1972

The scheduling of computer disk operations

Richard R. Reisman
Lehigh University

THE SCHEDULING OF COMPUTER DISK OPERATIONS

by

Richard Roy Reisman

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1972

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 2, 1972
Date

Gary E Whitehouse
Professor in Charge

Chairman of the Department of
Industrial Engineering

ACKNOWLEDGEMENTS

Thanks are due to Professor G. E. Whitehouse of Lehigh and to R. A. Stone of Western Electric for their support of this study.

I am also indebted to the other researchers in this field who made their works available to me, and especially to D. L. Stone, T. J. Teorey, T. B. Pinkerton, and G. H. MacEwen for helpful discussions on the telephone and by mail.

Finally, I wish to give special thanks to my fiancee, Christine Zubrovich, for her understanding and assistance during the course of this effort.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

There is a growing interest in the application of scheduling techniques to the two mechanical operations which are bottlenecks in the use of movable-head disk storage devices. A detailed simulation model is used to explore the effect of these techniques for varying numbers of disk modules.

Primary emphasis is on comparison of the two approaches to latency reduction based on the use of rotational position sensing (RPS): software queueing versus hardware queueing. It is found that both of these techniques can be highly effective for disk systems with many modules operating under substantial loads. Both give similar gains in mean response time and peak throughput capacity, but the hardware queuer can also yield large reductions in channel utilization.

For the techniques which apply to the other mechanical operation - the scheduling of seeks - results are mixed. Used alone, seek scheduling is found to be ineffective on configurations with many modules, but good results are obtained when the two classes of scheduling are used in combination.

In addition to the comparison of gross performance measures, a variety of statistics are collected on the internal operation of the disk system. These observations give insights into the subtle balances of the queueing process which are not readily obtained except by simulation.

CHAPTER I

Introduction

Modern computer systems rely heavily on on-line auxiliary storage devices to provide rapid access to massive quantities of data. Before any of this data can be processed, however, it must be transferred to the smaller central memory. Rotating magnetic drums and disks are the predominant on-line storage devices because of their large capacity and low cost. But these devices require mechanical motion in order to effect the transfer of data and therefore cause large delays of the fast central processor. As a result, access to auxiliary storage becomes a serious bottleneck in computer operation.

In multiprogramming operation, these input/output delays do not generally cause the central processor to wait idle since there are other processes (programs) in the system which may be activated to make use of this time. But these processes may also require records from the same device so that a queueing situation develops. This build-up of queues adds to the delays, but it also provides an opportunity to increase the throughput of I/O requests by clever scheduling. This is possible because the total access time for a series of requests is a (known) function of the sequence in which these requests are served. A scheduling rule which orders transfer requests in such a way as to reduce mechanical access time (by taking advantage of this knowledge) can thus increase throughput. Such rules have been applied in some major operating systems.

Of particular interest is the movable-head disk storage facility. This device is in very wide use in a wide variety of systems. It is slow compared with drums and head-per-track disks, but lower in cost. Where access speed is not a stringent requirement, the disk is the dominant device for on-line storage. The importance of efficient operation for such a heavily used device is clear. The most common disk configuration consists of multiple modules and this factor impacts greatly on the scheduling problem. Such configurations are the focus of this study. The RP-10 disk facility used on the Digital Equipment Corporation PDP-10 computer is examined in particular as a reasonably typical example.

The goal, then, is to evaluate and to gain understanding of the effectiveness of the relevant scheduling techniques. The basis for this investigation is a highly detailed simulation model of a multiple disk facility. The objective is not merely to compare these techniques for the standard gross performance measures (mean response time and peak throughput capacity), but also to obtain a variety of details which will elucidate the internal operation of such systems. No detailed study of this sort is known to exist in the literature.

Primary emphasis is placed on the use of techniques for minimizing rotational delay. These are techniques for scheduling the actual transfer operation based on the use of special hardware for rotational position sensing (RPS). There are two principal approaches - one in which the scheduling logic is implemented in software and the other which relies on a hardware queuer. Both have been implemented for major systems such as the PDP-10 and the IBM System/370, respectively. The operation and effectiveness of these rules (as well as the conventional mode of operation) are examined in some detail.

A different approach to disk scheduling involves methods for minimizing delays which occur in the arm positioning operation. These techniques are also considered, both alone and in combination with the rotational methods.

While a number of studies exist in these areas, they suffer from various limitations in scope and method. This study seeks to extend the understanding of disk scheduling techniques both in breadth and in depth. It is also hoped that the present results are given in sufficient detail to aid in the formulation and validation of analytic models. Two successful applications of this type are described.

CHAPTER II

Background

This chapter presents the necessary background for this study in three stages. First is a basic outline of the structure and operation of multiple disk storage facilities. Next, the general concepts and specific techniques for scheduling these devices are presented. Finally, attention focuses on the problems of evaluating the effectiveness of these techniques. This includes a broad discussion of the difficulties of defining a meaningful model, as well as a review of existing work.

2.1 Principles of Disk Operation

From the viewpoint of the disk service subsystem of a computer, transfer requests are requests to read or write a block of data at a specified position on the recording surface. The disk operation is characterized by the series of two mechanical operations required to move the recording heads to this specified position prior to the actual transfer. The disk pack or module is the basic unit of the magnetic recording medium and is actually a stack of individual disks spaced on a single spindle (the disk drive). Recording heads are mounted on movable arms between these disks, with one read/write pair per surface. Thus the recording space is contained in a cylinder and is conveniently described by cylindrical coordinates: radius, angle, and depth. All transfer requests are given in terms of these coordinates.

Depth is of secondary significance, since the heads corresponding to the desired surface may be switched at electronic speeds. The other two coordinates correspond to the two mechanical operations. Access at the proper radial position entails moving the head, much like cueing a record. Note that heads (one per surface) are ganged on a single comb-like mechanism and move as a unit. Thus, a cylindrical recording plane is accessible from each arm position--one circular track on each surface. This means that any angular position or depth in this cylinder can be reached without arm motion. These arm positions are commonly specified by cylinder number. Finally, to transfer a block, it is necessary to wait until it rotates into position under the head. Unlike the other operations, angular positioning is passive--no positive action is required, merely waiting for the continual rotation. Angular position is commonly specified by sector number where a finite number of equi-angular sectors are defined to span the disk surface. The delays involved, then, are that of arm motion (positioning or seek time) and that of rotational delay (latency). The actual transfer also takes time, of course, but there is no control over this delay.

The service times for these operations depend on the disparity between the current and desired position. Seek time depends only on the number of cylinders (radial distance) to be traversed. So the time for each request depends on the placement of the previous request. Latency depends only on the number of sectors (angular distance) separating the record requested from the current position of the heads. So latency varies with start time for the operation and is otherwise independent of previous requests. Seek times for the PDP-10/RP-10 disk are typical--increasing in a nearly linear fashion from a minimum of 20 ms. from one cylinder to the next to a maximum of 80 ms. for a full traverse. Latency for this model increases linearly to a maximum of 25 ms. for a full rotation.

Because the logic required to control a disk drive is expensive, it is usually shared by a number of individual drives. This controller together with its drives (commonly four or eight) comprise the disk facility. The controller is connected to the mainframe by a channel through which it receives all requests for its drives.

Sharing the controller (and its channel) naturally imposes limitations on the use of the disks. The most critical is that the controller can handle only one transfer operation at a time. This can create a bottleneck since there may be many drives with requests which are on-cylinder and ready to transfer at any given time. Such a transfer entails more than just the time to read a block. Since the desired block (or sector) is not, in general, directly under the heads when the controller is ordered to do the transfer, the controller must search the track until that block is encountered. This search time corresponds to the latency at the time of initiating the transfer. So the busy time for the transfer includes the time for the search as well as the time for the actual transfer.

The other limitation is that the controller is required to initiate all arm positioning operations. Once it has been started, the seek operation can be performed by the drive alone. When the proper cylinder has been reached, the drive signals completion to the controller. During this period the controller remains free to do other work until it chooses to respond to that signal. Thus any number of drives may be seeking simultaneously. Since seek initiation is done at electronic speeds, the load on the controller for this task is negligible. The limitation is that incoming positioning requests are delayed if the controller is busy.

Because of this interaction, it is important to coordinate the handling of seeks and transfers for effective operation. On completion of a transfer the controller will be given new work in the following sequence. First, seeks will be initiated on any free drive which has been requested. Only then are any new transfers started. This is quite sensible since the seeks can be initiated with little chance of delaying the transfer, while the transfer would otherwise delay the seeks significantly.

In addition to the sharing of the controller, it is not uncommon for more than one controller, each with a set of drives (or other devices), to share a single channel. This, again, is based on economic considerations--channels are expensive. Another limitation is thus imposed on disk services since the channel can only transmit one record at a time. Since this complication is of secondary interest, this study will be limited primarily to the case of a dedicated channel and the terms "channel" and "controller" will generally be used to mean both.

So the basic structure of the queueing system is of two types of servers which combine to perform each of two types of service. The servers are the controller (one per facility) and the drives (several in parallel). The services are seeks (which are initiated by the controller and the drive in concert, but completed by the drive alone) and transfers (which are performed by both servers in concert). Note that the drive is monopolized during the entire period of service for a request on that drive - seek, transfer queueing, and transfer - even though it is not active during the transfer queueing period.

This discussion has considered only the simple case of independent requests to transfer single blocks. Operation is complicated by requests to transfer multiple consecutive blocks and other kinds of combined operations (such as write-verify) which may occur in practice. These complications have usually been ignored in previous studies of disk scheduling and will not be considered here. The above discussion is also limited in that some movable-head disks do not operate quite as described here. They are atypical and will be ignored.

2.2 Scheduling Techniques

Before describing the techniques for scheduling disk operations, it should be emphasized that scheduling of requests is by no means the only approach to reducing auxiliary storage delays. In fact the traditional and dominant approach has been by way of file organization, which operates in a manner complementary to that of scheduling. The file organization approach is based on prediction of the patterns of requests which will be made of the disk. Using these predictions, the content and placement of blocks are controlled when they are first written in such a way that the anticipated processing sequences will require a minimum of mechanical motion. These techniques have evolved from a uniprogramming environment where prediction was straightforward. The unpredictable interweaving of many requesting processes that occurs in multiprogramming has profoundly reduced the power of this method.

Conversely, the scheduling approach does not attempt to control or predict the behavior of the requesting processes. It takes this as given and relies on knowledge of the present contents of the request queues to adjust the operation sequence within the operating system. So, unlike the file organization approach, the effectiveness of scheduling is enhanced by the queueing inherent in multiprogrammed operation and is insensitive to the lack of predictability.

Also of note is the attempt to reduce the number of mechanical operations by transfering batches of many blocks at once to and from an intermediate level of storage from which they can be accessed at high speed. The problem of how best to combine all of these methods for a given system is interesting and important, but must remain beyond the scope of this study.

The rest of this section contains a description of the scheduling techniques to be explored. All of these methods are at least mentioned in the existing literature. Following this introduction, the relevant efforts in evaluation of disk performance are reviewed.

In keeping with the usual manner of operation, the disk system to be scheduled is considered as two separate queueing systems which are scheduled independently. One queue (actually one for each drive) contains requests which await positioning of the arm to the proper cylinder. The other (for all drives on a control) contains requests which have been positioned and await transfer. This simplification can easily be justified on two grounds. Any rule which considered both operations would have difficulty meeting the stringent speed limitations. Also, newly arriving requests with short seek times would conflict with the planned transfer sequence so that even theoretical gains are questionable. So this distinction will be maintained here. The two problems, then, are:

1. Scheduling (for each drive) of the requests which await arm positioning on that drive, and
2. Scheduling (for the entire facility) of requests which have been positioned and await transfer.

This gives a two stage queue structure of the following form. The first stage consists of parallel queues for positioning, one for each drive. The output of that stage (on-cylinder requests) then funnels into a single queue for transfer, which is the second stage. Note that the number of requests in the transfer queue is effectively limited since there is usually only one request on-cylinder per drive.

Rotational Optimization

Major emphasis will be placed on techniques for minimizing the latency involved in disk transfers, for two reasons. First, it appears to be the more promising approach for multiple disk facilities, and second, it has been less explored.

The basic approach to latency reduction is simply to apply the standard SOT (shortest operation time) rule. Scheduling occurs when the controller has just completed a transfer and is free to initiate any other requests that are queued. If the number of the sector currently under the heads and of the sector containing the desired block are both known, it is easy to compute the latency for that operation. All one need do is search the queue for the request with the shortest latency and initiate it.

It is known that SOT gives good performance in terms of mean flow time [Conway, Maxwell, and Miller, 1967], and because of the nature of disk operation, it also gives increases in throughput. This arises from the fact that operation times (latencies) are sequence (time) dependent. By choosing requests which are nearby, those which are presently distant tend to be held in the queue until they happen to be nearby. Thus the average latency is reduced and more transfers can be processed in a given time. The rotational nature of disk operation impacts further on SOT scheduling. The usual flaw of the SOT rule is that requests with long operation times suffer excessive delay. But because latency is a cyclic function of time, a request with low priority at one time may have high priority the next.

This technique was initially applied to drum storage, which requires no seeking and is limited only by rotational delays. It has been shown to produce dramatic gains in throughput on that device (as much as twenty to one) [Denning, 1967]. The potential for disks is of course far less spectacular. In addition, the effect of latency reduction is a function of queue length and queue length is limited by the number of drives active on a given controller. So the potential of scheduling is further limited for small configurations.

The major practical difficulty in latency optimization is that of knowing the current and desired positions. In order to be able to sense the current rotational position, a drive must have special hardware. Only some of the more recent drives (such as the IBM 3330 and the DEC RP-10) include this hardware--the widely used IBM 2314 does not. In addition it is not otherwise required to know on what sector a requested block will be found and this information is not generally maintained in existing systems (although to do so is not difficult). So it must be cautioned that these methods are not immediately applicable for many installations.

Looking in greater detail, there are two distinct methods of latency minimization which are applicable to movable-head disks. The operation of these methods is closely tied to the availability of special hardware. One requires hardware for rotational position sensing (RPS) but relies on software for the queueing of requests. The other makes use of special hardware features for both functions. For convenience, the former will be referred to as the software approach (S-RPS) and the latter as the hardware approach (H-RPS). Both approaches have been implemented in recent major systems: software in the DEC PDP-10 (RP-10 disk) [Stone, 1970; DEC, 1971], and hardware in the IBM System/370 (3330 disk) [IBM, 1970, 1971a]. The exact form of the algorithms used in this study will be based on the documentation of these systems.*

* With one exception: the DEC PDP-10 algorithm permits alternation of SOT with occasional use of FCFS (normally once in ten times). Such alternation will not be considered here.

The software technique is quite straightforward in operation. On completion of a transfer the controller is free and a request is scheduled. The software obtains the current sector number and selects the nearest request from the queue (which is maintained in software). It then orders the controller to begin a transfer for that request. The controller handles the transfer just as it would in conventional first-come-first served (FCFS) operation--it searches until the block is found and performs the actual transfer, and is busy for this entire period (which is the sum of latency and the transfer time).

The hardware scheduling process is more involved. Its operation is based on the fact that since the time a block will be under the heads can be predicted, there is no need for the controller to be busy searching during the latency period. The special hardware in the controller allows it to disconnect from the module during this delay and to later reconnect in time to perform the actual transfer. Since this hardware can keep track of multiple requests (normally one on each drive), more than one transfer can be initiated at a time. Conflicts in the actual transfer--which still requires the exclusive service of the controller and channel--are handled by permitting reconnection only if the controller is not busy with another data transfer. When a reconnection attempt fails the request remains in the system and reconnection is attempted on each subsequent revolution. So the disk software need not make any selection among requests. All new on-cylinder requests are presented to the controller whenever it becomes idle. Thus, the scheduling is performed within the controller and is implicit in the mode of operation described above. By nature of this mode of operation the nearest request will be the next to be transfered.

This approach has an added attraction from the viewpoint of scheduling theory. The effect of the hardware queuer is to delay the final scheduling decision until the last possible moment. Since the queue size is limited (in both cases), even the nearest request will often be a significant distance away. If a new and less distant request arrives at such a time, the hardware will be able to take advantage of the opportunity, but the software will not--its controller does not permit preemption.

A more practical distinction concerns the use of the channel by the two methods. Because H-RPS involves disconnecting, the channel is freed during rotational delays. Thus the total busy time of the channel can be reduced. In an environment where it is attractive to share the channel with other devices, reduction of the load imposed by the disk facility could be quite important. It will be demonstrated later that this effect can be significant.

There is one complication in the operation of both methods which has not yet been mentioned. Because of imprecision in the RPS equipment and the necessary delays of hardware operations, the reading of current sector position cannot be taken to be exact.

If a transfer (S-RPS) or reconnection (H-RPS) began too late to find the start of the block, the controller and channel would be tied up in that search for an entire revolution. To avoid this risk of large delays, both methods allow a margin of safety in their scheduling procedure. For the PDP-10/RP-10 this safety factor is one sector (out of ten) [Stone, 1970]. So latencies for scheduling are computed as if the heads were some distance ahead of the indicated position. This portion of the latency can not be reduced by scheduling.

Stepping back to look at both stages of disk operation, a secondary benefit of latency reduction can be seen. Recall that seeks can be initiated only when the controller is free, and so may be delayed if a transfer is in progress. If latency is reduced, this delay will also be reduced. And here, also, the disconnection feature of H-RPS has an impact. These effects, however, are minor since the delays arise only when a request is made of an idle drive. Thus, the effect disappears as the load approaches saturation.

There are other approaches to latency reduction which have been suggested for disks by Weingarten [1968] and Sharma [1968]. These are oriented to message switching operations which are special in that they normally operate at very high request rates. These techniques will not be considered here.

Positioning Optimization

Techniques for seek optimization are more numerous but can be reduced to two basic concepts. One family of rules attempts to optimize the choice of the next request without regard to subsequent operations. The other rules operate in a manner intended to reduce total seek time for an entire group of requests. All of these techniques can be implemented entirely in software.

Again SOT is the obvious approach, taking the form of the shortest-seek-time-first (SSTF) rule. It is simple for the operating system to compare the current cylinder number with that required by each request and select that which is nearest. Again, the nature of the disk operation enables scheduling to improve throughput as well as mean flow time. In the case of seeks, however, the problem of excessive delay for some requests remains. A request is processed only if it is the nearest to the arm, and if new requests happen to be arriving near the arm, those which are distant remain at a disadvantage.

A simple way to avoid this problem is to alternate this rule with FCFS. This idea (which has received little note in the literature) is implemented in the DEC PDP-10 such that every n-th time the scheduler selects the oldest request instead of the nearest [DEC, 1971]. This limits the delay of any one request (at some--perhaps negligible--cost in throughput).

Motivated by this problem of discrimination, Denning [1967] proposes the SCAN rule. This rule orders all requests in the queue by cylinder position and processes them in that order, effectively sweeping back and forth across the entire surface of the disk. The direction of the arm motion is reversed only when there are no more requests in the current direction. As originally proposed, requests are handled during sweeps in either direction. This causes requests for either extreme to wait as much as two sweeps (away and back). Manocha [1969] considers a modification (circular SCAN) in which requests are processed in only one direction, each scan ending with a long seek back to the starting region. This rule has been implemented by IBM as an option under OS/360 [IBM, 1971b]. Frank [1969] suggests a version of SCAN which batches requests and satisfies all of each batch before considering new requests.

Comparison of these rules will not be considered here since that problem has recently been studied by Teorey and Pinkerton [1971] for the case of a single drive. The comparative performance of these rules should not be altered in a multiple drive situation. What is of interest is the effect of multiple drive operation on the effectiveness (relative to FCFS) of any such rule. Effectiveness may be diminished for two reasons: 1) queue length per drive may be relatively short, and 2) overall performance may be insensitive to improvements in seek performance. What is also of interest is the potential effectiveness of any such rule when used in combination with latency reduction techniques.

For this purpose, examination of a single rule is sufficient. Since the evaluation will be in terms of throughput, the rule which gives the best throughput will be used to set an upper bound on the potential gains. So this study will consider only the simple SSTF rule in spite of its possible practical disadvantages.

The techniques to be examined in this thesis, then, are three: the two methods of latency reduction described above and the one method for seek scheduling singled out here. Conventional FCFS operation is of course used as a standard of comparison. Existing work relevant to this effort is reviewed in the following section.

2.3 Performance Evaluation

General Observations

The growth of interest in scheduling disk operations is clearly reflected in the literature on disk performance evaluation. A number of comparative studies have been performed quite recently, and some have not yet been published. The performance of arm scheduling techniques has been treated most extensively, but there are some studies of rotational techniques. Before describing the findings of these comparisons, the methods and problems common to all of these studies will be outlined. Consideration is limited to theoretical methods - those which are based on mathematical models of disk operation. While empirical methods offer certain attractions, they do not figure in the existing literature (except to validate models) nor are they applied in this study.

Most of the studies are based on analytical methods, but some rely on simulation. In either case, the structure of the problem and the methods of solution are such that different studies vary widely in

1) operating environment modeled
2) simplifying assumptions and approximations
3) solution technique
4) performance measures obtained.

This diversity poses obvious problems to the attempt to make use of published results for the comparison of different scheduling techniques. Since each technique must be modeled as a separate case, two consistent analyses are required for comparison. While all of the major rules have been analyzed in the literature, the results of different authors are not easily compared. Existing results for comparison of scheduling rules are based on new analyses which were performed with that comparison in mind.

Since the simulation approach used in this thesis is largely independent of previous studies, no comprehensive review of disk performance analyses will be attempted here. This section will concentrate, instead, on the results of comparative studies. Methodology will be considered as it relates to the interpretation of these results. To complement this presentation, a bibliography of all papers found to be relevant to the evaluation of disk performance is provided. For a survey of some of the major analytic efforts, the thesis by MacEwen [1971a] is recommended.

The one aspect of disk performance evaluation which is most fundamental and least satisfactory is the model of the system load. The occurrence of I/O requests for a disk system may be modeled as a stochastic process defined by the probability distributions for the following characteristics:

1) arrival time

2) location

    a) module

    b) cylinder

    c) sector

3) block size.

For real systems in general, these distributions may be complex and not at all independent. There is considerable ignorance in this area because of the lack of any thorough empirical studies of I/O loads (such as have been done for the similar problem of time-sharing system loads). This is compounded by the wide variation among different types of systems. A number of simplifying assumptions are generally made with some argument for their realism.

To begin with, the distributions are assumed independent, for obvious reasons. A Poisson arrival process is commonly assumed. This seems reasonable for some systems, particularly those with many active processes with a good balance of I/O and computation. For systems with few active processes, a finite universe ("machine repair") model is more realistic. An extreme case of this model assumes a constant number of requests in the system, which corresponds to a system dominated by a few I/O-bound processes. Some studies have used this model, assuming a constant request load on each drive. This permits estimation of performance under saturation loading. The simple Poisson assumption is also questionable if a system is subject to bursts of I/O activity. In such a case, the saturation model might be realistic during the burst periods.

Uniform distribution of requests throughout the recording space is commonly assumed. This is not realistic for many systems, since the blocks for a given process are usually grouped together in conventional batch-oriented systems. It may be somewhat more realistic for time-sharing and real-time systems. Some studies [Fife and Smith, 1965; MacEwen, 1971a] have attempted more realism.

There is also variation in the block length distributions used. Many systems permit block sizes to vary up to a full track. These have been modeled as uniform [MacEwen, 1971a] or exponential [Teorey, 1971] distributions. Others, such as paged systems or time-sharing systems like the PDP-10, require a fixed block size. This is easily modeled.

It is clear that these questions of realism are a serious problem. Nevertheless, a comparison of scheduling rules under such assumptions is better than none at all. And realistic models are not without their drawbacks. McAulay [1970] uses CSS/360, an

OS/360 oriented simulator, to model the I/O activity of typical user jobstreams. The very complexity of this model makes it difficult to interpret and generalize the results. Perhaps a two pronged approach is required: simple models to gain understanding of the processes and for rough estimation, and detailed (simulation) models for accurate and realistic prediction.

In considering the effect of departures from the assumptions of the idealized models, note that the two classes of scheduling rules are sensitive to different aspects of the load. The seek rules operate independently on the individual drives; so it is access rate per drive, along with cylinder distribution by drive, which are the major factors. Transfer scheduling, on the other hand, works on a facility basis, normally with requests on each of several drives. Thus it is access rate across drives, along with sector distributions across drives, which are most important. Since little correlation of sector positions across drives would be expected in most systems, this factor is not of real import. It is the uniformity of load across drives that is of concern. Where there is some knowledge of these patterns, it may be practical to estimate correction factors for application to the results of the simplified models.

One aspect of the hardware operation which is generally simplified is the seek time function. This is usually assumed to be a linear function of the number of cylinders to be traversed.

Actual seek time functions may be quite irregular, and may depend on the location of the start and end positions as well as just the distance [Frank, 1969], but many are reasonably linear. The comparison of seek rules may be sensitive to this simplification but transfer rules should be little affected.

Another general consideration relevant to this study is that of measures for performance. The measure which is generally preferred is the total time in the system, or response time, for a given level of load. The mean response time is the simplest measure, but measures of the variability of response time--standard deviations or percentiles--are also desirable. Response time is generally considered to be the measure with the greatest real world significance. In some environments (such as real-time) few long waits can be tolerated, in others it is sufficient that the mean wait be short. In either case, throughput is traded off for some response requirement.

But throughput is still a useful and important measure. Analysis for maximum throughput is relatively simple, using the saturation model discussed above. Peak capacity can be an interesting measure, even when applied to an environment where saturation is unusual. And in an environment where large bursts of requests occur, this might be the most relevant measure of performance. A measure which is closely related to throughput is the effective data rate, which specifies the amount of data (rather than the number of blocks) transmitted in a given time. For any given block size, these two measures are equivalent, but when block sizes vary, data rate is usually a more meaningful quantity.

Some explanation is needed to clarify the use of the saturation model to obtain peak throughput figures. In cases where there is no seek scheduling, the disks operate at peak capacity as long as there is always one request in the queue for each module. Shorter queues will reduce activity, but longer queues will give no increase. Thus modeling of peak capacity is easy for this case. It is evident, however, that for the case of scheduled seeks, longer queue lengths will give some increase in throughput. In this case the absolute peak load cannot be modeled, but loads arbitrarily close to the peak can by fixing the queue length at some high value. Since long queues give diminishing returns, this approach is quite practical.

Another quantity of interest in the study of scheduled operation is the service time for the operation (seek or transfer) being scheduled. This gives a direct measure of the scheduling effect and may be easier to obtain than the response time. However, this quantity is not easily related to total system performance.

Any consideration of disk performance implies the assumption that improving disk performance will improve total system performance. This is clear when the disk is major bottleneck, but not at all obvious when the system is more balanced. The increased variability in individual response times which may be caused by scheduling becomes of special concern when gains may be marginal. This issue has not received much attention and will not be examined here except to note some very interesting findings by Nielsen [1971]. In a simulation of a sophisticated time-sharing system, Nielsen studied (among other things) the effect on total system performance of rotational techniques for fixed-head disks. Considering a variety of configurations and loads, he concluded that "the queuer technique can be surprisingly effective even where access is not a severe problem or where other constraints are beginning to bind." And with respect to the disadvantage of scheduling, "the improved system performance offset the greater variability in satisfying individual file requests, so that for the most part the variability in system response times to users actually decreased." While the fixed-head disk is certainly a more attractive device to schedule, these results appear promising for movable-head disks, as well.

One consideration which has not been discussed here is the problem of comparing performance between different devices. While this is not directly relevant to the present study, it should be noted since the most prominent H-RPS device is quite different in storage capacity from the most common conventional disks. The capacity per module of the 3330 is about three times that of the standard 2314-type devices, and the cost is correspondingly higher. So a given installation might have to chose between some number of standard size drives and a smaller number of "double-density" drives. This, in turn, means that a 3330 installation is more likely to be in the marginal region of RPS performance than if smaller drives were used. Meaningful comparison between drives of different sizes is a complex problem which can only be touched upon in this thesis.

Comparative Studies

Existing work on the comparative evaluation of disk scheduling techniques is reviewed in this section. Latency reduction is considered first, followed by seek strategies. Finally work on the combination of both methods is discussed.

The two techniques for rotational scheduling have been studied in relation to FCFS, but not in relation to each other. Fife and Smith [1965] compare S-RPS with FCFS for a range of parameters. The effect of H-RPS is explored by McAulay [1970] in a total system simulation. In addition, an analytic model of H-RPS is currently under development by Teorey [1971], but useful results are not yet available [Teorey, 1972].

The Fife and Smith analysis assumes a saturation load in order to compute peak capacity. While the intention of the study is to consider multiple positioners on one drive, this is virtually equivalent to the problem of multiple drives, each with a single positioner. One assumption of interest is that cylinder distribution is taken to be uniform except that 30 percent of all requests are taken to require no repositioning. Some such provision has the attraction of added realism for many common environments, but the thirty percent figure seems high if there are more than a very few active processes. Inclusion of this factor should enhance the effect of transfer scheduling since seek delays are reduced.

The accuracy of the results obtained is dependent upon two key approximations in the analysis (exponential positioning time and uniform remaining latency). Computation appears to be quite involved since the procedure for the scheduled case involves solving a system of linear equations for the stationary probability distribution of the states of the channel queue. It is apparently for this reason that results are not given for more than three drives.

Although the focus of the work is not on scheduling, some relevant comparisons are given in graphical form. Results are presented for variations in two parameters: block size and the ratio of positioning time to rotation time. As expected, the scheduling effect increases with both the number of drives (positioners) and the relative magnitude of the rotation time. Sizable gains are shown to be achieved by scheduling on three drives in most of the cases considered. Unfortunately, comparisons are not given for more than three drives. Because of this (and uncertainty about the impact of the simplifying assumptions) these results are not discussed in detail here.

McAulay [1970] simulates the operation of an entire computer system using H-RPS on a device with the parameters of the IBM 2314. A detailed model of a typical jobstream under OS/360 is used for the simulation, which includes computation, buffering, and overlapped sequential file processing. Considering up to four tasks (processes), McAulay concludes that gains of 5 to 30 percent in throughput are possible. Mixtures of RPS and non-RPS devices on a single channel are also considered and found to be unattractive (sometimes quite so). Because of the complexity of the simulated load, this work is difficult to interpret in detail and relate to other studies. It does indicate however that latency reduction for disks can have significant impact on total system performance. This complements the findings of Nielsen [1971], mentioned earlier.

The work of Teorey [1971] promises to provide a relatively simple and direct analysis for the mean response time for H-RPS given a Poisson access rate. While a final form of the model is not available at this writing, some very approximate results are derived for the IBM 3330. These figures suggest that sizable gains are obtained with eight drives, and that these gains become minor as the number of drives is reduced to two.

Considered as a group, these works suggest that transfer scheduling can be effective, but give little specific information. Both techniques are shown to work, but comparisons and details of just how they operate are not available, and for the case of scheduling on many drives, no useful results of any kind exist in the open literature.

In contrast, the scheduling of seeks has been dealt with in many papers, culminating very recently in two important studies. The first treatment of seek scheduling is in the paper by Denning

[1967] which compares SCAN with SSTF and FCFS. Like most of these studies, a single drive is considered. A major limitation of the analysis is that results are obtained for response time as a function of queue length but not of the Poisson access rate. Since for a given load, the two rules will operate at different equilibrium queue lengths, the values compared correspond to dissimilar loads. In a computational example with a queue length of ten, SSTF is shown to give significant improvements over FCFS, with SCAN showing about one third of that effect. In later studies, Manocha [1969] considers the circular-SCAN rule and Frank [1969] looks at a batched SCAN.

Teorey and Pinkerton [1971] review these studies and perform an interesting comparison of all the major seek rules using a simulation model of a single disk. Results are obtained for the IBM 2314 with fixed size blocks. SSTF is found to be most effective, with SCAN and related rules not too far behind. With a half second mean response time constraint, SSTF is shown to roughly double throughput capacity. Note that the 2314 has a long seek time (both absolutely and relative to rotation time) compared with more recent devices such as the RP-10 and the 3330. This makes demonstration on the 2314 favorable to seek scheduling.

The recent doctoral thesis by MacEwen [1971a] contains an exceptionally thorough analysis of SCAN and FCFS for multiple as well as single drives. Simulation and empirical measurements are used to verify the analysis techniques and results. The work is also unusual in that results are obtained for nonuniform cylinder distributions which are defined by the Beta distribution. Variation of the Beta parameter gives distributions ranging from uniform to highly peaked at the center. This allows some consideration of grouping effects, particularly the common practice of placing commonly used data on the central cylinders. Numerical results are given for the IBM 2314. Unfortunately, computational difficulty was such that results are not computed for more than four drives. For uniform cylinder distributions, SCAN is found to show significant gains for one through four drives. When the Beta parameter is set to model a significant tendency toward the central cylinders, the effect of SCAN is reduced but not eliminated.

These two efforts suggest that seek scheduling can be effective, but leave open two key questions. As the number of drives increases, the congestion at the transfer phase becomes dominant, so the effect of seek strategies should diminish after some point. It is important to know whether this occurs under normal operating conditions. It is also unclear whether it is worthwhile to combine the two levels of scheduling for additional gains.

Some work is underway in these areas, but little can be said at present. Stone and Turner [1971] are considering the operation of the RP-10 disk as used by the PDP-10 (software based rotational scheduling combined with alternating SSTF/FCFS seek scheduling).

While their analytic model is not in final form, this study is notable for obtaining empirical data on real disk drives in an experiment designed to match the conditions assumed in the analysis. Some simulation results are also presented.

Since operation with module queues fixed at one request per drive is equivalent to peak load with FCFS seek scheduling, this study shows the effect of S-RPS with and without the use of seek strategies. The absolute gain in throughput due to arm scheduling is shown to be roughly constant as the number of drives vary. This means that the relative gain declines. For example, considering the experimental results for a queue length of four requests per drive (compared ~~to~~ one per drive), there is a gain of about 70 percent on two drives, 40 percent on three and 30 percent on four. Increasing queue length to eight increases the absolute gain by only about a third. This is encouraging, in that most of the gain can be achieved with relatively short queue lengths. Looking at the simulation results for eight drives, the gains are only about 17 percent. So the effect definitely diminishes. Note that these are just the additional gains due to arm scheduling - the effect of transfer scheduling cannot be isolated in these results.

Teorey [1971] is also developing a model for such a combination of rules, specifically the use of a batched-SCAN rule with H-RPS. Unfortunately, useful results are not yet available.

In this area, then, it is evident that seek scheduling can be effective when used in addition to latency reduction techniques. Furthermore, it is seen that the effect of seek scheduling is reduced (but not eliminated) by congestion in the transfer stage. This reduction occurs well within the range of practical interest. This also provides an upper limit on the possible performance of seek scheduling alone.

To summarize the present knowledge of scheduled disk performance, there are results in the following areas:

1) some comparison of each of the rotational techniques to FCFS for some numbers of drives,
2) comparison of the major seek strategies for one or a few drives,
3) comparison between rotational scheduling with and without seek strategy for varying numbers of drives.

Areas for which there are no readily useful results are:

1) performance of the rotational techniques in relation to each other for varying numbers of drives,
2) effectiveness of seek strategies (alone) on configurations with many drives.

As suggested earlier, another limitation in existing works is that the results obtained are macroscopic performance measures which give little insight into just how these rules operate and why the results are what they are. This thesis will be concerned with all of these problems. Of course there is still the broader problem discussed above: the nature of real demand patterns and how to model them. This must remain beyond the scope of this work.

CHAPTER III

Simulation of a Multiple Disk Facility

3.1 Choice of Method

It is clear from the previous discussion that the results desired in this study cannot be readily obtained from existing analyses. The choice of simulation modeling over purely analytic methods for the new analyses is based on several considerations. The essence of these reasons is that this work is intended to be exploratory - considering a variety of models in ways which cannot be fully known in advance. So the flexibility of the simulation approach is more important than the transferability of analytic solutions. Variations to be considered include the nature of the requesting process and the block size distribution, as well as the two levels of scheduling. This flexibility also permits the simulator to be matched to the assumptions of various existing studies for validation.

Also involved in the exploratory orientation of this work is the desire for statistics on the microscopic behavior of the disk system in addition to the usual macroscopic performance measures. Simulation permits easy observation of such details as the queueing and service times for each stage of the disk operation and the distribution of channel queue length. It is hoped the value of these insights compensates for the parochiality of simulation.

An additional reason for using simulation is the need for a realistic model. The multiple disk system is not tractable to analysis without certain simplifying assumptions and approximations [MacEwen, 1971a]. So simulation (or some such validation technique) is still required to assure the applicability of such an analysis.

3.2 Model

Based on the above arguments, the major design goals for the simulator are three: flexibility, realism, and detail of output. Variations which must be modeled include device characteristics, configuration (number of modules), environment (load model), and scheduling technique. Statistics must be collected for macroscopic performance measures, such as response time and throughput, as well as whatever microscopic measures promise to be useful.

The simulator is written using GASP for ease of programming and flexibility. GASP permits easy handling of complex scheduling rules and readily adapts to the cyclic character of the disk system state changes. The simulator is designed to be modular to facilitate changes to the model, and consists of three major components:

1. device operation
2. scheduling (I/O supervisor)
3. request generation.

Device parameters may be varied to consider any appropriate hardware. Control parameters are used to select the scheduling rules, which are all built into the simulator. The two lower levels of the simulator-device operation and scheduling-are designed to operate under any requesting process. Thus the model could even be driven by a real request stream. The block length distribution is also easily varied.

Because of these variations, the simulation must be considered as not one, but a family of models with their respective assumptions. The assumptions describing the loads considered here are as follows:

1. Requesting process
   a. Poisson access rate, <u>or</u>
   b. Fixed load per module
2. Distribution in space
   a. Independent uniform distributions over modules, cylinders, sectors (for all runs)
3. Record length
   a. Fixed (for all experimental runs), <u>or</u>
   b. Uniform (for some validation runs only)

Essentially, the experiments in this thesis are based on two models, one with a Poisson load and one with a fixed load. However, some validation runs are made with other record length distributions.

Implementation of the fixed load model involves one complication which should be noted. The obvious approach to maintaining a fixed load is to issue a new request for a module whenever a transfer is completed for that module. However the normal sequence for scheduling on completion of a transfer is to start the next transfer (on some other drive) before returning control to the requesting process. Thus the new request is issued too late for a seek to be initiated at that time, and the seek is delayed by the length of the new transfer operation. Since the intent of the model is to maintain a specified number of requests in the seek queue for each module, some correction is required. This is easily accomplished by using a load generator which attempts to keep n + 1 requests in the system for each drive (where n is the number desired in the queue). Thus the standard saturation load of one request in the queue for each drive is implemented by keeping two requests in the system for each drive.

With respect to device operation, the simulation model is designed to match the operation of real disks in all significant details. All mechanical and queueing delays are simulated, including those for seek initiation and acceptance of seek-done interrupts. The time for purely electronic operations such as seek initiation and head switching is ignored as negligible. The safety factor for latency computation is included and is used to model the reconnection delay for H-RPS. In addition, each disk is started at a random rotational position to avoid errors due to synchronization.

The hardware model is defined by the number of sectors, number of cylinders, rotation time, seek time, safety factor, and of course, the number of drives. The seek time function needs some explanation. For lack of detailed specifications on the RP-10, seek time is assumed to be a linear function of the number of cylinders traversed. The effect of any error on this study should be minor. For other devices, this assumption of linearity matches the other studies of those devices. So seek time is specified by a minimum and maximum value.

The variables defining each run, then, are the following:

1. device type - hardware parameters
2. block length distribution - constant (or uniform)
3. block size
4. load type - Poisson or fixed
5. load level - access rate or queue length
6. number of drives
7. rotational scheduling technique - FCFS, S-RPS, or H-RPS
8. seek scheduling technique - FCFS or SSTF

The outputs of the simulator are many and varied. The average value (as well as standard deviation, coefficient of variation, minimum, maximum, and number of observations) is obtained for:

1. total response time
2. time in the positioning phase
3. time in the transfer phase
4. latency*

* For H-RPS, latency is defined to be the time to position to the record, starting from the initiation of transfer activity (set sector command) or from the completion of the last transfer, whichever occured last.

5. latency* conditioned on number in transfer queue (this also gives the distribution of the number in the transfer queue at the initiation of each transfer)
6. seek time
7. seek distance in cylinders

The time integrated average (as well as standard deviation, minimum, and maximum) is obtained for:

1. number of requests in the system
2. number waiting to seek
3. number of concurrent seek operations
4. fraction of time a transfer is in process
5. fraction of time seeks are not overlapped by transfer activity
6. fraction of time seeking, for each drive
7. fraction of time seek initiation is delayed because of a busy controller, for each drive
8. channel/controller utilization
9. drive utilization, for each drive
10. number in the transfer queue
11. number of seek done interrupts pending recognition by the controller
12. number in the positioning queue, for each drive

* For H-RPS, latency is defined to be the time to position to the record, starting from the initiation of transfer activity (set sector command) or from the completion of the last transfer, whichever occured last.

In addition, figures are obtained for:

1. access rate observed
2. throughput
3. efficiency (in terms of data rate)
4. number of reconnection attempts (for H-RPS)

This is a considerable variety of statistics, and naturally includes some items of minimal value. But it is obviously better to err on the side of too many statistics, especially where insight into operational details is a goal. Most of the items listed found some application in interpretation, validation, and credibility checking of the various runs, and some found important applications which were not fully anticipated at design time.

The operation of simulator is designed to permit a single long run to be divided into shorter sub-runs for statistical purposes. The system is modeled as a continuous long run, but separate and independent statistics are collected for each sub-run. This allows the transient start up period to be isolated and deleted. Such a transient period is allowed for all runs and may be assumed to have been deleted from all results discussed in this work. Division of a run into sub-runs also gives a simple measure of the statistical variation in the simulator operation.

For completeness, the following implementation notes are included: design, code, and test time - about one and one half man-months; program size - less than 800 lines of FORTRAN (not including standard GASP routines); and simulation speed - roughly 2000 requests per minute of CPU time (on the PDP-10).

3.3 Validation

The validity of the simulator was checked in a variety of ways. Most notable is the use of the empirical data on RP-10 throughput provided by Stone and Turner [1971], which indicated a high degree of accuracy.

Initial checking was for a single drive with Poisson input. The first step was examination of a short event trace for logical and statistical errors. The variety of statistics collected permitted cross checking of certain items which could be derived from combinations of the others. Verification of results for special cases which could be solved with standard queueing formulas was also performed. By using a very high rotation speed and zero seek time, all but the constant transfer time is eliminated. Thus the formula for a single server with Poisson input and constant service time is applicable. For an average of six such runs, the simulated queue length is 3.43 compared with a theoretical value of 3.40. Since the standard error for those runs is about 0.3, variability is the limiting factor. By eliminating transfer time and including seek times, a similar test can be made on the seek stage. The single server formula for general service times (mean and variance are measured) shows similar good agreement there.

Further validation for Poisson input on one drive was performed by matching the simulation to published results. The simulation by Teorey and Pinkerton [1971] makes use of an equivalent model with parameters for the IBM 2314. Results for mean number in the system given access rate [Teorey, 1972], were checked, with results as shown in Table 3.1(a). Again, run to run variation obscures any inherent disparities. Note that these runs are not at all short, consisting of six sub-runs of five hundred seconds (some 6000-7000 requests) each. Such a series requires about eighteen minutes of CPU time on the PDP-10. It is suggested that the apparent disparities near saturation are due to the closer approach to steady-state conditions achieved with these long runs. Similar comparison was made with computed values obtained using the analytic results of MacEwen [1971a: Section 4.2.2]. The simulator was modified to match his assumption of uniform record length distribution. Results show excellent agreement (see Table 3.1(b)).

Simulation of four drives with Poisson input was also checked against graphical results given by MacEwen [1971a: Figure 5.3]. This comparison is shown in Table 3.1(c). Again agreement is good.

In all of these runs with Poisson input, run to run variation is considerable. This arises primarily from the highly variable request load, which is eliminated in the case of fixed load models. In that case the variability is limited to the individual seek and transfer times. Excellent convergence is obtained with reasonably short runs.

TABLE 3.1(a)

VALIDATION RESULTS-POISSON LOAD

For one 2314 drive-Teorey [1972]

	Number in System			
Rate/Second	Teorey	Simulator	Standard Error	
10	2	1.84	0.08	(6x500 seconds each)
13	8	9.1	1.2	
13.5	18	24.4	10.8	

TABLE 3.1(b)

VALIDATION RESULTS-POISSON LOAD

For one 2314 drive-based on MacEwen [1971a]

	Response (ms)		
Rate/Second	MacEwen	Simulator	
6	163	162	(2x1000 seconds)

TABLE 3.1(c)

VALIDATION RESULTS-POISSON LOAD

For four 2314 drives-MacEwen [1971a]

	Response (ms)		
Rate/Second	MacEwen	Simulator	
25	230	215	(1000 seconds)

The fixed load simulation was checked against the empirical figures obtained by Stone and Turner [1971] with gratifying results. These empirical figures are obtained from the actual operation of up to four RP-10 disks under a special load designed to match the fixed load assumptions for a given queue length. As shown in Table 3.2(a), this simulator matches those figures to within a few percent. This indicates that the assumption of linear seek times is quite adequate. The simulation of SSTF was also checked against these figures with similar success (Table 3.2(b)).

Summarizing these findings in terms of accuracy, it appears that the simulator is extremely accurate for fixed load situations. For cases of Poisson input, variability is the limiting factor - extremely long runs are required to obtain high accuracy. It appears evident from the tests under fixed load that any inherent inaccuracies in the simulator are relatively minor and would impact only on very long runs. While this variability is a limitation in some possible applications of such a simulator, it is not a serious problem for the purposes of this work. For comparative analysis, it is precision rather than accuracy which is required. Absolute error is not important if it is sufficiently systematic to permit reliable comparisons. By making comparison runs with the same random number seed, identical request streams are generated for each. This eliminates the largest portion of run to run variation.

This principle may be more clear from the following viewpoint.

TABLE 3.2(a)

VALIDATION RESULTS-FIXED LOAD

For FCFS seeks, one request in the system per drive-Stone and Turner [1971], [Stone, 1972]

	Blocks/Second		
Drives	Stone	Simulator	
4	55.71	55.28	
3	45.18	44.49	(100 seconds each)
2	33.05	32.13	
1	19.18	18.47	

TABLE 3.2(b)

VALIDATION RESULTS-FIXED LOAD

For SSTF seeks, four requests in the system per drive-Stone and Turner [1971], [Stone, 1972]

	Blocks/Second		
Drives	Stone	Simulator	
4	70.13	71.44	(100 seconds)

If a given request stream is input to a realistic simulator, results will be accurate for that particular request stream, and comparisons of different systems will be valid, again for that particular load. That this load differs statistically from that specified by the mathematical model is a minor consideration. Use of a series of such matched runs, each diverging in different aspects from the expected load of the model, will assure that the comparison is not sensitive to these minor differences and can be considered representative for that model.

All runs with Poisson input were matched in this way and found to be completely consistent. Thus the comparative values may be considered reliable.

CHAPTER IV

Experimental Results

Before going into the detailed description and interpretation of the results obtained in this study, a brief summary of the experimentation is provided. This summary is intended to outline the specific cases examined and the types of statistics presented. A directory of figures and tables is included for convenient reference. The actual presentation and discussion of results begins in section 4.2.

4.1 Summary of Runs

Standard Parameters

All experimental runs were made with consistent parameters for the hardware, which correspond to the PDP-10/RP-10 facility. These are as follows:

1. Number of cylinders = 200
2. Number of sectors = 10
3. Rotation time = 25 ms
4. Seek time range = 20 to 80 ms
5. Safety factor = 1 sector

In addition, all runs are for a fixed block size equal to one sector (2.5 ms) except where stated otherwise. This also corresponds to the PDP-10.

It must be emphasized that all results given here apply directly only to the RP-10 disk. The behavior of devices with different timing characteristics will be different, perhaps quite so. It is hoped that the following discussion will take due note of the impact of the RP-10 characteristics, and some attempt is made to draw inferences about other devices, notably the IBM 3330. Great care should be taken in generalizing these findings.

Experiments with Poisson Load

Runs using the Poisson input model were made over a range of access rates to determine the variation of performance with load for each of the rotational techniques. The major performance measure obtained is mean response time, which is given for configurations of four, six, and eight drives. The case of eight drives is singled out as most interesting for intensive study and a variety of factors are examined.

The results of these experiments are presented in both graphical and tabular form as listed below. All values are averages over five runs of one hundred seconds each, which are preceded by a (single) start up period of fifty seconds. Note that all results are given in terms of a nominal value for the access rate. The rate actually observed differs slightly from this value and is given in Table 4.2 (a,b,c).

Figure/Table 4.2(a) Mean Response Time - 8 Drives

4.2(b) Mean Response Time - 6 Drives

4.2(c) Mean Response Time - 4 Drives

4.2(d) Mean Response Time - Combined Results

Figure/Table 4.3 Coefficient of Variation of Response Time

4.4 Channel Utilization

4.5 Mean Number in System

4.6 Mean Time in Transfer Stage

4.7 Mean Latency

4.8 Mean Number in Transfer Queue

4.9 Reconnection Failure Rate

4.10 Drive Utilization

4.11 Mean Number of Concurrent Seeks

Experiments with Fixed Load

The fixed load model is used for a variety of runs which include the effect of seek scheduling, variations in block size, and variations in the safety factor, as well as comparison of the three transfer rules. Choice of the saturation load model for these exploratory runs is for reasons of convenience. The rapid convergence is of course a prime factor. In addition, the single values for peak throughput are more easily compared than curves of response time as a function of access rate. The additional information contained in such curves is not important to the present goals. Considering the regularity of the results which are obtained in the previous section, it seems reasonable to assume that the effect of varying loads is similar for the cases considered here.

The results of these runs are presented as listed below. The run length in all cases was one hundred seconds with a one half second start up period. All figures are peak throughput in blocks per second except for the case of scheduled seeks. In that case the model does not correspond to an absolute maximum since larger queue lengths will give some improvement. Results given are for a constant queue of eight requests on each drive. This is a very high load and may reasonably be compared with the other values. A true peak value would require very long queues and is not of practical interest.

Figure/Table	4.1	Peak Capacity for Scheduled Transfers
	4.12	Effect of SSTF Scheduling
	4.13	Effect of Block Size
	4.14	Effect of Safety Factor

4.2 Effect of Rotational Scheduling

Macroscopic Performance Measures

It is clear that for heavy loads on many drives, rotational scheduling techniques are quite effective. The two scheduling rules are very close by most measures of performance, with H-RPS showing some advantage in most cases. As expected, the effect of scheduling disappears as load is reduced and with smaller configurations. The effect of H-RPS on channel utilization is significant in all cases and could be a major consideration.

The figures for maximum throughput obtained from the peak load model (Figure 4.1) give the clearest overall comparisons. For eight drives, scheduling increases peak capacity by nearly 45 percent. In this case H-RPS gives about two percent better

FIGURE 4.1

PEAK CAPACITY FOR SCHEDULED TRANSFERS

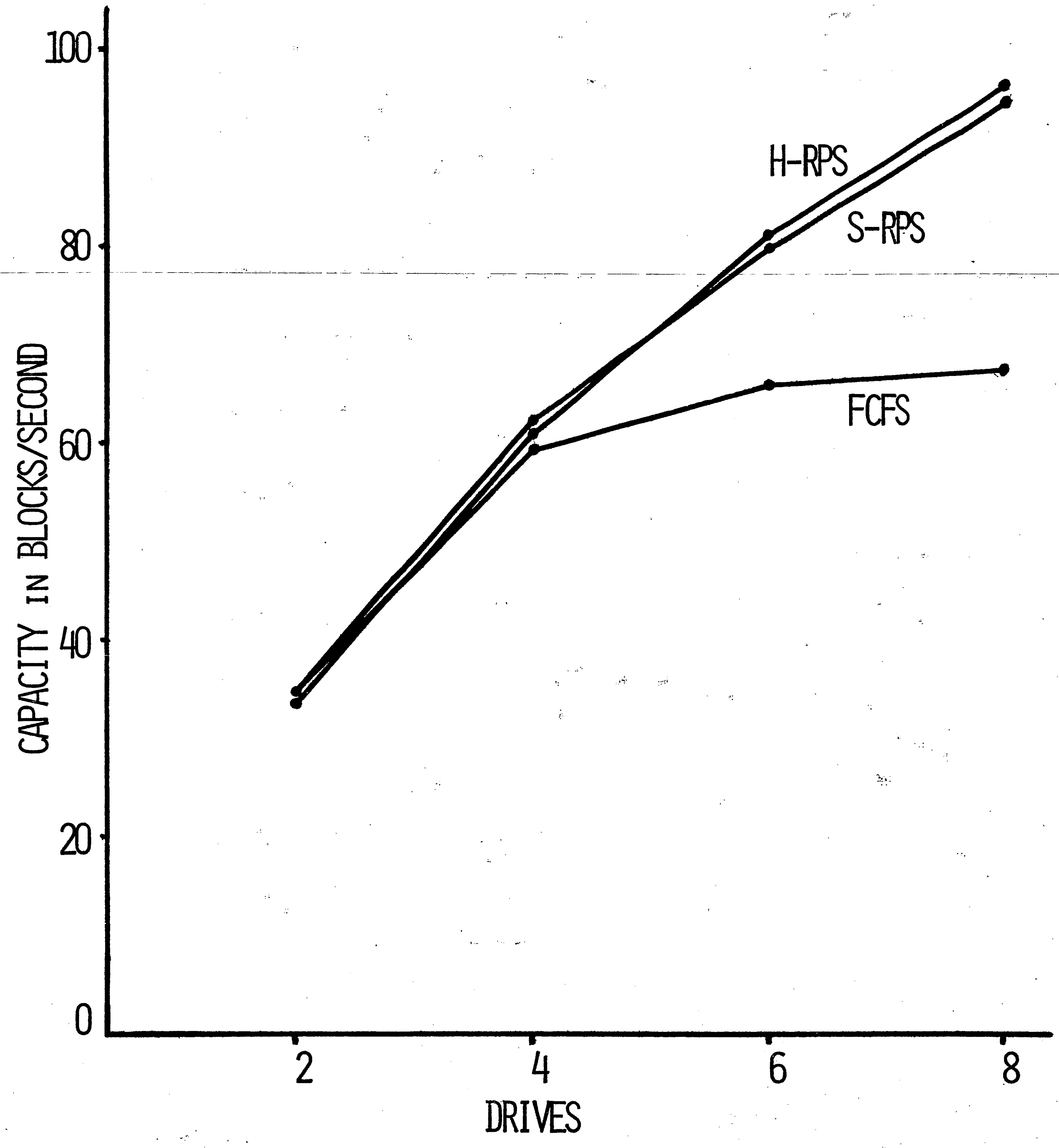


TABLE 4.1

PEAK CAPACITY FOR SCHEDULED TRANSFERS

	Blocks/Second		
Drives	H-RPS	S-RPS	FCFS
8	96.41	94.67	67.14
6	81.25	79.67	65.83
4	60.86	62.05	59.22
2	33.40	34.73	34.73

performance than S-RPS. With six drives, the gains are reduced to about 20 to 25 percent. Even so, note that six drives which are scheduled offer significantly more capacity than eight drives which operate conventionally. Going down to four drives the gains are only about 5 percent or less and for two drives there is no gain at all. This is clearly a result of the decreasing queue lengths.

Looking more closely, a number of interesting observations can be made. Most striking is the fact that the advantage of H-RPS for six and eight drives is reversed for two and four. Furthermore, for two drives, H-RPS is even worse than no scheduling at all (FCFS). This effect can be attributed to the difference in operation with respect to the safety factor. H-RPS can never handle a request which arrives with a latency smaller than the safety factor. Such a request must wait at least one revolution. With S-RPS, such a request will be handled if there are no alternatives in the queue. When queue sizes are small, this puts H-RPS at a disadvantage to the extent that for some requests it will do worse than FCFS. As queue sizes increase, this is balanced out by other gains. It must be emphasized that this effect disappears as the safety factor is reduced. For a device like the IBM 3330 which has a safety factor of only two sectors out of 128 (rather than one out of ten) this effect should not be significant.

Another observation of note concerns the extent to which the transfer operation is a bottleneck in the cases examined.

It can be seen that for FCFS, there is a clear case of diminishing returns as drives are added. This is due to the fact that transfer throughput is limited to 66 2/3 blocks per second regardless of the number of drives (average service time is 12.5 ms + 2.5 ms in all cases). For the RP-10, this is a clear limitation for configurations of more than four drives. With eight drives, heavy loads completely saturate the controller. With the introduction of scheduling, this congestion can be used to advantage. It is here that scheduling is most effective.

The results for Poisson input permit more complete investigation of these effects. The response time curves are pretty much what might be expected, based on the peak load figures. With eight drives (Figure 4.2(a)), the effect of scheduling is quite significant. Response times with scheduling remain reasonable long after FCFS has reached saturation. Considering the range in which FCFS is still operable, scheduling gives noticeable improvements in response time for the upper third of that range. The difference between H-RPS and S-RPS is very slight except for the heaviest loads. Curves for six drives (Figure 4.2(b)) are similar, but the differences are less pronounced. For four drives (Figure 4.2(c)), the differences are minor. This variation with the number of drives is seen more clearly in Figure 4.2(d).

The saturation of the transfer stage is clearly visible in the FCFS curve for eight drives. At low access rates the addi-

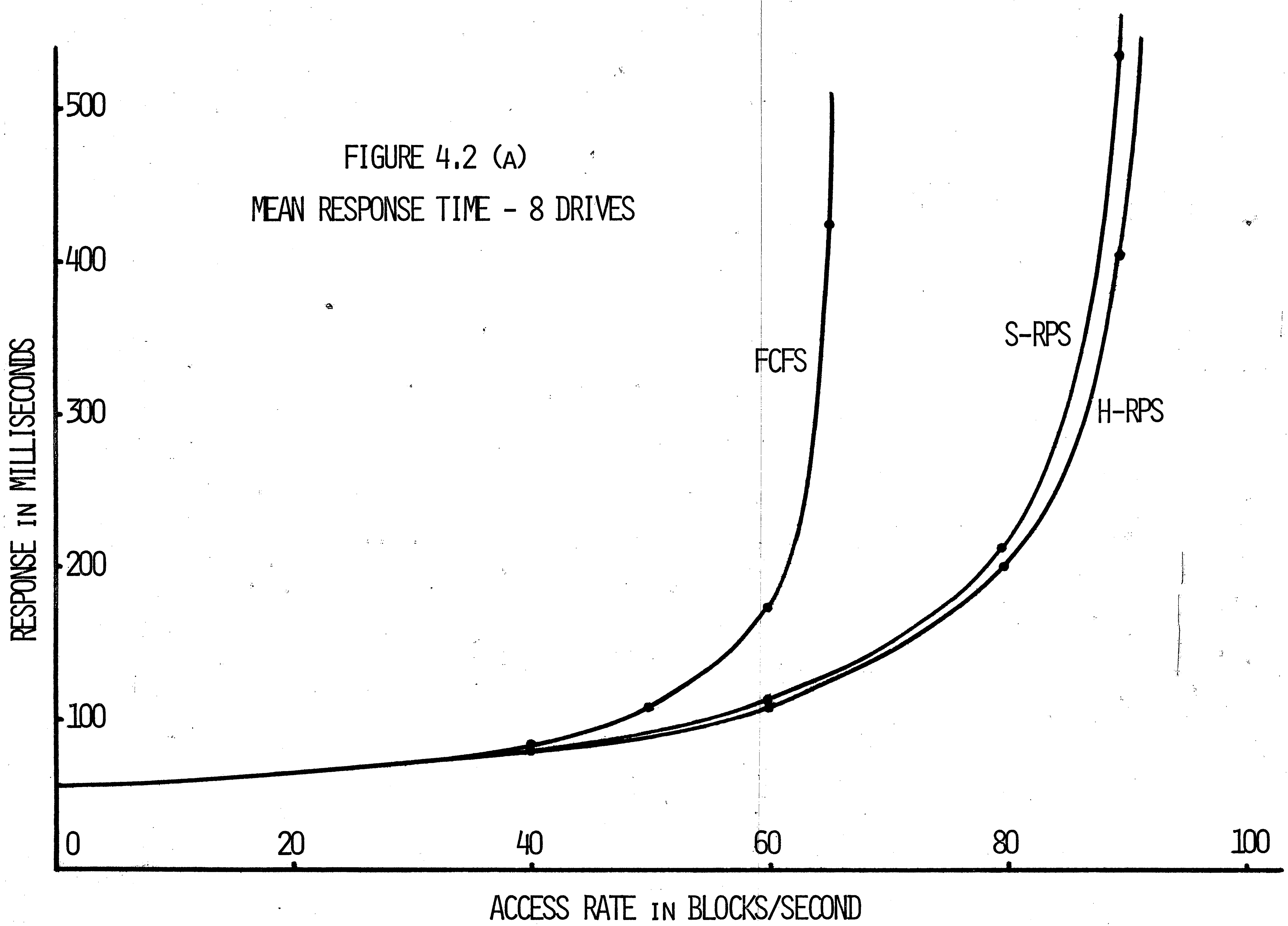


FIGURE 4.2 (A)

MEAN RESPONSE TIME - 8 DRIVES

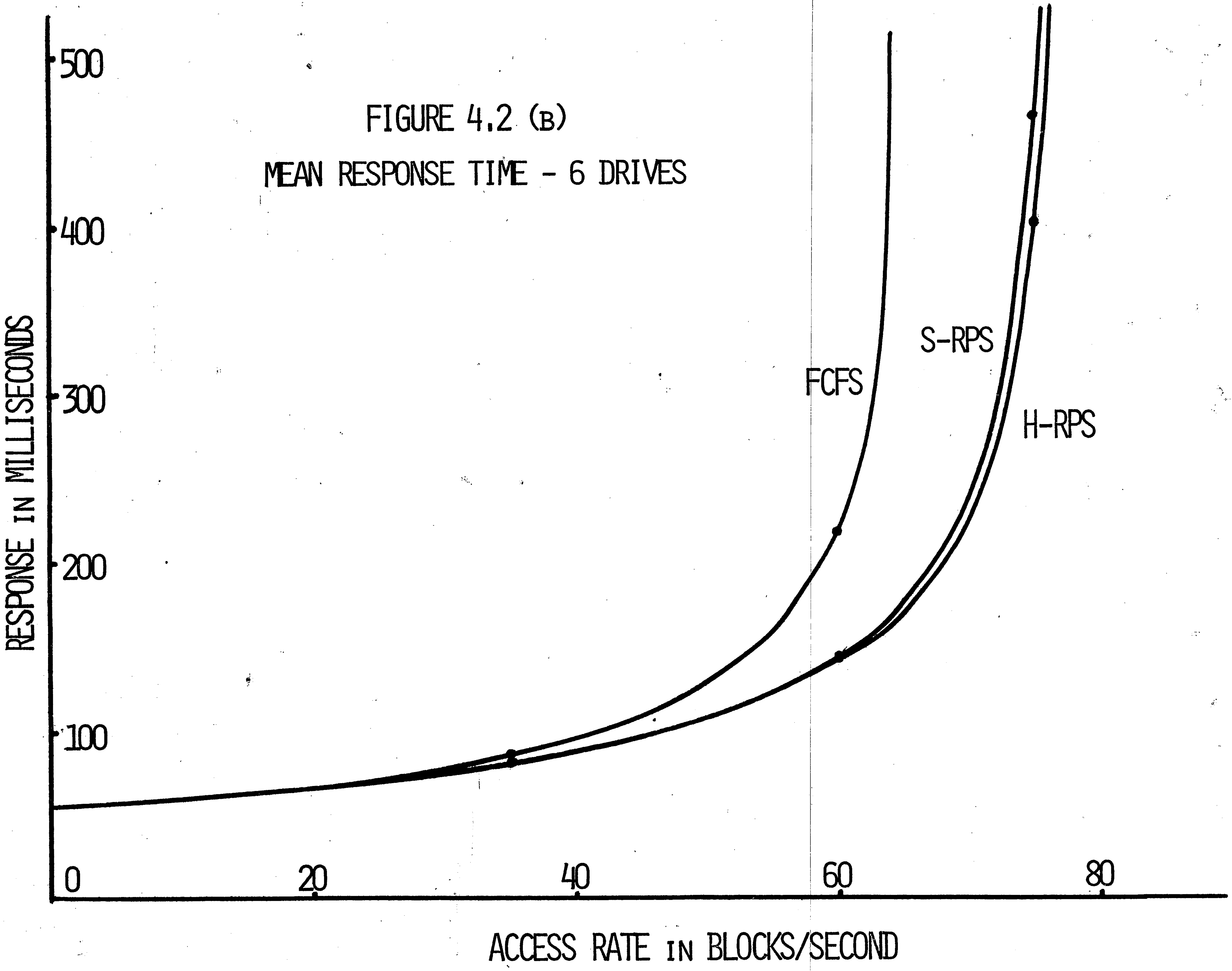


FIGURE 4.2 (B)

MEAN RESPONSE TIME - 6 DRIVES

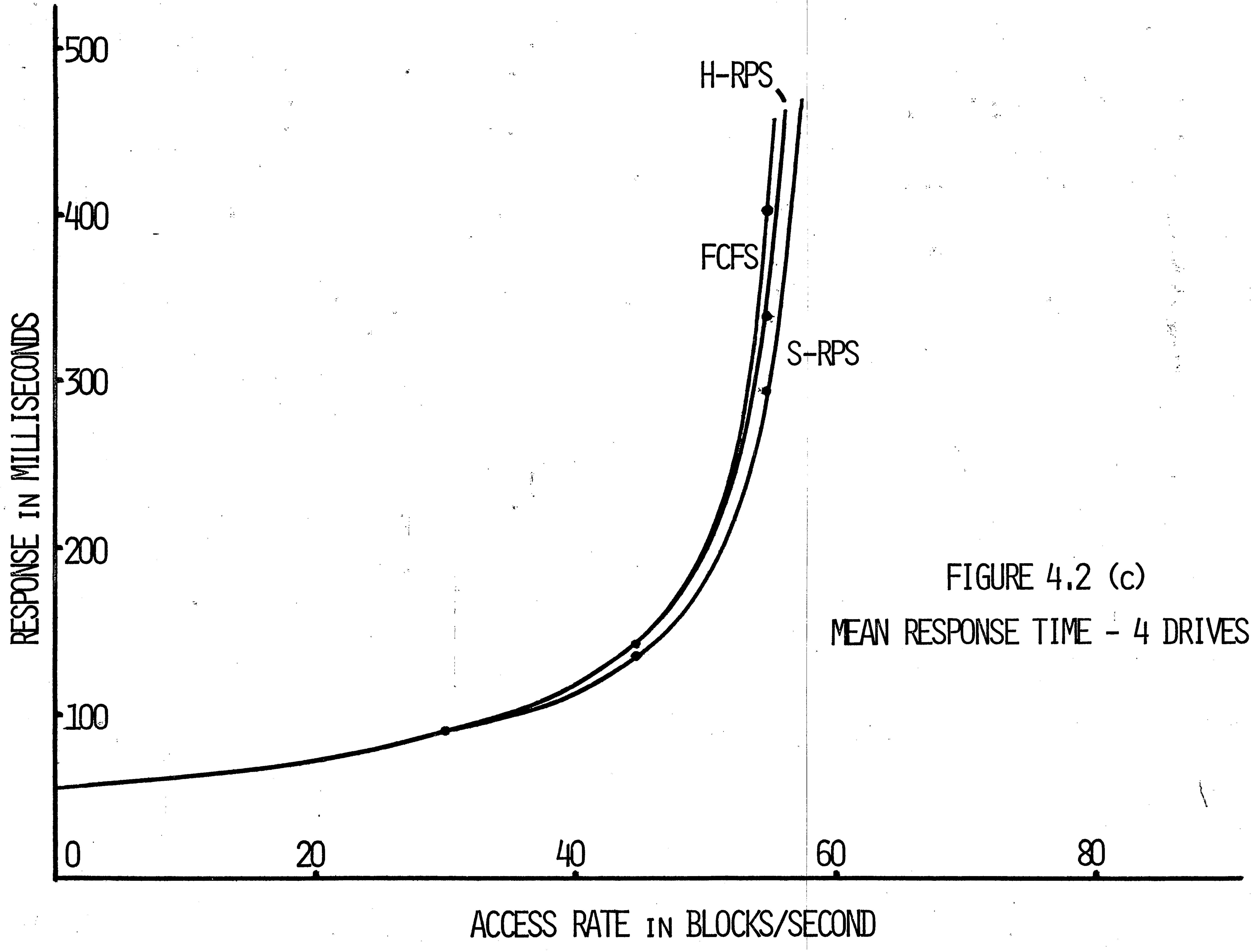


FIGURE 4.2 (c)

MEAN RESPONSE TIME - 4 DRIVES

FIGURE 4.2 (D)

MEAN RESPONSE TIME - COMBINED RESULTS

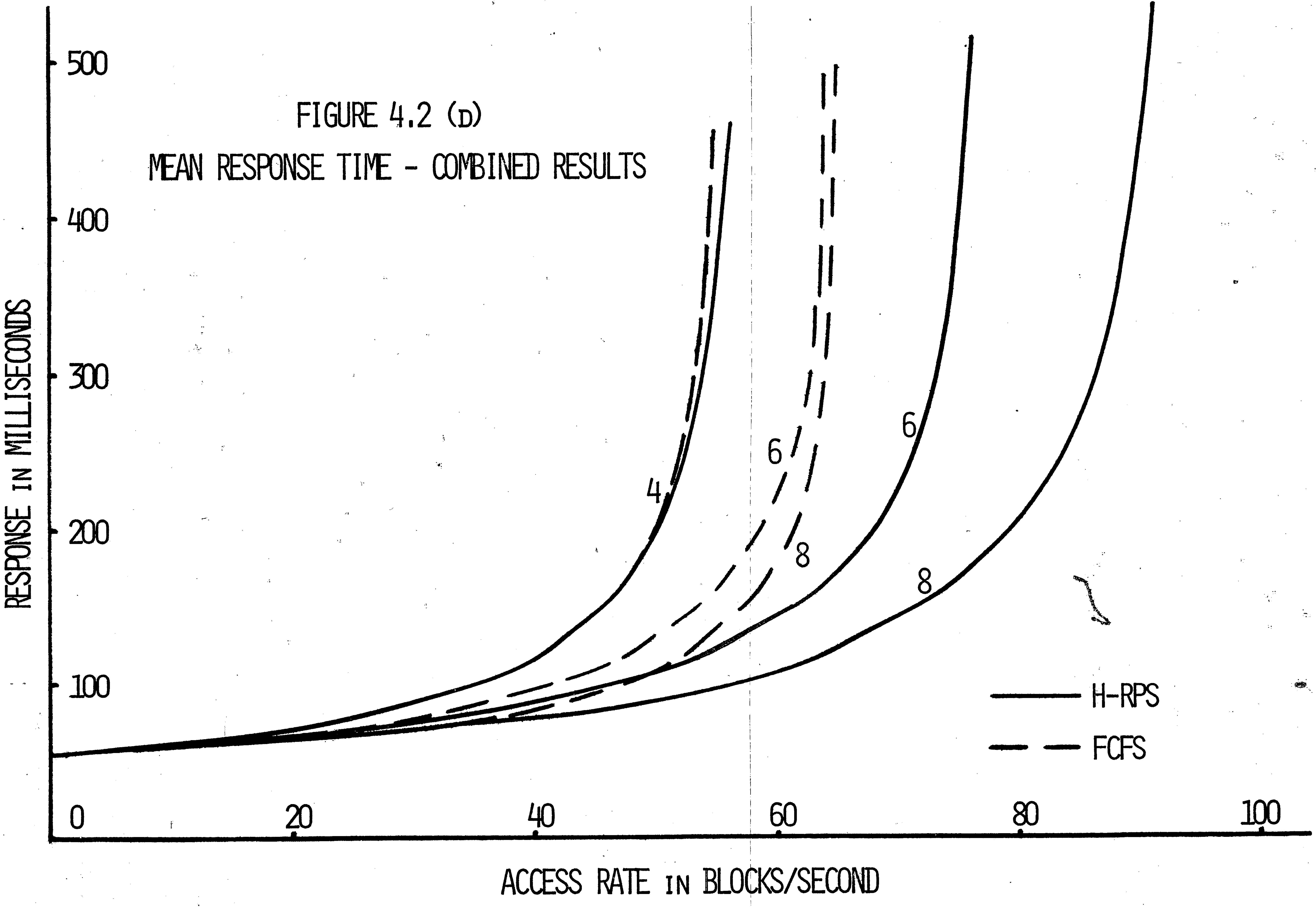


TABLE 4.2(a)

MEAN RESPONSE TIME-8 DRIVES

Access Rate	Response (ms)		
Requests/Second	H-RPS	S-RPS	FCFS
95(94.35)	1012.2		
90(89.30)	404.2	535.4	
80(79.52)	198.6	212.2	
65(64.62)			426.6
60(59.73)	106.82	111.62	172.06
50(49.76)			106.42
40(39.75)	79.12	81.18	85.38

TABLE 4.2(b)

MEAN RESPONSE TIME-6 DRIVES

Access Rate	Response (ms)		
Requests/Second	H-RPS	S-RPS	FCFS
75(75.28)	402.2	465.6	
60(60.14)	142.5	144.1	219.6
35(34.99)	80.9	81.9	84.3

TABLE 4.2(c)

MEAN RESPONSE TIME-4 DRIVES

Access Rate	Response (ms)		
Requests/Second	H-RPS	S-RPS	FCFS
55(54.70)	338.4	291.4	401.6
45(44.74)	140.6	134.9	141.1
30(30.07)	89.0	87.8	88.6

tional arms have a clear effect on performance, but as loads near the limit, this is washed out by the ceiling on transfer throughput. In other cases, a more balanced approach to saturation is apparent.

One word of caution should be interjected here. Since it is the length of the transfer queue which determines the scheduling effect, it is the number of drives in active use which must be considered. If the current system load makes use of only four out of eight drives, for example, performance will correspond to that for a four drive system. For conventional batch systems, this could severely limit the potential effect of transfer scheduling.

It is interesting to note that these gains in mean response time are not obtained at the cost of increased variation in response time. This is often a problem with SOT type rules (including the seek scheduling techniques). Figure 4.3 indicates that the coefficient of variation is never increased by scheduling (for the cases examined). In fact, a clear reduction in variation occurs in the region of heavy loading, especially for S-RPS. This is rather surprising, but the following explanation is suggested. It is the nature of disk scheduling techniques that they are most effective when they are most needed. Thus scheduling tends to level out the effect of transient overloads. Since the transfer queue is limited, no request is neglected for very long. The net result, then, is that the smoothing effect

FIGURE 4.3

COEFFICIENT OF VARIATION OF RESPONSE TIME

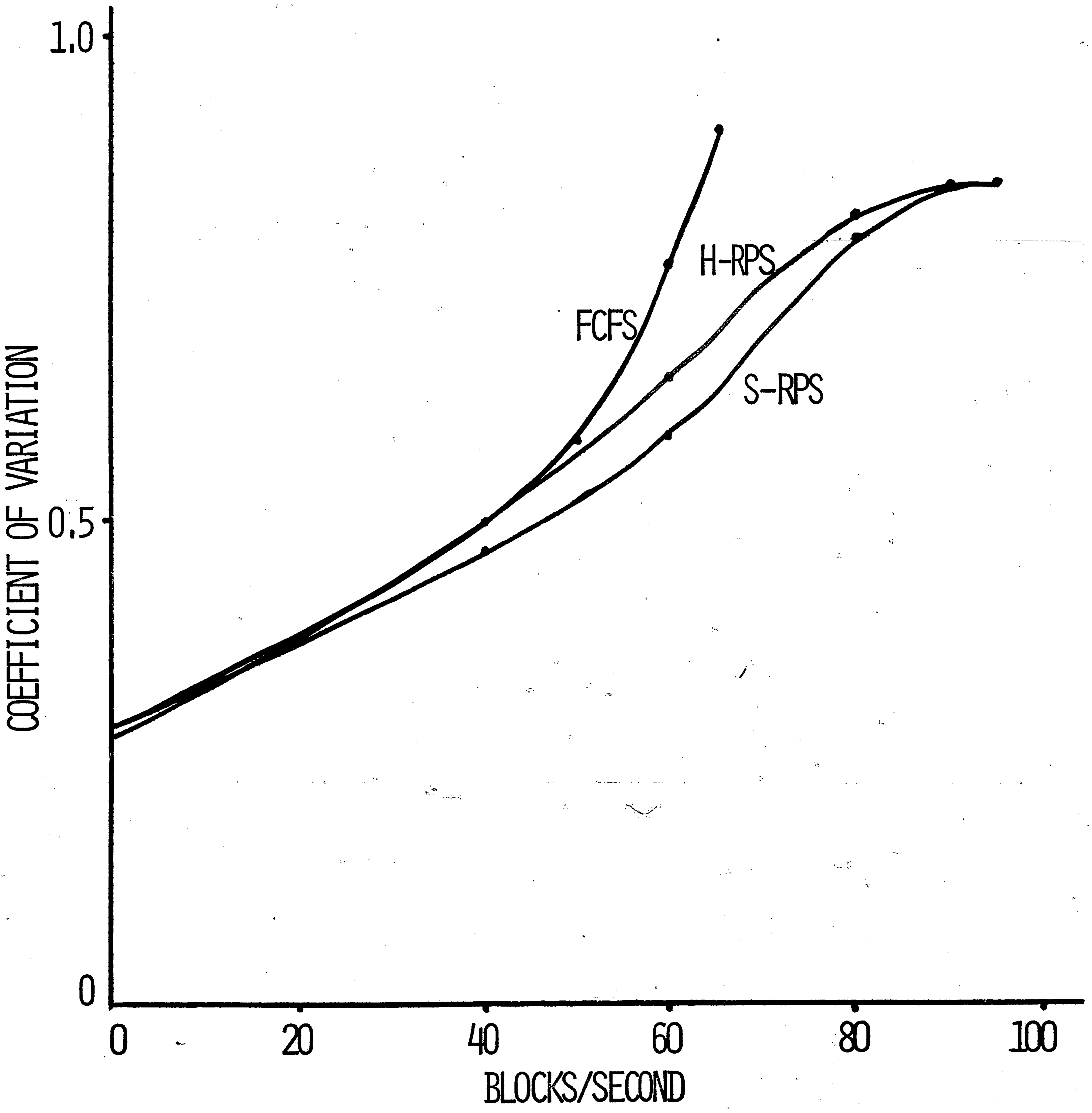


TABLE 4.3

COEFFICIENT OF VARIATION OF RESPONSE TIME

Rate/Second	H-RPS	S-RPS	FCFS
95	.848		
90	.846	.846	
80	.813	.790	
65			.900
60	.645	.584	.760
50			.579
40	.498	.467	.492

TABLE 4.4

CHANNEL UTILIZATION

Rate/Second	H-RPS	S-RPS	FCFS
95	.472		
90	.447	.995	
80	.398	.962	
65			.972
60	.299	.814	.896
50			.748
40	.199	.576	.597

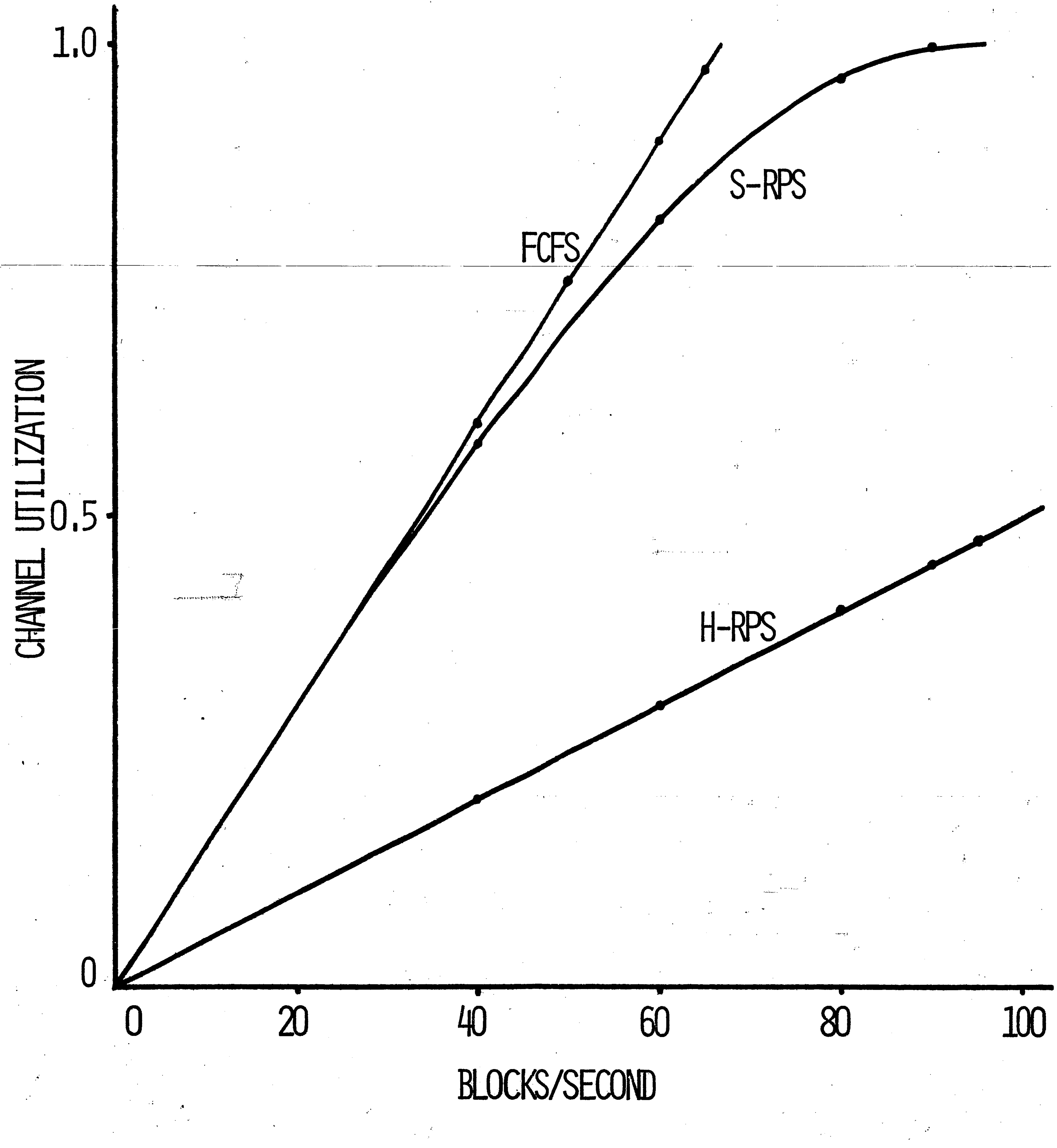
more than compensates for the sequence inversions due to scheduling and a reduction in variation is observed.

S-RPS is seen to yield lower variation than H-RPS at all loads tested. It is not clear just what is the cause for this, but two factors are suggested: the safety factor and the delayed scheduling of H-RPS. Both factors cause sequence inversions which would not occur under S-RPS. Another useful measure of variation is the maximum response time. Sub-run statistics for this maximum (which are not presented here) are in accord with the above results. So scheduling of transfers turns out to be attractive in terms of stability as well as efficiency.

It is in channel utilization that the two scheduling rules differ most, and this is clearly seen in Figure 4.4. Channel utilization is linear for both FCFS and H-RPS. This is predictable, since the average channel time per request is fixed in both cases. For FCFS, mean latency is independent of load; and with H-RPS, channel time is simply the safety factor time plus the actual transfer time. So H-RPS gives a channel utilization equal to one third of that for FCFS - this is just the ratio of the two per request channel times. For S-RPS, channel utilization is reduced only to the extent that latency is reduced. Under light loading, this reduction is negligible. Even under heavy loads, S-RPS requires more than twice as much channel time as H-RPS. This alone can be a very good reason for applying the latter technique. Remember, however, that this excess channel capacity

FIGURE 4.4

CHANNEL UTILIZATION



can be used effectively only for other similar devices [McAulay, 1970]. Sharing the channel with conventional devices results in serious degradation due to interference.

One other macroscopic measurement presented is the mean number of requests in the system (Figure 4.5). These results are similar in form to those for response time. They give a different viewpoint on the effect of I/O delays by showing the average number of requests outstanding at a given instant.

The results presented above are macroscopic in that they reflect the operation of the total system as a black box. As practical measures, they are of prime interest, but for educational purposes, they are too gross to really show what is happening. While a great deal can be inferred from this information, the reasoning must be indirect.

Internal Operation

To supplement the above results, a variety of detailed statistics on the internal operation of the system are examined. These details are quite helpful in elucidating the behavior of the three approaches to transfer sequencing. Close examination of these figures reveals that the macroscopic effects of scheduling are based on a rather delicate balance of opposing forces. Because of this subtlety it is difficult to predict or even explain macroscopic performance relationships without knowledge of such details.

For example, the observant reader may have noticed that the

FIGURE 4.5
MEAN NUMBER IN SYSTEM

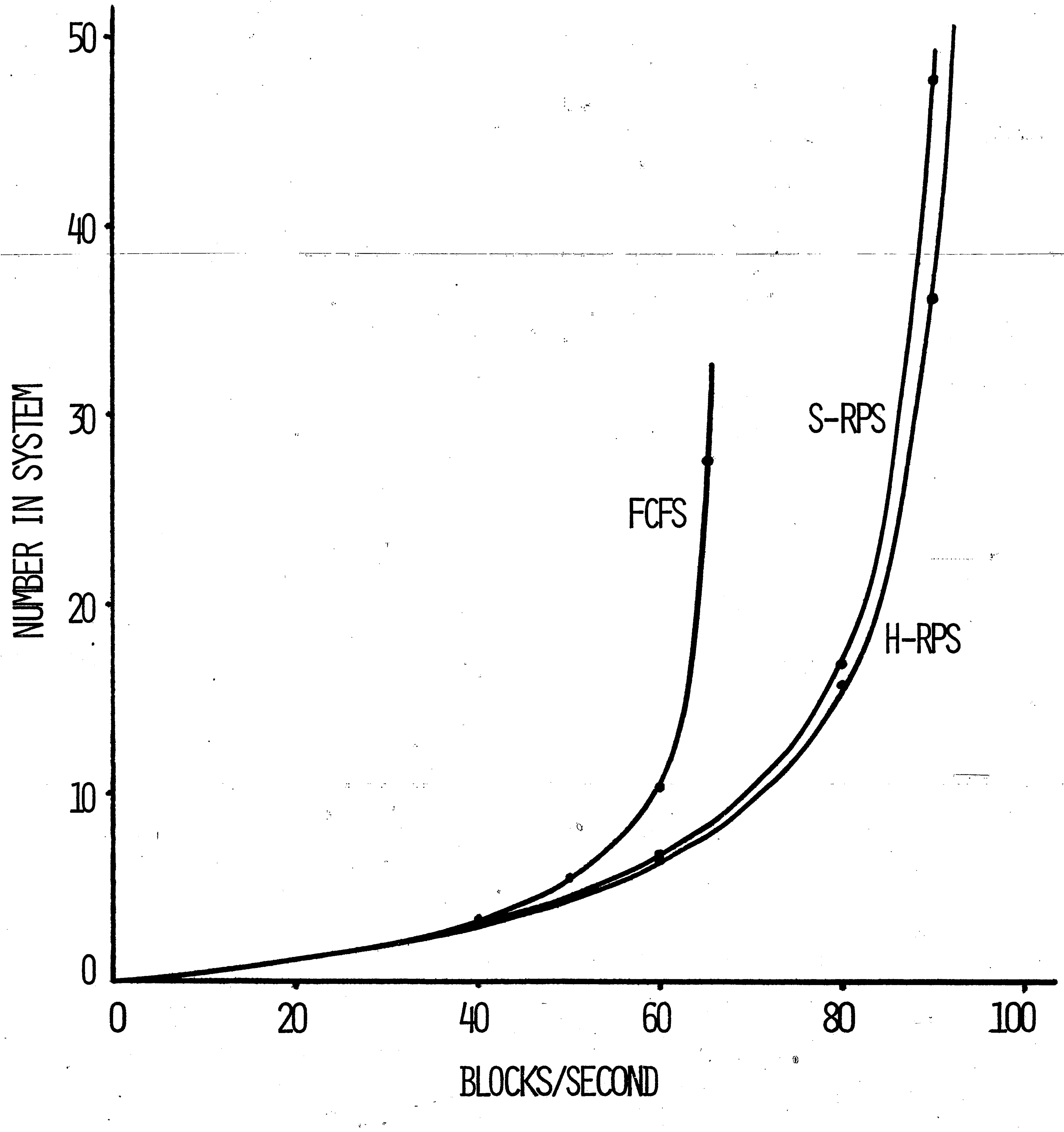


TABLE 4.5

MEAN NUMBER IN SYSTEM

Rate/Second	H-RPS	S-RPS	FCFS
95	95.7		
90	36.1	47.8	
80	15.8	16.9	
65			27.6
60	6.4	6.7	10.3
50			5.3
40	3.16	3.24	3.4

comparison of S-RPS and H-RPS appears to be inconsistent. It is shown that for four drives or less, H-RPS performs worse than S-RPS, and this is attributed to the short queue lengths. But the response time figures for eight drives show that H-RPS is consistently best in the range tested, even when queues are short. The reason for this is not at all obvious until more detailed information is considered.

Looking at the curves for average time in the transfer stage (Figure 4.6), it is seen that H-RPS actually does do worse than S-RPS under light loads. Since total response time for H-RPS is better for the same cases, improvements in the arm positioning stage must more than compensate for poor transfer performance. This is confirmed by statistics for time spent in the seek stage (not given here in direct form). Based on this knowledge, the effect can be attributed to the ability of H-RPS to start seeks earlier than the other rules. Note that this compensating effect disappears at very light loads (since the channel is rarely busy), so that there H-RPS should give worse total response times. This is not of practical importance and was not tested.

Considering the transfer time figures more generally, it is seen that the two scheduling rules give similar results, both better than FCFS at moderate and high loads. H-RPS, which is best at high loads, crosses the curve for S-RPS, and then for FCFS, as the load decreases.

The reduction in transfer time is, of course, an indirect

FIGURE 4.6

MEAN TIME IN TRANSFER STAGE

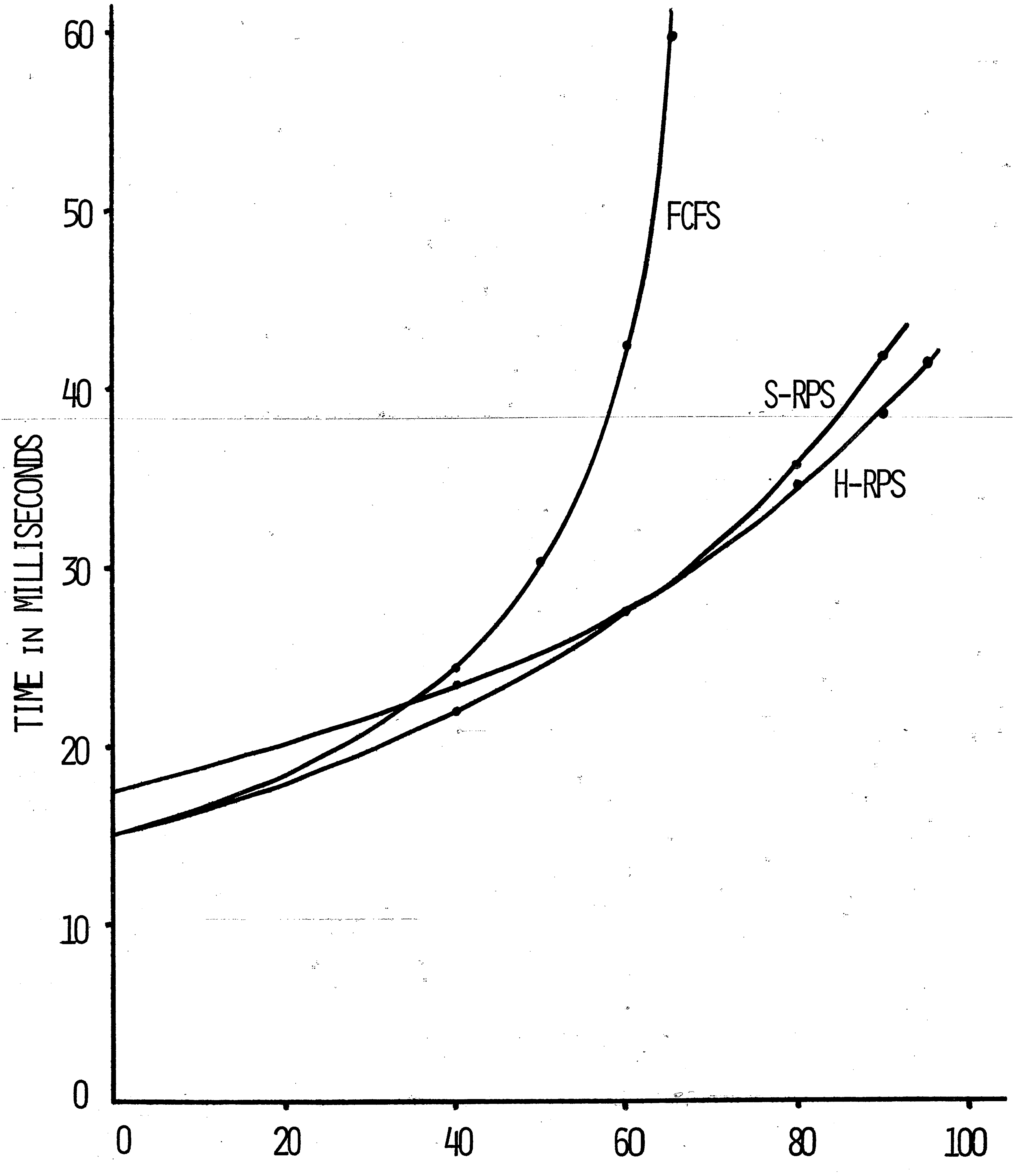


TABLE 4.6

MEAN TIME IN TRANSFER STAGE

	Milliseconds		
Rate/Second	H-RPS	S-RPS	FCFS
95	41.2		
90	38.3	41.5	
80	34.0	35.5	
65			59.6
60	27.6	27.6	42.3
50			30.1
40	23.3	21.8	24.1

TABLE 4.7

MEAN LATENCY

	Milliseconds	
Rate/Second	H-RPS	S-RPS
95	7.58	
90	8.04	8.64
80	8.96	9.59
60	10.75	11.12
40	12.53	11.97

effect of the reduction in latency which is the basic measure of the scheduling effect. So it is not surprising that the curves for latency (Figure 4.7) give a clearer and more basic view of this same effect. The differences in the operation of the three rules are quite graphically displayed by this statistic. As noted previously, the latency measured for H-RPS is defined to give comparisons which are consistent with the conventional usage as applied to the other rules. One surprising result for that case is the striking linearity of latency with the access rate (for moderate to high rates). This phenomenon remains unexplained, but it suggests some basic relationship which could be quite useful if it can be confirmed and identified. In any case, the increase in effectiveness under heavy loading is evident for both methods.

These reductions in latency are directly related to the length of the transfer queue. Specifically, it is the size of the queue at the time a request is selected which is the determining factor. Mean length is given in Figure 4.8 for both S-RPS and FCFS. Comparable results for H-RPS were not obtained, but comparison of time integrated mean queue lengths (not given here) indicates they should be similar to the values for S-RPS. It is interesting to observe that the mean queue length exceeds three only near saturation, so that scheduling operates predominantly on short queues.

One statistic which is of interest with respect to the opera-

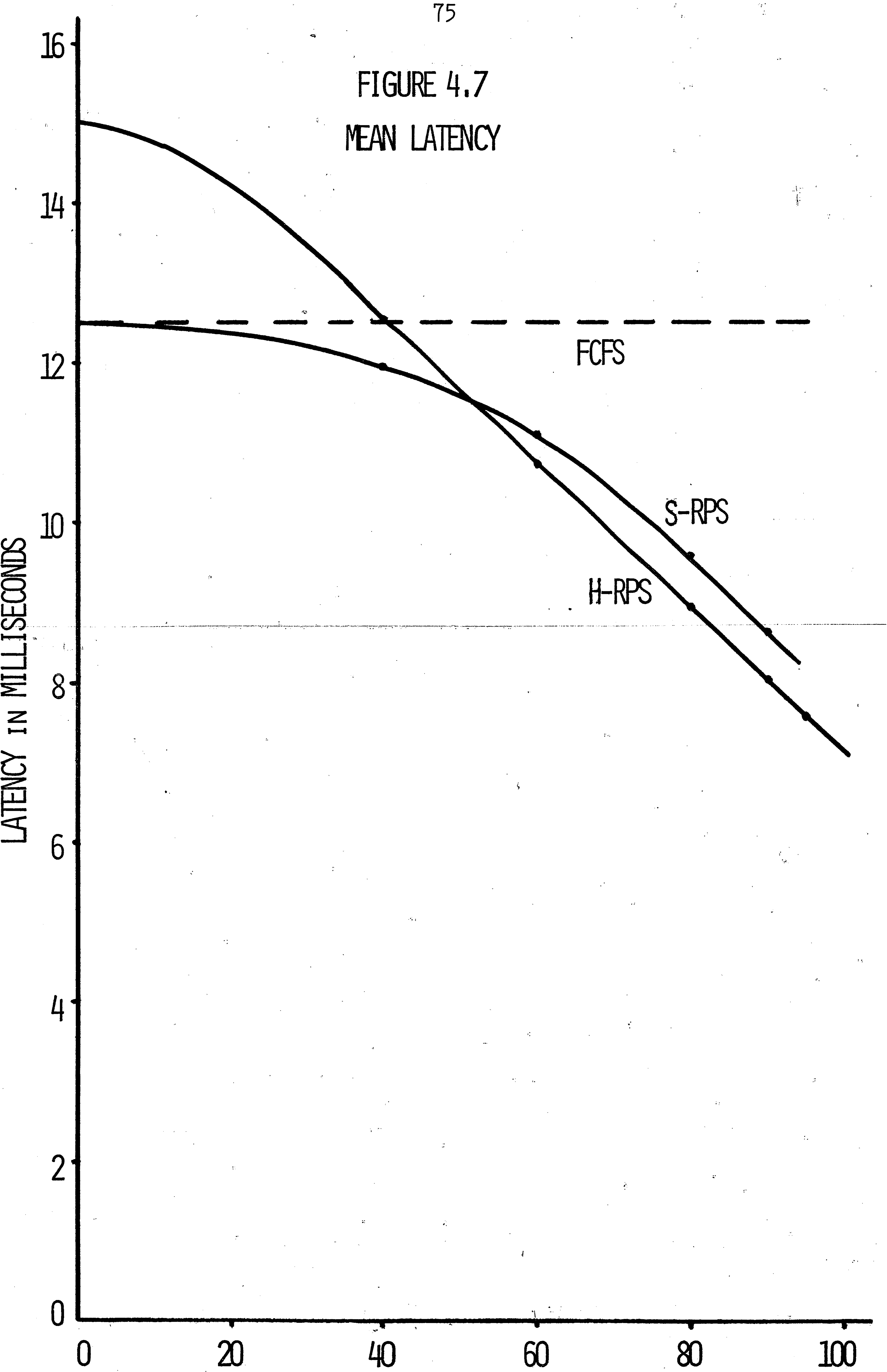


FIGURE 4.7
MEAN LATENCY
FCFS
S-RPS
H-RPS
LATENCY IN MILLISECONDS
16
14
12
10
8
6
4
2
0
0
20
40
60
80
100
BLOCKS/SECOND

FIGURE 4.8

MEAN NUMBER IN TRANSFER QUEUE

AT START OF TRANSFER

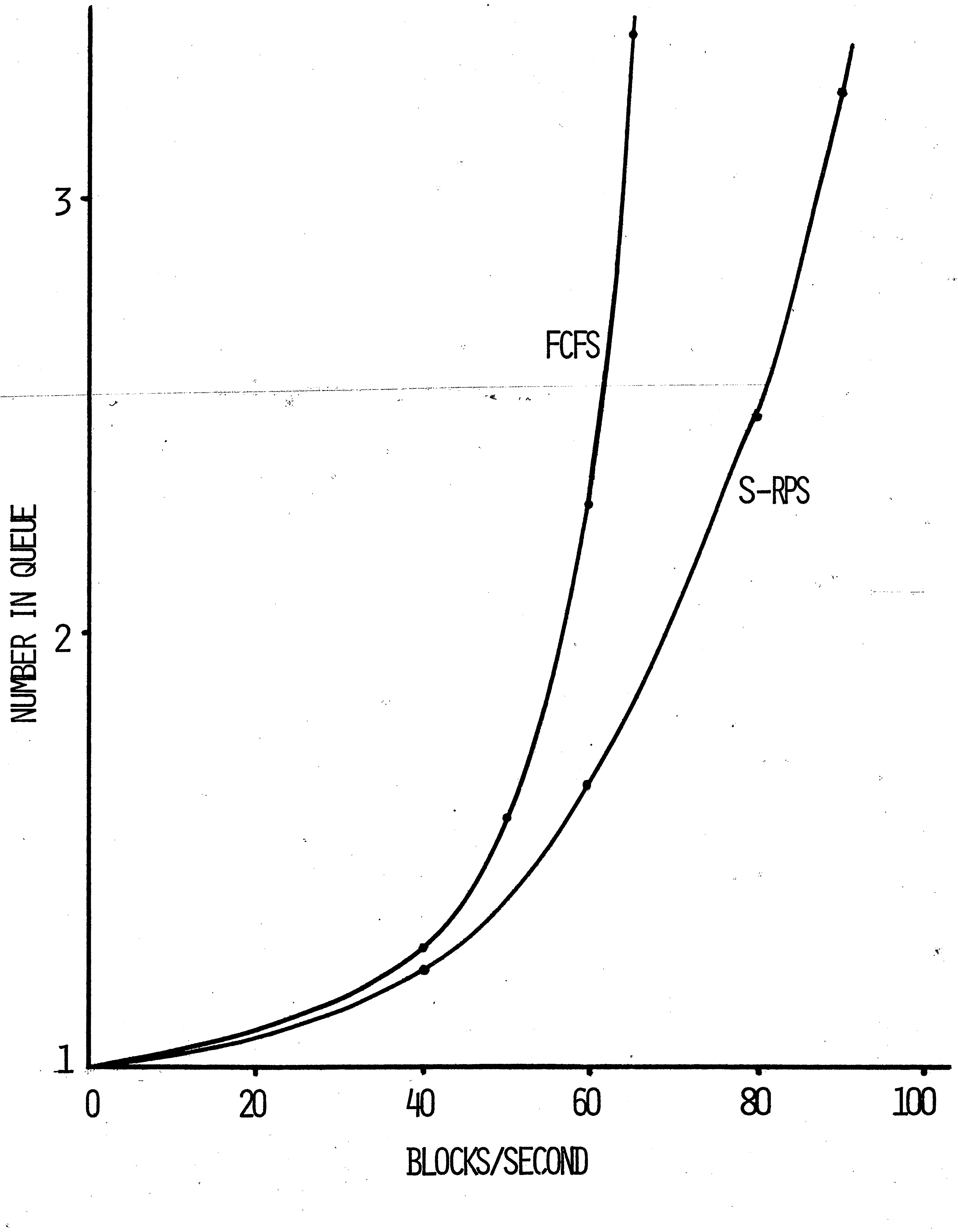


TABLE 4.8

MEAN NUMBER IN TRANSFER QUEUE
AT START OF TRANSFER

Rate/Second	S-RPS	FCFS
95		
90	3.24	
80	2.49	
65		3.37
60	1.64	2.29
50		1.57
40	1.22	1.27

TABLE 4.9

RECONNECTION FAILURE RATE

Rate/Second	H-RPS
95	.473
90	.442
80	.384
60	.275
40	.176

tion of H-RPS is the fraction of reconnection attempts which fail (Figure 4.9). This is equivalent to the mean probability of reconnection failure and is a measure of the extent of interference due to the other drives. One might expect this failure rate to be equal to (n-1)/n times the channel utilization, where n is the number of drives. This appears reasonable since the drive awaiting reconnection is no longer contributing requests (the case of two drives is easily visualized). The simulation results indicate that this is true at low access rates, but as the load increases, the failure rate approaches the full value of the channel utilization. The complication arises from the increasing arrival rate of on-cylinder requests, which take up the slack left by the waiting drive. Estimation of this statistic is central to the approach used by Teorey [1971] to obtain analytic solutions for H-RPS (see page 96).

Shifting attention to the seek operation, some statistics are obtained to clarify this aspect of the total system. The averages for drive utilization (Figure 4.10) indicate the extent to which seeks are a bottleneck. It is clear that seeking is not a major problem with FCFS, since the drive utilization is only .80 near saturation, while the channel utilization is .97. The introduction of scheduling permits increased use of the arms and a balance is achieved where the system becomes saturated only when both stages are saturated. At lower access rates, it is seen that the drives are not heavily used.

FIGURE 4.9

RECONNECTION FAILURE RATE

(H-RPS)

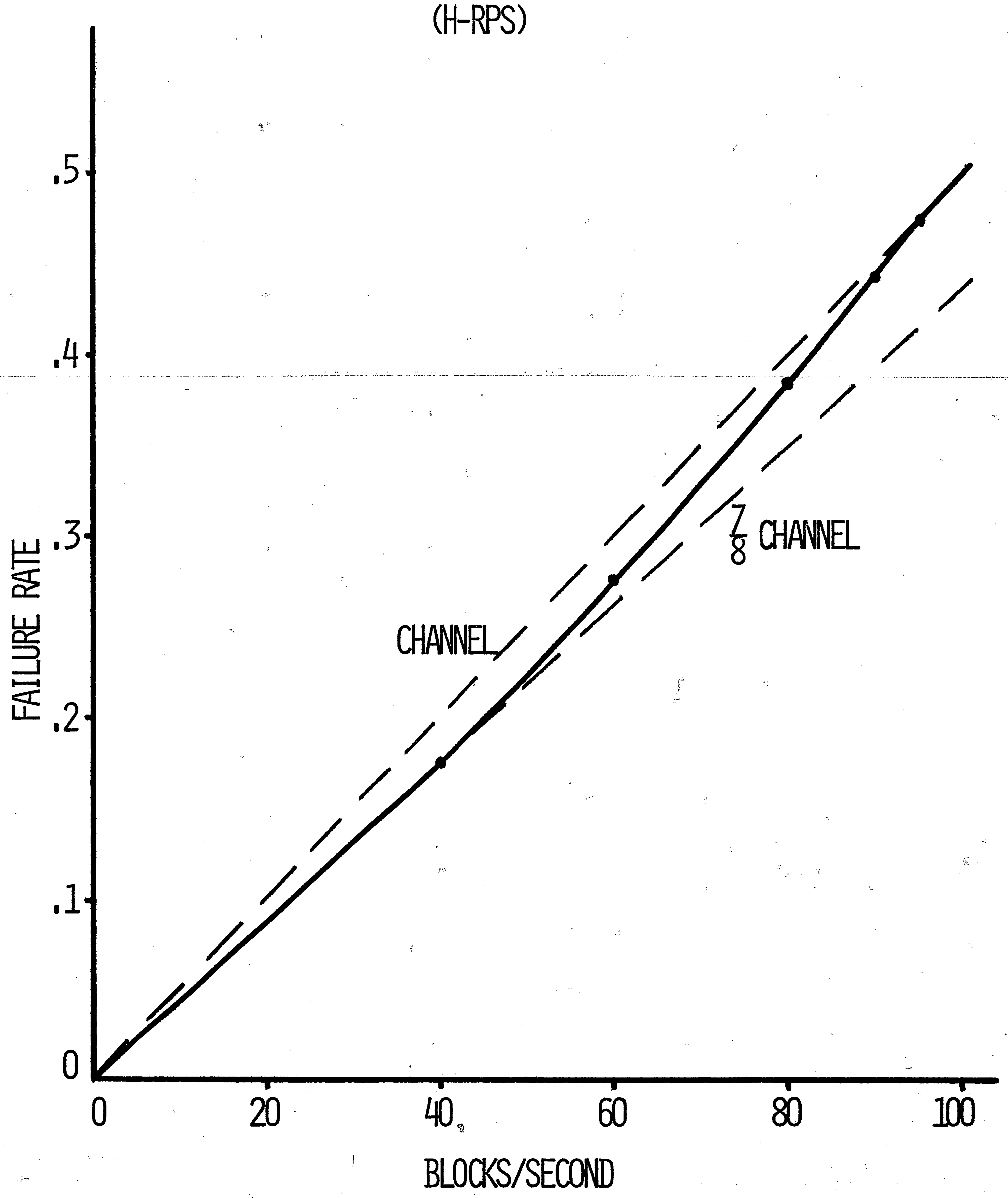


FIGURE 4.10

DRIVE UTILIZATION

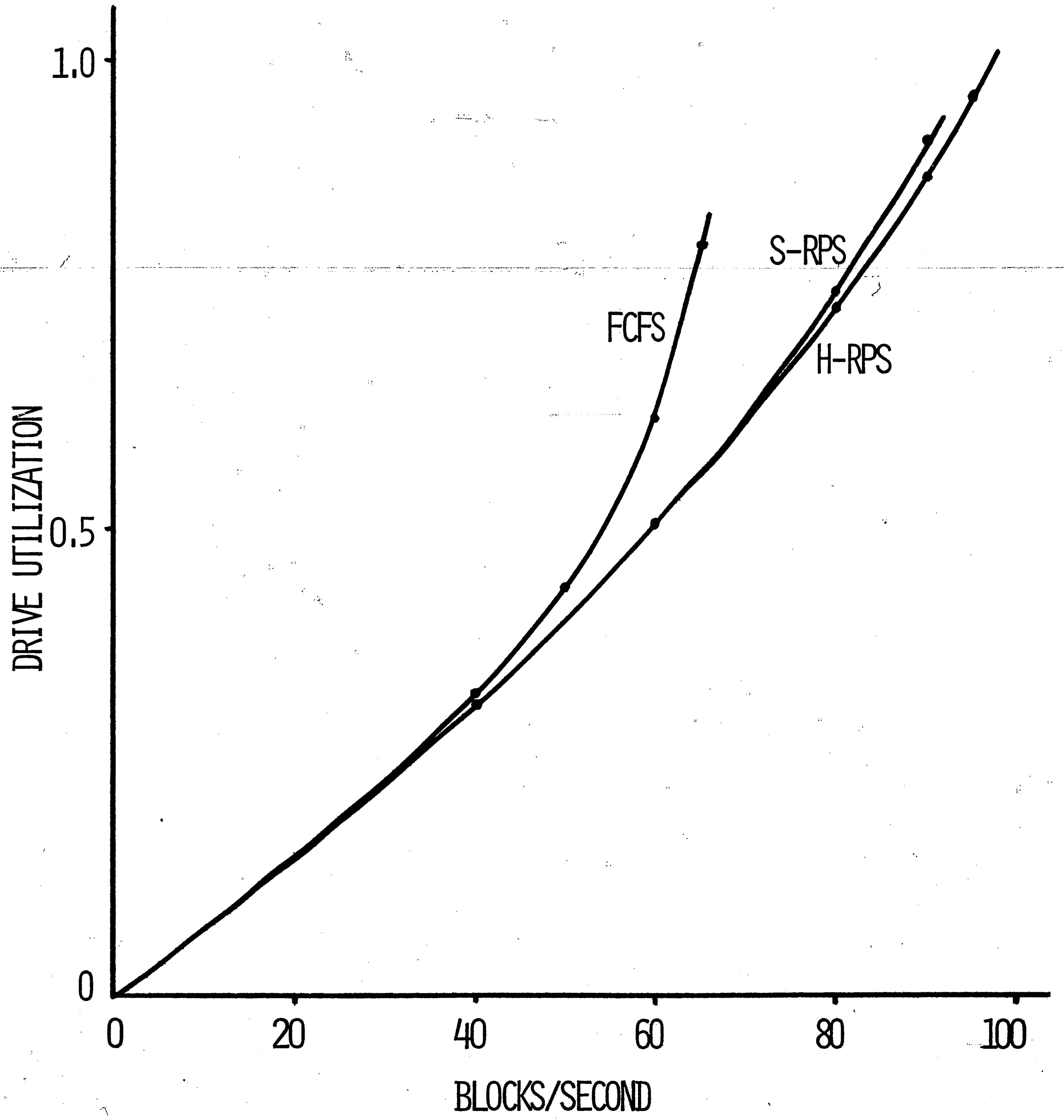


TABLE 4.10

DRIVE UTILIZATION

Rate/Second	H-RPS	S-RPS	FCFS
95	.955		
90	.872	.908	
80	.733	.748	
65			.804
60	.503	.503	.613
50			.435
40	.314	.306	.318

TABLE 4.11

MEAN NUMBER OF CONCURRENT SEEKS

Rate/Second	All Rules
95	3.8
90	3.6
80	3.2
65	2.6
60	2.4
50	2.0
40	1.6

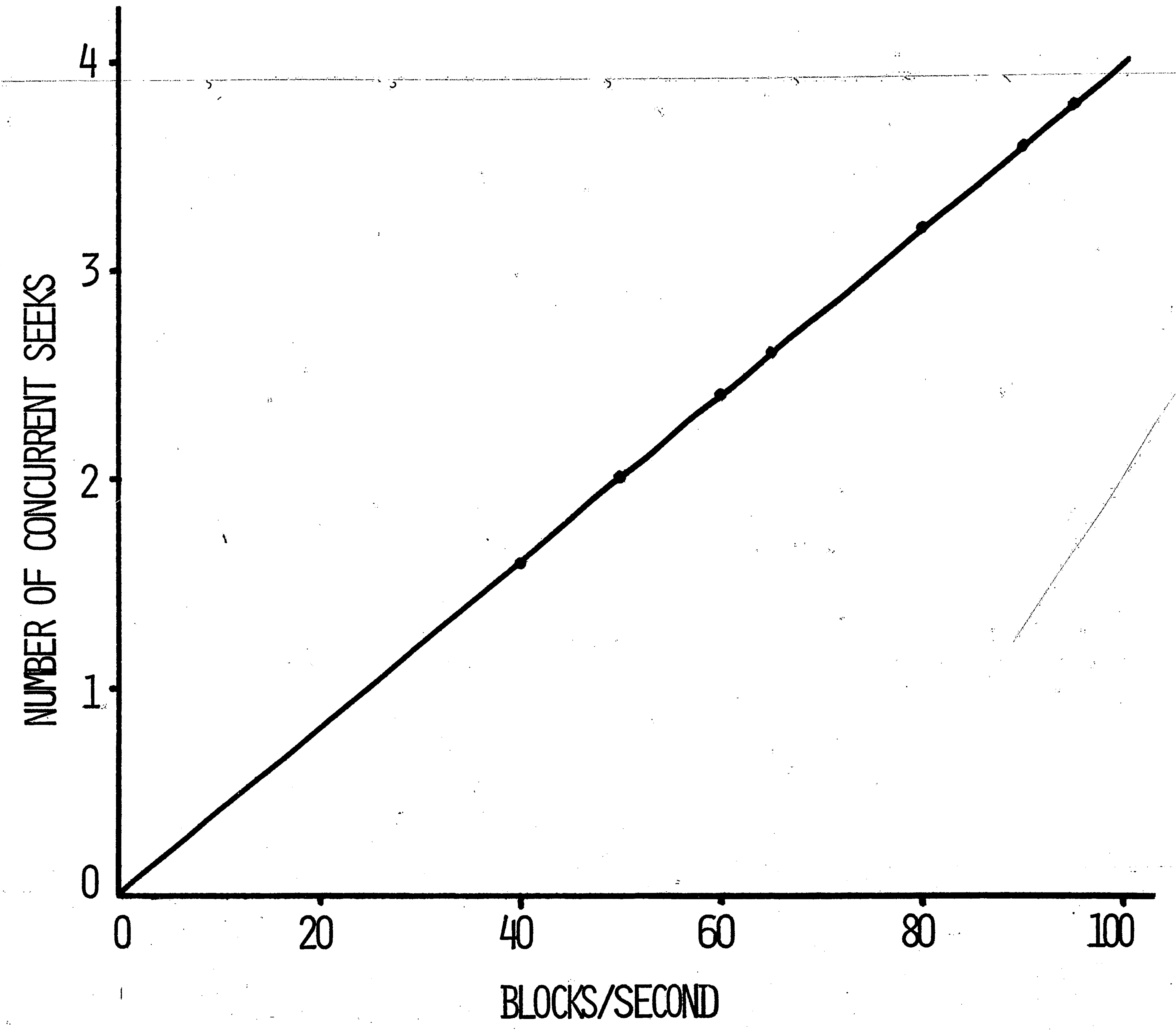
A related statistic which is of some interest is the average length of the seek queue for each drive. This is the variable which relates most directly to the possible effects of seek scheduling. It is found, however, that with uniform load distributions, this mean exceeds one only when loads are very heavy. Results for this statistic are not presented here, but the interested reader can easily estimate these values from results which are given. The mean number in the queue is just the mean number waiting in the system divided by the number of drives. The number waiting is just the number in the system minus the number in process, and this latter is just the drive utilization times the number of drives. Looking at the curves for mean number in the system, it is evident that queues are generally quite short.

Statistics on the average number of concurrent seek operations (Figure 4.11) are given to round out this view of disk operation. These figures are independent of the transfer sequence and are actually trivial to compute, being just the average seek time multiplied by the access rate. So the number of concurrent seeks increases linearly with the access rate. It is worth noting that for eight drives, this average never exceeds four.

Considered as a whole, these figures give a fairly comprehensive look at the behavior of a particular disk system with respect to latency reduction techniques. The following sections will consider what happens when these techniques are combined with seek scheduling, and then explore the effects of block size

FIGURE 4.11

MEAN NUMBER OF CONCURRENT SEEKS



and variations of the safety factor.

4.3 Effect of Seek Scheduling

The effect of combining a seek strategy with each of the rotational alternatives is shown in Figure 4.12. As mentioned above, these figures are not true peak values, but it is reasonable to use them as such. SSTF is used because it is the best seek rule in terms of throughput - this gives a ceiling on the gains obtainable with the other rules (which may be more practical).

It is found that for small configurations, SSTF gives good results, but for larger facilities, the gains are questionable. The most striking observation is that for eight drives, SSTF has no effect at all on peak capacity when transfers are FCFS. This could easily have been predicted, since it was shown that the channel is already saturated in that case. Gains which are limited to the seek operation have no effect on total performance. This does not mean that limited gains in response time cannot be obtained for heavy loads within these limits of capacity. In order to check this possibility, a special run was made for SSTF with a Poisson load of 60 requests/second. The effect on response time is quite small - 163.1 ms compared with 172.1 ms for FCFS - and in one of the sub-runs SSTF actually did slightly worse than FCFS. This poor showing is not surprising, since queue lengths rarely exceed one for such a load and as a result, the reduction in seek time is found to be less than ten percent.

FIGURE 4.12

EFFECT OF SSTF SCHEDULING

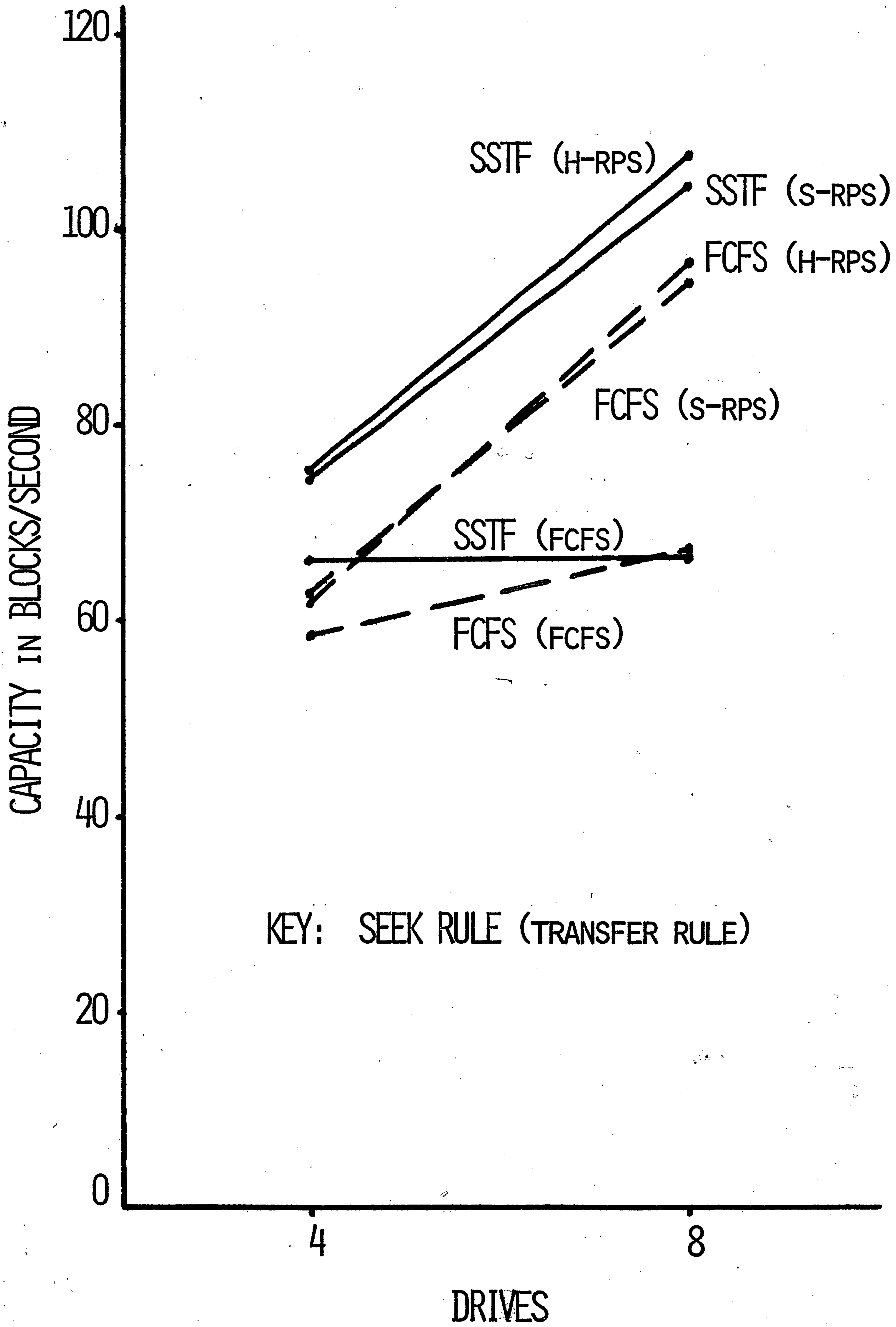


TABLE 4.12

EFFECT OF SSTF SCHEDULING

	Blocks/Second		
Drives	H-RPS	S-RPS	FCFS
8	107.43	104.50	66.74
4	75.21	74.43	66.07

For the cases where SSTF is combined with latency reduction (on eight drives), gains are obtained, but they are still limited by congestion in the transfer phase. With this use of transfer scheduling, the seek phase improvements can be applied in the form of longer transfer queues to obtain some gain in that phase. It is this secondary effect that results in the improved system performance. The gains here are on the order of ten percent which is not at all impressive. In such a case, it is clear that seek scheduling is of secondary importance to transfer scheduling. Nevertheless, the combined techniques give a total gain of 60 percent in peak throughput capacity.

With four drives, the relative effects are quite different. SSTF increases capacity about 12 percent for FCFS transfers and 20 to 25 percent for scheduled transfers. Actually neither approach is very effective when used alone on this configuration. When the two rules are combined, the effect is more than additive, but this is still not very impressive. The total gain for combined scheduling is about 27 percent.

The gains from seek scheduling when combined with a transfer rule show a pattern similar to that observed in the results of Stone and Turner [1971]. The absolute gain is roughly constant as the number of drives vary. Thus the relative gain diminishes with increasing numbers of drives. It is for small configurations that seek scheduling is most effective. Since it is for small configurations that latency reduction is least effective, the two classes of scheduling are complementary in this respect.

4.4 Other Factors

Experiments with the fixed load model were performed to explore the effect of different blocksizes and of the size of the safety factor. Some assessment of these effects is essential to attempts to extend the above results. No attempt at comprehensive study of these areas is made.

Considering blocks of fixed size up to one full track (Figure 4.13), it is found that transfer scheduling gives some gains in throughput capacity throughout this range. As expected, gains are greatest for the smaller blocks. With larger blocks, the latency - which can be reduced - becomes smaller relative to the irreducible transfer time. Consider also that with very large blocks the response time variability may become a problem. Based on these results, it appears that other block sizes up to perhaps a quarter track will result in behavior which is similar to the case examined here.

These results are displayed in two ways to emphasize the important distinction between throughput and data rate. Figure 4.13(a) shows the results as measured, in terms of throughput (in blocks per second). Naturally, fewer of the larger blocks can be transferred in a given time. However, these blocks each carry more data so that the effective data rate - the total amount of data transmitted per unit of time - is actually higher. When varying block sizes are involved, it is this latter quantity

FIGURE 4.13 (A)
EFFECT OF BLOCK SIZE
IN TERMS OF THROUGHPUT

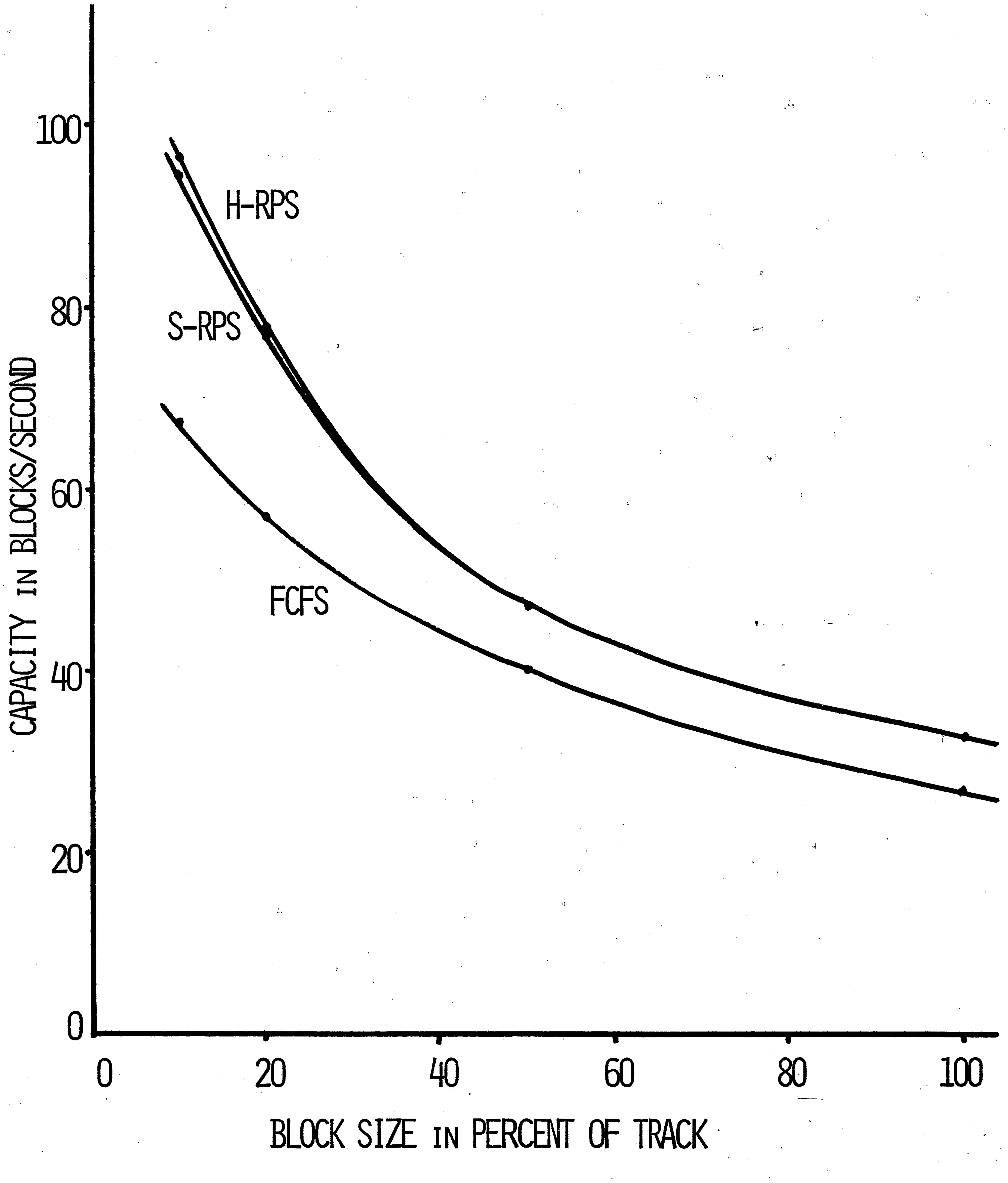


FIGURE 4.13 (B)

EFFECT OF BLOCK SIZE

IN TERMS OF EFFECTIVE DATA RATE

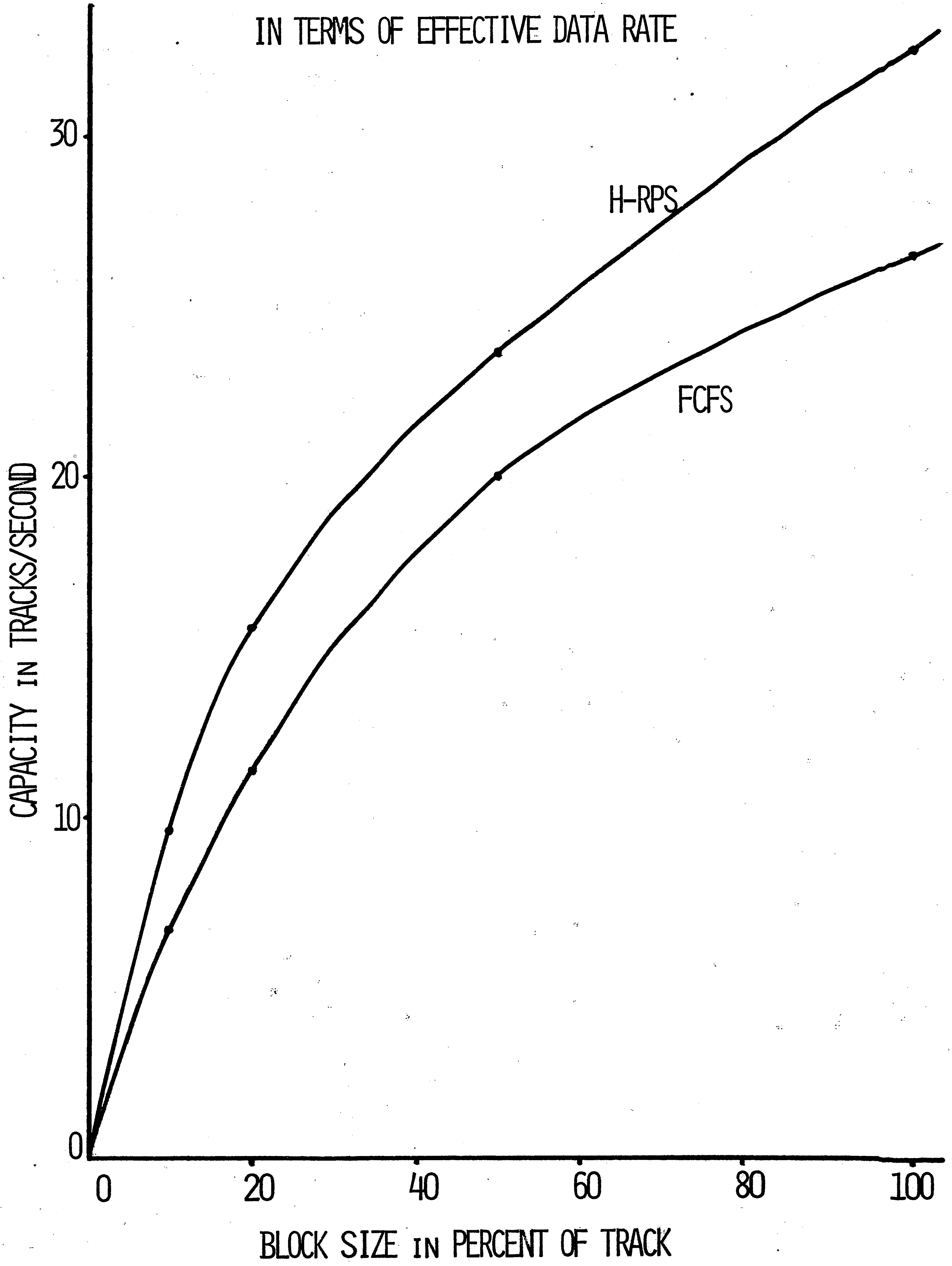


TABLE 4.13

EFFECT OF BLOCK SIZE

Size		Blocks/Second		
Milliseconds	% of Track	H-RPS	S-RPS	FCFS
25	100	32.68	32.44	26.63
12.5	50	47.23	46.92	39.92
5	20	77.96	76.89	56.90
2.5	10	96.41	94.67	67.14

TABLE 4.14

EFFECT OF SAFETY FACTOR

Seeks: FCFS

	Blocks/Second	
Safety Factor (sectors)	H-RPS	S-RPS
0		114.07
.5	107.58	104.54
1 (actual)	96.41	94.67
2	77.94	78.12

Seeks: SSTF (8 requests/drive)

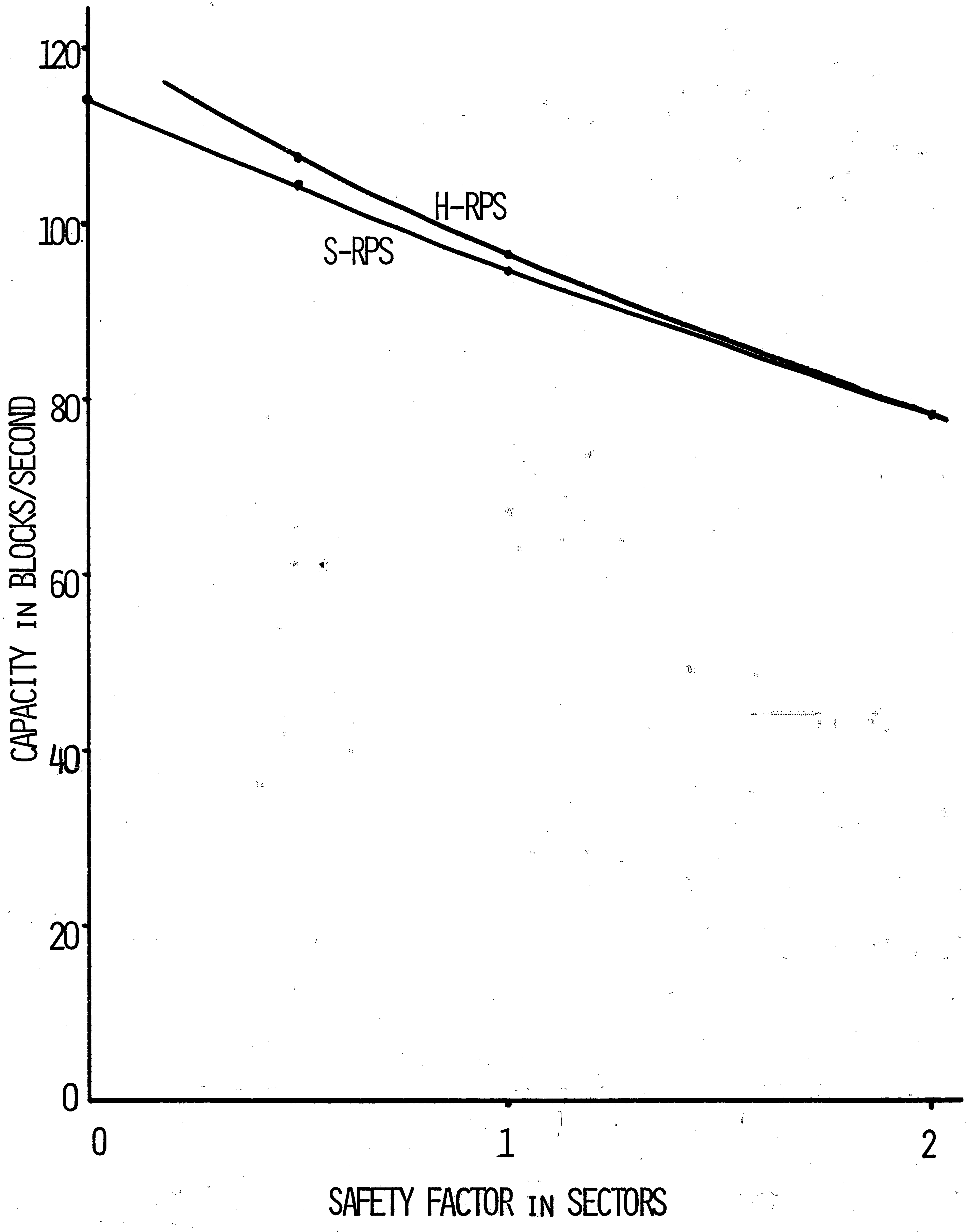
Safety Factor (sectors)	S-RPS
0	141.52
1 (actual)	104.50

which is generally the more meaningful measure of performance. Results are given in terms of this quantity (in tracks per second) in Figure 4.13(b). Those curves illustrate the basic principle that larger blocks make more efficient use of the channel. This is of course due to the increased ratio of transfer time to total service time for each request. The considerations which govern the choice of a block size are too complex to be dealt with here, but it is worth noting that scheduling permits a given data rate to be achieved with a smaller block size than would otherwise be necessary.

Experiments on sensitivity to changes in the size of the safety factor (Figure 4.14) indicate that this is an important consideration. Peak capacity of the RP-10 could be improved by about 21 to 23 percent if the safety factor could be reduced to zero. Reducing it to one half sector would give a proportionate part of this gain. Conversely, increasing the safety factor to two sectors would degrade performance by about 17 to 20 percent. The effect on channel utilization is even greater - this is reduced to one half of the present level as the safety is reduced to zero. In such a case, channel utilization for the case of eight drives would be only about .25 at saturation of the facility. Note also that when SSTF is used in conjunction with improved RPS, a double effect is achieved. With the transfer congestion reduced, more of the potential of seek scheduling can be realized

FIGURE 4.14

EFFECT OF SAFETY FACTOR



and the combined gain is quite dramatic. In such a case combined scheduling becomes much more attractive. So the precision of the RPS hardware is a significant factor. This finding indicates also that the H-RPS on the IBM 3330 has a significant advantage over the case examined here (at least in this aspect) since it has a much smaller safety factor - about 0.26 ms compared with 2.5 ms.

This completes the presentation of experimental results. It is regretted that practical limitations prevented further exploration using this simulator and that those results which were obtained could not be presented more completely. One special application of the simulator is described in the following chapter.

CHAPTER V

Application of the Simulator to Some Analytic Efforts

In addition to the above experimentation, the simulator developed for this thesis has been profitably applied to two analytic models under development by other workers. Errors of reasoning in the initial forms of these models [Stone and Turner, 1971] [Teorey, 1971] were uncovered by checking intermediate results against the appropriate simulation outputs. In both cases, these findings were communicated to the authors and revisions are in process. Discussion of these discoveries is included here for two reasons: they involve additional insights into the behavior of disks, and they make a good case study of the application of simulation to the development of analytic models.

In attempting to apply the Stone and Turner analysis in validation of the simulator, significant discrepancies were found for the number of requests on cylinder at the time a transfer is started. Further checking led back to the argument that the expected number of units seeking at those times is given by the ratio of expected seek time to expected transfer time. But what this ratio gives is the time integrated average over all time, not the average for those particular times. It was then hypothesized that the expected number at transfer start time would differ by one half from the overall average because an average of one seek is completed during each transfer. This was confirmed by the simulation results. Thus a correction for this difference is required in the analysis. Of course this error might have been discovered without any experimentation, but the point is it did not happen.

A similar discovery was made when attempting to check the initial version of the analysis by Teorey. The error here is the oversimplification discussed on page 78: taking the reconnection failure rate to be equal to be (n-1)/n times the channel utilization. Since the test point chosen happened to be near saturation, the simulated value was found to be nearly equal to the full channel utilization. Spurred by this discrepancy, the failure rate was graphed and the effect of increasing loads became obvious. Unfortunately, developing a correction here is not so trivial. Again, this is a case which, in retrospect, should not have required simulation, but did.

CHAPTER VI

Conclusions

6.1 The Effectiveness of Scheduling

This study attempts to assess and illuminate the effect of scheduling techniques on the operation of multiple disk systems. These techniques fall into two basic classes, both of which are explored with the simulation model. Major emphasis is placed on the methods for minimizing the rotational delay involved in scheduling transfer operations. The two alternative approaches of this class - hardware queueing and software queueing - are examined in some detail. The techniques of the other class are intended to minimize the delays in the arm positioning operation. Only one such rule is considered, that which selects the request with the shortest seek time first (SSTF). Since this rule is known to yield the best throughput of this class, these results can be taken to set an upper bound on the potential effectiveness of all such rules.

The major conclusion is that the methods of scheduling transfers based on latency reduction are highly effective when used on disk systems with many modules operating under substantial loads. Both approaches can significantly increase peak throughput capacity and provide sizeable improvements in response time under heavy loads. Of the two techniques, that which combines rotational position sensing with hardware scheduling (exemplified by the

IBM 3330 disk facility) has some significant advantages in performance. While the edge is small with regard to throughput and response time, the hardware approach offers a major reduction in channel requirements. This last effect occurs in all cases, without regard to configuration or load. For the other major approach to the sequencing of disk operations-the scheduling of seeks-results are mixed. Used alone, seek scheduling is ineffective on configurations with many modules, but good results are obtained when the two classes of scheduling are used in combination.

In assessing the potential impact of transfer scheduling, both the number of modules and the I/O load must be considered. This consideration is, of course, intimately related to the hardware timing characteristics for the devices to be used. Results obtained here are for the PDP-10/RP-10 disk, which is similar to many high speed 2314-type devices. It is found that transfer scheduling is highly effective for a facility with eight such drives - peak capacity is extended by nearly 45 percent. For six drives, this effect is cut roughly in half, and for four drives the gains are insignificant. This is a direct result of the fact that scheduling effect increases with the number of requests on-cylinder and waiting to transfer at a given time. It is the number of drives in active use simultaneously which determines the effect of rotational scheduling. If the load is not well distributed over the drives, gains will be lessened. In such a case the seek scheduling techniques become more attractive.

Considering performance in terms of response time, rotational techniques applied to eight drives yield gains whenever access rates are greater than about two thirds of the conventional (FCFS) saturation level. Loads which are uniformly light will not support a noticeable scheduling effect. However, if request loads are erratic, a low average rate may be deceptive. A related measure of performance, the variability of response time is also an important factor. While this is a problem with seek scheduling techniques, the variability is actually reduced when rotational techniques are used.

One other factor which impacts on the performance of rotational scheduling is the size of the blocks transferred. These techniques are most effective when the blocks are reasonably small (up to a quarter track). While throughput gains can be achieved with large blocks, response times may become excessive.

Obviously the question of whether such techniques should be applied to a given computer system is quite complex. One effect of scheduling which is more easily assessed is the decrease in channel utilization obtained when the hardware-RPS technique is employed. This effect is almost always significant. The important question is whether the channel time released can be used to operate other devices. These must be devices of a type which will not tie up the channel, such as other disk (or drum) facilities which use the channel in a similar fashion. If such sharing is practical, the hardware RPS technique is clearly desirable.

A key factor which limits the application of transfer scheduling methods is the need for special hardware. This naturally adds to the complexity of the decisions involved. The fact that seek strategies can be implemented in software is a definite advantage, and in comparing the two methods of transfer scheduling, the differences in hardware requirements are also significant. The advantages of the hardware RPS devices - low channel utilization and slightly better performance - must be weighed against the additional hardware costs. For both transfer techniques, there is one hardware consideration which directly relates to performance. This is the precision of the RPS device in terms of the size of the safety margin which must be allowed for error. Any reduction in this can be translated directly into improved performance. Limitations in the variety of hardware which is readily available will, of course, constrain all decisions on the application of these techniques.

Shifting to consideration of techniques for seek scheduling, results for multiple drives are mixed. Used alone on a facility with many drives, seek scheduling is not attractive. For the eight drive system tested, SSTF had no effect at all on peak capacity. This is due to the fact that the transfer operation is the limiting factor. For four drives, the effect was still minor. Thus it is for configurations of one or a very few drives that seek scheduling techniques can be profitably used in isolation.

When combined with a transfer technique, SSTF shows reasonable gains. This gain is moderate for eight drives and becomes more significant for fewer drives. Thus it complements the behavior of the transfer technique. For all around performance gains, such a combination appears quite attractive. It is just such a combination of techniques that is provided on the PDP-10.

In summary, both approaches to scheduling are attractive in certain cases, although transfer scheduling is probably the most generally useful of the two. Therefore the current appearance of devices and systems which apply these techniques can be viewed with favor. This is not to say that all such offerings are necessarily desirable to all users.

6.2 Other Observations

The goals which were set for the use of simulation in this study were achieved with considerable success. The simulator proved to flexible and accurate, and the emphasis on detailed output was found to be justified.

It is suggested that the specific findings outlined above comprise only one part of the value of this research. What is hoped to be of comparable significance is the insight into the general nature of disk operation which can be gained from the detailed observations presented here. While the specific findings of this study are not directly applicable to other devices, the underlying relationships are basic and readily generalized. The precise nature of these relationships involves a subtle balance of opposing effects. Because of this, intuitive conceptions of the queueing process are frequently deceptive in that they are oversimplified. The kind of internal details presented in this work are not readily obtained except through simulation and no other results of this type are known to exist in the literature. Thus it is believed that detailed study of the statistics obtained here would be illuminating to anyone who is involved in the assessment of disk performance.

Further, it is hoped that results are given in sufficient detail to be of value to researchers who are developing analytic models of disk performance. Such application would be both at the conceptual level and for preliminary testing of model predictions. It should be noted that these results have already proven useful in this application. Reasoning errors were uncovered in two analyses currently under development by other workers.

Comment should also be made on the other design goals set for this simulation. With respect to flexibility, the simulator proved easy to modify - implementation of a variety of models (some of which were not anticipated) was easily accomplished. This also simplified the validation process since results obtained by other workers could be duplicated. The accuracy of the model was found to be excellent. Precision is mixed: convergence is good for the saturation model, but variability is a problem for the Poisson input version. Even for that case, comparative values can be taken with some confidence. In all of these aspects, then, the decision to simulate is found to be justified.

6.3 Suggestions for Future Study

The most basic requirement for effective study of disk operation is the measurement of real world load patterns and the development of models for such loads. Empirical study of I/O load patterns should establish both the nature of the patterns for particular systems and the range of variation between systems. Classification of systems into families which have similar requirements would be especially useful. Models could then be developed to relate these factors to the appropriate measures of performance. A good example of a similar effort of this type is the study of user loads on time sharing systems.

Aside from this, the obvious area for work at the present time is the development of analytic formulations for performance under each method of transfer scheduling. Results for H-RPS would be especially useful in view of the current interest in the IBM 3330. Some such efforts are in progress and have been mentioned in this thesis.

One specific finding of this study warrants further investigation for possible application to the development of analytic models. This is the observation that latencies for H-RPS are linear with the access rate (for high rates), which is as yet unexplained. If the existence of a simple relationship can be confirmed and identified, this could be significant.

The appearance of the 3330 also opens another area. Because of the much enlarged storage capacity of this device over the more common 2314-type systems, the problem of device selection has become more complex. Simulation with a model like that used here could be used to explore the trade-offs involved.

One other useful application of a disk simulator would be an evaluation of existing analytic models for disk performance. A variety of models exist in the literature (most for FCFS), but it is difficult to tell which is best suited for any particular problem. A guide relating accuracy of results to computational effort required would be of value to configuration and system designers.

SELECTED BIBLIOGRAPHY

References Cited

Conway, R. W., Maxwell, W. L., and Miller, L. W. (1967). Theory of Scheduling. Addison-Wesley, Reading, Mass., 1967. p. 184.

Denning, P. J. (1967). Effects of scheduling on file memory operations. Proc. AFIPS 1967 SJCC, Vol. 30 Thompson Book Co., Washington, D. C., pp. 9-21.

Digital Equipment Corp. (1971). PDP-10 referenee handbook. Digital Equipment corp., Maynard, Mass., 1971.

Fife, D. W. and Smith, J. L. (1965). Transmission capacity of disk storage systems with concurrent arm positioning. IEEE Trans. on Elect. Comp. EC-14, (August 1965), pp. 575-582.

Frank, H. (1969). Analysis and optimization of disk storage devices for time-sharing systems. J. ACM 16, 4 (October 1969), pp. 602-620.

IBM. (1970). A guide to the IBM System/370 Model 155. GC20-1729-0, IBM, White Plains, (June 1970).

IBM. (1971a). Reference manual for IBM 3830 storage control and IBM 3330 disk storage. GA 26-1592-1, IBM, White Plains, (July 1971).

IBM (1971b). IBM System/370 operating system input/putput supervisor. GY28-6616-8, IBM, White Plains, (June 1971).

MacEwen, G. H. (1971a). Performance of disk storage devices in computer systems. Ph.D. thesis, University of Toronto, 1971.

MacEwen, G. H. (1971b). Performance of movable-head disk storage devices. (private communication-manuscript), Queens University, Kingston, Ontario, Canada, 1971.

Manocha, T. (1969). Ordered motion for direct-access devices. TR 21.318, IBM, Kingston, N. Y., May 16, 1969.

McAulay, S. E. (1970). Jobstream simulation using a channel multiprogramming feature. Proc. 4th Conf. on Appl. of Sim., December 9-11, 1970, New York, pp. 190-194.

Nielsen, N. R. (1971). An analysis of some time-sharing techniques. Comm ACM 14, 2 (February 1971), pp. 79-90

Sharma, R. L. (1968). Analysis of a scheme for information organization and retrieval from a disk file. Proc. IFIP 1968, North Holland Pub. Co., Amsterdam, pp. 853-859.

Stone, D. L. (1970). PDP-10 system concepts and capabilities. Proc. 1970 DECUS Europe 6, pp. 163-187.

Stone, D. L. (1972). private communication (telephone), February 7, 1972.

Stone, D. L. and Turner, R. (1971). Disk throughput estimation. (private communication-manuscript), Digital Equipment Corporation, Maynard, Mass., 1971.

Teorey, T. J. (1971). Disk storage performance with rotational position sensing. (private communication - manuscript), University of Wisconsin, Madison, Wisc., December, 1971.

Teorey, T. J. (1972). private communication (telephone), January 25, 1972.

Teorey, T. J. and Pinkerton, T. B. (1971). A comparative analysis of disk scheduling policies. Third ACM Symposium on Operating System Principles, Stanford, Cal., Oct. 18-20, 1971. also published in Comm ACM 15, 3 (March 1972), pp. 177-184.

Weingarten, A. (1968). The analytical design of real-time disk systems. Proc. IFIP 1968, North Holland Pub. Co., Amsterdam, pp. 860-866.

Related Works - Disk

Abate, J., Dubner, H., and Weinberg, S. B. Queueing analysis of the IBM 2314 disk storage facility. J. ACM 15, 4 (October 1968), pp. 577-589.

IBM. Analysis of some queueing models in real-time systems. GF20-0007-1, IBM, White Plains, 1971.

Martin, J. Design of Real-Time Computer Systems. Prentice-Hall, Englewood Cliffs, N. J., 1967, pp. 439-460.

Seaman, P. H., Lind, R. A., and Wilson, T. L. An analysis of auxiliary-storage activity. IBM Sys. J. 5, 3 (1966), pp. 158-170.

Stimler, S. Real-time Data-processing Systems. McGraw-Hill, New York, 1969, pp. 45-54, 59-66, 217-226.

Related Works - Drum and Fixed Head Disk

Abate, J. And Dubner, H. Optimizing the performance of drum-like storage. IEEE Trans. Comp. C-18, 11 (November 1969), pp. 992-997.

Burge, W. H. and Konheim, A. G. An accessing model. J. ACM 18, 3 (July 1971), pp. 400-404.

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

Denning, P. J. Queueing models for file memory operation. MAC-TR-21 (thesis), MIT, Proj. MAC, Cambridge, Mass., June 1965.

Manocha, T., Martin, W. L., and Stevens, K. W. Performance evaluation of direct access storage devices with a fixed head per track. Proc. AFIPS 1971 SJCC, Vol. 38, AFIPS Press, Montvale, N. J., pp. 309-317.

Weingarten, A. The Eschenbach drum scheme. Comm ACM 9, 7 (July 1966), pp. 509-512.

VITA

Personal History

Name:	Richard Roy Reisman
Date of Birth:	April 12, 1947
Place of Birth:	Morristown, New Jersey
Parents:	Julius and Sylvia

Education

Hanover Park Regional High School	Graduated 1964
Brown University Bachelor of Arts in Applied Mathematics	Graduated 1968
Lehigh University Candidate for Master of Science in Industrial Engineering	1970-1972

Professional Experience

Western Electric Company Newark, New Jersey Information Systems Staff Member	1968-1970
Western Electric Company Princeton, New Jersey Information Systems Staff Member	1970-1972

Professional Societies

Member of the Association for Computing Machinery