Copyright © 1974 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 53, No. 2, February 1974
Printed in U.S.A.

Performance Models of an Experimental Computer Communication Network

By J. F. HAYES

(Manuscript received March 12, 1973)

This paper reports the results of a performance study of an experimental computer communication network. The network is currently being designed and built in order to test concepts and techniques that may find future application. The network consists of synchronous digital transmission lines connected in loops to a Central Switch. User traffic enters the system through multiplexers connected to the synchronous lines. The Central Switch has the two-fold function of routing and controlling traffic.

Two multiplexing techniques were examined, Demand Multiplexing (DM) and Synchronous Time Division Multiplexing (STDM). In both techniques, user messages are blocked into fixed size packets, prior to multiplexing on the line. The synchronous line can carry these packets at a maximum rate of 4000 packet slots per second. In STDM each terminal is assigned a packet slot which recurs periodically. In contrast, for DM, packets are multiplexed on the line asynchronously into unoccupied packet slots. Alternative implementations of the DM technique were studied, one where each terminal transmits and receives at a maximum rate of 4000 packets per second and another where the maximum rate is 2000 packets per second.

As part of its message-handling function, the Central Switch buffers messages in transit. This allows User Terminals to transmit and receive messages with a degree of independence from one another. However, the terminals' strategy affects the amount of storage required in the Central Switch. In order to prevent the loss of information when there is insufficient buffering, there is a mechanism to inhibit traffic from User Terminals when the Central Switch buffer is near overflow. Due to this control of traffic, there is a relationship between the amount of data that flows through the switch and the amount of buffering in the switch.

Simulation results showed that there was little difference in delay performance between the two implementations of DM. However, an analysis *comparing DM and STDM showed a great difference in performance for all but the very heaviest line loadings. This difference increases as the number of terminals sharing the T1 line increases.*

Our study concentrated on two aspects of buffering in the Central Switch. We examined the relationship between throughput and the amount of storage available in the switch. The results of a simulation study showed that throughput can be quite high for all but minimal storage in the switch. Moreover, a strategy that dedicates buffers does quite well compared to common buffering. The second aspect of the study concentrated upon the User Terminal's strategy. Since each terminal acts independently, there may be strategies that make particularly high demands upon storage capacity in the Central Switch. An analysis showed that at the loadings where the system would be expected to operate, the user strategy in transmitting and receiving messages has little effect.

I. INTRODUCTION

An experimental computer communication network is currently being designed and built. The function of this network is to provide a flexible communication medium between computers, users, and peripheral devices. The network can accommodate sources with varying input-output rates and varying activity. Many of the components of the system employ techniques that are new. In order to gain insight into the operation of these components and thereby aid in design decisions, mathematical models were developed. The study of these models involved both analysis and simulation. The results are presented in the form of sets of curves.

The system under study consists of several T1 carrier lines,* configured as loops, connected to a Central Switch (see Fig. 1). The system is accessed through Terminal Interface Units (TIU) connected between User Terminals and the T1 line. In addition to forming an interface between the user and the T1 line, the TIU also does signaling. This signaling plays a role in switching calls and controlling the traffic flow.

There may be a wide variation of users accessing the system, ranging from Teletypes† to high-speed computer systems. The switch receives messages from all terminals and delivers messages to all addressed terminals so that any terminal in the system may communicate with

* The T1 carrier line is a digital synchronous short-haul transmission system operating at 1.544-Mb/s rate.

† Registered trademark of the Teletype Corporation.

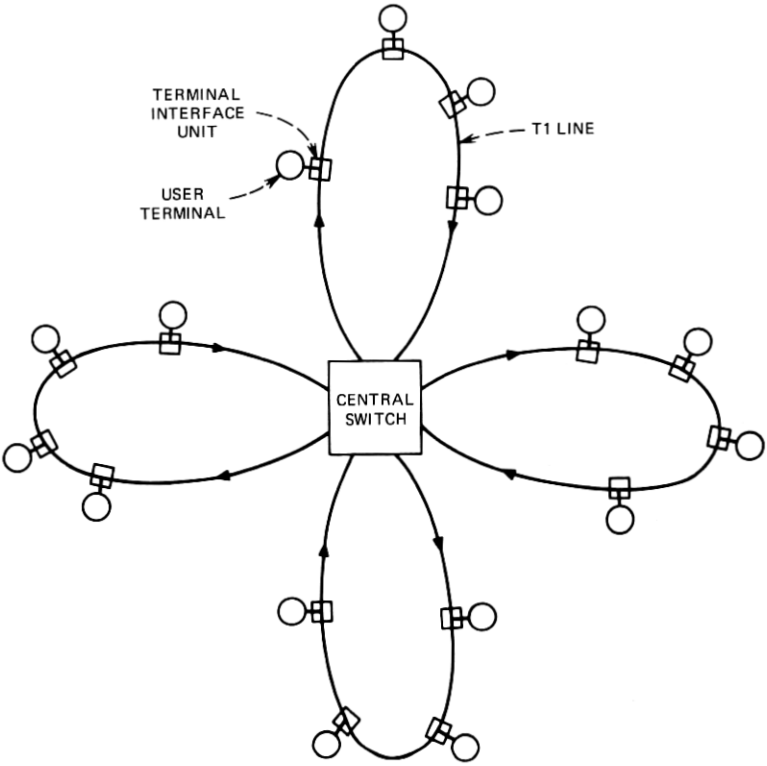

Fig. 1—Computer communication network.

any other terminal. All data pass through the switch even when two terminals are on the same T1 line.

The T1 line operates at a rate of 1.544 × 10^6 b/s. For purposes of synchronization and timing, the bit flow is divided into frames of 193 bits, with a flow of 8000 frames per second. The multiplexing arrangement in the system under study is such that a "network frame" consists of two adjacent T1 frames. Figure 2 indicates schematically how the 386 bits of the pair of T1 frames are allocated. The 50 bits required for framing and timing are part of the operation of the T1 carrier system. The assignment of the remaining 336 bits in the network frame is peculiar to the system under study.

User data are blocked into 256-bit packets and multiplexed on the line. Twenty-four bits of header information are attached to these information packets. (In the sequel we shall use the term "packet slot" in referring to this 280-bit block assigned to data and header.)

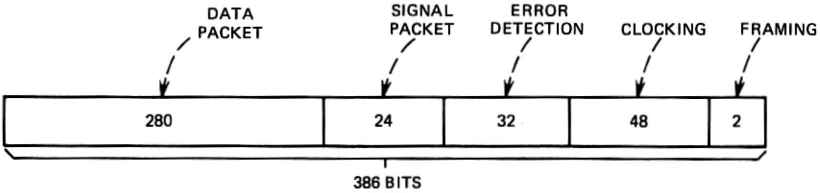

Fig. 2—Bit allocation of T1 frame pair.

Twenty-four of the 336 bits are used to carry signal packets. Signal packets convey control and routing information between the TIU and the switch. The remaining 32 bits of the frame pair are used for error detection in order to insure the integrity of the information in the signal and data packets.

From the foregoing we see that the information-carrying capability of the T1 loop is 4000 packets per second, each packet bearing 256 data bits, yielding a total information capacity of 1.024 Mb/s. There are many strategies that can be used to divide this capacity among the terminals connected to the loop. We shall evaluate the performance of two strategies, Synchronous Time Division Multiplexing and Demand Multiplexing. In Synchronous Time Division Multiplexing (STDM), each User Terminal is assigned a particular packet slot which recurs periodically. The terminal may multiplex data into its slot and receive data only in this same slot. For example, if there are ten terminals on the T1 loop and each terminal receives the same service, a particular terminal may multiplex packets on the line at a maximum rate of 400 packets per second. The time between packet multiplexing for these terminals is a constant 1/400 second.

In Demand Multiplexing (DM), packet slots are not assigned to a particular terminal. If a terminal has a packet to transmit to the switch, it inserts the packet into the first slot that is empty. Thus, unlike the STDM system, the flow of packets into the switch has no particular ordering as to originating terminal. So that the switch can sort packets according to their originating terminal, each packet has an address label in its 24-bit header. Similarly, information packets going from the switch to the terminal are not ordered and a header is required for each packet. Furthermore, each TIU must be able to recognize packets addressed to it. As we shall see, the number of bits required for addressing is relatively small.

Once a packet has been multiplexed on the loop either from the switch or from a terminal, it has priority over incoming traffic until it reaches its destination. A terminal must wait for an empty data packet slot before it can place a waiting packet onto the line. As in the STDM, the implementation also allows a terminal to place an outgoing packet into a slot from which it is removing an incoming packet.

We consider two implementations of the DM system, which correspond to the maximum speed at which terminals can transmit or receive. In the adjacent slot seizure implementation, terminals can transmit and receive at a 4000-packet-per-second rate. We consider an alternate implementation where the terminal is constrained to operate at a 2000-packet-per-second rate. In this case, a terminal can only write into or read from alternate packet slots.

A major component of the system is the Central Switch. The function of the switch is to route and control the flow of information. All messages generated at User Terminals pass through the switch where they are passed on to their destination terminals. Now the operation of the system (see Section V) is such that, as it may not be possible to deliver a message to its destination immediately, messages are temporarily stored in the switch. Also, destination terminals have some control over the way that these stored messages are read out of the switch's buffer.

The storage capability of the switch is not unlimited; therefore, the flow of information packets into the switch must be controlled. The switch does this by informing terminal TIU of the amount of storage currently available in the switch. The terminal does not transmit information packets when there is no room in the switch, but holds them until storage is available.

As mentioned earlier, models of the system were studied in order to gain insight into performance and thereby guide design decisions. The models studied are approximations to actual system operation. We felt that the study of more exact, hence more complicated, models would have involved far more time and effort, without a corresponding increase in insight.

We study the performance of multiplexing techniques on the loop as measured by message delay. In the switch we study packet storage requirements from two points of view, throughput and user strategy. Since the switch inhibits the flow of information when the storage in the switch is used up, there is a relationship between throughput and storage capacity. We also study the effect of different user readout strategies on switch storage requirements.

II. SUMMARY

As a guide to the reader, we pause to summarize the main body of the paper before plunging into details. In Section III, analytical and simulation approaches to the loop multiplexing problem are presented. Section IV is devoted to a discussion of our results on loop multiplexing. The relationship between storage capacity in the switch and throughput is considered in Section V. The results of a simulation study of capacity and throughput are presented in Section VI. In Section VII we consider the effect of a user's strategy on storage requirements in the switch. The results of this study are presented in Section VIII.

Although the analytical and simulation techniques used in our study are not restricted to a particular message distribution, we concentrated on the case where 30 percent of the messages are 32 packets long (8192 bits) and the remaining messages are one packet in duration (256 bits). This message distribution was our best guess at the actual distribution of messages in the system and reflects the fact that most terminals will, in fact, be computers. In the sequel we use the term variable message length to designate this distribution. The case where all messages are one packet in duration was also studied to some extent. In referring to this latter distribution we use the term constant message length.

The results of our studies of loop multiplexing are presented in Section IV. Simulation results for Demand Multiplexing indicate little difference in performance between alternate and adjacent slot seizure (see Figs. 6 and 7). The simulation was carried out for the variable message length distribution which consists of a large proportion of long messages. One would expect this distribution to be especially sensitive to the minimum time required to transmit and receive these long messages. In contrast, for the constant message length distribution, these maximum speeds, 2000 packets per second (alternate slot seizure) or 4000 packets per second (adjacent slot seizure), should have much less effect since the time to transmit a single packet is the same for both.

Analytical results show, not unexpectedly, that Demand Multiplexing yields better performance than Synchronous Time Division Multiplexing (see Fig. 8). Further, as the number of terminals served by a loop increases so also does the advantage of Demand Multiplexing (see Fig. 9). However, as the loading for a particular loop configuration increases, the difference in performance between DM and STDM decreases (see Fig. 8).

Results on information storage in the switch are presented in Sections VI and VIII. Our results indicate that, for all but minimum storage allocation, storage in the switch does not markedly affect throughput (see Figs. 12 and 13). The study also showed that, for the message distributions we considered, little is gained by dynamically allocating storage in the switch, as it is needed, holding nothing in reserve. Indeed, in certain circumstances, a static storage assignment does better simply because a static assignment insures reserves in the switch.

Our study showed that, for the loadings under which the system may be expected to operate, the effect of user strategy on switch storage requirements is not pronounced (see Figs. 14 and 15).

III. LOOP MULTIPLEXING

Two techniques for multiplexing data on the line are under consideration, Synchronous Time Division Multiplexing and Demand Multiplexing. In this section we shall present models designed to evaluate the performance of each of these techniques. These models are studied using both mathematical analysis and simulation. Simulation is necessary in situations where mathematical analysis is not possible.

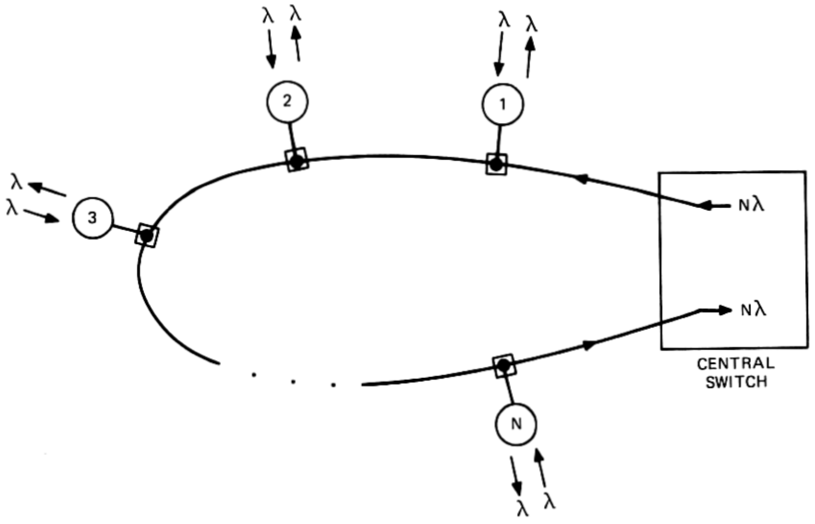

Fig. 3—Loop configuration.

A basic consideration in the design of the system is the response time to interactive users. An important component in this response time is message delay. We define message delay to be the time elapsing between the arrival of a message at a User Terminal and the departure of the last packet of the message from the terminal. Message delay is the criterion that we use to evaluate the multiplexing techniques under study.

We assumed that messages arrive at the User Terminals at a Poisson rate, and that the entire message arrives instantaneously. One would encounter this sort of behavior when a computer outputs directly from its memory to the loop, since the operation of the computer is at a much higher speed than the operation of the loop.

In both of the system configurations under study, the message delay may be broken up into two components, queuing delay and multiplexing delay. Recall that messages arrive at the station at a Poisson rate. Since it takes a nonzero amount of time to multiplex each message, there is a nonzero probability that when a message arrives at the station it must wait until previously arrived messages have been multiplexed onto the line. Once all prior messages have been multiplexed it takes additional time to multiplex the newly arrived message.

The problem of determining message delay for Synchronous Time Division Multiplexing and Demand Multiplexing with adjacent slot seizure is mathematically tractable. In order to study message delay in the DM case with alternate slot seizure, simulation is required.

A sketch of the loop model is shown in Fig. 3. N terminals are connected to the loop. The flow of data is unidirectional around the loop and is shown as counterclockwise in Fig. 3. The first terminal after the switch is labeled terminal number 1, the second terminal number 2, and so on. Messages arrive for multiplexing at a terminal at a rate of λ messages per second. We also assume that messages flow from the switch to each terminal at a rate of λ messages per second. The result is that the volume of traffic flow around the loop is symmetric. The total traffic flow from the switch to the terminals on the loop is $N\lambda$ messages per second. In the case of Demand Multiplexing this flow of return messages affects the operation of the loop. We shall ignore the interaction between the loop and the rest of the system by assuming that return message flow is Poisson. This assumption makes analysis possible and considerably simplifies simulation. We shall return to a consideration of this assumption after we present our models.

3.1 Synchronous time division multiplexing

As we have seen, the bit flow on the T1 loop is formated so that slots, into which information packets can be inserted, flow at a rate of 4000 per second. For Synchronous Time Division Multiplexing, each of these packet slots are assigned to particular terminals on a periodic basis. If there are N terminals connected to the T1 loop and each terminal is accorded equal treatment, then a packet slot is available every $T_c = NT_s$ seconds, where T_s is the duration of a packet slot. In the sequel we shall refer to T_c as the cycle time. For each terminal, we take the end of one cycle and the beginning of the next to be the end of the packet slot assigned to that terminal. We assume that a terminal may always write into its assigned slot even if it is simultaneously reading from the slot.

In order to develop an expression for the delay encountered by a message, let us assume that a message consisting of m_{L+1} packets arrives at a terminal whose buffer is empty, i.e., all previously arriving messages have been transmitted. If the message arrives w seconds before the end of a cycle, then a total of $w + (m_{L+1} - 1)T_c$ seconds elapse before the entire message is transmitted. Now if previous messages have not been transmitted, a newly arrived message suffers queuing delay as well as this multiplexing delay. For the purposes of analysis we categorize the packets of previously arrived messages into two classes: packets held over from previous cycles and packets that have arrived during the present cycle in the time interval $T_c - w$. We may write the total delay queuing and multiplexing as:

$$d_1 = qT_c + T_c \sum_{i=1}^{L} m_i + w + (m_{L+1} - 1)T_c . \tag{1}$$

In eq. (1), q is the number of packets remaining from previous cycles, L is the number of messages arriving in the interval $T_c - w$, and m_i is the number of packets in the ith of these L messages. The mean value of d_1 is shown in the appendix to be

$$\bar{d}_1 = T_c\left(\bar{m} - \frac{1}{2}\right) + \frac{\lambda T_c^2 \overline{m^2}}{2(1 - \lambda T_c \bar{m})}, \tag{2}$$

where $\bar{m}$ is the average message length in packets. Higher moments of d_1 can be found since, in eq. (1), the terms qT_c, $T_c \sum_{i=1}^{L} m_i + w$, and $(m_{L+1} - 1)T_c$ are independent random variables whose moment-generating functions can be calculated (see the appendix). Expressions for the moment-generating function of d_1 and for the mean-square value of d_1 are given in the appendix.*

3.2 *Demand multiplexing*

We now consider two implementations of Demand Multiplexing, adjacent slot seizure and alternate slot seizure. With adjacent slot seizure, a typical sequence of information slots leaving the switch might look as shown in Fig. 4. For purposes of explanation in Fig. 4 we assume messages are either three packets long or one packet long. The numbers in the slots correspond to the destination of the packet. No number in a slot indicates that it is empty. When the first three slots shown in Fig. 4 pass terminal 1, it is blocked. Terminal 1 may insert an information packet into slot 4 and into successive slots up to slot 10 when it is again blocked until slot 11. Terminal 1 also removes packets from slots 7 and 9. Terminal 2 removes the packets from slots 1, 2, and 3 and may insert information packets into these slots. Terminal 2 will be blocked when slots 10, 13, 15, 16, and 17 pass. Terminal 2 will also be blocked by terminal 1, if terminal 1 multiplexes packets onto the lines. The same rules apply to all of the other terminals on the loop. A terminal is free to insert data into empty slots and slots from which it removes data. Once an information packet has been inserted into an empty slot, it has priority over any other incoming packets.

Alternate slot seizure is similar except in one important respect. Since the terminal cannot receive nor transmit on adjacent information slots, it is limited to a maximum rate of 2000 packets per second. A typical flow of information is indicated in Fig. 5. Terminal 1 is blocked in slots 1, 3, and 5 but may transmit in slots 2 and 4. If terminal 1 begins by inserting a packet in slot 8, the next slot that is available to it is slot 11. These same rules apply to all of the other terminals in the loop.

The problem of message delay for Demand Multiplexing with adjacent slot seizure has received considerable attention recently.[2–4] Expressions for message delay that are relevant to our study can be found in Ref. 5. In order to make clear the assumptions that led to these expressions we shall sketch the analysis.

Basic assumptions of our study are that each terminal on the loop receives as much traffic as it transmits (see Fig. 3) and may write into slots containing packets addressed to it. Therefore, each

* A general analysis for which STDM is a special case is given in Ref. 1.

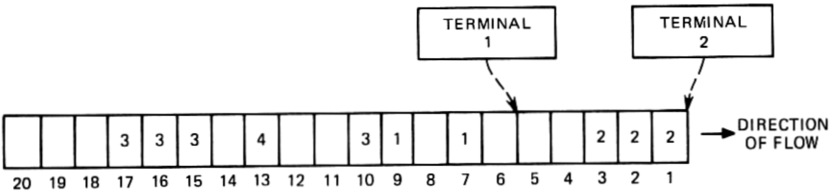

Fig. 4—Flow of data slots, adjacent slot seizure.

terminal "sees" the same traffic volume, $(N - 1)\lambda$ messages per second. The flow of traffic past each terminal consists of alternate busy and idle periods. Messages are multiplexed into line idle periods and are blocked by busy periods.

The probability distribution of message delay depends upon the distributions of the line busy and idle periods. In order to find these distributions, a basic assumption about the nature of the traffic flow out of the switch is necessary. We assume that the line busy and idle periods out of the switch are caused by the Poisson arrival of messages at the switch. This is not difficult to justify when there is light loading on the loops connected to the switch. Under light loading, messages arriving at User Terminals encounter little blocking and are conveyed immediately to the switch. Message arrival at the terminals is Poisson. For the message lengths and loadings we shall consider, the time between message arrivals is large compared to a slot time so that the discretization of message flow on the loop has little effect. As loading increases, the situation is less clear. However, messages arrive from all of the loops connected to the switch, tending to randomize message flow.

A line busy period at the output of the switch is initiated by a message arrival, which under the Poisson assumption is instantaneous. The busy period is lengthened if another message arrives while the previous message is being transmitted. The duration of a busy period is the same as the duration of the busy period of an M/G/1 queue

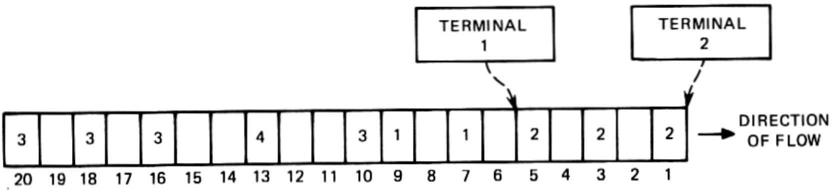

Fig. 5—Flow of data slots, alternate slot seizure.

for which results are well known.[6] Furthermore, under the Poisson assumption, the interval between message arrivals has a negative exponential distribution. Since each terminal sees $(N - 1)\lambda$ messages per second, the duration of a line idle period, as seen by a TIU, is negative exponential with mean $[(N - 1)\lambda]^{-1}$.

We assume that the statistics of line busy and idle periods are the same for all terminals on the loop. The removal of messages from the data stream may affect this assumption. Also, strictly speaking, these statistics also depend upon the multiplexing strategy of the switch. Further work involving simulation must be done to verify this assumption.

Messages are multiplexed packet-by-packet into empty slots. The multiplexing is interrupted by the advent of a busy period and is resumed when the busy period is over. Armed with the statistics of the line busy and idle periods, the message delay can be found. In the language of Queuing Theory, the model is that of a server that suffers periodic breakdowns. The results of the analysis appear as sets of curves that will be discussed presently.

From the foregoing we have an analytical approach for the calculation of delay in the case of Demand Multiplexing with adjacent slot seizure. There are inherent difficulties that preclude an analytical solution in the case of alternate slot seizure. The basic difficulty lies in calculating the durations of line busy and idle periods. For alternate slot seizure a terminal is blocked only if there are two or more interfering terminals. Thus a line idle period is terminated when a terminal begins transmitting, if there is at least one other terminal already transmitting.

Message delay in the case of alternate slot seizure was studied by means of simulation. The simulation program also could be used to study adjacent slot seizure. Although adjacent slot seizure can be analyzed, it was simulated primarily as a check.

In order to insure that the basic assumptions that underlie our model are understood, we outline the basic structure of the simulation program. The number of terminals on the loop is an input variable to the program. Input variables also determine the rate of message arrival, the length of long messages in packets, and the mixture of long and short messages. The simulation was carried out for the variable length message distribution. The basic time unit of the program corresponds to the duration of a packet slot, 1/4000 second. During each time unit a message may arrive to be multiplexed on the line. The random message arrival is simulated by comparing the output of a pseudorandom number generator to a threshold. If the test indicates that a message has arrived, the length of the message and the terminal to which it arrived are chosen randomly. A basic assumption here is that during a packet slot time no more than one message arrives at all N terminals. For the loadings of interest, this is not a restrictive assumption.

For each of the N terminals in the loop simulation, numbers are stored indicating the current number of packets and messages in the terminal buffer. In each basic time unit, terminal buffers $1, 2, \cdots, N$ are examined in succession until a nonempty buffer is found. If adjacent slot seizure is being simulated, a packet is removed from the first nonempty buffer. In the simulation of alternate slot seizure, a packet is removed from the first nonempty buffer from which a packet was not removed in the previous time unit. After either a packet has been removed from a buffer or all buffers have been examined, the program shifts to the next basic time unit and the cycle repeats beginning with message arrival.

Our interest is in the Nth terminal as it sees traffic from $N - 1$ other terminals. The line busy and idle periods seen by this terminal are measured. The program also measures the number of messages remaining in terminal N's buffer immediately after an entire message has been removed from the buffer. This measurement was not made every time a message departs, but periodically. The length of the period between measurements was varied from run to run. The reason for this is to guard against high correlation between measurements, thereby insuring independent samples.

The measurement of buffer contents upon message departure can be related to message delay. All of these messages remaining have arrived while the departing message was in the buffer. Since we know the rate of message arrival, we can estimate the delay or the length of time the departing message resided in the buffer. Estimates of the mean and the standard deviation of delay so derived will be shown in the next section of this paper.

IV. RESULTS OF LOOP STUDY

Results of the simulation and the analysis of the previous section are shown as curves which show the mean and the standard deviation of delay as a function of loading. From these curves we draw conclusions about the relative merits of the systems under study.

4.1 Demand multiplexing

On Fig. 6 are shown plots of average delay measured in the simulation as a function of loading for 5- and 20-terminal loops. The line loading is the portion of the time that the line is occupied. For comparison the results of the theoretical calculation of average delay is also shown in Fig. 6.

The simulation also yielded estimates of the standard deviation of message delay. Results are shown on Fig. 7, where the standard deviation of delay is shown as a function of loading. As in Fig. 6, the results of analysis for adjacent slot seizure calculations are shown.

In all of the curves the message distribution is the variable length message distribution defined earlier. The resulting average message length is 10.3 packets per message. The message arrival rate at each station in the loop in terms of loading, ρ, is $\rho/(10.3N)$ messages per slot time. (Each slot time is 1/4000 second.) Thus for 0.103 loading on a 20-terminal loop, messages arrive at a rate of 0.0005 message per slot time or 2 messages per second at each terminal.

For each of the simulations care was taken to insure that statistical

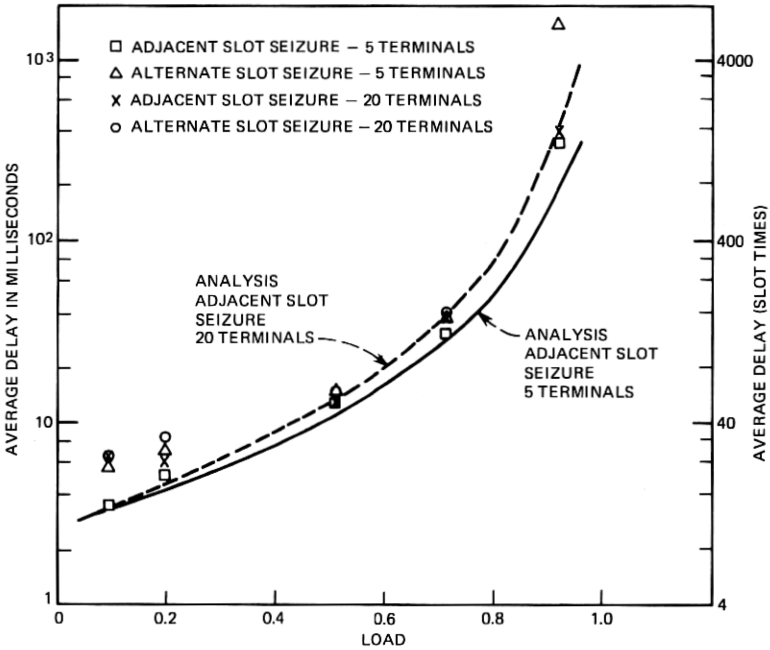

Fig. 6—Simulation results, average delay.

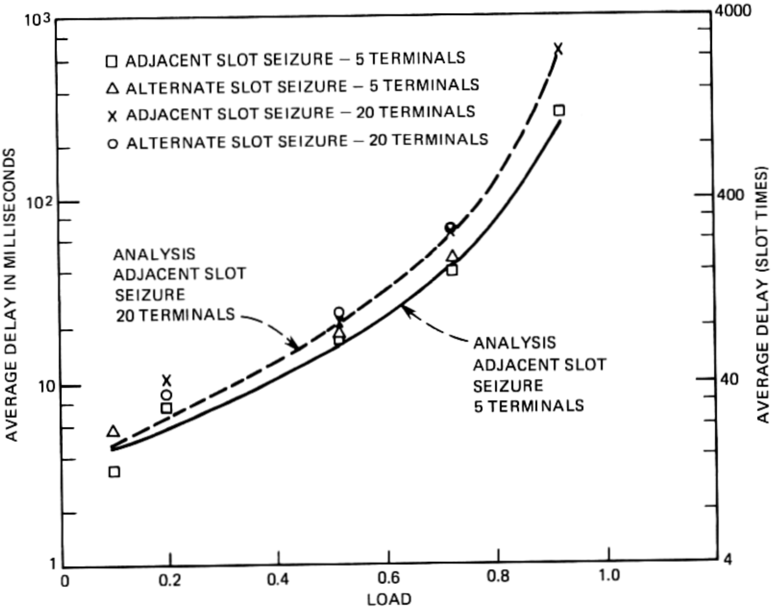

Fig. 7—Simulation results, standard deviation of delay.

equilibrium had been reached. The duration of runs and the random sequences used in the simulation were varied. The standard deviation of the estimates of the mean values of delay shown on Fig. 6 can be estimated. We assume that the measured standard deviations are the true standard deviations. The standard deviation of the mean is then the measured standard deviation divided by the square root of the number of samples. The results indicate that the standard deviation of the mean is small compared to the mean value. The standard deviation is largest relative to the mean at light loadings on the 20-terminal loop where it is approximately 5 percent of the mean.

There is a basic difficulty in making measurements at light loadings on a 20-terminal loop. Due to the relatively low departure rate, fewer independent samples can be gathered. Except for the lighter values of line loading on the 20-terminal loop there is good correspondence between the results of simulation and theory. Even at these lighter loadings the results of simulation are not so far from theory as to cast doubt on the simulation.

The simulation results show that adjacent slot seizure yields somewhat better delay performance than alternate slot seizure. For almost all values of line loading, adjacent slot seizure gives lower values of mean delay and standard deviation of delay. Moreover, measurements made on loops with 2, 10, and 64 terminals, not shown here, yield much the same result.

The reader will notice, however, that for most values of line loading, the difference between alternate and adjacent slot seizure is not large. The difference is small enough so that ease of implementation should probably determine the choice between the two.

Because of pressures of time we did not compute message delay distributions. However, estimates of the distribution of message delay can be calculated from the simulation values of mean and standard deviation. From the Tchebychev inequality we have

$$P_r[\text{delay} \geqq \mu + k\sigma] \leqq 1/k^2,$$

where μ is the mean and σ the standard deviation of the delay. On a 20-terminal loop with alternate slot seizure this inequality shows that at 0.515 loading 90 percent of the messages suffer delays of less than 83 milliseconds. However, the Tchebychev inequality often gives a rather loose bound. Under a rather tenuous assumption about the distribution of delay, one obtains a more optimistic result. If we compare means and standard deviations of delay we see that, for the same line loadings, the means and the standard deviations are roughly the same. For an exponentially distributed random variable the mean and the standard deviation are equal. If we assume that delay is exponentially distributed we find

$$P_r[\text{delay} \geqq x] = e^{-\alpha x},$$

where $1/\alpha$ is the standard deviation. For 0.515 loading on a 20-terminal loop, 90 percent of the messages have delay less than 50.5 milliseconds.

4.2 Comparison of multiplexing techniques

A second phase of our work on loop multiplexing was devoted to a comparison of Synchronous Time Division Multiplexing and Demand Multiplexing. In order to simplify analysis we have ignored the fact that for DM information packets must contain the address of the transmitting terminal. No such addressing is required for STDM. However, such addressing information is a negligible part of an information packet. For example, for a 64-terminal loop, only 6 bits are necessary to specify the address of a terminal. We feel that the small improvement in accuracy that could be attained by considering addressing did not justify the complications introduced into the analysis.

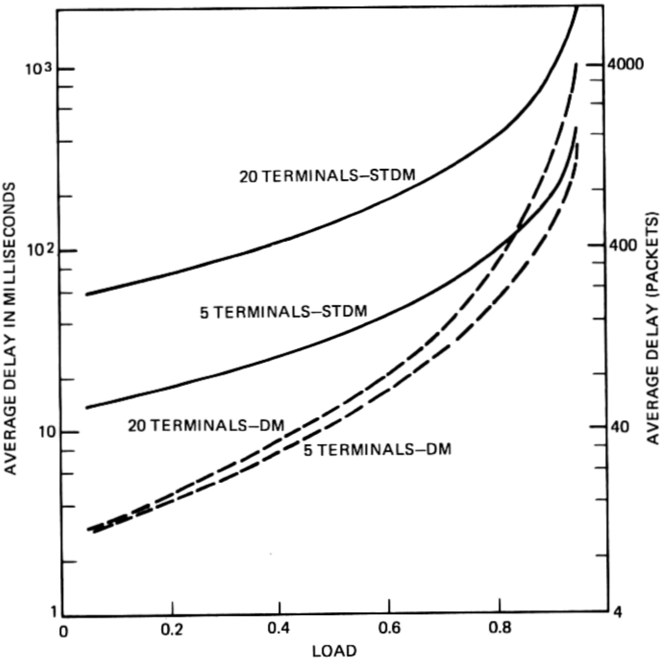

Fig. 8—Average delay versus loading in DM and STDM (30 percent of messages are 32 packets long).

Typical results are shown on Fig. 8 where delay in both packet times and milliseconds is shown as a function of line loading for the variable length message case. As the curves show, DM is clearly superior to STDM. This superiority is more pronounced at the lighter loading, where the multiplexing time is the strongest component of delay. In the absence of interfering traffic, the time required to multiplex a message in DM is an average of 10.3 slot times. In contrast, for an STDM system with N terminals, the average time required to multiplex a message is 10.3 × N slot times. As the loading increases, the difference between the two systems decreases. Line traffic in the DM system interferes with message multiplexing and as the load increases so does the interference.

Similar results have been obtained for the constant length message distribution. Computations of the standard deviation of delay for both constant and variable length message distributions also show the same basic pattern.

Another view of the performance is indicated on Fig. 9 where average delay is shown as a function of the number of terminals in the loop

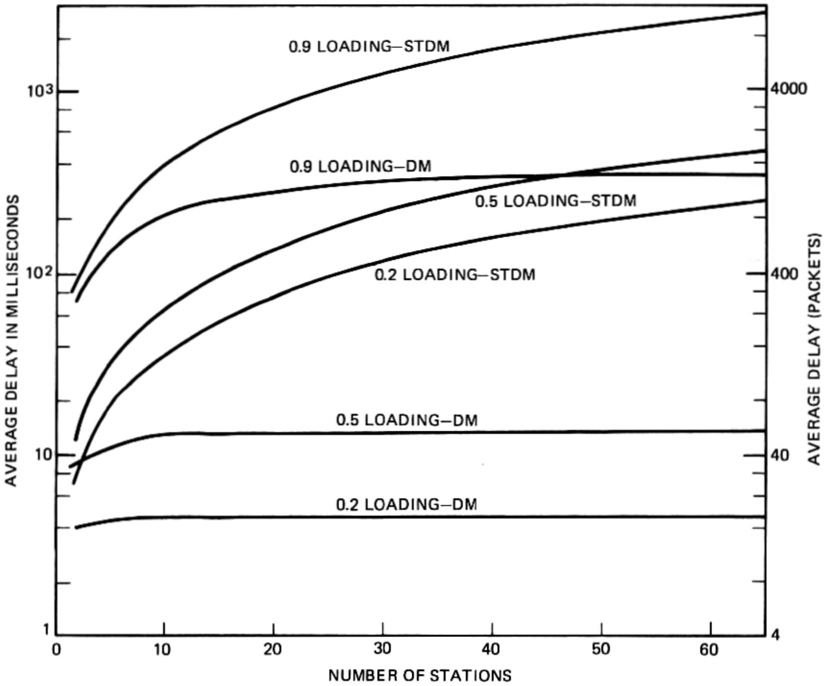

Fig. 9—Message delay versus number of stations (30 percent of messages are 32 packets long).

for fixed values of loading. In Fig. 9 the dependence of delay in the STDM system on the number of terminals in the loop is marked. There is little of this dependence in the case of DM. However, DM is more sensitive to changes in load than STDM. Notice the large jump in delay from 0.5 loading to 0.9 loading in the case of DM. Although we have not shown them, similar results obtain for the constant length message distribution.

V. SWITCH THROUGHPUT

The second phase of our work involved a study of buffering in the switch. Streams of data enter the switch from the loops connected to it. In the DM implementation entering packets are labeled as to originating terminal. In the STDM implementation each terminal has an assigned packet slot recurring periodically. System operation is such that, at any given time, each terminal in the system transmits to and receives from only one other terminal in the system. Information on which pairs of terminals are linked together is stored in the switch.

Therefore, given the origin of an information packet, the switch determines its destination by looking in a table.

A terminal can rapidly change the destination of the packets that it transmits. Stored in the Central Switch is a list of up to 64 possible correspondent terminals for each terminal. A terminal that is transmitting to terminal A, for example, may select a new destination, say terminal B. By means of signal packets (see Section I) the Central Switch is notified of this change in destination. After the change all information packets transmitted from the originating terminal are routed to terminal B. A terminal can select only from the list of its 64 correspondent terminals stored in the switch. However, this list can be altered by the originating terminal when it wishes to make connection with a new terminal or drop connection with an old. Again, signal packets are used to communicate between the terminal and the switch. The process of altering the list requires much more time then switching between terminals already on the list.

At a given instant of time, a terminal transmits to and receives from the same terminal. Further, each terminal in the system acts independently in selecting the correspondent terminal that is the destination of its packets. Thus a terminal may select a destination terminal that is, at that point in time, corresponding with a third terminal. In this event the packets that are transmitted are stored temporarily in the Central Switch. The Central Switch, again using signal packets, notifies the destination terminal that packets from a particular originating terminal are waiting to be delivered. It may happen that, for a particularly busy terminal, there may be messages from several different originating terminals stored in the switch waiting to be delivered. The receiving terminal is free to choose the order in which these messages are read out of the switch buffer.

In connection with this routing and selection procedure we use the term virtual channel as a notational shorthand. As we have seen, each User Terminal has stored in the switch a list of as many as 64 correspondent terminals. When a terminal selects the ith correspondent on this list, we say that the terminal selects the ith virtual channel. When we say that a terminal transmits and receives over virtual channel i we mean that the terminal transmits to and receives from the ith correspondent terminal on the list stored in the Central Switch.

As we have indicated, it may be necessary to store information packets in the switch before they can be delivered. As a practical necessity, the amount of storage in the switch is finite and under heavy loading conditions storage may be used up. In this situation the switch sends signal packets to User Terminals which inhibit transmission until storage is available in the switch.

Our study of packet storage capacity in the switch focused on two aspects of the problem, throughput and user strategy. Given the random nature of the message flow in the system, there will be occasions when all of the storage assigned to a channel is used up and the transmitting terminal is inhibited. If this condition occurs often enough, there will be a significant effect on the total throughput of data. Secondly, the user through his virtual channel selection strategy can affect the amount of storage that is required in the Central Switch. As we have noted earlier, a certain amount of time is required for a User Terminal to switch from one virtual channel to another. During this switching time, the terminal cannot read packets out of the switch. If User Terminals pursue a strategy calling for frequent switches, demands on switch storage may be too large.

In order to study throughput and the effect of user strategy on buffer requirements, a simplified model was constructed. The model is shown in Fig. 10. N independent data streams carrying λ messages per second flow into N buffers. These data streams represent traffic from correspondent terminals flowing over different virtual channels to the same destination terminal. The destination terminal's changing of virtual channels is represented by the switch in Fig. 10 moving from buffer to buffer. In the model the time required to switch buffers is taken to be either zero or eight packet times (1/4000 second).

In our study of switch throughput, two kinds of buffering were considered, dedicated and common. For dedicated buffering, each of the N buffers is a fixed size. When a buffer is filled, the transmitting terminal is informed and information packets are held at the User Terminal until there is room. In the case of common buffering, a fixed amount of storage is allocated for all N buffers. Each input line uses as much input capacity as it needs. Thus one input line can use up all of the common storage. Again, when there is no more room in the buffer, data flow is inhibited. In our study of throughput, we assume that the entire contents of a buffer are removed before moving on to a new buffer. The order in which buffers are examined is fixed and empty buffers skipped. In a later section we compare this to a strategy where switching takes place after a single message has been read out of a buffer.

A good deal of previous work on buffer occupancy has been based on a Poisson arrival model for messages in the input data stream. In the Poisson model, messages arrive instantaneously with an ex-

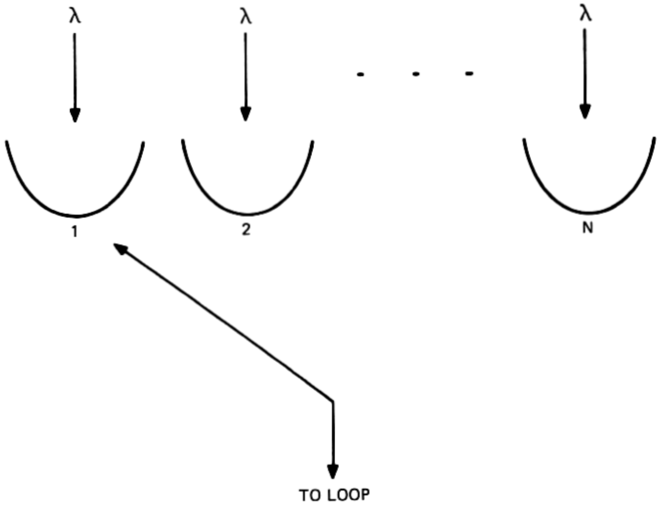

Fig. 10—Model of Central Switch.

ponentially distributed interval of time between messages. A more realistic model, for our study of throughput, is one in which messages arrive over a time interval proportional to the message length with the time between the beginning of one message and the end of the previous message being exponentially distributed. This latter model is more appropriate to buffering in the switch where the arrival and departure of messages is over T1 lines and the read-in and write-out rate of messages is the same. In Section VIII, results based on the continuous arrival model will be compared with the Poisson arrival model.

The model used in our study is indeed something of a simplification. Because terminals share the same T1 loop, messages are likely, especially in heavy loading, to be broken up when they are multiplexed. In our model we do not take this effect into account. For example, in our model a message with 32 packets would occupy 32 successive packet slots on the input line. In the actual system, there may well be gaps in these messages. Similarly, we assume that messages going to a particular destination terminal have sole access to the T1 line, when in fact the line is shared. Thus we assume that messages can be read out of buffers at will. Fortunately, these two effects tend to cancel out. In our model we read in faster and read out faster than reality. Also, we are not looking at absolute measures of performance, but are comparing different implementations. We felt that a more complicated input output model would not improve this comparison significantly.

Even this simplified model was not amenable to analysis and a Monte Carlo simulation program was written. The basic functions of the program is to measure the throughput as a function of storage capacity and to measure the average occupancy of the buffers. Input variables to the program determine the amount of storage available, the number of input lines and buffers, the time required to switch between buffers, and the probability of message arrival.

As in the program for loop multiplexing, each cycle of the program represents a packet slot time (1/4000 second). In each cycle, the program runs through three distinct parts of the program: input, output, and measurement. In the input portion, each input line is examined in turn. If the line is free, i.e., no message currently being delivered, a random test is performed. This test corresponds to the arrival of a message at a User Terminal in the system under study. If a message has arrived, another test determines its length. If either a new message has arrived or a message is already on the line, the contents of the line's buffer is checked. A packet is inserted in the buffer only if there is room. This packet insertion in the simulation program corresponds to a User Terminal transmitting a packet to the switch. If a line's buffer is full, the input process is suspended. This corresponds to storing a message or part of a message at a User Terminal. Until all of a previously generated message is fed into a buffer, no new messages can arrive.

The program is easily changed to handle either common or dedicated buffering. For common buffering, an input variable is the total storage available. When a packet is inserted in a buffer, this number is reduced by one. For dedicated storage, the storage for each line's buffer is an input variable. As a packet is inserted in a buffer, the amount of storage available for that buffer is reduced by one.

There is a simple relationship between the probability of message arrival and load. Let p be the probability that a message is generated in a slot. It can be shown that the portion of the slots on each input line that are busy is given by the expression

$$\ell = \frac{p\bar{m}}{p\bar{m} + 1 - p},$$

where $\bar{m}$ is the mean length of a message in packets. The assumption here is that there is no limitation on the content of the buffer into which the input line feeds. N input lines feed into the buffers, conse-

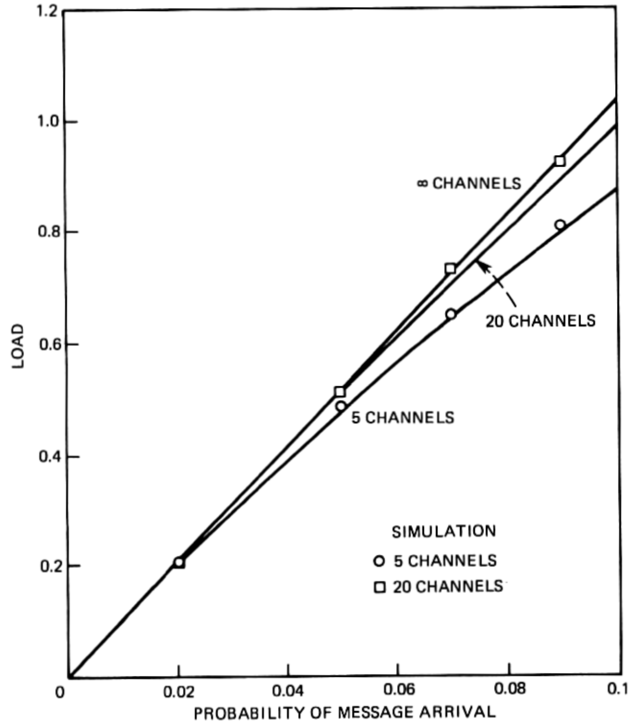

Fig. 11—Line load as a function of the probability of message arrival (30 percent of messages are 32 packets long).

quently the maximum occupancy of the line carrying messages out of the buffers is

$$L = \frac{pN\bar{m}}{p\bar{m} + 1 - p}. \tag{3}$$

The output line will attain this maximum occupancy if no time is required to switch between buffers. The relationship between loading, L, and the quantity pN, which we designate as the probability of message arrival, given in eq. (3) is plotted in Fig. 11.

In the output portion of the program, packets are removed from buffers and placed on the output line. This corresponds to a User Terminal receiving packets that have been stored in the switch. The program examines each of the N buffers in fixed order. Empty buffers are skipped and all of the contents of nonempty buffers are removed. One of the input variables to the program is the time required to switch between nonempty buffers. This corresponds to the time required by a User Terminal to select a new virtual channel. When a packet is removed from a buffer, the amount of storage available is increased by one up to some fixed amount.

In successive simulation runs the amount of packet storage available was varied with all other parameters held constant. For very large amounts of storage, there is always room in the buffers. As the amount of storage is decreased, it is increasingly likely that packet flow from an input line is inhibited. For relatively low amounts of storage, it will often happen that there is no room in the buffer. In this case, the flow of messages will be halted frequently and the number of packets flowing into the buffers per unit time will be reduced.

In the measurement portion of the program, the main focus was on throughput. Programs were run for 20,000 cycles, and the total number of packets that were fed into buffers were measured. By varying the total amount of storage available, with all other parameters fixed, one obtains the relationship between throughput and storage. Simulation runs were made for the constant and variable length message distributions. Measurements were also made of the total number of messages in the buffers. The results of these latter measurements will be considered in a later section dealing with user strategy.

VI. RESULTS OF SWITCH THROUGHPUT STUDY

Typical results of simulation are shown on Figs. 12 and 13 for 5 and 20 input lines, respectively. In obtaining the results shown on both figures the variable length message distribution was used. The switching time is 8 packet slots. If the line rate is 4000 packets per second, the time required to switch is 2 milliseconds. The curves show normalized throughput as a function of the total packet storage with message arrival probability as a parameter. For each loading the throughput is normalized to the throughput measured at very large storage capacity.

The basic configuration of the curves is as one might expect. As the storage capacity decreases, the throughput decreases. Further, the normalized throughput decreases faster for the larger values of loading.

The results show that, even for a limited amount of storage, the throughput is high. For example, if there are two packet slots for each buffer, the throughput is over 70 percent even for high loading. Results (not shown) for the case of zero switching time show that for this same amount of storage the throughput is over 90 percent.

A surprising result shown on Figs. 12 and 13 is that dedicated storage shows better performance than common storage when there is

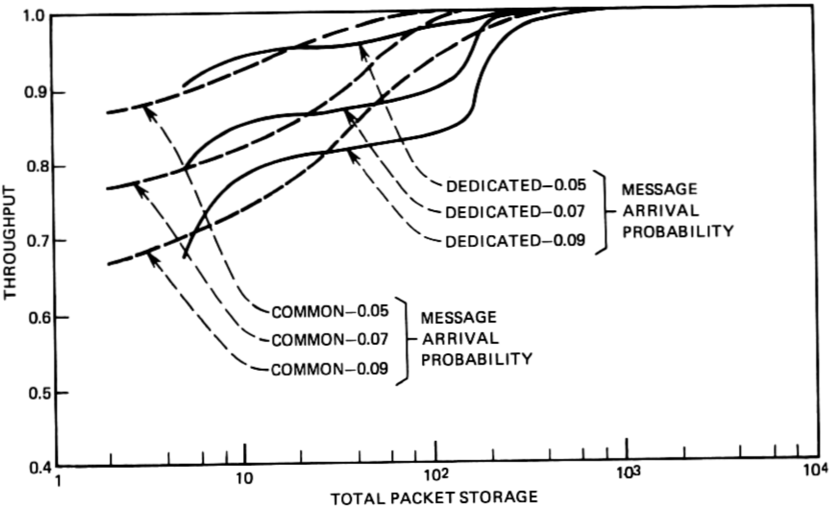

Fig. 12—Throughput versus packet storage for 5 input lines (switching time is equal to 8 slot times).

a limited amount of buffering available. A combination of factors produces this result. First of all, even though storage may be held in common, it is committed to input lines (virtual channels) in a specific

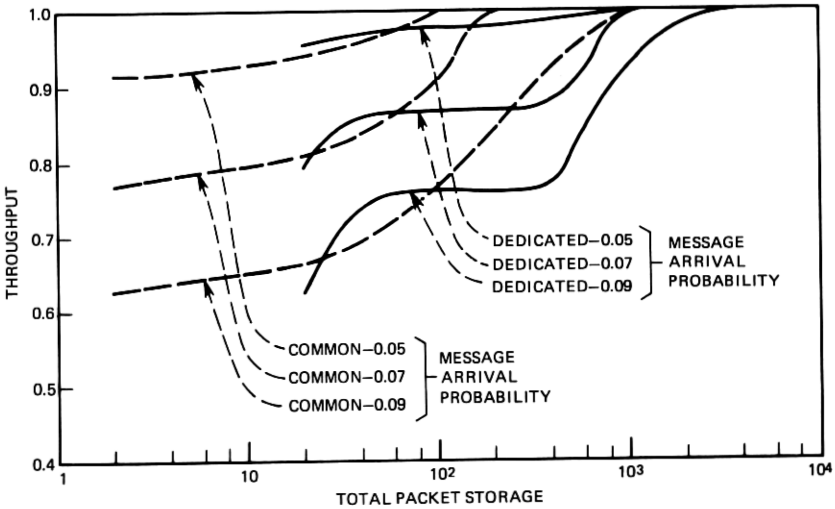

Fig. 13—Throughput versus packet storage for 20 input lines (switching time is equal to 8 slot times).

way that may be far from optimum. The preponderance of traffic is contained in messages that are 32 packets long. If the amount of storage held in common is limited, one channel may absorb all of the storage that is available in the switch. We have also simulated models where all messages are one packet long. In this case common storage is superior to dedicated. However, even in this case the difference between common and dedicated storage is not great.

In the foregoing, messages are generated regardless of whether there is room in the switch or not. We have also examined an implementation where messages are not generated until there is buffer space available. The results of a simulation study of this implementation are essentially the same as the results presented here.

Before concluding this section let us consider the reliability of the foregoing results. First of all, a good many simulation runs were made whose results are not shown here. In these runs, the number of channels, the starting points of the random sequence used in the Monte Carlo technique, and the running time of the simulation were varied. All of the results were in conformity with the results presented in this paper.

Recall that an input parameter to the program was the probability of message arrival. In Fig. 11 the theoretical relationship between loading and this quantity is shown. Also shown on Fig. 11 is the loading obtained in the simulation. As seen on Fig. 11, the results of simulation are within 5 percent of the theoretical values. This is additional indication that the simulation runs were long enough to obtain representative data sequences.

Estimates of the standard deviation of the throughput were made. This was done on each simulation run by measuring the throughput every 1000 cycles, obtaining a sequence of 20 points. (Recall that the simulation runs were 20,000 cycles long.) The mean, μ, and the variance, σ^2, of these points was calculated, giving an estimate of the mean and standard deviation of the throughput in 1000-cycle intervals. The sum of these points is the total throughput for the simulation run. If we assume that the throughputs for successive 1000-cycle intervals form a sequence of independent, identically distributed random variables, the variance of the throughput for a 20,000-cycle run is $20\sigma^2$. The coefficient of variation or the ratio of the standard deviation to the mean for a 20,000-cycle run is

$$\mathrm{CV} = \frac{20\mu}{\sqrt{20\sigma^2}}.$$

Our measurements show that in all cases the coefficient of variation is less than 0.1 and in many cases it is less than 0.05. This result means that with different starting points in the random sequences we would expect a relatively small variation around the points we have plotted on Figs. 12 and 13.

VII. BUFFER STORAGE REQUIREMENTS

At any given point in time, a User Terminal knows which virtual channels have messages waiting to be delivered. A terminal is free to select these virtual channels in any order. We assume that a terminal will not interrupt the reading out of a message in order to switch to a new channel. Since the channel selection procedure is entirely in the hands of the User Terminal, we studied the effect of different strategies on system storage requirements. Accordingly, a calculation of buffer occupancy statistics for different user strategies was performed.

As in the study of throughput we use the model shown on Fig. 10. Again, input lines correspond to virtual channels and switching between buffers corresponds to the selection of new virtual channels. In order to make the analysis tractable, we assume that messages arrive at a Poisson rate of λ messages per second over each input line. Further, we assume that eight packet slot times are required to switch buffers. In order to calculate bounds, we also consider the case where no time is required to switch.

Now if it is known which of the buffers are not empty, the worst strategy, in terms of buffer occupancy, is to always switch after reading a message out of a buffer. Thus, even if messages remain in a buffer, time is wasted in switching to a new buffer. In the sequel we shall refer to this strategy as "1-by-1." In contrast, the most efficient strategy is to cycle through the N buffers skipping empties and reading out the entire contents of nonempty buffers. This latter strategy is the one considered in the previous section on throughput. We shall refer to it as "empty before switch." An intermediate strategy involves switching at random. In this case, after a message has been read out of a buffer, one selects the next buffer at random from those having messages. It can be shown that, if there are N buffers, the probability of switching to a new buffer is 1-1/N. Thus with probability 1/N, two messages are read from the same buffer in succession.

We analyze the buffer occupancy of the 1-by-1 and the random strategies by using the theory of the M/G/1 queue. Messages arrive at all buffers at a Poisson rate $N\lambda$ messages per second. If the time required to switch between buffers is zero, we can take the service time in both strategies to be either 1 slot time or 32 slot times, depending on whether a message is long or short. An upper bound on storage requirement for the 1-by-1 strategy can be found by assuming that after each message is read out one always switches to a nonempty buffer. We can analyze this situation by adding the switch time to the time required to read out each message. Thus we have read out times of 9 slot times for short messages and 40 slot times for long messages. This is an upper bound because there is a nonzero probability that all of the messages are in the same buffer and there is no reason for the station to select a new virtual channel.

For the random strategy the service time is slightly different than 1-by-1. With probability $1/N$ no switching takes place and the service time is simply the time required to multiplex a message.

Let b be a random variable denoting the number of slots required to read a message out of a buffer, including switching time. We write $b = (m + w)T_s$, where w is the time required to switch in slot times. The random variable w is independent of m. If, in the case of random switching, 8 slot times are required to switch between buffers, the probability that $w = 8$ is 1-$1/N$ and the probability that $w = 0$ is $1/N$. From the analysis of the M/G/1 queue,[5] it can be shown that the mean number of messages in all N buffers is

$$\bar{n}_1 = N\lambda\bar{b} + \frac{(N\lambda)^2\overline{b^2}}{2(1 - \lambda N\bar{b})}, \tag{4}$$

where $\overline{b^i}$ is the ith moment of b and λ is the average message arrival rate in messages per second. The mean-square number of messages in the buffer is

$$\overline{n_1^2} = (N\lambda)^2\overline{b^2} + \frac{3\bar{n}_1(N\lambda)^2\overline{b^2} + (N\lambda)^3\overline{b^3}}{3(1 - \lambda N\bar{b})} + \bar{n}_1. \tag{5}$$

Our primary interest is in the number of data packets in switch buffers rather than in the number of messages. It can be shown that the mean and the mean-square number of packets in all N buffers is given respectively by

$$\bar{p}_1 = \bar{m}\bar{n}_1 \tag{6}$$

and

$$\overline{p_1^2} = \overline{n_1^2}(\bar{m})^2 + \bar{n}_1[\overline{m^2} - (\bar{m})^2]. \tag{7}$$

The foregoing considers packet storage requirements in all of the buffers in the switch. We shall focus our attention on the number of packets in individual buffers assigned to virtual channels. The mean number of packets in each buffer is simply $\bar{p}_1$ given by eq. (6) divided by the number of buffers N. In order to calculate the variance of the number of packets in individual buffers, it is necessary to assume that the contents of a buffer are independent of the contents of any other buffer. Under this assumption the variance of the number of packets in any buffer is given by

$$V_1 = [\overline{p_1^2} - (\bar{p}_1)^2]/N. \tag{8}$$

7.1 "Empty before switch" strategy

We now consider the strategy in which the User Terminal goes cyclically from buffer to buffer emptying the entire contents of each buffer. The User Terminal will not select a virtual channel which has no messages in its buffer. The analysis of the number of packets in a buffer under this strategy is mathematically difficult. However, we can obtain an upper bound by considering a strategy in which each of the N buffers is examined in turn (even empty buffers). Since time is wasted examining buffers which are known to be empty, the upper bound follows.

The analysis of the cyclic system is contained in Ref. 6. Since the contents of each buffer is random, the time required to cycle through all buffers is a random variable. It can be shown that the mean and the mean-square values of this quantity are

$$\bar{\tau}_c = \frac{N\bar{w}T_s}{1 - N\lambda\bar{m}T_s} \tag{9}$$

$$\overline{\tau_c^2} = \frac{N(N-1)(\bar{w} + \bar{\tau}_c\lambda\bar{m})^2T_c^2 + (\overline{w^2} + 2\bar{w}\bar{\tau}_c\lambda\bar{m} + \bar{\tau}_c\lambda\overline{m^2})T_s^2}{1 - N(\lambda\bar{m}T_s)^2}. \tag{10}$$

In the sequel we shall consider the time to switch between buffers, w, as fixed; therefore, $\bar{w} = w$ and $\bar{w}^2 = w^2$. The quantities $\bar{m}T_s$ and $\overline{m^2}T_s^2$ denote the first two moments of the time required to read a message out of a buffer. The mean number of messages in the buffer is

$$\bar{n}_2 = \lambda\bar{m}T_s + \frac{\bar{\tau}_c^2}{2\bar{\tau}_c}(1 + \lambda\bar{m}T_s). \tag{11}$$

The mean number of packets is

$$\bar{p}_2 = \bar{m}\bar{n}_2. \tag{12}$$

An expression for the mean-square number of packets in each buffer can be derived from the work presented in Ref. 6. This expression is rather lengthy and provides little insight; therefore, we shall omit it.

Results of computation using this expression will be presented in the sequel.

VIII. RESULTS OF BUFFER STORAGE REQUIREMENTS STUDY

The results of computations using the formula derived in the foregoing are shown in Figs. 14 and 15 for 5 and 20 buffers, respectively. In these figures the average occupancy of each buffer is shown as a function of load, which is the product of the message arrival rate and the average time required to read out a message, $N\lambda\bar{m}$. As expected, the lowest buffer occupancy occurs in the case where no time is required to select a new virtual channel. When an 8-slot-time channel select time is required, the technique with the lower occupancy depends upon the loading. At light loading, the "empty before switch" strategy shows poorer performance because time is wasted stopping at empty buffers. It must of course be remembered that this is only an upper bound for the "empty before switch" that selects only nonempty buffers. As the loading increases, there are fewer empty buffers and the performance of the "empty before switch" strategy improves relative to the 1-by-1 strategy.

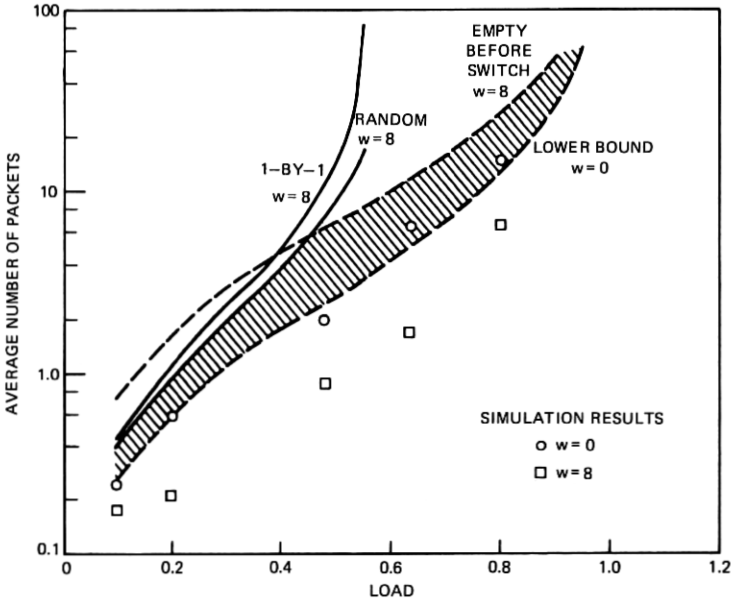

Fig. 14—Average number of packets in each buffer for 5 buffers (30 percent of messages are 32 packets long).

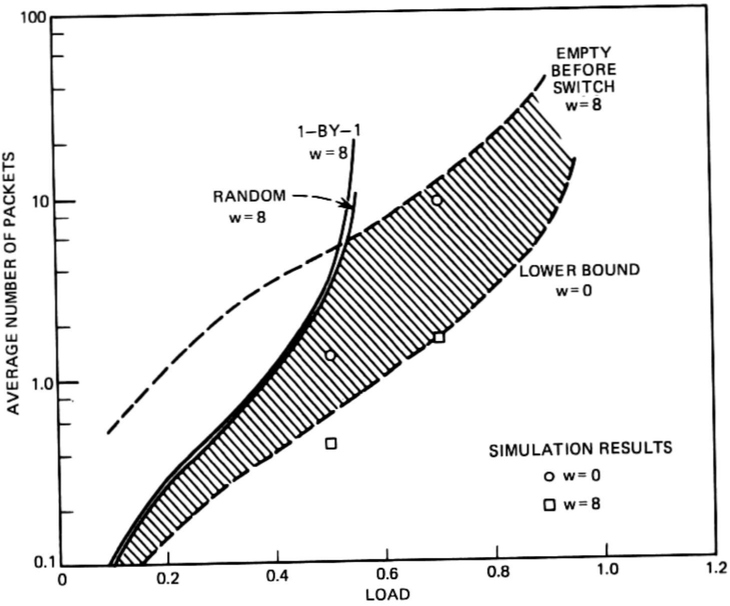

Fig. 15—Average number of packets in each buffer for 20 buffers (30 percent of messages are 32 packets long).

In the previous section we considered an "empty before switch" strategy that skipped empty buffers. As we have mentioned, the problem of calculating occupancy statistics for this technique is intractable. However, the results shown on Figs. 14 and 15 form bounds on the skipping empty technique. The shaded areas in the figures indicate the areas in which the statistics for this method lie.

If the system is operated below 0.5 loading, the difference between the different channel switching strategies is not very large. For example, for 20 channels and 0.4 loading (see Fig. 15) the average occupancy for 1-by-1 rotating strategy is 1.1 packets. For the "empty before switch" strategy skipping empties, the average occupancy is between 0.4 and 1.0 packet. As the load increases beyond 0.5 loading, the 1-by-1 strategy leads to saturation and the cyclic system is clearly superior.

Results for the standard deviations of buffer occupancy have been obtained. These results support the foregoing conclusions.

The simulation program discussed in the previous section computed means and standard deviations of buffer occupancy for an "empty before switch" strategy with skipping of empty buffers. The results are shown on Figs. 14 and 15. A comparison of analysis and simulation indicates that for the most part the analysis gives upper bounds to the simulation. This is not unexpected since, in the simulation program, messages arrive over an interval of time, whereas for the Poisson arrival model used in the analysis, messages arrive instantaneously.

IX. ACKNOWLEDGMENTS

The system described in this paper was conceived and built by A. G. Fraser.[7] The author would like to express his appreciation to Dr. Fraser for many enlightening discussions on the system while our work was being carried out. We would also like to thank Brian W. Kernighan who, with patience and unfailing good humor, answered many questions on the computer programming involved in the work.

APPENDIX

Delay in synchronous time division multiplexing

Equation (1) of the text is the following expression for the delay in Synchronous Time Division Multiplexing:

$$d_1 = qT_c + T_c \sum_{i=1}^{L} m^3 + w + (m_{L+1} - 1)T_c. \tag{1}$$

The definitions for each of the terms on the RHS of (1) are given in the text.

The delay, d_1, is the sum of the three mutually independent random variables qT_c, $T_c \sum_{i=1}^{L} m_i + w$, and $(m_{L+1} - 1)T_c$. Thus the moment-generating function for d_1 is the product of the moment-generating function for these three variables. In this appendix we shall calculate the moment-generating functions of each of these.

Recall that in STDM dedicated packet slots are available to each terminal cyclically every T_c seconds. We take the end of one cycle and the beginning of the next to be the end of a dedicated packet slot.

Let q_j be the number of packets remaining at the end of the jth cycle and let a_j be the number of packets arriving during the jth cycle. We can write

$$\begin{aligned} q_{j+1} &= q_j - 1 + a_{j+1} \qquad &&\text{for} \quad q_j + a_{j+1} > 0 \\ &= 0 &&\text{for} \quad q_j + a_{j+1} = 0. \end{aligned} \tag{13}$$

Writing this in a more compact form, we have

$$q_{j+1} = q_j + a_{j+1} - U(q_j + a_{j+1}), \tag{14}$$

where $U(x)$ is such that $U(x) = 1$ for $x > 0$ and $U(x) = 0$ for $x \leqq 0$. Taking expectation on both sides of (14) and assuming equilibrium (i.e., $Eq_{j+1} = Eq_j$) we find that

$$E[u(q_j + a_{j+1})] = E[a_{j+1}].$$

But

$$E[u(q_j + a_{j+1})] = P_r[q_j + a_{j+1} > 0].$$

Therefore,

$$P_r[u(q_j + a_{j+1} = 0)] = 1 - \lambda \bar{m} T_c. \tag{15}$$

We now find the moment-generating function. From (14) we have

$$\begin{aligned} E[e^{-sq_{j+1}}] &= E[e^{-sq_j - sa_{j+1} + U(q_j + a_{j+1})}] \\ &= \sum_{k=0}^{\infty} P_r[\bar{q}_j + a_{j+1} = 0]e^{-s[k-U(k)]} \\ &= P_r[\bar{q}_j + a_{j+1} = 0] + e^s \sum_{k=1}^{\infty} P_r[\bar{q}_i + a_{j+1} = k]e^{-sk}. \qquad (16) \end{aligned}$$

If we assume equilibrium has been reached, we may define

$$Q(s) = E[e^{-sq_j}]$$

for all j. From (16) after some manipulation we have

$$Q(s) = \frac{(1 - \bar{m}\lambda T_c)(e^{-s} - 1)}{e^{-s} - A(s)}, \tag{17}$$

where $A(s) \equiv E[e^{-a_j s}]$. Since messages arrive at a Poisson rate λ, it can be shown that

$$A(s) = e^{-\lambda T_c[1-M(s)]}, \tag{18}$$

where $M(s)$ is the generating function of the messages.

By successive differentiation it can be shown that the first two moments of q are

$$\bar{q} = \frac{\overline{a^2} - \bar{a}}{2(1 - \bar{a})}, \tag{19}$$

$$\overline{q^2} = \frac{\overline{a^3} - \bar{a} + 3\bar{q}(a^2 - 1)}{2(1 - \bar{a})}, \tag{20}$$

where $\bar{a}$, $\overline{a^2}$, and $\overline{a^3}$ are the first three moments respectively of the number of packets arriving in a cycle T_c. By successive differentiation of (18) we find that

$$\bar{a} = \lambda T_c \bar{m}, \tag{21a}$$

$$\overline{a^2} = (\lambda T_c)^2 \overline{m^2} + (\lambda T_c \bar{m})^2, \tag{21b}$$

$$\overline{a^3} = \lambda T_c m^3 + 3(\lambda T_c)^2 \bar{m}\overline{m^2} + (\lambda T_c \bar{m})^3, \tag{21c}$$

where $\bar{m}$, $\overline{m^2}$, and $\overline{m^3}$ are respectively the first three moments of the number of packets in a message.

We turn now to the second term in (1), $\bar{f} \triangleq w + T_c \sum_{i=1}^{L} m_i$. A message arrives at random during a cycle, w seconds before the end of a cycle. In the time interval $T_c - w$, L messages arrive, all of which have priority over the newly arrived message. Since message arrival is random, the quantities L and w are mutually dependent random variables. Conditioned on w, the probability that L messages arrive in the interval $T_c - w$ is

$$P_r[L \text{ messages in } T_c - w] = \frac{\lambda^L (T_c - w)^L}{L!} e^{-\lambda(T_c - w)}.$$

The random variable w is uniformly distributed in the interval $(0, T_c)$. Let $r(t)$ be the density function of the random variable $T_c m_i$. We may write

$$P_r[\tau < \bar{f} \leqq \tau + d\tau]$$
$$= \frac{1}{T_c} \int_0^{T_c} dw \sum_{L=0}^{\infty} \frac{\lambda^L (T_c - w)^L}{L!} e^{-\lambda(T_c - w)} r^{(L)}(\tau - w), \tag{22}$$

where $r^{(L)}(t)$ is the L-fold convolution of $r(t)$. The Laplace-Stieltjes transform of this can be shown to be

$$F(s) = \frac{e^{-\lambda T_c [1 - R(s)]} - e^{-s T_c}}{T_c \{s - \lambda[1 - R(s)]\}}, \tag{23}$$

where $R(s)$ is the L–S transform of $r(t)$. It can be shown that $R(s) = M(T_c s)$.

The first two moments of f can be found by successive differentiation of (23):

$$\bar{f} = \frac{T_c}{2} + \frac{\lambda T_c^2 \bar{m}}{2}, \tag{24}$$

$$\overline{f^2} = \frac{T_c^5 \lambda^3 (\bar{m})^3 + 3\lambda^2 T_c^4 \bar{m}\overline{m^2} + 3\lambda T_c^2 \bar{f} + T_c^2}{3(1 - T_c \lambda \bar{m})}. \tag{25}$$

The final term to be evaluated in (1) is $(m_{L+1} - 1)T_c$. Since the generating function of the message is $M(s)$, the generating function of this term is easily shown to be

$$G(s) = e^{sT_c}M(sT_c). \tag{26}$$

The first two moments of g are easily shown to be

$$\bar{g} = T_c(\bar{m} - 1), \tag{27}$$

$$\overline{g^2} = T_c^2(\overline{m^2} - 2\bar{m} + 1). \tag{28}$$

The mean value of delay is the sum of the terms $\bar{q}T_c$, $\bar{f}$, and $\bar{g}$. The variance of the delay can be calculated by summing the variances of qT_c, f, and g.

REFERENCES

1. A. G. Kohneim, "Service Epochs in a Loop System," presented at the 22nd Int. Symp. Computer-Communications Networks and Teletraffic, Polytech. Inst. Brooklyn, Brooklyn, N. Y., April 1972.
2. J. F. Hayes and D. N. Sherman, "Traffic Analysis of a Ring Switched Data Transmission System," B.S.T.J., *50*, No. 9 (November 1971), pp. 2947–2978.
3. R. R. Anderson, J. F. Hayes, and D. N. Sherman, "Simulated Performance of a Ring Switched Data Network," IEEE Trans. Commun., *COM-20*, No. 3 (June 1972), pp. 576–591.
4. B. Avi-Itzhak, "Heavy Traffic Characteristics of a Circular Data Network," B.S.T.J., *50*, No. 8 (October 1971), pp. 2521–2549.
5. D. R. Cox and W. L. Smith, *Queues*, London: Methuen and Co., 1961.
6. J. F. Hayes and D. N. Sherman, "A Study of Data Multiplexing Techniques and Delay Performance," B.S.T.J., *51*, No. 9 (November 1972), pp. 1983–2011.
7. A. G. Fraser, "Interconnecting Computers and Digital Equipment," internal report available upon request.