Communications Research Centre

THE EFFECT OF PROTOCOL ON THE RESPONSE TIME OF LOOP STRUCTURES FOR DATA COMMUNICATIONS

by

T.G. RICHARDSON

DEPARTMENT OF COMMUNICATIONS
MINISTÈRE DES COMMUNICATIONS

CRC REPORT NO. 1261

OTTAWA, AUGUST 1974

COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS
CANADA

THE EFFECT OF PROTOCOL ON THE RESPONSE TIME OF LOOP STRUCTURES FOR DATA COMMUNICATIONS

by

T.G. Richardson

(Communications Systems Directorate)

August 1974

CRC REPORT NO. 1261

OTTAWA

TABLE OF CONTENTS

THE EFFECT OF PROTOCOL ON THE RESPONSE TIME OF LOOP STRUCTURES FOR DATA COMMUNICATIONS

by

T.G. Richardson

ABSTRACT - *The loop or ring structure for data communications is considered under three different operating protocols. An inquiry-response type traffic pattern is assumed and response time is chosen as the measure of performance. Each loop protocol is analyzed through the use of approximate theoretical queueing formulae and these results are compared to those of a simulation. Conclusions are drawn as to the relative merits of the three alternatives. Mean response time is found to be about equal over the range of interest but important differences are observed in individual terminal behavior.*

1. INTRODUCTION

Recently[2-7,12-13] much attention has been directed at the loop or ring structure as a foundation for message switching in data communications. Typically information is transmitted unidirectionally from terminal to terminal around a loop. There may be nodal points on the loop which interconnect nets or hierarchies of loops, or other types of systems. Within the loop structure, a variety of protocols exist for transferring information from the terminal to the line[3,5-6,12-13]. The object of this report is to compare three of the most promising loop protocols, with mean response time chosen as the measure of performance. We consider N terminals and a single computer on a loop, with each terminal sending requests to the computer for processing and replies being returned from the computer to the terminal. The response time is the elapsed time between the message being loaded into the terminal-send-buffer and the reply being received completely in the terminal-receive-buffer, but excluding computer processing time. In the data communications environment envisaged here, i.e., short inquiry-type messages from remote terminals to a central computer such as occur in reservation, credit-verification, logistics or banking systems, the response time is acknowledged to be an important design parameter. The effectiveness of loops has been exploited for some time; hub polling, which is essentially a loop mode of operation, has been used in several airline reservation systems[14] because of the short response times achievable by loops.

The three loop protocols that are to be analyzed may be categorized as follows:

i) "Pass Control" Loop (PC Loop);

ii) "Slot Deletion" Loop (SD Loop);

iii) "Slot No Deletion" Loop (SND Loop).

In the PC Loop, as the name implies, control is passed from terminal to terminal around the loop. Only one terminal can be transmitting data on to the loop at a given point in time. When it has finished it passes control to the next terminal. This system has been proposed by Farmer and Newhall[3] and analyzed by Kaye[4] and Kaye and Richardson[5].

In the SD Loop a central controller generates fixed-length time slots which travel around the loop. If a terminal sees an empty slot, it may fill it with a message. The intended receive terminal takes the message and deletes the contents of the slot so that it can be re-used by itself or another terminal in the loop.

The SND Loop is again a slotted system but terminals, upon receipt of a message, copy the contents of the slot into their buffer and do not delete it. Thus slots cannot be re-used until their contents are deleted by the central controller. Slot loops have been proposed by Pierce[15] and analyzed by Hayes and Sherman[2,6], Spragins[5] and others.

Kaye and Richardson[1] have compared the PC loop to other conventional polling systems while Hayes and Sherman[2] have compared the SD Loop to the ALOHA random-access polling system and a shared polling system in which a portion of the return line (from computer to terminal) is dedicated to polls. In Ref. 1 it is shown that the PC Loop is definitely superior to conventional systems for short messages, or more precisely when $\rho m < D$, where ρ is the traffic intensity on the line, m is the message length and D is the fixed overhead time to poll all terminals (including synchronization times, poll signals, etc). Hayes and Sherman's analysis will be extended here to show that in fact the same conclusion holds for the SD Loop in relation to more conventional systems. Thus, evidence indicates that, for short inquiry type traffic, loops have an advantage in response time over other systems. Here we will compare the characteristics of the three loop systems to determine the advantages and disadvantages of each.

In the following three sections the loops will be treated in turn. The analysis will include a more detailed description of loop protocol, theoretical approximations to account for delay, and a comparison of these estimates with results from computer simulations. In Section 5 comparative results are presented and conclusions drawn as to performance. The appendix summarizes the methodology used to perform computer simulations of several systems.

Listed below is a glossary of terms used in the text. The double subscript "fi" refers to *forward* messages, from the i^{th} terminal to the computer, while subscript "ri" indicates messages *returned* from the computer to the i^{th} terminal. Subscripts f and r alone denote a mean quantity for all N terminals. In general, lower-case characters refer to terminal or computer quantities while upper case will be used for line quantities. All rates are normalized in bit times.

m = $m_f + m_r$ = sum of forward and return message lengths (bits)

λ = input traffic arrival rate ($bits^{-1}$)

ρ = traffic intensity

D = fixed overhead (bits)

N = number of terminals

$M_i, M_i^{(2)}$ = first and second moments of mean message length on the line at terminal i (bits)

Y_i = traffic arrival rate on the line at terminal i ($bits^{-1}$)

R_i = traffic intensity on line at terminal i

$L_i, L_i^{(2)}$ = first and second moments of busy period on the line at terminal i (bits)

T_s = slot length (bits)

In the analysis performed here included are the destination and return addresses and operation code information in the message length. Thus the traffic intensity ρ includes message overhead. In the case of extremely short messages this would be a significant percentage of the total message length. Throughout this report, bi-polar signalling is assumed; an extension of the analysis would be necessary for binary signalling.

2. PASS CONTROL (PC) LOOP

2.1 Loop Protocol

As pointed out in the previous section, in the PC Loop only one terminal can have control of the line. When the controlling terminal has completed the input of a message it terminates with an end-of-message code followed by a "pass control" bit which is set to indicate that the terminal has relinquished control. The next terminal on the line is free to take control. This is done by resetting the pass control bit so that following terminals are aware that control has been seized. Clearly a terminal must delay the bit stream by at least one bit in order to be able to reset the pass control bit if it wishes to seize control. Hence, if there are N terminals and a computer on a loop, control can be passed completely around the loop in N+1 bit times. This is effectively the fixed overhead D required to "poll" all terminals. If a terminal transmits a message longer than N bits the "front" of the message will be erased as it comes around to the sending terminal, which has blocked the line while transmitting. Of course the receiving terminal will have already copied the message into its receive buffer as it passed by.

There is no advantage in the destination terminal clearing or deleting the message as it passes by, since the line cannot be used past this point (even though there is no information on it) because control has not been passed on by the sending terminal. If we attempt to allow more than terminal

to have control, interference between messages can occur unless there is regulation of the spacing of the pass control bits (hence slotted systems). The disadvantage of the PC Loop is that portions of the line may be idle when they could be used. The advantage is that the fixed overhead D is at a minimum.

2.2 Theoretical Results

We assume identical Poisson distributed arrivals of messages at the N terminals. The return message arrivals from the computer to the terminals are also Poisson distributed and uncorrelated with the input message arrival times. Kaye and Richardson[1] obtained the following result for the sum of the delays at the terminals and at the computer as:

$$w \simeq \frac{D}{1-\rho} + \frac{\rho m}{(1-\rho)} \left[\left(\frac{m_f}{m} \right)^2 + \left(\frac{m_r}{m} \right)^2 \right] \tag{1}$$

Here D is the fixed delay in bits required to pass control around the loop (equal to N+1 bits), ρ is the total traffic intensity on the loop, m_f is the mean forward message length in bits inputted at the terminals, m_r is the mean return message length sent from the computer to the terminals and m is the overall mean message length. This is an approximate result to account for constant but unequal length messages emanating from the computer and terminals (the exact result of Chu and Konheim applies only to equal length messages). We conjecture that this approximate formula could be generalized to

$$w \simeq \frac{D}{1-\rho} + \frac{\rho m}{1-\rho} \left(\frac{m^{(2)}}{m^2} \right) \tag{2}$$

where $m^{(2)}$ is the second moment of message length. This reduces to the previous result with constant but unequal message lengths, but may prove to be useful for more general message length distributions. The total response time for a message is thus

$$Res = w + m + (N+1) \tag{2a}$$

which includes the sum of the two queueing delays, w, plus the time required for message transmission, which is the total message length, m, plus the (N+1)-bit delay encountered in going through all terminals and the computer. This excludes propagation delays. A simulation has been performed, the results of which are compared in Fig. 1 to the theoretical approximation. Note that the approximation is low when $m_r >> m_f$, $\rho m >> D$ and $1 > \rho >> 0$ (see Ref. 1 for an explanation of this).

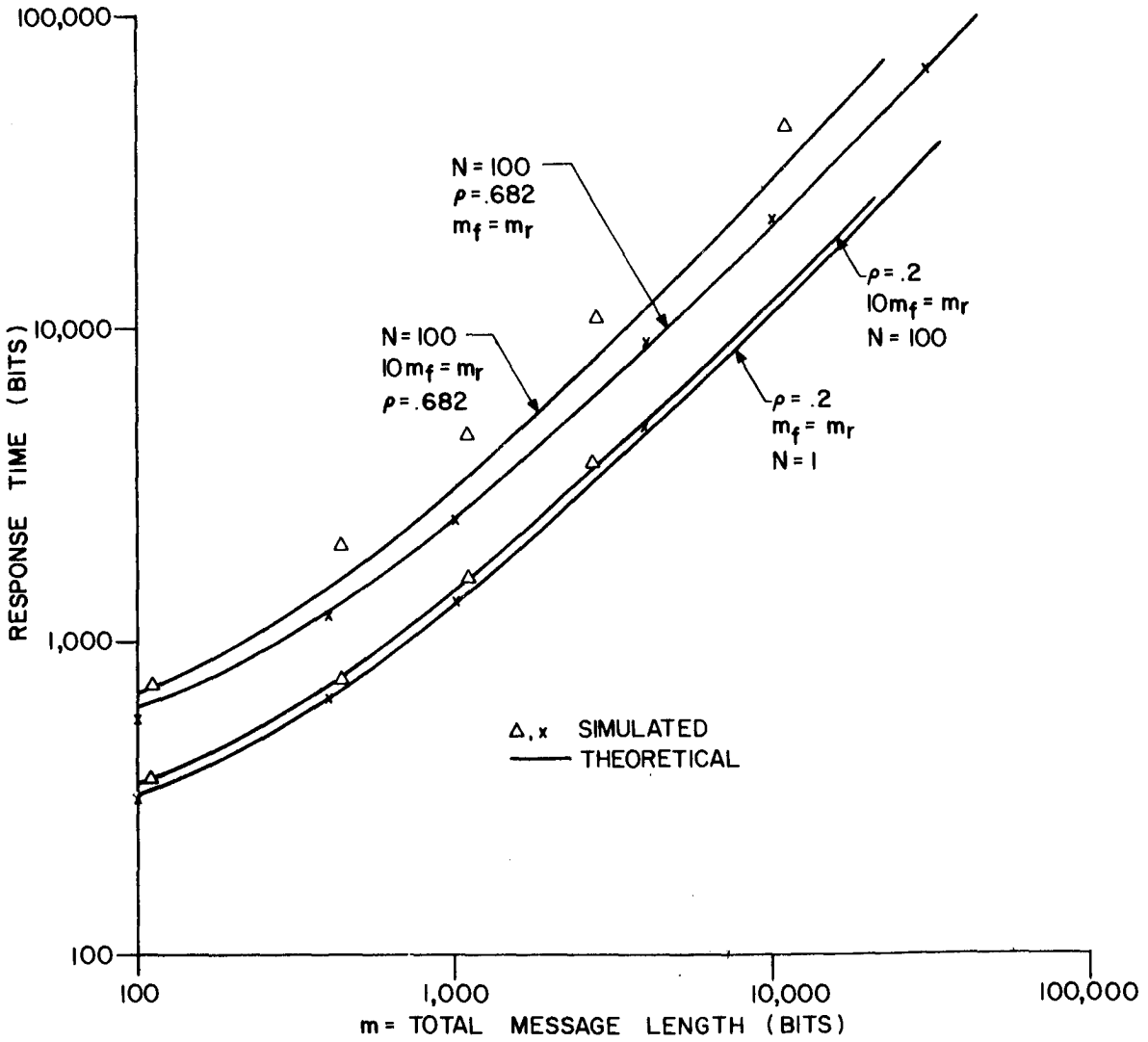

Fig. 1. Pass control (PC) loop, equation 2A with D = N+1

It should be noted that this loop and the others are viewed as synchronous digital systems. The loop controller provides synchronization timing when there is no traffic on the loop. Hence there is no modem set-up time or terminal-to-computer character synchronization time. This results in a significant reduction of the fixed overhead D; in the case of the PC loop we need only 1 bit per terminal.

3. SLOT DELETION (SD) LOOP

3.1 Loop Protocol

In the SD loop the contents of a slot are deleted by the receiving terminal. With N terminals and a computer on a loop, an address length of $[\log_2 N+2]$ bits is required† where [X] is the least integer greater than or equal to X. Thus the terminal must delay the slot by $[\log_2 N+2]$ bits in order to hold and delete the address and slot contents as the slot passes by. Thus the fixed overhead for a "poll" of all terminals is $D = (N+1)\ [\log_2(N+2)]$. For N = 100, for example, this is 707 bits, much greater than the 101 bits incurred in polling a PC loop. However, the SD loop allows more than one terminal to transmit data at the same time. Thus line efficiency is increased and long messages do not necessarily delay the input of other messages as they can in the PC loop.

3.2 Theoretical Results

3.2.1 Method "A"

To calculate the delay encountered by a message waiting at a terminal we will first build an approximation based on an alternating queue formula of Takacs[10] and a vacation time analysis for the M/D/1 queue of Cooper[9].

We assume that the computer is the first device on the loop after the central controller which is issuing free slots, followed by terminals 1,...,N. Now if we assume Poisson distributed arrivals of return messages at the computer and messages of constant length, the computer queue is an M/D/1 queue with a server vacation time of T_s bits, the slot length. (The server arrives at time t_o and does not return until time $t_o + T_s$.) The delay encountered by a message is then

$$w_r = \frac{\rho_r m_r}{2(1-\rho_r)} + \frac{T_s}{2} \qquad (3)$$

where the first term is the ordinary M/D/1 delay and $T_s/2$ accounts for the fact that on average the message which starts in a busy period arrives halfway through a slot or server vacation time. This is derived rigorously by Cooper[9].

The equations are stated for terminal 1. Since terminal 1 deletes return messages addressed to it, its transmit control looks for empty slots and observed a traffic intensity on the line of

$$R_1 = \rho_r - \rho_{r1}$$

† *This is because the address field will contain an "available" code when the slot is empty.*

Similarly the traffic arrival rate at this point is

$$Y_1 = \lambda_r - \lambda_{r1}$$

and the second moment of message length is

$$M_1^{(2)} = \left(\sum_{k=2}^{N} m_{rk}^2 \right) / (N-1)$$

We may regard the line at terminal 1 as a server which alternately services the line traffic "queue" until it is empty, and then services the terminal queue. The terminal queue in general is not emptied since the server must give priority to the line traffic. If it were emptied the protocol would be exactly that as defined by Takacs[10] and the formula for mean delay he develops would be exact with the addition of the half slot delay to account for the discrete nature of service starting times. In fact as an approximation to the actual situation the formula will be valid when the traffic generated at terminal 1 is small compared to the return traffic from the computer. In this case the probability of more than one message in the queue will be low, so that the queue will be emptied with high probability. The delay formula at terminal 1 is given by

$$wf_1 \simeq \frac{\rho_{f1} m_{f1}}{2(1-\rho_{f1})} + \frac{Y_1 M_1^{(2)} (1-\rho_{f1})^2 + \rho_{f1} m_{f1} R_1^2}{2(1-\rho_{f1})[(1-\rho_{f1})^2 (1-R_1)^2 - (\rho_{f1} R_1)^2]} + \frac{T_s}{2} \qquad (4)$$

The approximation is easily extended to the other terminals by defining

$$\begin{aligned}
R_i &= \rho_r - \sum_{i=1}^{k} \rho_{ri} + \sum_{i=1}^{k-1} \rho_{fi}, \\
Y_i &= \lambda_r - \sum_{i=1}^{k} \lambda_{ri} + \sum_{i=1}^{k-1} \lambda_{fi}, \\
M_i &= \left(\sum_{k=i+1}^{N} m_{rk} + \sum_{k=1}^{i-1} m_{fk} \right) / (N-1) \\
M_i^{(2)} &= \left(\sum_{k=i+1}^{N} m_{rk}^2 + \sum_{k=1} m_{fk}^2 \right) / (N-1)
\end{aligned} \qquad (5)$$

as the line quantities at terminal i. Then the delay is

$$w_{fi} \simeq \frac{\rho_{fi} m_{fi}}{2(1-\rho_{fi})} + \frac{Y_i M_i^{(2)} (1-\rho_{fi})^2 + \rho_{fi} m_{fi} R_i^2}{2(1-\rho_{fi})[(1-\rho_{fi})^2 (1-R_i)^2 - (\rho_{fi} R_i)^2]} + \frac{T_s}{2} \qquad (6)$$

Equation (6) is eqn. (4) with 1 replaced by i with the appropriate interpretation from eqn. (5) for line quantities at terminal i. Thus the total response time will be

$$\text{Res} = w_r + w_f + m + (N+1)[\log_2(N+2)] \qquad (7)$$

where w_r is given by (3), w_f is the mean of the w_{fi} over all i, and $m + (N+1)[\log_2(N+2)]$ is the transmission time including terminal delay.

3.2.2 Method B

Hayes and Sherman[6] have developed a delay formula for the first terminal downstream of the computer based on a busy period analysis. With some modification of their result, as discussed below, the forward delay for the first terminal, w_{f1}, is given by:

$$w_{f1} = Q_1 m_{f1} + \frac{m_{f1}\rho_{f1}(L+Q_1)^2}{2[1-\rho_{f1}(1+Q_1)]} + \frac{L_1^{(2)}}{L_1} \frac{Q_1}{2(1+Q_1)[1-\rho_{f1}(1+Q_1)]} + \frac{T_s}{2} \qquad (8)$$

L_1 and $L_1^{(2)}$ are the first and second moments of line busy period, which for an M/D/1 queue are

$$L_1 = \frac{M_1}{1-R_1}, \quad L_1^{(2)} = \frac{M_1^{(2)}}{(1-R_1)^3}$$

respectively. Q_1 is the ratio of line busy to line idle periods which for an M/D/1 queue is

$$Q_1 = \frac{M_1}{1-R_1} / \frac{1}{Y_1} \equiv \frac{R_1}{1-R_1}$$

Equation (8) is as given by Hayes and Sherman but with two differences. First, a half-slot delay is added as in the Method "A" analysis. Second it is assumed here that return messages to a terminal are deleted before forward messages

are inputed. Thus an input message can occupy a slot which is currently being deleted by the same terminal. This is not the case in Ref. 2.

To extend this result we simply write

$$Q_i = \frac{R_i}{1-R_i}, \quad L_i = \frac{M_i}{1-R_i}, \quad L_i^{(2)} = \frac{M_i^{(2)}}{(1-R_i)^3} \tag{9}$$

so that for terminal i

$$w_{fi} = Q_i m_{fi} + \frac{m_{fi}\rho_{fi}(1+Q_i)^2}{2[1-\rho_{fi}(1+R_i)]} + \frac{L_i^{(2)}}{L_i} \frac{Q_i}{2(1+Q_i)[1-\rho_{fi}(1+Q_i)]} + \frac{T_s}{2} \tag{10}$$

Clearly this is an approximation since one cannot claim that the events have a Poisson distribution at other than the first terminal. Furthermore there are other ways to make the extension to the formula which are also reasonable. However this method and Method A have been found to give good results. In Fig. 2 both are compared to simulation results. In Figs. 2 and 3, the lines have been fitted to the simulation points (not shown). For each simulated point corresponding values computed by each of the two methods are marked.

4. SLOT NO DELETION (SND) LOOP

4.1 Loop Protocol

In the SND loop there is no deletion of slot contents upon receipt. The contents are erased upon return to the central controller. Thus terminals need not delay slots by the address length. A delay of one bit suffices to allow a terminal to determine or indicate if a slot is empty or full. As in the PC loop this keeps overhead to a minimum since D = N+1. Furthermore it has the advantage of employing slots, namely the possibility of more than one simultaneous input. It has the disadvantage of not allowing slot re-use.

4.2 Theoretical Results

Exact results have been obtained by Spragins[5] for the case of all messages equal in length to the slot length. The forward delay for terminal i is

$$w_{fi} = \frac{m_{fi}}{2(1-P_i)(1-P_i-1)} \tag{11}$$

where

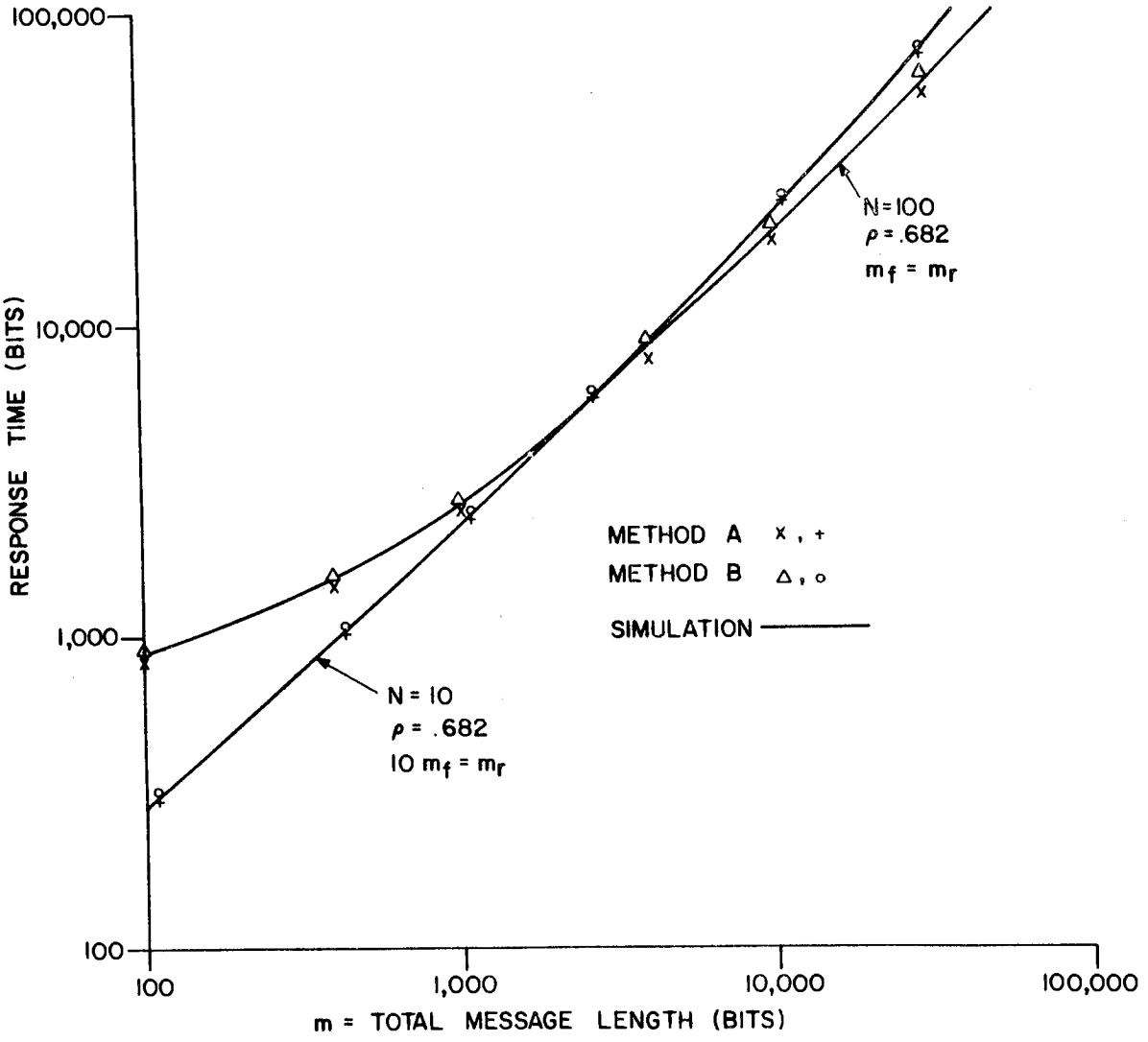

Fig. 2. Slot deletion (SD) loop, eqn. (7)

$$P_i = \rho_r + \sum_{i=1}^{k} \rho_{fi}$$

The return delay is exactly as in eqn. (3) so that total response is

$$Res = w_r + w_f + m + (N+1) \qquad (12)$$

with w_f being the mean of all w_{fi} in eqn. (11).

For unequal message lengths we found it necessary to resort to the approximate formulas as developed in methods A and B for the SD loop. We must redefine quantities to account for the no-deletion aspect as follows:

$$
\begin{aligned}
R_i &= \rho_r + \sum_{k=1}^{i=1} \rho_{fk} \\
Y_i &= \lambda_r + \sum_{k=1}^{i-1} \lambda_{fk} \\
M_i &= \left(\sum_{k=1}^{N} m_{rk} + \sum_{k=1}^{i-1} m_{fk} \right) / (N+i-1) \\
M_i^{(2)} &= \left(\sum_{k=1}^{N} m_{rk}^2 + \sum_{k=1}^{i-1} m_{fk}^2 \right) / (N+i-1)
\end{aligned}
\tag{13}
$$

These should be compared to eqn. (5). The response time result (12) is interpreted with w_f derived from (6) or (10) (methods A and B respectively), with the line quantities redefined by eqn. (13). Figure 3 compares this analysis to simulation results.

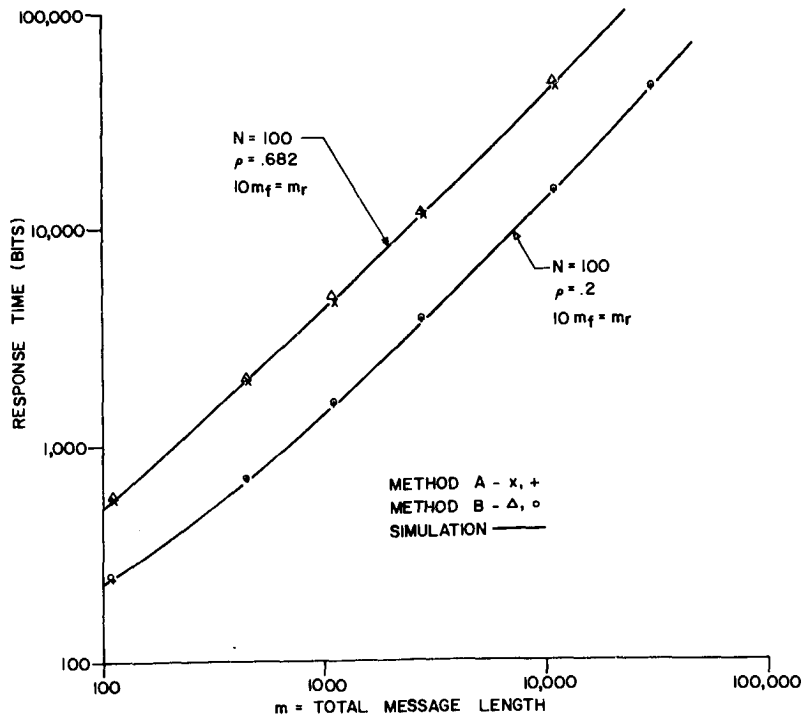

Fig. 3. Slot no deletion (SND) Loop, Eqn. (12)

5. RESULTS

5.1 Comparison of Response Times

5.1.1 Short messages

Figures 4 and 5 compare the performance of the three loops for a set of typical operating parameters. For short messages the SND slot loop system gives the best performance followed by the PC and SD protocols. Our analysis in the previous sections aids in indicating why this should be. First, the SD protocol is worst since it has relatively high overhead: for N = 100 the fixed delay is 707 bits whereas the SND and PC protocols have a fixed delay of only 101 bits. This added delay apparently predominates over the SD advantage of slot re-use. Secondly the SND protocol is better than the PC since it has the same fixed delay, but has the added advantage of allowing two or more terminals to be transmitting at once, thus decreasing overall delay.

5.1.2 Medium Length Messages

In each of the figures there is a region where performance is comparable. In the case of equal message lengths in and out from the computer (Fig. 4) comparable performance occurs for total message lengths in the order of 600 to 1000 bits. For computer output ten times the length of the input message the range is for total message lengths of about 250-700 bits (Graph 5).

5.1.3 Long Messages

Beyond the above ranges the SD slot system is best, followed by the PC and SND slot systems. In this range the advantage of the SD slot loop, slot deletion and re-use, comes to the fore by allowing simultaneous transmission. Specifically, when a long computer message is sent to low-numbered terminals these slots are re-used for input by the higher-numbered ones. Simulation shows that a high percentage of forward messages are accommodated in this way. In the SND slot loop such messages would have to wait for a slot that had not been used for return traffic *nor* filled by lower numbered terminals. The smaller overhead of the SND protocol is insignificant in this range. This is seen in eqn. (12), where, for a total message length of say 11,000 bits Res $\simeq$ 45,000 bits, while the fixed delay term N+1 is only 101 bits of this.

The advantage of the PC loop over the SND loop, assuming the same fixed overheads, is that a message arriving during an idle period must wait on the average only N/2 bits, while in the SND slot loop it must wait on the average half a slot which can be very large by comparison. On the other hand, the advantage of the SND slot loop is that it can transmit more than one message at a time on the loop. However, this benefit is marginal when the message lengths become much longer than the fixed delay around the loop; analysis shows that multiple transmission of long messages occurs infrequently, and that when it does occur the saving in delay is not significant.

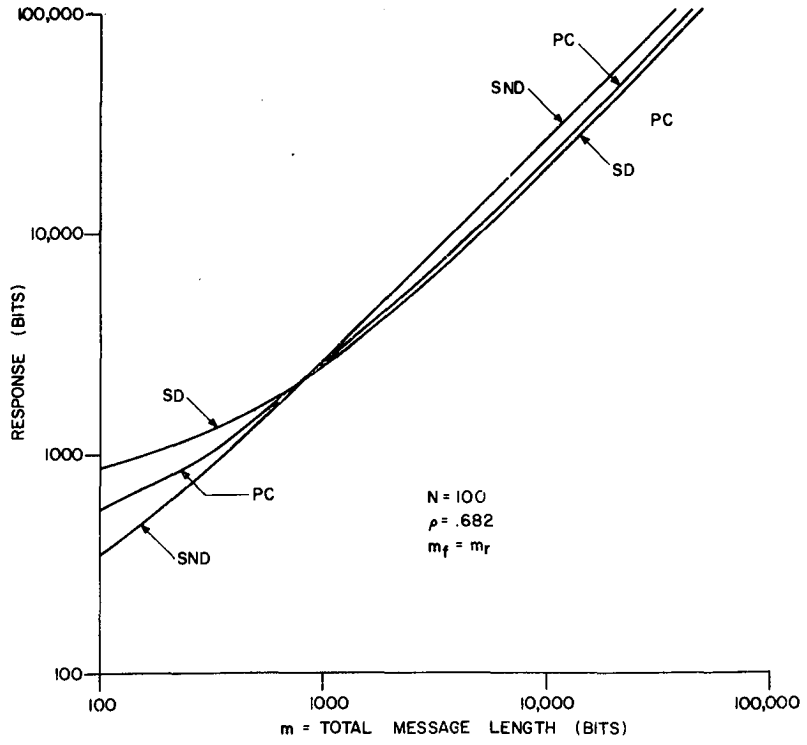

Fig. 4. Comparison of SD, PC & SND loops, with $m_s = m_r$

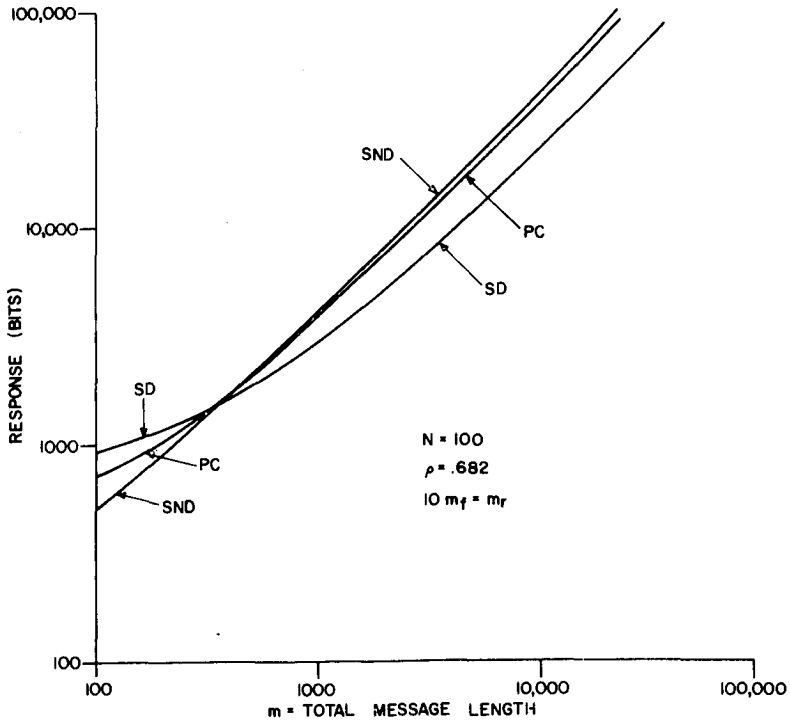

Fig. 5. Comparison of SD, PC and SND loops, with 10 $m_f = m_r$

The differences in overall response among the systems considered here are not particularly large, ranging from about 50% for short messages to 100% for long messages.

5.2 Effect on Performance of Position of the Terminal

For the PC loop, all terminals are treated equally since there is no point on the loop which is issuing free slots and the average interval between opportunities to transmit is equal for all terminals. Consequently the delay experienced at a terminal will depend only on the length of the queue at that particular terminal and is otherwise independent of the position of the terminal on the loop.

For the SD and SND loops the situation is very different. Consider first the SD loop in which there is deletion and re-use of slots. Suppose we have a traffic pattern in which the computer reply is much longer than the input message. Then the traffic density on the line will decrease as one moves farther downstream from the computer, since more slots will be emptied due to message deliveries than empty slots are filled due to message transmissions by the terminals. Hence terminals farther from the computer have more chance of seizing a free slot, which in effect will mean less delay. Table 1 compares theoretical values with simulation results, for a particular example.

TABLE 1

SD Loop Delay by Terminal $\rho = 0.682$
N = 10 Input Message 250 Bits Output Message 2500 Bits

	INPUT DELAY		
TERMINAL	METHOD A	METHOD B	SIMULTATION
1	3719	4065	3528
2	2647	2917	2471
3	1912	2124	1817
4	1392	1558	1385
5	1014	1143	1007
6	733	832	750
7	521	594	537
8	358	409	388
9	232	263	251
10	133	148	143
Average 1-10	1266	1405	1226

For the SND loop the situation is different in that no slots are deleted. Consequently the farther away from the computer the greater is the message density on the line. Table 2 shows how the delay increases. This effect becomes very significant for extremely long messages. For example,

the mean response time for a 3000 bit input message and 30,000 bit return message is 140,000 bits for an SND loop compared to about 75,000 bits in an SD loop. The difference here is due to the very long delay required to input messages from the high-numbered terminals in the SND loop.

TABLE 2

SND Loop Delay by Terminal $\rho = 0.682$
N = 10 Input Message 250 Bits Output Message 2500 Bits

	DELAY		
TERMINAL	METHOD A	METHOD B	SIMULATION
1	5528	5994	5431
2	5714	6195	5963
3	5910	6406	6297
4	6116	6628	6529
5	6334	6861	6557
6	6563	7108	6905
7	6804	7367	6926
8	7060	7642	7049
9	7330	7931	7423
10	7616	8238	7745
Average 1-10	6497	7037	6690

5.3 Effect of Computer Protocols

In all the theory and simulation discussed in previous sections we have assumed the standard "come right in" protocol as Eisenberg[8] has labelled it. This means that each queue is served until empty. An alternative is the "please wait" protocol in which only messages in the queue when the server arrives are considered for service. For a terminal with a very low arrival rate there is little difference in delay between one protocol and the other. However, changing the protocol on the computer queue can be significant. In the loops considered, the "please wait" protocol results in increased computer queue delay and decreased terminal queue delay as would be expected, since the computer service is more frequently interrupted. Total response remains about the same.

One benefit in the SD protocol is that the mean delay for terminals close to the computer is significantly decreased. This is shown in Table 3. The "% substituted" entry is the percent of messages transmitted by a terminal which utilizes an empty slot that previously carried a return message. Naturally this number is low for low numbered terminals because these terminals usually use a slot which was not previously used by return messages. The "please wait" protocol benefits these terminals since it generates free slots more often than the "come right in" protocol.

TABLE 3

Effect of queueing protocol

SD System $\rho = 0.682$ N = 10
Input Message 250 Bits Output Message 2500 Bits

TERMINAL	"COME RIGHT IN"		"PLEASE WAIT"	
	INPUT DELAY	% SUBSTITUTED	INPUT DELAY	% SUBSTITUTED
1	3419	0.17	1437	0.10
2	2497	0.29	1380	0.19
3	1888	0.35	1241	0.25
4	1457	0.42	1101	0.35
5	1002	0.46	831	0.40
6	709	0.51	606	0.48
7	564	0.54	502	0.51
8	427	0.59	430	0.57
9	252	9.61	248	0.59
10	143	0.64	140	0.64
Average 1-10	1224	0.46	787	0.41
Mean Response	6177	-	6126	-

6. CONCLUSIONS

One conclusion is that the three different protocols, proposed for loop communication systems serving transaction traffic all achieve similar mean response times for a wide range of message lengths. Variations of up to 2:1 are observed at the extremes of message length, with slot-deletion loops and slot-no-deletions loops being preferable for long messages and short messages respectively. In the range of message lengths from 250-1000 bits the differences are not significant. The pass-control loop has a performance lying between that of the other two types for all message lengths.

In addition to the variation in mean response time between the three protocols considered, an important difference is the variation between the mean response time for individual terminals caused by the two slot-oriented protocols. This was as great as 25:1 in a slot-deletion loop of modest proportions which was simulated with the least favored terminal having a mean response twice that of the overall mean. An advantage of the pass-control loop is that it treats all terminals equally.

In this report we have not considered the effect of the various control messages and acknowledgements which occur in almost all practical systems, and which are usually specific to any given system. Such messages would usually be shorter than required to fill slots in slot-oriented systems, resulting in an effective increase in overhead and a corresponding drop in line utilization efficiency. No attempt has been made to study the effect of changes in protocol that would improve a particular system. For example Gall and Mueller[14] show how the protocol in the SND loop can be modified to ensure

that least favored terminals are reserved a slot after a certain maximum wait time. This report, on the other hand, attempts to analyze three "pure strategies" in order to understand the underlying trade-offs that occur. Knowing these, it is felt that the analyst is then better prepared to consider protocol changes that optimize performance in a particular situation.

The general conclusion of this report is that other considerations, such as cost, flexibility and ease of control and bookkeeping may be allowed to influence the choice of system to be used in any given application without fear of affecting the response time too severely.

7. REFERENCES

1. Kaye, A.R. and T.G. Richardson, *A performance criterion and traffic analysis for polling systems,* INFOR, June 1973, pp. 93-112.

2. Hayes, J.F. and D.N. Sherman, *A study of data multiplexing techniques and delay performance,* BSTJ, 51, No. 9, November 1972, pp. 1983-2011.

3. Farmer, W.D. and E.E. Newhall, *An Experimental Distributed Switching System to Handle Bursty Computer Traffic,* ACM Symposium on Optimization of Data Communication Systems, Pine Mountain, Georgia, 1969, pp. 4-33.

4. Kaye, A.R., *Analysis of a distributed control loop for data transmission,* Paper No. II.5, PIB International Symposium XXII on Computer Communication Network and Teletraffic, 4-6 April 1972, pp. 47-58.

5. Spragins, J.D., *Analysis of loop transmission systems,* ACM/IEEE Second Symposium on Problems in the Optimization of Data Communication Systems, Palo Alto, October 1971, pp. 175-182.

6. Hayes, J.F. and D.N. Sherman, *Traffic analysis of a ring switched data transmission system,* BSTJ, 50, No. 9, November 1971, pp. 2947-2978.

7. Konheim, A.G. and B. Meister, *Polling in a multidrop communication system: waiting line analysis,* ACM/IEEE Second Symposium on Problems in the Optimization of Data Communication Systems, Palo, Alto, October 1971, pp. 124-130.

8. Eisenberg, M., *Queues with periodic service and changeover time,* ORSA, 20, March-April 1972, pp. 440-451.

9. Cooper, R.B., *Queueing served in cyclic order: Waiting times,* BSTJ, 49, 1965, pp. 399-413.

10. Takacs, L., *Two queues attended by a single server,* ORSA, 16, 1968, pp. 639-650.

11. Chu, W.W. and A.G. Konheim, *On the analysis and modelling of a class of computer communication systems,* IEEE, Trans. on Comms., 20, No. 3, June 1972, pp. 645-660.

12. Leibowitz, M.A., *An approximate method for treating a class of multi-queue problems,* IBM J of Res. and Deo. 5, pp. 204-209 (1961) and Note, October 1962, pp. 470-471.

13. Knight, J.R., *A case study: Airlines reservation systems,* Proc. of IEEE, Vol. 60, No. 11, November 1972, pp. 1423-1430.

14. Gall, D.A. and H.R. Mueller, *Waiting-time distributions and buffer overflow in priority queueing systems,* IEEE Trans. on Comms., 20, No. 5, October 1972, pp. 865-876.

15. Pierce, J.R., C.H. Coker and W.J. Kropfl, *An experiment in addressed block data transmission around a loop,* IEEE Conv. Rec., New York, March 1971, pp. 222-223.

A P P E N D I X

Due to the lack of exact theory for general loop protocols it was necessary to perform simultations to study loop performance under a variety of circumstances. Initially, several programs were written in GPDS (Xerox version of GPSS) both in batch and time-sharing environments. It was found that run-times were rather long because of the priority scheme on the Sigma 7. For this reason it was decided to use Fortran IV. Under the time-sharing system, Fortran jobs, run quickly and have a much smaller core requirement than the equivalent GPDS program. The logic of a simple loop is easy to program and alter so that no great programming effort was required. Collecting statistics is burdensome in Fortran, but since we are mainly interested in mean delays this problem was reduced. For more complex loop configurations we would probably revert to GPDS to keep the programming effort down.

The simulation first generates sequences of messages which are Poisson distributed with the specified rate. Destination and/or originating addresses are assigned in another vector. Then the executive part of the program searches to find the next vent that should occur. The executive also keeps track of various clock times and stores information for statistical analyses. Finally relevant statistics are gathered from some number of runs. Due to the sequential operating nature of a simple loop it is not necessary to have sophisticated starting and stopping procedures. In a hierarchy of loops, however, this would not be the case. The number and size of runs was basically determined to give a "reasonable" standard deviation to the mean value of the several run mean values. In general we aimed for a standard deviation of 5-15% of the mean value.

Runs times can become very long under certain circumstances. If, for example, the message length is very long compared to the delay time around the loop it will require many scans to pick up a message. The simulation is oriented in this way, in that it "looks ahead" one scan, or one slot time to see if there are candidates for input. This is a compromise between updating after each terminal is passed (very slow execution) and doing a comprehensive ordering with current and future events, chains, etc. (very slow programming).

No attempt has been made to use variance reduction techniques such as antithetic variables or stratified sampling. It is highly likely that one of these methods would be valuable in the very simple systems considered here but it is not obvious that they could be applied in more complex situations. They did not appear worthwhile to pursue for this reason.

CRC DOCUMENT CONTROL DATA

1. ORIGINATOR: Communications Systems

2. DOCUMENT NO: CRC Report No. 1261

3. DOCUMENT DATE: August 1974

4. DOCUMENT TITLE: The Effect of Protocol on the Response Time of Loop Structures for Data Communications

5. AUTHOR(s): T.G. Richardson

6. KEYWORDS: (1) Data Communications
(2) Protocol
(3) Loop Structures

7. SUBJECT CATEGORY (FIELD & GROUP: COSATI)

17 Navigation, Communications, Detection, and Countermeasures

17 02 Communications

8. ABSTRACT:

The loop or ring structure for data communications is considered under three different operating protocols. An inquiry-response type traffic pattern is assumed and response time is chosen as the measure of performance. Each loop protocol is analyzed through the use of approximate theoretical queueing formulae and these results are compared to those of a simulation. Conclusions are drawn as to the relative merits of the three alternatives. Mean response time is found to be about equal over the range of interest but important differences are observed in individual terminal behavior.

9. CITATION:

COMMUNICATIONS
CANADA