Computer Sciences Department
The University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

ANALYSIS OF TWO TIME-SHARING QUEUEING MODELS

by

Harry C. Heacox, Jr.
and
Paul W. Purdom, Jr.

Technical Report #119

March 1971

Analysis of Two Time-Sharing Queueing Models

by

Harry C. Heacox, Jr.
and
Paul W. Purdom, Jr.

Computer Sciences Department
University of Wisconsin
Madison, Wisconsin

Abstract. Two time-sharing models are described. One is the conventional round-robin model in which each customer receives at most q seconds of service at a time. If this completes his service requirement, he leaves the system; otherwise he joins the end of the queue to await his next turn. The second model is a modification of the round-robin system in which the amount of service per pass depends on the rate at which programs arrive in the system. The models are analyzed under the assumption of constant, non-zero overhead when the processor swaps one program for another. Expressions are derived for the mean waiting time in queue as a function of service requirement and for the mean system cost due to waiting time in queue.

Key Words and Phrases: time-shared system, scheduling algorithms, queueing theory, multiprogramming systems, cost effectiveness, round robin.

CR Categories: 4.32, 4.39, 5.5

Technical Report #119

ANALYSIS OF TWO TIME-SHARING QUEUEING MODELS

by

Heacox and Purdom

ERRATA

1. In the expression for β in the middle of page 18, the term "$\lambda[(1 - \delta/\mu + s]$" should read "$\lambda[(1 - \delta)/\mu + s]$".

2. The same error occurs in the expression for $W_{SQ}(k)$, equation (26), page 22. The term "$2s(1 - \delta/\mu$" should be "$2s(1 - \delta)/\mu$".

3. In the first equation on page 23, the term "$2s(\lambda - \delta)/\mu$" should be "$2s(1 - \delta)/\mu$".

4. Equation (28), page 23, should be

$$\lim_{q \to \infty} W_{SQ}(k) = \frac{\lambda/\mu}{1 - \rho} \left(\frac{1}{\mu} + s + \frac{\mu s^2}{2} \right)$$

5. Equation (42), page 31, should be

$$\lim_{q \to \infty} W_{SQ}(k) = \frac{\lambda/\mu}{1 - \rho} \left(\frac{1}{\mu} + s + \frac{\mu s^2}{2} \right)$$

INTRODUCTION

In recent years there has been considerable interest in various scheduling algorithms for time-shared computing systems. The literature contains many discussions of such algorithms, particularly the round-robin [3-6] and multiple level [4,6] schemes. However, for the most part, these efforts have neglected the "swap time" or overhead incurred when the processor switches its attention from one program to another. When the load on the system is light and the processor is idle much of the time, the effects of overhead are not too significant. However, when the system is heavily loaded, this overhead becomes important from the point of view of increasing the waiting time of programs in the system. The purpose of this paper is to examine two particular scheduling algorithms under the assumption of non-zero swap time.

The first model is the conventional round-robin discipline in which a program entering the system joins a queue and waits for its turn to be served by the single central processor. Time is allocated in relatively small quanta and, if the program does not complete its service requirement during its allocated quantum, it is placed at the end of the queue to await its next turn. This will be called the single-quantum (SQ) model. This model has been studied by Rasch [3], with results which differ somewhat from ours. The discrepancy and the reasons for it will be discussed in a later section.

The second model is a modification of the round-robin discipline which is due to Coffman [1]. As pointed out above, when the system is heavily loaded, as when the arrival rate of programs is high, the overhead inherent in the round-robin system causes increasingly serious degradation of performance from the standpoint

of the length of time programs must wait in the queue. To alleviate this problem, it is desirable to reduce the amount of time the system spends in swapping during periods of high arrival rates. In the Coffman model, if there is a new arrival during a quantum service and the program in service does not finish its processing requirement during the quantum, it is given an additional quantum. Arrivals during swaps have no effect on quantum allocation. Thus, a program operates until it completes its processing or runs for a complete quantum during which there are no new arrivals. Obviously, during periods of high arrival rates, this algorithm has the desired effect of reducing the system's swapping activities. Conversely, during periods of low arrival rate the model is quite similar to the conventional round-robin. This algorithm will be referred to as the multiple-quantum (MQ) model.

PRELIMINARY RESULTS

At this point we establish some quantities to be used in the later analysis. For both models we assume a Poisson input process, with the interarrival time distributed according to

$$A(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \qquad (1)$$

Then the mean interarrival time is $1/\lambda$ seconds, and the mean arrival rate is λ programs per second.

We assume an exponential distribution of service requirements given by

$$B(t) = \begin{cases} 1 - e^{-\mu t} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{2}$$

Thus the mean service requirement (exclusive of swap time) is $1/\mu$ seconds.

We are dealing with a continuous time model in which a program departs from the system as soon as its service requirement is satisfied. Then the amount of time actually used during a quantum; q, is distributed according to

$$F(t) = \begin{cases} 0 & t < 0 \\ 1 - e^{-\mu t} & 0 \leq t < q \\ 1 & t \geq q \end{cases} \tag{3}$$

The first two moments of F(t) are easily found:

$$E(t) = \int_{t=0}^{\infty} t dF(t) = (1 - e^{-\mu q})/\mu \tag{4}$$

$$E(t^2) = 2[1 - e^{-\mu q}(\mu q + 1)]/\mu^2 \tag{5}$$

We define a "loading factor," $\rho = \lambda E(\tau)$, where τ is the total service requirement (including swap time) of a program. This is just the ratio of the mean service requirement and the mean interarrival time. Clearly for $\rho \geq 1$ the system will saturate. That is, the queue length and waiting time will become arbitrarily large.

The probability that a program will not complete its processing during a quantum is given by

$$\delta = \int_{t=q}^{\infty} dB(t) \qquad \text{where } B(t) \text{ is given by (2)} \tag{6}$$

$$= e^{-\mu q}$$

For the MQ system we also wish to know the probability of an arrival during a quantum q. This is just the probability that the interarrival time is less than q and is given by

$$\gamma = \int_{t=0}^{q} dA(t) \qquad \text{where } A(t) \text{ is given by (1)}$$
$$= 1 - e^{-\lambda q} \tag{7}$$

Now, if a program has a processing requirement of t seconds, it will require k quanta, where $k - 1 < t/q \leq k$. Finally, we assume that the swap time is a constant, s seconds, and that the overhead is incurred when the program enters the processor, i.e. at the beginning of each quantum service.

We will discuss two parameters as measures of system efficiency. The first is $W(k)$, the mean waiting time in queue as a function of number of quanta required. $W(k)$ does not include the program's processing time or swap time, so that all programs requiring k quanta have the same waiting time. Thus, $W(k)$ can be viewed as mean waiting time as a function of processing requirement. Second, we will present a mean system delay cost as discussed by Rasch [3]. According to Rasch, the delay cost of a user with service requirement t and waiting time w is given by we^{-at}. Note that the factor e^{-at} is effectively a priority which can be adjusted by appropriate choice of the parameter a. In particular, for $a > 0$, the cost of delaying a short program for a time w is greater than the cost of delaying a long program for the same length of time. The converse is true for $a < 0$, while for $a = 0$, all program time requirements are treated equally. As we shall see, the expected value of we^{-at}, i.e. the mean system cost, gives us a means of adjusting system

parameters (in particular the quantum size) so as to optimize system performance. By this we mean that the most important programs, defined by the choice of a, will receive preferential treatment without undue degradation of service for the other programs. Following Rasch, we define the mean system cost by

$$\begin{aligned} C &= E(we^{-at}) \\ &= E\left[E(we^{-at}|k)\right] \end{aligned} \tag{8}$$

Since w and t are conditionally independent, given k,

$$C = E\left[E(w|k)\cdot E(e^{-at}|k)\right]$$

Note that this is only true if k is given. Now $E(w|k) = W(k)$, and

$$E(e^{-at}|k) = \int_0^{\infty} e^{-at} f(t,k)\, dt$$

where

$$f(t,k) = \begin{cases} \dfrac{\mu e^{-\mu t}}{e^{-\mu(k-1)q}\,(1 - e^{-\mu q})} & k-1 \leq \dfrac{t}{q} < k \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Thus,

$$C = \sum_{k=1}^{\infty} W(k)\, e^{-\mu(k-1)q}(1 - e^{-\mu q}) \int_0^{\infty} e^{-at} f(t,k)\, dt$$

$$= \sum_{k=1}^{\infty} W(k) \int_{(k-1)q}^{kq} \mu e^{-(\mu+a)t}\, dt \tag{10}$$

$$= \frac{\mu}{\mu+a}\left[1 - e^{-(\mu+a)q}\right] \sum_{k=1}^{\infty} W(k)\, e^{-(\mu+a)(k-1)q}$$

This completes the preliminary results.

SINGLE-QUANTUM MODEL

For the single-quantum model we have

THEOREM 1. Let W(k) be the mean waiting time in queue for a program requiring k quanta of processing. Then

$$W_{SQ}(k) = \frac{\lambda}{2}\left[s^2 + 2s(1-\delta)/\mu + E(t^2)\right]\frac{1-\rho\beta^{k-1}}{1-\beta} + \rho\delta Q\frac{1-\beta^{k-1}}{1-\beta} + \frac{\rho}{1-\rho}\left\{kQ + [m_1/\lambda - Q/(1-\beta)](1-\beta^k)\right\} \tag{11}$$

where

$\beta = \delta + \rho(1-\delta)$,

q = quantum size,

s = swap time,

$Q = q + s$, and

$m_1 = E(m)$ = mean queue length in equilibrium.

The proof is given in Appendix A.

Corollary 1. The mean system delay cost is given by

$$C_{SQ} = \frac{\mu}{\mu + a}\left\{\frac{\lambda(1-\delta\epsilon)}{2(1-\delta)(1-\beta\epsilon)}\left[s^2 + \frac{2s}{\mu}(1-\delta) + E(t^2)\right] + \frac{\rho m_1}{\lambda}\frac{1-\delta}{1-\beta\epsilon} + \rho Q\epsilon\frac{1-\delta\epsilon}{(1-\epsilon)(1-\beta\epsilon)}\right\} \tag{12}$$

where

$\epsilon = e^{-(\mu + a)q}$

MULTIPLE-QUANTUM MODEL

For the multiple-quantum model we have

THEOREM 2. The mean waiting time in queue in the MQ system for a program requiring k quanta of processing time is

$$W_{MQ}(k) = z_o + \frac{\rho(1-\zeta)}{1-\alpha}\left\{ m_1' + m_1 + \frac{1}{1-\alpha} - \left[m_1' + \alpha\left(m_1 + \frac{1}{1-\alpha} \right) \right] [\gamma + \alpha(1-\gamma)]^{k-1} + (k-1)\gamma - \frac{1-[\gamma+\alpha(1-\gamma)]^k}{(1-\alpha)(1-\gamma)} \right\} \quad (13)$$

where z_o = mean time to complete the service in progress at arrival,

ζ = probability that a program will not complete in a pass

m_1 = E(m) = mean queue length in equilibrium,

m_1' = mean number of programs behind the tagged unit at its first entry into the processor, due to the program in service at arrival, and

γ = probability of an arrival during a quantum q (7).

The proof is given in appendix B.

Corollary 2. the mean system delay cost for the MQ system is given by

$$C_{MQ} = \frac{\mu}{\mu+a}\left\{ z_0 + \rho(1-\zeta)\frac{m_1'\epsilon(1-\gamma) + m_1(1-\gamma\epsilon) + \gamma\epsilon^2(1-\gamma)/(1-\epsilon)}{1-\epsilon[\gamma+\alpha(1-\gamma)]} \right\} \quad (14)$$

where

$$\epsilon = e^{-(\mu+a)q}$$

Discussion of Results

We will compare the SQ and MQ models in three areas: mean number of swaps per program, mean waiting time, and mean system delay cost.

Considering mean number of swaps per program, we have

$$\bar{S}_{SQ} = \int_{t=0}^{\infty} k\,dB(t) \qquad \text{where } B(t) \text{ is given by (2)}$$

$$= 1/(1 - \delta) \qquad \text{where } \delta \text{ is given by (6)}$$

$$\bar{S}_{MQ} = \int_{t=0}^{\infty} \sum_{n=1}^{k} np_n(k)\,dB(t) \qquad \text{where } p_n(k) \text{ is given by (29), appendix A}$$

$$= (1 - \gamma\delta)/(1 - \delta) \qquad \text{where } \gamma \text{ is given by (7)}$$

These results correspond to those obtained by Coffman [1]. Note that the MQ system does indeed reduce the number of swaps per program, as intended.

A comparison of mean waiting time (averaged over all programs) as a function of arrival rate is given in figure 1. Note that for light loads ($\lambda < 0.6$), the two models agree very closely. But as the load increases, they begin to differ markedly. At $\lambda = 0.8$, we have $E[W_{MQ}(k)] \simeq 0.8\ E[W_{SQ}(k)]$. More significantly, the single-quantum system saturates ($\rho = 1$) at $\lambda = 0.887$, while the multiple-quantum system does not reach saturation until $\lambda = 0.954$. Obviously, the MQ system gives substantial improvement in efficiency under near-saturation conditions, although it should be pointed out that it does so at the expense of the advantages of a time-sharing system.

We next consider the behavior of W(k) as a function of arrival rate for various values of k. Figure 2 gives a comparison of the SQ and MQ models, showing W(k) for k = 1 and k = 3. The other parameters are q = 0.5 seconds, s = 0.05 seconds, and $\mu = 1.0\ \text{sec}^{-1}$ (the average program makes two passes). Note that

for programs shorter than average the single-quantum model gives better service at low arrival rates, but that, as expected, at high arrival rates (i.e. high system load) the MQ model is superior. For longer-than-average programs the MQ system is better no matter what the arrival rate.

Figures 3, 4, and 5 show the variation of system delay cost as a function of quantum size for various values of the system parameters. Notice that, for zero overhead, the optimum quantum size in terms of delay cost would be zero, i.e. the processor-shared model investigated by Coffman and Kleinrock [4]. But for non-zero swap time there is a value q_{min} such that, for $q \leq q_{min}$ the system saturates. Depending on the particular parameters, it is usually possible to choose an optimum value for q so as to minimize the mean system cost in the sense of providing the best service for the "high-priority" users. This group is essentially defined by the choice of the cost weighting factor, a. As can be seen in figure 4, the choice of q_{opt} is quite dependent on the value of a. Note also that, in figure 4, the curve for $a = 0$ is just the mean waiting time averaged over all programs.

The single-quantum system with overhead has been discussed by Rasch [3], and figure 6 gives a comparison between his results and ours. Here we plot mean waiting time versus number of quanta for $q = 0.5$ seconds, $s = 0$ and 0.05 seconds, $\mu = 1.0\ \text{sec}^{-1}$, and $\lambda = 0.8\ \text{sec}^{-1}$. As shown the models do not agree, the differences being more pronounced for long programs than for short ones, and for non-zero swap time than for zero swap time. Notice also that for $s = 0$ and $k = 1$ the two models give identical results, while this is not true for $s \neq 0$ and $k = 1$.

This discrepancy can be attributed to a slight difference between the two models, namely that Rasch places the swap at the end of the quantum, while our model puts it at the beginning. As Rasch points out, for $q = \infty$ (i.e. a batch-processing system) his model gives $E(\tau) = 1/\mu$ rather than $1/\mu + s$ as this paper gives. In particular, this means that there is no overhead involved in going from one program to the next if the outgoing program has completed processing.

But this is not the only difference between the two models, as shown by the fact that they do not agree even for $s = 0$ if a program requires more than one quantum. Thus, there is a more serious discrepancy in the models. Rasch assumes that if we exclude the tagged unit, the queue has its equilibrium length at all times. That is, the tagged unit's waiting time on each pass is just the processing time for one mean queue length. But this is only true for the first pass. For subsequent passes we must take into account arrivals during the processing of the tagged unit, thus increasing the length of the queue and the tagged unit's waiting time. In effect, the arrival of the tagged unit perturbs the system and it requires some time to return to equilibrium.

Finally, it should be pointed out that the models presented here have been extensively tested by comparing them with computer simulations. These consisted of simulating the execution of 50,000 programs per case, with arrival rates varying from 0.1 sec^{-1} to 0.8 sec^{-1}, for both zero and non-zero swap time. For arrival rates greater than 0.8 sec^{-1} the rounding errors in the simulator become appreciable. Arrival times and service requirements were obtained by means of a random number generator using a Tausworthe generator to fill a 64-member array and a linear

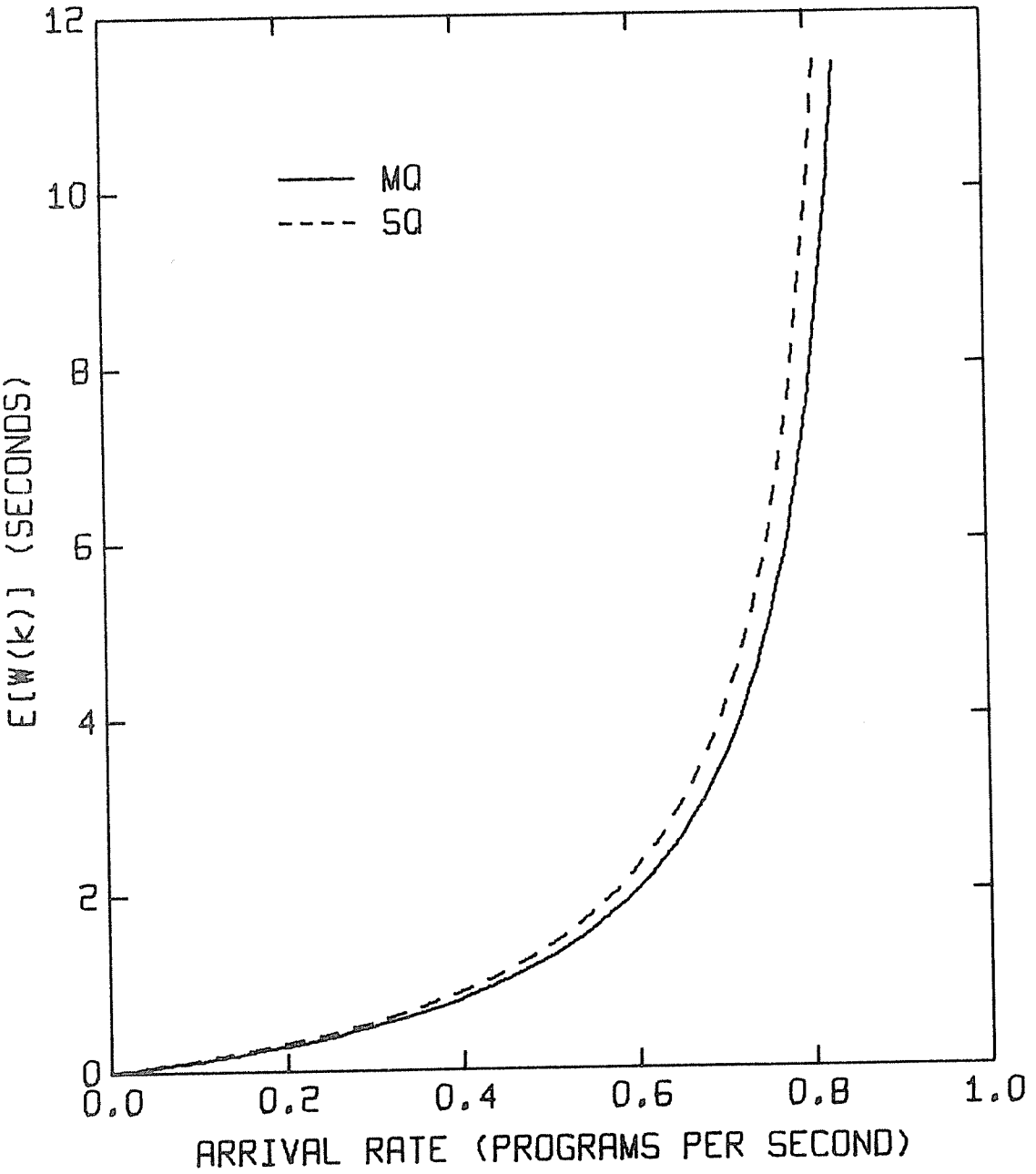

FIGURE 1. WAITING TIME IN QUEUE AVERAGED OVER ALL PROGRAMS FOR THE SQ AND MQ MODELS. $q = 0.5$ SECONDS, $s = 0.05$ SECONDS, AND $\mu = 1.0$ SEC^{-1}.

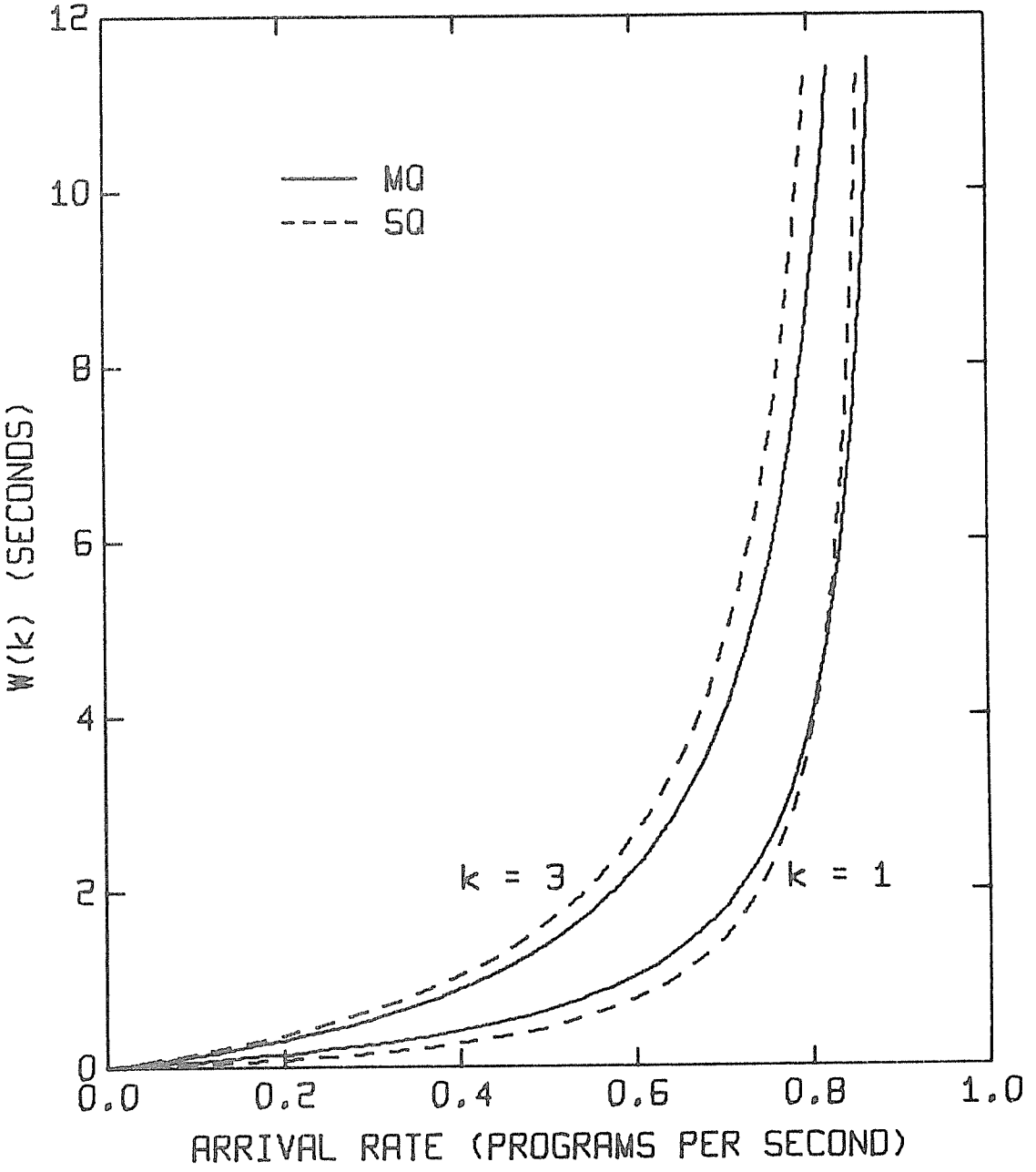

FIGURE 2. $W_{SQ}(k)$ AND $W_{MQ}(k)$ VS. ARRIVAL RATE FOR $k = 1$ AND $k = 3$. $q = 0.5$ SECONDS, $s = 0.05$ SECONDS, AND $\mu = 1.0$ SEC^{-1}.

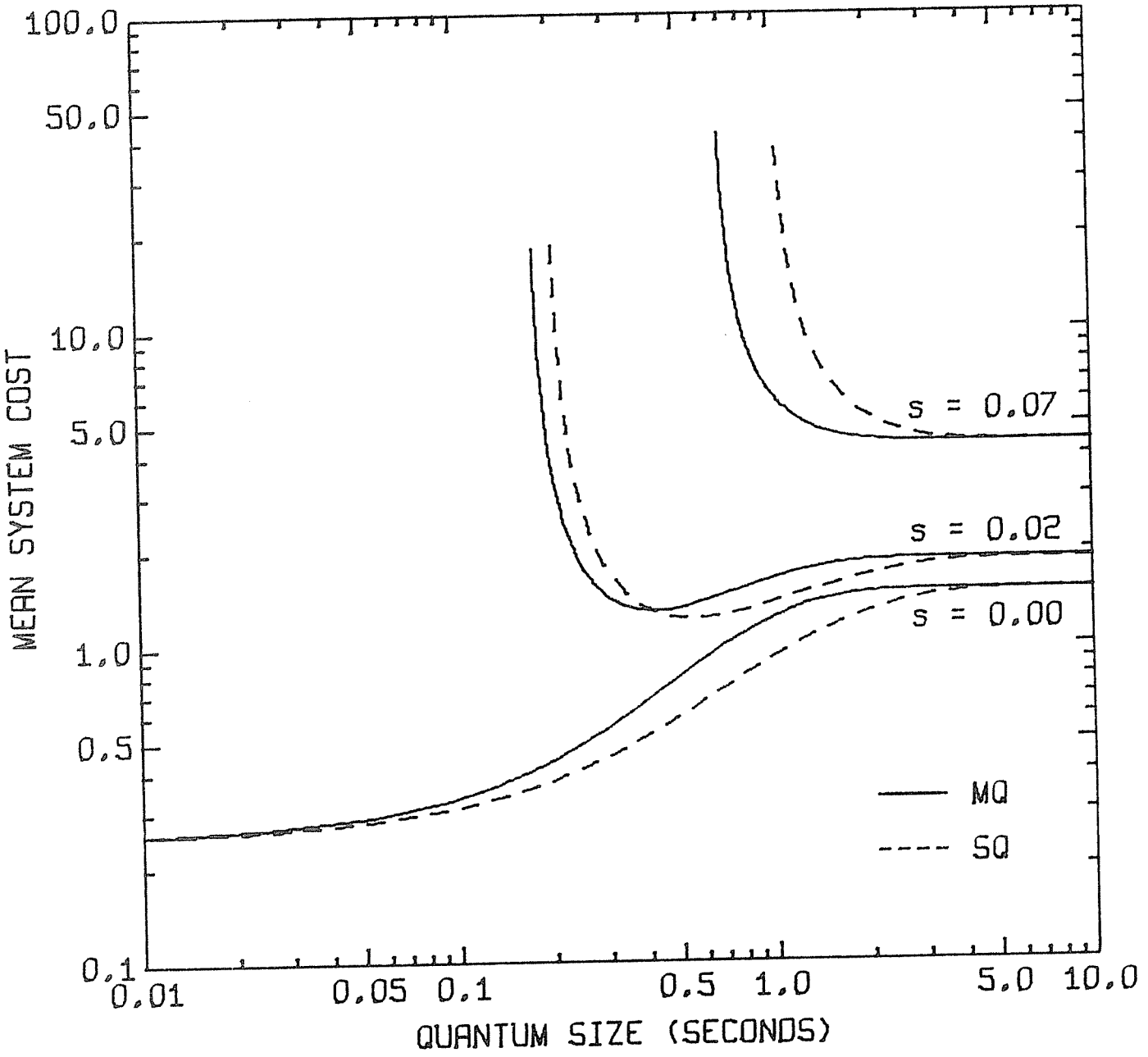

FIGURE 3. MEAN SYSTEM COST AS A FUNCTION OF QUANTUM SIZE AND SWAP TIME FOR THE SQ AND MQ MODELS. λ = 0.9 SEC^{-1}, μ = 1.0 SEC^{-1}, AND a = 5.0.

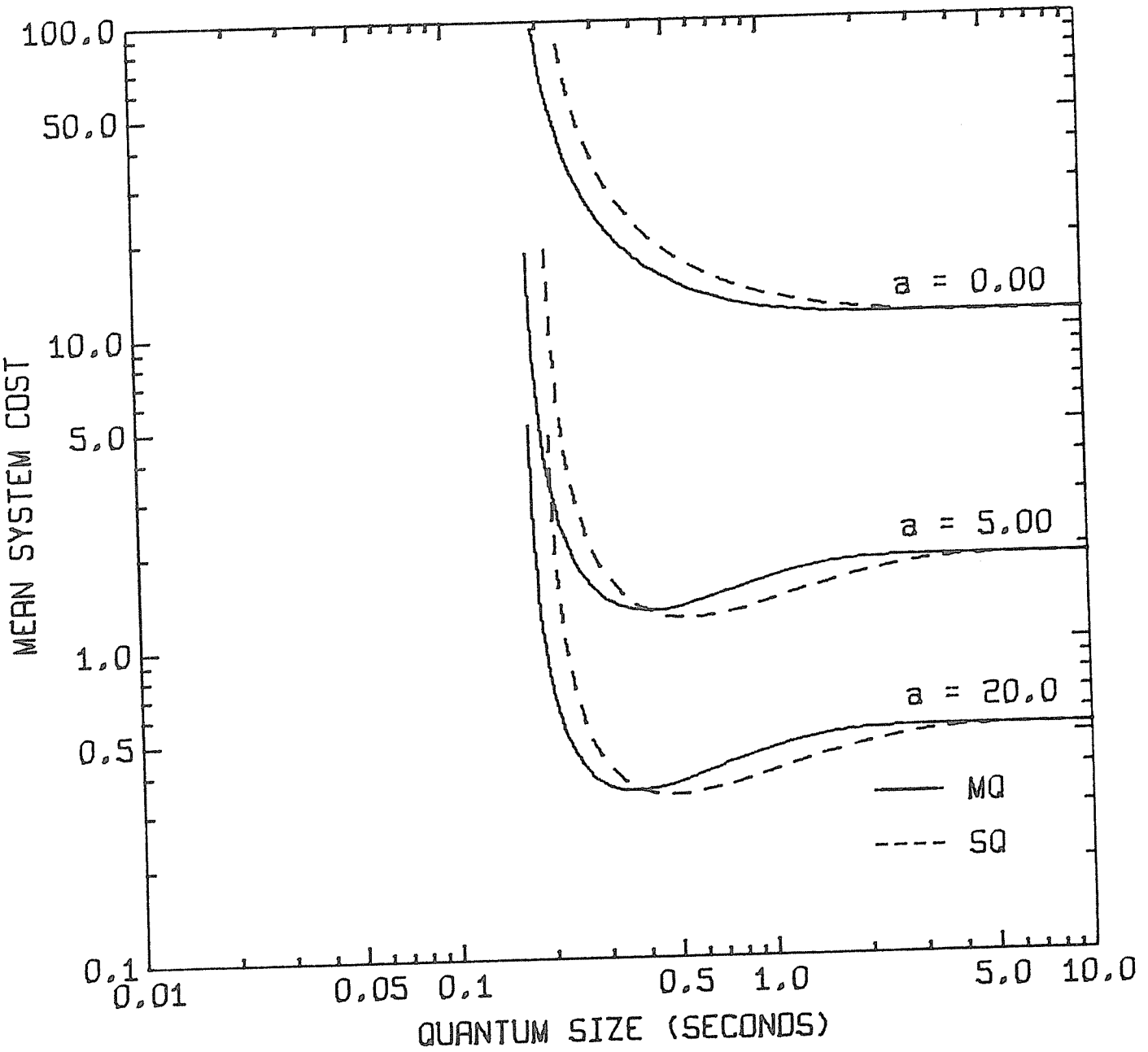

FIGURE 4. MEAN SYSTEM COST AS A FUNCTION OF QUANTUM SIZE AND COST WEIGHTING FACTOR FOR THE SQ AND MQ MODELS. λ = 0.9 SEC^{-1}, μ = 1.0 SEC^{-1}, AND s = 0.02 SECONDS.

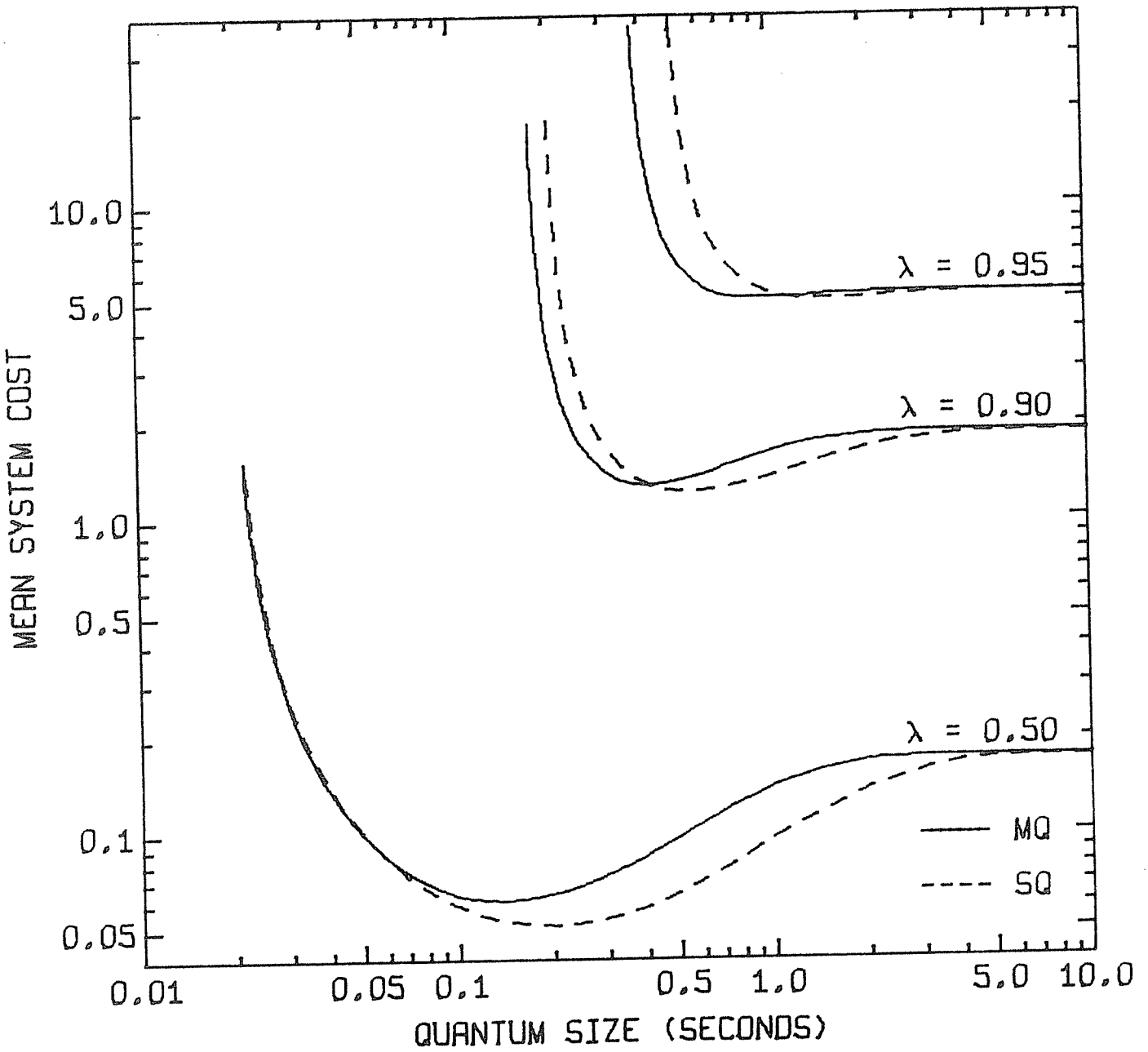

FIGURE 5. MEAN SYSTEM COST AS A FUNCTION OF QUANTUM SIZE AND ARRIVAL RATE FOR THE SQ AND MQ MODELS. μ = 1.0 SEC^{-1}, s = 0.02 SECONDS, AND a = 5.0.

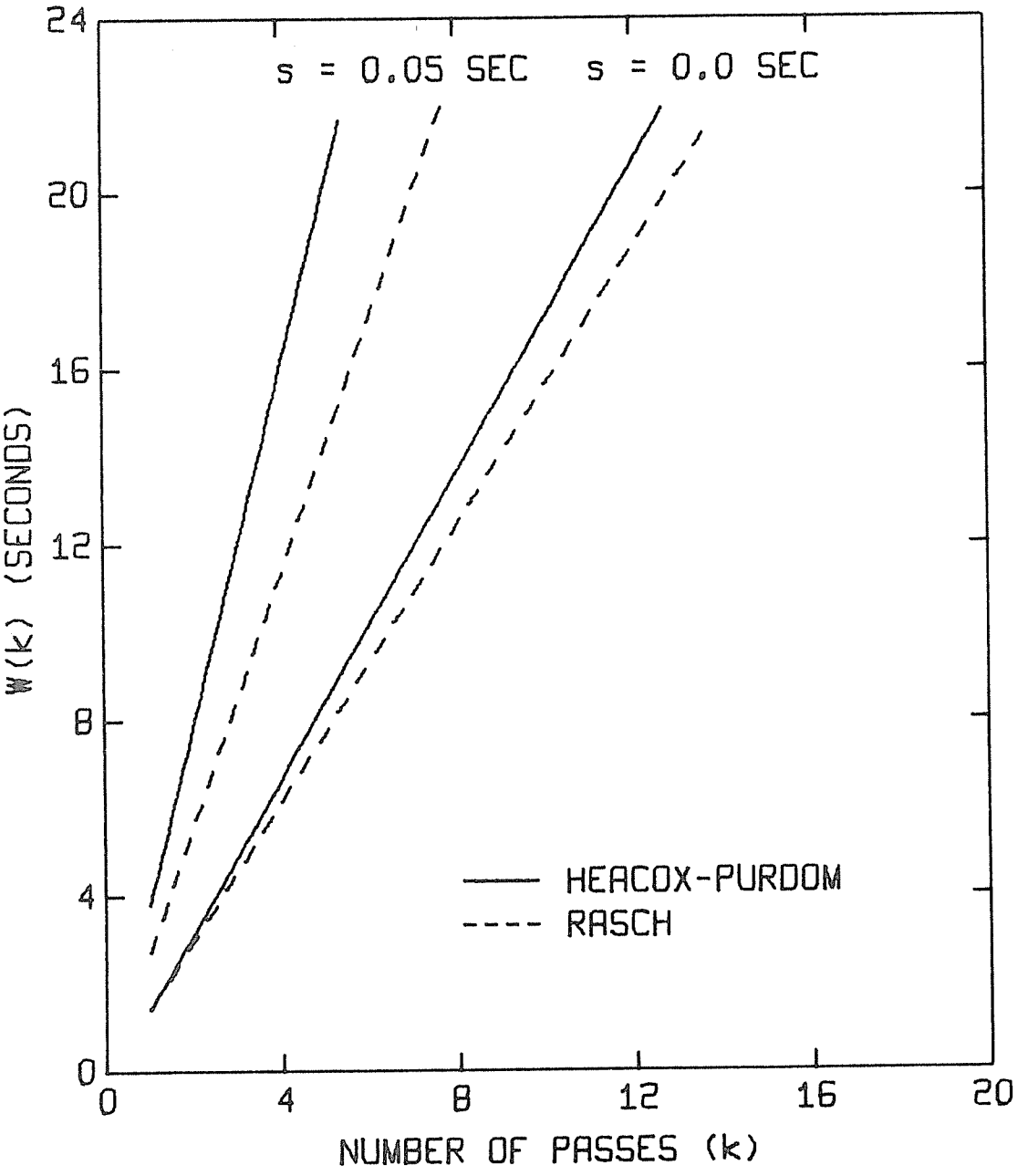

FIGURE 6. W(k) VS. NUMBER OF PASSES FOR THE HEACOX-PURDOM AND RASCH SINGLE-QUANTUM MODELS. $q = 0.5$ SECONDS, $\lambda = 0.8$ SEC^{-1}, $\mu = 1.0$ SEC^{-1}.

congruential (mod 2^{23}) generator to index the array. In almost all the cases tested, differences between the models and the simulations have been less than 5 percent.

APPENDIX A. Proof of Theorem 1.

Using the method of Coffman and Kleinrock [4], we consider a "tagged unit" requiring k quanta of service which arrives with the system in equilibrium. In considering its waiting time in queue, we break the time up into two parts, T_1 and T_2. A program contributes to T_1 if it was being swapped or processed when the tagged unit arrived, or if it arrives during swapping or processing of a program contributing to T_1. A program contributes to T_2 if it was waiting in queue at arrival of the tagged unit, or if it arrives during swapping or processing of a program contributing to T_2.

$$W(k) = E_k(T_1) + E_k(T_2) \tag{15}$$

We first consider T_2. Let y_i be the tagged unit's waiting time in queue on its i^{th} pass (excluding waiting time due to the program possibly operating at arrival). Then we have

$$E_k(T_2) = E\left[\sum_{i=1}^{k} y_i\right] = \sum_{i=1}^{k} E(y_i).$$

Now if m_i is the expected number of programs ahead of the tagged unit when it joins the end of the queue prior to its i^{th} pass, we have

$$E(y_i) = m_i[E(t) + s]$$

where $E(t)$ is given by (4). Then

$$E_k(T_2) = \sum_{i=1}^{k} m_i[E(t) + s]$$

Since δ is the probability that a program, having operated for a quantum, will need more service, evidently

$$m_i = \delta m_{i-1} + \lambda E(y_{i-1}) + \lambda Q$$

where δm_{i-1} is the mean number of programs returning from the preceding pass and $\lambda E(y_{i-1})$ and λQ represent arrivals during processing of the queue and the tagged unit, respectively, on the preceding pass. Thus,

$$\begin{aligned} m_i &= m_{i-1}[\delta + \lambda E(t) + \lambda s] + \lambda Q \\ &= \beta m_{i-1} + \lambda Q \end{aligned}$$

where

$$\begin{aligned} \beta &= \delta + \lambda [E(t) + s] \\ &= \delta + \lambda [(1 - \delta/\mu + s] \end{aligned}$$

We proceed by induction to obtain

$$m_i = \beta^{i-1} m_1 + \lambda Q \frac{1 - \beta^{i-1}}{1 - \beta}$$

Thus

$$E_k(T_2) = [E(t) + s] \sum_{i=1}^{k} \left[\frac{\lambda Q}{1 - \beta} + \beta^{i-1} \left(m_1 - \frac{\lambda Q}{1 - \beta} \right) \right]$$

$$= \frac{E(t) + s}{1 - \beta} \left[\lambda k Q + \left(m_1 - \frac{\lambda Q}{1 - \beta} \right) \left(1 - \beta^k \right) \right] \qquad (16)$$

We now require an expression for m_1. We have assumed that the system is in equilibrium at the arrival of the tagged unit, so $m_1 = E(m)$, the mean queue length in equilibrium. From Saaty [2], p. 183,

$$m_1 = \frac{\lambda^2 E(\tau^2)}{2(1-\rho)} \tag{17}$$

where $\tau = t + js$, $j - 1 < \frac{t}{q} \leq j$

$$E(\tau^2) = \int_{t=0}^{\infty} \tau^2 dB(t)$$

$$= \int_{t=0}^{\infty} (t + js)^2 dB(t)$$

$$\rho = \lambda E(\tau)$$

$$= \lambda \int_{t=0}^{\infty} \tau dB(t)$$

Integrating, we obtain

$$E(\tau^2) = (2/\mu^2)\left[1 + \mu s \frac{1 + \mu s/2}{1 - \delta} + \mu^2 s \delta \frac{Q}{(1-\delta)^2}\right] \tag{18}$$

and

$$\rho = (\lambda/\mu)[1 + \mu s/(1 - \delta)] \tag{19}$$

Now let m_i' be the number of programs <u>behind</u> the tagged unit on its i^{th} pass due to the program in service upon arrival. The tagged unit will have to wait through the processing of these programs on its <u>next</u> pass (since they are behind the tagged unit on the i^{th} pass). Then

$$E_k(T_1) = z_0 + \sum_{i=1}^{k-1} m_i' [E(t) + s]$$

where z_0 is the time required to finish the quantum service in progress at arrival. Note that the sum goes only to $k - 1$.

For z_0 we must consider the probabilities, π_s and π_q, of arriving during a swap and during a quantum, respectively. We have

$$z_0 = \pi_s [\bar{s} + E(t)] + \pi_q \bar{q}$$

where $\bar{s}$ and $\bar{q}$ are the expected time remaining in the swap and quantum, respectively, and $E(t)$ is given by (5). Now $\pi_q = \frac{\lambda}{\mu}$ and

$$\bar{q} = \frac{E(t^2)}{2E(t)}$$

so

$$\pi_q \bar{q} = \frac{\lambda}{\mu} \frac{E(t^2)}{2E(t)} = \frac{\lambda E(t^2)}{2(1-\delta)} \tag{20}$$

Now we observe that π_s is just the "arrival rate" of swaps times the length of a swap: $\pi_s = \lambda_s s$. But λ_s is the arrival rate of programs at the end of the queue, either from outside or after not completing during a quantum of processing.

$$\lambda_s = \lambda + \delta\lambda_s$$
$$= \lambda/(1-\delta)$$

and

$$\pi_s = \frac{\lambda s}{1-\delta} \tag{21}$$

Finally, since the swap time, s, is a constant, the expected time remaining at arrival of the tagged unit is $s/2$. Then

$$z_0 = \frac{\lambda s}{1-\delta} \left[s/2 + \frac{1-\delta}{\mu} \right] + \frac{\lambda E(t^2)}{2(1-\delta)}$$
$$= \frac{\lambda}{2(1-\delta)} \left[s^2 + 2s(1-\delta)/\mu + E(t^2) \right] \tag{22}$$

Returning to m_i' we observe that

$$m_i' = m_{i-1}' [\delta + \lambda E(t) + \lambda s]$$

$$= \beta m_{i-1}'$$

Again proceeding by induction, we obtain

$$m_i' = \beta^{i-1} m_1'$$

Now m_1' is simply the probability, η , that the unit in service at arrival will need more service, plus any arrivals during z_o. Letting σ be the probability of arriving during a "non-terminal" quantum, we have

$$\eta = \pi_s \delta + \sigma$$

σ is the arrival rate of nonterminal quanta times the expected length of a non-terminal quantum. Then

$$\sigma = \lambda_s \delta q$$

$$= \frac{\lambda \delta q}{1 - \delta} \tag{23}$$

and

$$\eta = \frac{\lambda s \delta}{1 - \delta} + \frac{\lambda q \delta}{1 - \delta}$$

$$= \frac{\lambda Q \delta}{1 - \delta} \tag{24}$$

Thus,

$$m_1' = \eta + \lambda z_o$$

$$= \frac{\lambda Q \delta}{1 - \delta} + \frac{\lambda^2}{2(1 - \delta)} [s^2 + 2s(1 - \delta)/\mu + E(t^2)]$$

and

$$E_k(T_1) = z_o + \sum_{i=1}^{k-1} \beta^{i-1} m_1' [E(t) + s]$$

$$= z_o + (\eta + \lambda z_o) \frac{1 - \beta^{k-1}}{1 - \beta} [(1 - \delta)/\mu + s]$$

$$= z_o \left[1 + \rho(1 - \delta) \frac{1 - \beta^{k-1}}{1 - \beta}\right] + \frac{\rho\eta}{\lambda} (1 - \delta) \frac{1 - \beta^{k-1}}{1 - \beta}$$

Substituting for z_o and η we obtain

$$E_k(T_1) = \frac{\lambda}{2(1 - \delta)} [s^2 + 2s(1 - \delta)/\mu + E(t^2)] \left[1 + \rho(1 - \delta) \frac{1 - \beta^{k-1}}{1 - \beta}\right]$$

$$+ \rho\delta Q \frac{1 - \beta^{k-1}}{1 - \beta} \qquad (25)$$

Finally, we substitute (16) and (25) into (15) and, noting that $\beta = \delta + \rho(1 - \delta)$ and $\lambda(E(t) + s)/(1 - \beta) = \rho/(1 - \rho)$, we obtain

$$W_{SQ}(k) = (\lambda/2) \left[s^2 + 2s(1 - \delta/\mu + E(t^2)\right] \frac{1 - \rho\beta^{k-1}}{1 - \beta} + \rho\delta Q \frac{1 - \beta^{k-1}}{1 - \beta}$$

$$+ \frac{\rho}{1 - \rho} \left[kQ + \left(\frac{m_1}{\lambda} - \frac{Q}{1 - \beta}\right) \left(1 - \beta^k\right)\right] \qquad (26)$$

This completes the proof of Theorem 1.

It is interesting to consider the limiting behavior of this model as $q \to 0$ (and therefore s necessarily approaches zero). This is the processor-shared model of Coffman and Kleinrock. We require that q and s approach zero in such a way that $(q+s)/q$ approaches one and kq approaches t, where t is the program's service requirement. From Coffman and Kleinrock [4], page 570, we have

$$\lim_{q \to 0} \beta^k = \lim_{q \to 0} [\rho + (1 - \rho)\delta]^k, \quad \delta = e^{-\mu q}$$

$$= e^{-\lambda t(1-\rho)}$$

Now, considering the first term in the expression for $W_{SQ}(k)$, eq (26), noting that $1 - \beta = (1 - \rho)(1 - \delta)$, and using the approximation $1 - \delta \sim \mu q$ for $0 < q << 1$, it is easy to show that

$$\lim_{q \to 0} \frac{\lambda}{2} [s^2 + 2s(\lambda - \delta)/\mu + E(t^2)] \frac{1 - \rho\beta^{k-1}}{1 - \beta} = 0$$

Again approximating $1 - \delta$ by μq, the following limits can be obtained:

$$\lim_{q \to 0} \frac{Q}{1 - \delta} = \frac{1}{\mu} \quad \text{and} \quad \lim_{q \to 0} m_1 = \frac{\rho^2}{1 - \rho}$$

Hence $$\lim_{q \to 0} W_{SQ}(k) = \frac{\rho}{\mu} \frac{1 - e^{-\mu t(1-\rho)}}{1 - \rho} + \frac{\rho}{1 - \rho} \left\{ t + \left[\frac{1}{\lambda} \frac{\rho^2}{1 - \rho} - \frac{1}{\mu} \frac{1}{1-\rho} \right] \left[1 - e^{-\mu t(1-\rho)} \right] \right\}$$

$$= \frac{\rho t}{1-\rho} \tag{27}$$

As expected, this result is identical with that obtained by Coffman and Kleinrock as the limit of the round-robin model without overhead.

Another interesting limit is

$$\lim_{q \to \infty} W_{SQ}(k) = \frac{\rho}{1 - \rho} \left(\frac{1}{\mu} + s \right) \tag{28}$$

which is the waiting time for a batch-processing system, as one would expect. Noting that, as q approaches ∞, k approaches one, the limit is easy to obtain and the details are left to the reader.

APPENDIX B. Proof of Theorem 2.

An obvious difference between the single- and multiple-quantum models is that in the SQ system a program requiring k quanta makes exactly k passes, while in the MQ system such a program may make n passes where $1 \le n \le k$. From Coffman [1], we have

$p_n(k)$ = probability that a k-quantum program will make n passes

$$= \binom{k-1}{n-1} \gamma^{k-n} (1-\gamma)^{n-1} \tag{29}$$

where γ is given by (7).

We define $w_n(k)$ to be the mean waiting time in queue for a k-quantum program which makes exactly n passes. Then

$$W_{MQ}(k) = \sum_{n=1}^{k} p_n(k)\, w_n(k) \tag{30}$$

We now wish to find an expression for $w_n(k)$. We again break up the waiting time into T_1 and T_2, as in the proof of theorem 1. And

$$w_n(k) = E_{k,n}(T_1) + E_{k,n}(T_2) \tag{31}$$

We again consider T_2 first. We have

$$E_{k,n}(T_2) = \sum_{i=1}^{n} E(y_i)$$

where y_i is again the waiting time on the i^{th} pass due to programs in the queue at arrival of the tagged unit. If m_i is the expected number in the queue on the i^{th} pass (ahead of the tagged unit) and if τ_1 is the mean processing time per program per pass, then

$$E(y_i) = m_i(\tau_1 + s)$$

and

$$E_{k,n}(T_2) = \sum_{i=1}^{n} m_i(\tau_1 + s) \tag{32}$$

Now define ζ to be the probability that a program, having just finished a pass, will require more service. Assume that the tagged unit received $k_{n,i-1}$ quanta on the $(i-1)^{st}$ pass — implying that there were $k_{n,i-1} - 1$ arrivals during that processing. Finally, $\lambda m_{i-1}(\tau_1 + s)$ programs will have arrived during processing of programs ahead of the tagged unit on the $(i-1)^{st}$ pass. Then we have

$$m_i = \zeta m_{i-1} + \lambda m_{i-1}(\tau_1 + s) + E[k_{n,i-1} - 1]$$

Following Coffman, we take $k_{n,i} = k/n$ for all i. Then, letting $\alpha = \zeta + \lambda(\tau_1 + s)$

$$m_i = \alpha m_{i-1} + k/n - 1$$

Proceeding by induction

$$m_i = \alpha^{i-1} m_1 + (k/n - 1)\frac{1 - \alpha^{i-1}}{1 - \alpha} \tag{33}$$

Again we assume that the system is in equilibrium at arrival of the tagged unit, so that

$$m_1 = E(m) = \frac{\lambda^2 E(\tau^2)}{2(1 - \rho)}$$

where here $\tau = t + js$

$$E(\tau^2) = \int_{t=0}^{\infty} \sum_{j=1}^{i} (t + j\,s)^2 p_j(i)\,dB(t),\ i - 1 < \frac{t}{q} \leq i$$

$$= (2/\mu^2)\left[1 + \mu s \frac{1 - \gamma\delta}{1 - \delta} + \mu^2 s \frac{\delta q(1 - \gamma) + s(1 - \gamma\delta)(1 - 2\gamma\delta + \delta)/2}{(1 - \delta)^2}\right] \tag{34}$$

where δ and γ are given by (6) and (7), respectively. Also,

$$\rho = \lambda E(\tau) = (\lambda/\mu) \left[1 + \mu s \frac{1 - \gamma\delta}{1 - \delta}\right]$$

Now, for ζ, Coffman gives $\zeta = \delta(1 - \gamma)/(1 - \gamma\delta)$. For τ_1 we observe that the probability that a program will take j quanta on a single pass, without completing its service requirement, is given by $\gamma^{j-1}(1 - \gamma)\delta^j$. The probability of taking j quanta, completing on the j^{th} is given by $\gamma^{j-1}\delta^{j-1}(1 - \delta)$. Thus

$$\tau_1 = \sum_{j=1}^{\infty} \left\{\gamma^{j-1}\delta^j(1 - \gamma)jq + \gamma^{j-1}\delta^{j-1}(1 - \delta)\left[(j - 1)q + E_1(t)\right]\right\}$$

where $E_1(t)$ is the mean time taken by a program to which q seconds have been allocated, assuming the program completes during the q seconds ($t < q$).

$$\tau_1 = \delta q(1 - \gamma) \sum_{j=1}^{\infty} j(\gamma\delta)^{j-1} + q(1 - \delta) \sum_{j=1}^{\infty} (j - 1)(\gamma\delta)^{j-1}$$

$$+ (1 - \delta)E_1(t) \sum_{j=1}^{\infty} (\gamma\delta)^{j-1}$$

$$= \frac{\delta q + (1 - \delta)E_1(t)}{1 - \gamma\delta}$$

$$E_1(t) = \frac{\int_0^{q^-} t\,dF(t)}{\int_0^{q^-} dF(t)}$$

where the integrals go to q^- since it is assumed that $t < q$

$$= \frac{(1 - \delta)/\mu - \delta q}{1 - \delta}$$

Therefore,

$$\tau_1 = \frac{1}{\mu} \frac{1 - \delta}{1 - \gamma\delta}$$

$$= \frac{1}{\mu}(1 - \zeta) \qquad (36)$$

Substituting (33) into (32), we obtain

$$E_{n,k}(T_2) = \sum_{i=1}^{n} \left[\alpha^{i-1} m_1 + (k/n - 1) \frac{1 - \alpha^{i-1}}{1 - \alpha} \right] (\tau_1 + s)$$

$$= \frac{\tau_1 + s}{1 - \alpha} \left[k - n + \left(m_1 - \frac{k/n - 1}{1 - \alpha} \right) (1 - \alpha^n) \right] \qquad (37)$$

We now turn our attention to T_1, the waiting time due to a program in service at arrival of the tagged unit. Consider first the expected time, z_o, to completion of the service in progress. There are three possibilities. With probability π_s the tagged unit may arrive during a swap. With probability π_q it may arrive during a quantum. With probability π_{cq} this "arrival" quantum may be a complete one, i.e. one in which the program in service does not finish its processing requirement. In the latter case, since we hypothesize the arrival of the tagged unit, the program in service will get an additional quantum, and, using the memoryless property once again, will operate for τ_1 seconds after the completion of the current quantum. Then

$$z_0 = \pi_s [\bar{s} + \tau_1] + \pi_q \bar{q} + \pi_{cq} \tau_1$$

where $\bar{s}$ and $\bar{q}$ are the expected time remaining in the swap and quantum, respectively.

From (20) we have

$$\pi_q \bar{q} = \frac{\lambda E(t^2)}{2(1-\delta)}$$

and

$$\pi_{cq} = \sigma = \frac{\lambda \delta q}{1-\delta} \quad \text{from (23).}$$

Since the swap time is constant $\bar{s} = \frac{s}{2}$. As in the single-quantum case $\pi_s = \lambda_s s$. But here we have

$$\lambda_s = \lambda + \zeta \lambda_s$$
$$= \lambda/(1 - \zeta)$$

Then

$$z_0 = \frac{\lambda s}{1 - \zeta} \left(\frac{s}{2} + \tau_1\right) + \frac{\lambda E(t^2)}{2(1-\delta)} + \frac{\lambda \delta q}{1-\delta} \tau_1 \qquad (38)$$

Now if η is the probability that the program in service at arrival will return for more service after its current pass, then the number of programs behind the tagged unit on its first pass (and therefore ahead of it on its second pass) due to this program is

$$m_1' = \eta + \lambda z_o$$

In considering η, note that the probability of arriving during either a swap or a non-terminal quantum is just $\pi_s + \sigma$. In either case, due to the memoryless property, the probability that the program will not complete is ζ. Then

$$\eta = \zeta(\pi_s + \sigma)$$

Since these programs are all behind the tagged unit, they will receive at most n - 1 passes, so

$$E_{kn,}(T_1) = z_o + \sum_{i=1}^{n-1} m_i' (\tau_1 + s)$$

But

$$m_i' = \alpha m_{i-1}' = \alpha^{i-1} m_1'$$

Thus,

$$E_{k,n}(T_1) = z_o + m_1'(\tau_1 + s) \frac{1 - \alpha^{n-1}}{1 - \alpha}$$

and, noting that $\tau_1 + s = \rho(1 - \zeta)$

$$w_n(k) = E_{k,n}(T_1) + E_{k,n}(T_2)$$

$$= z_o + m_1'\rho(1 - \zeta) \frac{1 - \alpha^{n-1}}{1 - \alpha} + \frac{\rho(1-\zeta)}{1 - \alpha} \left[k - n + \left(m_1 - \frac{k/n - 1}{1 - \alpha} \right) (1 - \alpha^n) \right] \quad (39)$$

Multiplying eq. (39) by $p_n(k)$, eq (29), substituting into eq (30) and carrying out the summation gives

$$W_{MQ}(k) = z_0 + \frac{1}{\lambda} \frac{\rho(1 - \zeta)}{1 - \alpha} \left\{ m_1' + m_1 + \frac{1}{1-\alpha} - [m_1' + \alpha (m_1 + \frac{1}{1 - \alpha})][\gamma + \alpha(1 - \gamma)]^{k-1} + (k - 1)\gamma - \frac{1 - [\gamma+\alpha(1-\gamma)]^k}{(1-\alpha)(1-\gamma)} \right\} . \quad (40)$$

which establishes Theorem 2.

Again it is interesting to consider the processor-shared limit of the MQ model. We expect to obtain the same result as for the SQ model since the probability of an arrival during a quantum approaches zero, as does the additional

processing time allocated as the result of such an arrival. Again we require that q and s approach zero in such a way that $(q + s)/q$ approaches one and kq approaches t, the service time requirement. We observe that

$$\lim_{q \to 0} [\gamma + \alpha(1-\gamma)]^k = \lim_{q \to 0} \alpha^k$$

$$= \lim_{q \to 0} [\rho + (1 - \rho)\zeta]^k$$

Since $$\lim_{q \to 0} \zeta = \lim_{q \to 0} \frac{\delta(1 - \gamma)}{1 - \gamma\delta}$$

$$= \lim_{q \to 0} \delta$$

we can again use the result of Coffman and Kleinrock

$$\lim_{q \to 0} \alpha^k = \lim_{q \to 0} \beta^k = e^{-\mu t(1-\rho)}$$

Furthermore, noting that $1 - \alpha = (1 - \rho)(1 - \zeta)$ and using the approximation $1 - \zeta \simeq 1 - \delta \simeq \mu q$, $0 < q << 1$, it is easy to show that

$$\lim_{q \to 0} z_0 = 0, \quad \lim_{q \to 0} m_1' = \rho \quad \text{and} \quad \lim_{q \to 0} m_1 = \frac{\rho^2}{1-\rho}$$

Also $$\lim_{q \to 0} (k - 1)\gamma = \lim_{q \to 0} (k - 1)(1 - e^{-\lambda q})$$

$$= \lim_{q \to 0} (k - 1)\lambda q$$

$$= \lambda t$$

Then

$$\lim_{q\to 0} W_{MQ}(k) = \frac{1}{\lambda}\frac{\rho(1-\zeta)}{1-\alpha}\left\{\rho + \frac{\rho^2}{1-\rho} + \frac{1}{1-\alpha} - [\rho + \alpha(\frac{\rho^2}{1-\rho} + \frac{1}{1-\alpha})]\alpha^{k-1} + \lambda t - \frac{1-\alpha^k}{(1-\alpha)(1-\gamma)}\right\}$$

$$= \frac{1}{\lambda}\frac{\rho}{1-\rho}\left\{\rho[1 - e^{-\mu t(1-\rho)}] + \frac{\rho^2}{1-\rho}[1 - e^{-\mu t(1-\rho)}] + \frac{1 - e^{-\mu t(1-\rho)}}{1-\alpha} - \frac{1 - e^{-\mu t(1-\rho)}}{(1-\alpha)(1-\gamma)} + \lambda t\right\}$$

$$= \frac{1}{\lambda}\frac{\rho}{1-\rho}\left\{\lambda t + [\frac{\rho}{1-\rho} - \frac{\gamma}{(1-\rho)(1-\zeta)(1-\gamma)}][1 - e^{-\mu t(1-\rho)}]\right\}$$

$$= \frac{\rho t}{1-\rho} \qquad (41)$$

This result is identical with that for the SQ model.

Finally, it is easy to show that

$$\lim_{q\to\infty} W_{MQ}(k) = \frac{\rho}{1-\rho}(\frac{1}{\mu} + s) \qquad (42)$$

As we would expect, this is the result for a batch-processing system.

ACKNOWLEDGMENTS

The authors would like to thank the Space Astronomy Laboratory, University of Wisconsin, for providing time on its IBM 1130 for the simulations of the time-sharing algorithms. The Laboratory is supported by the National Areonautics and Space Administration under Contract NAS 5-1348. We are also indebted to Stephen M. Stigler, Department of Statistics, University of Wisconsin, for his criticism and comments. Finally we would like to thank the referees for several valuable suggestions and clarifications.

REFERENCES

1. Coffman, E. G., Jr. Analysis of two time-sharing algorithms designed for limited swapping. JACM 15, 3 (July 1968), 341-353.

2. Saaty, T. L. Elements of Queueing Theory. McGraw-Hill, New York, 1961.

3. Rasch, P. J. A queueing-theory study of round-robin scheduling of time-sharing computer systems. JACM 17, 1 (January 1970), 131-145.

4. Coffman, E. G., Jr., and Kleinrock, L. Feedback queueing models for time-sharing systems. JACM 15, 4 (October 1968), 549-576.

5. Adiri, Igal. Computer time-sharing queues with priorities. JACM 16, 4 (October 1969), 631-645.

6. Schrage, L. E. Some queueing models for a time-sharing facility. Ph.D. thesis, Cornell University, Ithaca, N. Y., February, 1961.

7. Apostol, T. M. Mathematical Analysis. Addison-Wesley, Reading, Mass., 1957.