A computational procedure is derived analytically to evaluate the input/output buffer storage requirements in a data exchange.

Validity of the analysis is substantiated by comparing—in a typical instance—the analytical results with those obtained by simulation.

Storage requirements for a data exchange

by I. Delgalvis and G. Davison

One of the fundamental problems in designing a real-time system is to determine the storage requirements of the data exchange used to connect the communication network and central processor. The difficult part of the problem is to find the amount of data exchange storage needed for input/output buffer storage. In this paper, an analytical method is developed.

Exchange storage

exchange storage areas

Different exchange storage areas may be identified in terms of their usage, as shown in Figure 1.

- *Input/output storage* — containing blocks of storage assigned to the communication channels to receive or to send messages. Input/output storage may be subdivided into *buffer* and *output message queue.*
- *Permanent storage* — containing programs and tables throughout the entire operation.
- *Programming queue* — containing messages waiting in the exchange for the appropriate program to handle them.
- *Data channel queue* — containing messages waiting to be transmitted to the processor.
- *Auxiliary storage queue* — containing messages to be read from, or sent to, some external storage associated with the exchange.

In most systems, the program and table areas of the exchange are fixed, whereas the data channel and auxiliary storage queue areas vary with time and are always relatively small. The I/O buffer is for assembly of incoming messages and disassembly of outgoing messages. The output message queue is used to store messages that are to be transmitted over lines already in use. This queue is usually stored in an auxiliary storage, such as disk files or drums. Hence, the problem of finding the amount of exchange storage required is reduced to finding how much storage is needed for buffering.

buffer usage

In most buffers, the incoming and outgoing traffic moves in parallel—messages are received and sent over a particular line independently of the other lines. The character is used as a unit of measure for both message length and buffer size (one character of a message occupies one character of storage in the buffer). In the buffer, the messages come in over the lines, one character at a time, with the time between character arrivals being constant for a given line. When transmission is completed, the entire message is instantaneously transferred to the program queue. Similarly, messages to be sent come into the buffer area from the queue area instantaneously and go out, a character at a time, over the lines. Two messages cannot be assembled or disassembled at the same time; hence, if one wanted to provide the maximum area that would ever be used, one only has to find the maximum message length and multiply it by the number of lines that use the area. But if long messages rarely occur, as is usually the case, the exchange would be overdesigned and unduly expensive.

Buffer usage distribution

allocation of buffer storage

Buffer usage distribution is governed, in part, by the method used to allocate the storage. One may think of exchange storage as

Figure 1 Exchange storage areas

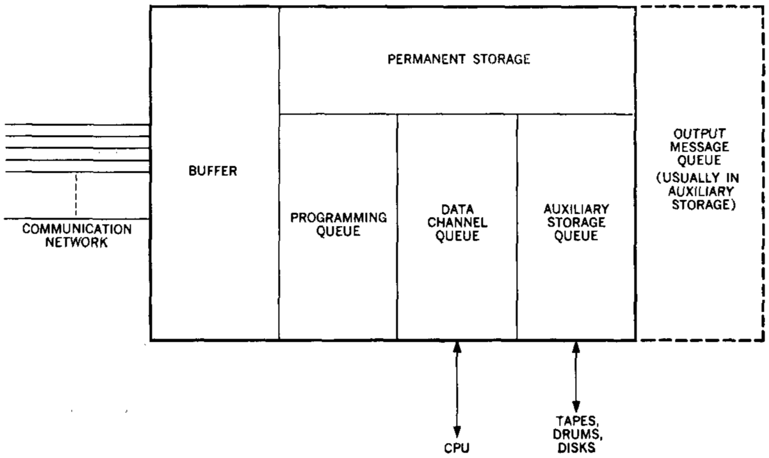


Figure 2 Method 1 buffer storage allocation

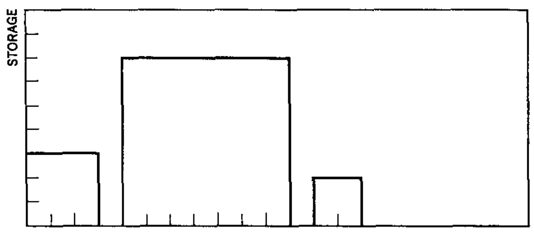
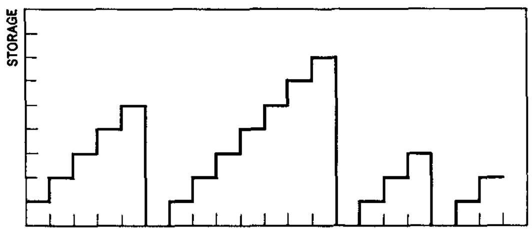


Figure 3 Method 2 buffer storage allocation



consisting of a common pool of character storage locations from which the buffer receives and returns the storage as needed. The common practice is to allocate to the incoming message a whole block of character storage locations at a time. Similarly, when a message is sent out, the storage is returned in blocks. The size of the blocks varies with application and the particular exchange. The two most common ways of allocating and retrieving storage are:

Method 1. As soon as a message is to be transmitted, a block of storage corresponding to the size of the message is taken from the pool and reserved for the message. All of this storage is held until the message has been transmitted and removed from the buffer. The process is represented graphically in Figure 2.

Method 2. A block of storage is assigned to a line when the exchange starts receiving a message from the line. If the message fills this block, another block of storage of the same size is taken from the pool and assigned to the line. This process continues until transmission of the message is completed, as shown in Figure 3.

message length, traffic rate

The next factors to be considered are the distribution of message lengths and the traffic rate. In most systems, the origin and length of message are only statistically determinate. Usually, enough is known about the messages to obtain:

- The probability distribution of the occurrence of message lengths. The notation $f(X)$ is used to denote the *probability of a message of length X occurring.*
- The rate at which messages originate.

Consequently, the probability that a line is sending or receiving a message at some random time t may be computed. This probability is commonly called *line utilization* and will be denoted by ρ.

The first problem to be solved is to find the probability of using k characters of storage at some random time t. For simplicity, first suppose the line is sending and receiving messages continuously ($\rho = 1$). Then, storage utilization with Method 1 and Method 2 allocation are shown in Figures 4 and 5, respectively.

Method 1 allocation

To find the distribution of storage usage for one line with Method 1 allocation of storage blocks, let:

Figure 4 Method 1 storage utilization

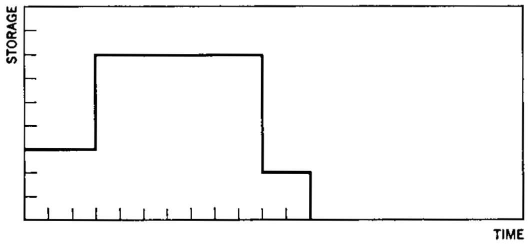
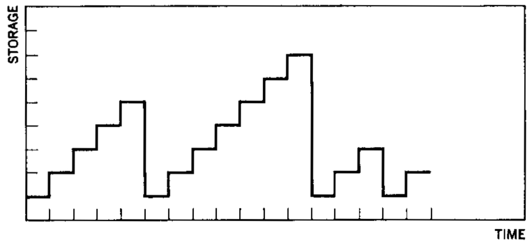


Figure 5 Method 2 storage utilization



T = a long interval of time,
N_T = number of messages that occur in T,
N_{XT} = number of messages during T of length X, and
C = time between successive character assembly.

Then, the total number of characters occurring in T is T/C, and the probability of a message being X characters long is given by

$$f(X) = \lim_{T\to\infty} (N_{XT}/N_T).$$

The number of characters of storage used by messages of length X during time T is XN_{XT}. From Figure 4, it is easily seen that this is the number of time units, C, during which X characters of storage are used. Hence, the relative frequency of using X characters of storage is

$$\frac{XN_{XT}}{T/C}.$$

The total number of characters occurring during T is the sum over all the message lengths of each message length times the frequency of that message length,

$$T/C = \sum_X XN_{XT}.$$

Therefore, the relative frequency of using X characters of storage is

$$\frac{XN_{XT}}{\sum_X XN_{XT}}.$$

By dividing the numerator and denominator of the latter quantity by N_T and then taking the limit over T, we obtain $g(X)$, the probability of using X characters of storage:

$$g(X) = \lim_{T\to\infty} \frac{XN_{XT}/N_T}{\sum_X XN_{XT}/N_T} = \frac{Xf(X)}{\sum_X Xf(X)},$$

the last equality having been obtained by noting that

$$\lim_{T\to\infty} \sum_X X \frac{N_{XT}}{N_T} = \sum_X X \lim_{T\to\infty} \frac{N_{XT}}{N_T}$$

holds, since messages of infinite length do not exist in reality.

Method 2 allocation

The distribution of storage usage for one line using the Method 2 allocation of storage to the incoming messages is found in a similar manner. First of all, let us assume that the block consists of one character. From Figure 5 we observe that M_{KT}, which denotes the number of times that K blocks are used during T, can be expressed as follows:

$$M_{1T} = N_{1T} + N_{2T} + N_{3T} + \cdots + N_{KT} + \cdots = \sum_X N_{XT},$$

$$M_{2T} = N_{2T} + N_{3T} + \cdots + N_{KT} + \cdots = \sum_{X>1} N_{XT},$$

$$M_{KT} = N_{KT} + \cdots = \sum_{X>K-1} N_{XT}.$$

Hence, the relative frequency of using K blocks during T is

$$\frac{\sum_{X>K-1} N_{XT}}{\sum_X XN_{XT}},$$

and the probability $g(K)$ of using K characters of storage is

$$g(K) = \frac{\sum_{X>K-1} f(X)}{\sum_X Xf(X)}.$$

Now, if the block size is greater than one character, say b characters, the problem becomes more complicated. The number of times that Y blocks are used in the interval T is

$$\sum_{X>(Y-1)b}^{bY} [X - (Y - 1)b]N_{XT} + \sum_{X>bY} bN_{XT},$$

and the relative frequency of using Y blocks is

$$\frac{\sum_{X>(Y-1)b}^{bY} [X - (Y - 1)b]N_{XT} + \sum_{X>Yb} bN_{XT}}{\sum_X XN_{XT}}.$$

Hence, $g(Y)$, the probability of using Y blocks, is

$$g(Y) = \frac{\sum_{X>(Y-1)b}^{bY} [X - (Y - 1)b]f(X) + b \sum_{X>bY} f(X)}{\sum_X Xf(X)}.$$

A similar development yields the following expression for Method 1 allocation when the block size, b, is greater than 1:

$$g(Y) = \frac{\sum_{X>(Y-1)b}^{bY} Xf(X)}{\sum_X Xf(X)}.$$

If the utilization is not 1, then

$$h(0) = 1 - \rho \qquad \text{and} \qquad h(Y) = \rho g(Y) \qquad \text{for } Y > 0,$$

where $h(Y)$ is the probability of using Y blocks of storage when the line utilization is ρ.

density functions

In summary, the density function $h(Y)$ for the buffer storage usage distribution for one line takes the following forms:

Method 1 allocation

If the block size b is equal to 1,

$$h(Y) = \begin{cases} 1 - \rho & \text{for } Y = 0 \\ \dfrac{\rho Y f(Y)}{\sum\limits_X X f(X)} & \text{for } Y > 0, \end{cases} \tag{1}$$

and if $b > 1$,

$$h(Y) = \begin{cases} 1 - \rho & \text{for } Y = 0 \\ \rho \dfrac{\sum\limits_{X > (Y-1)b}^{bY} X f(X)}{\sum\limits_X X f(X)} & \text{for } Y > 0. \end{cases} \tag{2}$$

Method 2 allocation

If $b = 1$,

$$h(Y) = \begin{cases} 1 - \rho & \text{for } Y = 0 \\ \rho \dfrac{\sum\limits_{X \geq Y} f(X)}{\sum\limits_X X f(X)} & \text{for } Y > 0, \end{cases} \tag{3}$$

and if $b > 1$,

$$h(Y) = \begin{cases} 1 - \rho & \text{for } Y = 0 \\ \rho \dfrac{\sum\limits_{X > (Y-1)b}^{bY} [X - (Y-1)b] f(X) + b \sum\limits_{X > bY} f(X)}{\sum\limits_X X f(X)} & \text{for } Y > 0. \end{cases} \tag{4}$$

numerical example

Figure 6 gives an example of message length distribution using blocks of characters. In this example, we assume a Method 2 allocation. Since the distribution is given in terms of whole blocks rather than in characters, there is no possibility of having a partially filled block. Therefore, we can use Equation (3) in terms of whole blocks instead of characters to calculate the probability of a single line using any given number of blocks at a random point in time. Thus, the probabilities of 0 and 1 blocks in use may be found, respectively, as follows:

$$h(0) = 1 - 1$$

$$h(1) = 1\left(\frac{0 + \frac{1}{4} + \frac{1}{4}}{1(0) + 2(\frac{1}{4}) + 3(\frac{1}{2}) + 4(\frac{1}{4})}\right) = \tfrac{1}{3}.$$

Similarly, $h(2) = \frac{1}{3}$, $h(3) = \frac{1}{4}$, and $h(4) = \frac{1}{12}$.

Figure 6 Illustrative example involving Method 2 allocation

Figure 6(a) Message length distribution

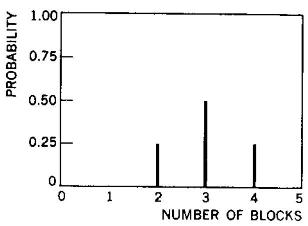


Figure 6(b) Buffer block usage during typical time interval

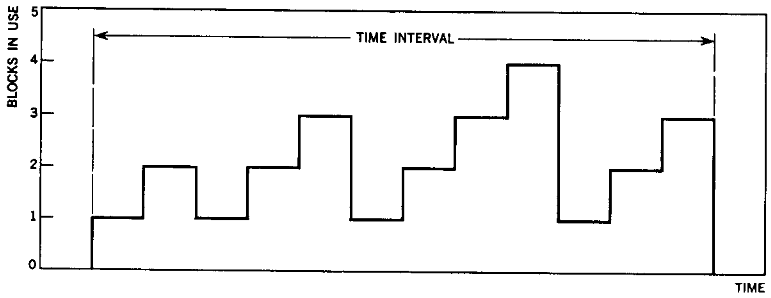
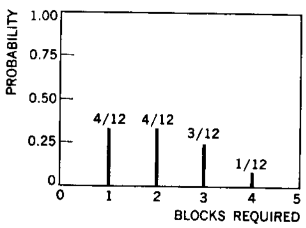


Figure 6(c) Buffer block usage at random time



Moments of buffer usage distribution

Rather than obtain the distribution of buffer storage usage for a message exchange with a number of input and output lines, we merely obtain the moments. This is a much simpler task, but the results are sufficient to effectively design the system.

If we denote the mth moment of $f(X)$ as $E(X^m)$, and the mth moment of $h(Y)$ as $E(Y^m)$, then the moments of $h(Y)$ are given as follows:

Moments, Method 1 allocation

$$E[Y^m] = \sum_Y Y^m h(Y) = \frac{\rho}{\sum_X Xf(X)} \sum_Y Y^{m+1} f(Y) = \rho \frac{E[X^{m+1}]}{E[X]},$$

so that in particular:

$$E[Y] = \rho \frac{E[X^2]}{E[X]},$$

$$E[Y^2] = \rho \frac{E[X^3]}{E[X]}, \tag{5}$$

$$E[Y^3] = \rho \frac{E[X^4]}{E[X]}.$$

Moments, Method 2 allocation

$$E[Y^m] = \sum_Y Y^m h(Y)$$

$$= \frac{\rho}{\sum_X Xf(X)} \sum_Y Y^m \Big(\sum_{X \geq Y} f(X)\Big) = \frac{\rho}{E[X]} \sum_Y f(Y) \left(\sum_{X=1}^{Y} X^m\right),$$

and by using the relations

$$\sum_{X=1}^{Y} X = \frac{Y(Y+1)}{2}, \qquad \sum_{X=1}^{Y} X^2 = \frac{Y(Y+1)(2Y+1)}{6},$$

and

$$\sum_{X=1}^{Y} X^3 = \frac{Y^2(Y+1)^2}{4},$$

formulas for the first three moments are obtained:

$$E[Y] = \frac{\rho}{E[X]}\left[\frac{E[X^2] + E[X]}{2}\right],$$

$$E[Y^2] = \frac{\rho}{E[X]}\left[\frac{2E[X^3] + 3E[X^2] + E[X]}{6}\right], \quad (6)$$

$$E[Y^3] = \frac{\rho}{E[X]}\left[\frac{E[X^4] + 2E[X^3] + E[X^2]}{4}\right].$$

Note that we have derived the moments of only those buffer storage usage distributions involving no partially filled blocks. This is appropriate, since the corresponding moments of the distributions are much easier to compute and the accompanying errors are relatively small, with the results being slightly high.

example

The first three moments of the message length distribution associated with data displayed in Figure 6a can be computed to obtain:

$$E(X) = 3.00, \qquad E(X^2) = 9.50, \qquad \text{and} \qquad E(X^3) = 31.50.$$

From these values, using (6), we may obtain the mean and variance of the number of blocks required to buffer a single line:

$$E[Y] = \frac{\rho}{E[X]}\left[\frac{E[X^2] + E[X]}{2}\right] = \frac{1}{3}\left[\frac{9.5 + 3.0}{2}\right] = 2.083,$$

$$E[Y^2] = \frac{\rho}{E[X]}\left[\frac{2E[X^3] + 3E[X^2] + E[X]}{6}\right]$$

$$= \frac{1}{3}\left[\frac{2(31.5) + 3(9.5) + 3.0}{6}\right] = 5.25,$$

$$\sigma_Y^2 = E[Y^2] - (E[Y])^2 = 5.25 - (2.083)^2 = 0.911.$$

It should be noted that these computations depend only on the moments—no other knowledge of the distribution is required. Equation (6) is exact only when the length of all messages is an integral number of blocks. When this condition does not exist, the exact mean and variance can be obtained from the values $h(Y)$ of the storage distribution. These values, in turn, are calculated from the message length distribution and utilization, using Equation (2) or Equation (4). However, the latter calculation requires knowledge of the entire message length distribution—not just of the moments.

exchange with N lines

Communication networks use three types of lines:

- *Simplex* — lines over which traffic moves in only one direction.
- *Full duplex* — lines over which traffic can move in both directions simultaneously.
- *Half duplex* — lines over which traffic can move in both directions, but only one direction at a time.

From the exchange point of view, each full duplex line actually consists of two simplex lines, one used for output and the other for input. First, we find the storage required by the N_I lines which are used for input. If Y_{iI} is the number of storage units used by Line i, the number of units used by all N_I lines is a new variable, Z_{NI}, where

$$Z_{NI} = Y_{1I} + Y_{2I} + \cdots + Y_{NI}.$$

Since Y_{iI} are independent, the mean and variance of Z_{NI} are the sums, respectively, of the means and variances of the Y_{iI}. If the message length distribution and the line utilization are the same for all the lines, the mean and variance of Z_{NI} for Method 1 allocation are:

$$E[Z_{NI}] = N_I \rho_I \frac{E[X^2]}{E[X]}$$

and

$$\sigma^2_{Z_{NI}} = N_I \rho_I \left[\frac{E[X^3]}{E[X]} - \rho_I \left(\frac{E[X^2]}{E[X]}\right)^2\right].$$

In the case of Method 2 allocation, the mean and variance of Z_{NI} are:

$$E[Z_{NI}] = N_I \rho_I \left(\frac{E[X^2] + E[X]}{2E[X]}\right)$$

and

$$\sigma^2_{Z_{NI}} = N_I \rho_I \left[\frac{2E[X^3] + 3E(X^2] + E[X]}{6E[X]} - \rho_I \left(\frac{E[X^2] + E[X]}{2E[X]}\right)^2\right].$$

Similarly, for N_O output lines (if Y_{iO} denotes the number of storage units used by line i, and Z_{NO} denotes the sum of the Y_{iO}), the corresponding equations for the mean and variance are obtained from the foregoing equations by replacing I by O.

A problem arises with half duplex lines when different methods of storage allocation are used for message assembly and disassembly. Let Z_I be the number of storage units used when a message is coming in over a half duplex line, and Z_O be the number of storage units used when a message is going out over the same line. The first two moments of the distribution of storage used for both assembling and disassembling the messages over this line are:

$$E[Z_{IO}] = \frac{\rho_I}{\rho_I + \rho_O} E[Z_I] + \frac{\rho_O}{\rho_I + \rho_O} E[Z_O],$$

$$E[Z^2_{IO}] = \frac{\rho_I}{\rho_I + \rho_O} E[Z^2_I] + \frac{\rho_O}{\rho_I + \rho_O} E[Z^2_O],$$

where

ρ_I = probability of assembly process taking place, and

ρ_O = probability of disassembly process taking place.

If N_H half duplex lines are hooked up to the exchange, the moments of the distribution of the number of storage units used for both assembly and disassembly of the messages over all N_H lines are:

$$E[Z_{N_H}] = N_H(E[Z_{IO}]),$$

$$E[Z^2_{N_H}] = N_H(E[Z^2_{IO}] - E^2[Z_{IO}]) + N^2_H(E[Z_{IO}])^2.$$

Let Z denote the total number of storage units used by all lines. Then

$$Z = Z_{NSO} + Z_{NSI} + Z_{N_H},$$

$$E[Z] = E[Z_{NSO}] + E[Z_{NSI}] + E[Z_{N_H}],$$

$$E[Z^2] = E[Z^2_{NSO}] + E[Z^2_{NSI}] + E[Z^2_{N_H}] + 2E[Z_{NSO}]E[Z_{NSI}] + 2E[Z_{NSO}]E[Z_{N_H}] + 2E[Z_{NSI}]E[Z_{N_H}],$$

where

Z_{NSO} = number of storage units used by outgoing simplex lines,

Z_{NSI} = number of storage units used by incoming simplex lines,

Z_{N_H} = number of storage units used by the half duplex lines.

Note that $E[V^2]$ can be obtained from σ^2_V from the relationship

$$E[V^2] = \sigma^2_V + (E[V])^2$$

for any variable V.

Design of the buffer area

general procedure

Very few systems can be designed to process all traffic 100 percent of the time. In most cases, the required buffer area would be excessive. Small infrequent delays are much less costly than the extra hardware that would be needed.

Hence, when designing a system, the percentage of time that the system should be able to process all traffic must be fixed before one tries to find the size of the system required.

The variable Z, being the sum of independent random variables, has a distribution which is approximately normal—the greater the number of lines, the better the approximation. Thus, it is possible to compute the amount of buffer storage to be provided, Z', so that the system overflows a specified percentage of the time. Z' is computed from the following expression:

$$Z' = E[Z] + h\sigma_Z,$$

where the value of h is chosen to permit overflow the specified percent of the time, the value of h being obtained from standard normal distribution tables. Thus, the amount of storage to provide adequate buffering of the lines for any given percentage of the time can be computed.

A technique often used to reduce the amount of buffer storage needed is to segment the messages and to buffer only one segment at a time. In this case, we have a new message length distribution, the segmented message length distribution, which governs the storage usage. If a small enough segment is chosen, the segmented message length distribution closely resembles a constant (equal to the length of the segment) distribution. Fairly good approximations can be obtained by using the constant distribution, thus simplifying the calculations.

line speed, message length

It is interesting to note that the amount of core storage needed for buffering and for the output message queue is independent of the line speed. High- and low-speed lines with the same message length distribution and utilization require exactly the same amount of I/O area core storage. Although many more messages come in and go out on the high-speed line, each message spends a proportionately shorter time in the I/O area. Thus, high-speed lines are more efficient, as far as core storage in the message exchange is concerned, because many more messages can be handled while using no more I/O area core storage than is required for a low-speed line.

Another important point to note is the effect of the message length on the buffer storage requirement. An increase in the length of a message increases not only the amount of core storage required to contain the message but also the length of time that the message must be stored, since the transmission time is proportional to the message length. At the same time, line utilization is also increased. The buffer storage requirement increases approximately with the square of the message length. In cases of core storage shortage, a significant storage saving may result from a change in message format that gives a decrease in message length, even if the decrease is only slight.

A check on the analysis

The analytical procedures developed in this paper have been tested empirically by studying a particular system involving an IBM 7740 data exchange.

The 7740 hardware uses Method 1 allocation with block sizes of 30 characters for the outgoing messages, and Method 2 allocation for the incoming messages. In addition, it is possible to connect a number of files to store messages waiting to be processed or sent. If all incoming and outgoing transmission lines are of the

Table 1 Comparisons of simulation results with the analytical results

Number of lines	*Line utilization*	*Simulated average*	*Analytical average*	*Simulated standard deviation*	*Analytical standard deviation*
1	0.5148	13.45	13.26	22.88	22.54
10	0.5128	131.84	132.09	70.23	71.25
20	0.5144	70.66	70.59	19.89	18.74

low-speed type, the output message queue is very large and, in most cases, stored on disk files. Hence, a fairly good estimate of the overall core storage requirements of the exchange can be obtained from results of this paper.

The core storage requirement of the system studied was determined both by the analytical methods of this paper and by simulation. The results obtained agreed very closely, as shown in Table 1.

The simulation was accomplished by means of the GPSS II simulator described in Reference 1. In Reference 2, the use of GPSS II in simulating a data processing system is discussed and suggests the general simulation technique employed in the present study.

ACKNOWLEDGMENT

The authors wish to thank Wei Chang, P. H. Seaman, and C. E. Skinner for their helpful suggestions in writing this paper.

CITED REFERENCES

1. R. Efron and G. Gordon, "A General Purpose Digital Simulator and Examples of Its Application: Part I—Description of the Simulator," *IBM Systems Journal* **3**, No. 1, 22 (1964).
2. C. R. Velasco, "A General Purpose Digital Simulator and Examples of Its Application: Part II—Simulation of a Telephone Intercept System," *IBM Systems Journal* **3**, No. 1, 35 (1964).