Time-shared Systems: A Theoretical Treatment

LEONARD KLEINROCK

Department of Engineering, University of California, Los Angeles

ABSTRACT. Time-shared computer (or processing) facilities are treated as stochastic queueing systems under priority service disciplines, and the performance measure of these systems is taken to be the average time spent in the system. Models are analyzed in which time-shared computer usage is obtained by giving each request a fixed quantum Q of time on the processor, after which the request is placed at the end of a queue of other requests; the queue of requests is constantly cycled, giving each user Q seconds on the machine per cycle. The case for which $Q \to 0$ (a processor-shared model) is then analyzed using methods from queueing theory. A general time-shared facility is then considered in which *priority groups* are introduced. Specifically, the pth priority group is given g_pQ seconds in the processor each time around. Letting $Q \to 0$ gives results for the priority processor-shared system. These disciplines are compared with the first-come-first-served disciplines. The systems considered provide the two basic features desired in any time-shared system, namely, rapid service for short jobs and the virtual appearance of a (fractional capacity) processor available on a full-time basis. No charge is made for swap time, thus providing results for "ideal" systems. The results hold only for Poisson arrivals and geometric (or exponential) service-time distributions.

1. *Introduction*

Interest in time-shared computing systems has been growing at an increasing rate in recent years. A number of such systems have been cropping up in various places throughout the country [1–5]. The motivation for such interest is toward encouraging the interaction between the user (programmer) and the computer itself. Furthermore, it is recognized that the availability of computers must be increased so rapidly that we may soon find it expedient to offer computational and processing capacity as a "public utility." A natural way to do this is to provide the public with access to computers on a time-shared basis (not unlike the telephone company's use of graded trunk lines), thus providing a high efficiency for the user as well as for the computer facility.

Time-shared systems are often designed with the intent of appearing to a user as his personal processor (where, ideally, he is unaware of the presence of any other users). Of course, no such ideal systems can guarantee a full-capacity full-time machine to any user (in the time-shared mode), but rather they offer a fractional-capacity "full-time" machine to each user. In the ideal case, at any time, the fraction of the total capacity offered to any user will be just[1] the inverse of the number of users currently requesting service (i.e., we assume an harmonic variation of individual capacity with number of users).

Unfortunately, very little work has been carried out in analyzing the behavior of

This research was sponsored by the Applied Mathematics Division, Office of Research, United States Air Force under Grant No. AFOSR-700-65. This work represents a generalized version of the author's paper, "Theory of Queues Applied to Time-Shared Computer Systems, which appeared in the 1966 IEEE Region Six-Conference Record.

[1] This is generalized in the priority model described in Section 2.

time-shared systems from a mathematical viewpoint. In this paper we proceed in that direction. We begin with a queueing model of a time-shared system recently analyzed by Kleinrock [6] in which the user present at the head of the queue receives Q seconds of service on the processor and is then returned to the tail of the queue. Thus, a user cycles around the queue n times, where nQ is the number of seconds of processing time he requires. This is called the "round-robin" model (see Section 2).

The round-robin model does not focus attention on any swap-time charges (i.e., the cost in time for removing the old job from and placing the new job on the processor). If desired, however, it can be added into the interval Q as an additional time cost. In this paper, we take the view that the best performance attainable in a time-shared system is achieved when swap time is assumed to be identically zero. By "best" we mean in terms of some average job-waiting times in the queue. Indeed, we wish to establish upper limits of performance for time-shared processors and do so by assuming zero swap time.

If we consider a round-robin system in which we allow $Q \to 0$, we arrive at an interesting model of a processor-shared system in which users are cycling around at an infinite rate, receiving an infinitesimal quantum of service infinitely often. When the total service time received equals a user's required processing time, he then leaves the system. Indeed, we see that this is identical to a model in which each user receives continuous processing at a rate C/k operations (say additions) per second when there are a total of k users in the system (where C is the capacity, in operations per second, of the processor).

We also consider the more general case of processor-sharing in which we have P *priority* groups. Members from the pth group ($p = 1, 2, \cdots, P$) receive g_pQ seconds of service each time around the cycle. As $Q \to 0$, we then obtain the priority processor-shared system, which represents a fairly general ideal (in the sense of zero swap-time and zero wait on queue) time-shared system.

In Section 2, we carefully define these three models.

2. *Queueing Models of Time-shared Facilities*

The Round-Robin Model. Our point of departure is the discrete time model of a time-shared processor studied by Kleinrock [6]. In this model, it is assumed that time is quantized with segments each Q seconds in length. At the end of each time interval, a new unit (or job) arrives in the system with probability λQ (result of a Bernoulli trial); thus, the average number of arrivals per second is λ. The service time (i.e., the required processing time) of a newly arriving unit is chosen independently from a geometric distribution such that for $0 \leq \sigma < 1$,

$$s_n = (1 - \sigma)\sigma^{n-1} \quad n = 1, 2, 3, \cdots , \tag{1}$$

where s_n is the probability that a unit's service time is exactly n time intervals long (i.e., that its service time is nQ seconds).

The procedure for servicing is as follows: A newly arriving unit joins the end of the queue and waits in line in a first-come-first-served fashion until it finally arrives at the service facility. The server picks the next unit in the queue and performs one unit of service upon it (i.e., it services this job for exactly Q seconds). At the end of this time interval, the unit leaves the system if its service (processing) is finished;

if not, it joins the end of the queue with its service partially completed, as shown in Figure 1. Obviously, a unit whose processing requirement is nQ time units long will be forced to join the queue n times in all before its service is completed.

An assumption must be made regarding the order in which events take place at the end of a time interval. Consider two types of systems. The first system allows the unit in service to be ejected from the service facility (and then allows it to join the end of the queue if more service is required for this unit), and instantaneously after that a new unit arrives (with probability λQ). This is referred to as a *late-arrival system.* The second system reverses the order in which these events are allowed to occur, giving rise to the *early-arrival system.* In both systems, a new unit is taken into service at the beginning of a time interval.

Processor-shared Model (*No Priorities*). If we assume zero swap-time, we may consider the case of a round-robin system in which $Q \rightarrow 0$. We must be careful in taking this limit, since the service time nQ also goes to zero in this case and our model loses all meaning. Consequently, let us agree to keep the average service time constant as $Q \rightarrow 0$. This involves changing σ, the decay rate in eq. (1) such that $\sigma \rightarrow 1$ as $Q \rightarrow 0$. Specifically, we have that

$$\bar{n} = \sum_{n=1}^{\infty} ns_n = \frac{1}{1 - \sigma}$$

and, defining

$$\frac{1}{\mu C} = \text{average service requirement (in seconds)},$$

we obtain

$$\frac{1}{\mu C} = \frac{Q}{1 - \sigma} = \text{constant as } Q \rightarrow 0 \text{ and } \sigma \rightarrow 1$$

or

$$\sigma = 1 - \mu CQ. \tag{2}$$

Thus, the limiting operation we consider is where $Q \rightarrow 0$ and $\sigma \rightarrow 1$ in the manner expressed in eq. (2). The result of this limit is that the required service l (in operations) is exponentially distributed with paremeter μ, namely,

$$p(l) = \mu e^{-\mu l}, \tag{3}$$

where l is the length of the job.

We have chosen to assume that the length l of a job is given in *number of opera-*

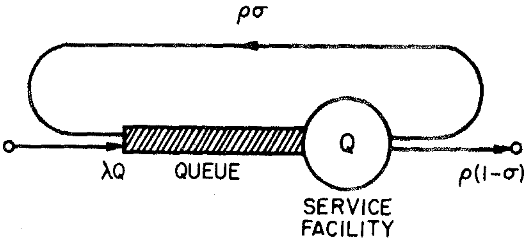

FIG. 1. Round-robin time-shared service system

tions instead of in seconds, thus making the user requirement independent of the machine on which it is serviced. We then define, for any processor, a quantity

C = capacity of a processor in operations (say, additions) per second.

The service time for a job then becomes l/C seconds, with a mean service time of $1/\mu C$ seconds.

The arrival mechanism in the limit then becomes Poisson with an average arrival rate of λ customers per second. This model reduces to a system in which a user is processed at a rate C/k operations per second when there are k users sharing a computer of capacity C. This processing rate varies as new users enter and old ones leave the system. We are here assuming an harmonic variation of individual processing rate with number of customers. (See Figure 2.)

Priority Processor-shared Model. This is a generalization of the processor-shared system considered above. Here we assume that the input traffic is broken up into P separate priority groups, where the pth group has a Bernoulli arrival pattern at an average rate of λ_p customers per second and a geometrically distributed service requirement whose mean is $1/(1 - \sigma_p)$ operations. For the $Q \to 0$ case, we give a member of the pth priority group g_pQ seconds of service each time he cycles around the queue (see Figure 3).

For $Q \to 0$, holding fixed $1/\mu_pC = Q/(1 - \sigma_p)$, this model then reduces to a processor-shared model with a priority structure wherein a member from group p received at time t a fraction f_p , where

$$f_p = \frac{g_p}{\sum_{i=1}^{P} g_i n_i} \tag{4}$$

of the total processing capacity C (here n_i is the number of customers from priority group i present in the system at time t). We note that we then have, for the pth group, Poisson arrivals (λ_p per second) and exponential service with an average of $1/\mu_pC$ seconds. The nonpriority processor-shared model considered earlier is the special case $g_p = 1$ for all p.

The interest in this model is that it can be used to give preferential service to certain groups of users. For convenience, we may consider that the higher the value

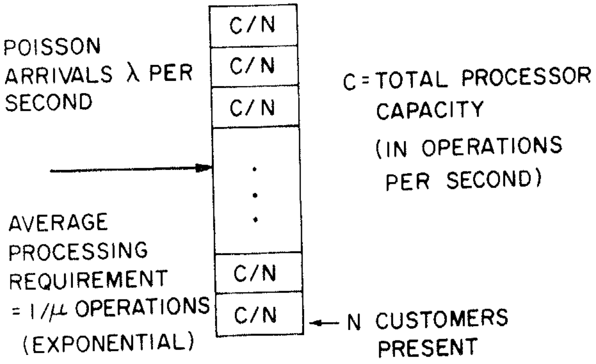

FIG. 2. Processor-shared model with N in system

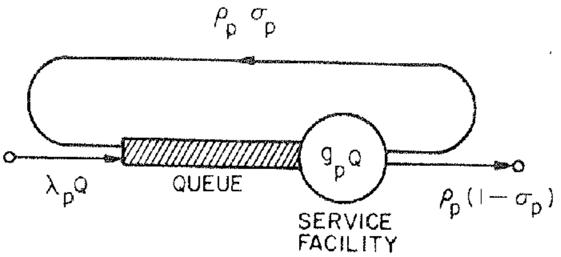

FIG. 3. Round-robin time-shared service system with priorities ($Q > 0$)

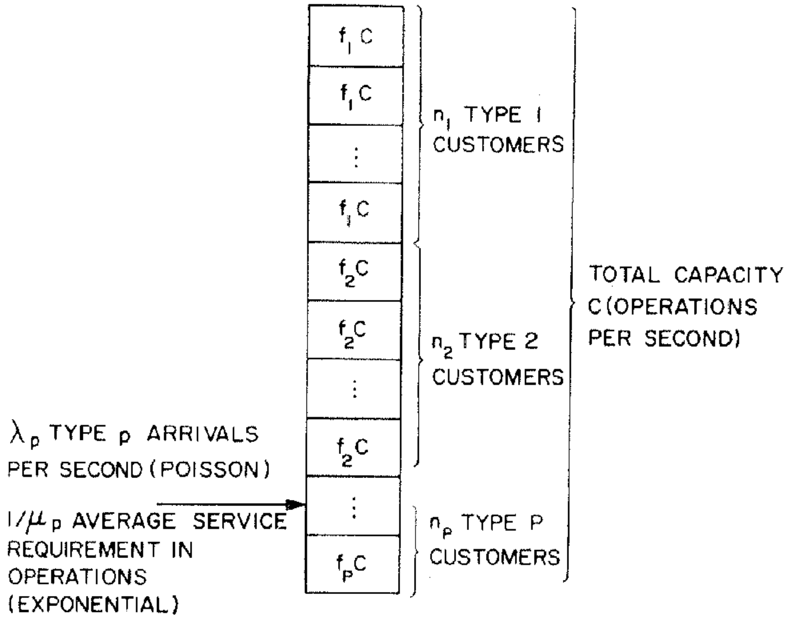
f_1C
f_1C
f_1C
f_2C
f_2C
f_2C
f_pC
n_1 TYPE 1 CUSTOMERS
n_2 TYPE 2 CUSTOMERS
n_p TYPE P CUSTOMERS
TOTAL CAPACITY C (OPERATIONS PER SECOND)
λ_p TYPE p ARRIVALS PER SECOND (POISSON)
$1/\mu_p$ AVERAGE SERVICE REQUIREMENT IN OPERATIONS (EXPONENTIAL)

FIG. 4. Priority processor-shared model with n_p type p in system

of p, the higher is considered the priority of the group. In such a case, we may assume that g_p is a monotonically increasing function of p (although we do not need this for the subsequent development).

A diagram of the priority processor-shared system is shown in Figure 4.

We observe that the two processor-shared models are ideal in the sense that swap-time is assumed to be zero and in that customers are given immediate use of the processor (although only a fractional capacity f_pC).

3. *Results for Time-shared Systems*

Round-Robin System. The Round-Robin system has already been studied [6]. We present the results of that analysis here.

THEOREM 1. *The expected value T_n of the total time*[2] *spent in the late-arrival system*

[2] T_n is the sum of the time spent in the queue and the time spent in the service facility.

for a job whose service time is nQ seconds is

$$T_n = \frac{nQ}{1-\rho} - \frac{\lambda Q^2}{1-\rho}\left[1 + \frac{(1-\sigma\alpha)(1-\alpha^{n-1})}{(1-\sigma)^2(1-\rho)}\right], \tag{5}$$

where

$$\alpha = \sigma + \lambda Q, \tag{6}$$

$$\rho = \frac{\lambda Q}{1-\sigma}. \tag{7}$$

Furthermore, the expected number E_r of customers in the system is given by

$$E_r = \frac{\rho\sigma}{1-\rho}. \tag{8}$$

THEOREM 2. *The expected value T_n' of the total time spent in the early-arrival system for a unit whose service time is nQ seconds is*

$$T_n' = \frac{nQ}{1-\rho} - \rho Q - \frac{\lambda Q^2 \rho}{1-\rho}\left[1 + \frac{(1-\sigma\alpha)(1-\alpha^{n-1})}{(1-\sigma)^2(1-\rho)}\right], \tag{9}$$

where α and ρ are as defined before. The expected number E_r' of customers in the system is given by

$$E_r' = \frac{\rho(1-\lambda Q)}{1-\rho}. \tag{10}$$

THEOREM 3. *The expected value, T_n'', of the total time spent in the strict first-come-first-served system*[3] *for a unit whose service time is nQ seconds is*

$$T_n'' = \frac{QE_r}{1-\sigma} + nQ, \tag{11}$$

where E_r is defined in eq. (8).

We remark here that there is no significant difference in performance between the late- and early-arrival systems. In [6] it is shown that a good approximation to T_n is

$$T_n = nQE_r + nQ. \tag{12}$$

When we compare eqs. (11) and (12), we see that for units which require a number of service intervals less (greater) than $1/(1-\sigma)$, the round-robin waiting time for the late-arrival system is less (greater) than the strict first-come-first-served system. One notes, however, that the average number of service intervals $\bar{n}$ is exactly $1/(1-\sigma)$. Thus, for this approximate solution, the crossover point for waiting time is at the mean number of service intervals. This effect is observable in Figures 5-7 in Section 4.

Processor-shared System. The Processor-Shared model considers the limit of the round-robin model in which $Q \to 0$ and $\sigma = 1 - \mu CQ$, giving a Poisson arrival mechanism with an average of λ units arriving per second and an exponential

[3] This is our reference system and corresponds to the more usual case where a unit receives its complete processing requirement the first time it enters service.

service distribution with an average of $1/\mu$ operations per customer. In this case, the early- and late-arrival systems become identical, and we have the following.

THEOREM 4. *The expected value $T(l)$ of the total time spent in the processor-shared system for a customer requiring l operations is*

$$T(l) = \frac{l/C}{1-\rho} \tag{13}$$

where

$$\begin{aligned} &\rho = \lambda/\mu C, \\ &C = \textit{capacity of the processor in operations per second.} \end{aligned} \tag{14}$$

The expected number E of customers in the system is

$$E = \frac{\rho}{1-\rho}. \tag{15}$$

PROOF. We define l, the required number of operations for a customer as

$$l = \lim_{\substack{Q\to 0 \\ \sigma\to 1}} nCQ. \tag{16}$$

This limit is meaningful in that the distribution on n (eq. (1)) shows that as $\sigma \to 1$, extremely large values of n occur. Indeed, we may calculate the probability distribution for l,

$$\begin{aligned} Pr[L \le l] &= \lim_{\substack{Q\to 0 \\ n\to\infty}} Pr[nCQ \le l] \\ &= \lim_{\substack{Q\to 0 \\ n\to\infty}} \sum_{i=1}^{l/QC} (1-\sigma)\sigma^{i-1} \\ &= \lim_{\substack{Q\to 0 \\ n\to\infty}} 1 - (1-\mu CQ)^{l/QC} \\ &= 1 - e^{-\mu l}, \end{aligned}$$

which proves that l is exponentially distributed with mean $1/\mu$ operations as stated in eq. (3).

We now consider the limiting form for eqs. (5) and (9). We have that

$$\begin{aligned} (1-\sigma\alpha) &= [1-(1-\mu CQ)(1-\mu CQ+\lambda Q)] \\ &= Q[2\mu C - \lambda - Q(\mu C)^2(1-\rho)] \end{aligned}$$

and

$$\begin{aligned} \lim_{Q\to 0} 1 - \alpha^{n-1} &= \lim_{Q\to 0} 1 - (1-\mu CQ+\lambda Q)^{(l/QC)-1} \\ &= \lim_{Q\to 0} 1 - [1-\mu CQ(1-\rho)]^{(l/QC)-1} \\ &= 1 - e^{-\mu l(1-\rho)} \end{aligned}$$

and

$$(1-\sigma)^2 = (1-1+\mu CQ)^2 = (\mu CQ)^2.$$

Thus, defining

$$T(l) = \lim_{\substack{Q \to 0 \\ \sigma \to 1}} T_n ,$$

we obtain from eq. (5),

$$\begin{aligned} T(l) &= \lim_{\substack{Q \to 0 \\ \sigma \to 1}} \left\{ \frac{nQ}{1 - \rho} - \frac{\lambda Q^2}{1 - \rho} \left[1 + \frac{(1 - \sigma\alpha)(1 - \alpha^{n-1})}{(1 - \sigma)^2(1 - \rho)} \right] \right\} \\ &= \frac{l/C}{1 - \rho} - \lim_{Q \to 0} \frac{\lambda\ Q^2}{1 - \rho} \left\{ 1 + \frac{[1 - e^{-\mu l(1-\rho)}]Q[2\mu C - \lambda - Q(\mu C)^2(1 - \rho)]}{(\mu C Q)^2(1 - \rho)} \right\} \\ &= \frac{l/C}{1 - \rho} - \lim_{Q \to 0} \frac{\lambda Q}{1 - \rho} \left\{ Q + \frac{[1 - e^{-\mu l(1-\rho)}][2\mu C - \lambda - Q(\mu C)^2(1 - \rho)]}{(\mu C)^2(1 - \rho)} \right\} \\ &= \frac{l/C}{1 - \rho} . \end{aligned}$$

Thus, we have established eq. (13) for the late-arrival system; but it is clear that $\lim_{Q \to 0} T_n = \lim_{Q \to 0} T_n'$ and so eq. (13) is true for the early-arrival system also. From eqs. (8) and (10) it is clear that $E = \lim_{Q \to 0} E_r' = \lim_{Q \to 0} E_r$. This completes the proof of Theorem 4.

In Section 4 these results are compared with those of the round-robin model.

Priority Processor-shared System. In the Priority Processor-shared system, we have P priority groups with Poisson arrivals at an average rate of λ_p per second and an exponentially distributed service requirement with a mean of $1/\mu_p$ operations ($p = 1, 2, \cdots, P$). For a processor of capacity C operations per second, we assign a customer from the pth priority group a capacity $f_p C$ when there are n_i type-i customers in the system; f_p is given by eq. (4), namely,

$$f_p = \frac{g_p}{\sum_{i=1}^{P} g_i\, n_i} . \tag{4}$$

For such a system, we have the following theorem.

THEOREM 5. *The expected value $T_p(l)$ of the total time spent in the priority processor-shared system for a customer from priority group p who requires l operations is*

$$T_p(l) = \frac{l}{C} \left[1 + \sum_{i=1}^{P} \frac{g_i\, \rho_i}{g_p(1 - \rho)} \right], \tag{17}$$

The expected number, E_p, of type p customers in the system is

$$E_p = \frac{\rho_p}{1 - \rho} \left[1 + \sum_{i=1}^{P} \left(\frac{g_i}{g_p} - 1 \right) \rho_i \right], \tag{18}$$

where

$$\rho_p = \frac{\lambda_p}{\mu_p\, C}$$

and

$$\rho = \sum_{p=1}^{P} \rho_p$$

and where $g_p > 0, \quad p = 1, 2, \cdots, P$.

PROOF. We carry out this proof by assuming $Q > 0$ and that customers from the pth priority group are given g_pQ seconds of service each time they cycle around the round-robin queue. We assume that the pth priority group has a geometrically distributed number of service intervals n with decay parameter $0 \leq \sigma_p < 1$ (see eq. (1)) and that a single arrival of type p occurs during the interval Q with probability λ_pQ (and that no such arrival occurs with probability $1 - \lambda_pQ$). Then when we let $Q \to 0$, we arrive at the priority processor-shared model.[4] The proof follows the approach used in [6] to solve the round-robin system.

Accordingly, let us consider the arrival of a unit (the tagged unit) from priority group p and which requires nQ seconds of service (processing). Let

E_p = expected number of type p customers in the system,

$T_p(n)$ = expected time spent in the system (queue plus service) for the tagged unit,

D_k = expected delay (time spent) between the completion of the tagged unit's $(k - 1)$st ejection from service and its kth ejection from service.[5]

Clearly then,[6]

$$T_p(n) = \sum_{k=1}^{n/g_p} D_k . \tag{19}$$

We now define

N_{ki} = expected number of type-i customers served between the completion of the tagged unit's $(k - 1)$st ejection from service and its kth ejection from service.

Thus,

$$D_k = \sum_{i=1}^{P} N_{ki} g_i Q \tag{20}$$

and so

$$T_p(n) = \sum_{k=1}^{n/g_p} \sum_{i=1}^{P} N_{ki} g_i Q. \tag{21}$$

We now derive a general form for N_{ki}. Upon its arrival to the system, the tagged unit finds a certain number of type-i units in the queue, the expected value of which is E_i by definition. Note that the service facility is empty whenever a new unit enters the system. Thus,

$$N_{1i} = E_i + \delta_{ip} , \tag{22}$$

where

$$\delta_{ip} = \begin{cases} 0, & i \neq p, \\ 1, & i = p. \end{cases}$$

[4] We also assume a late-arrival system; however, this choice is unimportant, since early and late arrival systems are identical for $Q \to 0$.

[5] We complete this definition by assuming that its 0th ejection from service is completed at its time of arrival to the system.

[6] We do not worry about n/g_p being an integer, since we shortly allow $Q \to 0$, which effectively converts this sum to an integral.

The addition of δ_{ip} is due to the tagged unit's first service interval. Now, each E_i-unit of type i will remain in the system with probability σ_i, and so $\sigma_i(N_{1i} - \delta_{ip})$ of them will contribute to N_{2i}. In addition, during the time $Q \sum_{j=1}^{P} g_j(N_{1j} - \delta_{jp})$, devoted to servicing these units found in the original queue, we expect λ_i new units of type i to arrive per second, and so we must also add $\lambda_i Q \sum_{j=1}^{P} g_j(N_{1j} - \delta_{ip})$ more units to N_{2i}. Besides all this, for $n > 1$, we must add one more (the tagged unit itself) to N_{2p}, giving

$$N_{2i} = \sigma_i(N_{1i} - \delta_{ip}) + \lambda_i Q \sum_{j=1}^{P} g_j(N_{1j} - \delta_{jp}) + \delta_{ip}. \tag{23}$$

In calculating N_{3i}, we see that a fraction σ_i of the type-i units which were served during the D_2 time interval will remain in the system; i.e., $\sigma_i(N_{2i} - \delta_{ip})$ type-i will remain. In addition, during the time

$$D_2 = Q \sum_{j=1}^{P} g_j(N_{2j} - \delta_{jp}),$$

we expect $\lambda_i D_2$ new units to arrive on the average. Also, for $n > 2$, we must add one more (the tagged unit again) to N_{3p}.

However, we now notice a new effect entering, namely, the presence of a type-i unit which arrived (with probability $\lambda_i g_p Q$) at the conclusion of the first service interval of the tagged unit. This additional unit was placed in back of the tagged unit when it arrived and therefore did not appear in N_{2i}. From now on, however, it will appear as an additional $\lambda_i g_p Q$ added to each N_{ki} for $k \geq 3$. Thus,

$$N_{3i} = \sigma_i(N_{2i} - \delta_{ip}) + \lambda_i Q \sum_{j=1}^{P} g_j(N_{2j} - \delta_{jp}) + \delta_{ip} + \lambda_i g_p Q. \tag{24}$$

For N_{ki} $(k \geq 3)$ we repeat the arguments used for finding N_{3i}, with the substitutions N_{ki} for N_{3i} and $N_{n-1,i}$ for N_{2i}. Thus, for $k \geq 3$ we obtain

$$N_{ki} = \sigma_i(N_{k-1,i} - \delta_{ip}) + \lambda_i Q \sum_{j=1}^{P} g_j(N_{k-1,j} - \delta_{jp}) + \delta_{ip} + \lambda_i g_p Q. \tag{25}$$

We now make use of the limit $Q \rightarrow 0$ under the condition $\sigma_p = 1 - \mu_p C Q$ (so that $\sigma_p \rightarrow 1$ for all p and the average number of operations required for type-p units remains fixed at $1/\mu_p$). Applying this limiting operation to eq. (25) yields, for $k \geq 3$,

$$\bar{N}_{ki} \equiv \lim_{Q \rightarrow 0} N_{ki} = N_{k-1,i}, \quad k \geq 3. \tag{26}$$

We have

$$\bar{N}_{1i} = N_{1i} = E_i + \delta_{ip} \tag{27}$$

and, from eqs. (23) and (27), we obtain

$$\bar{N}_{2i} = \bar{N}_{1i} = E_i + \delta_{ip}. \tag{28}$$

Applying eq. (28) to eq. (26) for $k = 3$ and repeating the process for all k, we obtain the simple result

$$\bar{N}_{ki} = E_i + \delta_{ip}, \quad k = 1, 2, 3, \cdots. \tag{29}$$

We now consider the limit of $T_p(n)$ as $Q \rightarrow 0$ and define

$$T_p(l) = \lim_{Q\rightarrow 0} T_p(n) \tag{30}$$

From eqs. (21), (29), and (30), we then obtain

$$T_p(l) = \lim_{Q\rightarrow 0} \sum_{k=1}^{n/g_p} \sum_{i=1}^{P} N_{ki}\, g_i\, Q = \sum_{i=1}^{P} g_i \lim_{Q\rightarrow 0} \sum_{k=1}^{n/g_p} N_{ki}\, Q = \sum_{i=1}^{P} \frac{g_i}{g_p} (E_i + \delta_{ip}) \lim_{Q\rightarrow 0} nQ.$$

But from eq. (16) we have that $\lim_{Q\rightarrow 0} nQ = l/C$, and so

$$T_p(l) = \frac{l}{C} \sum_{i=1}^{P} \frac{g_i}{g_p} (E_i + \delta_{ip})$$

or

$$T_p(l) = \frac{l}{C}\left(1 + \sum_{i=1}^{P} \frac{g_i}{g_p} E_i\right). \tag{31}$$

We must now evaluate E_i, the expected number of type-i customers in the system. We make use of Little's [7] result, which states that, in general, the expected number of units in a queueing system which has reached equilibrium is equal to the product of the average input rate of these units to the system and the expected time spent by these units in the system. His result holds for priority systems as well, and so we have the set of P linear simultaneous equations in the E_i, namely,

$$E_p = \lambda_p T_p, \quad p = 1, 2, \cdots, P,$$

where T_p is the average time that type-p units spend in the system. By definition,

$$T_p = \int_0^\infty p(l) T_p(l)\, dl.$$

From eqs. (3) and (31) we obtain

$$T_p = \frac{1}{\mu_p C}\left(1 + \sum_{i=1}^{P} \frac{g_i}{g_p} E_i\right) \tag{32}$$

and so we must solve

$$E_p = \frac{\lambda_p}{\mu_p C}\left(1 + \sum_{i=1}^{P} \frac{g_i}{g_p} E_i\right) \quad p = 1, 2, \cdots, P$$

or

$$E_p = \rho_p\left(1 + \sum_{i=1}^{P} \frac{g_i}{g_p} E_i\right) \quad p = 1, 2, \cdots, P. \tag{33}$$

We assert that the solution to the set of eqs. (33) is given by eq. (18), namely,

$$E_p = \frac{\rho_p}{1-\rho}\left[1 + \sum_{i=1}^{P}\left(\frac{g_i}{g_p} - 1\right)\rho_i\right]. \tag{18}$$

We check this assertion by substituting eq. (18) in eq. (33) and testing the (conjectured) identity

$$\frac{\rho_p}{1-\rho}\left[1 + \sum_{i=1}^{P}\left(\frac{g_i}{g_p} - 1\right)\rho_i\right] = \rho_p\left\{1 + \sum_{i=1}^{P} \frac{g_i}{g_p}\left(\frac{\rho_i}{1-\rho}\right)\left[1 + \sum_{j=1}^{P}\left(\frac{g_j}{g_i} - 1\right)\rho_j\right]\right\}$$

or

$$1 + \sum_{i=1}^{P} \left(\frac{g_i}{g_p} - 1\right) \rho_i = 1 - \rho + \sum_{i=1}^{P} \frac{g_i \rho_i}{g_p} \left[1 + \sum_{j=1}^{P} \left(\frac{g_j}{g_i} - 1\right) \rho_j\right]$$

or

$$\sum_{i=1}^{P} \frac{g_i \rho_i}{g_p} = \sum_{i=1}^{P} \frac{g_i \rho_i}{g_p} + \sum_{i=1}^{P} \frac{g_i \rho_i}{g_p} \sum_{j=1}^{P} \frac{g_j \rho_j}{g_i} - \sum_{i=1}^{P} \frac{g_i \rho_i}{g_p} \rho$$

or

$$0 = \sum_{i=1}^{P} \rho_i \sum_{j=1}^{P} g_j \rho_j - \rho \sum_{i=1}^{P} g_i \rho_i$$

$$0 = 0,$$

thus establishing the required identity and validating the asserted solution for E_p.

We must now evaluate (for subsequent use),

$$\begin{aligned}\sum_{i=1}^{P} g_i E_i &= \sum_{i=1}^{P} g_i \left(\frac{\rho_i}{1-\rho}\right) \left[1 + \sum_{j=1}^{P} \left(\frac{g_j}{g_i} - 1\right) \rho_j\right] \\ &= \sum_{i=1}^{P} \frac{g_i \rho_i}{1-\rho} + \sum_{i=1}^{P} \frac{g_i \rho_i}{1-\rho} \sum_{j=1}^{P} \left(\frac{g_j}{g_i} - 1\right) \rho_j \\ &= \sum_{i=1}^{P} \frac{g_i \rho_i}{1-\rho} + \sum_{i=1}^{P} \frac{\rho_i}{1-\rho} \sum_{j=1}^{P} g_j \rho_j - \sum_{i=1}^{P} g_i \rho_i \sum_{j=1}^{P} \frac{\rho_j}{1-\rho}.\end{aligned}$$

Thus

$$\sum_{i=1}^{P} g_i E_i = \sum_{i=1}^{P} \frac{g_i \rho_i}{1-\rho}. \tag{34}$$

We now substitute eq. (34) in eq. (31) to obtain finally

$$T_p(l) = \frac{l}{C} \left[1 + \sum_{i=1}^{P} \frac{g_i \rho_i}{g_p(1-\rho)}\right],$$

thus establishing eq. (17) and concluding the proof of Theorem 5.

In Section 4, this priority processor-shared model is compared with the other two models studied.

For completeness, we also consider a strict first-come-first-served system with the same input and service requirements as in our priority model. To this end, we have

THEOREM 6. *The first-come-first-served system with a priority input yields, for customers with l required operations, a total expected time in system as follows:*

$$T(l) = \frac{l}{C} + \frac{\rho/\mu C}{1-\rho}, \tag{35}$$

where

$$\frac{1}{\mu C} = \frac{\rho}{\sum_{p=1}^{P} \lambda_p}. \tag{36}$$

PROOF. The result follows directly from classical queueing theory results. We proceed as follows. Upon entering, our tagged unit finds F_p customers from priority group p present in the system, each of which will take, on the average $1/\mu_p C$ seconds of service. His own time in service will be exactly l/C seconds; thus,

$$T(l) = \frac{l}{C} + \sum_{p=1}^{P} \frac{F_p}{\mu_p C}. \tag{37}$$

But from Little [7] we have

$$\lambda_p T_p = F_p. \tag{38}$$

But for the first-come-first-served case, $T_p = T$ for all p. Thus, from eq. (37)

$$T = \int_0^\infty p(l)T(l)\,dl = \frac{1}{\mu C} + \sum_{p=1}^{P} \frac{F_p}{\mu_p C}, \tag{39}$$

where

$$p(l) = \sum_{p=1}^{P} \frac{\lambda_p}{\lambda} \mu_p e^{-\mu_p l}$$

and

$$\lambda = \sum_{i=1}^{P} \lambda_i,$$

giving

$$\frac{1}{\mu C} = \sum_{p=1}^{P} \frac{\lambda_p}{\lambda C} \mu_p \int_0^\infty l e^{-\mu_p l}\,dl$$
$$= \sum_{p=1}^{P} \frac{\lambda_p}{\lambda C \mu_p} = \frac{\rho}{\lambda},$$

where ρ is defined as in Theorem 5. This establishes eq. (36). Thus, eq. (38) becomes

$$\lambda_p\left(\frac{1}{\mu C} + \sum_{i=1}^{P} \frac{F_i}{\mu_i C}\right) = F_p, \quad p = 1, 2, \cdots, P. \tag{40}$$

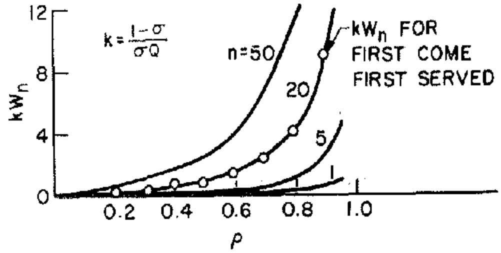

FIG. 5. $[(1 - \sigma)/(\sigma Q)]W_n$ for the late-arrival round-robin system ($\sigma = 19/20$)

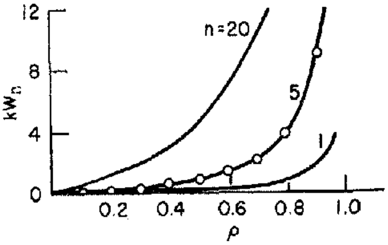

FIG. 6. $[(1 - \sigma)/(\sigma Q)]W_n$ for the late-arrival round-robin system ($\sigma = 4/5$)

The solution to this set of equations is readily seen to be

$$F_p = \frac{\lambda_p}{\lambda}\left(\frac{\rho}{1-\rho}\right),$$

which when substituted into eq. (37) gives

$$\begin{aligned} T(l) &= \frac{l}{C} + \sum_{p=1}^{P} \frac{\lambda_p}{\lambda\mu_p C}\frac{\rho}{1-\rho} \\ &= \frac{l}{C} + \frac{\rho^2}{\lambda(1-\rho)} \\ &= \frac{l}{C} + \frac{\rho/\mu C}{1-\rho}, \end{aligned}$$

which completes the proof of Theorem 6.

We note that for $P = 1$ we have the (nonpriority) processor-shared system.

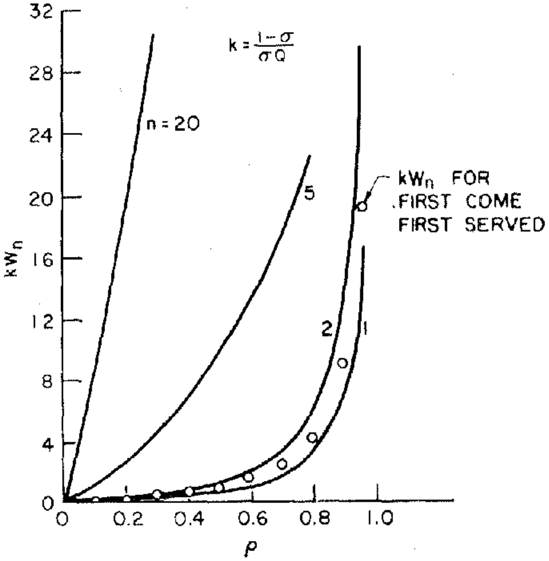

FIG. 7. $[(1 - \sigma)/(\sigma Q)]W_n$ for the late-arrival round-robin system ($\sigma = 1/5$)

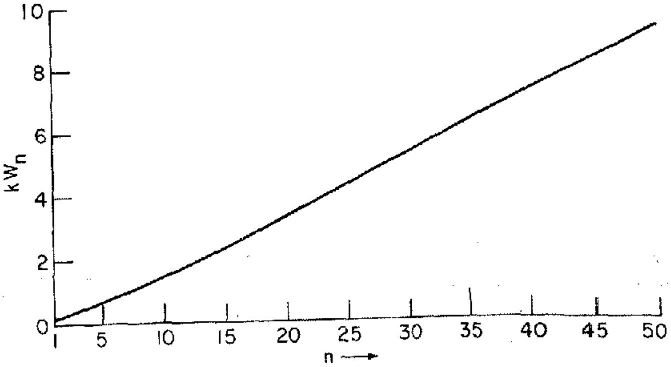

FIG. 8. $[(1 - \sigma)/\sigma Q]W_n$ for the late-arrival round-robin system as a function of n ($\rho = 1/2$, $\sigma = 4/5$)

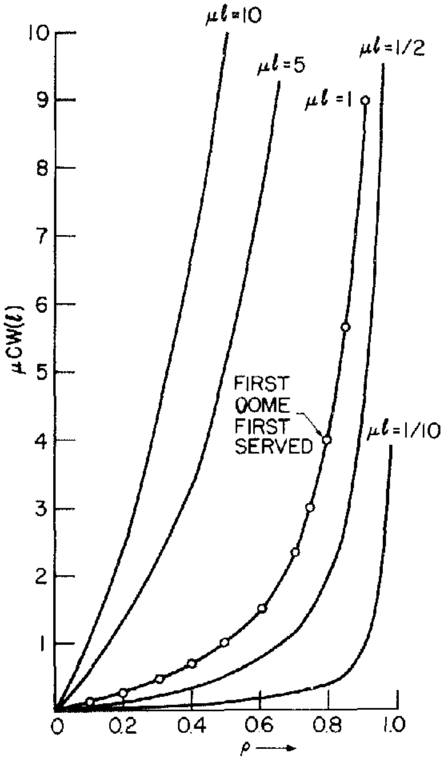

FIG. 9. Processor-shared system: performance as a function of ρ for various μl

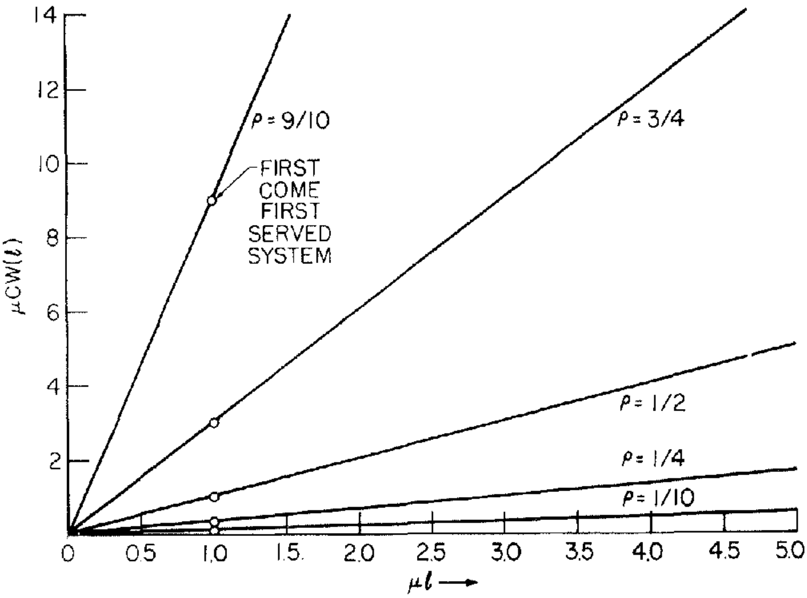

FIG. 10. Processor-shared system: performance as a function of μl for various ρ

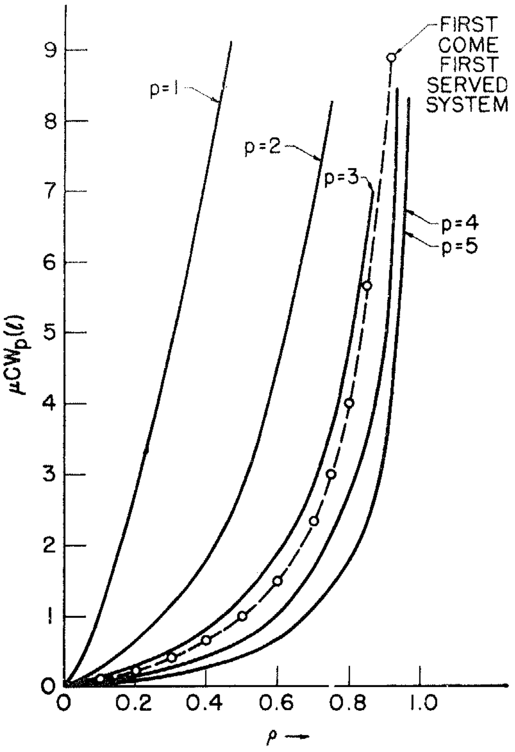

FIG. 11. Priority processor-shared system: performance as a function of ρ for $g_p = p^2$ ($p = 1, 2, 3, 4, 5$), $\mu_p = \mu$, $\lambda_p = \lambda/P$, and $\mu l = 1$

4. *Discussion, Examples, and Comparison of the Systems*

Having considered three models of time-shared systems, we now wish to compare their performance among themselves as well as with the first-come-first-served systems. The basis of comparison will be the average conditional *additional* delay experienced by a customer (conditioned on his required processing as well as on his priority). We define the additional delay as the difference between the time that a customer spends in the time-shared system and the time he would spend in the system if no other customers were present (in a first-come-first-served model, this is merely his time in queue); i.e., let

$W_p(l)$ = average additional delay experienced by a customer from priority group p who requires l operations in service (obvious analogous definition for $W_p(n)$ and $W(n)$ in the $Q > 0$ case).

We have[7]

$$W_p(l) = T_p(l) - \frac{l}{C}. \tag{41}$$

In the most general model, we wish to display curves of $W_p(l)$ as a function of l

[7] Obviously, for $Q > 0$ we have $W_p(n) = T_p(n) - nQ$.

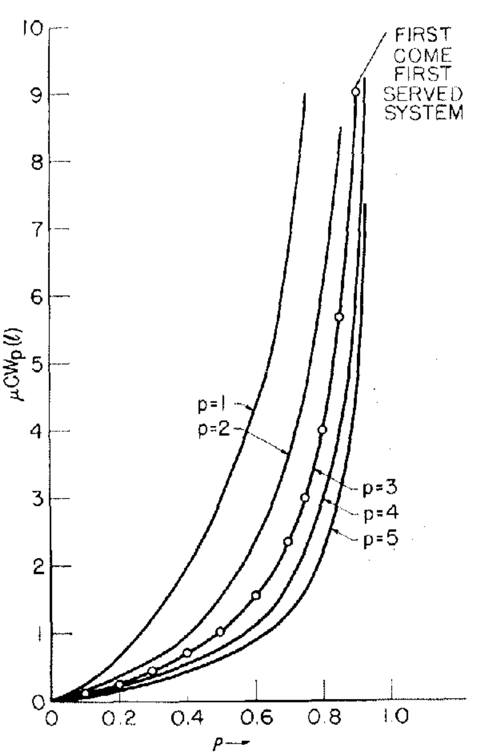

FIG. 12. Priority processor-shared system: performance as a function of ρ for $g_p = p$ ($p = 1, 2, 3, 4, 5$), $\mu_p = \mu$, $\lambda_p = \lambda/P$, and $\mu l = 1$

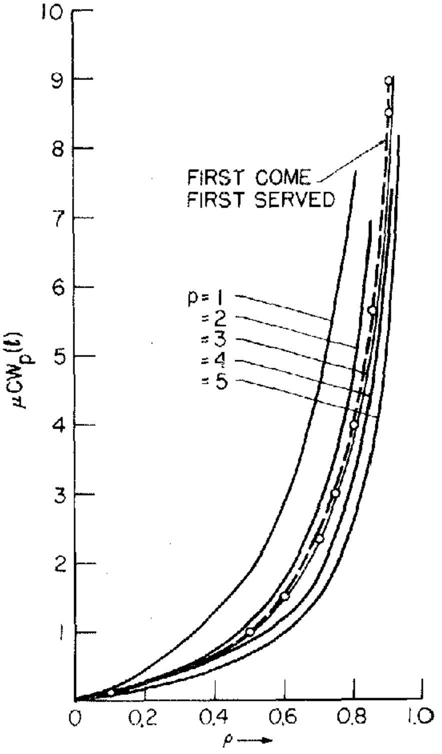

FIG. 13. Priority processor-shared system: performance as a function of ρ for $g_p = \log_2 (1 + p)$ ($p = 1, 2, 3, 4, 5$), $\mu_p = \mu$, $\lambda_p = \lambda/P$, $\mu l = 1$

and as a function of ρ with p as a parameter. Furthermore, we choose to plot

$$\frac{1 - \sigma_p}{\sigma_p Q} W_p(n)$$

rather than $W_p(n)$ for purposes of a convenient normalization, which, in the case for $Q \to 0$ becomes $\mu_p CW_p(l)$. Below we present these curves for various examples.

Round-Robin System. In Figures 5–7, curves[8] of $[(1 - \sigma)/\sigma Q]W_n \equiv kW_n$ are plotted to show the general behavior of the round-robin structure for the late-arrival system. On each graph, (circled) points corresponding to the first-come-first-served case have also been included. The normalization $(1 - \sigma)/\sigma Q$ used is such that for the first-come-first-served case we obtain the curve $\rho/(1 - \rho)$, which is a function only of ρ.

Figures 5–7 indicate the accuracy of the approximation discussed above, in which the crossover point for waiting times is at the mean number of service intervals, $1/(1 - \sigma)$. In Figures 5 and 6 there is no noticeable difference (on the scale used)

[8] These are the same curves as in Kleinrock [6]. In these curves, ρ was varied by fixing σ and varying λQ (recall $\rho - \lambda Q/(1 - \sigma)$).

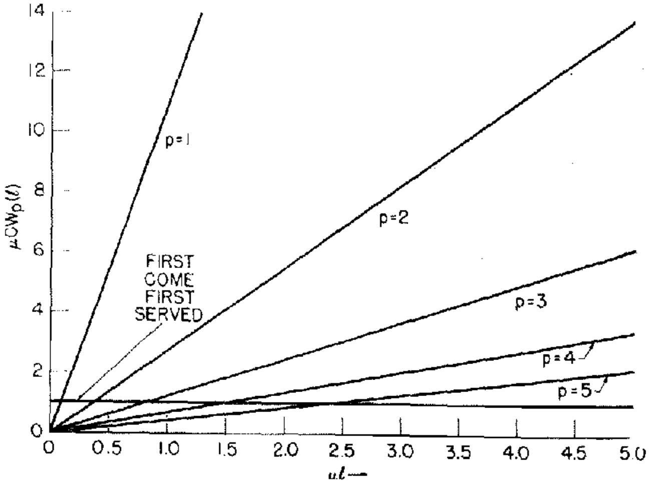

FIG. 14. Priority processor-shared system: performance as a function of μl for $g_p = p^{22}$ $(p = 1, 2, 3, 4, 5)$, $\mu_p = \mu$, $\lambda_p = \lambda_p = \lambda/P$, and $\rho = 1/2$

between the first-come-first-served points and the curve for $n = 1/(1 - \sigma)$; moreover, in Figure 7 the points fall between the curves for $n = 1$ and $n = 2$, since $1/(1 - \sigma) = 1.25$.

In Figure 8, we plot kW_n as a function of n for $\rho = \frac{1}{2}$, $\sigma = \frac{4}{5}$. In all these curves (Figures 5–8) we observe that by introducing the round-robin system, one manipulates the relative waiting time for different jobs and thus imposes a method of time-sharing which gives preferential treatment to *short* jobs.

Processor-shared System. In Figures 9 and 10, we plot $\mu CW(l)$ as a function of ρ (for various μl) and as a function[9] of μl (for various ρ), respectively.

In Figure 10, the circles indicate the values of $\mu CW(l)$ for the strict first-come-first-served system (see Theorem 6). Again we see the preferential treatment given to shorter jobs, and again we see that the break-even point for jobs is the average job length $(\mu l = 1)$.

Priority Processor-shared Model. For these curves, we let $\mu_p = \mu$, $\lambda_p = \lambda/P$, $P = 5$ for $p = 1, 2, \cdots, 5$. In Figures 11–13 we show $\mu CW_p(l)$ as a function of ρ for various p and for $\mu l = 1$. Figure 11 is for $g_p = p^2$, Figure 12 is for $g_p = p$, and Figure 13 is for $g_p = \log_2 (p + 1)$. In each of these figures, the circles correspond to the strict first-come-first-served system (which compares the treatment as a function of p for the two systems).

In Figures 14–16 we show $\mu CW_p(l)$ as a function of μl for various p and for $\rho = 1/2$. Again $g_p = p^2$, $g_p = p$, and $g_p = \log_2 (p + 1)$ for Figures 14, 15, and 16, respectively. In each of these figures, the circles correspond to the behavior of a first-come-first-served system. (On these axes, it is a constant additional delay, independent of μl.)

In both processor-shared models, $W_p(l)$ approaches zero as $\rho \to 0$ for all l and p.

In all the curves presented, we see that the effect of introducing a time-sharing discipline is to reduce the average waiting time for customers with "short" service

[9] $\mu l = l/(1/\mu)$ is the length of a job normalized with respect to its average length.

(processing) requirements at the expense of those customers with "longer" service requirements. For the nonpriority cases (i.e., the first two models studied), we observe that customers with service (processing) requirements less (greater) than the average requirement spend, on the average, less (greater) time in the system, compared with a strict first-come-first-served system.

In the priority processor-shared system, we see a similar trend (i.e., short jobs wait less than long jobs), and, in addition, we give preferential treatment (shorter waiting) to certain select high-priority groups. The effect now is that for job lengths below some critical value (dependent upon p, the priority group) a customer does

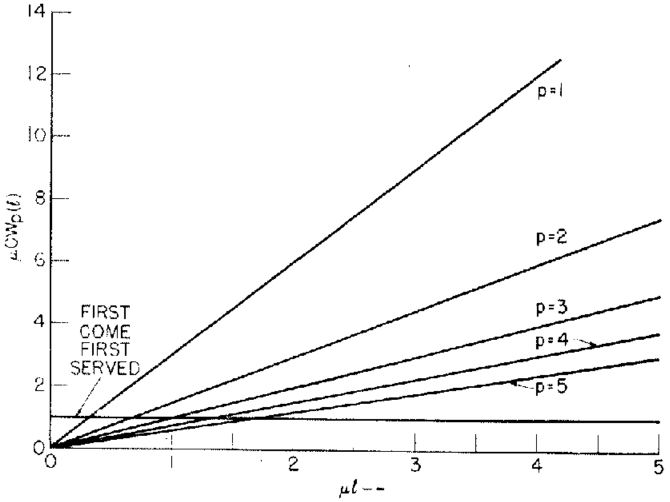

FIG. 15. Priority processor-shared system: performance as a function of μl for $g_p = p$ $(p = 1, 2, 3, 4, 5)$, $\mu_p = \mu$, $\lambda_p = \lambda/P$, and $\rho = 1/2$

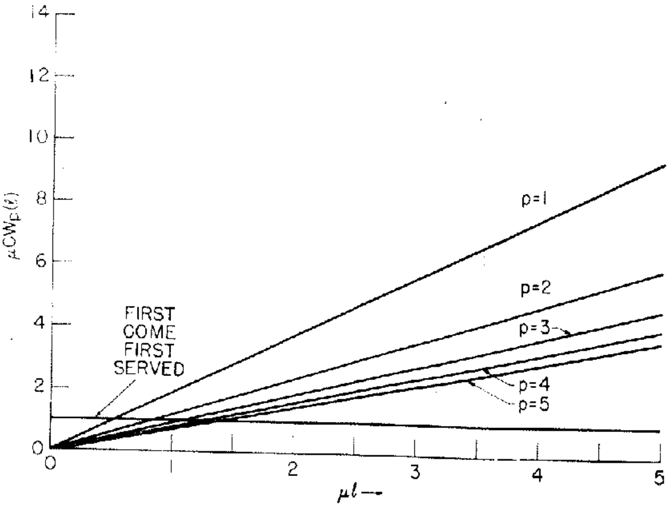

FIG. 16. Priority processor-shared system: performance as a function of μl for $g_p = \log (1 + p)$ $(p = 1, 2, 3, 4, 5)$, $\mu_p = \mu$, $\lambda_p = \lambda/P$, and $\rho = 1/2$

better (waits less) in the time-shared system than in a first-come-first-served system. This critical length is monotonically increasing with p. The degree and manner in which the different priority groups receive treatment depends upon the function g_p and may be varied over a considerable range of relative performance.

Conclusion

In this paper, we have considered several models of time-shared processing systems. These models provide the basic features desired in such systems, namely, rapid service for short jobs and the virtual appearance of a (fractional capacity) processor available on a full-time basis.

The most general model, the priority processor-shared system, not only provides the above features but also allows the population of customers to be divided into priority classes where the higher priority groups receive preferential treatment compared with the lower priority groups.

The assumption of zero swap-time results in models which provide the best possible performance of such time-shared systems. Comparison of these systems with the strict first-come-first-served systems showed the relative improvement (or deterioration) of performance as a function of service requirement and priority group.

REFERENCES

1. Fano, R. M. The MAC system: the computer utility approach. *IEEE Spectrum 2*, No. 1 (Jan. 1965), 56–64.
2. Lichtenberger, W. W. and Pirtle, M. W. A facility for experimentation in man-machine interactions. Proc. AFIPS 1965 Fall Joint Comput. Conf., Vol. 27, Pt. I, 1965, pp. 589–598.
3. Forgie, J. W. A time- and memory-sharing executive program for quick response on-line applications. Proc. AFIPS 1965 Fall Joint Comput. Conf., Vol. 27, Pt. 1, 1965, pp. 599–609.
4. McCarthy, J. Time-sharing computer systems. In *Management and the Computer of the Future*, M. Greenberger, Ed. MIT Press, Cambridge, Mass., 1962, pp. 221–236.
5. Schwartz, J. I., Coffman, E. G. and Weissman, C. A general purpose time-sharing system. Proc. AFIPS, 1962 Spring Joint Comput. Conf., 1962, pp. 335–344.
6. Kleinrock, L. Analysis of a time-shared processor. *Naval Res. Logistics Quart. 11*, 10 (March 1964), 59–73.
7. Little, J. D. C. A proof of the queueing formula $L = \lambda W$. *Operations Res. 9* (1961), 383–387.

received may, 1966