STAN-CS-72-317

SU-SEL-72-048

An Analysis of Drum Storage Units

by

Samuel H. Fuller and Forest Baskett

August 1972

Technical Report No. 26

This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract N-00014-67-A-0112-0044 and by Mr. Fuller's fellowships from the National Science Foundation and the Fannie and John Hertz Foundation. Computer time was provided by the Atomic Energy Commission under Contract AT-(04-3)515.

DIGITAL SYSTEMS LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY · STANFORD, CALIFORNIA

STAN-CS-72-317 SEL-72-048

AN ANALYSIS OF DRUM STORAGE UNITS

by

Samuel H. Fuller and Forest Baskett

August 1972

Technical Report No. 26

DIGITAL SYSTEMS LABORATORY

Dept. of Electrical Engineering Dept. of Computer Science

Stanford University

Stanford, California

This work was supported by the Joint Services Electronics Program (U. S. Army, U. S. Navy and U. S. Air Force) under contract N-00014-67-A-0112-0044 and by Mr. Fuller's fellowships from the National Science Foundation and the Fannie and John Hertz Foundation. Computer time was provided by the Atomic Energy Commission under contract AT-(04-3)515.

AN ANALYSIS OF DRUM STORAGE UNITS

ABSTRACT

This article discusses the modeling and analysis of drum-like storage units. Two common forms of drum organizations and two common scheduling disciplines are considered: the file drum and the paging drum; first-in-first-out (FIFO) scheduling and shortest-latency-time-first (SLTF) scheduling.

The modeling of the I/O requests to the drum is an important aspect of this analysis. Measurements are presented to indicate that it is realistic to model requests for records, or blocks of information to a file drum, as requests that have starting addresses uniformly distributed around the circumference of the drum and transfer times that are exponentially distributed with a mean of 1/2 to 1/3 of a drum revolution. The arrival of I/O requests is first assumed to be a Poisson process and then generalized to the case of a computer system with a finite degree of multiprogramming.

An exact analysis of all the models except the SLTF file drum is presented; in this case the complexity of the drum organization has forced us to accept an approximate analysis. In order to examine the error introduced into the analysis of the SLTF file drum by our approximations, the results of the analytic models are compared to a simulation model of the SLTF file drum.

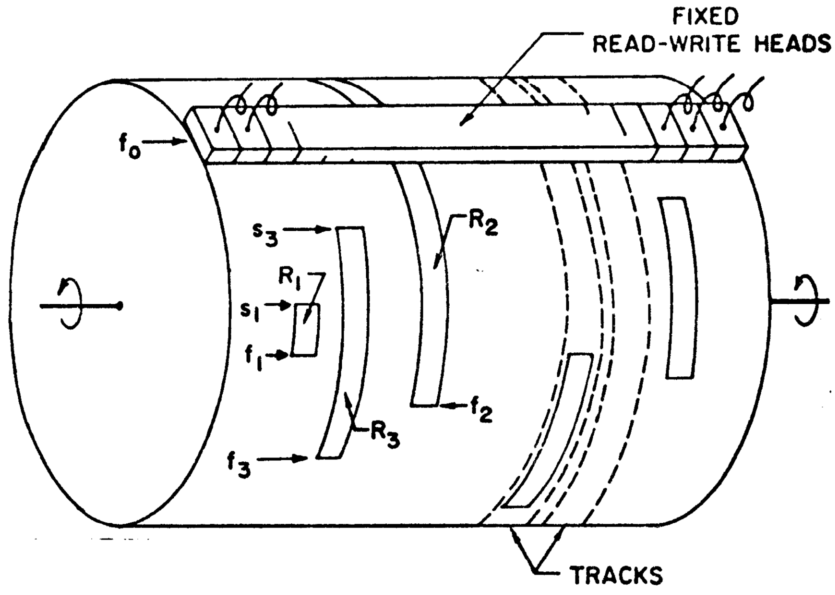
TABLE OF CONTENTS

LIST OF FIGURES

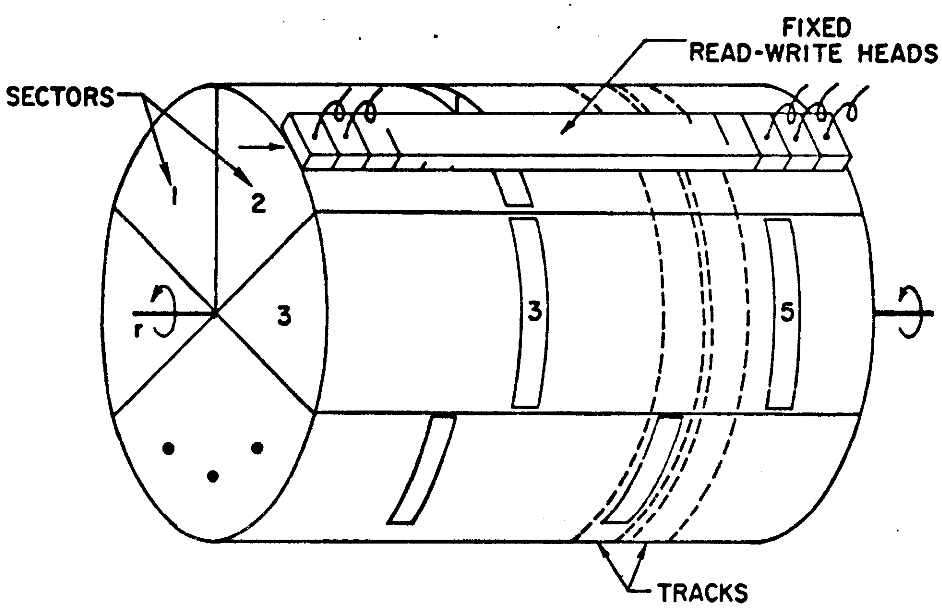
1. Introduction

Gains in the performance of computer systems are not solely related to the power of the central processor. In particular, the I/O structure of computer systems has been an increasing cause of concern because of its relatively poor performance and high cost with respect to the central processor. This article focuses attention on one major form of I/O processor, the drum-like storage unit. Examples of drum-like stores include fixed-head disks, acoustic delay lines, and large semi-conductor shift registers, as well as storage units that actually contain physically rotating drums as shown in Fig. 1.1.

The purpose of this paper is to investigate the consequences of using a device such as a drum that must operate under the constraints of rotational delays. In the models of drum storage units that follow, every attempt has been made to keep the models as simple and free from obscuring details as possible while carefully describing those aspects of the drum relating to rotational delays. No attempt is made here to relate the capacity of the drum to the performance of the computer system. We are focusing on the response and service times of the drum and their effect on the utilization of the computer system. Furthermore, no attempt is made to differentiate read requests from write requests on the drum; when a request is made to 'process' an I/O request, only a starting address and length will be given. This assumption accurately models most drums currently in operation and should provide a base from which to analyze drum scheduling algorithms that exploit the opportunity to write a record in the most convenient empty space on the drum rather than into a fixed location.



(a) A storage unit organized as a file drum.



(b) A storage unit organized as a paging drum.

Figure 1.1. Two common drum organizations: the file drum and the paging drum.

This article deals with two drum organizations that encompass the majority of drum-like stores that are in use or that have been proposed: the file drum and the paging drum. A drawing of a file drum is shown in Fig. 1.1(a). The drum rotates at a constant angular velocity, with period τ, and the read-write heads are fixed. Blocks of information, often called records, or files, are read or written onto the surface of the drum as the appropriate portion of the drum passes under the read-write heads. Once a decision has been made to process a particular record, the time spent waiting for the record to come under the read-write heads is called rotational latency or just latency. With a drum storage unit organized as a file drum we do not constrain the records to be of any particular length nor do we impose restrictions on the starting position of records. Let the random variable S_i denote the starting position of record i and the random variable R_i denote the length of record i. For convenience, let our unit of length be the circumference of the drum and hence S_i and R_i are in the half open interval $[0,1)$.

A drum storage unit organized as a paging drum is shown in Fig. 1.1(b); the drum rotates at a constant angular velocity, as in the case of a file drum, with period τ and the records are recorded on the drum's surface in tracks. Unlike a file drum, however, a paging drum partitions all of its tracks into equal sized intervals called sectors. The records are required to start on a sector boundary and the record lengths are commonly constrained to be one sector long. As we will see in the course of our analysis, this organization allows improvements in performance not possible with a drum organized as a file drum.

In the analysis of both of the drum organizations just described two scheduling algorithms are considered: FIFO and SLTF. First-in-first-out (FIFO), scheduling is a simple scheduling policy that merely services the I/O requests in the order in which they arrive at the drum. FIFO scheduling is sometimes called first-come-first-serve (FCFS) scheduling, for obvious reasons. Shortest-latency-time-first (SLTF), is a scheduling discipline well suited for storage units with rotational latency. At all times, an SLTF policy will schedule the record that comes under the read-write heads first as the next record to be transmitted. For example, in Fig. 1.1(a), assuming the drum is not transmitting record 2, an SLTF policy will schedule record 5 as the next record to be processed. An SLTF algorithm never preempts the processing of a record once transmission of the record has begun. SLTF scheduling is often called shortest-access-time-first (SATF) scheduling. The word 'access', however, is an ambiguous term with respect to storage units (it is used both to denote the time until data transfer begins as well as the time until data transfer is complete) and to avoid possible confusion we will use the SLTF mnemonic. While SLTF is not the optimal policy to use in all situations, some remarks can be made about its near-optimality [Fuller, 1972A], and it enjoys the important practical feature that it is straightforward to implement in the hardware of the drum controller [IBM, 1971; Burroughs, 1970]. Another drum scheduling algorithm, although not considered any further in the article, that may be of practical value is shortest-processing-time-first (SPTF), i.e. service the record whose sum of latency and transmission time is the smallest. Variants of SPTF scheduling include policies which do, or do not, allow preemptions of a request once transmission has begun. Other scheduling algorithms have been developed that are superior to SLTF under assumptions more restrictive than those considered here [Fuller, 1971]

Figure 1.2 applies to both paging and file drums. It defines the basic time intervals and events associated with servicing an I/O request on a drum. The four events involved in this processing of an I/O request are: (1) arrival of the I/O request at the drum, (2) decision by the scheduler that the I/O request is the next to be serviced, (3) the start of the record comes under the read-write heads and transmission begins, and finally (4) transmission of the record is completed. If the I/O request finds the drum idle upon arrival, events (1) and (2) occur at the same instant. The interval of time between events (2) and (3) is the rotational latency of the I/O request.

A drum using a SLTF scheduling algorithm may push a request back onto the queue between events (2) and (3) if a new request arrives that can begin transmission before the currently selected event. Neither SLTF nor FIFO scheduling algorithms allow a request to be preempted after event (3).

The <u>waiting time</u>, or response time, of an I/O request will be denoted by the random variable W and includes the time from event (1), the arrival of the request, until (4), the completion of the request. This definition of wait time was chosen, rather than from events (1) to (2) or (1) to (3), since the interval from events (1) to (4) directly measures the time a process must wait before receiving a response to an I/O request.

The <u>utilization of the drum</u>, call it u_d, is the long term fraction of time the drum is transmitting information. Note that only the fraction of time a drum actually transmits information is included, i.e.

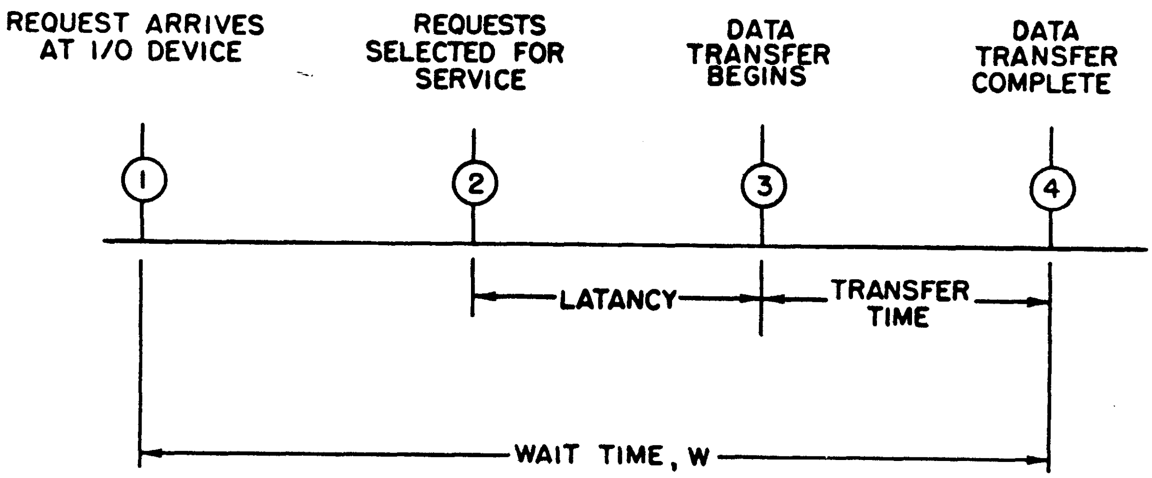


Figure 1.2. Intervals and events associated with servicing an I/O request.

between events (3) and (4), and intervals of rotational latency is not included in our definition of u_d.

A wide variety of drum performance measures have been used. For instance, some popular measures are: expected waiting time of an I/O request as a function of queue length, or arrival rate, or traffic intensity; cpu and drum idle times as a function of arrival rate or traffic intensity; throughput as a function of drum speed, etc. Obviously, some measures of performance convey more information than others and many measures convey the same information but with varying degrees of clarity.

One measure that is used widely in the evaluation of computer systems is the utilization of the central processor, call it u_c. u_c is a measure that is easily monitored in practice and has the advantage that it bears a strong correlation with our intuitive concept of "throughput".

In those cases where the utilization of the drum and central processor cannot be used, in particular when the drum is analyzed independently of the other processors in the computer system, the expected waiting time of the I/O requests appears to be an appropriate measure of performance. The expected waiting time has a more direct interpretation than the two other common measures of processor performance, the expected queue size or the length of the busy periods.

In this article we will present an analysis of each of the four major drum organizations: the FIFO file drum, the FIFO paging drum, the SLTF file drum, and the SLTF paging drum. All the organizations except the SLTF file drum can be precisely modeled if we assume I/O requests to the drum form a Poisson arrival process, and in these cases we will present expressions for the expected waiting time of I/O requests at the storage units. In the case of the SLTF file drum several models are presented and compared with a simulation model of an SLTF file drum to evaluate their utility and the validity of their approximations. Finally, we remove the assumption of Poisson arrivals and explore the performance of a model of a computer system consisting of a central processor and an SLTF file drum.

2. Analysis of the FIFO Drum Scheduling Discipline

This section discusses the first-in-first-out (FIFO) scheduling discipline applied to drum storage units. Expressions are developed for the expected waiting time for I/O requests to file drums, paging drums, and a variation of the paging drum, the sectored file drum.

The FIFO file drum. First, let us consider the case of a storage unit organized as a file drum with FIFO scheduling. This is the simplest drum organization analyzed in this article, but it is worthwhile to consider explicitly each of the assumptions required to construct a tractable model even in this simple case.

The simplest arrival process to handle mathematically, and the one we will initially use to model the arrival of I/O requests to the drum is the Poisson process. This arrival process has been widely studied [cf. Cox and Smith, 1961; Feller, 1968], and the fundamental properties of a Poisson arrival process are: any two time intervals of equal length experience an arrival(s) with equal probability, and the number of arrivals during disjoint intervals are independent random events. The Poisson assumption implies that the probability of k arrivals in an arbitrary interval of time, t, is

$$\Pr\{k \text{ arrivals in interval } t\} = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

and the interarrival intervals have the exponential density function

$$\Pr\{\text{interarrival time} = t\} = \lambda e^{-\lambda t}.$$

In the more specific terms of I/O requests arriving at a drum, the Poisson assumption requires that the degree of multiprogramming* is sufficiently large that the central processors generating requests to the drum are never idle and that the times between the generation of I/O requests can be modeled as independent, exponentially distributed random variables. In general, central processor utilization is not near unity; however, several current computer systems have been reported to enjoy this property [Sherman, Baskett, and Browne, 1971; Kimbleton and Moore, 1971]. The Poisson arrival assumption is removed in Sec. 5, and the more general case of drum performance in a computer system with arbitrary central processor utilization is studied. In this section, however, we will pursue the analysis of drum storage units with Poisson arrivals because it is our hope that the relatively straightforward results obtained will provide insight, at the most fundamental level, as to how a processor with rotational latency performs. As we progress through discussions of the various drum organizations, it will become evident that a device with rotational latency possesses several subtle, but significant, properties not encountered in more conventional processors. In the analysis that follows it is necessary to describe the starting addresses and the record lengths more completely than to

---

* The number of jobs, or processes, actively using the main memory resources of the computer system.

merely note they are 'random variables' as was done in the introduction. Figure 2.1 is of considerable help in this respect; it is a histogram of the observed service time, i.e. latency plus transmission time, for I/O requests to the drum storage units on the IBM 360/91 computer system at the Stanford Linear Accelerator Center where a FIFO discipline is used to schedule the I/O requests on the drums [Fuller, Price, and Wilhelm, 1971]. The shape of this histogram suggests the following model for I/O requests: let the starting address of a record, S_i, be a random variable with the uniform density function

$$f_S(t) = \frac{1}{\tau} \qquad 0 \le t < \tau \tag{2.1}$$

$$= 0 \qquad \text{elsewhere;}$$

and let the record lengths, with mean $\overline{R}$, have the exponential density function

$$f_R(x) = (1/\overline{R})e^{-x/\overline{R}}, \quad x \ge 0 , \tag{2.2}$$

and if we let $\mu = 1/(\tau\overline{R})$, then μ is the reciprocal of the mean transmission time and

$$f_T(t) = \mu e^{-\mu t} , \quad t \ge 0 .$$

If we assume the S_i's are independent, then it follows immediately that the rotational latency associated with a record, denoted L_i, has the same uniform distribution as the starting addresses, i.e. Eq. (2.1). The service time is just the sum of the random variables L_i and τR_i, and the density function of the service time is the convolution of $f_S(t)$ and $f_T(t)$:

$$g(t) = \int_{-\infty}^{\infty} f_T(t-\omega)\ f_R(\omega)\ d\omega$$

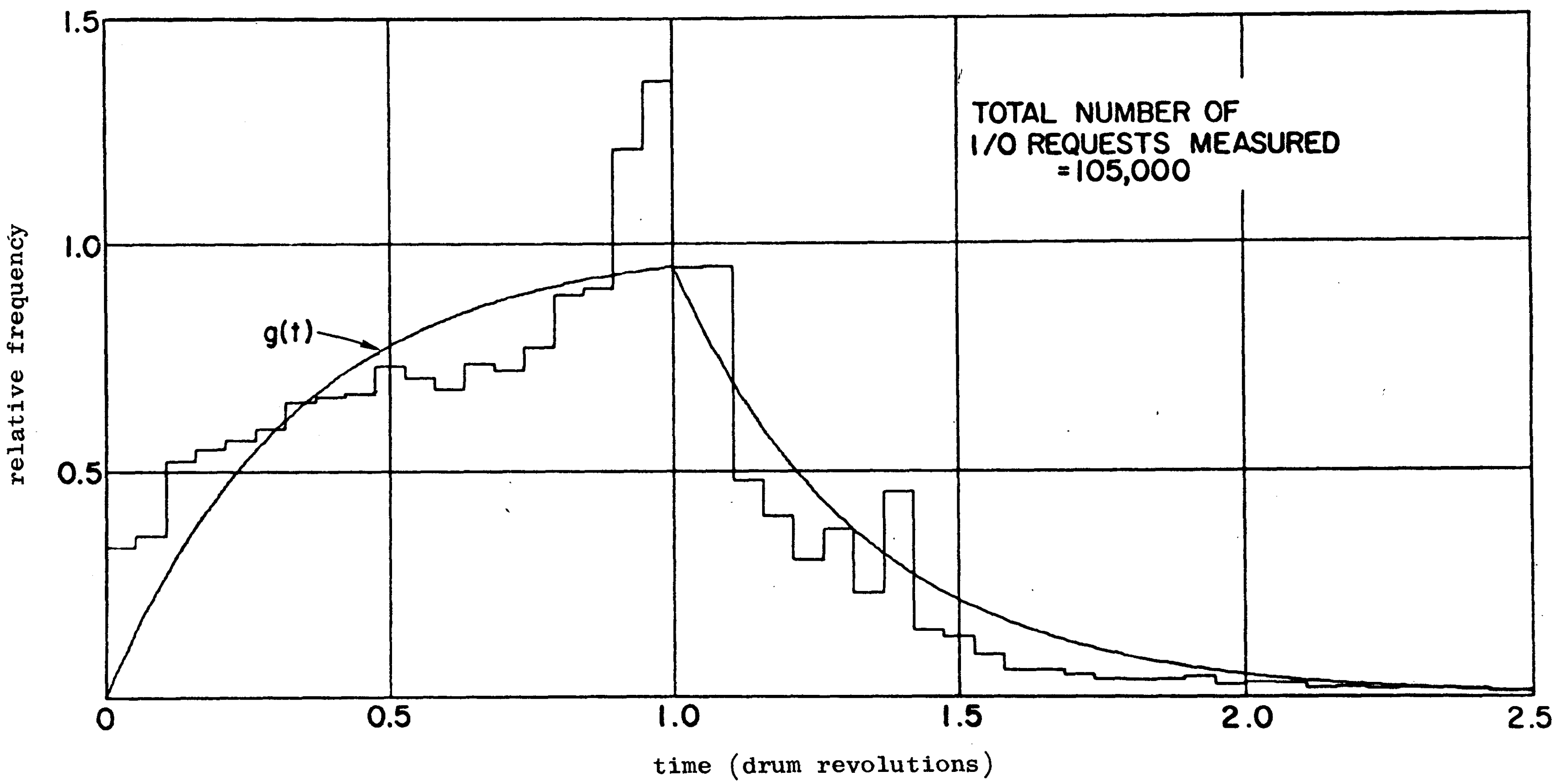


Figure 2.1. Histogram of latency plus transfer times for a FIFO file drum.

$$= 1 - e^{-\mu t} \qquad , 0 \le t < \tau ;$$
$$= (1 - e^{-\mu\tau})\, e^{-\mu[t-\tau]} \qquad , t \ge \tau .$$

The relatively good fit of g(t) to the histogram when $\mu = 3/\tau$ indicates the appropriateness of this model of I/O requests and we will use it in our analysis of file drums.

The above assumptions concerning the arrival process and the attributes of I/O requests completely specifies our model of a file drum with a FIFO scheduling discipline. The model is in the form of the classic M/G/1 model, i.e. Poisson arrivals (M), general service time distribution (G), and one server (1), and we can use the Pollaczek-Khinchine formula [cf. Cox and Smith, 1961; Saaty, 1961] to give us the expected waiting time of an I/O request at the drum:

$$\overline{W} = (\tfrac{1}{2} + \overline{R})\left[1 + \frac{\xi(1+C^2)}{2(1-\xi)}\right]\tau \tag{2.3}$$

where

$$\xi = \lambda(\tfrac{1}{2} + \overline{R})\tau$$

$$C = \frac{\frac{\tau^2}{12} + \sigma^2}{(\frac{1}{2} + \overline{R})^2\tau^2} \qquad \text{(coefficient of variation of service time)}$$

For exponentially distributed record lengths, $\sigma = 1/\overline{R}$ in the above equations and we can achieve some simplifications. In this case, however, it is not necessary to make the exponential assumption; we need only assume the record transfer times are independent, identically distributed random variables with mean $\overline{R}\tau$ and variance σ^2.

<u>The FIFO paging drum</u>. The extension of the above result to a FIFO paging drum is not as straightforward as might be expected. The problem lies in accurately describing the drum when it is idle. In a FIFO file

drum, the idle state is truly a Markov (memoryless) state. That is, when the drum is idle, the distance from the read-write heads to the starting address of the arriving I/O request, D_i, can be accurately modeled as a random variable with a uniform distribution, Eq. (2.1). The duration of the idle period, or any other fact about the drum's history, has no effect on the distribution of D_i.

In contrast, the idle state of a FIFO paging drum does not enjoy a similar Markovian property. The reason is readily evident: starting addresses of I/O requests always occur at sector boundaries and when a paging drum becomes idle it does so at sector boundaries. Consequently, the duration of the drum idle period has a significant effect on the latency required to service the I/O request arriving at the idle drum.

With the above comment serving as a cautionary note, let us proceed with the analysis of a FIFO paging drum. As with the file drum, assume the arrival of I/O requests form a Poisson process with parameter λ. Moreover, suppose there are k sectors on the drum and I/O requests demand service from any one of the k sectors with equal probability. In most paging drums records are required to be one sector in length and we will assume this to be true in our analysis, and consequently $\overline{R} = 1/k$.

With the above assumptions, the FIFO paging drum is an example of an abstract model developed by C. E. Skinner [1967]. The significance of Skinner's analysis is best appreciated by considering the approximations others have made in their analysis of a FIFO paging drum [Denning, 1967].

Skinner's model is depicted in Fig. 2.2. The processor services a request in time A, where A is a random variable with arbitrary distribution $F_A(t)$. After servicing a request, the processor becomes

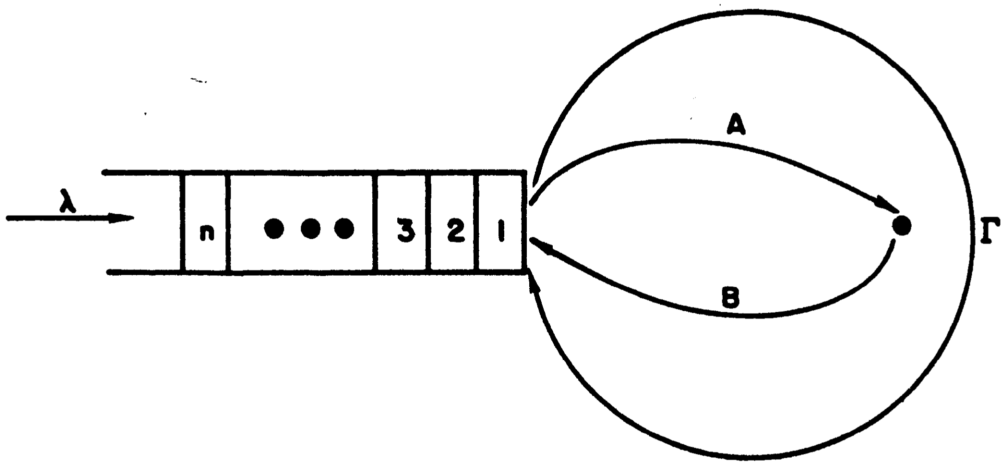


Figure 2.2. Skinner's model of a processor with latency.

latent for time B, where B is a random variable, possibly dependent on A, with distribution $F_B(t)$. Let Z denote the sum of A and B, and $F_Z(t)$ the distribution function of Z. After a service-latency cycle, the processor inspects the queue to see if there are any outstanding requests. If the queue is not empty, a new service period begins. If the queue is empty, the processor begins a new latency period of time Γ, where Γ is a random variable, independent of A and B, with distribution $F_\Gamma(t)$.

The major departure of Skinner's model from the classic M/G/1 model, i.e. the models described by the Pollaczek-Khinchine formula, is that it no longer assumes the idle state has the memoryless property. The only points in time that enjoy the Markovian property in the Skinner model are those instants at which the processor inspects the queue. The Laplace-Stieltjes transform of the waiting time for Skinner's model is [Skinner, 1967]:

$$\underset{\sim}{W}(s) = \frac{(1 - \lambda\overline{Z})(\underset{\sim}{\Gamma}(s) - 1)\underset{\sim}{A}(s)}{\lambda\,\overline{\Gamma}\,[1 - \frac{s}{\lambda} - \underset{\sim}{Z}(s)]}$$

Consequently, the expected waiting time of a request at the server is:

$$\overline{W} = \lim_{s \to 0} \{-\underset{\sim}{W}'(s)\} = \frac{\overline{\Gamma^2}}{2\overline{\Gamma}} + \frac{\lambda\overline{Z^2}}{2(1-\lambda\overline{Z})} + \overline{A} \qquad (2.4)$$

where l'Hospital's rule must be used twice to find the limit as $s \to 0$.

The interpretation of Skinner's model as a FIFO paging drum is straightforward. From the assumptions that the starting addresses are independent random variables that are equally likely to have the value of any of the k sector boundaries and all records are one sector long it follows that

$$F_A(t) = \frac{(i-1)}{k}, \quad \frac{(i-1)\tau}{k} < t \leq \frac{i\tau}{k} \text{ and } i = 1,2,\ldots,k.$$

$$F_A(t) = 1, \quad t > \frac{(k+1)\tau}{k}$$

The first two moments of A are

$$\overline{A} = \frac{k+1}{2k}\tau$$

$$\overline{A^2} = \frac{(k+1)(k+\frac{1}{2})}{3k^2}\tau^2 .$$

Since all the records are one sector long, the paging drum is in a position to begin servicing a new request immediately after finishing the previous one; this is reflected in Skinner's model by requiring B to always be 0. This leads to the simplifying result that $F_Z(t) = F_A(t)$.

If the queue is empty after the paging drum finishes servicing a request, it must remain latent for the time to traverse one sector before examining the queue again. Hence, for a paging drum Γ is not a random variable at all, it is always τ/k, and

$$\overline{\Gamma} = \frac{\tau}{k}$$

$$\overline{\Gamma^2} = (\frac{\tau}{k})^2$$

Therefore, the mean waiting time for I/O requests at a FIFO paging drum is

$$\overline{W} = \{(\frac{1}{2} + \frac{1}{k}) + \frac{\xi(1 + \frac{1}{2k})}{3(1-\xi)} \tag{2.5}$$

where

$$\xi = \lambda(\frac{1}{2} + \frac{1}{k})\tau = \lambda(\frac{1}{2} + \overline{R})\tau .$$

<u>The FIFO sectored file drum.</u> Suppose the record lengths are exponentially distributed rather than the constant size required by a paging drum, but assume records are still constrained to begin at sector

boundaries. This type of drum will be called a sectored file drum, and the IBM 2305* is a good example of this type of drum organization. The IBM 2305 has 128 sectors, its track capacity is roughly 3/4ths of the track capacity of an IBM 2301, and from the histogram in Fig. 2.1 it is clear most records will be considerably longer than a sector. Skinner's model is adequate to describe the behavior of a sectored file drum using the FIFO scheduling discipline. Let A be the sum of the latency plus the transmission time, as in the case of a paging drum, and we see immediately that

$$\overline{A} = \left(\overline{R} + \frac{1}{2} - \frac{1}{2k}\right)\tau$$

Let B be the time required to get from the end of the record to the next sector boundary, and let R_p denote the sum of the record length and B, i.e. R_p is the record length rounded up to the nearest integer number of sectors. The probability mass function of R_p is

$$\Pr\{R_p = \frac{i\tau}{k}\} = (e^{\mu\tau/k} - 1)e^{-i\mu\tau/k}, \qquad i = 1,2,\ldots$$

and

$$\overline{R_p} = \frac{\tau}{k(1-e^{-\mu\tau/k})}$$

$$\overline{R_p^2} = \frac{2\tau^2 e^{-\mu\tau/k}}{k^2(1-e^{-\mu\tau/k})^2}$$

In order to find $\overline{Z}$ and $\overline{Z^2}$, it is more helpful to treat Z as the sum of R_p and the latency interval rather than the sum of A and B. Hence

---

* As will be discussed later, a more efficient way to schedule a position-sensing, sectored drum like the IBM 2305 is with a SLTF scheduling discipline.

$$\overline{Z} = [\frac{k-1}{2k} + \frac{1}{k(1-e^{-\mu T/k})}]T$$

$$\overline{Z^2} = \frac{T^2(k-1)(1-e^{-\mu T/k})^2[(k-\frac{1}{2})(1-e^{-\mu T/k})+3] + 3e^{-\mu T/k}[1-e^{-\mu T/k}]}{3k^2(1-e^{-\mu T/k})^3}$$

Now using Eq. (2.4) we see that the expected waiting time for I/O requests at a sectored file drum with FIFO scheduling is

$$\overline{W} = (\frac{1}{2} + \overline{R})T + \frac{\lambda\overline{Z^2}}{2(1-\overline{Z})} \qquad (2.6)$$

Note that in the limit as $k \to \infty$, the above equation approaches the Pollaczek-Khinchine formula for a FIFO file drum, Eq. (2.3).

3. Analysis of the SLTF Drum Scheduling Discipline

In this section we attempt to provide the same analysis for the shortest-latency-time-first (SLTF) scheduling discipline that we provided for the FIFO scheduling discipline in the last section. We will continue to model the I/O requests as a Poisson arrival process, and both file and paging drum organizations are considered. In contrast to FIFO scheduling, it is considerably simpler to analyze an SLTF paging drum than a SLTF file drum, and in fact several articles exist that analyze an SLTF paging drum with a Poisson arrival process [Coffman, 1969; Skinner, 1967]. The difficulty defining an exact, tractable model of a SLTF file drum leads here to the presentation of three alternate models that vary in the approximations they make as well as the complexity of their analysis and results.

The SLTF paging drum. For the analysis of an SLTF paging drum, let us use the same notation that was developed for the FIFO paging drum, as well as the same assumptions concerning the I/O requests: all requests are for records of size 1/k and a request is equally likely to be directed to any one of the k sectors. Referring back to Fig. 2.2, let the queue shown be for the I/O requests of a single sector, rather than the entire paging drum as in the case of the FIFO scheduling discipline. It follows as an obvious consequence of the Poisson assumption that the arrival process at an individual sector queue is also a Poisson process with rate λ/k. Let $A = \tau/k$, the time to transmit a record, and let $B = (k-1)\tau/k$, the time to return to the start of the sector after finishing the service of a request. Γ is the time between inspections of the sector queue after the queue is found to be empty, and hence Γ is simply the period of the drum revolution, τ. Interpreting Fig. 2.2 as a sector queue of a SLTF paging drum is a simple application of Skinner's model since neither A, B, nor Γ is a random variable. Note that

$$\overline{A} = \frac{\tau}{k}$$

$$\overline{Z} = \overline{\Gamma} = \tau$$

$$\overline{Z^2} = \overline{\Gamma^2} = \tau^2$$

Therefore, using Eq. (2.4), the expected waiting time for I/O requests at a SLTF paging drum is:

$$\overline{W} = \{(\frac{1}{2} + \frac{1}{k}) + \frac{\rho}{2(1-\rho)}\}\tau , \qquad 0 \le \rho < 1 . \tag{3.1}$$

$$\rho = \lambda\tau/k = \lambda\overline{R}\tau$$

Define the utilization of a drum, denoted u_d, to be the equilibrium, or long term, probability that the drum is transmitting information. From basic conservation principles it follows that the utilization of a paging drum is $\lambda\tau/k$ and hence $u_d = \rho$.

Coffman [1969] derives this same result from first principles* and those interested in a more complete discussion of SLTF paging drums are encouraged to read Coffman's article.

The SLTF file drum. For the remainder of this section we turn our attention to the SLTF file drum. This form of drum organization is becoming more important to understand as drums that provide hardware assistance to implement SLTF scheduling gain wider acceptance.

Let us use the same model of I/O requests for the SLTF file drum that was used for the FIFO file drum; in other words, the successive arrival epochs form a Poisson process with parameter λ; starting addresses are independent random variables, uniformly distributed about the drum's circumference, Eq. (2.1); and the record lengths are exponentially distributed, Eq. (2.2).

The most difficult aspect of an SLTF file drum to model is its latency intervals. Since the initial position of the read-write heads of the drum is not related (correlated) to the starting addresses of the outstanding I/O requests, S_i, it follows that the distance from the read-write heads to S_i, denoted D_i, is uniformly distributed between zero and a full drum revolution and hence

$$\Pr\{D_i > t\} = \frac{1-t}{\tau} \qquad 0 \le t < \tau;\ 1 \le i \le n.$$

Since the distances to the starting addresses from the read-write heads are independent,

$$\Pr\{D_1 > t \text{ and } D_2 > t \text{ and } \ldots \text{ and } D_n > t\} = \left[\frac{(1-t)}{\tau}\right]^n,\ 0 \le t < \tau.$$

---

* Coffman's definition of $\overline{W}$ does not include the data transmission time, and hence his expression for $\overline{W}$ is smaller than Eq. (3.1) by the quantity τ/k.

The SLTF scheduling discipline requires that the first record processed is the one whose starting address is the first to encounter the read-write heads; call the time until processing begins L_n. We can now state the cumulative distribution function of L_n, as well as its density function, mean, and variance

$$F_n(t) = \Pr\{L_n < t\} = 1 - [\frac{(1-t)}{\tau}]^n , \qquad 0 \le t < \tau; \tag{3.2}$$

$$f_n(t) = F_n'(t) = \frac{n(1-t)^{n-1}}{\tau^n} , \qquad 0 \le t < \tau;$$

$$\overline{L}_n = \frac{\tau}{n+1} ;$$

$$\mathrm{var}(L_n) = \{1 - \frac{2n}{n+1} + \frac{n}{n+2} - \frac{1}{(n+1)^2}\}\tau^2$$

Although the above distribution of L_n is relatively simple, significant simplification in the analysis will result by replacing Eq. (3.2) with the exponential distribution

$$G_n(t) = 1 - e^{-(n+1)t/\tau}, \qquad t \ge 0 . \tag{3.3}$$

Let L_n' be the random variable with distribution $G_n(t)$. $G_n(t)$ has several attractive properties as an approximation to $F_n(t)$:

$$\overline{L_n'} = \overline{L_n} = \frac{\tau}{n+1} ,$$

and if we let C_n' be the coefficient of variation for L_n' and C_n be the coefficient of variation for L_n we have

$$C_n' = \sqrt{\frac{n}{n+2}} < 1 = C_n$$

but

$$\lim_{n \to \infty} C_n' = 1$$

Consequently, $G_n(t)$ becomes a better approximation to $F_n(t)$ as the depth of the queue at the drum increases. We cannot ignore the fact $G_n(t)$ is a rough approximation for small n, but note how quickly C'_n approaches 1:

$$C'_1 = .577$$

$$C'_2 = .707$$

$$\vdots$$

$$C'_{10} = .910$$

$$\vdots$$

The above discussion makes no mention of how well the higher order moments (greater than 2) of $G_n(t)$ approximate $F_n(t)$. However, we feel somewhat justified in ignoring these higher moments since both the Pollaczek-Khinchine formula, Eq. (2.2), and the Skinner formula, Eq. (2.4) show that the expected waiting time (and queue size) is only a function of the first two moments of the service time. Unfortunately neither the Pollaczek-Khinchine or Skinner formulas directly apply here since service time, in particular latency, is queue size dependent. However, it appears likely that the first two moments of the service time are also the dominant parameters in a SLTF file drum and comparison of models based on the exponential approximation to latency shown in Sec. 4 behave very similarly to models not using the approximation.

Figure 3.1 is the model of a SLTF file drum based on the assumptions discussed above. We have a Poisson arrival process with parameter λ, a single queue, and the server is made up of two exponential servers in series. The second server, with servicing rate μ, models the transmission of records and reflects our assumption of exponentially distributed record lengths.

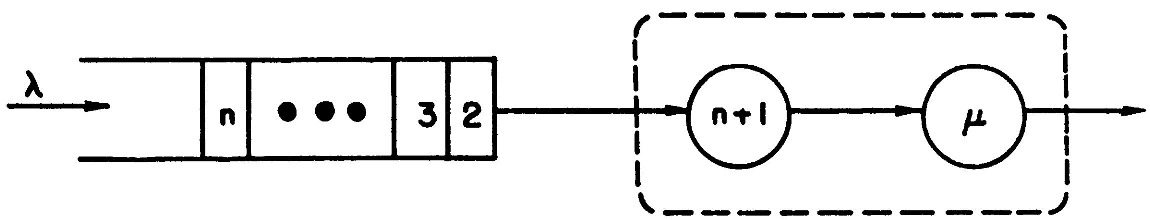


Figure 3.1. Two-stage Markov model of the SLTF file drum.

The first of the two servers has service time distribution $G_n(t)$, and hence service rate $(n+1)/\tau$, and models the latency incurred accessing a record when there are n I/O requests demanding service. This 'latency' server is the result of our recent discussion and one further assumption: we assume that the latency is purely a function of queue depth and not related to the past performance of the drum. Although this is normally a very good approximation, it is not entirely true. In Sec. 4 we will explore why this is an approximation and show its relation to Feller's waiting-time paradox [Feller, 1970].

Since both servers in our model have exponential service times and the arrival process is Poisson, our model is a birth-and-death Markov process. Let $E_{o,n}$ be the state where n I/O requests are queued or in service and the latency server, server 0, is active; define $E_{0,0}$ as the idle state. Similarly, let $E_{1,n}$ be the state with n I/O requests and the transmission server, server 1, active. If $P_{i,j}(t)$ is the probability of being in state $E_{i,j}$ at time t, then the differential equations describing the SLTF file drum are

$$P'_{0,0}(t) = -\lambda\ P_{0,0}(t) + \mu\ P_{1,1}(t),$$

$$P'_{1,1}(t) = -(\lambda + \mu)\ P_{1,1}(t) + \frac{2}{\tau}\ P_{0,1}(t),$$

$$P'_{0,n}(t) = -(\lambda + \frac{n+1}{\tau})\ P_{0,n}(t) + \lambda\ P_{0,n-1}(t) + \mu\ P_{1,n+1}(t), \qquad n = 1,2,\ldots;$$

$$P'_{1,n}(t) = -(\lambda+\mu)\ P_{1,n}(t) + \lambda\ P_{1,n-1}(t) + \frac{n+1}{\tau}\ P_{0,n}(t), \qquad n = 2,3,\ldots$$

In this article we are primarily interested in the steady state solution, or more precisely, the solution at statistical equilibrium. Consequently let $p_{i,j} = \lim_{t \to \infty} P_{i,j}(t)$, and the above set of differential equations reduce to the following set of recurrence relations (often called balance equations):

$$\lambda p_{0,0} = \mu\, p_{1,1}\ , \tag{3.4}$$

$$(\lambda+\mu)p_{1,1} = \frac{2}{\tau}\, p_{0,1}\ , \tag{3.5}$$

$$(\lambda + \frac{n+1}{\tau})\, p_{0,n} = \lambda p_{0,n-1} + \mu\, p_{1,n+1}\ , \qquad n = 1,2,\ldots; \tag{3.6}$$

$$(\lambda+\mu)p_{1,n} = \lambda p_{1,n-1} + \frac{n+1}{\tau}\, p_{0,n+1}\ , \qquad n = 2,3,\ldots \tag{3.7}$$

The most direct solution of the above set of balance equations lies in working with their associated generating functions:

$$P_0(z) = \sum_{0 \le n < \infty} p_{0,n}\, z^n$$

$$P_1(z) = \sum_{0 < n < \infty} p_{1,n}\, z^n\ .$$

Equation (3.6), using Eq. (3.4) as an initial condition yields

$$z\, P_0'(z) + (1 + \lambda\tau - \lambda\tau z)\, P_0(z) = \frac{\mu\tau}{z}\, P_1(z) + p_{0,0}\ . \tag{3.8}$$

Similarly, Eqs. (3.5) and (3.7) yield

$$\tau(\mu + \lambda - \lambda z)\, P_1(z) = z\, P_0'(z) + P_0(z) - p_{0,0}\ . \tag{3.9}$$

$P_1(z)$ can be eliminated from the above set of simultaneous equations and we get a linear, first-order differential equation in $P_0(z)$:

$$P_0'(z) + (\frac{1}{z} - \lambda\tau + \frac{\lambda\tau\rho}{\rho z-1})\, P_0(z) = \frac{1}{z}\, p_{0,0} \tag{3.10}$$

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

$$\int(\frac{1}{z} - \lambda\tau + \frac{\lambda\tau\rho}{\rho z-1})dz = \ln z - \lambda\tau z + \lambda\tau\cdot\ln(\rho z-1) + C$$

as an integrating factor, we find the following explicit form for $P_0(z)$

$$P_0(z) = \frac{p_{0,0}e^{\lambda\tau z}}{z(1-\rho z)^{\lambda}}\left[\int e^{-\lambda\tau z}(1-\rho z)^{\lambda\tau}dz + C\right] \qquad (3.11)$$

where C is the constant of integration. We can eliminate C by replacing the indefinite integral by the correct definite integral. Clearly $\lim_{z \to 0} P_0(z) = p_{0,0}$, and thus C vanishes if the indefinite integral is replaced by a definite integral with limits from 0 to z:

$$P_0(z) = \frac{e^{\lambda\tau z}p_{0,0}}{z(1-\rho z)^{\lambda}}\int_0^z e^{-\lambda\tau w}(1-\rho w)^{\lambda\tau}dw$$

A more useful generating function for the subsequent analysis than $P_0(z)$ is

$$Q(z) = \sum_{0 \le n < \infty} (p_{0,n} + p_{1,n})z^n = P_0(z) + P_1(z) .$$

From Eqs. (3.8) and (3.9) it follows that

$$P_1(z) = \frac{\rho z}{1-\rho z} P_0(z)$$

and hence

$$Q(z) = \frac{1}{1-\rho z} P_0(z)$$

$$Q(z) = \frac{e^{\lambda\tau z}p_{0,0}}{z(1-\rho z)^{-\lambda\tau+1}}\int_0^z e^{-\lambda\tau w}(1-\rho w)^{\lambda\tau}dw \qquad (3.12)$$

It is impossible to integrate, in closed form, the integral in the above equations. However it is in the form of the well-known, and extensively tabulated, incomplete gamma function [Abramowitz and Stegun, 1964]:

$$\gamma[\alpha,x] = \int_0^x e^{-y}\, y^{\alpha-1}\, dy$$

Restating Eq. (3.12) in terms of the incomplete gamma function gives

$$Q(z) = \frac{\mu\, e^{(\lambda z-\mu)\tau} p_{0,0}}{\lambda z} \left\{ \frac{\gamma[\lambda\tau+1,\ -\mu\tau] - \gamma[\lambda\tau+1, (\lambda z-\mu)\tau]}{[(\lambda z - \mu)\tau]^{\lambda\tau+1}} \right\}.$$

It is now possible to find an explicit expression for $p_{0,0}$, the equilibrium probability that the drum is idle, since it is clear that $\lim_{z \to 1} Q(z) = 1$. Hence, from Eq. (3.12)

$$p_{0,0} = e^{-\lambda\tau}(1-\rho)^{\lambda\tau+1} \left\{ \int_0^1 e^{-\lambda\tau w}(1-\rho w)^{\lambda\tau}\, dw \right\}^{-1} \tag{3.13}$$

Equation (3.12) can be used to find the mean queue length, $\overline{L}$:

$$\overline{L} = \lim_{z \to 1} Q'(z) = \lambda\tau - 1 + \frac{\rho(\lambda\tau+1) + p_{0,0}}{1-\rho}$$

Using Little's formula, $\overline{L} = \lambda\overline{W}$ [cf. Jewell, 1967], we can state the expected waiting time of I/O requests at an SLTF file drum for this two-stage Markov model:

$$\overline{W} = \frac{\mu\tau+1}{(\mu-\lambda)\tau} + \frac{1}{\lambda\tau} \left[\left\{ \int_0^1 \left[\frac{e^{\lambda\tau(1-w)}(1-\rho w)}{1-\rho} \right]^{\lambda\tau} dw \right\}^{-1} - 1 \right] \tag{3.14}$$

Figure 3.2 displays the expected waiting time, $\overline{W}$, as a function of both λ and μ for this two-stage model of the SLTF file drum.

The expression for $\overline{W}$ is not given in terms of the incomplete gamma function since numerical integration of the integral in Eq. (3.14) is routine.

In the analysis of the two-stage model of an SLTF file drum just discussed, a concerted attempt was made to closely approximate the behavior of the drum. It is interesting, even if for comparative purposes only, to briefly discuss a simplification of the two-stage model. The

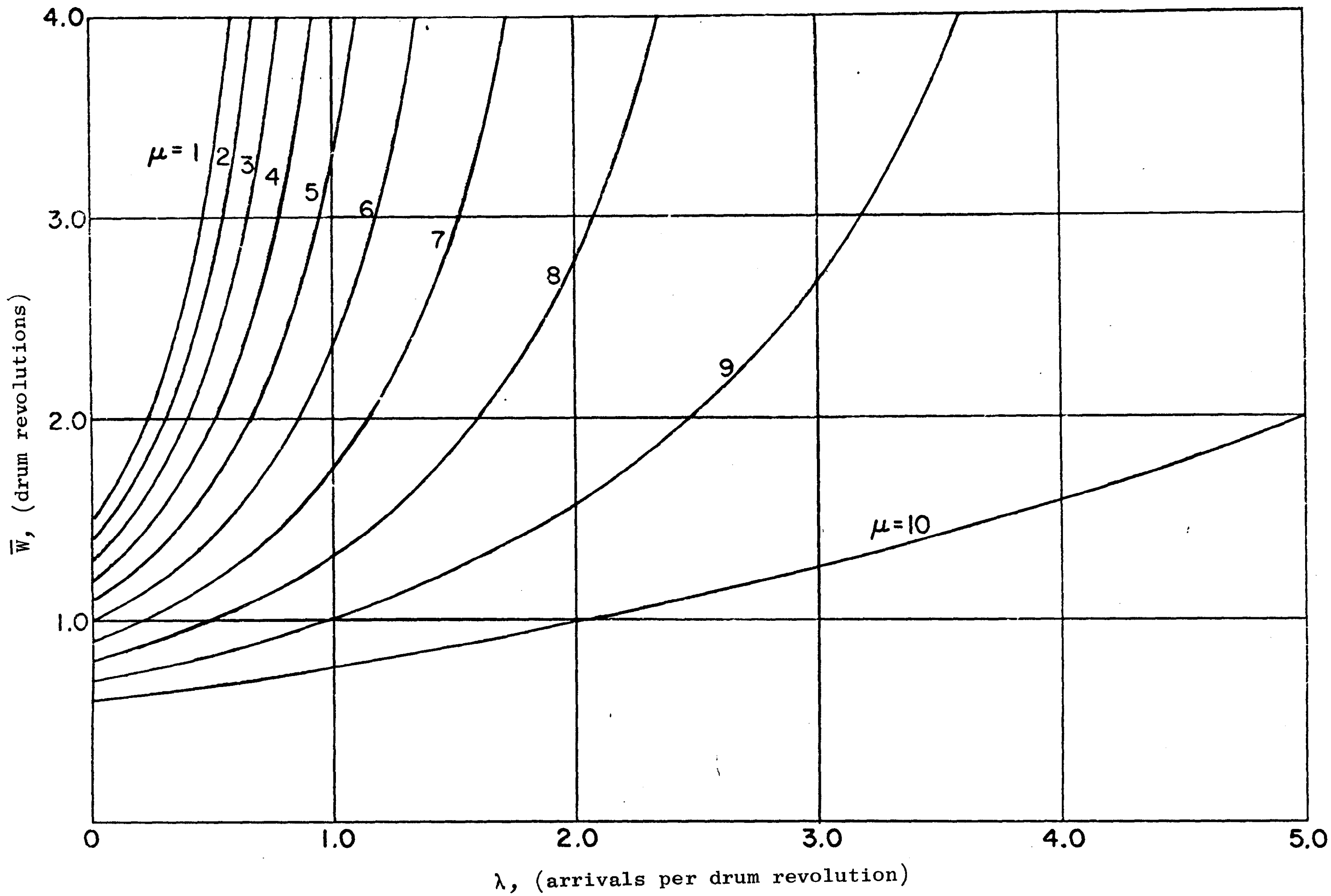


Figure 3.2. $\overline{W}$, the expected waiting time, of the two-stage Markov model of the SLTF file drum.

most obvious simplification is to replace the two series-coupled servers in Fig. 3.1 by a single exponential server with the same mean service rate. Let μ_n denote the mean service rate of the single server when there are n I/O requests in the queue, then

$$\frac{1}{\mu_n} = \frac{\tau}{n+1} + \frac{1}{\mu} = \frac{\mu\tau + n + 1}{\mu(n+1)} .$$

It follows in a straightforward manner that the balance equations for the one-stage model, analogous to Eqs. (3.4)-(3.7), are

$$\lambda\, p_0 = \mu_1\, p_1 , \tag{3.15}$$

$$(\lambda + \mu_n)p_n = \lambda p_{n-1} + \mu_{n+1}\, p_{n+1} , \qquad n = 1,2,\ldots \tag{3.16}$$

The above set of recurrence relations can be solved directly with forward substitution and yield:

$$p_1 = \frac{\lambda}{\mu_1} p_0 ,$$

$$p_2 = \frac{\lambda^2}{\mu_1\mu_2} p_0 ,$$

and in general

$$p_n = \frac{\rho^n}{\mu\tau + 1} \binom{\mu\tau + n + 1}{n + 1} p_0 , \qquad n \geq 0 ; \tag{3.17}$$

where $\rho = \frac{\lambda}{\mu}$.

The sequence $\{p_n\}$ must sum to unity, i.e.

$$\sum_{0 \leq n < \infty} \frac{\rho^n}{\mu\tau+1} \binom{\mu\tau + n + 1}{n + 1} p_0 = 1$$

and hence

$$p_0^{-1} = \frac{1}{\mu\tau+1} \sum_{0 \leq n < \infty} \binom{\mu\tau + n + 1}{n + 1} \rho^n .$$

With the aid of the binomial theorem the above relation reduces to

$$p_0 = \frac{\rho(\mu T+1)(1-\rho)^{\mu T+1}}{1-(1-\rho)^{\mu T+1}} \tag{3.18}$$

Using Eqs. (3.17) and (3.18) we see that for this simple model of a SLTF file drum we are able to get explicit expressions for the probability of being in any state E_n. However, only $\overline{W}$ will be used in comparing this single-stage model to the two-stage model and $\overline{W}$ is most easily determined from the generating function for the model.

$$P(z) = \sum_{0 \le n < \infty} p_n z^n$$

$$= \sum_{0 \le n < \infty} \frac{\rho^n}{\mu T+1} \binom{\mu T + n + 1}{n + 1} p_0 \, z^n$$

Applying the binomial theorem as before, and using the expression for p_n in Eq. (3.17),

$$P(z) = \frac{(1-\rho)^{\mu T+1}[1-(1-\rho z)^{\mu T+1}]}{z(1-\rho z)^{\mu T+1}[1-(1-\rho)^{\mu T+1}]} \tag{3.19}$$

The expected waiting time for I/O requests at the single-stage model of a SLTF file drum is

$$\overline{W} = \lim_{z \to 1} P'(z) = \frac{1}{\lambda}\left(\frac{\rho(\mu T+1)}{(1-\rho)(1-(1-\rho)^{\mu T+1})} - 1\right). \tag{3.20}$$

For our third model of the SLTF file drum we turn to an article by Abate and Dubner [1969]. Although the majority of their paper is concerned with a particular variant of the SLTF scheduling discipline implemented by Burroughs [1970], their approach can be applied to the SLTF file drum discussed here.*

---

* Abate and Dubner do in fact briefly discuss the model of a SLTF file drum presented here, and Eq. (3.22) is their Eq. (14), with appropriate changes in notation.

Abate and Dubner's analysis, when applied to a file drum with pure SLTF scheduling, is surprisingly simple. They divide the waiting time into three, independent terms:

$$W = D + R + \tau(K - 1) \quad (3.21)$$

D is the distance from the read-write heads to the start of the I/O request at the instant of the request's arrival. Consistent with our previous discussions, D is a random variable uniformly distributed from zero to a full drum revolution; hence $\overline{D} = 1/2$. R is the length of the record that must be read or written to the drum; Abate and Dubner make no assumptions concerning the distribution of D; they only use the mean of the record length, $\overline{R}$. With the same conservation argument following Eq. (3.1), Abate and Dubner also note the drum utilization is just λ/μ, or ρ, and that on the average, the drum is free to begin servicing a record at its starting address with probability $(1 - \rho)$, which is the equilibrium probability the drum is not busy transmitting a record. Furthermore, they assume successive attempts to read a record can be modeled as independent, Bernoulli trials with probability of success $(1 - \rho)$. The random variable K in Eq. (3.21) is the number of trials required until the record is serviced, and hence K is geometrically distributed with probability mass function:

$$\Pr\{K = k\} = (1 - \rho)\rho^{k-1}, \qquad k = 1,2,\ldots$$

and mean

$$\overline{K} = \frac{1}{1 - \rho} .$$

Therefore, the expected waiting time for I/O requests at Abate and Dubner's model of an SLTF file drum is:

$$\overline{W} = \{\tfrac{1}{2} + \overline{R} + \frac{\rho}{1 - \rho}\}T \ . \qquad (3.22)$$

This concludes the development of models for the SLTF file drum. Unlike the FIFO file drum, the FIFO paging drum, and the SLTF paging drum, we do not have a model that is exact; in each of the three models of the SLTF file drum, assumptions are made that do not exactly reflect the actual behavior of a SLTF file drum.

4. Verification of SLTF file drum models

We have presented three different models of the SLTF file drum: the one-stage Markov model; the 2-stage Markov model, and Abate and Dubner's model. In order to resolve the relative merits of these models we will compare each of them to the results of a simulation model of the SLTF file drum. The simulation uses all of our original assumptions, i.e. (1) Poisson arrival process, (2) exponential distribution of record lengths and (3) starting addressing uniformly distributed around the surface of the drum.

The precision of the summary statistics of the simulation model is described in detail in [Fuller, 1972B]. All the points on the graphs in this article represent the result of simulation experiments that are run until 100,000 I/O requests have been serviced; this number of simulated events proved sufficient for the purposes of this article. The sample means of the I/O waiting times, for example, are random variables with standard deviations less than .002 for ρ = .1 and slightly more than .1 for ρ = .75.

A plot of the results from the models of the SLTF file drum are shown in Fig. 4.1 for an expected record size of 1/3 of the drum's

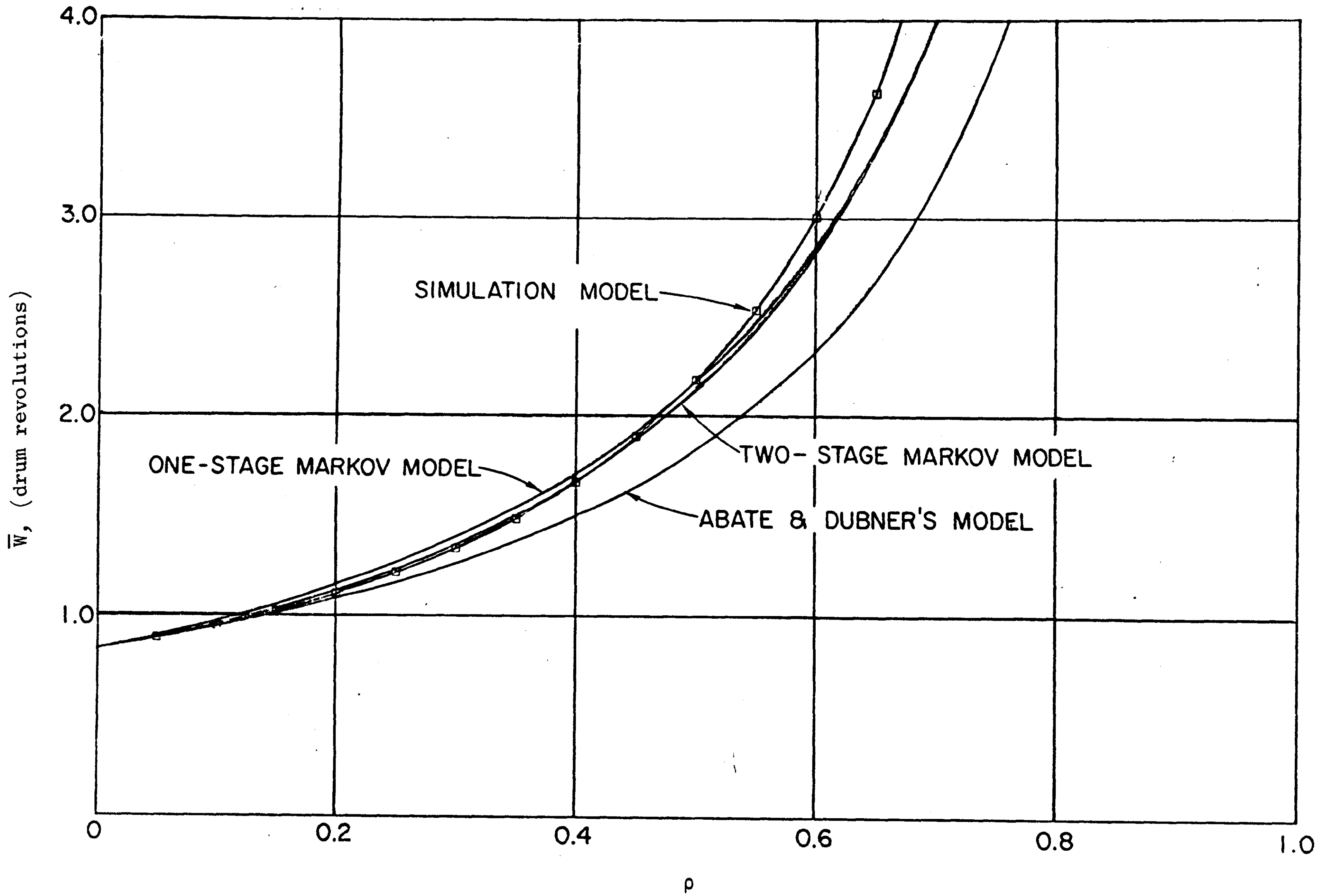


Figure 4.1. The expected waiting time of the four models of the SLTF file drum for $\overline{R} = 1/3$.

circumference. Specifically, $\overline{W}$, the mean waiting time for I/O requests at the SLTF file drum are shown as a function of drum utilization, ρ. Figure 4.2 is a similar plot except the expected records size is 1/8 of the drum's circumference. We use 1/3 because this is close to the measured value of record sizes discussed in Sec. 2 and 1/8 because Abate and Dubner thought their model should be a good approximation for $\overline{R} < 1/4$. For $\overline{R} = 1/3$ and $\rho < .45$ the results are encouraging: the two-stage model tracks the simulated waiting time very closely, but is a slight overestimate; the one-stage model follows the simulation fairly closely but is more of an overestimate than the two-stage model; and Abate and Dubner's model is an underestimate, but they warned their model might not apply very accurately for $\overline{R} > 1/4$. When we estimated the latency intervals by the exponential distribution with mean of $\tau/(n+1)$, we slightly overestimated the coefficient of variation. It has been shown in other queueing models that the mean waiting time is positively related to the mean service time as well as the coefficient of variation of the service time. Therefore, it is not surprising that the two-stage model is an overestimate of the results found by simulation. Lumping the latency and transmission servers into one server in the one-stage model further over-approximates the coefficient of variation of service time, even though the mean is still exact, and hence the one-stage model is a larger overestimate of the waiting time than the two-stage model.

Examination of Fig. 4.2 for small ρ, i.e. $\overline{R} = 1/8$ and $p < .45$, shows several significant features of the SLTF file drum models. Most striking is the degradation of the one and two-stage models. The reason, however, is quite simple; since the record lengths are not uniformly

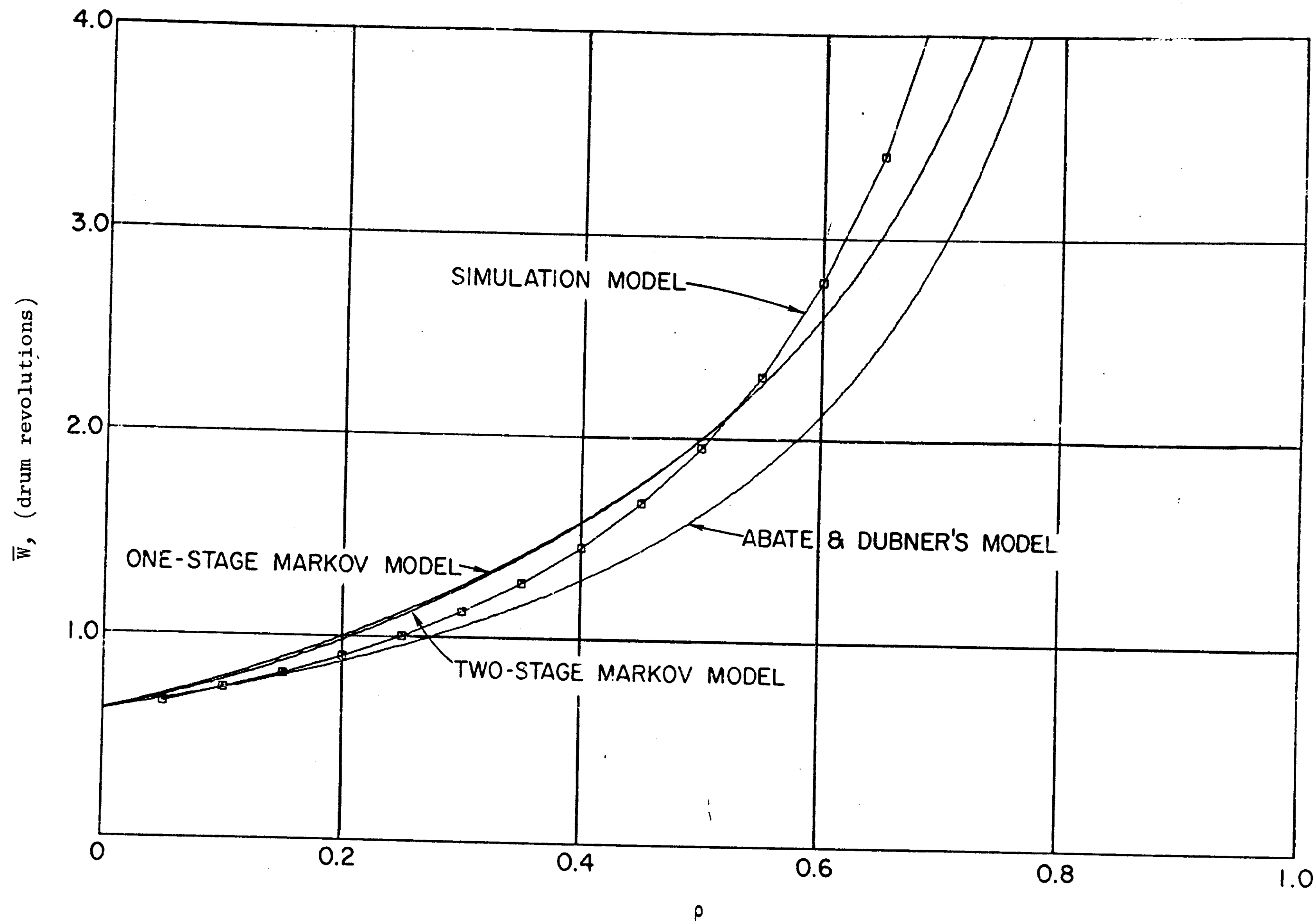


Figure 4.2. The expected waiting time of the four models of the SLTF file drum for $\overline{R} = 1/8$.

distributed over some integer number of drum revolutions, the finishing address of an I/O request is not independent of its starting address. In particular, as the expected record length goes to zero, the average wait time for an I/O request approaches $\tau/2$, regardless of the arrival rate of I/O requests.

Figure 4.2 illustrates an interesting phenomenon with respect to the Abate and Dubner model: no change, to within the accuracy of the simulation model, is detectable between the difference in the simulation model and Abate and Dubner's model for $\overline{R} = 1/3$ and $R = 1/8$. Counter to original intuition [Abate and Dubner, 1969], no improvement in the Abate and Dubner model is detected as $\overline{R}$ becomes smaller. If instead of comparing the curves for $\overline{R} = 1/3$ and $\overline{R} = 1/8$ at points of equal ρ, as is suggested by Figs. 4.1 and 4.2, we compare them at points of equal λ, then as $\overline{R}$ is reduced, we do see a significant improvement in the Abate and Dubner model.

The most outstanding feature of Figs. 4.1 and 4.2 is when ρ becomes large, i.e. $\rho > 0.45$: the one and two-stage models, as well as Abate and Dubner's model, are underestimates of the expected waiting time found by the simulation model, and the underestimates become increasingly pronounced as $\rho \to \infty$. The following illustration gives an intuitive explanation for this phenomena. Suppose we have n outstanding I/O requests. From the arguments that lead to Eq. (3.3) we know the expected distance from the read-write heads to the first starting address is $1/(n+1)$. However, the reasoning can be applied to distances opposite to the drum's rotation and again the distance from the read-write heads to the closest starting address behind the read-write heads is $1/(n+1)$. Hence the expected size of the interval between the starting addresses, punctuated by the read-write heads, is 2/(n+1). If we consider the n I/O requests by themselves, however, since the starting addresses are independent random variables, from symmetry it follows that the expected distance between adjacent starting addresses is 1/n. Therefore we see that when we randomly position the read-write head on the drum, we are most likely to fall into a larger than average interval between starting addresses. In other words, those starting addresses that end large intervals are most likely to be chosen first by an SLTF schedule and as the SLTF schedule processes I/O requests, the remaining starting addresses exhibit an increasing degree of clustering, or correlation. A more complete discussion of this phenomenon, and related topics, are discussed by Feller [1970] under the general heading of <u>waiting-time paradoxes</u>.

Let us consider in more detail why the two-stage model underestimates the wait time. Let $S_{(1)}$, $S_{(2)}$, and $S_{(3)}$ denote the starting addresses in order of increasing distance from the read-write head, denoted H, and let $F_{(1)}$ be the finishing address of the record beginning with $S_{(1)}$. Let $F_{(1)}$ be a random variable uniformly distributed around the circumference of the drum, and for this example suppose it is independent of $S_{(1)}$. Suppose initially there are n I/O requests, the $S_{(i)}$'s, H, and $F_{(1)}$'s make up n+2 random variables, and they can form the following three basic configurations with respect to H:

$$H \; F_{(1)} \; S_{(1)} \; S_{(2)} \cdots S_{(n)} \tag{4.1a}$$

$$H \; S_{(1)} \cdots S_{(n)} \; F_{(1)} \tag{4.1b}$$

$$H \; S_{(1)} \cdots S_{(j)} \; F_{(1)} \; S_{(j+1)} \cdots S_{(n)} \; ; \quad 1 < j < n-1 \tag{4.1c}$$

We know that the density function of the distance between any two adjacent random variables is $(n+2)(1-x)^{n+1}$, $0 \leq x < 1$, and has a mean of $1/(n+2)$. Moreover, an obvious extension of the above expression leads to the following distribution functions for the difference between any two random variables separated by k-1 other points on the drum [Feller, 1970]:

$$(n+2)\binom{n+1}{k-1} x^{k-1}(1-x)^{n-k} , \qquad 0 \leq x < 1 .$$

with mean $k/(n+2)$.

Referring to the n+2 situations in (4.1) we can see that

$$\Pr(L_n = t) = (n+1)(1-t)^n , \qquad 0 \leq t < \tau$$

and

$$\overline{L}_n = \tau/n+1$$

and this is precisely what we found $\overline{L}_n$ to be by a much simpler argument in Sec. 3. However, now consider $\overline{L}_{n-1}$ in the same example. In (4.1a) and (4.1b) the latency is the distance from $F_{(1)}$ to $S_{(2)}$ and in the n-1 cases of (4.1c) the latency is the distance from $F_{(1)}$ to $S_{(j+1)}$. Hence

$$\overline{L}_{n-1} = \frac{1}{n+1} \cdot \frac{2}{n+2} + \frac{1}{n+1} \cdot \frac{3}{n+2} + \frac{n-1}{n+1} \cdot \frac{1}{n+2}$$

$$= \frac{n+4}{(n+2)(n+1)} > \frac{1}{n} .$$

The above inequality illustrates that latency is dependent on the past history of the drum; and that unlike the SLTF and FIFO paging drums, as well as the FIFO file drum, the SLTF file drum does not experience a Markov epoch upon completion of an I/O request. The only instances in which the drum's future performance truly uncouples from the past behavior is when the drum is idle. This fact makes the precise analysis of a SLTF file drum, even with the simple arrival process described here, very difficult.

In concluding this section we might pause to consider the relative errors introduced by the approximations used to make a tractable model of the SLTF file drum. In the best analytic model, the two-stage Markov model, we made three assumptions, all with regard to latency, that are not entirely correct: (1) we approximated Eq. (3.2) by the exponential distribution Eq. (3.3), (2) we assumed the position of the read-write heads at the end of a record transmission is independent of its position at the start of transmission, and (3) we assumed that latency is a function only of the current queue length and not influenced by previous queue sizes. The first two approximations are dominant for $\rho_d < .4$ but the error introduced by these approximations is slight for large $\overline{R}$, and as $\overline{R}$ becomes small, the second assumption causes large overestimations of numbers. However, for larger ρ, the third approximation, related to Feller's waiting time paradox, dominates and the error it introduces becomes severe as $\rho \to 1$. Consequently, attempts to make a significant practical improvement over the two-stage model should not dwell on removing the exponential approximation to the latency interval, but rather on the second and third approximations.

5. An Empirical Model of the SLTF File Drum

In this section we develop a simple, empirical expression for the expected waiting time of the SLTF file drum. Such an empirical expression has a limited utility, but in conjunction with the other models available for the SLTF file drum it can be a useful tool.

The expressions for the expected waiting time for the FIFO file drum, the FIFO paging drum, the SLTF paging drum, and Abate and Dubner's model of the SLTF file drum all have the basic hyperbolic form:

$$\overline{W} = \{\tfrac{1}{2} + \overline{R} + b(\frac{\rho}{1-\rho})\}\tau \qquad (5.1)$$

and consequently this form is a likely candidate for a model of the SLTF file drum. All that must be empirically determined is the coefficient b, and whether or not an expression of the form of Eq. (5.1) is adequate to describe the SLTF file drum. Figure 5.1 is a plot of

$$[\overline{W} - (\tfrac{1}{2} + \overline{R})](1 - \rho) \qquad (5.2)$$

for all four models of the SLTF file drum for $\overline{R} = 1/3$. Curves that are of the form of Eq. (5.1) will appear as straight lines in Fig. 5.1. Note Abate and Dubner's model appears as a straight line with a slope of one, and both of the Markov models approach finite, nonzero value as $\rho \rightarrow 1$, indicating they are also fundamentally hyperbolic in form. The simulation points, however, do not appear to approach a finite value as $\rho \rightarrow 1$, and hence Eq. (5.1) does not capture all the significant behavior of the SLTF file drum. In order to model the part of the SLTF waiting time that is growing faster than $\rho/(1-\rho)$, we will add another term to Eq. (5.1) to get:

$$\overline{W} = \{\tfrac{1}{2} + \overline{R} + b\frac{\rho}{1-\rho} + c(\frac{\rho}{1-\rho})^2\}\tau \ . \qquad (5.3)$$

From Fig. 5.1 we see that for small ρ, the simulation curve appears to have a slope of about one, and hence if Eq. (5.2) is an adequate model of the SLTF file drum, the expression

$$[\overline{W} - (\tfrac{1}{2} + \overline{R}) - \frac{\rho}{1-\rho}]\frac{(1-\rho)^2}{\rho}\tau \qquad (5.4)$$

should appear as a straight line when plotted as a function of ρ. Figure 5.2 shows Eq. (5.4) as a function of ρ. Clearly the simulation results are not growing as fast as Eq. (5.3) suggests and it over-estimates the rate of growth of $\overline{W}$ as $\rho \rightarrow 1$.

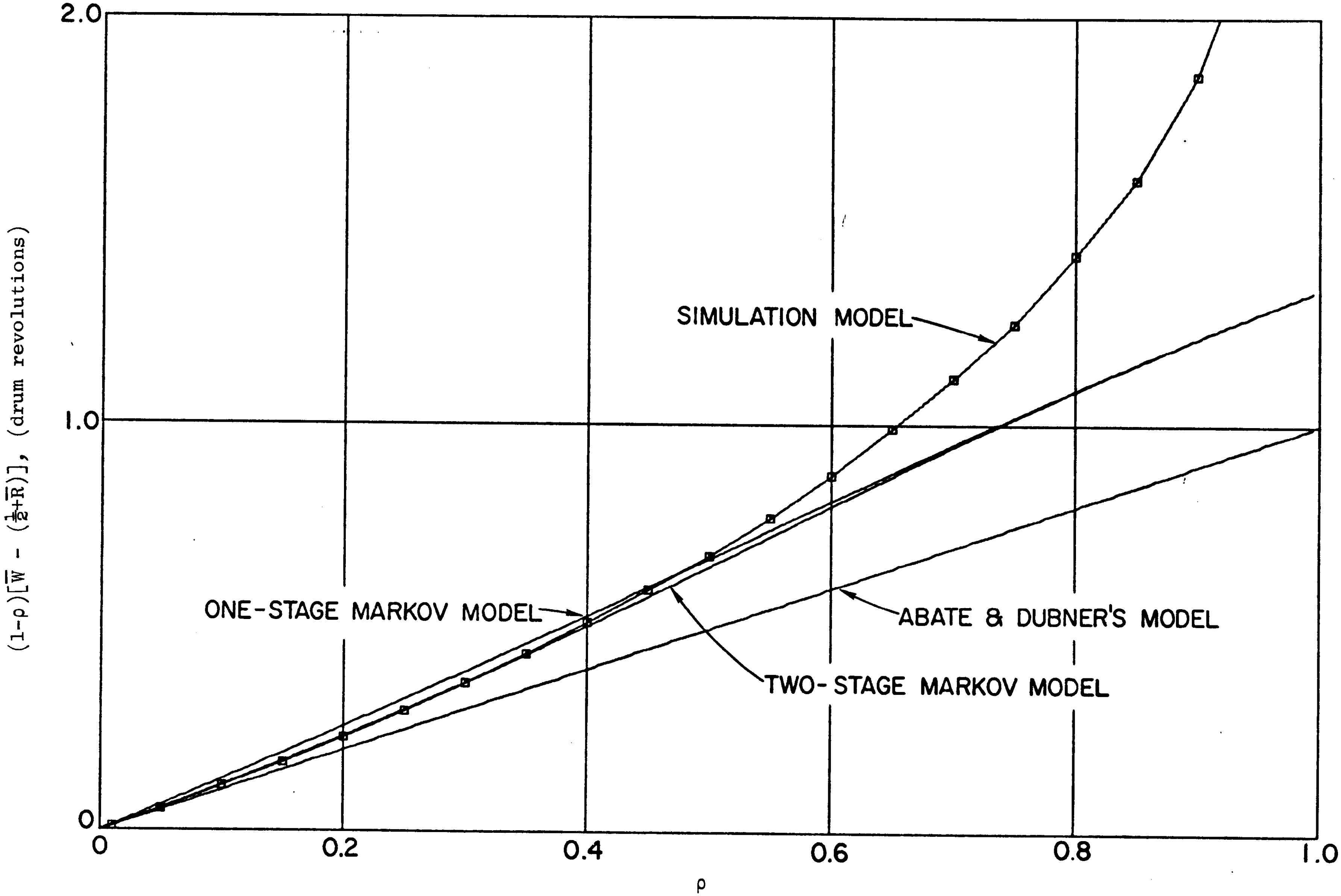


Figure 5.1. The expected waiting time of the SLTF file drum models transformed to test hyperbolic dependence on $(1-\rho)$.

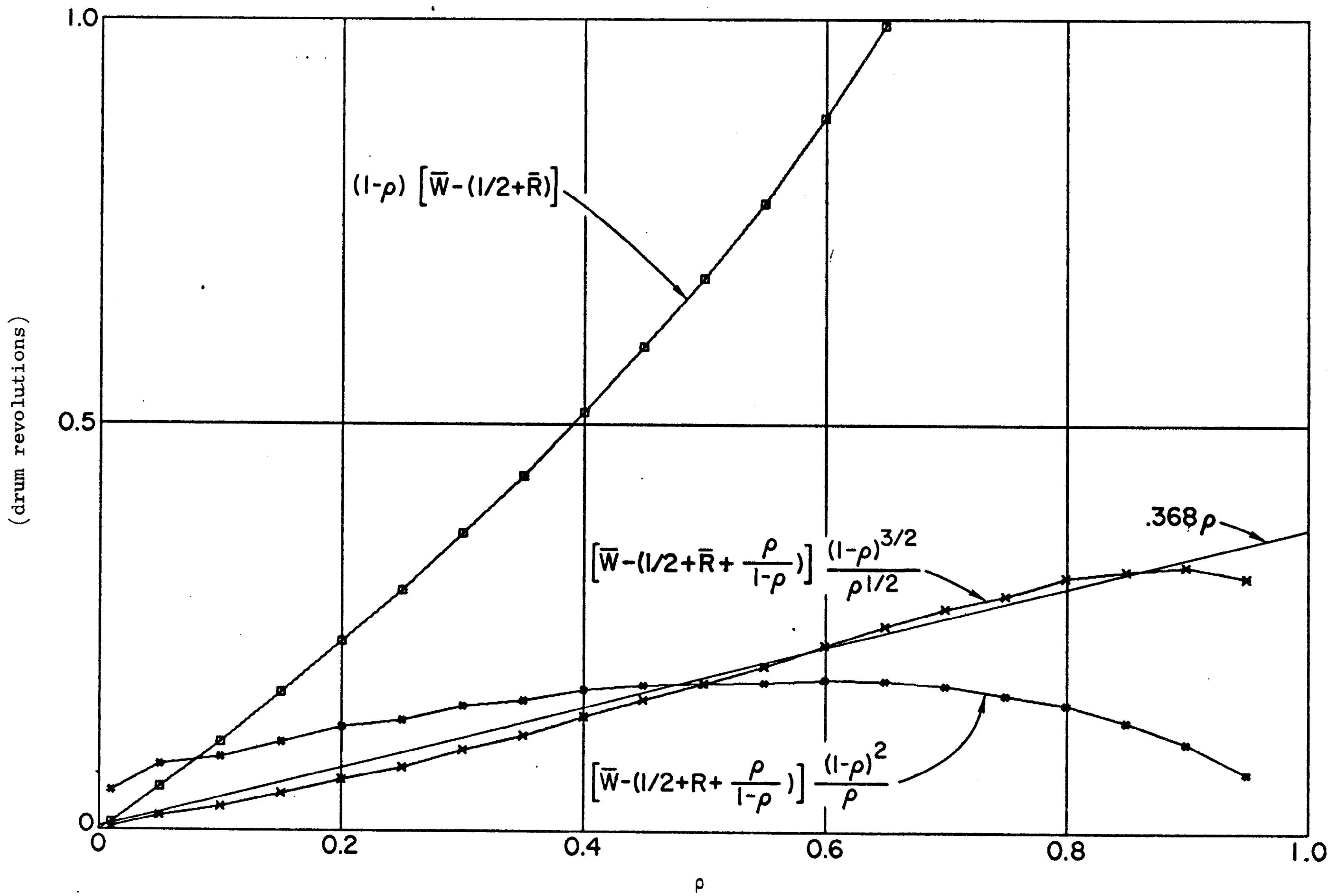


Figure 5.2. Additional transformations of the expected waiting time for the simulation model.

Another expression for $\overline{W}$ that may lead to an adequate empirical model is

$$\overline{W} = \{\tfrac{1}{2} + \overline{R} + \frac{\rho}{1-\rho} + c(\frac{\rho}{1-\rho})^{3/2}\}\tau \ . \tag{5.5}$$

In Fig. 5.2 we have plotted

$$[\overline{W} - (\tfrac{1}{2} + \overline{R}) - \frac{\rho}{1-\rho}] \frac{(1-\rho)^{3/2}}{\rho^{1/2}} \tag{5.6}$$

and show that to a first approximation it has a slope of .368. This is an approximate model, but until we have more understanding of the SLTF file drum, and have some idea of the form of the expected waiting time, it is pointless to add refinements to Eq. (5.5). Therefore, our empirical model is

$$\overline{W} = \{\tfrac{1}{2} + \overline{R} + \frac{\rho}{1-\rho} + .368(\frac{\rho}{1-\rho})^{3/2}\}\tau \tag{5.7}$$

and Fig. 5.3 shows the expected waiting time of the empirical and simulation models for $\overline{R} = 1/3$. The closeness of the empirical model to the simulation results makes them almost indistinguishable in the figure. Eq. (5.7) is also a very good model for $\overline{R} = 1/8$.

6. Cyclic queue models of central processor-drum computer systems

The previous models of drum storage units have all considered the performance of a drum as an isolated system by means of the Poisson assumption. That is, requests arrive at the drum from some undefined source such that the interarrival times follow an exponential distribution. In this section we extend the previous models by removing the Poisson assumption and consider a simple case of the type of environment that generates requests to be processed by the drum. Cyclic queue models with a FIFO file drum can be treated by Gaver's [1967]

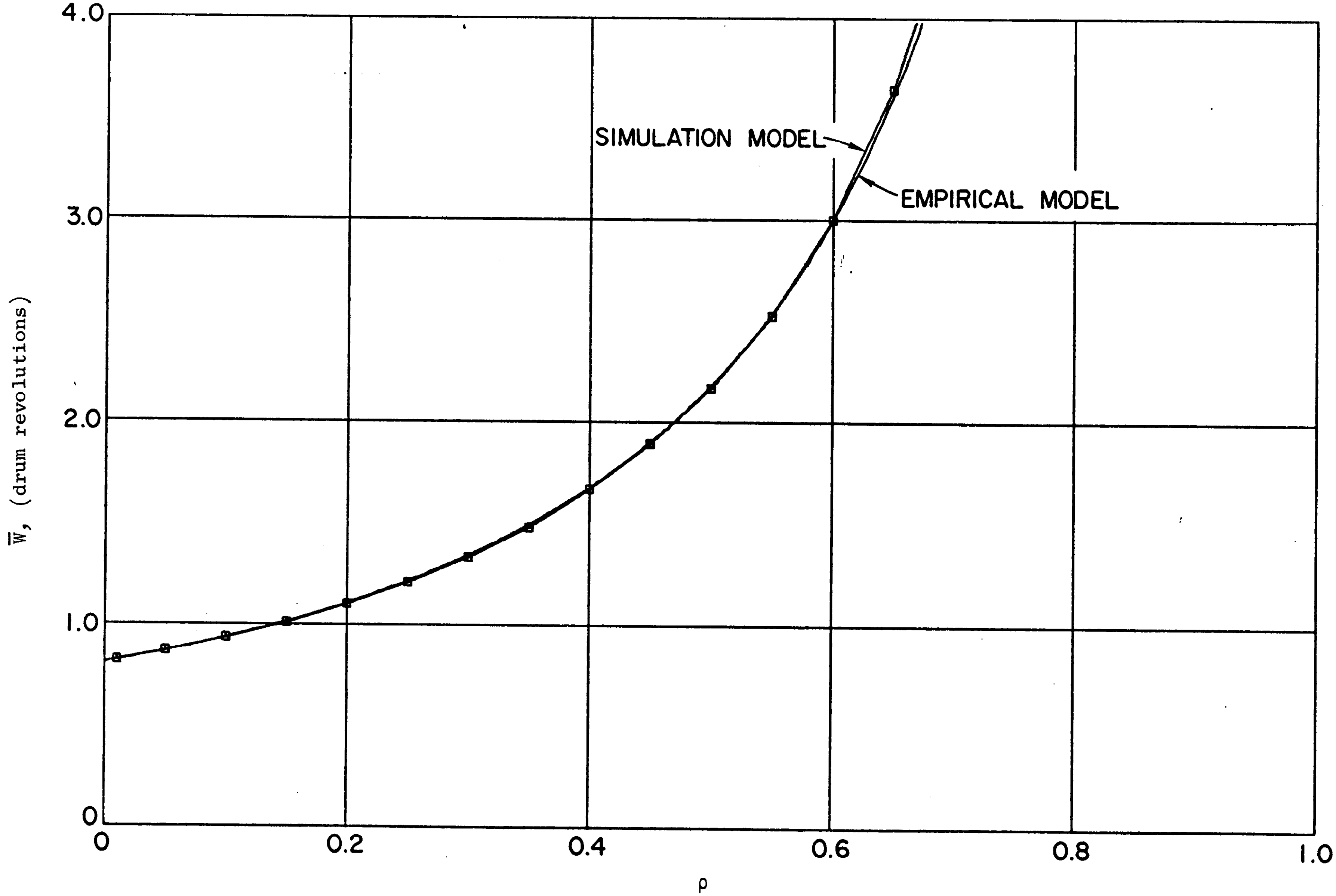


Figure 5.3. The expected waiting time of the empirical and simulation models of the SLTF file drum for $\mu = 3$.

analysis, and for a SLTF and FIFO paging drum, an exact analysis has not been found. The models of this section are cyclic queue models with two processors. One of the processors is a model of an SLTF file drum and the other processor represents a central processing unit (CPU). A fixed number of customers (jobs) alternate between waiting for and receiving service at the CPU and waiting for and receiving service at the drum. Thus the completions of one processor form the arrivals at the other processor. The structure of actual computer systems is typically more complex than this simple cyclic model. Such a model does allow us to consider the feedback effects of drum scheduling disciplines and to evaluate the accuracy of the models and their appropriateness for use in more complex feedback queueing models of computer systems. Closed queueing models of the type considered by Jackson [1963] and Gordon and Newell [1967] can be analyzed when the service rate of a processor is a function of the queue length at that processor and the service time is exponentially distributed. The one-stage model of an SLTF file drum is an important example of this type of processor.

The two-stage cyclic model. Figure 6.1 shows a cyclic queue model incorporating the two-stage Markov model of an SLTF file drum of the previous sections and a single CPU. The CPU server processes requests from its queue and its completions are routed to the drum queue. Completions at the drum are routed to the CPU queue. There are a fixed number, m, of customers in this cycle. This number, m, is called the degree of multiprogramming. The CPU processing times are exponentially distributed with parameter λ. The drum is composed of two stages, the first representing latency time, exponentially distributed with rate $(n+1)/\tau$ where n is the size of the drum queue and τ is the period of

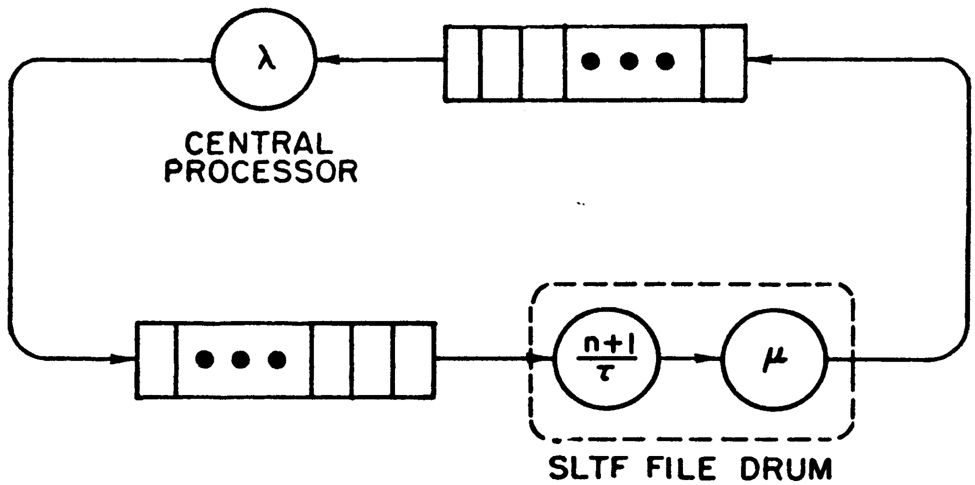


Figure 6.1. Cyclic queue model of CPU-drum system with two-stage Markov model of SLTF file drum.

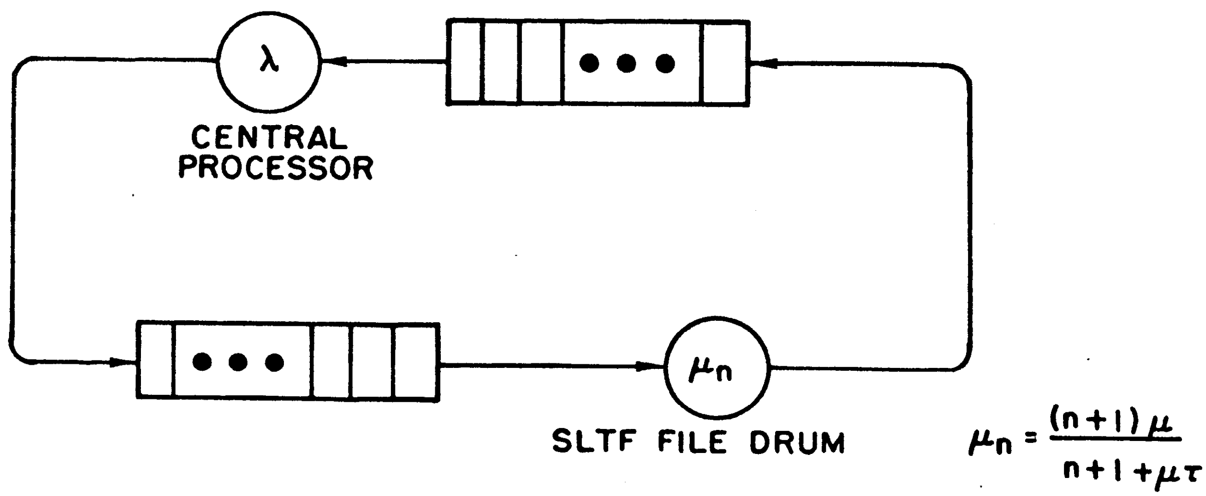


Figure 6.2. Cyclic queue model of CPU-drum system with one-stage Markov model of SLTF file drum.

revolution of the drum. The second stage represents transmission time, exponentially distributed with rate μ.

We can easily write down the balance equations describing the equilibrium distribution of queue sizes for this model. Let $p_{0,n}$ be the steady state probability of having n I/O requests in the drum queue and having some request in the first stage, i.e., the drum is latent. When the drum reaches the first starting address of some customer in the queue, it begins data transmission by passing to the second stage. Denote the steady state probability of transmission with n requests at the drum by $p_{1,n}$. For convenience let $p_{0,0}$ denote an idle drum and let $p_{1,0}$ be identically zero. The balance equations are then:

$$\begin{aligned}
&\lambda p_{0,0} = \mu p_{1,1} \\
&(\lambda+\mu)p_{1,n} = \frac{(n+1)}{\tau} p_{0,n} + \lambda p_{1,n-1}, \qquad 1 \le n < m \\
&(\lambda+\frac{n+1}{\tau})p_{0,n} = \mu p_{1,n+1} + \lambda p_{0,n-1}, \qquad 1 \le n < m \qquad (6.1)\\
&\mu p_{1,m} = \frac{(m+1)}{\tau} p_{0,m} + \lambda p_{1,m-1} \\
&(\frac{m+1}{\tau})p_{0,m} = \lambda p_{0,m-1} .
\end{aligned}$$

These equations can be transformed to the following form

$$\begin{aligned}
&p_{1,n} = \rho_0(p_{0,n-1} + p_{1,n-1}), \qquad 1 \le n \le m \\
&p_{0,n} = \rho_n(p_{0,n-1} + p_{1,n}), \qquad 1 \le n < m \qquad (6.2)\\
&p_{0,m} = \rho_m \, p_{0,m-1}
\end{aligned}$$

where $\rho_o = \lambda/u$ and $\rho_n = \lambda\tau/(n+1)$. Remembering that $p_{1,0} = 0$, we have 2m+1 equations in 2m+2 variables. We get the final (nonhomogeneous) equation by recalling that all the variables must sum to unity in order to represent a discrete probability distribution. Now note that if a value for $p_{0,0}$ were known, the values of all the other variables could be computed directly from Eqs. (6.2). Note also that $p_{0,0}$ is a factor of all the other variables; that is, if $p_{0,0}$ were incorrect by a factor of α, all the other values computed from Eqs. (6.2) would be incorrect by a factor of α. Thus to find the correct value for $p_{0,0}$ assume some arbitrary, nonzero value for $p_{0,0}$, say one, and use Eqs. (6.2) to compute the sum of all the variables. The reciprocal of the resulting sum is the factor by which the initial value of $p_{0,0}$ was incorrect. Thus if we assumed an initial value of one, the correct value would be the reciprocal of the sum. Now we can either correct the computed values of the other variables or recompute them using the correct value of $p_{0,0}$.

The expected waiting time at the two-stage drum, $\overline{W}$, (the queueing time plus the service time) will be

$$\overline{W} = \sum_{n=0}^{m} n(p_{0,n} + p_{1,n}) .$$

The utilization of the CPU will be

$$u_c = 1 - (p_{0,m} + p_{1,m}) .$$

<u>The one-stage cyclic model.</u> Figure 6.2 shows a cyclic queue model using the one-stage Markov model of an SLTF file drum. There is one exponential server for the drum, the service rate of which is queue size dependent. Let μ_n be the service rate when n customers are in the drum queue. Since we want the mean service time to be $\tau/(n+1) + 1/\mu$, where $1/\mu$ is the mean transmission time, we see that

$$\mu_n = \frac{(n+1)\mu}{n+1+\mu\tau} .$$

The balance equations for this model are

$$\lambda p_0 = \mu_1 p_1$$

$$(\lambda+\mu_n)p_n = \lambda p_{n-1} + \mu_{n+1} p_{n+1} , \qquad 1 \le n < m$$

$$\mu_m p_m = \lambda p_{m-1} .$$

These are equations for a simple queue with arrival and service rates dependent on queue size [cf. Cox and Smith, p. 43] so the solution is

$$p_n = \left(\prod_{i=1}^{n} \rho_i \right) p_0 , \qquad 1 \le n \le m$$

$$p_0 = \left\{ \sum_{i=0}^{m} \prod_{j=0}^{i} \rho_j \right\}^{-1}$$

where $\rho_n = \lambda/\mu_n$.

Comparison of the models. Figure 6.3 shows the expected waiting time, $\overline{W}$, for a drum service request versus the ratio of drum transmission time and computing time, λ/μ, when the expected record size is one-third of a drum revolution for three models and four different degrees of multiprogramming, $m = 2, 4, 8, 16$. The model results depicted by the square plotting symbol are from a modification of the simulation model discussed in the earlier sections. The drum server captures the true latency of an SLTF drum. A CPU server and a fixed number of customers have been substituted for the Poisson source. The model results depicted by the smooth curves are from the two-stage model. The one-stage model results are indicated by the curves with the plotting ● symbol. Both of the analytic models are very close to the simulation model except for large degrees of multiprogramming ($m > 8$), but the two-stage model gives slightly better results than the one-stage model.

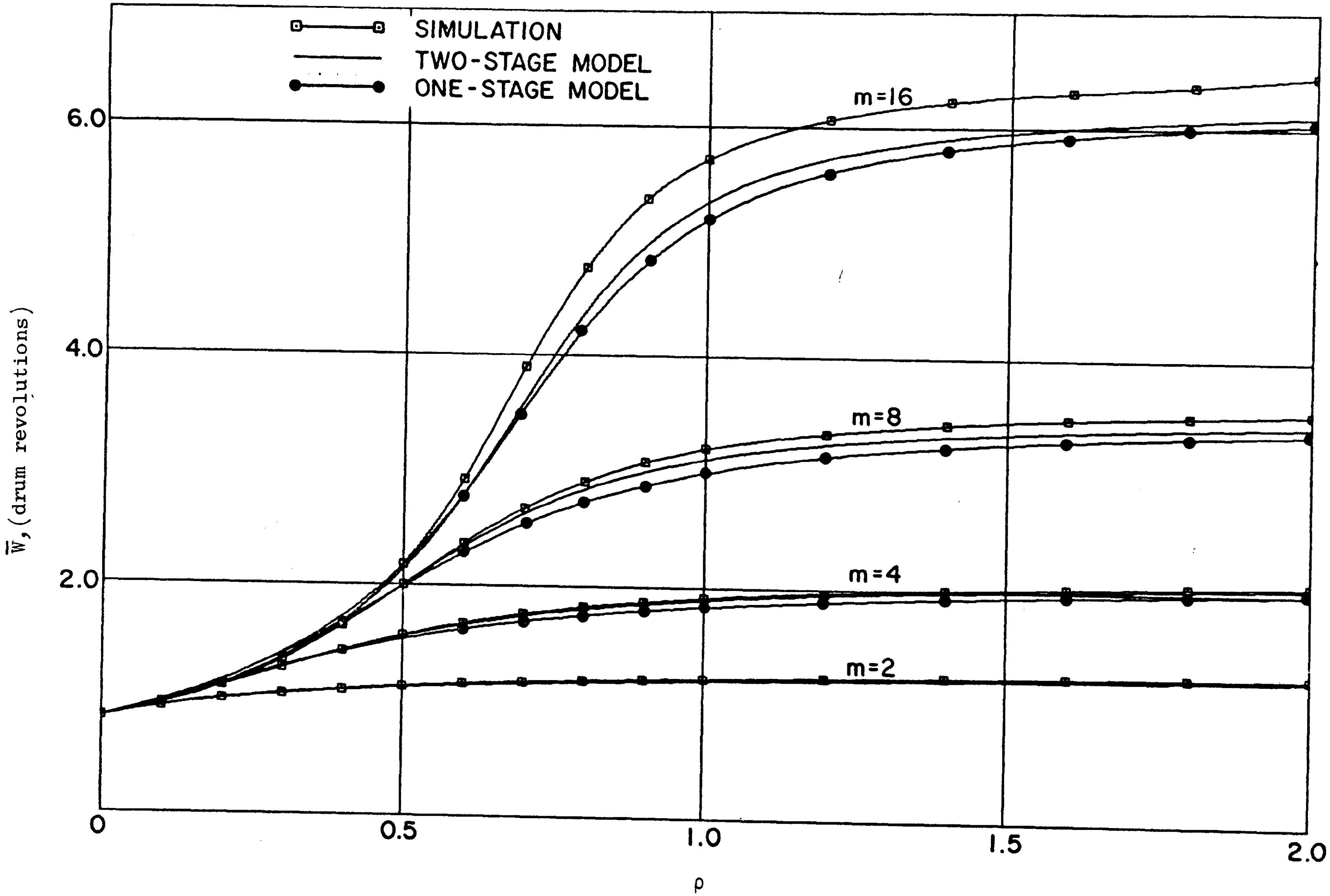


Figure 6.3. The expected waiting time of the three cyclic queue models of the CPU-drum system for $\overline{R} = 1/3$.

Figure 6.4 shows the CPU utilization for all of the cases shown in Fig. 6.3. The results from the analytic models seem even better for this normalized measure of system performance.

Figure 6.5 shows the expected waiting time for the three models and four degrees of multiprogramming versus the ratio of transmission and computing when the expected record length is one-eighth of a drum revolution. The problems with the approximations of the analytic models are becoming more apparent in this figure. The analytic models overestimate the expected waiting time except for large values of λ/μ and large degrees of multiprogramming. Again, in Fig. 6.6, the CPU utilization for the same cases as Fig. 6.5, the models still give very good results.

Comparison of Figs. 6.3 and 6.5 with Figs. 4.1 and 4.2 show that the expected waiting time for the cyclic models follows the expected waiting time for the Poisson source models until the CPU utilization falls away from 100%. Then the expected waiting times flatten out and approach the asymptote determined by having all m customers in the drum queue at all times. Comparison of Fig. 6.4 and Fig. 6.6 show that for a given ratio of transmission and computing, large records are to be preferred over short records. For a given quantity of work less latency will be incurred if records are large since fewer records will be transmitted. However, it is expected that the penalty associated with short records will be less severe in an SLTF system than in a FIFO system since the total incurred latency is reduced by SLTF scheduling.

The degree of agreement among the curves for the different models encourages us to use the analytic models of SLTF file drums in more complex models of computer systems at least until a more exact treatment

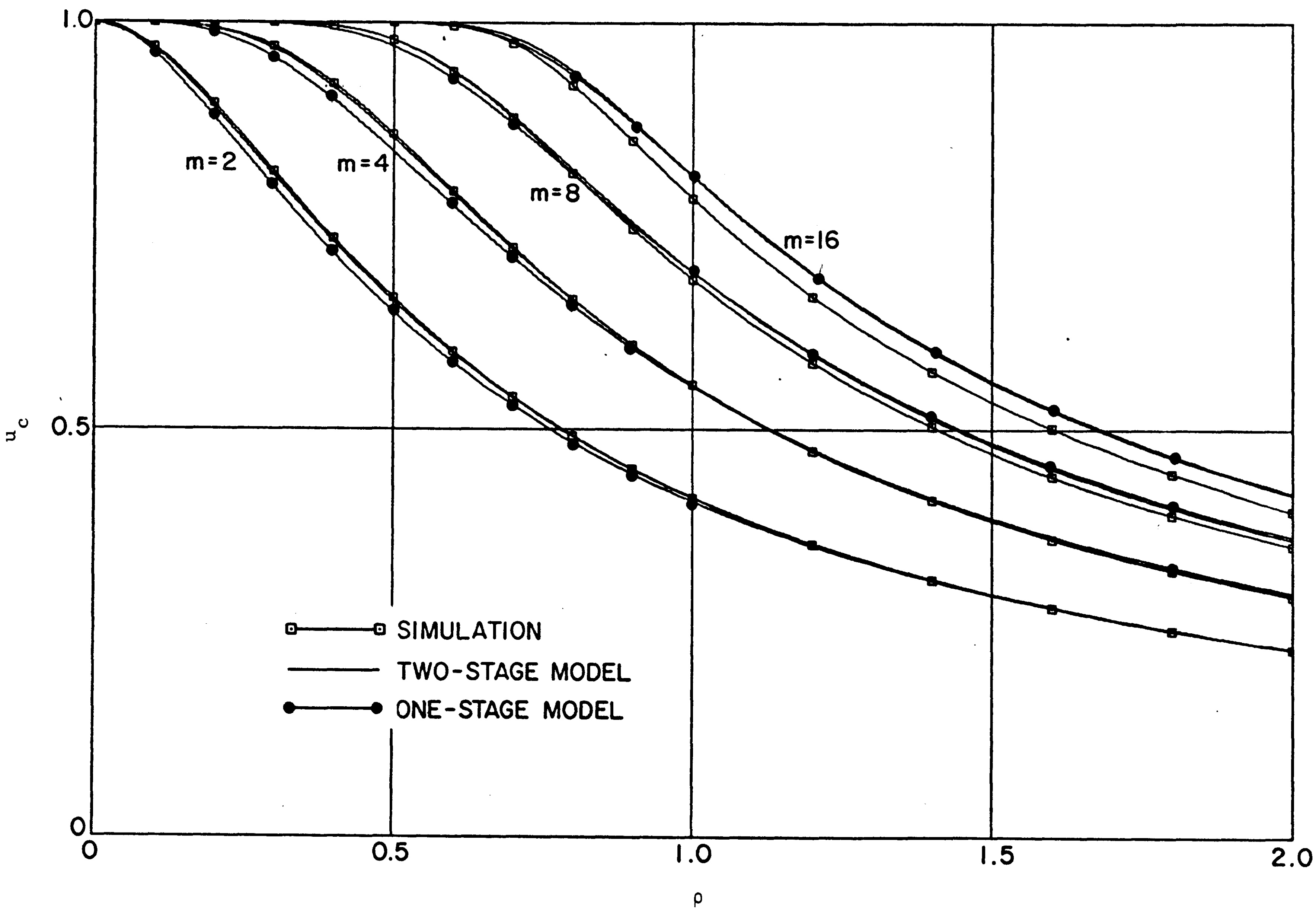


Figure 6.4. Central processor utilization of the three cyclic queue models of the CPU-drum system for $\overline{R} = 1/3$.

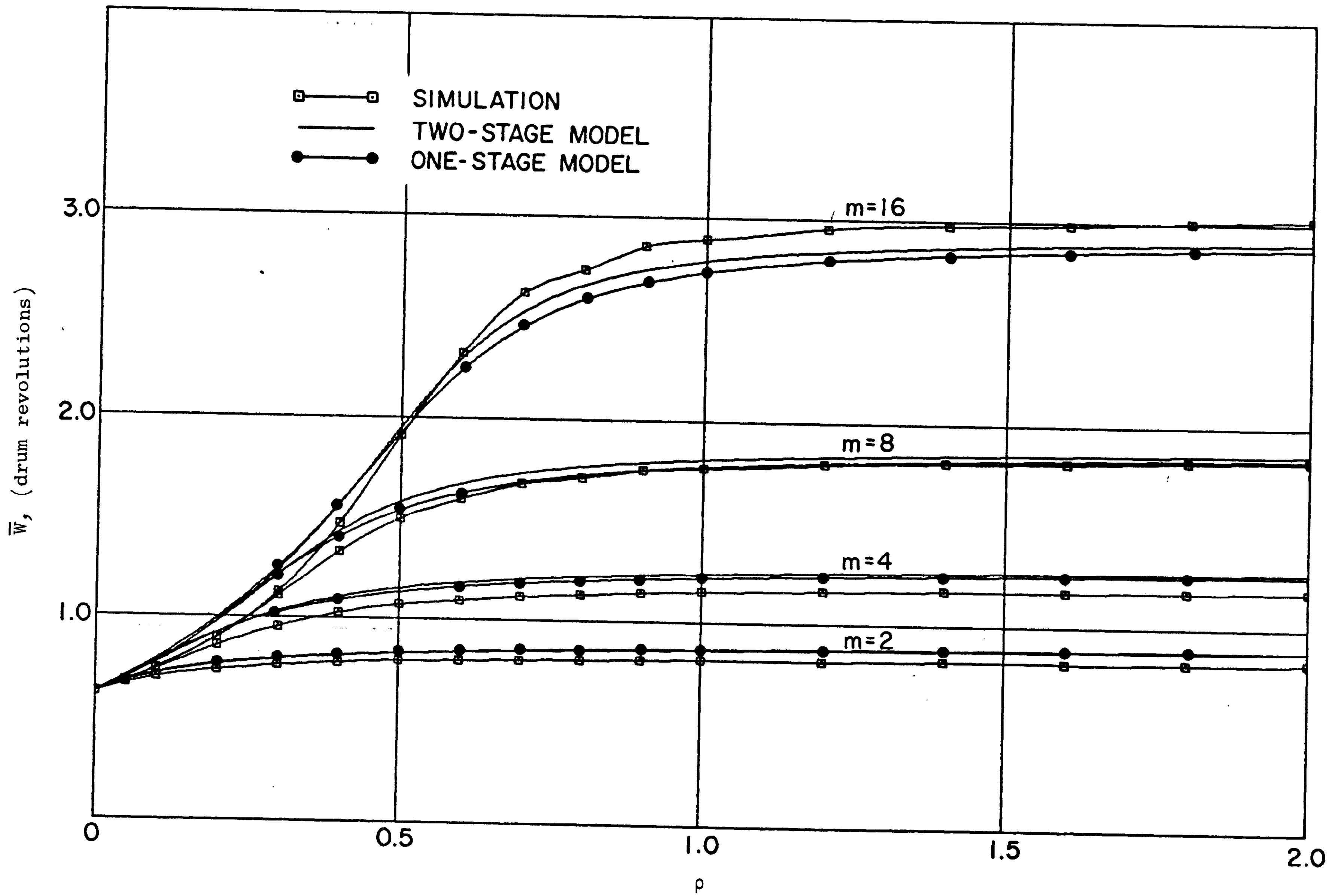


Figure 6.5. The expected waiting time of the three cyclic queue models of the CPU-drum system for $\overline{R}$ = 1/8.

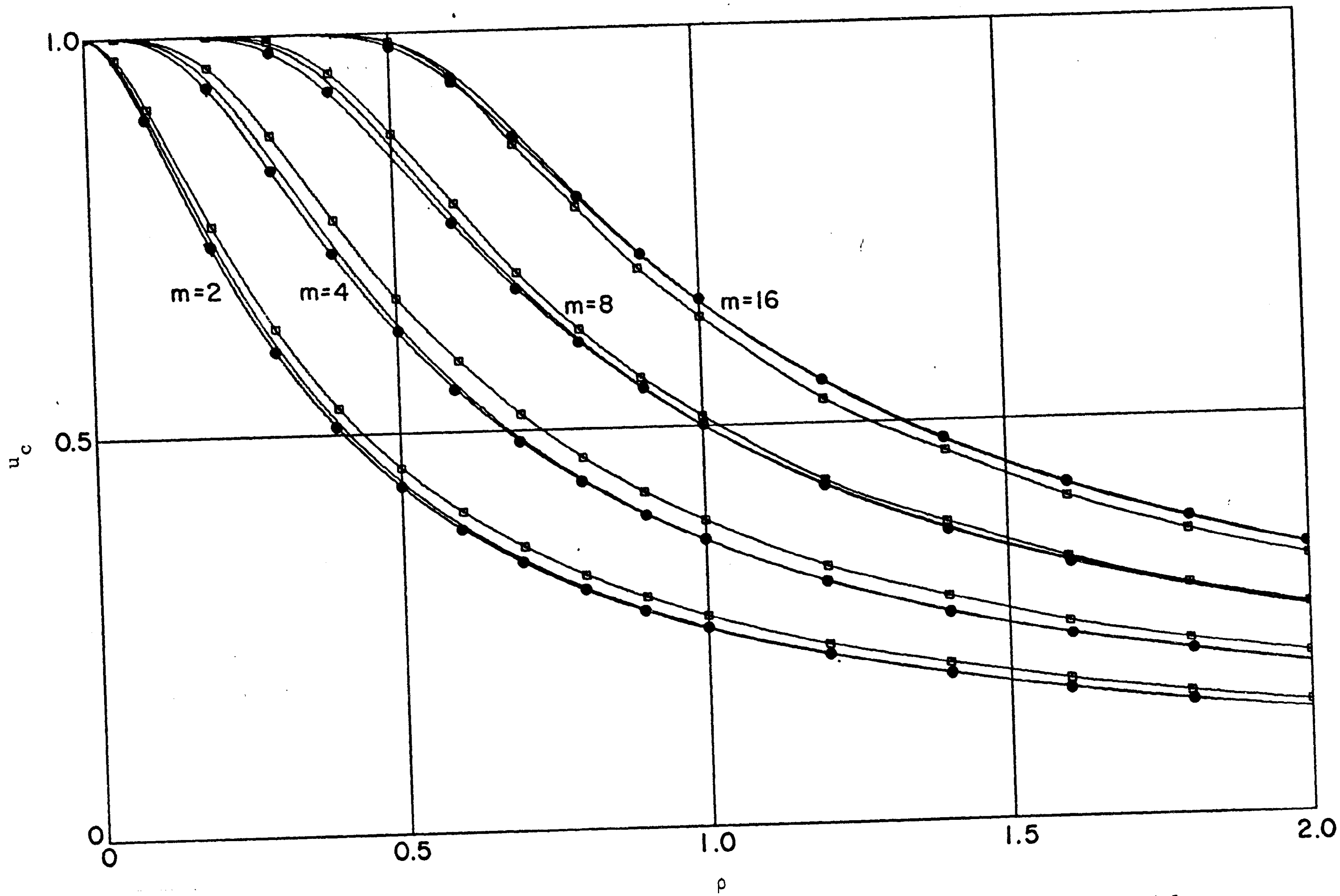


Figure 6.6. The central processor utilization of the three cyclic queue models of the CPU-drum system for $\overline{R} = 1/8$.

of SLTF scheduling can be developed. Even if an exact analysis were available, that analysis might not be compatible with the analysis used to treat a larger model in which an exact model was imbedded. Thus, for example, the one-stage model may be useful for some time because it fits naturally into queueing network models of computer systems.

7. Conclusions

We have analyzed two types of drum-like storage organizations, file drums and paging drums, and two types of scheduling, first-in-first-out and shortest-latency-time-first. For the FIFO file drum with Poisson arrivals we observed that the Pollaczek-Khinchine formula gives exact results for the expected waiting time of a request to the drum. For a FIFO paging drum, a FIFO sectored file drum, and an SLTF paging drum, all with Poisson arrivals, different interpretations of Skinner's model yield exact expressions for the expected waiting times. For the SLTF file drum with Poisson arrivals, two new approximate models are developed and an earlier approximate model is discussed and all analytic results are compared with results from a simulation model. The weak points of the approximate models are identified and the reasons for the errors are discussed. Table 7.1 shows the expressions for $\overline{W}$, the expected waiting time of I/O requests for the four drums discussed in this paper. Note that all the expressions are hyperbolic in form, with vertical asymptotes at ξ on ρ, except for the SLTF file drum. In Sec. 5 we showed the expected waiting time for I/O requests at the SLTF file drum grows faster than hyperbolically as $\rho \to 1$. Figure 7.1 graphically illustrates the relative performance of the different drum organizations and scheduling disciplines. (The two-stage Markov model is used for the

Table 7.1. Expressions for the Expected Waiting Times of I/O Requests at a drum with Poisson arrivals.

scheduling discipline	drum organization	$\overline{W}$
FIFO	file	$(\frac{1}{2} + \overline{R})\left[1 + \frac{\xi(1+C^2)}{2(1-\xi)}\right]\tau$
FIFO	paging	$\{\frac{1}{2} + \overline{R} + \frac{\xi(1 + \frac{1}{2k})}{3(1-\xi)}\}\tau$
SLTF	file	two-stage Markov model $\frac{\mu\tau+1}{(\mu-\lambda)\tau} + \frac{1}{\lambda\tau}\left[\left\{\int_0^1 \left[e^{\lambda\tau(1-w)}(1-\rho w)\right]^{\lambda\tau} dw\right\}^{-1} -1\right]$ empirical model $\{\frac{1}{2} + \overline{R} + \frac{\rho}{1-\rho} + .368(\frac{\rho}{1-\rho})^{3/2}\}\tau$
SLTF	paging	$\frac{1}{2} + \overline{R} + \frac{\rho}{2(1-\rho)}$

$$\rho = \lambda\overline{R}\tau; \quad \xi = \lambda(\tfrac{1}{2} + \overline{R})\tau = \frac{\lambda\tau}{2} + \rho .$$

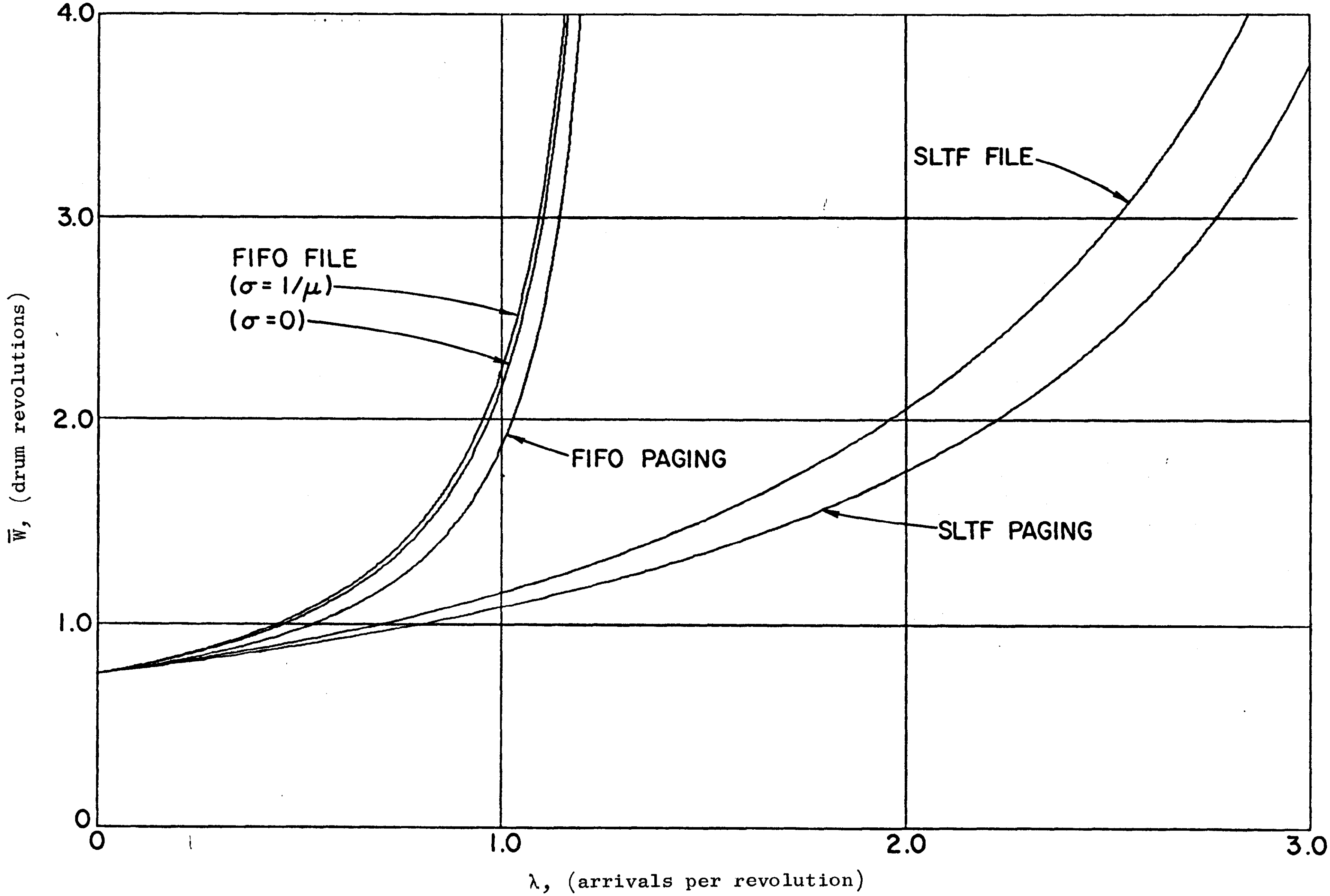


Figure 7.1. The expected waiting time for the different drum organizations and scheduling disciplines where $\overline{R} = 1/k = 1/4$.

SLTF file drum).

The one-stage and two-stage Markov models are incorporated into a cyclic queueing model and these results are compared with simulation results. The comparisons indicate the suitability of these models of SLTF file drums for use in more complex queueing network models of computer systems. A reasonably accurate model of an SLTF paging drum that could be easily incorporated into larger queueing network models would be a valuable addition to this work.

REFERENCES

Abate, J. and Dubner, H. (1969) Optimizing the performance of a drum-like storage. IEEE Trans. on Computers C-18, 11 (Nov., 1969), 992-996.

Abramowitz, M. and Stegun, I. A. (1964) Handbook of Mathematical Functions. National Bureau of Standards, Applied Mathematics Series 55 (June, 1964).

Burroughs. (1970) Burroughs B6375 disk file optimizer. New Product Announcement, Burroughs Corp., Detroit, Mich. (Feb., 1970).

Coffman, E. G. (1969) Analysis of a drum input/output queue under scheduling operation in a paged computer system. J. ACM 16, 1 (Jan., 1969), 73-90.

Cox, D. R. and Smith, W. L. (1961) Queues. Monographs on Applied Probability and Statistics, Chapman and Hall Ltd., London. (1961).

Denning, P. J. (1967) Effects of scheduling on file memory operations. Proc. AFIPS SJCC 30 (1967), 9-21.
Keywords: drum scheduling, queueing models.

Feller, W. (1968) An Introduction to Probability Theory and Its Applications, Volume 1. Wiley and Sons, Inc., New York, N.Y. (1968).

Feller, W. (1970) An Introduction to Probability Theory and Its Applications, Volume 2. Wiley and Sons, Inc., New York, N.Y. (1970).

Fuller, S. H. (1971) An optimal drum scheduling algorithm. Technical Report No. 12, Digital Systems Laboratory, Stanford University, Stanford, Calif. (April, 1971).

Fuller, S. H. (1972A) The expected difference between the SLTF and MTPT drum scheduling disciplines. Technical Report No. 28, Digital Systems Laboratory, Stanford University, Stanford, Calif. (August, 1971).

Fuller, S. H. (1972B) A simulator of computer systems with storage units having rotational delays. Technical Note 17, Digital Systems Laboratory, Stanford University, Stanford, Calif. (August, 1972).

Fuller, S. H., Price, T. G., and Wilhelm, N. C. (1971) Measurement and Analysis of a multiprogrammed computer system. IEEE Workshop on Measurement and Performance Evaluation, Argonne, Ill. (Oct. 1971).

Gaver, D. P. (1967) Probability models for multiprogramming computer systems. J. ACM 14 (July, 1967), 423-438.

Gordon, W. J. and Newell, G. F. (1967) Closed queueing systems with exponential servers. Operations Research 15 (1967), 254-265.

IBM. (1971) IBM 2835 storage control and IBM 2305 fixed head storage module. Form GA26-1589-2.

Jackson, J. F. (1963) Jobshop-like queueing systems. Management Science 16, 1 (Oct. 1963), 131-142.

Jewell, W. S. (1967) A simple proof of: $L = \lambda W$. Operations Research 15 (1967), 1109-1116.

Kimbleton, S. R. and Moore, C. G. III (1971) A limited resource approach to system performance evaluation. Technical Report No. 71-2, Department of Industrial Engineering, College of Engineering, The University of Michigan, Ann Arbor, Mich. (30 June, 1971).

Saaty, T. L. (1961) Element of Queueing Theory. McGraw-Hill, New York, N.Y. (1961).

Sherman, S., Baskett, F., and Browne, J. C. (1971) Trace driven modeling and analysis of CPU scheduling in a multiprogramming system. ACM Workshop on System Performance Evaluation, Harvard Univ. (April, 1971), 173-199.

Skinner, C. E. (1967) Priority queueing systems with server-walking time. Operations Research 15, 2 (1967), 278-285.