (exponential) service at the processor of mean length s_1, the job either terminates with probability ϕ, or it enters the disk queue to be given an (exponential) service of mean length s_2. At the completion of its disk service, the job reenters the processor queue, etc. If n_1 is the number at the processor (queue plus server) and n_2 the number at the disk (queue plus server), then we enforce a maximum multiprogramming level of M by requiring that $n_1 + n_2 \leq M$. If, on arrival, a job finds $n_1 + n_2 = M$, then it is queued externally and enters the computer system (instantaneously) *when* there is room. We denote the number

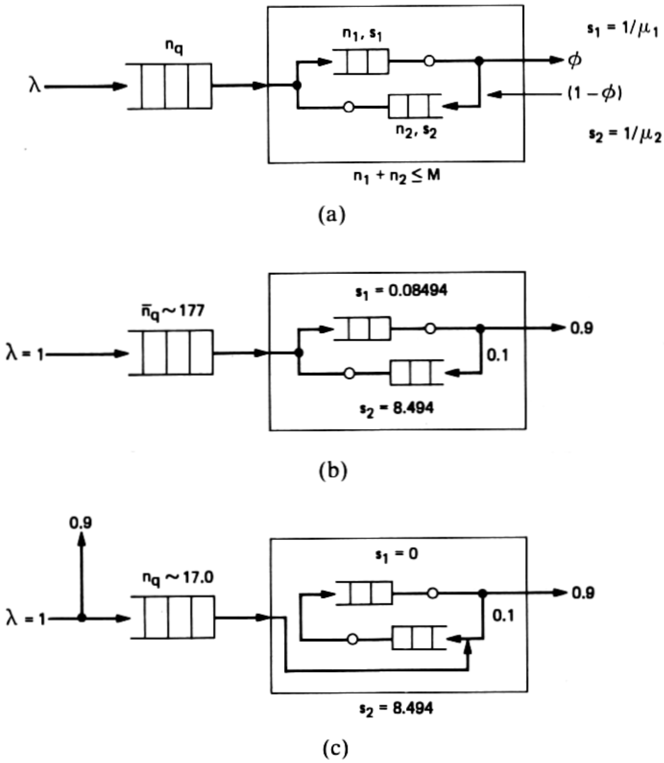


Fig. 2—A simple multiprogrammed computer system. (a) Basic system. (b) Specific parameter system. (c) Disk only model.

at this external queue by n_q and further assume that the arrival process is Poisson.

This system is studied in Ref. 2 where an algorithm is given for obtaining the (exact) solution. An interesting interpretation of the state-dependent server model as a limiting case of this model leads Konheim and Reiser[2] to conclude that the important accuracy parameter is ϕ, i.e., small ϕ implies small errors. This is completely consistent with the error analysis given in Ref. 1. Moreover, if ϕ is small, then clearly *both* $\mu_1 = 1/s_1$ and $\mu_2 = 1/s_2$ must be large compared to the mean arrival rate λ, i.e., the conditions for decomposibility in the sense of Courtois.[3,4]

Figure 3 shows the percent error made in using the state-dependent server model to approximate the mean waiting time for the system of Fig. 2a for $M = 2$. (The exact results were taken from Ref. 2.) We see that, indeed, the error increases with increasing ϕ; however, we see that there is also a very strong dependence on $r = \mu_2/(1 - \phi)\mu_1$, which is a measure of the balance in the system. Thus, apparently, neither ϕ nor r alone can totally explain the errors. To obtain some insight into the underlying cause, we consider briefly a specific example. Figure 2b shows the basic system under consideration with the parameter values that correspond to the maximum error on Fig. 3. As indicated, the true mean external queue length ($\bar{n}_q$) is about 177, while the state-dependent model predicts about 18. (Note that, while the relative error is about −90 percent, the true mean queue size as seen by arrivals is a factor of 10 greater than predicted—a catastrophic error for system performance prediction.)

To gain some insight into the underlying cause of this large error, consider the following example. First, we assume that $s_1 = 0$. Second, we "thin" the stream at arrival epochs to the total system by removing jobs that will never need the disk; thus, the input to this system is just $(1 - \phi)\lambda$ and the system consists essentially of just a disk (see Fig. 2c). The equilibrium state distribution for this new system is equivalent to that for an M/M/1 queuing system with intensity $\hat{\lambda} = (1 - \phi)\lambda$, and service rate $\hat{\mu} = \phi\mu_2$ (this can be readily seen by writing the system's birth-death equations). The mean number in the total system is then given by $\hat{N}_S = \hat{\rho}/(1 - \hat{\rho})$, where $\hat{\rho} = \hat{\lambda}/\hat{\mu}$. Using $\lambda = 1$, $\phi = 0.9$ and $\mu_2 = 1/s_2 = 0.118$ (the parameter values that lead to the value of $\bar{n}_q = 177$ noted above), we find that $\hat{N}_S = 18.9$ and also $\hat{n}_q = 17.0$. Now for this same system, with $s_1 \equiv 0$, if we retained all customers we would expect to see the same number of customers in the external queue that needed disk requests but, on the average, 9 customers that did not need the

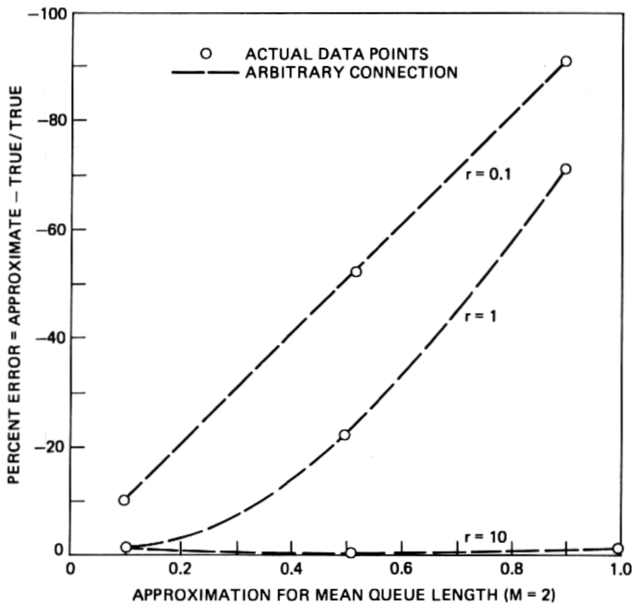


Fig. 3—Accuracy of the state-dependent server model.

disk (and hence have 0 service time) for each customer that did. Thus we are led to a value of $n_q^{\bar{B}} = 170$ in the external queue. If $s_1 > 0$, clearly $\bar{n}_q > n_q^{\bar{B}}$. Thus this simple "bottleneck" analysis tells us that $\bar{n}_q$ must be greater than 170, yet the state-dependent server model tells us it is 18.

The phenomenon not captured by the state-dependent server model is that, by keeping a maximum value of $M = 2$ on the number in the comptuer system, whenever two jobs are at the disk, we have no potential for job completions for a relativley long time, and hence the buildup of a large queue. That is, a relatively small number of heavy system resource users are delaying a larger number of light system resource users, i.e., "processor sharing" cannot help small jobs if they are not let into the computer system.

It is somewhat disconcerting that the state-dependent server model does so poorly, since the occurrence of this phenomenon may be masked in a complex computer system. For example, the same phenomenon would occur if we had a multiclass system in which the majority of the customers belonged to classes which required little system resources, but a small minority of customers required extensive use of system resources. The result is that occasionally the heavy resource users saturate the system causing long delays. This is further compounded by the well-known length biasing effect familiar in the analysis of the M/G/1 queue, i.e., customers are more likely to arrive during long service times—the *variability* in the service time distribution impacts on customer-viewed delays.

Knowing the cause of the large errors, we now turn our attention to an approach that does not suffer from the same problem.

IV. AN IMPROVED APPROXIMATION FOR CUSTOMER VIEWED DELAYS

We first consider the case $M = 1$. Here we note that the total system is *exactly* described by an M/G/1 queuing system and hence the mean number in the "external" queue is just given by the Pollaczek-Khinchin formula

$$\bar{n}_q = \frac{\lambda RP}{1 - \lambda S},$$

where P is the probability the server is busy, S is the mean service time, and R is the mean forward recurrence time of the service time process.

Now consider the more general multiprogramming system of Fig. 1a. Let $\bar{w}_q$ denote the mean delay in the external queue as seen by arrivals; then, by Little's law we have

$$\bar{n}_q = \lambda \bar{w}_q, \tag{2}$$

where

$$\bar{n}_q = \text{mean number in the external queue}$$

$$\lambda = \text{mean (Poisson) arrival rate.}$$

We develop another relation for $\bar{w}_q$ which, when combined with (2), will allow us to solve for $\bar{w}_q$. For this prupose, we define the following:

P_M—the probability that an arrival finds M in the computer system, i.e., the probability that an arrival will be delayed in the external queue.

$S(M)$—the mean interdeparture time when there are M in the computer system.

R_M—the mean forward recurrence time of the interdeparture process when there are M in the computer system.

Following the derivation for $\bar{w}_q$ in an M/G/1 system (e.g., See Ref. 5), we have

$$\bar{w}_q = E\{w_q/w_q > 0\}P(w_q > 0) = E\{w_q/w_q > 0\}P_M. \tag{3}$$

If we assumed that the interdeparture process under saturation (i.e., M in the computer system) were approximately a renewal process, then we would also have that

$$E\{w_q/w_q > 0\} \approx R_M + E\{n_q/w_q > 0\}S(M). \tag{4}$$

Our approximation method is to assume that (4) holds for the system under consideration.

If the closed computer system can be represented by a Markovian queuing network, then a sufficient condition for this assumption to hold is that the system reach equilibrium "instantaneously" when it becomes saturated, i.e., when an arrival finds $M - 1$ in the system. (The Markovian property will ensure that the system can be described by the equilibrium state distribution for the remainder of the saturation period, i.e., until a departure leaves $M - 1$ in the system.) We would expect (4) to be approximately true if the majority of the customers arriving during saturation saw the system in equilibrium, i.e., if either the time to reach equilibrium upon saturation were small compared to the interarrival time, λ^{-1}, or if the total number served during a saturation period were large compared to the number that are served before equilibrium is reached. Note that these assumptions are *necessary* for the applicability of the state-dependent server model, but clearly not *sufficient* as evidenced by the M/G/1 example noted above.

Assuming the validity of (4), (3) now yields

$$\bar{w}_q \approx R_M P_M + \bar{n}_q S(M). \tag{5}$$

Combining (2) and (5), we obtain

$$\bar{w}_q \approx \bar{w}_q(A) = \frac{R_M P_M}{(1 - \lambda S(M))} \tag{6}$$

or

$$\bar{n}_q \approx \bar{n}_q(A) = \frac{\lambda R_M P_M}{(1 - \lambda S(M))}. \tag{7}$$

Now, given an arbitrary multiprogramming computer system, we can compute (or at least approximate) $S(M)$, P_M, and R_M, and hence use (6) or (7) as an approximation. In the next section, we show how to obtain the quantities needed for this approximation method for the class of systems given by Fig. 2a and compare the results with the exact solution obtained from Ref. 2. Before proceeding, we note that the use of the state-dependent server model is equivalent to making our equilibrium assumption *and* assuming that $R_M = S(M)$. Thus, if we use $S(M)$ and P_M as given by the state-dependent server model and *any* improved (over $S(M)$) estimate of R_M, we will most likely obtain an improved approximation to $\bar{w}_q(\bar{n}_q)$. Indeed, one would expect the ratio of the exact value of $\bar{n}_q$ to the state-dependent server model value to be on the order of R_M/S_M— a quantity which, as we shall see, can be quite large.

V. ILLUSTRATION OF THE METHOD—AN EXAMPLE

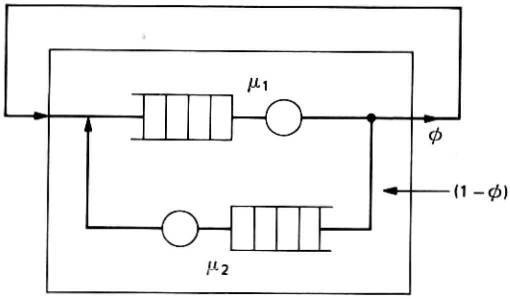
In this section, we study the application of our approximation method to the system of Fig. 2a. We begin by studying the closed system, and in particular its transient behavior, to determine the validity of our assumptions. Next we show how we can approximate the quantities needed for eqs. (6) and (7), in particular, R_M, and finally we give some numerical results comparing our approximation with the exact results from Ref. 2 and results obtained via the state dependent server model.

5.1 Behavior of closed system

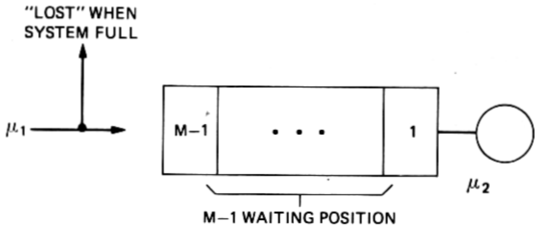
We first note that the system under study is equivalent to a single queuing station with a finite waiting room (see Fig. 4). (Note that this fact has been used by Reiser and Kobayashi[6] to study the impact of a nonexponential server in a closed queuing network.) The equilibrium state probabilities for our closed system are given by

$$P_{n_1} = \frac{(1-r)r^{n_1}}{(1-r^{M+1})} \qquad r \neq 1$$

$$= \frac{1}{M+1} \qquad r = 1, \tag{8}$$



(a)



(b)

Fig. 4—Equivalent closed and open systems. (a) Closed system. (b) Open system.

where

M = multiprogramming level

n_1 = number at processor ($n_2 = M - n_1$ at disk)

$$r = \frac{\mu_2}{(1-\phi)\mu_1}.$$

In particular, note that

$$P_0 = \frac{1}{\sum_{i=0}^{M} r^i}. \tag{9}$$

For this system, the transition probabilities $P_{ij}(t) = P\{j$ at processor at time t given i at processor at time $0\}$ are also known[7] and are given by

$$P_{ij}(t) = \frac{(1-r)r^j}{1-r^{M+1}} + \frac{2}{M+1} r^{1+(j-i)/2} \sum_{k=1}^{M} \frac{F_{ik}F_{jk}\, e^{-(1-\phi)\mu_1 t f_k}}{f_k} \tag{10}$$

where

$$F_{ik} = \sin(i+1)\theta_k - r^{-1/2}\sin i\theta_k$$

$$f_k = 1 + r - 2\sqrt{r}\cos\theta_k$$

and

$$\theta_k = \frac{\pi k}{(M+1)}, \; k = 1, 2, \cdots M.$$

In particular, the dominant time constant,* $T(M)$, is given by

$$\begin{aligned} T(M)^{-1} &= (1-\phi)\mu_1 f_1 \\ &= (1-\phi)\mu_1 + \mu_2 - 2\sqrt{(1-\phi)\mu_1\mu_2}\cos\frac{\pi}{M+1} \end{aligned} \tag{11}$$

Thus if $T(M)$ is small compared to λ^{-1}, one would expect that our equilibrium conditions would be met and suitable values for $S(M)$, R_M, and P_M could be found.

Note from (11) we have

$$\begin{aligned} T(1)^{-1} &= (1-\phi)\mu_1 + \mu_2 \\ T(M)^{-1}_{M\to\infty} &= (((1-\phi)\mu_1)^{1/2} - \mu_2^{1/2})^2. \end{aligned} \tag{12}$$

Hence, in general, we see that if *either* μ_1 or μ_2 is large, we tend to reach equilibrium rapidly and our approximation method should be good. However, we also see that the rate at which equilibrium is approached tends to decrease as M increases and, for M large, a balanced (closed) system ($(1-\phi)\mu_1 = \mu_2$) can result in a very slow approach to equilibrium.

5.2 Determination of needed quantities

We assume in this section that (4) holds exactly and, in particular, that the system has the "instantaneous" equilibrium property noted above.

5.2.1 Determination of S(M)

Under our equilibrium assumption, $S(M)$ is given by

$$S(M) = [\phi\mu_1(1-P_0)]^{-1}, \tag{13}$$

i.e., the value that would be used in the state-dependent model. (Note P_0 is given by (9).)

5.2.2 Determination of R_M

First, define R_{M,n_1} as the mean forward recurrence time given the (Poisson) arrival found n_1 at the processor, so that

$$R_M = \sum_{n_1=0}^{M} R_{M,n_1} P_{n_1}. \tag{14}$$

---

* By dominant time constant we mean the inverse of the smallest (nonzero) exponent in the expansion (10). That is, for t equal to this value, all transient terms will have decreased by at least a factor of e^{-1}.

Now for $M = 1$ we clearly have

$$R_{1,0} = \frac{s_1 + s_2}{\phi}$$

$$R_{1,1} = s_1\phi + (1 - \phi)\frac{(s_1 + s_2)}{\phi},$$

so that

$$R_1 = \frac{s_1 + s_2}{\phi} - (s_2 + (1 - \phi)s_1)(1 - P_0).$$

(Recall that our method is exact for $M = 1$.)

For $M > 1$, we note that the quantities R_{M,n_1} satisfy the following second-order system of difference equations.

$$R_{M,0} = R_{M,1} + \frac{1}{\mu_2}$$

$$R_{M,n_1} = \frac{\mu_2}{(\mu_1 + \mu_2)} R_{M,n_1+1} + \frac{\mu_1(1 - \phi)}{(\mu_1 + \mu_2)} R_{M,n_1-1} + \frac{1}{(\mu_1 + \mu_2)}, \; 0 < n_1 < M$$

$$R_{M,M} = (1 - \phi)R_{M,M-1} + \frac{1}{\mu_1}.$$

Writing this in the simpler notational form (with the obvious correspondences):

$$\begin{aligned} T_0 &= T_1 + d_0 \\ T_i &= aT_{i+1} + bT_{i-1} + d, \; 0 < i < M \\ T_M &= (1 - \phi)T_{M-1} + d_1 \end{aligned} \tag{15}$$

and noting that

$$1 - a - b = 1 - \frac{\mu_2}{\mu_1 + \mu_2} - \frac{\mu_1(1 - \phi)}{\mu_1 + \mu_2} > 0 \quad \text{for} \quad \phi < 1$$

and

$$1 - 4ab = 1 - 4\frac{\mu_1\mu_2(1 - \phi)}{(\mu_1 + \mu_2)^2} = \frac{(\mu_1 - \mu_2)^2 + 4\mu_1\mu_2}{(\mu_1 + \mu_2)^2} > 0,$$

we see that the general solution for this system is given by (e.g., see Ref. 9)

$$T_i = Ar_1^i + Br_2^i + \frac{d}{1 - a - b}, \tag{16}$$

where

$$r_1 = \frac{1 + \sqrt{1 - 4ab}}{2a}$$

$$r_2 = \frac{1 - \sqrt{1 - 4ab}}{2a}.$$

The constants A, B are readily obtained from the two boundary conditions of (15). R_M is then given by (14). Thus we can readily solve for R_M for the class of systems of Fig. 2a. However, for more general systems this might pose a difficult problem, although, as noted, any approximate value, in particular a lower bound for R_M, offers the potential for improvement over the state dependent server model.

5.2.3 Determination of P_M

Rather than obtain a better estimate for P_M than that given by the state-dependent server model, we content ourselves with using the latter. Thus the difference between the results we will give for our approximation and those from the state-dependent server model are due only to the replacement of $S(M)$ by R_M. For completeness, we give a short derivation of P_M assuming we have a (Markovian) state-dependent server model.

Let B_M be the mean length of time from an upward entry to M in the system till a downward entry to $M - 1$ in the system and let I_{M-1} be the mean length of time from a downward transition to the state $M - 1$ till an upward transition to the state M. Then, clearly,

$$P_M = \frac{B_M}{B_M + I_{M-1}}. \tag{17}$$

Under our Markovian assumptions,

$$B_M = \frac{S(M)}{1 - \lambda S(M)} \tag{18}$$

while for I_{M-1} we have the recursion

$$I_0 = \frac{1}{\lambda}$$

$$I_{M-1} = \frac{1}{\lambda + \mu(M-1)} + \frac{\mu(M-1)}{\lambda + \mu(M-1)}(I_{M-2} + I_{M-1}), \quad M \geq 2, \tag{19}$$

where again $\mu(k) = 1/S(k)$.

Thus we have all the quantities needed to compute our approximation. We are also in a position to compare the accuracy of our approximation as a function of the dominant time constant, $T(M)$

from eq. (11) and the accuracy of the state-dependent model as a function of R_M.

5.3 Numerical results

Table I shows the quantities needed to compute our approximation for the parameter values used in Ref. 2 for the case $M = 2$.

Table II compares the accuracy of our approximation with that of the state-dependent server model. Shown are the ratio of $\bar{n}_q$ (exact) to $\bar{n}_q$ (state-dependent server model) and the ratio of the mean forward recurrence time R_M to the mean interdeparture time $S(M)$. As anticipated, these ratios compare very closely. Hence, $R_M/S(M)$ appears to be an excellent measure of the adequacy of the state-dependent server model. Also shown on Table II is the similar ratio for our approximation, and the quantity $T(M)$. We see first a significant improvement over the state-dependent server model (maximum error of 13 percent vs a factor of 10) and second excellent correlation of the error with the dominant time constant.

Note that the approximation does quite well even when $T(2) > \lambda^{-1}$, a fact that might be expected when the *mean* queue lengths are quite large. However, on comparing, for example, for $\phi = 0.9$, $S(2) = 0.95238$, the cases $r = 10$ and $r = 0.1$, we see that the dominant time constant and the *mean* queue length do not give the whole story. The impact of the variability of the queue length and the determination of the quantity P_M on the accuracy of our approximation are currently being investigated.

Table I—Quantities used in the approximation

Input Parameters					
$\lambda = 1$					
ϕ	μ_1	μ_2	S (2)	R_2	P_2
0.1	10.60	95.36	0.95238	0.9525	0.9106
	15.75	14.18		0.9720	0.9180
	106.0	9.536		1.054	0.9106
	12.61	113.5	0.80000	0.8003	0.6508
	18.75	16.88		0.8181	0.6737
	126.1	11.35		0.8855	0.6508
0.5	2.119	10.60	0.95238	0.9528	0.9106
	3.150	1.575		1.186	0.9180
	21.19	1.060		1.834	0.9106
	2.523	12.61	0.80000	0.8015	0.6508
	3.750	1.875		0.9900	0.6737
	25.23	1.261		1.543	0.6508
0.9	1.177	1.118	0.95238	0.9582	0.9106
	1.750	0.1750		2.952	0.9180
	11.77	0.1177		8.668	0.9106
	1.402	1.402	0.80000	0.8076	0.6508
	2.083	0.2083		2.415	0.6737
	14.02	0.1402		7.293	0.6508

Table II—Average value of external queue length, $\bar{n}_q$ (E – Exact[2]; A – our approximation; SD – State-dependent server model; — Not available in Ref. 2)

			$\lambda = 1$				
ϕ	$S(2)$	r	$\bar{n}_q(E)$	$\frac{\bar{n}_q(E)}{\bar{n}_q(A)}$	$\frac{T(2)}{\lambda^{-1}}$	$\frac{\bar{n}_q(E)}{\bar{n}_q(SD)}$	$\frac{R_2}{S(2)}$
0.1	0.95238	10	18.21	1.000+	0.013	1.000+	1.000+
		1	18.93	1.008	0.071	1.030	1.025
		0.1	20.17	1.000+	0.010	1.107	1.107
	0.80000	10	2.603	1.000+	0.011	1.000+	1.004
			—	—	0.059	—	1.023
		0.1	2.882	1.001	0.011	1.107	1.107
0.5	0.95238	10	18.23	1.001	0.120	1.001	1.000+
		1	23.45	1.026	0.635	1.278	1.245
		0.1	35.81	1.021	0.120	1.967	1.925
	0.80000	10	2.606	1.001	0.101	1.001	1.002
		1	3.440	1.033	0.533	1.276	1.238
		0.1	5.116	1.020	0.101	1.965	1.925
0.9	0.95238	10	18.37	1.003	1.15	1.009	1.006
		1	64.26	1.130	5.71	3.500	3.100
		0.1	176.7	1.067	1.08	9.700	9.101
	0.8000	10	2.625	1.002	0.909	1.008	1.00
		1	9.420	1.132	4.80	3.495	3.081
		0.1	25.21	1.064	0.909	9.700	9.117

VI. CONCLUDING REMARKS

We have shown that the errors made in using the state-dependent server model to characterize multiprogramming computer systems can be explicitly related to the concept of the mean forward recurrence time, R_M, of the interdeparture process of the system under saturation. Moreover, we have shown how this latter quantity can be used to obtain significantly improved approximations for the mean waiting time in the external queue.

For a simple system, we have shown how all the needed quantities can be computed and how the accuracy of our approximations can be assessed. For more general systems involving multiple chains and priority classes, the computation of R_M may be extremely difficult. However, as we have seen, *some* estimate of R_M may be necessary to insure that an accurate model has been constructed. That is, without any knowledge of R_M one cannot determine delays as seen by customers, nor size the buffer needed to queue requests, etc.

From a design point of view, the variability in the output process as evidenced by R_M must also be considered. For example, if there are, say, two classes of customers with only a minority being heavy service consumers, then, *if possible*, it clearly is an advantage to replace the single value for a multiprogramming level M by one for each class. That is, form two external queues and limit the number of class i in the system to M_i.

Our approach can also be used to obtain approximations to other quantities of interest. The basic idea is to treat the computer system as an $M/G/1$ queue—when it is saturated. For example, one can match the mean and variance of the output process by a suitable distribution for G and obtain approximate external queue delay distributions. One can also couple this type of analysis with more detailed descriptions of the time in the computer system—once a job enters. For example, Mitrani[8] looks at the *total system* time of a customer for our simple computer system and considers the multiprogramming level that minimizes this. His approach is to treat in detail the time *in* the computer system and to add to this an external delay. His implicit assumption that $R_M = S(M)$ can readily be removed to obtain more accurate results. The method given here would also benefit from improved estimates of P_M, particularly for lightly loaded systems with $M \gg 1$. This, as well as extensions and a more detailed characterization of the accuracy of our approximation, are currently being studied.

Finally, we reemphasize that, while the incorporation of R_M into an analysis may be difficult, it may very well be necessary. However, to estimate R_M in a running system, one need only measure the mean and variance of interdeparture times when the system is saturated.

VII. ACKNOWLEDGMENTS

I would like to thank H. Heffes, R. J. T. Morris, and D. L. Jagerman for many helpful discussions and comments. The reviewer's suggestions are also greatly appreciated.

REFERENCES

1. B. Avi-Itzhak and D. P. Heyman, "Approximate Queueing Models for Multiprogramming Computer Systems," Operations Research, *21* (1973), pp. 1212–1230.
2. A. G. Konheim and M. Reiser, "Finite Capacity Queueing Systems with Applications in Computer Modelling," SIAM J. on Computing, *7* (1978), pp. 210–229.
3. P. J. Courtois, "Decomposability, Instabilities and Saturation in Multiprogramming Systems," Comm. ACM, *18* (1975).
4. P. J. Courtois, "Error Analysis in Nearly-Completely Decomposable Stochastic Systems," Econometrica, *43* (1975).
5. R. B. Cooper, *Introduction to Queueing Theory*, New York: MacMillan, 1972.
6. M. Reiser and H. Kobayashi, "The Effects of Service Time Distributions on System Performance," Information Processing, *74*, Amsterdam: North-Holland, 1974.
7. J. Riordan, *Stochastic Server Systems*, New York: John Wiley, 1962.
8. I. Mitrani, "Non-priority Multiprogramming Systems Under Heavy Demand Conditions—Customers' Viewpoint," J. ACM, *19*, No. 3 (July 1972), pp. 445–452.
9. E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, New York: John Wiley, 1966.