THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 51, No. 9, November, 1972
Printed in U.S.A.

A Study of Data Multiplexing Techniques and Delay Performance

By J. F. HAYES and D. N. SHERMAN

(Manuscript received May 25, 1972)

This paper is concerned with the study of data multiplexing techniques which provide local distribution for user populations whose source characteristics may be categorized as Inquiry/Response. The techniques studied are Polling, Random Access, and a Loop System. In each method a group of users is multiplexed onto the same line which is connected to a Central Processor. The Central Processor forms the interface between the local system and long haul facilities, but does no computation on arriving messages other than to route them appropriately. The advantage of such systems is flexibility of operation and economy through sharing of equipment at the Central Processor. We focus our attention on the average round-trip message delay and use this as a measure for comparisons of the three techniques.

Source models, in terms of calls per busy hour and number of bits per message for users as well as for computer responses, are developed which are appropriate in an Inquiry/Response context. Other factors taken into consideration are transmission rates, synchronization delay time, and allocation of available capacity to the transmission of overhead information, such as polling messages and acknowledgment traffic.

The results of the study are presented in graphic form, where the average delay due to traffic in the system is plotted as a function of the number of stations. Principal conclusions can be summarized as follows:

Polling—It is found that this system is sensitive to the synchronization delay which takes place each time a user station transmits to the central facility. For reasonable choices of system parameters as many as 100 stations can share the Central Processor without exceeding an average of 1-second round trip message delay.

Random Access—This system is not sensitive to synchronization delay. Over most of the range of load parameters, the Random Access system shows lower average delay than the other two systems.

Loop System—The study revealed that this system has average delay

performance comparable to the other two. The Loop system is not sensitive to synchronization delay.

I. INTRODUCTION

One of the major functions of data communication networks is providing means by which users can access a distant computer in an *interactive* manner. Time-sharing, inquiry-response and credit checking are examples of applications which have a common communications requirement. They require rapid set up times for access to their respective computers and, as a consequence of their interactiveness, often require high-capacity return channels for return traffic.

Providing access to a distant computer involves providing two major data transmission media, viz. long haul facilities and local distribution. (See Figs. 1 and 2.) While there are important problems associated with each part, we shall focus our attention on local distribution techniques. We are motivated in this direction by the relatively high cost of providing local distribution compared to the total cost of transmission.

The results presented in the sequel are most appropriate to the situation where users desiring access to distant computers are geographically clustered. In this instance, traffic from several users is concentrated at what are designated as User Stations. We shall consider multiplexing techniques whereby traffic from User Stations is conveyed to a central point which we call the Central Processor. The role of the Central Processor is to route or direct message flow to and from User Stations and computers.

This paper is devoted to an analysis of the roundtrip delay of three techniques which multiplex traffic onto common facilities. The analysis is based on models of user traffic and computer responses to user mes-

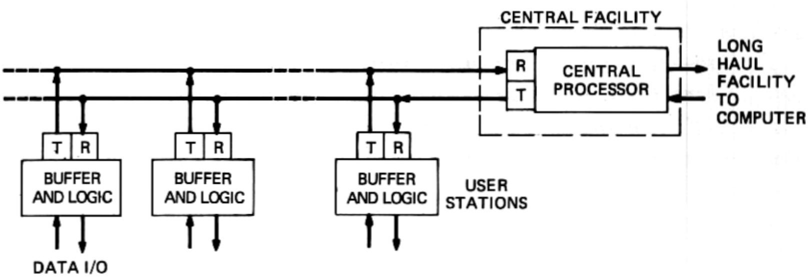

Fig. 1—Polling and random access systems.

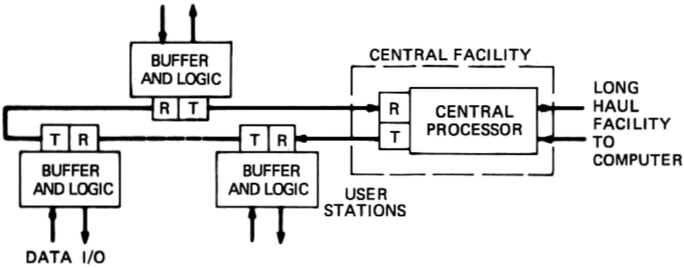

Fig. 2—Loop system.

sages. In the applications we consider, traffic from the user is bursty, i.e., short periods of activity followed by relatively long idle periods. Each message from a user elicits a response from the computer.

In the sequel we shall analyze the three alternatives for local distribution:

(*i*) Polling—(See Fig. 1) Stations are polled by a Central Processor and transmit over a common line. Return traffic is multiplexed over a second common line by the Central Processor.

(*ii*) Random Access—(See Fig. 1) Here User Stations transmit, at will on a common line, all messages that are generated. Positive acknowledgments are issued by the Central Processor upon error-free reception of the messages. Until an acknowledgment is received, the message is held in the station's buffer. Again, return messages travel over a second common line.

(*iii*) Loop System[1]—(See Fig. 2) User Stations share the same line. Traffic already on the line has priority, and newly generated messages are multiplexed into gaps in the line traffic.

In each of these techniques, traffic to and from different User Stations share common facilities. The basic difference among the techniques lies in the means of controlling traffic accessing the system through the User Station. In the Polling System, the Central Processor exercises tight control over entering messages. As we shall see, in order to do this, overhead is incurred which causes an increased delay. In contrast, entering traffic in the Random Access System is loosely controlled by the Central Processor and delay attributable to overhead is reduced considerably. In the Loop System, as in the Random Access System, messages are multiplexed on the line at the User Station without direct control by the Central Processor.

Our analysis of performance concentrates on buffering or queueing delay. The results of this analysis will be used to calculate the roundtrip queueing delay encountered by a user-generated message. A message going to the computer from a particular station is competing for the line with messages from other stations. Consequently it encounters delay before it has sole access to shared facilities. Further, messages returning from the computer may be delayed in a queue at the Central Processor before being transmitted over the common line. We shall designate the sum of these two delays as the roundtrip delay. Of course, a message will encounter other delays such as propagation delay from the Central Processor to a distant computer and service time in the computer. However, these delays are independent of the local distribution system delays and can simply be added to the roundtrip delay that we calculate.

A common thread running through each of these systems is interactive queues. At each of the User Stations storage is assigned, if necessary, to queue up messages. However, since all the queues share the same server, the queues are not independent of one another. The exact treatment of interactive queues is mathematically difficult, therefore, in carrying out the analysis, certain approximations have been made.

The three systems described above are compared on the basis of average roundtrip delay. Since we have computed the moment generating functions of forward and return delay, higher order moments can be found easily enough. From these higher order moments one can calculate other measures of performance. In so doing, the question of correlation between forward and roundtrip delay must be considered. We did not concern ourselves with this correlation since it has no bearing on the average roundtrip delay.

In developing the models for the systems, we have attempted to take into account the constraints encountered in real communications systems. In the concluding section of the paper, examples using parameters of the Digital Data System[2] are presented.

II. GENERAL CONCLUSIONS

At the conclusion of the analytical investigations, we spend the final section of this paper in describing the delay performance of the systems. There also we compare, where possible, the advantages and disadvantages of various possible implementations. While the details appear later, it is worthwhile to give some general conclusions about the systems' behavior here.

For a given number of stations (N) and call rate (λ), the average message delay in the Polling System depends strongly on how quickly receivers can synchronize (s) as well as the available channel capacity. For instance, one can imagine that in the Polling System, polls take place over a separate channel of capacity δ. Now let us suppose that two separate systems could be designed, the first having zero synchronization time and the other taking 10 ms for the Central Processor to synchronize the polls. We would find as the number of stations increased, that the 10 ms spent at each station ultimately degrades the performance substantially. This effect can be seen by comparing Figs. 3 and 4 where we have depicted the above situation. For a large number of stations ($\sim$ 100) the two lower right families of curves rotate counterclockwise, indicating increasing delay. Note that even if we poll with capacity $\delta = 100$ bits/s (the upper right families of curves in Figs. 3 and 4), the delay is almost independent of the synchronization time and call rate. This is primarily due to the fact that we are forced to poll fewer stations (to maintain reasonable delay) and thus incur a smaller overhead induced delay.

If the aim of the system is to serve a small number of stations each of which has a high calling rate, the synchronization time is relatively unimportant. Thus it may be desirable to use a low speed line with a small synchronization time. In Fig. 5, we depict this situation and note that it is possible in this case to achieve similar performance for a small number of stations ($\leqq 25$) as is achieved with high speed line (Fig. 3).

Since the Central Processor in the Random Access technique exercises loose control over traffic flow, there is a much smaller effect of overhead and correspondingly smaller delay (see Figs. 6 and 7).

In the applications we consider, the volume of return traffic is greater than the volume of forward traffic, therefore, the first station in a Loop system suffers the largest delay. In considering the Loop system we focus on this station. Analytical results obtained in the sequel indicate that the Loop System compares favorably with each of the previous techniques. In the Loop System, such quantities as synchronization time and retransmission delay do not arise. What is relevant in this system is the processing delay, which involves examining the addresses of packets passing through the station.

III. GLOSSARY OF IMPORTANT QUANTITIES

N—number of User Stations

λ—arrival rate of messages to the User Station (messages/s or calls/busy hour)

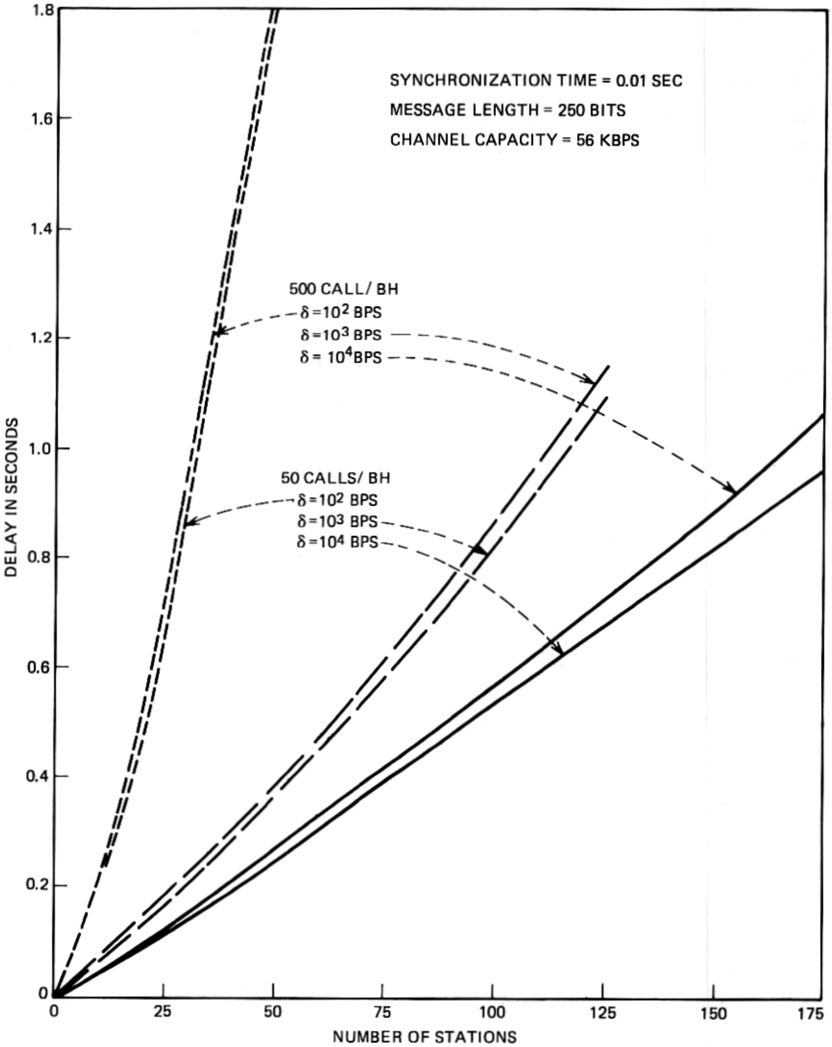

Fig. 3—Polling system delay in forward direction.

B_F—message length generated at User Stations (bits)
C_F—capacity of forward link (bits/s)
m—time duration of message while on forward link (s)
B_R—return message length (bits)
C_R—capacity of return link (bits/s)

M—time duration of message returning from the Central Processor (s)

w—minimum time between polls in Polling System and minimum retransmission time in Random Access System (s)

$\bar{t}_c$, $\overline{t_c^2}$—mean and second moment of the time required to poll and readout messages in Polling System

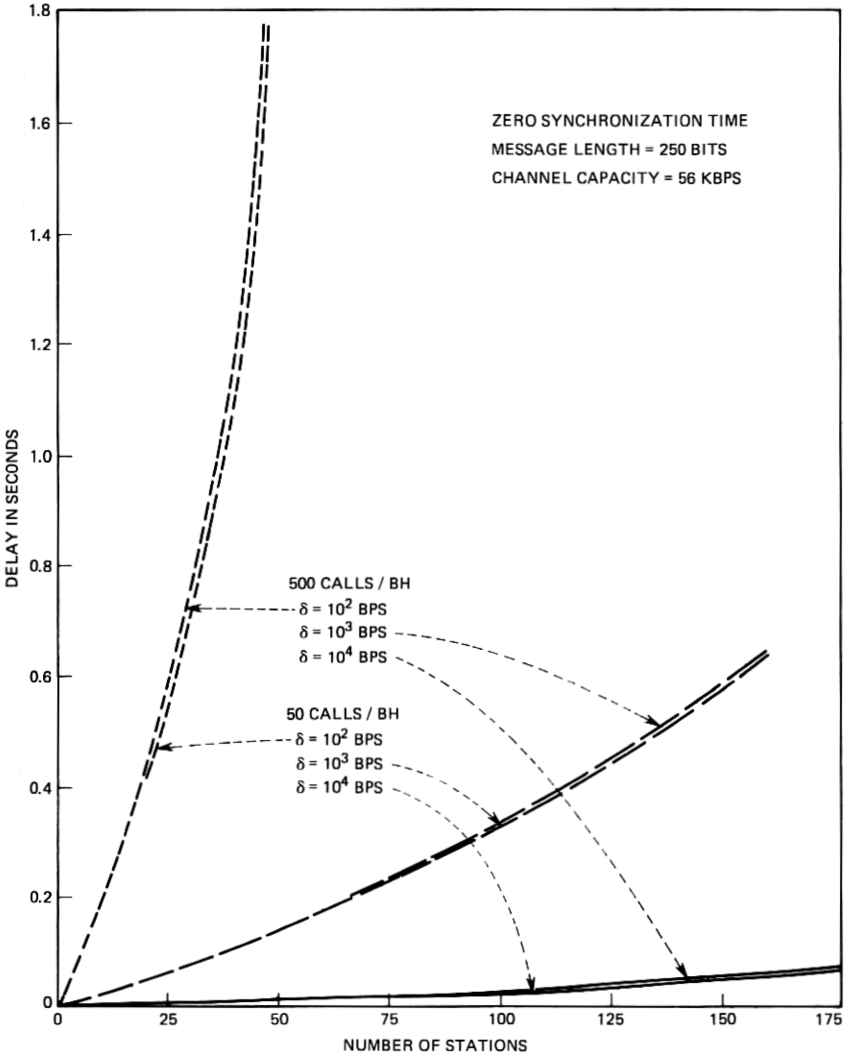

Fig. 4—Polling system delay in forward direction.

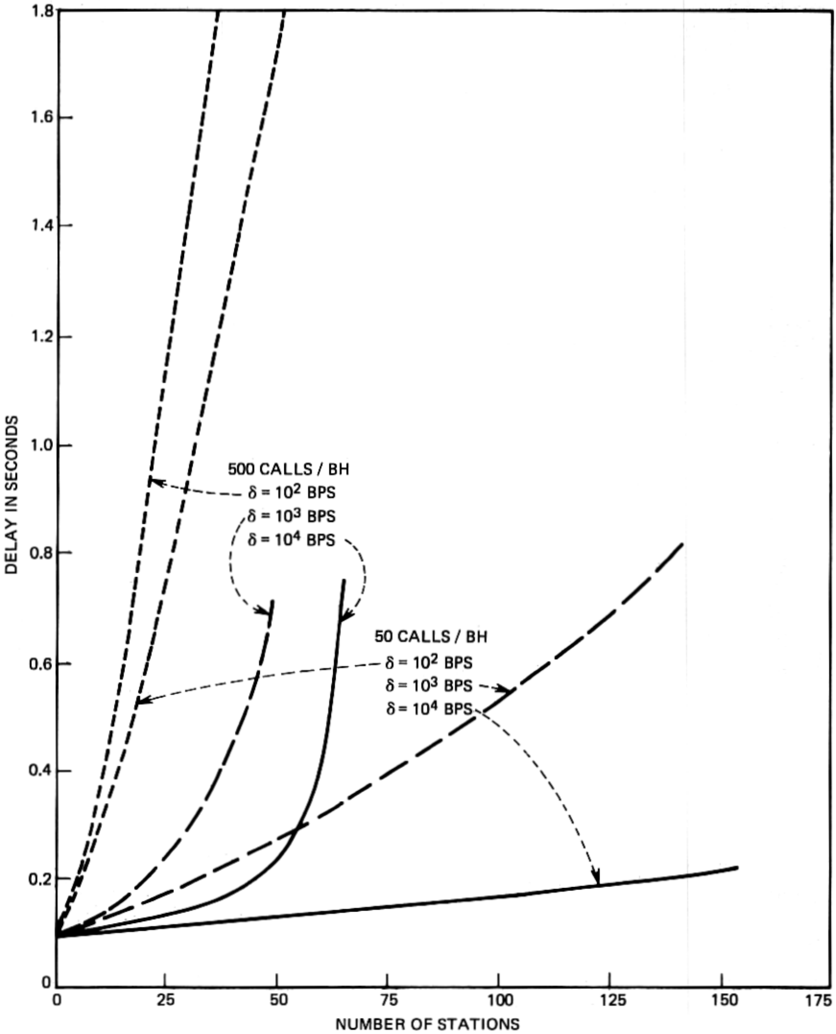

Fig. 5—Polling system delay in forward direction.

ρ—intensity of arriving traffic

d_{RA}—inbound message delay in Random Access System (s)

d_L—inbound message delay in Loop System (s)

d_P—inbound message delay in Polling System (s)

d_R—message delay on return link (s)

δ—capacity of return link allocated to polling messages in Polling

System and to positive acknowledgements in Random Access System (bits/s)

s—synchronization time (s).

IV. SOURCE MODELS

As indicated in the introduction we are primarily interested in the bursty, interactive user. Such users are encountered, for example, in credit checking and Inquiry-Response applications. The models that we shall develop are also applicable to time-sharing computer systems where users access a distant computer. Traffic characteristics of such users have been studied,[3,4] so that estimates of parameters characterizing the source (holding time, rate, etc.) are available.

We model user traffic as consisting of Poisson arrivals of short, fixed-length messages. The arrival rate is designated as λ either in calls per busy hour or in messages per second. Each message consists of B_F bits. If the line serving a User Station has a capacity of C_F bits per second then $m = B_F/C_F$ seconds are required to transmit each message.

In our analysis we shall assume that each User Station has unlimited buffer capacity so that buffer overflow never occurs. From the delay formulas that we develop, one can also derive buffer occupancy statistics, and thus imply the buffer size required for small overload probability.

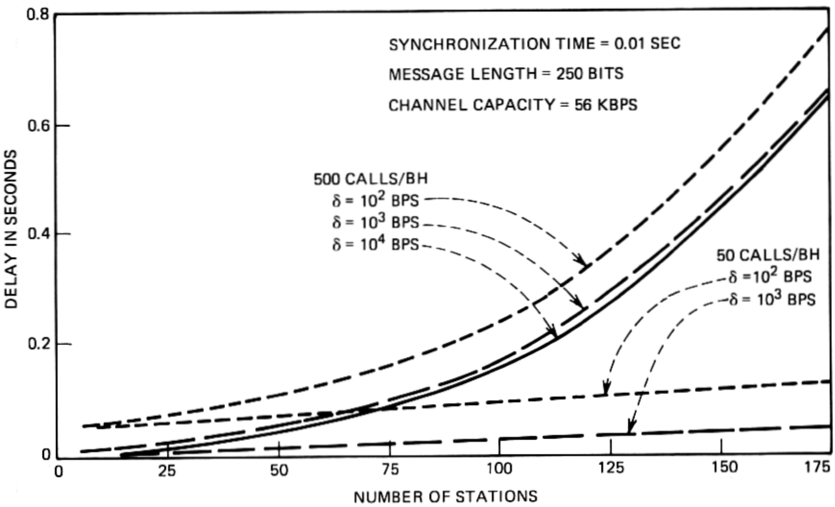

Fig. 6—Random access system, delay in forward direction, fixed-time retransmission.

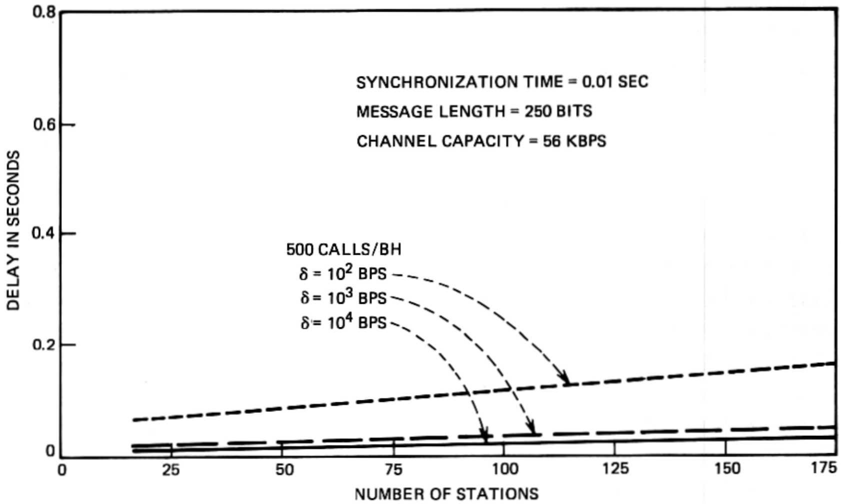

Fig. 7—Random access system, random retransmission, forward delay.

For the message arrival rates that we consider, the station buffers need not be very large in terms of message lengths to achieve adequately small overflow probability.

User messages arriving at the Central Processor are routed to a computer. Each message elicits a response from the computer. All of these return messages pass back again through the Central Processor. If messages arrive at the N user stations at average rate λ messages per second, then the average rate of return message flow through the Central Processor is $N\lambda$ messages per second. We assume that each return message has a fixed length of B_R bits, although our analysis can just as well be carried out for random message lengths.

A basic assumption in our analysis is that the return message flow is Poisson, i.e., messages arrive at the Central Processor at random. Several conditions serve to randomize the return message flow. It may well be that messages sharing the same local facilities are destined for different computers, so that the returns from different computers will tend to be uncorrelated. The computers will be serving many local communities of users, introducing a random delay between successive messages returning to the same Central Processor.

Under the assumption of a Poisson process of return messages, buffering must be provided at the Central Processor. Messages are read out of the buffer and transmitted to the User Station at a constant bit rate.

V. POLLING SYSTEM

The polling technique is not new, and the reader is referred to Martin[5,6] for discussions on the present approach and others. The type of polling under study is 'roll-call' polling in which the Central Processor works down a list of users and, in sequence, interrogates User Stations sharing the same line. Upon being polled, the User Station may transmit messages. Since the generation of messages is random with respect to the time of polling, messages must queue in a buffer located at the User Station until a polling message arrives. As indicated earlier, we focus our attention on the average delay encountered by a message.

The arrangement of lines and User Stations for the Polling System is shown in Fig. 1. User Stations are bridged across common lines. Notice that, in contrast with the star connection, only one receiver and one transmitter is required at the Central Processor. The Polling System can be implemented using a "daisy chain" configuration in which the line between two user stations loops back to the Central Processor.

Since polling messages are transmitted over a common line, User Stations must be capable of recognizing messages addressed to it. The same line is used in common by messages returning from the computer. Again the User Station must be able to recognize return messages addressed to it. As we shall see, there is an interplay between forward and return message delay, in as much as polling messages and return traffic share the same physical line.

The first component of the round-trip message delay that we consider is the delay at the User Station that a message encounters before it is put on the line. The calculation of this delay is complicated by the fact that the number of messages queueing at a particular station strongly depends upon the number of messages stored in all of the other stations. Since arrivals to any station are at a Poisson rate, the less frequently a station is polled, the more messages accumulate. Moreover, as messages accumulate, the polling cycle lengthens.

The polling problem has been analyzed by Leibowitz[7] by assuming a form of independence between queues.* The results of his analysis using this assumption compares well with the results of a rigorous analysis for a two-queue network. The nature of the independence assumption is such that the largest error occurs in a two-queue system.

* A rigorous solution of the polling problem has recently been found by Eisenberg.[8] However, in our application, which may involve a large number of stations, Leibowitz's approach leads to more tractable results.

Of primary importance in the calculation of delay in the forward direction is a parameter that we designate as the 'walk-time' between stations. In the situation where a station has no message to transmit, the Central Processor must still spend time interrogating it. Time is required to transmit a polling message. The speed with which polling messages can be transmitted is limited, since messages returning from the computer share the same line. Further, after polling a station, the Central Processor must wait for a possible response. This time involves, for example, establishing synchronization between the User Station and the Central Processor. We shall return to the walk-time after we have considered its effect on delay.

In calculating the delay of a message in this system, it is necessary to characterize the time between polls of a particular station. From the moment generating function derived in Appendix A we obtain the mean and the mean square values of the cycle time

$$\bar{t}_c = \frac{Nw}{1 - N\rho} \tag{1a}$$

$$\overline{t_c^2} = \frac{N(N-1)(w + \rho\bar{t}_c)^2 + N[w^2 + 2w\bar{t}_c\rho + \bar{t}_c m\rho]}{1 - N\rho^2}. \tag{1b}$$

In writing (1a) and (1b), we take advantage of the fact that the walk-time is a constant value, w, and the message length is a constant value $m = B_F/C_F$. Also we define $\rho = \lambda\bar{m}$.

Messages that arrive at a User Station are buffered, awaiting the arrival of the poll. When the poll arrives, messages are read out of the buffer on a first come first served basis. Thus a message must wait until the station is polled as well as until messages that have arrived before it are read out of the buffer. In Appendix A, the generating function of delay is derived. From this generating function it can be shown that the average delay is given by

$$\bar{d}_P = \frac{\overline{t_c^2}}{2\bar{t}_c}(1 + \rho) + m \tag{2}$$

where $\bar{t}_c$ and $\overline{t_c^2}$ are given by eqs. (1a and b).

Equations (1a) and (1b) show that the walk time, w, is an important parameter in the calculation of delay. We turn now to assigning values to it. Since the polling message must contain the address of the station being polled, it must be at least $\log_2 N$ bits long. Further, since it shares the line from the processor to the station with messages returning from the computer, not all of the return line capacity is available to transmit polling messages. We treat this situation by splitting the capacity of the return line. Suppose that of the C_R bit-per-second capacity of the return line, δ bits per second, interleaved with the message bits, are allocated entirely for polling messages. The minimum time required to transmit a polling message is then

$$r = \{[\log_2 N] + 1\}/\delta \tag{3}$$

where $[x]$ is the largest integer less than x. The remaining $C_R - \delta$ bit-per-second capacity of the return line is allocated to return message traffic. This allocation may be effected by either FDM or TDM.

Other methods may be used to share the return line between polling and return messages. For example, messages may be transmitted over the same channel with priority given to polls or user traffic. The difficulty with this approach is that there may be a large difference in the lengths of the two kinds of messages. Thus, even though a polling message is short and has priority, it may wait a relatively long time until a transmission of a return message is completed.

Since User Stations continuously receive from the same point, the Central Processor, no time is required to establish synchronization for return or polling messages. This may not be true in the reverse direction since the Central Processor is receiving from a different station on each poll. Thus in the calculation of walk-time it may be necessary to allow time for the establishment of synchronization between the User Station and the Central Processor. Let this time be denoted s. The walk-time between stations is then the sum.

$$w = s + \{[\log_2 N] + 1\}/\delta. \tag{4}$$

The effect achieved by splitting return line capacity on return message delay can now be calculated. Return messages are B_R bits in duration and arrive at the central processor at a Poisson rate of $N\lambda$ messages per second. The time required to transmit each message is $B_R/(C_R - \delta)$ seconds. The delay of a message in the Central Processor may be found from the analysis of an M/G/1 queue.[9] It can be shown that the average return message delay is

$$\bar{d}_R = \frac{N\lambda[B_R/(C_R - \delta)]^2}{2[1 - B_R\lambda N/(C_R - \delta)]} + B_R/(C_R - \delta). \tag{5}$$

The average roundtrip delay is the sum of $\bar{d}_P$ and $\bar{d}_R$.

VI. RANDOM ACCESS SYSTEM

The Random Access system was suggested by the University of Hawaii's ALOHA System.[10] The configuration of this system is the same as shown for the Polling System in Fig. 1. As in the Polling System, messages arrive at each User Station at rate λ messages per second.

Messages are transmitted to the Central Processor on a first come first served basis. The length of a message in bits and the line rate are such that m seconds are required to transmit each constant length message. This constant length includes overhead bits required by the Random Access technique.

Messages are transmitted at random by each station in the system. There is a probability of messages from different stations overlapping. In order to detect message overlap, parity check bits are transmitted along with each message and error detection is performed upon reception. Only if a message is received error free is an acknowledgment transmitted to the User Station. Until a User Station recognizes a positive acknowledgment addressed to it, the message is held in a buffer. If, after a specified period of time, no acknowledgment is received, the message is retransmitted. This continues until an acknowledgment for a particular message is received.

Messages returning from the computer are buffered and sent to the User Stations in the same manner as the Polling System. As in the Polling System, messages must contain addresses since messages destined for different stations share the same line.

A basic relation governing traffic in the Random Access system has been derived by Abramson.[10,11] In this analysis, messages are assumed to enter the common line at Poisson rate. Suppose that a particular message is transmitted at time t_o. This message will encounter no interference if no other message is put on the line in the time interval $t_o \pm m$. We have then

$$Pr\,[\text{message retransmission}] = 1 - \exp\,(-2Rm) \tag{6}$$

where R is the total transmission rate on the line, including retransmissions. The rate of retransmission on the line is $R(1 - e^{-2Rm})$.

Now messages arrive at all user stations at rate λN messages per section. If there is no continuous buildup of messages at User Stations, we must have the relationship.

$$R = N\lambda + R(1 - \exp\,(-2Rm)).$$

This reduces to

$$N\lambda m = Rm \exp\,(-2Rm). \tag{7}$$

An examination of eq. (7) discloses that the maximum information transmission rate is $N\lambda m = 1/2e \approx 0.18$. As one attempts to exceed this, the retransmission rate increases so fast that less information gets through.

The basic assumption in this analysis is that line traffic, including retransmissions, forms a Poisson process. Simulation studies[12] show that eq. (7) holds up to $N\lambda m \cong 0.1$. For higher loadings, this simulation shows a much higher retransmission rate and, presumably, the Poisson assumption does not hold.

A basic quantity in the calculation of delay is the time that the User Station waits before retransmission if no positive acknowledgment for a prior transmission is received. Let us consider how this 'timeout' period affects the system performance that a user might experience. If two messages do interfere, then each station must initiate a retransmission. If the times of each retransmission are the same, then interference will persist. Thus some strategy must be used to avoid persistent interference as well as to provide for a short 'timeout' delay. To more fully appreciate what is meant by 'short' we have analyzed two strategies for retransmission.

The first strategy assigns to each station a fixed or designated 'timeout' delay. This approach has the advantage that it completely avoids persistent interference. It has the disadvantage that some stations will have large (on the order of the number of stations times a fixed delay per station) delay.

The second strategy (called randomized retransmission) attempts to make use of the fact that it is generally two stations that interfere, and that it is not necessary to distinguish between all N stations in the system. This approach asks interfering stations to select a retransmission time by selecting it from a random sequence of retransmission times. If each station has a different sequence, then there is a small probability of persistent interference. The approach has the advantage of shortening the retransmission delay, but has the disadvantage that there is a non-zero probability of repeated interference.

In the next two subsections we discuss the delay analysis for these strategies. Detailed calculations may be found in Appendix B.

6.1 *Fixed Timeout Delay*

Consider first the Random Access System implemented with fixed timeout retransmission. Suppose that a message is transmitted from a User Station at time $t = 0$. Since we must allow for synchronization and for the transmission of an acknowledgment message, the message can be acknowledged no sooner than time $t = s + \{[\log_2 N] + 1\}/\delta + m$. If no acknowledgment is received at this time, station i waits for $2im$ seconds ($i = 1, 2, \cdots, N$) and retransmits. This is repeated until a positive acknowledgment is received. If we assume that interferences on successive retransmissions are independent, we have from eq. (6)

$$Pr\,[\text{message clearance time} = w + m + kT_i]$$
$$= Pr\,[k \text{ retransmissions}] = [1 - \exp(-2Rm)]^k \exp(-2Rm) \qquad (8)$$

where $w = \{[\log_2 N] + 1\}/\delta + s$ and $T_i = w + 2mi$.

Average message delay at each station as well as higher moments of delay may be calculated by calling upon results from the analysis of an M/G/1 queue.[9] Messages arrive at a Poisson rate of λ messages per second and each message has a geometrically distributed service time as given by (8). From the generating function derived in Appendix B, it can be shown that the average delay for a message at station i is

$$\bar{d}_{RA} = \frac{2[wP - 2Nm(1-P)] - \lambda[w^2P + m^2 + (2Nm)^2(1-P)]}{2\{1 - \lambda[wP + m - 2Nm(1-P)]\}} + m \qquad (9)$$

where $P = \exp(2Rm)$.

The return message delay in the random access system is the same as in the Polling System. Fixed length messages of M bits return at rate $N\lambda$ messages per second, with $C_R - \delta$ bits per second capacity available for transmission. The resulting message delay is given by eq. (5). The roundtrip delay is the sum of $\bar{d}_{RA}$ [eq. (9)] and $\bar{d}_R$ [eq. (5)].

6.2 *Randomized Retransmission Delay*

We turn now to consider a random retransmission technique. After an initial transmission, the User Station waits a minimum of $s + ([\log_2 N] + 1)/\delta$ seconds. If no acknowledgment has been received, the message is retransmitted after a randomly-distributed time interval. If no acknowledgment is received after this second transmission, the process is repeated. In our calculations we have taken the random timeout interval to be exponentially distributed with mean $1/\alpha$.

The probability of interference on the initial transmission is given by eq. (6). On subsequent retransmissions the probability of interference depends on α. For example, if α is very large, then retransmission for the two interfering stations occurs almost immediately after

$$t = s + \frac{[\log_2 N] + 1 + m}{\delta}$$

and the probability of interference is high. As α decreases, the probability of interference decreases. We approximate the probability of interference on retransmission as

$$Pr\,[\text{retransmission interference}] = 1 - \exp\,[-2m(R + \alpha)]. \tag{10}$$

The average time between retransmissions is

$$\bar{t}_r = s + \{[\log_2 N] + 1\}/\delta + 1/\alpha + m. \tag{11}$$

In Appendix B, the generating function of the time required to clear a message from a station's buffer is derived. From this we can show that the mean and the mean square times to clear a message from a station buffer are respectively

$$\bar{b} = w \exp\,(-2Rm) + X(1/\alpha + 2w) \exp\,\{+4m(R + \alpha)\} + m \tag{12a}$$

$$\overline{b^2} = w^2 \exp\,(-2Rm) - \frac{2Xw^2}{\alpha \exp\,\{-4m(R + \alpha)\}} + \frac{2X(1 + 2w\alpha)(1 + w\alpha + w\alpha Y)}{\alpha^2 \exp\,\{-6m(R + \alpha)\}} + 2m\bar{b} + m^2 \tag{12b}$$

where

$$X = [1 - \exp\,(-2Rm)] \exp\,\{-2m(R + \lambda)\}$$

and

$$Y = 1 - \exp\,\{-2m(R + \lambda)\}.$$

From the theory of the M/G/1 queue, it can be shown that the average delay is

$$\bar{d}_{RA} = \bar{b} + \frac{\alpha\overline{b^2}}{2(1 - \alpha\bar{b})} + m \tag{13}$$

where $\bar{b}$ and $\overline{b^2}$ are given by eqs. (12a) and (12b).

Notice from (10) and (11) that decreasing α reduces the probability of interference while increasing the average retransmission delay. Thus there is an optimum value of α that balances these effects, yielding a minimum average delay.

VII. LOOP SYSTEM

The configuration of User Stations and Central Processor for the Loop System is shown in Fig. 2. As in the Polling and Random Access Systems, the line between User Stations can be looped back to the Central Processor. Traffic flow on the line is in terms of fixed size message slots. Messages arriving at a User Station are multiplexed into these slots, one message to each slot. Traffic that is already on the line has priority; consequently a User Station must wait for an empty message slot. Traffic returning from the computer is addressed. At each station these addresses are examined. If a message is addressed to a particular station, it is taken off the line. Notice that in the Loop System, return and forward messages share the same line, consequently the total line length is half that required for Polling and Random Access Systems serving the same User Stations.

Messages arrive at each User Station at a Poisson rate of λ messages per second. These messages are multiplexed on a first come first served basis. If a message arrives when the buffer is empty, it still must wait until the line is free before it can be multiplexed.

Analytical and simulation results for this system have been presented in Refs. 13, 14, and 15. It was shown that the average message delay is approximated by the expression.

$$t_L = m(1 + \rho_L) + \frac{m\rho(1 + \rho_L)^2}{2[1 - \rho(1 + \rho_L)]} + \frac{\overline{l^2}}{\bar{l}}\,\frac{\rho_L}{2(1 + \rho_L)[1 - \rho(1 + \rho_L)]} \tag{14}$$

where ρ_L is the ratio of the average durations of line busy and idle periods. $\bar{l}$ and $\overline{l^2}$ are respectively the mean and the mean square values of the durations of the line busy periods. In deriving eq. (14) we make use of the fact that the message length is a constant value of m seconds.

A crucial quantity in the calculation of message delay is the line busy period. The characterization of the line busy period is simplified somewhat if we look at delay for what is usually the most critical station. In many applications the User Station receives more data than it transmits and as a consequence the traffic diminishes as one moves around the loop from the Central Processor.

We have modeled the traffic to the Central Processor as consisting of Poisson arrivals of fixed length messages. Messages arrive at rate $N\lambda$ messages per second and $M = B_R/C_R$ seconds are required to transmit each message. The busy period of the line out of the Central Processor is the busy period of an M/D/1 queue. It can be shown that[9]

$$\bar{l} = \frac{B_R/C_R}{1 - \rho_R} \tag{15a}$$

and

$$\overline{l^2} = \frac{(B_R/C_R)^2}{(1 - \rho_R)^3} \tag{15b}$$

where

$$\rho_R = B_R \lambda N / C_B .$$

The average duration of a line idle period is $1/N\lambda$. We have then

$$\rho_L = \frac{B_R N \lambda / C_R}{1 - \rho_R} = \frac{\rho_R}{1 - \rho_R}. \tag{16}$$

In order to provide a valid comparison with the other two systems, we must consider the processing time of a message at each station. A message, generated at the first station after the computer, passes through all of the other stations on the loop. At each station the address of the message is examined, entailing a delay of $\{[\log_2 N] + 1\}/C_R$ where $[x]$ is the largest integer less than x. Notice that since each station is receiving continuously from the same adjacent station, there is no synchronization delay. The cumulative delay of a message in going from the first User Station to the Central Processor is then

$$\bar{d}_L = \bar{t}_L + N([\log_2 N] + 1)/C_R \tag{17}$$

where $\bar{t}_L$ is given by eq. (14).

The return message delay for the Loop System is similar to the return message delay in the previous two systems. The delay in seconds is given by eq. (5) with $\delta = 0$. Thus the roundtrip delay is the sum of $\bar{d}_L$ [eq. (17)] and $\bar{d}_R$ [eq. (5)].

VIII. EXAMPLES OF SYSTEM BEHAVIOR

In this section, results of computations using the equations for *average* delay derived in the foregoing are presented. In presenting these results we were faced with the difficulty of choosing sets of parameters that provide meaningful comparisons. There is such a wide latitude in the choice of parameters for each of the three systems that one could easily bury the reader in a mass of curves and tables. Therefore, we have limited ourselves to relatively few cases illustrating system behavior. It is not difficult to supplement the results we present here since the expressions we have derived for average delay are relatively easy to evaluate.

Calculations have been made using values of user-related parameters (λ, B_F and B_R) which are appropriate in an Inquiry-Response context. Values for those parameters related directly to implementation (C_F, C_R and s) are chosen with the Digital Data System[2] in mind.

The results of the computations are shown in the form of sets of curves where average message delay is shown as a function of the number of user stations in the system. The rate at which messages arrive at User Stations, λ, is given two values, 50 calls per busy hour and 500 calls per busy hour, representing, respectively, light and heavy calling rates. In the application we have in mind, messages in the forward direction tend to be short. We have taken this message length to be $B_F = 250$ bits. Three values were used for the lengths of messages returning from the computer 2.5×10^3, 5×10^3, and 10^4 bits. We note that this latter value is approximately the number of bits required to fill a CRT display.

Since the systems we consider achieve economies by sharing, it is reasonable to put as many stations as possible in a system by choosing high line capacities C_F and C_R. In the Digital Data System, for example, 56 kbits per second are available transmit to local stations over short distances. Thus we take $C_R = C_F = 56 \times 10^3$ bits per second. This choice is tempered by the fact that, in order to achieve this high rate, synchronous operation is required. As we have noted in connection with the Polling and Random Access systems, the Central Processor receives data from different stations on each transmission. A low estimate for the time required to adjust synchronization from reception to reception at this speed is $s = 10$ ms. As we shall see, this value of s may lead to high values of delay for the Polling System. Therefore, for comparison we examine Polling Systems where transmission in the forward direction is asynchronous with $C_F = 2400$ bits per second and $s = 1/2400$ seconds. We have also examined fully synchronous operation where $s = 0$ and $C_F = 56 \times 10^3$ bits per second.

The parameter δ comes into play in the Polling and the Random Access system. Recall that δ is the portion of the return channel capacity allocated to the transmission of polling messages (Polling System) or positive acknowledgments (Random Access System). By varying this parameter, message delay in the forward direction is traded off against message delay in the reverse direction. For any particular system configuration, there is an optimum value for δ. However we have examined the effect of varying this parameter by choosing $\delta = 10^2$, 10^3 and 10^4 bits per second.

Sets of curves of delay in the forward direction for the Polling System are shown in Figs. 3, 4 and 5. In Figs. 7 and 8, return delay is shown as a function of the number of stations. As one might expect, traffic characteristics are important in judging the merits of implementations of the Polling System. For $\lambda = 50$ calls per busy hour, the 2,400 bps implementation (Fig. 5) yields lower delay than in the 56 kbps system

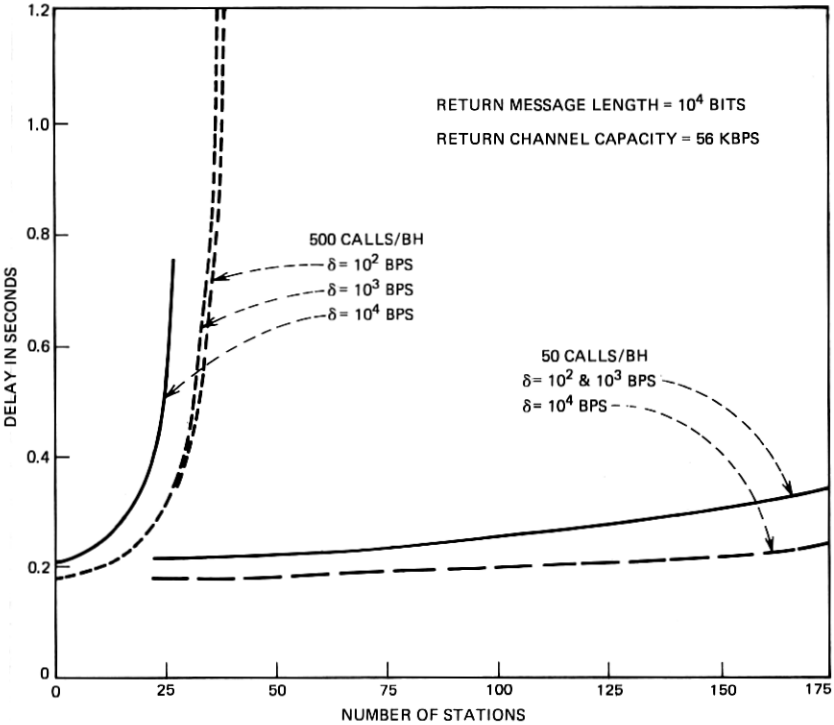

Fig. 8—Polling and random access system return delay.

with 0.01-second synchronization time (Fig. 3) in almost all cases. In fact, at this calling rate the fully synchronous 56 kbps system (Fig. 4) has delay which is lower than that of the 2,400 bps system by a relatively small amount. This advantage disappears under the heavy calling rate (500 calls per busy hour). The lower speed implementation is far more sensitive to calling rates.

Delay is sensitive to the parameter δ, which is that portion of the return channel allocated to polling messages. It is clear from that, for the traffic we consider, $\delta = 10^2$ bps is entirely too small. By increasing δ to 10^3 bps, there is a large decrease in forward delay and a small increase in return delay. It is not clear that further advantage is obtained if δ is increased to 10^4 bps. On Figs. 8 and 9, return delay may be very large for $\delta = 10^4$ bps and, for the same number of stations, low for $\delta = 10^3$ bps (e.g., $N = 130$ stations, $M = 2.5 \times 10^3/56 \times 10^3$ seconds on Fig. 9). For any given set of traffic characteristics there is an optimum value of δ which minimizes total delay.

The results of the computation of average forward delay for the Random Access System are shown on Figs. 6 and 7. The curves of average return delay are the same as for the Polling System and are shown on Figs. 8 and 9. Unlike the Polling System, the effect of s and δ does not accumulate with the number of stations, and consequently is not very sensitive to these parameters. The Random Access System is more sensitive to calling rate than the Polling System primarily because calling rate affects the probability of a message being retransmitted [see eqs. (6) and (7)]. In Fig. 6, the results shown are for the station with the longest fixed timeout interval ($i = N$) in eq. (11).

The fixed-time retransmission implementation of the Random Access System (see Fig. 6) compares very well with the Polling System. For example, for $s = 0.01$ seconds, $\delta = 10^3$ bits per second and $\lambda = 500$ calls per busy hour, the 100 station delay in the Polling System is nearly 0.9 seconds, (see Fig. 3) while the corresponding delay in the Random Access System is less than 0.2 seconds (see Fig. 6). The Random Access System with fixed-time retransmission also performs well in comparison with the $s = 0$ implementation of the Polling System.

Recall that in the Random Access random retransmission strategy, the timeout interval is exponentially distributed with mean $1/\alpha$. By a process of trial and error we have found that $\alpha = N\lambda$ yields a rough minimum of average delay for $N > 10$ stations. This value was used

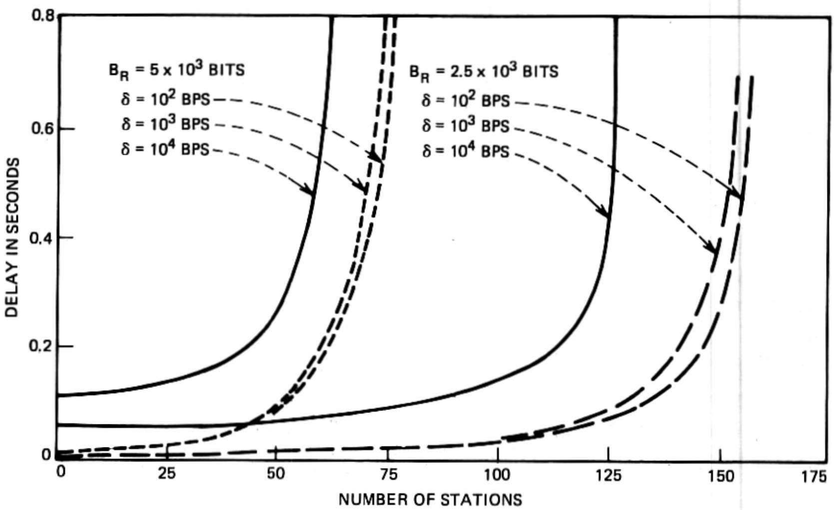

Fig. 9—Polling and random access system return delay; $\lambda = 500$ calls/BH.

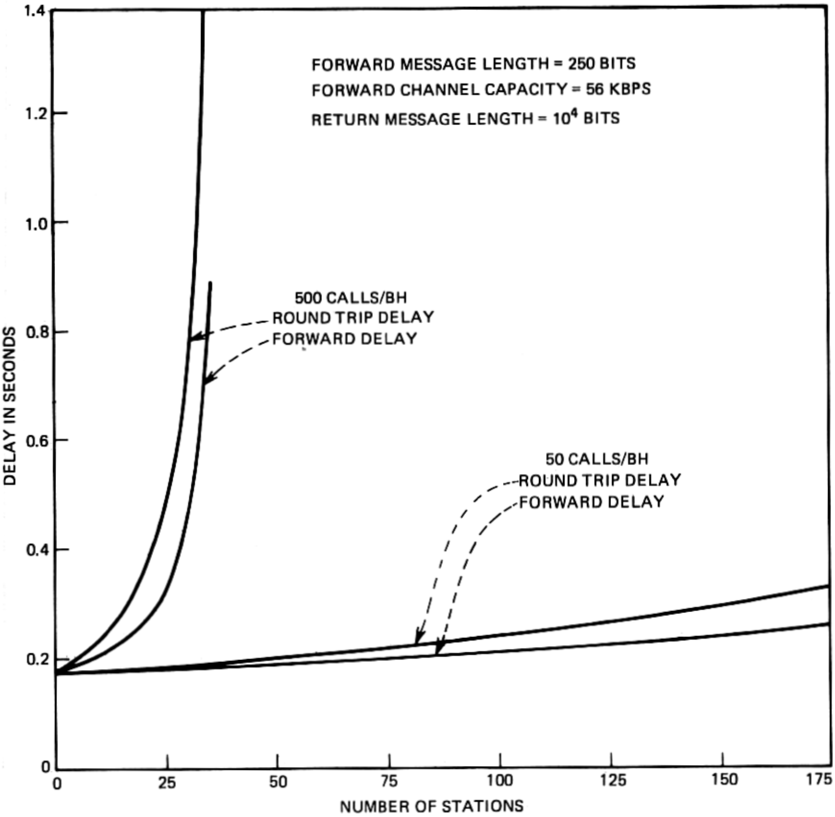

Fig. 10—Loop system forward and roundtrip delay.

to obtain the results shown in Fig. 7. As we see from Figs. 6 and 7, this realization of the Random Access System yields performance that is superior to fixed-time retransmission. Random retransmission compares favorably with the best implementation of the Polling System (see Fig. 4).

Our comparison may be biased somewhat in favor of the Random Access System since we have not taken into account overhead in that system. Recall that in order to detect errors, parity check bits along with information bits may be transmitted from the User Station. This will lengthen the message from the 250 bit message we have considered. A larger value of m will cause increased delay by increasing the probability of retransmission [see eqs. (6) and (7)] and by increasing the timeout interval [see eq. (11)]. However, we felt that it would not be necessary to lengthen the message very much, and consequently delay would be approximately the values we have shown.

The results of average delay calculations for the Loop System are shown on Figs. 9 and 10. Forward and roundtrip delay for the station on the line immediately after the Central Processor are shown. Roundtrip delay for the Loop System is even more sensitive to return message duration than the Polling or Random Access Systems. This is because messages going from the user terminal to the Central Processor are blocked by return messages. However, in the stable region below the knee of the delay curve, the Loop System gives performance comparable to the best implementation of the Polling and Random Access Systems. For example, for $M = 2.5 \times 10^3/5.6 \times 10^4$ seconds, $\lambda = 500$ calls per busy hour, and $N = 100$ stations, the following average roundtrip delay estimates are obtained for each of the three systems: Loop System: 0.12 seconds (Fig. 11), Polling System: 0.16 seconds (Figs. 4 and 9), and Random Access System: 0.16 seconds (Figs. 7 and 9).

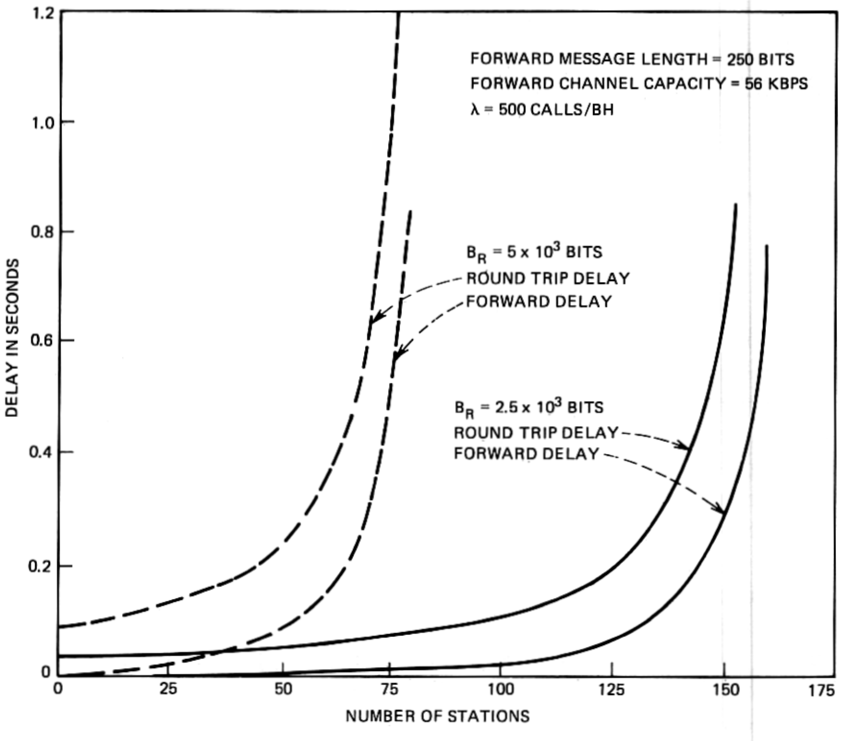

Fig. 11—Loop system forward and roundtrip delay.

IX. ACKNOWLEDGMENTS

We would like to thank R. J. Pilc for initial suggestions on the applicability of polling to local distribution, and M. Eisenberg for his assistance in developing the structure of the queueing model in the polling system.

APPENDIX A

In this appendix the moment generating function of forward delay in a Polling System is derived. Although other schedules may be analyzed, we concentrate on the situation where each station is polled once and only once in a cycle. Messages arrive for multiplexing at the User Stations at a Poisson rate of λ messages per second. In carrying out the derivation, we shall denote the generating function of the message length in seconds as $M(u)$. The generating function of the walk-time is denoted as $W(u)$. In the text, we shall apply our results to the case where message length and walk times are constants.

We consider first the "cycle time" of the polling sequence. This quantity is the time interval between polls of a particular station. As the polling sequence goes through a complete cycle, a random number of messages is encountered in each station's buffer. The analysis is simplified considerably if we assume that this number of messages is independent and identically distributed from station to station. Under this assumption, Leibowitz[7] shows that the moment generating function of the number of messages in each station's buffer at the time of polling is given by

$$P(x) = \{W(\lambda - \lambda x)P(\phi)\}^N \tag{18}$$

where $\phi = M(\lambda - \lambda x)$.

Since messages arrive at a Poisson rate, a relationship between the number of messages in the buffer and the cycle time can be derived.

$$Pr\ [n \text{ messages in a buffer at polling time cycle time } = \tau] = \exp(-\lambda\tau)(\lambda\tau)^n/n!.$$

Averaging over the cycle time we have

$$Pr\ [n \text{ messages in a buffer at polling time}] = \int_0^\infty \frac{\exp(-\lambda\tau)(\lambda\tau)^n}{n!}\, p_c(\tau)\, d\tau$$

where $p_c(\tau)$ is the probability density of the cycle time. By taking the Laplace-Stieltjes transform here it can be shown that,

$$P(x) = T_c(\lambda - \lambda x) \tag{19}$$

where $T_c(s)$ is the moment generating function of the cycle time. From (18) and (19) we have

$$T_c(u) = [W(u)T_c(\lambda - \lambda M(u))]^N. \tag{20}$$

Differentiating (20) and setting $u = 0$ yields the results on mean and mean square delay shown in eqs. (1a) and (1b) respectively.

Now a message arriving at a station must wait until the station is polled, and until all messages that have arrived before it have been multiplexed on the line. We first derive the generating function of the time the message must wait in the queue. We shall work under the assumption that messages arriving while prior messages are being multiplexed must wait until the next poll. Thus the queueing time of a customer can be written

$$d_P = \tau + \sum_{i=1}^{k} m_i \tag{21}$$

where τ is the time interval until the next poll and m_i , $i = 1, 2, \cdots, k$ are the lengths of k prior messages in the buffer. From (21) we can write the density of the delay d_P as

$$P(T) = \Pr[T < d_P \leqq T + dT] = \int_0^\infty d\tau \sum_{k=0}^{\infty} p_m^{(k)}(T - \tau)p(k, \tau) \tag{22}$$

where $p_m(s)$ is the probability density of the message length and $p(k, \tau)$ is the joint density of k and τ. It can be shown from elementary probabilistic arguments that

$$p(k, \tau) = \int_\tau^\infty dx \frac{p_c(x)}{\bar{t}_c} \frac{\lambda(x - \tau)^k}{k!} \exp\{-\lambda(x - \tau)\} \tag{23}$$

where $p_c(x)$ is the probability density of the cycle time. From (22) and (23) it can be shown that the moment generating function of the queueing delay is

$$T_{PQ}(u) = \frac{T_c\{\lambda - \lambda M(u)\} - T_c(u)}{\bar{t}_c[u - \lambda + \lambda M(u)]} \tag{24}$$

where $T_c(u)$ is the moment generating function of the cycle time. In order to find the total message delay, we must add the multiplexing time to the queueing delay. Since these quantities are independent random variables, the moment generating function of the message delay is

$$T_P(u) = T_{PQ}(u)\cdot M(u). \tag{25}$$

By differentiating $T_P(u)$ with respect to u and setting $u = 0$, the expression for delay given by eq. (2) of the text can be found.

APPENDIX B

In this appendix the generating function of message delay in the forward direction for the Random Access System is derived. Recall that after transmitting a message, a User Station waits for a positive acknowledgment from the Central Processor. The minimum time required to receive a positive acknowledgment is

$$w = \frac{[\log_2 N] + 1}{\delta} + s. \tag{26}$$

If no acknowledgment is received, the original message is retransmitted. We consider two transmission strategies; a fixed transmission time which is different for each station, and a random transmission time.

Under the fixed retransmission strategy, each station's transmission interval is different by at most $2m$ seconds, thus the retransmission interval for the ith station is $2mi$. This interval obviates the possibility of the same two stations repeatedly interfering with one another. If k retransmissions of a message are necessary to clear a message from the station's buffer, the total clearance time is

$$\tau = w + m + k(w + mi) \tag{27}$$

for the ith station.

If message flow on the line is Poisson, then the probability of two messages interfering with one another is

$$\Pr\,[\text{interference}] = 1 - \exp\,(-2Rm). \tag{28}$$

If we assume that subsequent retransmissions are independent trials, the total number of trials required to clear a message is geometrically distributed. We have

$$\Pr\,[k \text{ retransmissions}] = [1 - \exp\,(-2Rm)]^k \exp\,(-2Rm). \tag{29}$$

From (27) and (29) the moment generating function of the time required to clear a message is

$$S_{1i}(u) = E\{\exp\,(-u[w + m + kT_i])\} = \exp\,(-um)[\exp\,(uw + 2Rm) - \exp\,(2Rm - 2uim) + \exp\,(-2uim)]^{-1} \tag{30}$$

where

$$T_i = w + 2mi.$$

In the language of queueing theory, $S_{1i}(u)$ corresponds to the generating function of the service time of a customer (message). Messages arrive for multiplexing at a Poisson rate of λ messages per second. The message delay can be found from the theory of the M/G/1 queue. We have

$$D^i_{RA}(u) = \frac{(1 + \lambda S'_{1i}(0))uS_{1i}(u)}{u - \lambda + \lambda S_{1i}(u)}. \tag{31}$$

By substituting (30) into (31), differentiating with respect to u and setting $u = 0$, the expression for the mean value of delay given in eq. (9) can be found.

The derivation of the generating function of message delay for random retransmission proceeds along the similar lines. The retransmission interval is a geometrically-distributed random variable with mean value $1/\alpha$ for each station. If k retransmissions are required, then the time to clear a message is

$$\tau = (k + 1)w + m + \sum_{j=1}^{k} \xi_j \tag{32}$$

where ξ_j, $j = 1, 2, \cdots, k$ are exponentially distributed random variables. The distribution of k is slightly different than in the case of fixed time-out retransmission. If two stations have interfered, there is a non-zero probability that they will interfere on the subsequent retransmission. We account for this phenomenon by approximating the probability of retransmission by the expression

$$\text{Pr [interference on retransmission]} = 1 - \exp(-2(R + \alpha)m). \tag{33}$$

The probability of no retransmissions is

$$\text{Pr [no retransmission]} = \exp(-2Rm). \tag{34}$$

The probability of k retransmissions is

$$\text{Pr [}k\text{ retransmissions]} = [1 - \exp(-2Rm)][1 - \exp(-2(R + \alpha)m)]^{k-1} \cdot \exp(-2m(R + \alpha)); \quad k = 1, 2, \cdots. \tag{35}$$

From (31), (33) and (34) it can be shown that the generating function of message clearance is

$$S_{2i}(u) = E\left[\exp\left(-(k+1)w + m - \sum_{j=1}^{k} \xi_i\right)u\right]$$

$$= \exp(-um)\left[\exp(-2Rm - wu) + \frac{\lambda[1 - \exp(-2Rm)]\exp(-2m(R+\alpha))\exp(-2wu)}{\lambda + u - \lambda\exp(-wu)[1 - \exp(-2m(R+\alpha))]}\right]. \tag{36}$$

The generating function of message delay can be found from the theory of the M/G/1 queue. The generating function of message delay is the same as in (31) with $S_{1i}(u)$ replaced by $S_{2i}(u)$. The first two moments of the time needed to clear a message given by (12a) and (12b) are found from successive differentiations of (36). The formula for average delay given in (13) is well known.[9]

REFERENCES

1. Pierce, J. R., "Network for Block Switching of Data," IEEE Conv. Rec. (March 1971).
2. Muench, P. E., "Bell System Private Line Digital Data Service," IEEE Conv. Rec. (March 1971).
3. Jackson, P. E., and Stubbs, C. D., "A Study of Multiaccess Computer Communications," AFIPS, Conf. Proc., *34*, p. 491.
4. Jackson, P. E., and Fuchs, E., "Estimates of Distributions of Random Variables for Certain Computer Communications Traffic Models," Proc. ACM Conf., Pine Mountain, Georgia (October 1969).
5. Martin, J., *Teleprocessing Network Organization*, Englewood Cliffs: Prentice-Hall Inc., 1970.
6. Martin, J., *Design of Real-Time Computer Systems*, Englewood Cliffs: Prentice-Hall Inc., 1967.
7. Leibowitz, M. A., "An Approximate Method for Treating a Class of Multiqueue Problems," IBM J. Res. Development, *5*, No. 3, 1961, pp. 204–209.
8. Eisenberg, M., "Queues with Periodic Service and Changeover Time," to be published, J. Oper. Res.
9. Cox, D. R., and Smith, W. L., *Queues*, Methuen and Co. Ltd., 1961, pp. 52–59.
10. "The ALOHA System," University of Hawaii Technical Report, 1969, AFOSR 70-0416 TR, AD 701417.
11. Abramson, N., "The ALOHA-System-Another Alternative for Computer Communications," University of Hawaii Technical Report B70-1, April 1970, AD 707853.
12. Bortels, W. H., "Simulation of Interference of Packets in the ALOHA Time-Sharing System, University of Hawaii Technical Report B70–2, March 1970.
13. Hayes, J. F., and Sherman, D. N., "Traffic Considerations of A Ring Switched Data Transmission System," B.S.T.J. *50*, No. 9 (November 1971), pp. 2947–2978.
14. Anderson, R. R., Hayes, J. F., and Sherman, D. N., "Simulated Performance of a Ring Switched Data Network," *COM-20*, No. 3, IEEE Trans. on Comm. Tech. (June 1972).
15. Avi-Itzhak, B., "Heavy Traffic Characteristics of a Circular Data Network," B.S.T.J. *50*, No. 8 (October 1971), pp. 2521–2549.