

Approximating Pre-emptive Priority Dispatching in a Multiprogramming Model

Abstract: The formulation of the closed queuing network model of a multiprogramming computer system is generalized to allow each task to have its own set of facility service rates and I/O device selection probability distribution. In the model, processor sharing is assumed for different types of customers. It is shown through a series of investigations that the model reasonably approximates pre-emptive priority dispatching.

Introduction

The closed queuing network model first investigated by Jackson [1] and by Gordon and Newell [2] has been applied by Arora and Gallo [3], Baskett [4], Buzen [5], Mitrani [6], Moore [7], and Tanaka [8] to multiprogramming computer systems. Recently, Adiri [9] reviewed the application of queuing network models to computer systems. In a closed queuing network model of a multiprogramming system, N tasks circulate in the queuing network (Fig. 1). Each task, after leaving the computer processor queue, randomly selects and enters an input/output (I/O) device queue. When service for a task by an I/O device is completed, the task enters the computer processor queue. Service is dispatched to tasks in each queue on a first-come, first-served (FCFS) basis.

In the closed queuing network models, all tasks are assumed to be indistinguishable, having the following statistical characteristics:

1. Service times in the computer processor queue and in the M I/O device queues are statistically independent and have exponential distributions with parameters μ_0 , and μ_1, \dots, μ_M , respectively.
2. Selection of one of the M I/O devices for any task that is departing from the computer processor queue is governed by a probability function p_j , the probability of selecting the j th I/O device queue, where $0 \leq p_j$, and $\sum_{j=1}^M p_j = 1$.

The indistinguishability of tasks limits the usefulness of the closed queuing network model of a multiprogramming computer system. The computer processor service

times (compute intervals) for tasks have been empirically observed to be more closely approximated by a hyper-exponential distribution than by an exponential distribution [4, 7]. This can result from some tasks having very high computer processor service rates (I/O-bound and paging-bound tasks), while other tasks have moderate to very low computer processor service rates (compute-bound tasks). Another reason for distinguishing among tasks is that computer systems having multiple software subsystems (applications) have become increasingly common. The model required to represent such systems has to account for the effect on performance of each subsystem (task).

When tasks having widely differing computer processor servicing rates are multiprogrammed using a FCFS dispatching algorithm, then tasks with high service rates become queued up in the computer behind tasks with low service rates. The load on the system is imbalanced. Therefore some form of pre-emptive priority dispatching algorithm for the computer processor is appropriate for balancing the system load across the I/O devices. A reasonable multiprogramming model should represent the main effects that such a dispatching algorithm has on performance.

In a multiprogramming computer system, each task selects an I/O device based on its own pattern of usage. In general, it is not possible to represent this by a single probability function (assumption 2). For example, it would be difficult to represent that situation for which some I/O devices are utilized by only one task. In this

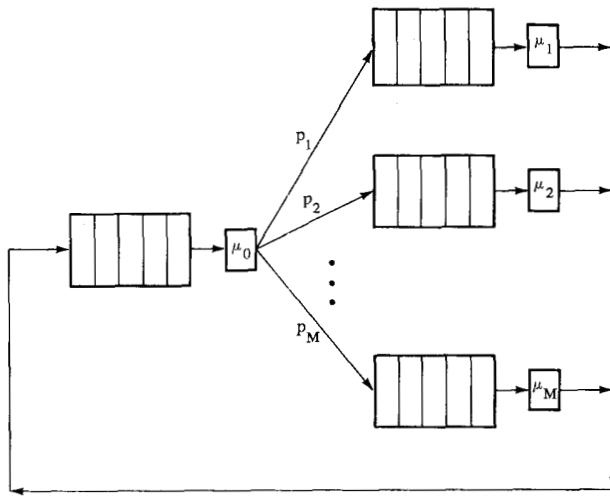


Figure 1 Closed queuing network model of a multiprogramming computer system.

case, no congestion would occur at these I/O devices and the utilization of the computer processor would be higher than if congestion had occurred.

In developing a more realistic model of a multiprogramming computer system, assumptions 1 and 2 were modified. Each task was assumed to have its own computer processor service rate and its own probability distribution for selecting an I/O device. The problem formulation used permitted a pre-emptive priority dispatching algorithm to be approximated. The model developed is referred to as the *multiprogramming model with N distinguishable tasks*.

Analysis

Posner and Bernholtz [10] proposed a solution for the probability distribution of the state of a closed queuing network model having N customers derived from L customer classes. Associated with each customer class are its own set of service rates and its own queue selection probability distribution for each queue in the network. Posner and Bernholtz also took into account the travel time of a customer in going from one queue to another. Their model, with the travel times assumed to be zero, served as the basis for the formulation used here. Recently, Muntz and Baskett [11] applied a technique developed by Chandy [12] to extend the applicability of queuing network models with L customer classes to more general service stations.

Let there be N distinguishable tasks (N classes of customers with one customer in each class) and M I/O devices in the closed queuing network of a multiprogramming computer system. Let $N^* = \{1, \dots, N\}$ be

the set of indices for the tasks and $M^* = \{0, 1, \dots, M\}$ be the set of indices for the computer processor and the M I/O device queues. The service rate of task i at station j is defined to be $\mu_j(i)$, $j \in M^*$ and $i \in N^*$. The I/O device selection probability function for task i is defined to be

$$p_j(i) \geq 0, \sum_{j \in M^*} p_j(i) = 1, \text{ and } p_0(i) = 0,$$

where $p_j(i)$ is the probability of task i selecting the j th I/O device queue.

Let the state of the system be defined by the N -tuple

$$J = (j_1, j_2, \dots, j_N), j_i \in M^* \text{ and } i \in N^*,$$

and let S be the set of N -tuples representing all possible states of the system such that $J \in S$. The N -tuple J indicates the queue occupied by each task, i.e., task 1 is in queue j_1 , task 2 is in queue j_2 , etc. The probability that the i th task is in service, given that it is in queue j_i and the state of the queue network is J , is represented by the probability function $q_j(j_i)$, where $J \in S$, $i \in N^*$, $j_i \in M^*$, and

$$\sum_{\substack{k \in N^* \\ j_k = j_i}} q_j(j_k) = 1.$$

We assume

$$q_j(j_i) = \left(\sum_{k \in N^*} \varepsilon_{j_i j_k} \right)^{-1}, \text{ where } \varepsilon_{j_i j_k} = \begin{cases} 0 & \text{if } j_i \neq j_k \\ 1 & \text{if } j_i = j_k \end{cases}$$

which states that the probability that the i th task is in service in the j_i th queue is equal to the reciprocal of the number of tasks in the j_i th queue. This assumption renders the multiprogramming model with N distinguishable tasks equivalent to a generalization of the processor shared model [11].

The servicing rate of task i in queue j_i when the state of this queue network is J is given by $\mu_{j_i}(i) q_j(j_i) P_J$, where P_J is the steady state probability of the queue network being in state J . The P_J are computed from the detailed balance equations, which equate the rate at which a particular task leaves an I/O device queue to the rate at which it enters the I/O device queue from the computer processor queue. The detailed balance equations have the simplified form

$$\mu_{j_i}(i) q_j(j_i) P_J = \mu_0(i) q_{j^*}(j_i^*) p_{j_i}(i) P_{J^*},$$

where $K = (j_1, \dots, j_i, \dots, j_N)$, $j_i^* = 0$, and $J = (j_1, \dots, j_i, \dots, j_N)$, $j_i \neq 0$ for $i \in N^*$. The i th task leaves the computer processor and enters the j_i th I/O device queue when the state of the system goes from K to J .

The rate of transition out of the given state requires interpretation. For the sake of simplifying the discussion, let

$$\mu_0(1) > \mu_0(2) > \dots > \mu_0(N)$$

and the state of the system be $(0, 0, \dots, 0)$, i.e., all the tasks are in the computer processor queue. Then the possible transition rates out of this state are

$$\begin{aligned} \mu_0(1) \frac{1}{N} P_{(0,0,\dots,0)} &> \mu_0(2) \frac{1}{N} P_{(0,0,\dots,0)} > \dots \\ &> \mu_0(N) \frac{1}{N} P_{(0,0,\dots,0)}. \end{aligned}$$

Therefore a transition out of state $(0, 0, \dots, 0)$ is most likely to occur because task 1 completes service in the computer processor queue, then because task 2 completes service, etc. This approximates the sequence in which tasks would be serviced when dispatched by a pre-emptive priority dispatching algorithm that assigns priority based on the service rate of a task. The higher the computer service rate, the higher the priority. Tasks with the same service rate have the same priority. In the next section, we show how the multiprogramming model with N distinguishable tasks approximates such a pre-emptive priority computer dispatching algorithm.

The form of the detailed balance equations simplifies the analysis. For example, let $N = 4$ tasks, $M = 2$ I/O devices, and the state of the system be $J = (2, 2, 2, 0)$; i.e., tasks 1, 2, and 3 are in the I/O device 2 queue and task 4 is in the computer processor queue. The service rate of task 1 out of the I/O device 2 queue is

$$\frac{1}{3} \mu_2(1) P_{(2,2,2,0)} = \frac{1}{2} \mu_0(1) p_2(1) P_{(0,2,2,0)}$$

An equation that is independent of the sequence of state transitions taken between the two states can be derived for $P_{(2,2,2,0)}$ in terms of $P_{(0,0,0,0)}$. The following is one such sequence of state transitions: $(2, 2, 2, 0)$, $(0, 2, 2, 0)$, $(0, 0, 2, 0)$, and $(0, 0, 0, 0)$. From the detailed balance equations we obtain

$$P_{(2,2,2,0)} = \frac{3 \mu_0(1)}{2 \mu_2(1)} p_2(1) P_{(0,2,2,0)}$$

$$P_{(0,2,2,0)} = \frac{2 \mu_0(2)}{3 \mu_2(2)} p_2(2) P_{(0,0,2,0)}$$

and

$$P_{(0,0,2,0)} = \frac{1 \mu_0(3)}{4 \mu_2(3)} p_2(3) P_{(0,0,0,0)}$$

Therefore

$$P_{(2,2,2,0)} = \frac{1 \mu_0(1) \mu_0(2) \mu_0(3)}{4 \mu_2(1) \mu_2(2) \mu_2(3)} p_2(1) p_2(2) p_2(3) P_{(0,0,0,0)}$$

In general, due to the assumed form for the $q_j(j_i)$ (the processor shared model), an equation for any P_J can be derived in terms of $P_{(0,0,\dots,0)}$. The derived equations are independent of the sequence of state transitions selected. By introducing more notation, a general form for P_J , $J \in \mathcal{S}$, can be derived. Let I^* be the set of indices of all tasks

that are not in the computer processor queue when the state of the system is $J = (j_1, j_2, \dots, j_N)$. That is,

$$I^* = \{i(1), i(2), \dots, i(L) | j_{i(\ell)} \neq 0, \ell = 1, \dots, L\}.$$

The number of tasks not in the computer processor queue is L when the state of the system is J , and $i(\ell)$, $\ell = 1, \dots, L$, are the corresponding indices of these tasks. In our example, for the state $J = (2, 2, 2, 0)$, $I^* = \{1, 2, 3\}$, since tasks 1, 2, and 3 are not in the computer processor queue. For a given $J \in \mathcal{S}$, let $J_0 = (0, 0, \dots, 0)$, \dots , $J_L = J$ be a sequence (not necessarily unique) of states of the queue network that results when the state of the network goes from J_0 to J_L by having the $i(\ell)$ th task complete service in the computer processor and enter the $j_{i(\ell)}$ th I/O device queue, where $i(\ell) \in I^*$. This sequence of state transitions, $\{J_k\}_{k=0}^L$, is used to determine the $q_{J_\ell}(j_{i(\ell)})$ present in the solution of P_J in terms of $P_{(0,0,\dots,0)}$. The solution for P_J is

$$P_J = P_{(0,0,\dots,0)} \prod_{\ell=1}^L \frac{\mu_0[i(\ell)]}{\mu_{j_{i(\ell)}}(i(\ell))} \frac{q_{J_{\ell-1}}(j_{i(\ell)})}{q_{J_\ell}(j_{i(\ell)})} p_{j_{i(\ell)}}(i(\ell)),$$

where $i(\ell) \in I^*$ and $j_{i(\ell)} = 0$ for the $i(\ell)$ th task in state $J_{\ell-1}$ and $j_{i(\ell)} \neq 0$ for state J_ℓ ; i.e., in the ℓ th transition the $i(\ell)$ th task leaves the computer processor queue and enters an I/O device queue.

Probability $P_{(0,0,\dots,0)}$ may be determined directly from the normalization equation

$$\sum_{J \in \mathcal{S}} P_J = 1.$$

As a simple example, consider the case in which $N = 2$ tasks and $M = 2$ I/O devices, as shown in Table 1. The solutions of P_1 through P_8 in terms of P_0 are

$$P_1 = \frac{\mu_0(1)}{\mu_1(1)} \frac{1}{2} p_1(1) P_0,$$

$$P_2 = \frac{\mu_0(1)}{\mu_2(1)} \frac{1}{2} p_2(1) P_0,$$

$$P_3 = \frac{\mu_0(2)}{\mu_1(2)} \frac{1}{2} p_1(2) P_0,$$

$$P_4 = \frac{\mu_0(2)}{\mu_2(2)} \frac{1}{2} p_2(2) P_0,$$

$$P_5 = \frac{\mu_0(1) \mu_0(2)}{\mu_1(1) \mu_1(2)} p_1(1) p_1(2) P_0,$$

$$P_6 = \frac{\mu_0(1) \mu_0(2)}{\mu_2(1) \mu_1(2)} \frac{1}{2} p_2(1) p_1(2) P_0,$$

$$P_7 = \frac{\mu_0(1) \mu_0(2)}{\mu_1(1) \mu_2(2)} \frac{1}{2} p_1(1) p_2(2) P_0, \text{ and}$$

$$P_8 = \frac{\mu_0(1) \mu_0(2)}{\mu_2(1) \mu_2(2)} p_2(1) p_2(2) P_0.$$

Pre-emptive priority computer dispatching

For illustrative purposes, this section presents the results of using the multiprogramming model with N distinguishable tasks (the analytical model) in three separate investigations. In each investigation, three tasks and three I/O devices were assumed. An investigation consisted of comparing the effects that two different sets (two cases) of I/O device selection probability distributions had on the utilizations and average queue lengths of the computer processor and the I/O devices. Only the results on utilization are presented. For each set of I/O device selection probability distributions, the results for three sets of I/O device service rates were computed. This was done to show the sensitivity of utilization of the computer processor to changes in the service rates of the I/O devices. Tables 2 and 3 summarize the values of the parameters assumed and analyzed in each investigation. In the second and third investigations, simulation models of pre-emptive priority dispatching and FCFS dispatching were used to compare results using the simulation models to the results obtained from the analytical model.

Table 1 Detailed balance equations.

J	State symbol	One set of equations
(0,0)	0	
(1,0)	1	$\mu_1(1)P_1 = \mu_0(1) \frac{1}{2} p_1(1)P_0$
(2,0)	2	$\mu_2(1)P_2 = \mu_0(1) \frac{1}{2} p_2(2)P_0$
(0,1)	3	$\mu_1(2)P_3 = \mu_0(2) \frac{1}{2} p_1(2)P_0$
(0,2)	4	$\mu_2(2)P_4 = \mu_0(2) \frac{1}{2} p_2(2)P_0$
(1,1)	5	$\mu_1(1) \frac{1}{2} P_5 = \mu_0(1)p_1(1)P_3$
(2,1)	6	$\mu_2(1)P_6 = \mu_0(1)p_2(1)P_3$
(1,2)	7	$\mu_1(1)P_7 = \mu_0(1)p_1(1)P_4$
(2,2)	8	$\mu_2(1) \frac{1}{2} P_8 = \mu_0(1)p_2(1)P_4$

Table 2 Computer processor service rates and I/O device selection probability distributions used in the three investigations.

Investigation	Task	Computer processor service rate	I/O device selection probability distribution					
			Case 1			Case 2		
			Dev. 1	Dev. 2	Dev. 3	Dev. 1	Dev. 2	Dev. 3
1	1	1.0	1.0	0.0	0.0	0.333	0.333	0.333
	2	1.0	0.0	1.0	0.0	0.333	0.333	0.333
	3	1.0	0.0	0.0	1.0	0.333	0.333	0.333
2	1	0.1	1.0	0.0	0.0	0.50	0.25	0.25
	2	10.0	0.0	1.0	0.0	0.50	0.50	0.00
	3	10.0	0.0	0.0	1.0	0.50	0.00	0.50
3	1	0.1	1.0	0.0	0.0	0.50	0.25	0.25
	2	1.0	0.0	1.0	0.0	0.50	0.30	0.20
	3	10.0	0.0	0.0	1.0	0.50	0.40	0.10

Very different computer processor utilization can result by assuming that all tasks have the same I/O device selection probability distribution instead of assuming the distribution to be different for each task. The first investigation illustrates this. The three computer processor service rates were all equal to 1.0. Therefore no computer processor pre-emptive priority dispatching was appropriate. In the first case each task selected only one I/O device, such that there was no congestion in the I/O de-

Table 3 Three sets of I/O device service rates used in each investigation.

I/O device	Service rate		
	Set 1	Set 2	Set 3
1	0.25	0.25	0.125
2	0.25	0.125	0.25
3	0.25	0.125	0.25

Table 4 Utilization of the facilities for the two cases of the first investigation.

I/O device service rates	Facility	Case 1	Case 2
Set 1	Processor	0.549	0.398
	I/O Device 1	0.732	0.530
	I/O Device 2	0.732	0.530
	I/O Device 3	0.732	0.530
Set 2	Processor	0.405	0.246
	I/O Device 1	0.763	0.328
	I/O Device 2	0.856	0.656
	I/O Device 3	0.856	0.656
Set 3	Processor	0.480	0.296
	I/O Device 1	0.848	0.789
	I/O Device 2	0.748	0.394
	I/O Device 3	0.748	0.394

vice subsystem. In the second case each task had the same I/O device selection probability distribution (this is the multiprogramming model for three indistinguishable tasks). Table 4 shows that a substantially different computer processor utilization resulted in the two cases.

In the second investigation, there were two I/O-bound tasks (service rates of 10) and one compute-bound task (service rate of 0.1). Pre-empting the servicing by the computer processor of the computer bound task by any one of the I/O bound tasks was appropriate in this situation. In the first case investigated, each task selected only one I/O device, such that there was no congestion in the I/O device subsystem. For this case, the results of the FCFS simulation model and the analytical model show that utilization of the computer processor was not sub-

stantially affected by the dispatching algorithm. Most of the utilizations of the I/O devices were substantially lower for the FCFS dispatching algorithm. The FCFS algorithm caused a bottleneck to develop in the computer processor. The I/O-bound tasks were typically waiting in the computer processor queue behind the compute-bound task. In the second case computer processor utilization was higher for the FCFS dispatching algorithm, but at the cost of much lower I/O device utilization. The I/O-bound tasks tended to remain queued behind the compute-bound task even in the I/O device queues. Tables 5 and 6 summarize the results of the second investigation, indicating a good agreement between the analytical model and the simulation model of pre-emptive priority dispatching.

Table 5 Utilization of the facilities for case 1 of the second investigation.

<i>I/O device service rates</i>	<i>Facility</i>	<i>Simulation model pre-emptive priority</i>	<i>Analytical model</i>	<i>Simulation model FCFS</i>
Set 1	Processor	0.765	0.738	0.750
	I/O Device 1	0.235	0.276	0.272
	I/O Device 2	0.976	0.956	0.480
	I/O Device 3	0.975	0.956	0.458
Set 2	Processor	0.731	0.726	0.735
	I/O Device 1	0.269	0.281	0.278
	I/O Device 2	0.987	0.979	0.585
	I/O Device 3	0.987	0.979	0.566
Set 3	Processor	0.604	0.589	0.566
	I/O Device 1	0.395	0.432	0.463
	I/O Device 2	0.974	0.962	0.598
	I/O Device 3	0.975	0.962	0.619

Table 6 Utilization of the facilities for case 2 of the second investigation.

<i>I/O device service rates</i>	<i>Facility</i>	<i>Simulation model pre-emptive priority</i>	<i>Analytical model</i>	<i>Simulation model FCFS</i>
Set 1	Processor	0.533	0.606	0.669
	I/O Device 1	0.860	0.820	0.445
	I/O Device 2	0.383	0.410	0.267
	I/O Device 3	0.397	0.410	0.323
Set 2	Processor	0.508	0.521	0.587
	I/O Device 1	0.665	0.629	0.414
	I/O Device 2	0.604	0.629	0.417
	I/O Device 3	0.619	0.629	0.445
Set 3	Processor	0.425	0.456	0.570
	I/O Device 1	0.936	0.946	0.717
	I/O Device 2	0.262	0.236	0.197
	I/O Device 3	0.267	0.236	0.193

In the third investigation, there were a compute-bound task, an I/O-bound task, and a task with a moderate computer processor service rate, with service rates 0.1, 10, and 1.0, respectively. In the first case investigated, there was no congestion in the I/O device subsystem, while in the second case there was substantial congestion. The conclusions derived from the results presented in Tables 7 and 8 are similar to those made in the second investigation. The analytical model reasonably approximates pre-emptive priority dispatching, which is typically used when dispatching service to tasks with widely differing service rates.

Summary

The tasks in a multiprogramming computer system are usually heterogeneous in their usage of system resources. This is especially true for systems supporting multiple applications. The closed queuing network model of a multiprogramming computer system can be rendered more realistic by generalizing it, so that it accounts for distinctly different tasks in the system. When this is done, the effects of different mixes of tasks on the utilizations of the computer processor and I/O devices can be investigated. In fact, it was found that pre-emptive priority computer dispatching can be approximated by a

Table 7 Utilization of the facilities for case 1 of the third investigation.

<i>I/O device service rates</i>	<i>Facility</i>	<i>Simulation model pre-emptive priority</i>	<i>Analytical model</i>	<i>Simulation model FCFS</i>
Set 1	Processor	0.825	0.810	0.807
	I/O Device 1	0.218	0.245	0.281
	I/O Device 2	0.796	0.694	0.401
	I/O Device 3	0.972	0.951	0.439
Set 2	Processor	0.750	0.770	0.773
	I/O Device 1	0.281	0.262	0.293
	I/O Device 2	0.879	0.822	0.541
	I/O Device 3	0.986	0.977	0.601
Set 3	Processor	0.706	0.694	0.691
	I/O Device 1	0.376	0.394	0.468
	I/O Device 2	0.790	0.714	0.533
	I/O Device 3	0.975	0.955	0.691

Table 8 Utilization of the facilities for case 2 of the third investigation.

<i>I/O device service rates</i>	<i>Facility</i>	<i>Simulation model pre-emptive priority</i>	<i>Analytical model</i>	<i>Simulation model FCFS</i>
Set 1	Processor	0.625	0.684	0.748
	I/O Device 1	0.749	0.711	0.445
	I/O Device 2	0.502	0.485	0.308
	I/O Device 3	0.229	0.226	0.145
Set 2	Processor	0.553	0.582	0.625
	I/O Device 1	0.511	0.513	0.376
	I/O Device 2	0.704	0.688	0.533
	I/O Device 3	0.333	0.338	0.284
Set 3	Processor	0.485	0.524	0.581
	I/O Device 1	0.919	0.898	0.658
	I/O Device 2	0.269	0.300	0.218
	I/O Device 3	0.142	0.149	0.091

generalization of the processor-shared model. This is significant, since some form of pre-emptive priority computer dispatching is usually used in multiprogramming computer systems for balancing the load on the computer processor(s) and the I/O devices.

Acknowledgment

The model reported in this paper was motivated in part by the efforts of D. T. Brown of the Systems Development Division to extend the applicability of queuing models to multiprogramming computer systems.

References

1. J. Jackson, "Networks of Waiting Lines," *Operations Research* 5, 518 (1957).
2. W. J. Gordon and G. F. Newell, "Cyclic Queuing Systems with Exponential Servers," *Operations Research* 15, 254 (1967).
3. S. Arora and A. Gallo, "The Optimal Organization of Multiprogrammed Multi-Level Memory," *Proc. ACM SIGOPS Workshop on System Performance Evaluation*, New York, April 1971, p. 104.
4. F. Baskett, *Mathematical Models of Multiprogramming Computer Systems*, Ph.D. Thesis, Univ. of Texas, Austin, December 1970.
5. J. Buzen, "Analysis of System Bottlenecks Using a Queuing Network Model," *Proc. ACM SIGOPS Workshop on System Performance Evaluation*, New York, April 1971, p. 82.
6. I. Mitrani, "Non-Priority Multiprogrammed Systems Under Heavy Demand Conditions—Customer Point of View," *J. ACM* 19, No. 3, 445 (1972).
7. C. Moore, *Network Models for Large Scale Time Sharing Systems*, Ph.D. Thesis, Univ. of Michigan, Ann Arbor, April 1971.
8. H. Tanaka, "An Analysis of On-Line Systems Using Parallel Cyclic Queues," *J. Inst. Elect. and Com. of Japan* 53-C, 10, 756 (1970).
9. I. Adiri, *Queuing Models for Multiprogrammed Computers*, IBM Research Report RC 3802, Yorktown Heights, N.Y., March 30, 1972.
10. M. Posner and G. Berholtz, "Closed Finite Queuing Networks with Time Lags with Several Classes of Units," *Operations Research* 16, 977 (1968).
11. R. Muntz and F. Baskett, "Open, Closed, and Mixed Networks of Queues with Different Classes of Units," to be published in *J. ACM*.
12. K. Chandy, "The Analysis and Solutions for General Queuing Networks," *Proc. Sixth Annual Princeton Conf. on Information Sciences and Systems*, Princeton University, March 1972.

Received February 9, 1973; revised June 4, 1973

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598