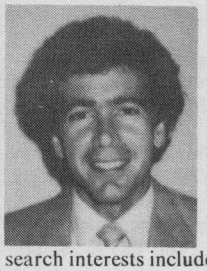


Rami R. Razouk was born in Cairo, Egypt, in 1953. He received the B.S. degree in engineering. In 1975 he received the M.S. degree in computer science and in 1980 he received the Ph.D. in computer architecture, both from the University of California, Los Angeles.

Presently, he is with the Department of Computer Science, University of California, Los Angeles. Since 1977 he has led the SARA design group under the direction of Dr. Gerald Estrin. His research interests include modeling, simulation, and analysis of computer systems.

Dr. Razouk is a member of Phi Beta Kappa and Tau Beta Pi.

Gerald Estrin (S'48–A'51–M'56–F'68) was born in New York, NY. He received the B.S., M.S., and Ph.D. degrees from the University of Wisconsin, Madison, in 1948, 1949, and 1951, respectively.

He served as Research Engineer at the Institute for Advanced Study, Princeton University, where he participated in the design of one of the earliest large digital computers. Subsequently, he served as the Director of the Electronic Computer Project at the Weizmann Institute of Science, Israel, where he led the development of WEIZAC, the first large-scale electronic computer outside of the United States or Western Europe. In 1956 he became a member of the faculty at the University of California, Los Angeles and in 1979 received University-wide recognition when the Regents approved his appointment as an Above Scale Professor. Since July 1979 he has been Chairperson of the UCLA Computer Science Department. He is leading research projects in the computer architecture area including the modeling, measurement, and synthesis of computer systems and programs, multilevel simulation of computers, and microprocessor-based networks.

Dr. Estrin is a Guggenheim Fellow, a member of the Board of Directors of Systems Engineering Laboratories, Ft. Lauderdale, FL, and a member of the Board of Governors of the Weizmann Institute of Science, Rehovot, Israel.

A Tradeoff Study of Switching Systems in Computer Communication Networks

PARVIZ KERMANI, MEMBER, IEEE, AND LEONARD KLEINROCK, FELLOW, IEEE

***Abstract*—This paper is concerned with a comparison study of three switching techniques used in computer-based communication networks: circuit switching, message (packet) switching, and *cut-through* switching. Our comparison is based on the delay performance as obtained through analytic models of these techniques. For circuit switching, the model reflects the phenomenon of channel reservation through which it can be shown that when circuit switching is used, data communication networks saturate rapidly. Through numerical examples, it is shown that the boundary between the areas of relative effectiveness of these switching techniques depends very much on the network topology (more precisely the path length of communication), the message length, and the useful utilization.**

***Index Terms*—Circuit switching, computer communication networks, cut-through switching, message switching, packet switching, performance evaluation, store-and-forward.**

I. INTRODUCTION

IN this paper we present a comparison study of some proposed and existing switching techniques employed in computer-based communication networks.

Manuscript received February 8, 1980; revised July 1, 1980. This work was supported by the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-77-C-0272. This paper was presented at the International Conference on Communications, Boston, MA, 1979.

P. Kermani is with the IBM T.J. Watson Research Center, Yorktown Heights, NY 10598.

L. Kleinrock is with the Department of Computer Science, University of California, Los Angeles, CA 90024.

A tradeoff study of switching systems involves consideration of a wide range of issues; however, in most cases, cost and delay are the two major criteria. Network cost consists of installation and maintenance cost (from the owner's point of view) and service cost (from the users' point of view). Consideration of cost requires either specializing the study toward a specific system or making strong assumptions regarding the physical structure and hardware costs of the switching nodes and the communication media. Considering the rapid advances of today's technology, with a trend toward lower cost and better performance, any hardware cost assumptions soon become obsolete due to their dependence on technology. For these reasons we choose not to base our comparison study on cost and consider only the criterion of network delay.

This type of comparison study has been the subject of considerable interest for some years. Most published studies compare the performance of circuit switching (CS) employed in telephone networks with store-and-forward switching (which has been used in one form or another in most recent data communication networks). The object is to identify those situations (e.g., traffic pattern, network topology, etc.) in which one switching technique out-performs the other.

Among the earliest published reports on this subject is [1], in which the issue of signaling in CS networks was studied. Later, reports [2] and [3] are extensions of this work. In [4] a detailed comparison based on the network delay between

0018-9340/80/1200-1052$00.75

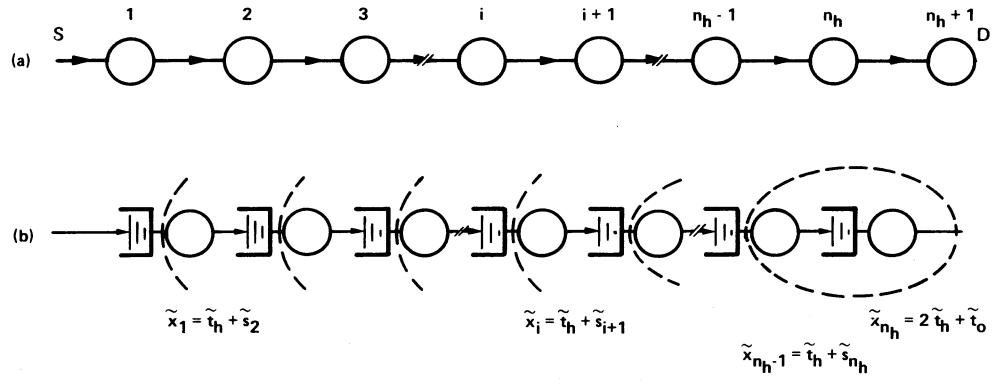


Fig. 1. Tandem queue model of a communication path.

switching schemes is made. This work confirmed the intuitive understanding that at higher traffic rates store-and-forward switching results in a better delay performance, whereas for longer message lengths CS is superior to the other switching technique. In [5] a comprehensive comparison study was made from two points of view: delay performance and usage cost. This study again confirmed the previous results.

We hasten to point out that almost all of the previous works (except [1]) have neglected to include the effect of channel reservation in their analytic modeling of CS. The fact is, channel reservations seriously degrade the throughput capability of CS as we shall see shortly. By accepting some simplifying assumptions (which were first introduced in [1]) we are able to develop a more accurate analytic model than has previously been used.

The aim of the present study is to give a reasonably realistic and quantitative performance comparison of three switching systems: circuit switching (CS); message (and packet) switching (MS); and cut-through switching (CTS, described below). Throughout our study we do not differentiate between packet switching and message switching; we consider both methods as members of the larger class of store-and-forward switching systems.

In the following sections, we develop analytic models for the performance of these switching systems and then present some numerical results and performance curves.

II. Delay-Models

To simplify the analysis, we consider a specific communication path from source S to destination D [Fig. 1(a)] and study the message delay along this path under different switching techniques. In our path model, we have $n_h + 1$ nodes which are connected in tandem by n_h trunks, each of capacity C bps. Messages enter the network at node 1 and leave the network at node $n_h + 1$, i.e., they travel over n_h hops. For CS we assume each trunk is divided into $N_{ch} \geq 1$ channels, each of capacity C/N_{ch}. The channels are assumed to be noiseless. Messages to be transmitted are considered to be a continuous stream of $\tilde{l}_m$ bits. To each message we append an overhead of $\tilde{l}_h$ bits; the overhead is provided for the addressing header in MS and CTS, or the reservation signal in CS. The network delay (the path delay) is the interval of time from the moment a message is submitted to node 1 until it is received at node $n_h + 1$. In the formulation below we assume that the input processes of message arrivals to all nodes is Poisson with rate λ (messages/second), and that message lengths (data plus header) are distributed exponentially.

A. Message Switching

In message switching [6], messages are transmitted in a hop-by-hop fashion through the network. Each message carries its destination address in its header. At each intermediate node the message must be completely received before it can be forwarded toward its destination node. If the selected outgoing channel is busy, the message is queued while awaiting transmission. Accepting the independence assumption of Kleinrock [7], the average delay at each node becomes

$$T = \frac{(\bar{l}_m + \bar{l}_h)/C}{1 - \lambda(\bar{l}_m + \bar{l}_h)/C}$$

where $\bar{l}_m$ and $\bar{l}_h$ are the average values of $\tilde{l}_m$ and $\tilde{l}_h$, respectively, and λ is the input rate of messages (we use $1/\mu = \bar{l}_m$). The average end-to-end delay of MS is then $n_h T$, or

$$T_{MS} = \frac{(\bar{l}_m + \bar{l}_h)/C}{1 - \lambda(\bar{l}_m + \bar{l}_h)/C} n_h. \tag{1}$$

B. Cut-Through Switching

In cut-through switching [8], [9] the operation is similar to MS, with the difference that messages do not have to be received completely at an intermediate node before being transmitted out of the node toward the destination. After the header of a message is received, the outgoing channel can be selected, and if this selected channel is free, the message may start transmission out of the node immediately, even though its tail has not yet arrived in the node. If, however, after the reception of the header, it is found out that the outgoing channel is busy, the operation follows that of MS, i.e., the message must be received completely before being sent out toward the destination node. Therefore, a message can *cut through* intermediate nodes and save the unnecessary buffering delay if the outgoing channel is free. A complete performance analysis of this system has been reported elsewhere [8], [9]. In the Appendix at the end of this paper we derive expression for the average delay for cut-through switching to be

$$T_{CTS} = T_{MS} - (n_h - 1) \times \left(1 - \lambda \frac{\bar{l}_m + \bar{l}_h}{C}\right)\left(1 - \frac{\bar{l}_h}{\bar{l}_m + \bar{l}_h}\right)\frac{\bar{l}_m + \bar{l}_h}{C} \quad (2)$$

where T_{MS} is given by (1).

C. Circuit Switching

With circuit switching, a complete path of communication must be set up between two parties before the communication begins. Path setup is established through a signaling process. Before transmission of a message, a (reservation) signal is sent towards the destination. While traveling node by node towards the destination node, the signal reserves channels along the path. If, at any intermediate node, it cannot find a free channel, it waits for a channel to become free (while holding the channels it has reserved so far), at which time the signal reserves it and goes to the next node to repeat the same process. By the time the signal reaches the destination node, a path has been reserved between the source and the destination nodes. Fig. 2 shows the structure of a node in a communication network. A message from outside the network arrives in box (or queue) A and sends a request-for-connection (RFC) signal to establish a path between this node and its destination node. Box B is a queue in which signals wait to reserve one of the message channels between the two adjacent nodes (service facility C). Having reserved a channel, the signal uses the same channel for transmission to the next node, at which time the signal goes through the same process until it arrives at the destination node. When the signal reaches its destination, the originating message is notified (through a reverse signalling process called request-for-transmission (RFT) using the path which has already been set up), at which time the message can start transmission. Note that during the RFT and the message transmission all of the channels on the path are used simultaneously, hence, these last transmissions are similar to a one-hop transmission. After completion of the message transmission, the reserved channels are released. We assume that no further signaling is required for releasing the reserved channels.

The above model is usually referred to as a *forward reservation-individual signaling channel* system. Reservation of message channels can also be done while the signal is coming back from the destination node (backward reservation). Signaling channels can also be separate from message channels (i.e., all signals use a common signaling channel), which is called *common signaling.* In this case, when a message channel is reserved by a signal, the signal joins another queue to be transmitted over the signaling channel.

What we just described is the *delay* mode of operation. A circuit switched network may (and usually does) operate in a *loss* mode, in which case when a reservation signal encounters a busy channel, the reservation process terminates and the calling party is dropped from the network. In this paper we assume the CS network operates under delay mode, as its performance measure (delay) in this case is comparable with those of MS and CTS.

An exact analysis of a CS network leads to the development of a network model, the solution of which requires treatment of a multiple server queueing system with a non-Poisson input process and nonexponential service time; such systems, to date, have not in general been solved. A partial solution of such a model is reported in [8]. In order to develop a model which is analytically tractable, we consider a *path* of a communication network [Fig. 1(a)] and consider the traffic between the source node S and the destination node D. Fig. 1(b) shows the tandem queue model of the communication path. In this figure each trunk is abstracted as an N_{ch} server, FIFO queueing system. Because of the reservation process, the service time of each queue is affected by waiting times in the succeeding queues. Let $\tilde{x}_i$ (with average $\bar{x}_i$) be the service time at the ith queue, and $\tilde{s}_i$ (with average $\bar{s}_i$) be the total delay at the ith node $1 \le i \le n_h$. Furthermore, let $\tilde{t}_m = \tilde{l}_m N_{ch}/C$ and $\tilde{t}_h = \tilde{l}_h N_{ch}/C$ be transmission times of a message and a signal (with an average $\bar{t}_m$ and $\bar{t}_h$), respectively. The service time in the rightmost queue (the n_hth queue) is

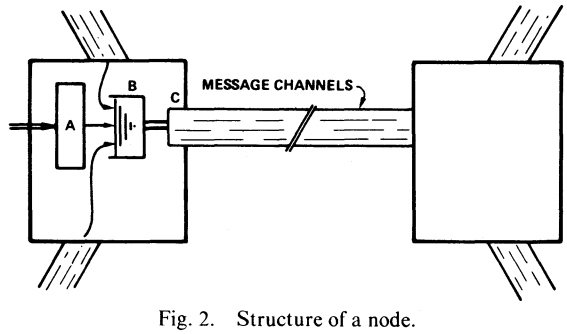


Fig. 2. Structure of a node.

$$\tilde{x}_{n_h} = \tilde{t}_h + \tilde{t}_h + \tilde{t}_m = 2\tilde{t}_h + \tilde{t}_m.$$

The first $\tilde{t}_h$ corresponds to the "request-for-connection" signal from node n_h to node $n_h + 1$, the destination node. After this, a "request-for-transmission" signal originates from node $n_h + 1$ and is sent to node 1. Because a complete path is already set up, this transmission takes only $\tilde{t}_h$ seconds. After reception of this signal, the message is transmitted from node 1 (which takes $\tilde{t}_m$ seconds). The service time at queue $n_h - 1$ is affected by the system time (waiting plus service time) of the n_hth queue, so we have

$$\tilde{x}_{n_h-1} = \tilde{t}_h + \tilde{s}_{n_h}.$$

In general, we have the following expression for the service time at the ith queue

$$\tilde{x}_i = \tilde{t}_h + \tilde{s}_{i+1} \qquad 1 \le i < n_h \quad (3a)$$

$$\tilde{x}_{n_h} = 2\tilde{t}_h + \tilde{t}_m \quad (3b)$$

and for the average values we have

$$\bar{x}_i = \bar{t}_h + \bar{s}_{i+1} \qquad 1 \le i < n_h \quad (4a)$$

$$\bar{x}_{n_h} = 2\bar{t}_h + \bar{t}_m. \quad (4b)$$

In order to simplify the analysis we accept a generalized version of the independence assumption [7].

Assumption: The distribution of service time at each of the queueing systems shown in Fig. 1(b) is a negative exponential with the average values determined by (4), and is stochastically independent of the service time of its succeeding nodes.

Assuming that the input process of messages to the first node is Poisson, by virtue of the above assumption, the input process

of all of the queues on the path will be Poisson and the queueing system at each node can be treated as an $M/M/N_{ch}$ queue [6], N_{ch} being the number of servers.

To find the path delay (the time between the arrival of a message at the first node till the completion of its transmission), one must start from the last queue (n_hth) and iteratively proceed to the first queue; the mean system time of queue 1, $\bar{s}_1$, is the total path delay, i.e.,

$$T_{CS} = \bar{s}_1 = \bar{x}_1 + \bar{t}_h \tag{5}$$

where $\bar{x}_1$ is given by (4).

Remarks:

1) It has been shown [2], [3], [5] that in the "common signaling" method, a proper signaling channel capacity is vital to the proper operation of the network. In fact, in order to obtain the optimal performance, the signaling channel capacity should be dynamically adjusted to the traffic in the network. Because the emphasis of the present study is to study the effect of channel holding time on the overall performance, we base our study on an "individual signaling channel" model, hence, we are not faced with this optimization problem.

2) We should point out that the way we have treated the holding time distribution is not accurate in a narrow sense. In [10] it is shown that under certain assumptions (in particular in networks with no feedback), the distribution of the end-to-end delay in a Markovian network of queues is given by a sum of independent and exponentially distributed random variables. Though the system under our investigation differs from a Markovian network (mainly because of the reservation phenomenon), in view of the results in [10], perhaps some other distributions (e.g., Erlangian) for the service time would be a better choice; however, any distribution except negative exponential would complicate the solution. As we shall see shortly, even this simple and approximate model clearly shows the effect of channel reservation and holding time. In all of the previous reports, except in [1], the queueing system at each node is considered as an $M/M/m$ system in which the average service time is the same as the average transmission time of a message, an assumption which is far from realistic (unless the model is for a fully connected network); our model corrects that defect.

III. Discussions of Results

Based on the models developed in the previous section, we present numerical comparisons of the network delay for the three switching systems: CTS, MS, and CS. The network model is the one shown in Fig. 1(a) and our comparison is based on network delay only, i.e., we will disregard the access delay to the network, as this component of the delay is common to all three systems. We should point out that, as a result of disregarding access delay, our results are slightly biased in favor of MS and to some extent CTS. Access delay is essentially a one-hop transmission. For large message lengths and, in particular, in light load conditions, access delay is smaller in CS than in MS and to some extent in CTS. A total trunk capacity of $C = 50$ kbps is used and only for CS the trunks are subdivided into N_{ch} channels (hence, for CTS and MS we consider single channel trunks only). The average header (or signal) length is assumed to be $\bar{l}_h = 100$ bits and the average message length will be specified for each case. Lastly, we also assume that channels are noiseless. The useful utilization, which for simplicity will be also referred to as utilization, denoted by $\rho = \lambda \bar{l}_m / C$, is the fraction of the total trunk capacity used by the useful information (i.e., the message). This is usually different from (and less than) the effective utilization ρ_e, which is the utilization caused by the useful information plus the header (or signal) plus the holding time of the reserved channel. In our comparison study, we study the effect of four parameters: input rate λ, average message length $\bar{l}_m$, path length $\bar{n}_h$, and number of channels per trunk N_{ch} (this last parameter is only important for CS).

We first consider the network delay for a path of average length $n_h = 4$ [Fig. 3(a)]. The average message length (useful information) is 1500 bits. This figure shows the normalized path delay (delay/$(\bar{l}_m/C)$) for CTS, MS, and for CS when the trunk is split into one, two, and four channels.

Some observations can be made on these curves. The CTS delay is always smaller than the MS delay (in [8] and [9] it is shown that when the trunks are noiseless, this is *always* the case). For the particular parameters of this figure, the delay of CTS is always less than CS delay; however, this is not always the case. As we will see in the following figures, in some situations network delay for CS becomes less than the CTS delay.

Comparing CS with MS, we see that for a small number of channels N_{ch} and at low input rates, the CS delay is less than the MS delay. This is because at low input rates the waiting time to reserve a channel is small, therefore, path setup time is small. Once a path is set up, the message is transmitted without incurring further delay in the intermediate nodes. On the other hand, in MS a message must be assembled at all of the intermediate nodes, a process that causes excessive delay. When the number of channels is large (e.g., $N_{ch} = 4$), the CS channel capacity is small and even though the path setup is negligible, transmission time of a message on a (small capacity) channel is excessive and the total path delay becomes larger than MS. As the input traffic increases, Fig. 3(a) shows that CS, with a small number of channels per trunk, saturates very quickly (see the delay curve for $N_{ch} = 1$) and the delay curve climbs rapidly. This rapid saturation is the result of the excessive waiting time to reserve a channel and as we already know, this delay is reflected in the channel holding time. By further increasing the number of channels per trunk, because the waiting time is reduced the network does not saturate so quickly (see the CS delay curve for $N_{ch} = 4$), even though the transmission time increases. Comparing this curve with the delay curve of MS, we observe an interesting behavior. For very small input rates, the MS delay is less than the CS delay; however, as input rate increases, there is a crossover between the two delay curves and for a certain interval the CS delay becomes less than the MS delay. For a further increase of input rate, there is a sharp increase in the CS delay and MS again becomes better than CS (from the delay point of view). The crossover occurs because for a certain range of input rates the combined waiting time and reassembly delay at the intermediate nodes for MS becomes more than the path setup delay

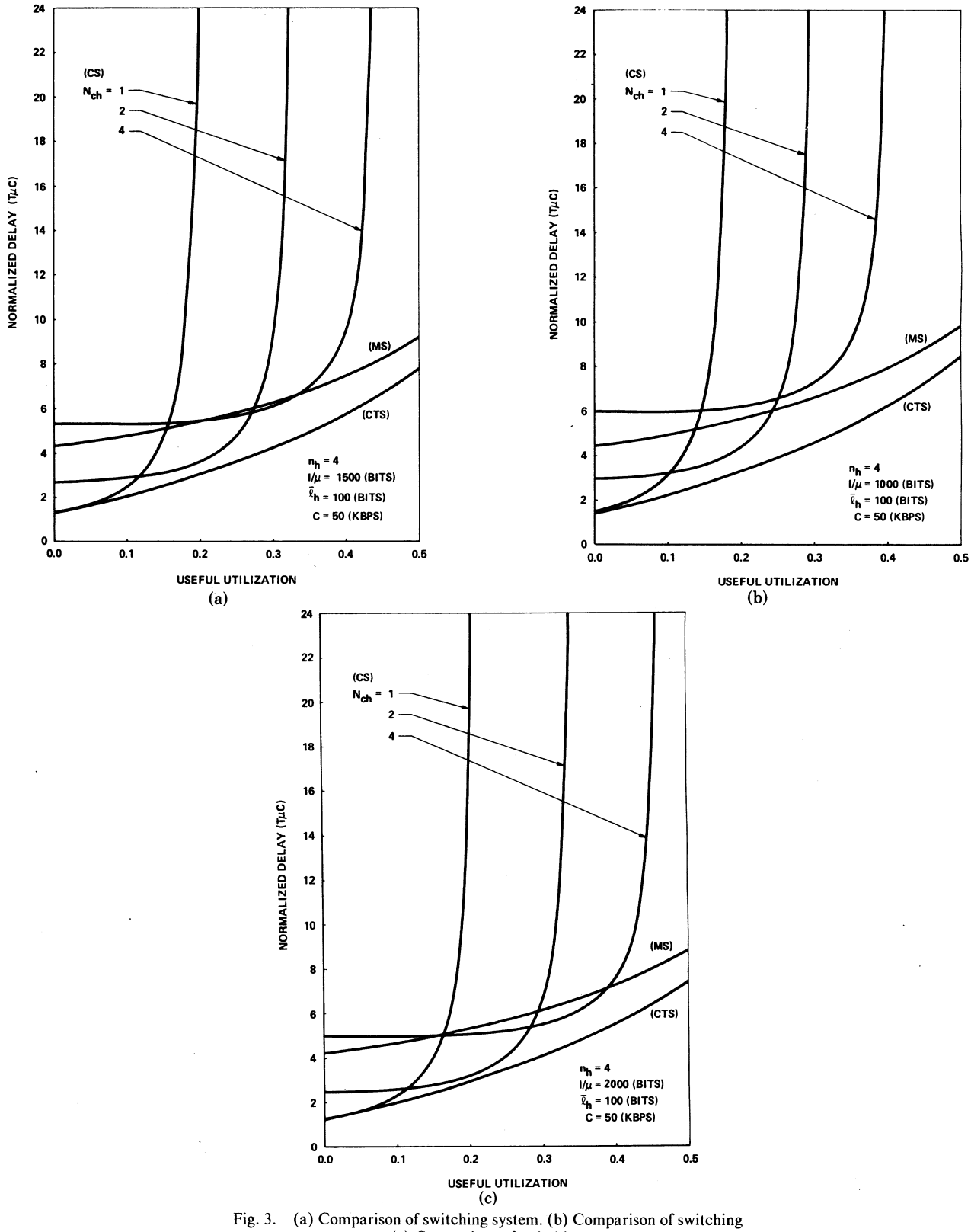


Fig. 3. (a) Comparison of switching system. (b) Comparison of switching systems. (c) Comparison of switching systems.

and message transmission time for CS. We should notice that the interval at which CS becomes better than MS depends very much on the parameters of the system. For example, for a smaller average message length [$\bar{l}_m$ = 1000 bits; Fig. 3(b)], there is no crossover point for the MS and CS delay curves when N_{ch} = 4 (in Fig. 3(b) crossover does occur at larger values of N_{ch}). For comparison, we have also presented delay curves for the average message length of 2000 bits in Fig. 3(c) and a longer path length n_h = 8 in Fig. 4. Comparison of Fig. 3(a) with Fig. 4 shows that for long path lengths, CS is more advantageous than MS in a larger region. This is because in MS it is necessary to assemble the message in all of the inter-

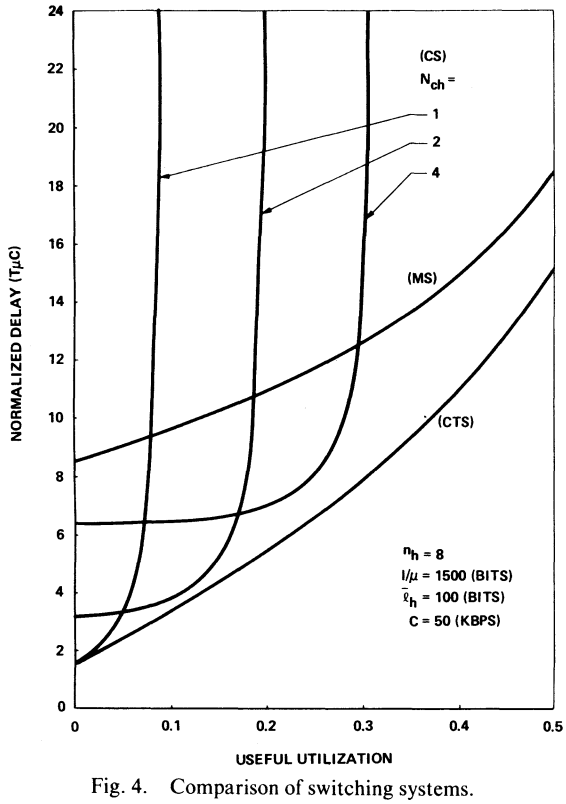


Fig. 4. Comparison of switching systems.

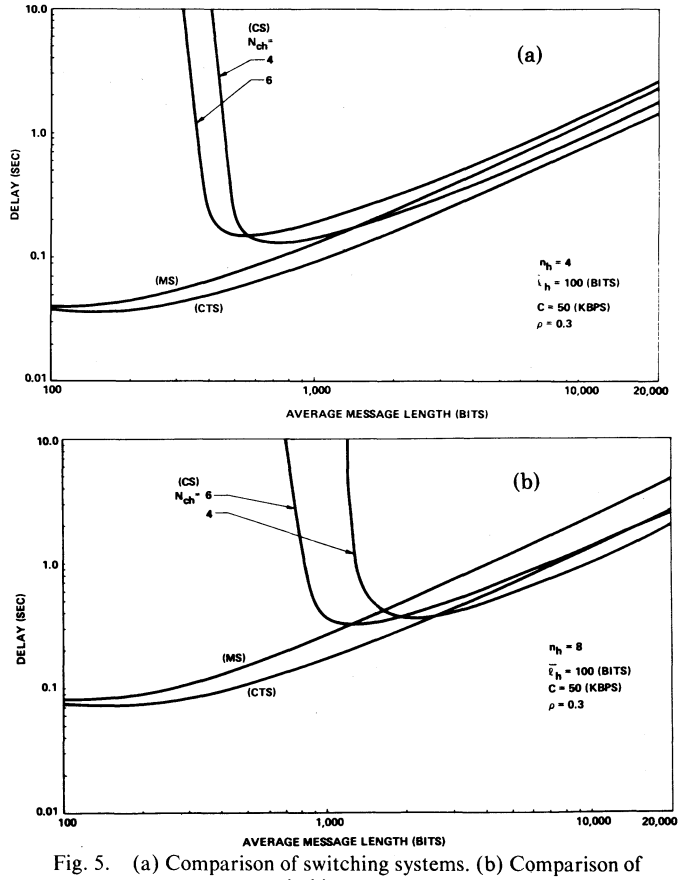


Fig. 5. (a) Comparison of switching systems. (b) Comparison of switching systems.

mediate nodes and so the network delay becomes large, whereas in CS, once a path is set up, there is no further delay due to intermediate nodes.

Fig. 5(a) and (b) show the effect of message length on network delay. In these figures the useful utilization is kept constant ($\rho = 0.3$); hence, the input rate varies as the message length changes (the average header, or signal, length is kept constant; $\bar{l}_h = 100$ bits). For very small message lengths, the network delay for all the three switching systems is unbounded, a consequence of the high-input rate to keep the useful utilization constant (in these figures MS and CTS delay curves become unbounded for $\bar{l}_m < 100$ bits which have not been plotted). As the message length increases, the network delay first decreases and then grows again. As before, we observe that the CTS delay is always less than the MS delay. For CS, after a sharp decrease (which indicates a rapid recovery from saturation), the delay escalates; however, the rate of increase for the delay in CS is less than the rate for the other switching systems and eventually the CS delay becomes less than the MS and CTS delays. This phenomenon can be seen better in Fig. 5(b) where the path length is long. This figure also shows that at longer path lengths CS becomes more advantageous in reducing the network delay. Comparison of the delay curves for CS indicates that the saturation point for a larger number of channels per trunk is higher (i.e., the network can accept larger input rates).

In Figs. 6–8 the delay of MS is compared with the CS delay at different utilization-message length combinations. We consider three path lengths; short $n_h = 2$; medium $n_h = 4$; and long $n_h = 8$, (Figs. 6, 7, and 8, respectively). For each path length we consider 6 different numbers of channels $N_{ch} = 1$, 2, 4, 6, 8, and 10 (Figs. 6(a)–(f), etc.). Note that MS always uses the full capacity of the trunk and there is no splitting involved in this case. Again, the header length is kept at 100 bits and the total trunk capacity is 50 kbps in all cases we study. In these figures the following notation has been used:

cross-hatch:	MS delay < CS delay
space (blank):	MS delay > CS delay
dot ("·"):	MS delay < ∞; CS delay unbounded
"X":	Both the MS and CS delay unbounded

For a fixed path length the area that CS is operational (the network is not saturated) expands with an increase in the number of channels per trunk. This area is denoted by cross-hatched or blank area (the region designated by "·" and/or "X" is where the CS delay is unbounded). For a given number of channels, as the path length increases, the area that CS is operational shrinks. This effect can be seen by comparing, for example, Figs. 6(f), 7(f), and 8(f) with each other. For short path lengths a large number of channels is not advantageous to CS as compared to MS (Fig. 7), although a small number of channels shrinks the operational region of CS.

We can carry out the same kind of comparison between CS and CTS. As we pointed out earlier, CS manifests its advantage when the path length is long; for short path lengths (e.g., $n_h < 8$), CS is hardly ever better than CTS. For a large path length (e.g , $n_h = 8$), the figure becomes very similar to its counterpart figure, Fig. 8; however the area more favorable to CS shrinks. This is due to the fact that the CTS delay is always less than the MS delay. The interested reader is referred to [8].

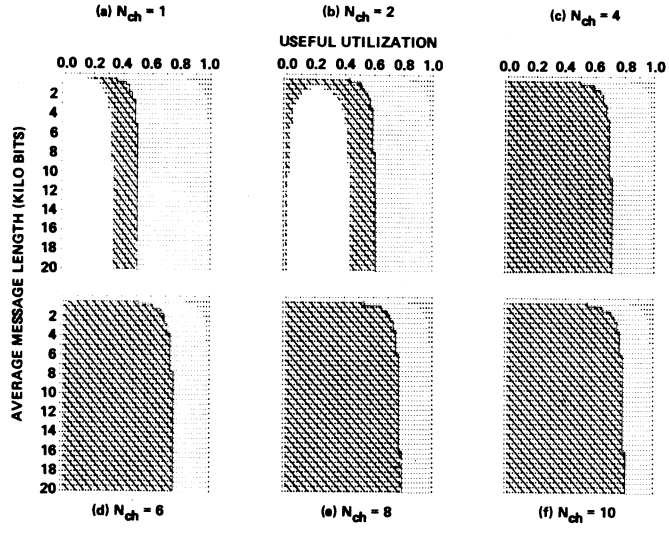


Fig. 6. MX versus CS ($n_h = 2$).

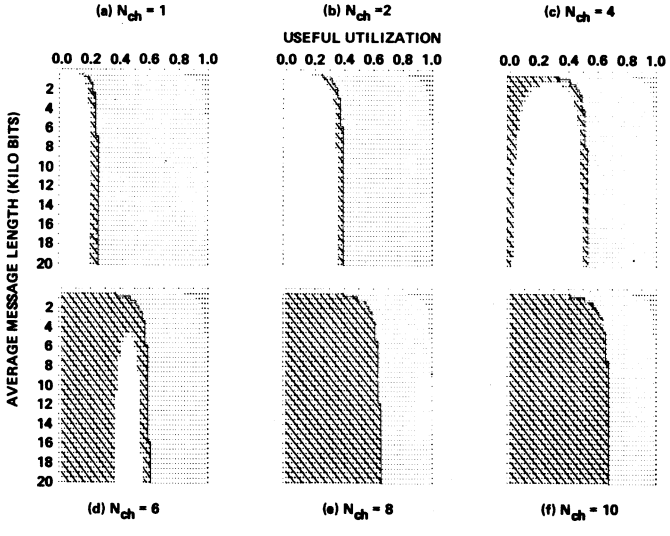


Fig. 7. MS versus CS ($n_h = 4$).

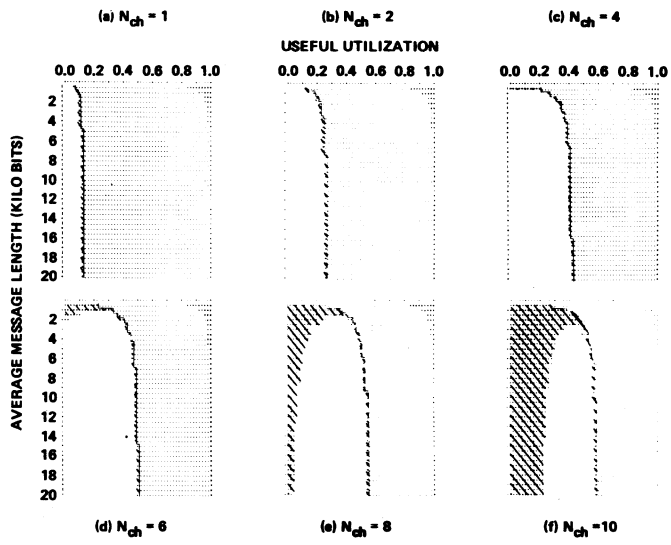


Fig. 8. MS versus CS ($n_h = 8$).

TABLE I
COMPARISON OF SWITCHING TECHNIQUES

		SMALL $n_h = 2$		MEDIUM $n_h = 4$		LARGE $n_h = 8$	
		$500 \le l_m \le 2000$	$2000 < l_m$	$500 \le l_m \le 2000$	$2000 < l_m$	$500 \le l_m \le 2000$	$2000 < l_m$
SMALL $\rho \le .1$	S $N_{ch} = 1$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
	M $2 \le N_{ch} \le 4$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
	L $4 < N_{ch} \le 20$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
MEDIUM $.1 \le \rho \le .4$	S $N_{ch} = 1$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
	M $2 \le N_{ch} \le 4$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
	L $4 < N_{ch} \le 20$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
LARGE $\rho > .4$	S $N_{ch} = 1$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
	M $2 \le N_{ch} \le 4$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS
	L $4 < N_{ch} \le 20$	CTS≅CS<MS	CTS<CS<MS	CTS<MS<CS	CTS<MS≅CS	CTS<MS≅CS	CS<CTS<MS

Table I shows the results of our comparison study. In this table we have chosen two intervals for the message length ($500 \le \bar{l}_m \le 2000$ and $2000 < \bar{l}_m$) and three intervals (small, medium, and large) for the other three parameters (ρ, N_{ch}, and n_h). The symbol "<" is used to indicate that the delay due to the switching system on the left-hand side of the symbol is less than the delay due to the switching technique on the right-hand side. In the case that a selected range is too large to determine a firm advantage between two switching systems, we have used the symbol ≅.

Summarizing our observations, the following remarks can be made:

1) The reservation operation in CS causes a substantial decrease in the network capacity. For this reason CS is not a good choice when the traffic is high (Table I).

2) For large messages, large path lengths, large number of channels, and a moderate utilization, CS can usually outperform the other two switching systems (Table I).

3) When the capacity of the trunk is kept fixed, proper selection of the number of channels per trunk is very critical in the proper operation of the CS system. In fact, this number has an opposing effect on the performance of the network: when the number of channels is small, then the network saturates quickly; on the other hand, a large number of channels results in a large delay. (The problem of determining the optimal number of channels per trunk has been studied in depth in [8].)

4) Based on our study, the decision as to which form of switching to use should be based on a careful study of the parameters of the network.

If the data stream is intermittent, i.e., if a communication session consists of an alternating sequence of message blocks and idle periods, depending on the duration of the idle periods and the message lengths, holding the connection during the idle periods will significantly affect these results. We have not studied such cases; however, the analytic model developed in this section can easily be modified to encompass this case.

From the performance curves presented in this section, it is clear that the ratio $\bar{l}_m/(\bar{l}_m + \bar{l}_h)$ plays a crucial role in the location of the boundary between the area of relative effectiveness between these switching techniques. Because of space limitation, we have not shown this dependency explicitly.

IV. CONCLUSION

In this paper we first presented a mathematical model for the delay performance of circuit switching which incorporated

the effect of reservations. By using this model we carried out a comparison study between the delay performance of three switching systems: circuit switching, message switching, and cut-through switching. Our study showed that the boundary between the areas of relative effectiveness of these switching techniques depends very much on the network topology (which was reflected in the path length), the message length (more precisely on the ratio $\bar{l}_m/(\bar{l}_m + \bar{l}_h)$), and the useful utilization. For the circuit switched system the number of channels per trunk is also an important parameter. While this study confirmed the previous understanding, it also quantitatively showed the effect of some parameters (e.g., the path length), which so far have been disregarded. Based on our studies, when the number of channels per trunk in CS is properly selected, CS is a better choice of switching system than MS and/or CTS at relatively low utilizations and large message lengths and when the communication path is long. In any other circumstance, CTS outperforms CS and MS. For small path lengths and moderate-sized messages, MS outperforms CS; for large utilization, MS outperforms CS. In general, the decision on the choice of switching method depends on the parameters of the system, and no single switching technique is optimum under all conditions.

The analytic models, especially the one for CS, are approximate; more exact models are, in general, complex to solve and inflexible to use (see for example [8] for an exact model for CS). We believe, however, that the comparison results presented in this paper are valid as the approximations affect all systems uniformly.

We did not differentiate between packet-switching and message-switching; these two methods were considered members of the larger class of store-and-forward switching systems. In particular, we used a model for message-switching to represent store-and-forward switching performance. This is valid when message lengths are small and each packet comprises a message. However, for large message lengths the results are biased against store-and-forward switching, as with packet switching one can achieve a better delay performance in this case. One should particularly bear this consideration in mind in studying the performance curves in Fig. 5(a) and (b).

APPENDIX

End-to-End Delay in Cut-Through Switching

We first find the average number of times a message encounters free channels on its way to the destination.

Whenever a message enters a node, the outgoing channel is free with probability $(1 - \rho)$ where $\rho = \lambda(\bar{l}_m + \bar{l}_h)/C$ is the utilization of channels (recall that we assume all channels have the same utilization). Due to the independence assumption, the number of times a message encounters idle channels has a binomial distribution and we have

$$\Pr[\tilde{n}_c = k] = \binom{n_h - 1}{k} (1 - \rho)^k \rho^{n_h - k - 1} \quad 0 \le k \le n_h - 1$$

where $\tilde{n}_c$ is the number of times a message encounters free channels and n_h is the path length. The average of $\tilde{n}_c$, $\bar{n}_c$ is therefore,

$$\bar{n}_c = \mathrm{E}[\tilde{n}_c] = \sum_{k=0}^{n_h - 1} k \Pr[\tilde{n}_c = k] = (n_h - 1)(1 - \rho). \tag{A1}$$

Note that if $n_h = 1$, then $\bar{n}_c = 0$, viz. in a one-hop path there is no intermediate node and no "cut-through" is made.

We now find the average delay. Each time a free node is encountered, a nodal service time is saved. However, this service time is conditioned on the event that the waiting time is zero. So we have

$$T_{\mathrm{MS}} - T_{\mathrm{CTS}} = \bar{n}_c \, \mathrm{E}[\tilde{s} | \tilde{w} = 0] \tag{A2}$$

where $\tilde{s}$ is the total delay in a node and $\tilde{w}$ is the waiting time in a node. Considering the fact that a message can be sent out only after its header is received, we have

$$E[\tilde{s} | \tilde{w} = 0] = \frac{\bar{l}_m}{C}$$

and (A2) is changed to

$$T_{\mathrm{MS}} - T_{\mathrm{CTS}} = \bar{n}_c \frac{\bar{l}_m}{C}. \tag{A3}$$

Where $\bar{l}_m/C$ is the saving in delay at each node where a cut-through is made. Using $\bar{n}_c$ from (A1), we have

$$T_{\mathrm{CTS}} = T_{\mathrm{MS}} - (n_h - 1)\left(1 - \lambda \frac{\bar{l}_m + \bar{l}_h}{C}\right)\frac{\bar{l}_m}{C}$$

or

$$T_{\mathrm{CTS}} = T_{\mathrm{MS}} - (n_h - 1)\left(1 - \lambda \frac{\bar{l}_m + \bar{l}_h}{C}\right) \times \left(1 - \frac{\bar{l}_h}{\bar{l}_m + \bar{l}_h}\right)\frac{\bar{l}_m + \bar{l}_h}{C}.$$

REFERENCES

[1] E. Port and F. Closs, "Comparison of switched data networks on the basis of waiting times," IBM Zurich Res. Center, Zurich, Switzerland, Tech. Rep. RZ405 (#14721), Jan. 1971.

[2] F. Closs, "Message delay and trunk utilization in line-switched and message-switched data networks," in *Proc. 1st USA–Japan Comput. Conf.*, Tokyo, 1972, pp. 524–530.

[3] F. Closs, "Time delay and trunk capacity requirements in line-switched and message-switched networks," in *Int. Switching Symp. Re.*, Boston, MA, 1972, pp. 428–433.

[4] K. Itoh, T. Kato, O. Hashida, and Y. Yoshida, "An analysis of traffic handling capacity of packet switched and circuit switched networks," in *Proc. 3rd Data Commun. Symp.*, St. Petersburg, FL, Nov. 1973, pp. 29–37.

[5] K. Kummerle and H. Rudin, "Packet and circuit switching: A comparison of cost and performance," in *Nat. Telecommun. Conf. Proc.*, Dallas, TX, Nov. 1976, Vol. III, pp. 42–5.1 to 42–5.7, also IBM Zurich Res. Center, Zurich Switzerland, Rep. RZ 805 (#27122), Nov. 1976.

[6] L. Kleinrock, *Queueing Systems, Vol. II: Computer Applications.* New York: Wiley-Interscience, 1976.

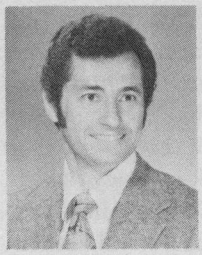
[7] ——, *Communication Nets: Stochastic Message Flow And Delay.* New York: McGraw-Hill, 1964.

[8] P. Kermani, "Switching and flow control techniques in computer communication networks," Ph.D. dissertation, Comput. Sci. Dep., Sch. of Eng. and Applied Sci., Univ. of California, Los Angeles, Dec. 1977.

[9] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique," *Computer Networks*, vol. 3, pp. 267–286, Sept. 1979.
[10] J. W. Wong, "Distribution of end-to-end delay in message-switched networks," *Computer Networks*, vol. 2, pp. 44–49, Feb. 1978.

Parviz Kermani (M'79) received the B.S. degree from the University of Tehran, Iran in 1969, the M.S. degree in mathematics from the University of Waterloo, Canada in 1973, and the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA) in 1977.

From 1974 to 1977 he participated in the ARPA Network Project at UCLA as a Postgraduate Research Engineer and did research in the design and evaluation of computer communication networks. In 1978 he received a postdoctoral fellowship from IBM and was a member of the research staff in the Computer Science Department of UCLA. Since December 1978 he has been with the IBM T.J. Watson Research Center, Yorktown Heights, NY, and has been involved in design of routing and flow control mechanisms for computer networks. His current interests are in the area of design, control, and evaluation of data communication networks and distributed systems.



Leonard Kleinrock (S'55–M'64–SM'71–F'73) received the B.E.E. degree from City College, New York in 1957 and the M.S.E.E. and Ph.D.E.E. degrees from the Massachusetts Institute of Technology, Cambridge, in 1959 and 1963, respectively.

In 1963 he joined the faculty of the School of Engineering and Applied Science, University of California, Los Angeles, where he is now a Professor of Computer Science. His research spans the fields of computer networks, computer systems modeling and analysis, queueing theory, and resource sharing and allocation, in general. At UCLA, he directs a group in advanced teleprocessing systems and computer networks. He is the author of three major books in the field of computer networks: *Communication Nets: Stochastic Message Flow and Delay* (New York: McGraw-Hill, 1964; also New York: Dover, 1972); *Queueing Systems, Vol. I: Theory* (New York: Wiley-Interscience, 1975); and *Queueing Systems, Vol. II: Computer Applications* (New York: Wiley-Interscience, 1976). He has published over 100 articles and contributed to several books. He serves as consultant for many domestic and foreign corporations and governments and is a referee for numerous scholarly publications and a book reviewer for several publishers.

Dr. Kleinrock is a Guggenheim Fellow and has received various outstanding teacher and best paper awards, including the 1976 Lanchester prize for the outstanding paper in operations research, and the ICC 1978 Prize-Winning Paper Award. He was recently elected to the National Academy of Engineering in recognition of his pioneering research contributions and educational leadership in the field of computer communications networks.

A Distributed Control Algorithm for Reliably and Consistently Updating Replicated Databases

GEORGES GARDARIN, MEMBER, IEEE, AND WESLEY W. CHU, FELLOW, IEEE

***Abstract*—This paper presents a deadlock-free and distributed control algorithm for robustly and consistently updating replicated databases. This algorithm is based on local locking and time stamps on lock tables which permit detection of conflicts among transactions executed at different sites. Messages are exchanged in the network whenever a transaction commitment occurs, that is, at the end of every consistent step of local processing. Conflicts among remote transactions are resolved by a roll back procedure. Local restart is based on a journal of locks which provides backup facilities. Performance in terms of the number of messages and volume of control messages of the proposed algorithm is compared with that of the voting and centralized locking algorithms. These results reveal that the proposed distributed control algorithm performs, in most cases, comparably to the centralized locking algorithm and better than the voting algorithm.**

***Index Terms*—Concurrency, deadlock, distributed control, locking, lock table, recovery, replicated databases, time stamps, two-step commit.**

Manuscript received January 14, 1980; revised May 22, 1980. This work was supported by IRIA, SIRIUS, and the U.S. Office of Naval Research under Contract N00014-75-C-0650. This is a generalized paper that was presented at the 6th Data Communication Symposium, Pacific Grove, CA, November 1979.

G. Gardarin is with the Institut de Programmation, Le Chesnay, France.

W. W. Chu is with the Department of Computer Science, School of Engineering and Applied Science, University of California, Los Angeles, CA 90024.

I. INTRODUCTION

THE algorithms proposed for updating replicated databases can be separated into two classes: global locking and time stamping. For the global locking scheme, either centralized control [13], [16] or distributed control [4] can be implemented. However, both cases require many control messages. For the time stamping method [17], [1], timing information permits the ordering of transaction updates. However, the addition of a time stamp to every data element

0018-9340/80/1200-1060$00.75