Queues of requests for access to auxiliary storage play a major role in every teleprocessing application. Assuming that access requests are randomly distributed, a queuing model is formulated; formulas are obtained for the mean and variance of the response-time distribution, as well as for the utilization factors of the access channel and the storage modules.

Samples of analytical and simulation results are given.

On teleprocessing system design

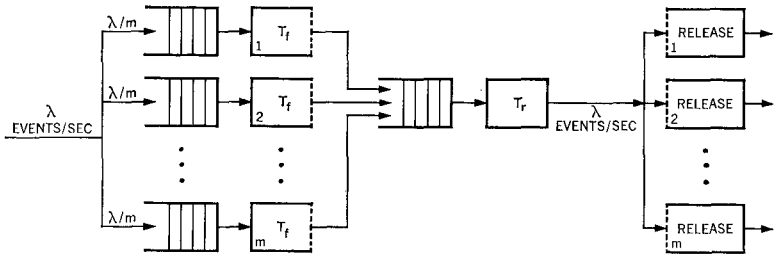
Part IV An analysis of auxiliary-storage activity

by P. H. Seaman, R. A. Lind, and T. L. Wilson

Teleprocessing applications can be expected to generate an appreciable flow of references to files, tables, program segments, and messages in various phases of processing. Since this activity may place a heavy load on the auxiliary-storage devices, it is important to determine the effect of this load on system performance. This problem is complicated by the fact that many variables can affect equipment utilization in many ways. As a result, no entirely satisfactory way of estimating the performance is known—short of a fairly elaborate simulation study.

An analytical approach to the file system can be useful, nevertheless. For example, if we make assumptions such as independent arrivals and uniform random accessing, we can formulate a mathematical model and solve for rough estimates of response times and equipment loads. Such a model can contribute to an understanding of the system in two ways. First, it provides quantitative estimates that accurately portray one possible set of operating conditions. Second, it draws attention to the areas in which operations can be controlled to improve upon the assumed operations. If operating conditions are intelligently controlled, it is safe to assume that the calculated estimates will be somewhat overconservative. Moreover, if simulation is subsequently employed to obtain more highly accurate estimates, the conservative estimates can serve to target the parameter region most deserving of simulation.

Figure 1 Schematic model of auxiliary-storage system



Following this line of reasoning, the paper first introduces a queuing model that applies to a typical auxiliary-storage system with direct-access file organization. Formulas are derived for the average system response time and for the utilization factors of the file modules and channel. Discussion then follows of modifications to this basic model necessary for estimating performance with an index sequential file organization. Finally, the paper concludes with a discussion of the most important controllable variables and ways for improving the file system performance.

A queuing analysis of a direct access system

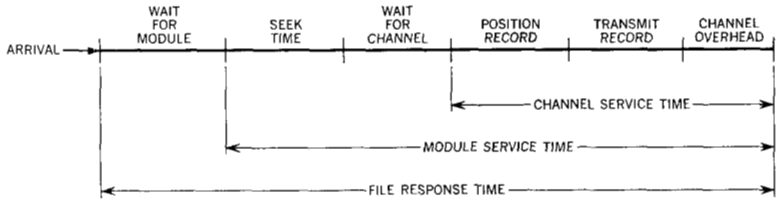
A file from which data can be retrieved directly using a given or generated address is said to have direct-access organization.

An idealized model[1] of such a file system is shown in Figure 1. The file is made up of m independent modules, such as drum or disk units. Every module is identical, with an *average seek time* T_f. Each module is fed by an independent module queue in which incoming file requests (events) wait if the module is busy with previous events. Each module feeds a single channel queue in which the processed events wait when the channel is busy transmitting previous items. The channel transmits processed events from the modules back to the requesting source with an *average service time* T_r. Modules are not released to process a new event until the old event has cleared through the channel. The average arrival rate of events processed through the system, i.e., *system throughput,* is denoted by λ.

The following assumptions are made: assumptions

- Events arrive at each of the m module queues in independent Poisson streams with a mean arrival rate of λ/m.
- The module queues are unlimited and are served in FIFO (first-in, first-out) order.
- Seeks may be initiated without the availability of the channel.
- The seek-time distribution of each module has a mean of T_f and a variance of σ_f^2.

Figure 2 Timing diagram for direct-access file



- Once a particular module arrives at the channel queue, it remains busy and cannot arrive again until it has been serviced by the channel. The channel queue may contain up to m units, which are served in FIFO order.[2]
- The service time of the channel is an exponentially distributed random variable with mean T_r and variance σ_r^2.
- No events are created or lost within the file system.

As a consequence of the last assumption, any point in the system (through which all the events in process must pass) experiences a mean rate of passage of λ events per unit of time. However, the distribution of traffic may change from point to point.

analysis

The main problem is to determine the mean response time of the file system. This is defined as the mean elapsed time between initiation of an event and receipt of an answer. Let T_q denote this interval, which will be called *mean file response time.*

After being accepted from its module queue, each event experiences a servicing delay until its answer is transmitted from the channel to the requesting source. The average time for this service, called *mean module service time,* is denoted by T_s.

The timing schematic of Figure 2 illustrates the relations among component times spent during the passage of an event through the file system.

The two relevant utilization factors pertain to the module and channel. Let ρ_m denote *module utilization,* which represents the proportion of time a given module is tied up servicing events. ρ_m is calculated by multiplying the mean time to service one event by the mean number of events to that module per unit time:

$$\rho_m = T_s\left(\frac{\lambda}{m}\right) \tag{1}$$

Let ρ_c denote *channel utilization,* the proportion of time the channel is tied up servicing events. This is calculated in a similar manner:

$$\rho_c = T_r\lambda \tag{2}$$

Since each module and module queue is independent, we may treat the system at this stage as m independent unlimited queue

systems, each with a mean arrival rate of λ/m and a mean service time T_s. It is well known that the mean response time for a single-server, unlimited queuing system is given by[3]

$$T_q = \frac{T_s}{1 - \rho_m}\left[1 - \frac{\rho_m}{2}\left(1 - \frac{\sigma_s^2}{T_s^2}\right)\right] \tag{3}$$

where σ_s^2 is the variance of the service time distribution.

Our problem now reduces to finding expressions for T_s and σ_s^2. To consider T_s first, note from Figure 2 that module service time consists of three parts: seek time within a module, wait time in the channel queue, and service time within the channel. Hence,

$$T_s = T_f + T_c + T_r \tag{4}$$

Although T_f and T_r are known or easily calculated, we must determine T_c, the wait time in the channel queue. It is generally true that the mean waiting time in queue for service is given by the mean length of the queue, L_c, divided by the mean arrival rate to that queue. The mean arrival rate to the channel queue is λ (due to the last of our assumptions). Hence,

$$T_c = L_c/\lambda \tag{5}$$

and it remains to determine the mean length of the channel queue.

Let us focus attention on the channel queue in Figure 1. From the standpoint of the channel queue, events from each of the individual modules appear to arrive at a mean rate, say w, provided that the module is not already in the queue—in which case its arrival rate is zero. The arriving events are assumed to be exponentially distributed. Thus, the overall arrival rate to the channel queue is $(m - n)w$, where n is the number of modules already in the queue, including the one being serviced by the channel. This is clearly an example similar to the "machine repair problem."[4]

Let P_n denote the probability that n modules are waiting in the channel queue (including the module being serviced). If we let z denote $(wT_r)^{-1}$, the steady-state equation governing this process is

$$(m - n)P_n = zP_{n+1} \tag{6}$$

The solution to (6) is given by

$$P_n = \frac{z^{m-n}}{(m - n)!}\left[\sum_{n=0}^{m}\frac{z^n}{n!}\right]^{-1} = \frac{e_{m-n}(z)}{E_m(z)} \tag{7}$$

where $e_n(z)$ and $E_m(z)$ are the individual and cumulative Poisson terms, respectively. That is,

$$e_n(z) = \frac{e^{-z}z^n}{n!} \quad \text{and} \quad E_m(z) = \sum_{n=0}^{m} e_n(z)$$

By definition, P_0 is the probability that no modules are waiting or in service; that is, that the channel is idle. Therefore, $(1 - P_0)$ must be the probability that the channel is busy. But this is the meaning of channel utilization. Therefore,

$$\rho_c = 1 - P_0 = 1 - \frac{e_m(z)}{E_m(z)} = \frac{E_{m-1}(z)}{E_m(z)} \tag{8}$$

However, ρ_c can be calculated from Equation 2, and therefore one can determine w by an iterative process on a computer, or from tables of the Poisson terms.[5]

The mean length of the channel queue is defined by

$$L_c = \sum_{n=1}^{m} (n - 1)P_n = m - z\rho_c - \rho_c = m - \lambda\left(\frac{1}{w} + T_r\right) \tag{9}$$

Finally, we substitute (9) into (5)

$$T_c = \frac{m}{\lambda} - T_r - \frac{1}{w} \tag{10}$$

and (10) into (4)

$$T_s = T_f + \frac{m}{\lambda} - \frac{1}{w} \tag{11}$$

Next consider the determination of σ_s^2. As in the case of (4), we may write

$$\sigma_s^2 = \sigma_f^2 + \sigma_c^2 + \sigma_r^2 \tag{12}$$

where the three terms on the right are the variances of the seek time, channel queue waiting time, and channel service time, respectively. The first and last of these can be calculated from the known distributions. The variance of the channel queue waiting time may be estimated from the formula for the variance of the mean queue length, assuming that the queue is ordered.

$$\sigma_L^2 = \sum_{n=1}^{m} (n - 1)^2 P_n - L_c^2 \tag{13}$$

Working the equation through, we arrive at

$$\sigma_c^2 \doteq \left(\frac{\sigma_L}{\lambda}\right)^2$$

$$\left(\frac{\sigma_L}{\lambda}\right)^2 = \frac{1}{\lambda}\left[(1 + z - \rho_c)T_r - (1 - \rho_c)(2 + z)\left(\frac{m}{\lambda} - \frac{1}{w}\right)\right] \tag{14}$$

sample calculation

To carry through a sample calculation, consider six modules of an IBM 2302 disk file connected through a channel to a computer and assume a throughput of 20 file events per second. Then $m = 6$ and $\lambda = 0.020$ events/msec. Assuming uniform random accessing, let the seek-time distribution for the disk module be the one given in Table 1. Then the weighted mean of the seek times is given by

$$T_f = (0.032)(50) + (0.164)(120) + (0.800)(180) = 165 \text{ msec}$$

We assume that records are stored randomly and that the basic record size is one-fifth of a disk track. If rotation time T_v is 33.3 msec., a single-record block is transmitted in 6.67 msec. Some records may require two continuous basic records, so that a double-record block is transmitted in 13.33 msec. We shall assume the record distribution shown in Table 2.

After the channel has selected a module from the channel queue, it must wait, on the average, one-half of a disk revolution to position the record. It then transmits the record (in time T_d). Channel control time or overhead (T_o) is assumed to require 1 msec. Because

$$T_r = \tfrac{1}{2}T_v + T_d + T_o$$

we have

$$T_r = 16.7 + (0.65 \times 6.67 + 0.35 \times 13.33) + 1.0$$

$$= 26.7 \text{ msec}$$

and

$$\rho_c = \lambda T_r = 0.020 \times 26.7 = 0.534$$

Determine w from Equation 8

$$0.534E_m(z) = E_{m-1}(z)$$

From tables,[5]

$$z = (26.7w)^{-1} = 9.6$$

and therefore w equals 0.00392 events/msec.

Remaining values can be calculated directly:

$$T_s = 165 + 300 - 256 = 209 \text{ msec} \qquad \text{(by Eq. 11)}$$

$$\rho_m = 209 \times 0.02/6 = 0.697 \qquad \text{(by Eq. 1)}$$

$$\sigma_c^2 = 50[(1 + 9.6 - 0.5)26.7 - 0.466(2 + 9.6)(300 - 256)]$$

$$= 1500 \qquad \text{(by Eq. 14)}$$

$$\sigma_f^2 = (0.032)(50)^2 + (0.164)(120)^2 + (0.800)(180)^2 - (165)^2$$

$$= 1044$$

$$\sigma_r^2 = \frac{33.3^2}{12} + (0.65)(6.67)^2 + (0.35)(13.33)^2 - 9^2 = 94$$

$$\sigma_s^2 = 1500 + 1044 + 94 = 2638 \qquad \text{(by Eq. 12)}$$

$$T_a = \frac{209}{1 - 0.697}\left[1 - \frac{0.7}{2}\left(1 - \frac{2638}{43500}\right)\right] = 463 \text{ msec} \qquad \text{(by Eq. 3)}$$

In many cases, one may desire a much simpler approximation for file response time—one that does not require w, thereby avoiding the necessity for tables. Such an approximation will now be described.

Table 1 Assumed seek-time distribution

Seek time (ms)	*Distribution*
0	0.004
50	0.032
120	0.164
180	0.800

Table 2 Assumed record distributions

Type	*Length (ms)*	*Distribution*
Single-record block	6.67	0.65
Double-record block	13.33	0.35

an approximation

Table 3 Simulation results

Case	ρ_c	ρ_m	T_s	T_q
	λ = 10 events/second			
1	0.267	0.330	198	252
2	0.267	0.328	197	251
3	0.274	0.343	197	253
	λ = 15 events/second			
1	0.400	0.507	203	315
2	0.400	0.505	202	319
3	0.402	0.498	199	315
	λ = 20 events/second			
1	0.534	0.697	209	463
2	0.534	0.697	209	490
3	0.552	0.697	202	467
	λ = 22 events/second			
1	0.588	0.781	213	618
2	0.588	0.781	213	660
3	0.605	0.759	202	548
	λ = 25 events/second			
1	0.668	0.907	218	1360
2	0.668	0.295	222	1900
3	0.676	0.844	203	812

If there were a large number of file modules, the arrival rate at the channel queue would be nearly independent of the length of the queue for reasonable channel utilizations. Thus the "machine service" model could be approximated by a single channel, unlimited queue model with response given as in Equation 3

$$T_c + T_r = \frac{T_r}{1 - \rho_c}\left[1 - \frac{\rho_c}{2}\left(1 - \frac{\sigma_r^2}{T_r^2}\right)\right] \tag{15}$$

This equation may be modified for smaller numbers of file modules by replacing the channel utilization ρ_c by a channel blocking factor D,

$$D = \frac{m - 1}{m}\,\rho_c \tag{16}$$

This represents the channel utilization due to all other modules except the one arriving. (Note: the analysis can be extended to systems with unequal traffic loads to each module by using this approximation.) In addition, the ratio of the variance to the mean

squared for channel service time is usually small and can be eliminated. This results in

$$T_c + T_r = \frac{T_r}{1 - D}\left[1 - \frac{D}{2}\right] \quad \text{or} \quad T_c = \frac{DT_r}{2(1 - D)} \tag{17}$$

A reasonable approximation for the variance of channel response time (assuming the response time to be exponentially distributed) is

$$\sigma_c^2 + \sigma_r^2 = (T_c + T_r)^2 \tag{18}$$

The approximate mean module service time now becomes

$$T_s = T_f + \frac{DT_r}{2(1 - D)} + T_r \tag{19}$$

and the module service variance is

$$\sigma_s^2 = \sigma_f^2 + (T_c + T_r)^2 \tag{20}$$

These values may be substituted into Equation 3 to determine mean file response time. (The ratio σ_s^2/T_s^2 is often small and can be omitted, simplifying the calculation even further.)

simulation check

In an effort to verify the accuracy of the solutions, the model was simulated for the system parameters given in the sample problem. The simulation results (shown in Table 3) are plotted as Case 3 in Figure 3. Case 1 shows the response time given by Equation 3 using the system service time given in Equation 11. Case 2 shows the response time using the approximate service time of Equation 19.

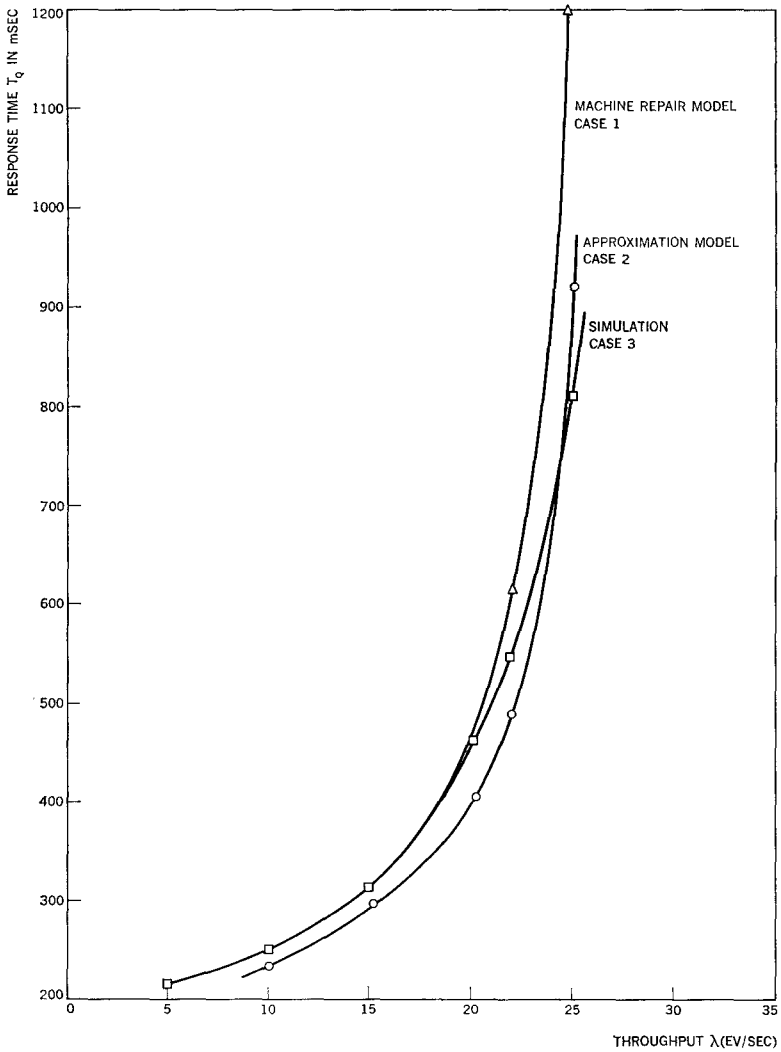
For values of ρ_m above 0.7, it is clear that the calculated response times for Case 1 begin to be appreciably higher than simulation. Assuming that the simulation accurately reflects the state of affairs, the following argument is offered in explanation.

In solving the "machine repair problem" for T_c, we assume that the channel service time is exponentially distributed. However, the true distribution actually has less variation, and events tend to wait less time in the channel queue. Thus, actual response times for the system can be expected to be somewhat lower than those calculated. Also, the distribution of the arrivals to the channel queue, with a mean of w, is not Poisson as assumed, but dependent upon the combined distributions the system arrival rate and the module seek time; this results in less variation than would be expected of the Poisson distribution, and therefore contributes to the shorter response time.

Modifications of the model

In the indexed sequential file organization based on the cylinder concept of auxiliary storage devices, data records are stored sequentially in increasing order on the contents of a specified key field. To facilitate random access, a cylinder index for each data set (there may be several per module) contains the key

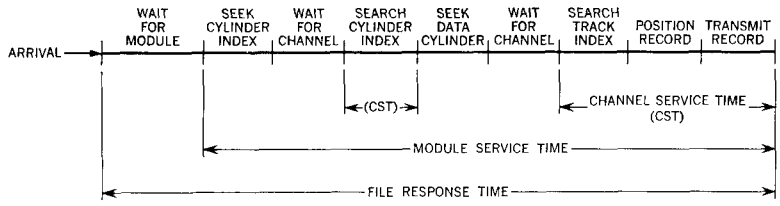
Figure 3 Estimated response times



of the last record stored in each cylinder of the module. (There are also master indices to the cylinder indices, but we are assuming these to be in main storage.) The cylinder index is usually stored on the module itself. In addition, each cylinder contains a track index; only a negligible head-switching time is required to reach the correct data track from this index.

To obtain a record from a file organized in such a manner, the steps shown in Figure 4 are required. Note that the timing diagram assumes that the file module is held throughout the operation, implying that the cylinder index is on the same module as the data referenced. If this is not the case, or if the module is released following the cylinder index search, the mathematical model of the previous section may be employed simply by doubling

Figure 4 Timing diagram for index sequential file



the traffic rate λ, since each file reference involves two independent accesses, one for an index and one for the data. By holding the module following the cylinder index search, a second module waiting-time is eliminated, thus reducing the overall file response time.

However, channel traffic is doubled whether or not the module is held; if the arrival rate of file requests is λ, the rate across the channel is 2λ. The channel service time, T_r, is the average of the two sequential services, a first that searches the cylinder index with service time T_{r1}, and a second that searches the track index and transmits the data record with service time T_{r2}. Thus,

$$T_r = \tfrac{1}{2}(T_{r1} + T_{r2})$$

For example, on a disk file, suppose that the cylinder index is allocated one track and the track index another track. Assuming that each index search reads half the index and that half a disk revolution is required to position the referenced data, we have

$$\begin{aligned} T_r &= \tfrac{1}{2}[(\tfrac{1}{2}T_v + T_o) + (\tfrac{1}{2}T_v + \tfrac{1}{2}T_v + T_d + T_o)] \\ &= \tfrac{3}{4}T_v + \tfrac{1}{2}T_d + T_o \end{aligned}$$

where T_v denotes disk revolution time, T_d denotes data transmission time, and T_o denotes channel control time.

Channel waiting time can then be approximated, as in Equation 17

$$T_c = \frac{DT_r}{2(1 - D)} \quad \text{where} \quad D = 2\left(\frac{m - 1}{m}\right)\lambda T_r$$

This again assumes traffic to all m modules is identical. If not, the channel blocking factor should include only that channel traffic due to modules other than the one in question.

Module service time is then given as

$$\begin{aligned} T_s &= T_{f1} + T_c + T_{r1} + T_{f2} + T_c + T_{r2} \\ &= T_{f1} + T_{f2} + 2(T_c + T_r) \end{aligned}$$

where T_{f1} and T_{f2} are the mean seek times to the cylinder index and data cylinder, respectively. It must be noted that some operations, such as a "write update," may not require a second search of the indices, but address information obtained from

previous reads can be used in seeking the referenced location. These operations should be properly weighted in the calculation for module service time. The variance of module service time may be approximated by

$$\sigma_r^2 = \sigma_{f1}^2 + \sigma_{f2}^2 + 2(T_c + T_r)^2$$

Mean file response time can then be found from Equation 3.

Remarks on operational considerations

There are usually several ways to improve file response time by changing the structural parameters of the system. Each of these ways seeks to achieve its end by modifying a different component of the response time. The most effective method or combination of methods to employ depends upon the parameters of the given system. In postulating a mathematical model, we assume that an overall file access rate λ is given. This rate, it should be noted, is partially under the control of the system designer. For example, if messages are buffered as segments in main storage and assembled or disassembled from auxiliary storage, λ is affected by the choice of segment length. The shorter the segment length, the higher the value of λ. However, the choice of the segment length also affects the amount of main storage required. The shorter the segment length, the less main storage needed for the buffer pool; the higher the value of λ, the more main storage needed for the access queue to auxiliary storage. To get reasonably close to a suitable balance, the system designer clearly must be prepared to perform a certain amount of iterative juggling of parameters.

Rather than decreasing the overall file access rate, the rate to each module may be decreased by adding more modules, thus distributing the traffic more widely. This is most effective if the channel utilization is low, and the module utilization is relatively high.

The mathematical model also assumes that any of the m storage modules are equally likely to be named in an access request. This is not a conservative assumption, i.e., the usual effect of gross violations of the assumption is to lengthen average response time. The system designer should keep this assumption in mind when allocating space for data sets. In some cases, it is well to divide a data set into m portions, with one portion residing on each module. This distributes accesses more evenly and results in greater device overlap.

Most file modules, other than storage drums, have large seek times as compared to the other system parameters. Thus, reduction of the mean seek time has a significant affect on the improvement of file operation. The obvious way to do this is to substitute a faster unit, but the improvement in speed must be balanced against increased cost. However, because faster speed results in less main storage being held by partially processed data waiting

for file action, the gain in main storage may be worth the extra cost. Another factor to be weighed when considering a faster unit is that engineering considerations may dictate a smaller storage capacity. If the capacity utilization is low, as with many message switching systems, the reduction may not be critical. However, if the utilization is high, the approach may not be feasible; then we must turn to other methods of reducing seek time.

For lack of specific data to the contrary, we usually assume that every location in auxiliary storage is equally likely to be accessed by a file request. Under such an assumption of uniform random accessing, each file unit has a characteristic mean seek time and variance. Any steps that the system designer can take to reduce the randomness of accesses usually reduces the average seek time.

Basically, there are two ways of reducing randomness. The first employs application knowledge of record usage in allocating space for records. For example, the most frequently used records may be stored as neighbors, or records that follow one another in algorithmic sequence may be stored together. The design of many storage devices makes it advantageous to store high-activity records near the center of the module. For example, in the indexed sequential mode, it is advantageous to store the cylinder index in the center of the module, thus reducing the maximum travel to the outermost data cylinder.

The second method is to sort the request queues for each module on addressed location. In this case, the queued requests are not handled in order of arrival, but in some order depending upon record address. In one such technique, the device arm is moved unidirectionally throughout its complete range—serving as many requests as possible—before returning to the initial location for another scan. Although this technique involves additional queue processing, it becomes very efficient (has a near-minimal mean) and very stable (has a low variance) at high throughput rates.

Where channel utilization is high, we speak of a "channel-bound" file system. Ways must be found to relieve the channel congestion in order to improve the file operation. One way to do this is to decrease channel throughput. As previously indicated, this may be accomplished by using a direct-access rather than an indexed sequential organization, if possible. Another possibility is to add another channel, sending half the throughput to each. This is an expensive solution, but is sometimes the only possibility with enough potential for improvement.

Any possible reduction in channel service time will also improve the performance of channel-bound file systems. Employment of faster devices being one means to this end, the remarks made for improved seek time apply here also. Another possibility may be to use shorter file records, but this often leads to an increased access rate that more than counteracts any gain achieved. A very

effective means of reducing channel service is to eliminate the rotational delay required for positioning the record prior to transmission. One method is to employ a buffer into which data can be read without engaging the channel; then, when the channel is ready for it, the buffer content is instantly available. This does not remove the rotational delay from the total module service time, but only from the channel service time.

Summary

To provide estimates of device utilizations, queue lengths, and response times, a queuing model for a hypothetical auxiliary-storage system is formulated and analyzed. The limitations of the model are emphasized, as they are important in application of the model. On the other hand, useful ways of extending the utility of the model are also discussed.

CITED REFERENCES AND FOOTNOTES

1. The given derivation is an extension of work once carried out by Professor J. Wolfowitz of Cornell University on a consulting basis for project SABRE.
2. In current channels, the channel queue is not FIFO ordered. Rather, the lowest-numbered device requesting service on the lowest-numbered control unit is chosen for service. Although this affects the distribution of waiting times in the queue, the mean waiting time is the same as if the queue were ordered.
3. D. Kendall, "Some problems in the theory of queues," *Journal of the Royal Statistics Society, Series B (Methodological)* **13**, No. 2, 155 (1951).
4. W. Feller, *An Introduction to Probability Theory and Its Applications*, John Wiley & Sons, New York, Volume I, Second edition, 416–418 (1957).
5. General Electric Company, *Tables of the Individual and Cumulative Terms of the Poisson Distribution*, D. Van Nostrand Company, Princeton, New Jersey (1962).