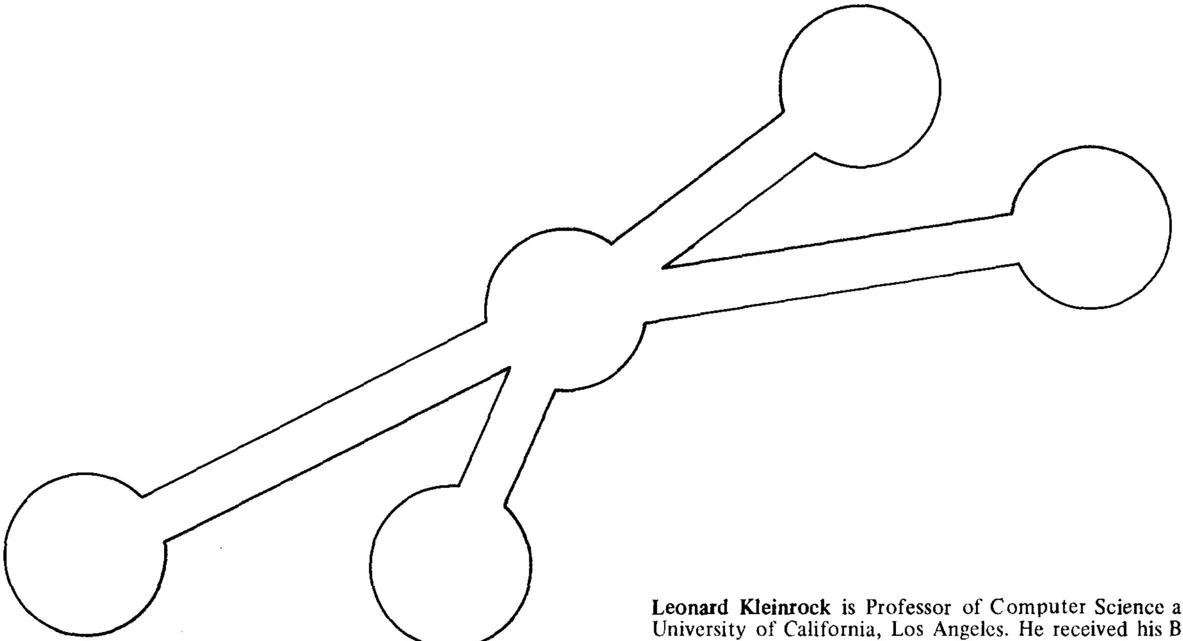
Virtual Cut-Through: A New Computer Communication Switching Technique

Parviz Kermani and Leonard Kleinrock

Computer Science Department, University of California, Los Angeles, Los Angeles, California 90024, USA

In this paper a new switching technique called virtual cut-through is proposed and its performance is analyzed. This switching system is very similar to message switching, with the difference that when a message arrives in an intermediate node and its selected outgoing channel is free (just after the reception of the header), then, in contrast to message switching, the message is sent out to the adjacent node towards its destination before it is received completely at the node; only if the message is blocked due to a busy output channel is a message buffered in an intermediate node. Therefore, the delay due to unnecessary buffering in front of an idle channel is avoided. We analyze and compare the performance of this new switching technique with that of message switching with respect to three measures: network delay, traffic gain and buffer storage requirement. Our analysis shows that cut-through switching is superior (and at worst identical) to message switching with respect to the above three performance measures.



Leonard Kleinrock is Professor of Computer Science at the University of California, Los Angeles. He received his B.E.E. at the City College of New York in 1957 and his M.S.E.E. and Ph.D.E.E. at the Massachusetts Institute of Technology in 1959 and 1963 respectively. In 1963 he joined the faculty of the School of Engineering and Applied Science at the University of California, Los Angeles. 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 large group in advanced teleprocessing systems and computer networks. He serves as consultant for many domestic and foreign corporations and governments and he is a referee for numerous scholarly publications and a book reviewer for several publishers. He was awarded a Guggenheim Fellowship for 1971–1972 and is an IEEE Fellow "for contributions in computer-communication networks, queueing theory, timeshared systems, and engineering education".

* This research was supported by the Advanced Research Projects Agency of the Department of Defense under contract MDA 903-77-C-0272.

Computer Networks 3 (1979) 267–286

1. Introduction

One of the basic problems in a computer communication network design is to select the switching method. Early computer networks were built over the existing telephone networks using principles of telephone switching, namely *line (circuit) switching.* With circuit switching, a complete path of communication links must be set up between two parties before the real communication begins. This set up is accomplished via a signalling message. The path is tied up during the entire session between the two parties. This includes the set up time plus any idle periods during which the parties are silent. Once a path is set up, no further signalling for addressing purposes is necessary; thus, in a circuit switched network, a path, once set up, implicitly provides all the addrsssing information.

When applied to data communication networks, circuit switching suffers from some drawbacks. One is the slow path set up which delays transfer of messages from sender to receiver. Circuit switching was originally used in telephone networks designed for human communication. A telephone conversation is typically an order of magnitude longer than the signalling delay. On the other hand, measurement studies [1] and [2] conducted on time sharing systems indicate that computer communication data streams are bursty. That is, they consists of frequent short messages with occasional long ones. If the circuit is released after each message, then the excessive signalling delay, especially for short messages, is a serious disadvantage of this switching technique. Another drawback is the low channel utilization due to the fact that the channels on a path are tied up but actually are not being used during the idle periods. That is, the dynamic assignment of paths is not dynamic enough. Fig. 1a shows the network delay in a circuit switched system. It is assumed that there is no interfering traffic and that the number of intermediate nodes in the path is 2.

In order to achieve a better channel utilization, one may think of relinguishing the channels on a path during periods in which the parties are silent. This brings us to the idea of *(store-and forward) message switching.* In this method messages are routed toward their destination node without establishing a path beforehand; rather, the paths are assigned dynamically. Through provision of a storage facility at each node, message are stored in intermediate nodes and then are sent forward to a selected adjacent node (hence the name store-and-forward). This selection is made by a well-defined decision rule referred to as the routing algorithm. The process is repeated until the message reaches the destination node. By attaching addressing bits to the header, each message carries information regarding its destination. Since we do not allocate communication links into complete paths for specific source-destination pairs of nodes, each link is statistically shared by many nodes. Fig. 1b shows the network delay in a message switching system.

Examination of Fig. 1b suggests that a better utilization and further reduction in network delay is possible by dividing a message into smaller pieces called packets. This method is called packet switching and is shown in Fig. 1c. In packet switching, each packet (instead of a message) carries its own addressing information. This introduces extra overhead; however, by dividing messages into packets, a message can use a number of links on a path simultaneously. This allows more complete resource sharing (here the resource is the communication link), a higher channel utilization and lower net delay [3].

From Fig. 1b and 1c we observe that the extra delay is incurred because we do not permit a message (or packet) to be transmitted out of a node before it is received completely. We not that when a message arrives in an intermediate node and its outgoing channel is free, it actually does not need to be completely received in the node before being transmitted out. In fact, it can be transmitted out immediately after the outgoing channel is identified, i.e., after the reception of the message header and the selection of outgoing channel by the routing procedure. That is, we need only buffer when we encounter a busy channel, but we may avoid the delay due to unnecessary buffering in front of an idle channel. This is essentially a hybrid mixture of circuit switching and packet switching techniques. Fig. 1d shows the delay when this "*virtual cut-through*" method is used. If the packet encounters busy channels at all of the intermediate nodes, the outcome is exactly the same as in packet switched network. On the other hand, if all of the intermediate channels are free, the outcome is similar to a circuit switched system. Cut-through switching is most advantageous when the network load is low.

In this paper we study and analyze the performance of this virtual cut-through switching technique and compare it with message switching. The development of the analytical models is based on a number of simplifying assumptions. Unless it is necessary for the development of the models, we avoid discussing the physical imple-

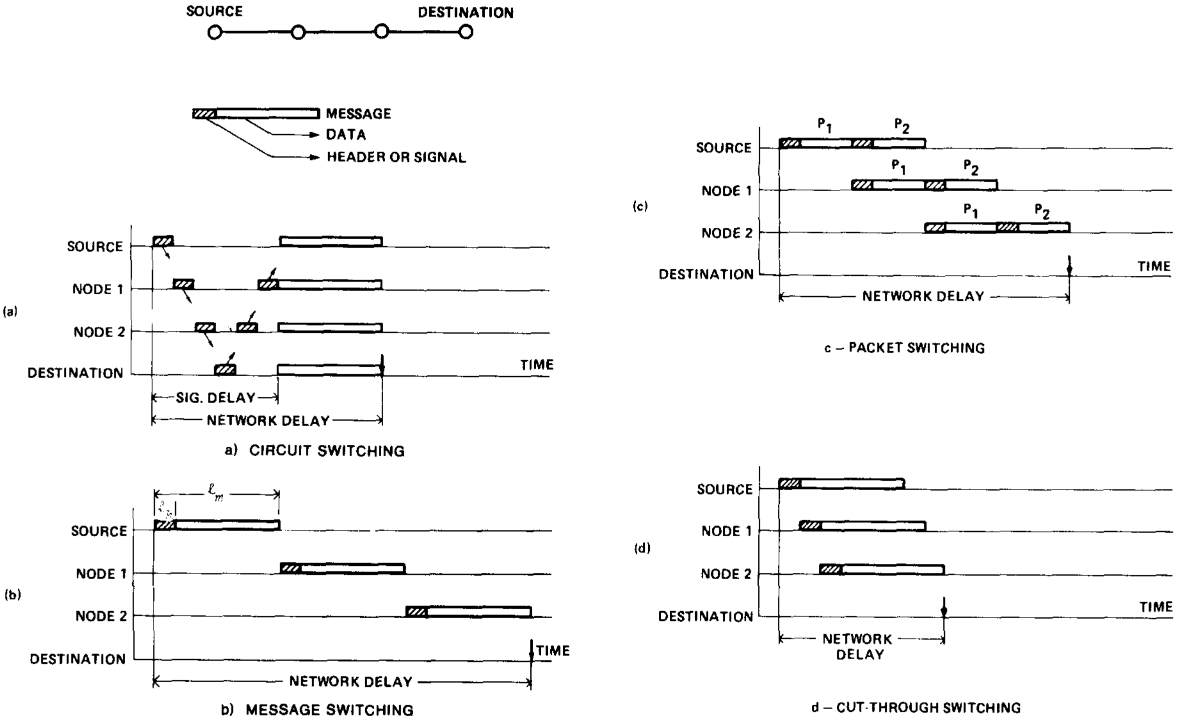


Fig. 1. Operation of Different Switching Systems.

mentation and the protocol design for this switching scheme. These areas are beyond the scope of this paper and require further research. The reader is however cautioned that the quantitative results presented in this paper should be viewed as an approximation (and in some cases, bounds) for the performance of the real system, and that because of practical limitations, the advantage of the proposed scheme may be reduced.

We start with some assumptions and terminology.

2. Assumptions

The essential difference between message and packet switching systems is that in packet switching a message may consist of multiple packets. Measurements of existing networks, in particular the ARPANET [4], show that the average number of packets per message is very close to 1 (1.1 packet per message in the ARPANET); therefore, in this paper we assume that messages consist of single packets and we no longer differentiate between these two systems.

Consider a "tagged" message in a message switching computer net. Whenever the message arrives in an intermediate node in which the outgoing channel is free, we say the message arrives in a "free" node (this happens with probability $(1-P_w)$, P_w being the probability of having to wait). Furthermore, because the message can be sent out immediately after its header (rather than the entire message) is received, we say a "cut-through" has occurred. If, after the reception of the header, the message cannot be sent out immediately (due to busy channels), then it *must* be received completely before being transmitted out (i.e. partial cuts are not allowed). The reasons for this assumption are two-fold. First, in case of noisy channels the message can be error checked when it is blocked (more about this in Section 4.2). Second, the analysis becomes simpler by accepting this assumption.

Throughout the analysis we make the following assumptions

1- External Poisson message arrivals.

2- Exponentially distributed message lengths.
3- Infinite nodal capacity.
4- Deterministic routing.
5- Independence assumption [5].
6- Balanced network (defined below).
7- Negligible Propagation delay.

Definition

A network is said to be *balanced* if the utilization factor of all of its channels is the same. Note that this property does not imply any assumption on the topology of the network.

Assumptions 1 and 2 assign specific distributions to message arrivals and lengths *. Measurement studies [1] on multi-access computer systems justify these assumptions to some extent. In accepting Assumption 2 we are faced with a further implication which is raised by the fact that we will assume each message can fit in a buffer of a constant length. We delay discussion of this limitation until Section 7. Nodal storage limitation can cause blocking, and consequently retransmissions, or loss of messages, hence an increase in delay or a decrease in throughput [6]. It is generally accepted [7], [8], that with a reasonable storage size the above assumption is fairly accurate. Most store-and-forward networks employ a variant of adaptive routing schemes, consequently, the validity of assumption 4 depends critically on the difference of behavior between deterministic and adaptice techniques. Fultz [9] has found that with a proper deterministic routing scheme (e.g., shortest path, etc.) the delay performance of both routing schemes are in close agreement. The implications imposed by the independence assumption are thoroughly discussed in [5]. In short, this assumption is quite acceptable as long as the network does not contain long chains with no interfering traffic. We use Assumption 6 as a means to avoid complexity in our analysis and while we have not made specific assumptions regarding the topology of the network, there are certain topologies (e.g., grid or torus [6]) in which this assumption is easily realizable. For a more detailed discussion regarding implications of the above assumptions, the interested reader is referred to [5] and [6].

The implications and limitations imposed by these assumptions will affect our analysis of both the cut-through as well as the message switching systems. Considering the fact that in our study we mainly compare the performance of these two switching systems with each other, and that these assumptions affect the two schemes equally, the comparison results should be reasonable and close to reality.

The analysis is carried out for multiple channel links. We use the word link for the communication media between nodes. A link of capacity C can be split into $N_{ch}(\geqslant 1)$ channels, each of capacity C/N_{ch}.

3. Analysis of the Number of Cut-throughs

Although the average number of cut-throughs is not a good measure of performance, it is a quantity needed in the evaluation of the network delay. In this section we investigate some of its properties.

Theorem I

The average number and the generating function of the number of cut-throughs in a balanced network with channel utilization ρ are given by

$$\bar{n}_c = (\bar{n}_h - 1)(1 - P_w) \tag{1}$$

and

$$C_\rho(z) = \frac{N[P_w + z(1 - P_w)]}{P_w + z(1 - P_w)} \tag{2}$$

where $\bar{n}_c$ is the average number of cut-throughs, $\bar{n}_h$ is the average number of hops (or path length), P_w is the

* The assumption is that a message consists of data bits and a header. The header is used, among other things, for addressing purposes.

probability of waiting at an arbitrary channel (no cut), $C_\rho(.)$ is the generating function of the number of cut-throughs when the utilization factor of the network channels is ρ and $N[.]$ is the generating function of the number of hops.

Proof: Appendix 1.

For the special case of single channel links, $P_w = \rho$, the utilization of channels, and so Eqs. (1) and (2) reduce to

$$\bar{n}_c = (\bar{n}_h - 1)(1 - \rho) \tag{3}$$

$$C_\rho(z) = \frac{N[\rho + z(1-\rho)]}{\rho + z(1-\rho)} \tag{4}$$

These equations show that when the average number of hops ($\bar{n}_h$) increases, so does the average number of cut-throughs. This is intuitively clear since, as the number of intermediate nodes increases, there is a greater chance to experience more cuts. They also show that $\bar{n}_c$ is a decreasing function of ρ which means that in a less crowded network the average number of cuts is larger.

Special Cases

(I) $\rho = 1$ This implies $P_w = 1$ and $\bar{n}_c = 0$, i.e., no cut-throughs are made.

(II) $\rho = 0$ This implies $P_w = 0$ and $\bar{n}_c = \bar{n}_h - 1$.

In case (I), with probability one, all of the intermediate nodes are busy upon arrival of the message. The network then behaves like a pure message switched system. In (II), with probability one, all of the intermediate nodes are free and all of them are cut through; thus it resembles a circuit switched system. Later in this paper we show performance curves for the system.

4. Delay Analysis

An important performance measure for a computer communication net is the average source to destination message delay T, defined below:

$$T = \sum_{i,j} \frac{\gamma_{ij}}{\gamma} Z_{ij}$$

where γ_{ih} is the average number of messages entering the network per second with origin i and destination j, γ is the total arrival rate of messages from external sources, and Z_{ij} is the average message delay for messages with origin i and destination j.

For a message switched system a straight forward application of the Little's result [10] to the queueing model leads to the following expression for the average network delay, [5]. (Here we use the symbol T_m to indicate that this delay is for message switching systems).

$$T_m = \sum_{i} \frac{\lambda_i}{\gamma} T_i$$

where T_i and λ_i are the average delay and the traffic rate on channel i, respectively.

Calculation of T_i is, in general, a nontrivial and currently unachieved task; however, by accepting the assumptions of Section 2, especially the independence assumption of Kleinrock [5], we are in a position to reduce the network of queues model to the model first studied by Jackson [11]. By virtue of his results, each node behaves stochastically independent of the other nodes and similar to an $M/M/m$ system, for which the average delay is known [12]. For the special case of uniform ρ all of the nodal delays are identical and we have

$$T_m = T_i \sum_i \frac{\lambda_i}{\gamma}$$

In [5] it is shown that $\Sigma_i \lambda_i/\gamma = \bar{n}_h$, so we get

$$T_m = \bar{n}_h T_i \tag{5}$$

So far we have not made any assumption regarding channel error rate. We will initially give an analysis of delay for a network with noiseless channels. It turns out that for such a system the analysis is fairly simple and a closed form expression for network delay is obtainable. However, when the network contains noisy channels, as we will see shortly, the network delay can be calculated through a fairly complicated iterative routine. We will present a delay analysis of these two systems; however, later we deal only with networks with noiseless channels.

4.1. Delay Analysis for a Network with Noiseless Channels

When channels are error free, each nodal delay becomes similar to the system delay of an $M/M/m$ queueing system. So we have

$$T_i = \frac{N_{ch}}{\mu C} + \frac{P_w}{\mu C(1-\rho)}$$

and

$$T_m = \left(\frac{N_{ch}}{\mu C} + \frac{P_w}{\mu C(1-\rho)}\right) n_h \tag{6}$$

where N_{ch} is the number of channels per link, $1/\mu$ is the average message length, and C is the total capacity of the link (notice that capacity of each channel is C/N_{ch}). For the special case of $N_{ch} = 1$ we get

$$T_m = \frac{1}{\mu C(1-\rho)} \bar{n}_h = \frac{\bar{n}_h}{\mu C - \lambda}$$

For the average network delay in the cut-through system we have the following result:

Theorem 2

The average network delay in the cut-through system is given by

$$T_c = T_m - (\bar{n}_h - 1)(1 - P_w)(1 - \alpha)\, t_m \tag{7}$$

where $t_m(=1/\mu C)$ is the average transmission time of a message on a channel, t_h is the average transmission time of a header and $\alpha = t_h/t_m \leqslant 1$ (note that a header is part of a message).

Proof: Appendix 2.

For the special case of $N_{ch} = 1$ and $\alpha = 0$, we have $T_m = \bar{n}_h t_m/(1-\rho)$ and $P_w = \rho$ and Eq. (7) reduces to

$$T_c = \frac{\bar{n}_h t_m}{1-\rho} - (\bar{n}_h - 1)(1-\rho)\, t_m \tag{8}$$

From Eq. (7) we have

$$T_m - T_c = \begin{cases} (\bar{n}_h - 1)(1-\alpha)\, t_m & \text{for } \rho = 0 \\ 0 & \text{for } \rho = 1 \end{cases} \tag{9}$$

which shows that if all of the intermediate nodes are busy ($\rho = 1$ and $P_w = 1$), no reduction in delay is made. On the other hand, when $\rho = 0$ and there is no intermediate busy node, the transmission times of the intermediate channels are saved. This fact becomes clear when we set $a = 0$. For $a = 1$ Eq. (7) gives $T_c = T_m$; this is intuitively clear because $\alpha = 1$ implies that a message is completely composed of its header and the cut-through system is identical to message switching.

4.2. Delay Analysis for a Network with Noisy Channels

A detailed analysis of a network with noisy channels is very complicated. The difficulty is that in a network with unreliable channels there must be some combination of acknowledgment and timeout mechanisms to guarantee correct delivery of messages; modeling of such mechanisms is usually an involved task [13], [14] and [15]. In this paper we accept a very simplified model for this mechanism. For message switching we assume if a message is received with some bits in error (which we assume happens with a constant probability P_e), it is retransmitted immediately (i.e. instantaneous negative and positive acknowledgment). We further assume that with retransmission traffic, the network is still balanced (Definition 1).

If the average delay for one pass transmission is $\bar{s}$, the total nodal delay becomes

$$T_i = \frac{\bar{s}}{1 - P_e} = \left(\frac{N_{ch}}{\mu C} + \frac{P_w}{\mu C(1 - \rho_e)}\right)/(1 - P_e)$$

and the network delay for message switching will be

$$T_m = \left(\frac{N_{ch}}{\mu C} + \frac{P_w}{\mu C(1 - \rho_e)}\right)\bar{n}_h/(1 - P_e) \tag{10}$$

In the above expression ρ_e is the effective network traffic. The useful traffic is given by

$$\rho = (1 - P_e)\,\rho_e \tag{11}$$

For the cut-through system the operation is somewhat more involved. We assume that a message is error checked only in its destination node or any intermediate node which it cannot cut through; we call this type of node a *final node* (which is not necessarily the same as the destination node although the destination node is always a final node). It is not that a message cannot be checked for error in an intermediate node through which it makes a cut, but even if it is recognized that a message contains some bits in error (this can be done only after the last bit of a message arrives in a node), the message has already started its transmission out of the node and there is no way to stop the transmission and to notify the next node(s) not to accept the message. In a final node, because the message is received completely, error checking (and stopping) can be done. If it is recognized that the message is erroneous, it has to be retransmitted through all of the intermediate nodes (including the first node on the cut-through path). As before, we further assume that acknowledgments (positive and/or negative) are instantaneous. Fig. 2 makes these operations clear. In this figure a message is to be transmitted from source node S to destination node D. The message has already made its way to an intermediate node i (in which it has been blocked). When it starts transmission from this node, it makes cuts through nodes $i + 1$, $i + 2$ and $i + 3$ and is again blocked at node $i + 4$. At this node, however, after error checking, the message is found to have some bits in error. Through a negative acknowledgment (which we assume is instantaneous), it is retransmitted from node i. In the retransmission process, however, it can make a cut only through node $i + 1$ and is blocked at node $i + 2$. Depending on whether it is received incorrectly or not, further retransmission from node i is necessary or it can start making its way toward its destination node D (after queueing delay) *.

The delay analysis for the cut-through system in a network with noisy channels is fairly complicated and in fact no closed form expression for it has been derived. The conditional delay, when the number of hops is constant ($\tilde{n}_h = n$) is given below.

* Note that in cut-through switching a header should carry information regarding not only the source-destination, but also the last node in which the message was blocked (the last final node). This results in a larger header length for cut-through switching than message switching. We will ignore this fact in our analysis in order not to make the derivations complex. The reader is however cautioned that the performance of the cut-through system will be lower than what is predicted by our model.

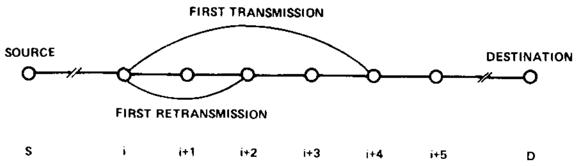


Fig. 2. Example of Retransmission in the Cut-Through System.

Theorem 3

The average network delay for the cut-through system in a network with noisy channels is given by

$$T_c = T_c^{(n)}(n) \tag{12}$$

where

$$= \sum_{k=0}^{j-1} \{T_{cc}^{(k)}(i, j) + T_c^{(i-k+1)}(i)(1 - P_e)^{k+1} + T_c^{(j)}(j)[1 - (1 - P_e)^{k+1}]\} P_c^{(j)}(k) \qquad (i > j > 0) \tag{13}$$

$$T_c^{(j)}(i) = \frac{\sum_{k=0}^{j-1} [T_{cc}^{(k)}(i, j) + T_c^{(j-k+1)}(i)(1 - P_e)^{k+1}]\, P_c^{(j)}(k)}{\sum_{k=0}^{j-1} (1 - P_e)^{k+1} P_c^{(i)}(k)} \qquad (i = j > 0)$$

$$= 0 \qquad (j = 0)$$

and

$$T_{cc}^{(k)}(i, j) = \begin{cases} \mathrm{E}[\tilde{s}] + kt_h = \dfrac{N_{ch}}{\mu C} + \dfrac{P_w}{\mu C(1 - \rho_e)} + kt_h & (i = j) \\ \mathrm{E}[\tilde{s} | \tilde{w} > 0] + kt_h = \dfrac{N_{ch}}{\mu C} + \dfrac{1}{\mu C(1 - \rho_e)} + kt_h & (i > j) \end{cases} \tag{14}$$

and

$$P_c^{(j)}(k) = \begin{cases} (1 - P_w)^k P_w & (0 \leqslant k < j - 1 \text{ and } j > 1) \\ (1 - P_w)^k & (k = j - 1) \end{cases} \tag{15}$$

where $\tilde{s}$ and $\tilde{w}$ are the service time and the waiting time in a node, respectively, ρ_e is the effective traffic and n is the path length.

For a proof of this theorem the interested reader is referred to [14].

If we set $P_e = 0$, after some algebra Eq.(12) reduces to Eq. (7), the delay for the cut-through system in a network with noiseless channels.

The useful traffic is related to effective traffic according to the following theorem.

Theorem 4

For an n-hop cut-through system in a balanced network with noisy channels, the useful traffic ρ is determined according to the following relationships

$$\rho = \frac{n}{N_t(n)} \rho_e \tag{16}$$

where

$$N_t(n) = \begin{cases} \dfrac{\sum\limits_{k=0}^{n-1;} [(k+1) + N_t(n-k-1)(1-P_e)^{k+1}]\, P_c^{(n)}(k)}{\sum\limits_{k=0}^{n-1} (1-P_e)^{k+1} P_c^{(n)}(k)} & (n > 0) \\ 0 & (n = 0) \end{cases} \tag{17}$$

where $P_c^{(j)}(k)$ is given by Eq. (15). $N_t(n)$ is the total number of times a message is transmitted when the path length is n; note that $N_t(n) \geqslant n$.

For a proof of this theorem the reader is referred to [14].

Let us consider some special cases of interest

I- If $P_w \to 1$ then it can be shown that Eqs. (10) and (12) become identical. This is intuitively clear as in this case $\overline{n}_c = 0$ and retransmissions take place only over one hop in both systems.

II- If $n = 1$, again Eqs. (10) and (12) become identical (on a one-hop path there cannot be any cut-throughs!).

III- If $P_e = 0$ then both Eqs. (11) and (16) reduce to $\rho = \rho_e$. Clearly, in this case there is no retransmission and the effective traffic is the same as the useful traffic.

In Section 6 we present some performance curves which show the effect of channel errors on network performance, with the exception of these cases, we assume that network channels are noiseless in the remainder of this paper.

5. Traffic Gain

Analysis of Eq. (7) shows that

$$T_m(\rho) - T_c(\rho) \geqslant 0 \qquad 0 \leqslant \rho \leqslant 1$$

(here we use $T_m(\rho)$ and $T_c(\rho)$ to indicate the delay at a given traffic load, ρ). Below, we present performance curves which explicitly show this fact. This property indicates that at the same traffic level the network delay in a cut-through system is less than in message switching. Equivalently, for the same network delay the cut-through system can handle more traffic; see the sketch Fig. 3 in which p_m and p_c are defined as the throughput carried by message switching and cut-through switching, respectively, at a constant network delay. In this section we calculate the throughput increase ($\rho_c - p_m$). (Actually, what we find is the difference between channel utiliza-

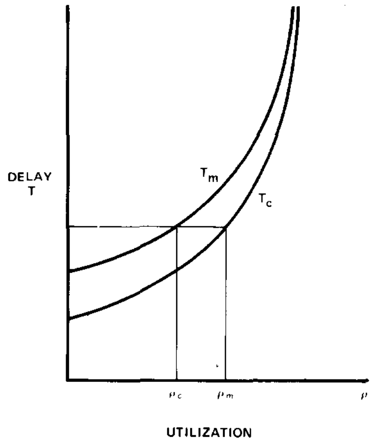


Fig. 3. Sketch of Traffic Gain in the Cut-Through System.

tion of the two systems at the same delay; this gives us a measure of traffic gain).

Setting $T_m(\rho_m) = T_c(\rho_c)$ we get the condition

$$\bar{n}_h\left(\frac{N_{ch}}{\mu C}+\frac{P_{w_m}}{\mu C(1-\rho_m)}\right)=\bar{n}_h\left(\frac{N_{ch}}{\mu C}+\frac{P_{w_c}}{\mu C(1-\rho_c)}\right)-(\bar{n}_h-1)(1-P_{w_c})(1-\alpha)\frac{N_{ch}}{\mu C} \tag{18}$$

where P_{w_m} or P_{w_c} are the waiting probabilities at a node for which the utilization factor is ρ_m or ρ_c, respectively (recall that we are assuming noiseless channels).

For $N_{ch} > 1$, P_w is a complicated expression of ρ and N_{ch} and it is not possible to find ρ_c in terms of ρ_m explicitly. For this reason we solve Eq. (18) analytically only for the case where $N_{ch} = 1$. For higher values of N_{ch} numberical evaluation techniques are necessary. For $N_{ch} = 1$ we have $P_w = \rho$ and Eq. (18) reduces to

$$1-\rho_m=\frac{1-\rho_c}{1-\eta(1-\rho_c)^2(1-\alpha)}$$

where $\eta = (\bar{n}_h - 1)/\bar{n}_h$, $0 \leqslant \eta < 1$. From the previous equation we obtain the throughput to be

$$\rho_c-\rho_m=\frac{1+2\eta(1-\alpha)(1-\rho_m)^2-[1+4\eta(1-\alpha)(1-\rho_m)^2]^{1/2}}{2\eta(1-\alpha)(1-\rho_m)} \tag{19}$$

When $\bar{n}_h = 1$ $(\eta = 0)$, then $\rho_c = \rho_m$; i.e., there is no gain. This is the case for a fully connected net. For $\bar{n}_h \to \infty (\eta \to 1)$ we have

$$\lim_{\bar{n}_h\to\infty}(\rho_c-\rho_m)=\frac{1+2(1-\alpha)(1-\rho_m)^2-[1+4(1-\alpha)(1-\rho_m)^2]^{1/2}}{2(1-\alpha)(1-\rho_m)} \tag{20}$$

Note that when $\bar{n}_h \to \infty$, T_m and T_c both grow to infinity; however, in the limit, there is still a throughput gain in using the cut-through system.

6. Some Performance Curves and Simulation Results

Fig. 5 shows the number of cuts as a function of ρ, the channel utilization as given in Eq. (1). The network configuration and traffic pattern is shown in Fig. 4. This is a tandem queue in which, except for the first and the last 2 nodes, traffic passes through 3 hops (2 internediate nodes). The reason for selecting this topology was to create a uniform traffic load on each channel. For a single channel system $\bar{n}_c$ is a linear function of ρ. Fig. 5 also shows some simulation results on this tandem queue model for $N_{ch} = 1$, 2, and 4. In the simulation program we used all the assumptions of Section 2, but we assumed the message lengths to be of constant length (instead of exponentially distributed), and also did not use the independence assumption. As our analytic model predicts, when the traffic is light ($\rho \approx 0$), the average number of cuts is equal to $(\bar{n}_h - 1)$ which is 1.6 for our network; whereas, for a heavily loaded system ($\rho \approx 1$) the average number of cuts becomes zero. In spite of the differences between the underlying assumptions of the analytic and the simulation model, the match between the simulated and the analytic results is very encouraging.

Figs. 6a to 6c show the normalized delay curves (μCT_c) as a function of useful channel traffic. Recall that throughput is not necessarily equal to the effective channel traffic; they are related to each other according to

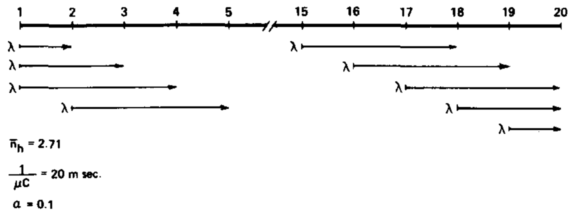


Fig. 4. The Simulated Tandem Queues.

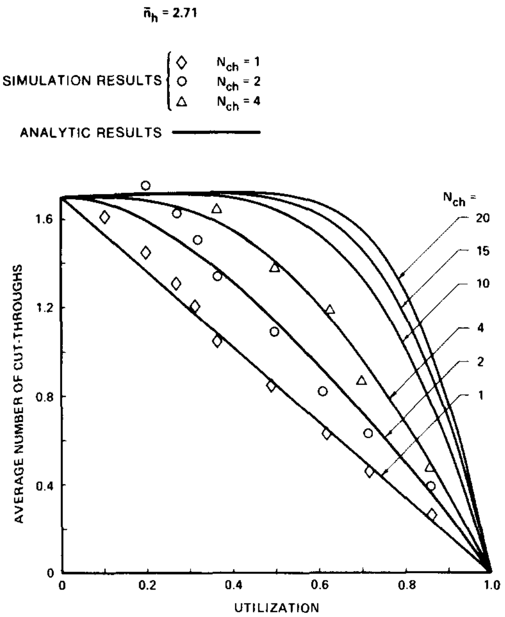


Fig. 5. Average Number of Cut-Throughs.

Eqs. (11) and (16). The curves shown are for a number of cuts equal to 3 and for different values of $\alpha(=t_h/t_m)$ and P_e (channel error rate). Although channel error rate affects performance of both message switching and the cut-through system, it is interesting to note that this parameter has a more serious effect in the cut-through system than in the message switching system. In fact for a very noisy channel (large P_e) the performance of message switching may become better than the cut-through system (for the parameters of these graphs this happens approximately when $P_e > 0.4$). The reason for this phenomenon is that in the cut-through system, if a retransmission becomes necessary, it must take place over more than one hop; Eq. (16) makes this fact clear. Fig. 6c shows this effect is addition to a rather interesting phenomenon. Here the channel error rate is (unreasonably) high, $P_e = 0.45$. When traffic is low the chance of making many cut-throughs is high, but the received messages therefore have a greater chance of having bits in error; thus the retransmission overhead and hence the delay is high. As traffic increases, althoug the cut-through properties are less effective, we are saved unnecessary long retransmissions and the network delay decreases. Further increase in traffic results in higher delay due to queueing phenomena.

Perhaps a better comparison is to find the ratio T_c/T_m. We have

$$\frac{T_c}{T_m} = 1 - \frac{\bar{n}_h - 1}{\bar{n}_h} \frac{(1 - P_w)(1 - \alpha)N_{ch}}{N_{ch} + P_w/(1 - \rho)} \tag{21}$$

This ratio is shown in Figs. 7a to 7c. We see from Eq. (21) and from Fig. 7a that when $\alpha = 0, N_{ch} = 1$ and $\rho = 0$ then T_m is $\bar{n}_h$ times larger than T_c. This is intuitively clear since, in the cut-through system, messages can snake through all of the intermediate nodes without experiencing any delay (recall that $\alpha = 0$); however, in message switching the delay incurred at each intermediate node is t_m, the transmission time of a message over a single channel. As the channel utilization (or the normalized throughput) approaches 1, cut-through switching and message switching delay become identical (see also Figs. 5 and 6).

Some limiting cases

Let us study the limiting value of T_c/T_m as a function of the variables α, $\bar{n}_h$ and $N_{ch}(0 \leqslant \alpha \leqslant 1, 1 \leqslant \bar{n}_h, 1 \leqslant N_{ch})$.

When $\bar{n}_h = 1$ (i.e. in a fully connected net), then, as we expect, $T_c/T_m = 1$ for all possible values of ρ, the utili-

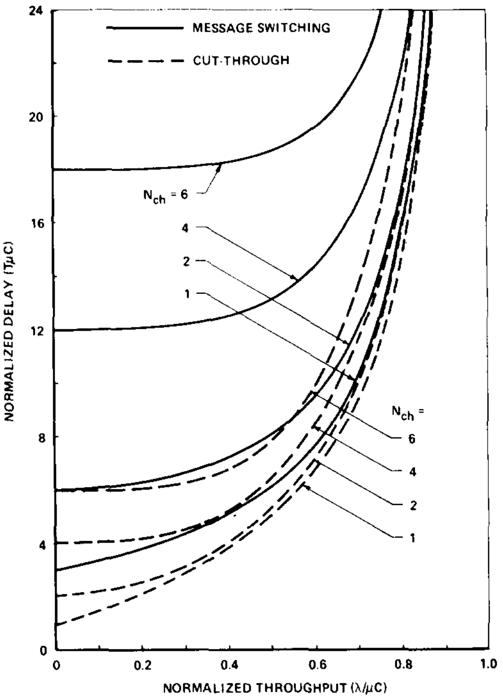


Fig. 6a. Delay vs. Throughput.

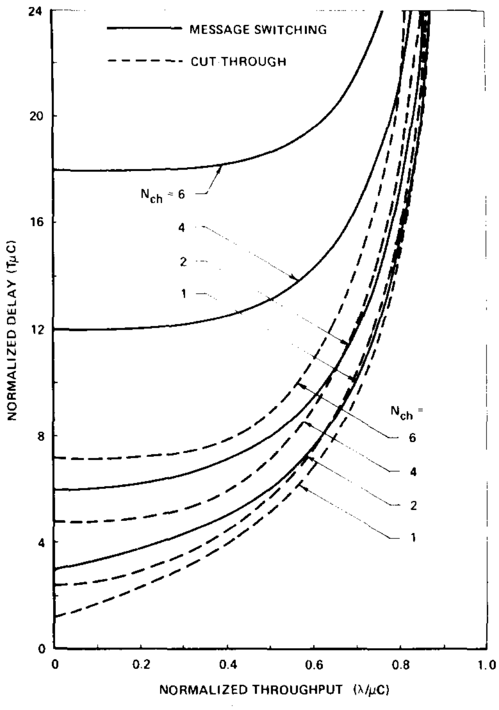
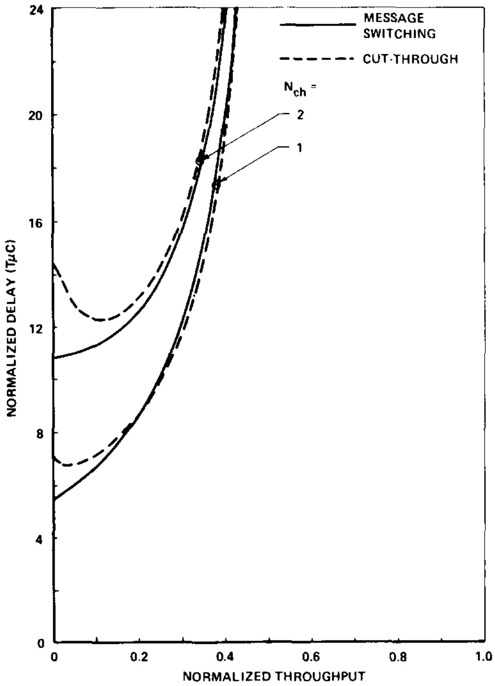


Fig. 6b. Delay vs. Throughput.



Fjg. 6c. Delay vs. Throughput

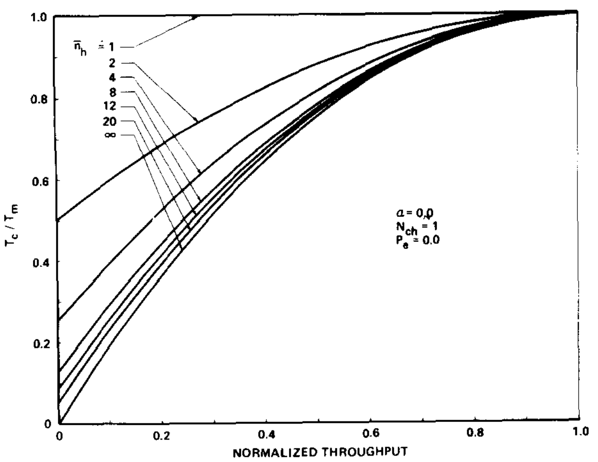


Fig. 7a. Ratio of the Delay in the Two Switching Systems

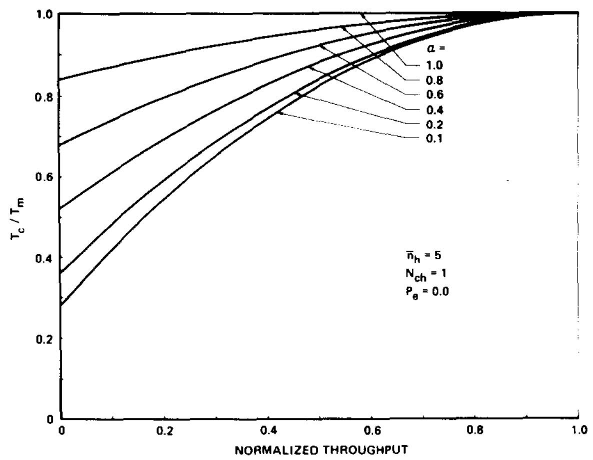


Fig. 7b. Ratio of the Delay in the Two Switching Systems.

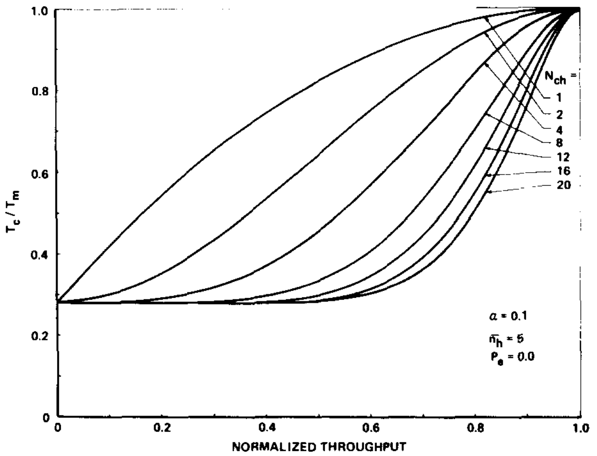


Fig. 7c. Ratio of the Delay in the Two Switching Systems.

zation, as shown in Fig. 7a. For $\bar{n}_h \to \infty$ the ratio becomes

$$\lim_{\bar{n}_h \to \infty} \frac{T_c}{T_m} = 1 - \frac{(1 - P_w)(1 - \alpha) N_{ch}}{N_{ch} + P_w/(1 - \rho)} \tag{22}$$

The interesting point is that, even though T_c and T_m both become unbounded as $\bar{n}_h \to \infty$, their ratio has a well defined limit. In fact Eq. (21) shows that T_c/T_m is a strictly decreasing function of $\bar{n}_h$; the longer the path length, the more advantageous is cut-through.

Looking at the dependency of T_c/T_m on α, it is no surprise that Eq. (21) shows that smaller values of α result in a better performance. In fact for $\alpha = 1$, both system are identical; this behavior is shown in Fig. 7b.

The dependency of T_c/T_m on N_{ch} is shown in Fig. 7c. As expected, we see that larger values of N_{ch} result in a lower value of T_c/T_m.

Some simulation results for the throughput-delay performance are shown in Fig. 8. The simulated network is once again the tandem queue system shown in Fig. 4. The discrepancy between the simulated and the calculated values of T_c and T_m is the result of the topology of the network and the message length distribution. The analytic

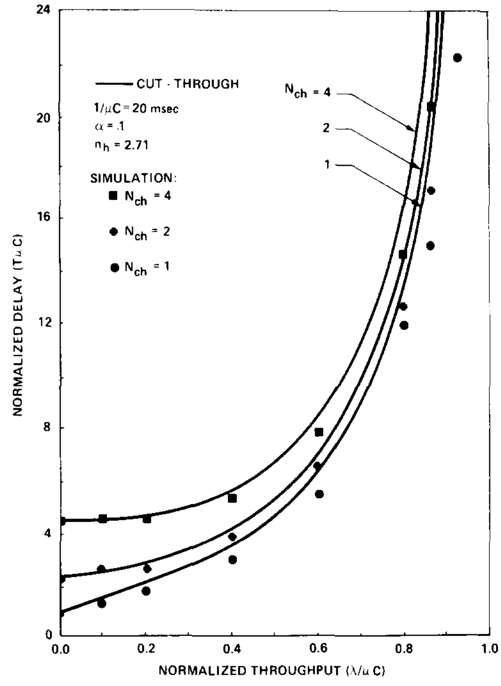


Fig. 8. Some Simulation Results.

results are for exponentially distributed message lengths, whereas in the simulation program we have used fixed length messages.

The traffic gain is shown in Fig. 9; The curves are for single channel links (Eq. (19)). This figure shows that at the same network delay, cut-through switching can carry more traffic than message switching, and that the difference is higher when the network is lightly loaded.

The curves show that higher values of $\overline{n}_h$ or smaller values of α result in a better gain. For $\overline{n}_h = 1$ or $\alpha = 1$ both systems behave identically.

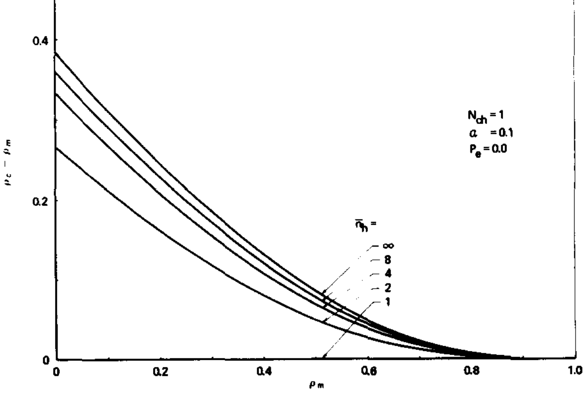


Fig. 9. Traffic Gain.

7. Storage Requirements

In a store-and-forward message switching system, buffer storage is provided at the switching nodes. Although the cost of storage is decreasing, it nevertheless is an important component of the overall network cost. The purpose of this section is to calculate the storage requirements for the cut-through system and compare it with the requirements of the message switching system. Throughout this section we deal with noiseless channels. We start with some assumptions regarding buffer allocation in the two systems.

7.1. Buffer Allocation in Message Switching

Storage is divided into buffers of equal size which can accommodate a message of maximum length. Each buffer can be used by only one message at a time, even though such a message may not fill its entire buffer. Within a node, a message is never allocated more than one buffer. With this buffer management, one might think that the assumption of exponentially distributed message length should be replaced with a truncated exponential distribution. However, measurements of the existing networks, expecially the ARPANET [4], show that the average message length is a small fraction of the maximum packet length (250 bit average compared to 1000 bit maximum for the ARPANET). In such cases where the average message length is much smaller than the buffer size, we find that the exponential message length assumption is not unreasonable. Regarding the messages which are being transmitted, we assume that when the first bit of a message arrives in a node, a full buffer is immediately allocated to the message. This means that all messages in transmission use two buffers simultaneously (one in the node transmitting and one in the mode receiving). We assume that acknowledgment is instantaneous. As a result, the moment a message is completely received at a node, its source buffer is relinquished.

7.2. Buffer Allocation in Cut-Through Switching

There are two types of buffers in this system: message buffers and header buffers (with lengths l_m and l_h, respectively). A header buffer is used by a message when it cuts through a node. The buffer is allocated when the first bit of a message arrives at a node. After the reception of the message header, if it is recognized that the selected outgoing channel is free, no further storage is allocated to this message and it starts being transmitted out (however, this header buffer is used while the message is snaking through the node). If, however, the outgoing channel is busy, then the message must be completely received in the node and (the remainder of) a message buffer is allocated to it (this happens at a final node, i.e. the destination node or any intermediate node where a cut is not possible). We note that a message may be using a number of header buffers simultaneously, but it can use at most two message buffers at one time. Regarding the acknowledgment, we assume that a copy of a message is kept at the source node, or at any intermediate node in which it is blocked (i.e. through which it cannot make a cut), until it is completely received at a succeeding final node. If this final node is the destination node then, after completion of transmission out of the network, both message buffers are released simultaneously. If, however, this final node is not the destination node then a copy of the message is kept in this final node and the preceding message buffer is released (i.e. after reception of the first acknowledgment, the copy of the message at the source node is dropped and its buffer is released).

7.3. Storage Requirement: Results

Let $\bar{S}_m$ be the expected value of the total amount of storage required at a node in message switching and let $\bar{S}_c$ be the same quantity for the cut-through system. As before, we use l_m and l_h for message buffer and header buffer size. We also assume that

$$\alpha = \frac{t_h}{t_m} = \frac{l_h}{l_m} = \frac{\text{header buffer size}}{\text{message buffer size}}$$

For storage requirements in the message switching and the cut-through switching systems we have the following theorem.

Theorem 5

The expected amount of storage required at a node in message switching and cut-through switching are given by:

$$\bar{S}_m = (\bar{N}_m + N_{ch}\rho)\, l_m \tag{23}$$

and

$$\bar{S}_c = \left\{\bar{N}_m + N_{ch}\rho\left(1 - 2\frac{\bar{n}_h - 1}{\bar{n}_h}(1-\alpha)(1-P_w)\right)\right\} l_m \tag{24}$$

where N_m is the average number of messages at each node.

Proof: See Appendix 3.

Note that if $P_w = 1$ and/or $\alpha = 1$ (i.e. no cut is possible or a message carries only addressing and control information) then $\bar{S}_m = \bar{S}_c$. To compare storage requirements of the two switching methods, we study the behavior of the ratio $\phi = \bar{S}_c/\bar{S}_m$

$$\phi = \frac{\bar{N}_m + N_{ch}\rho\left(1 - 2\frac{\bar{n}_h - 1}{\bar{n}_h}(1-\alpha)(1-P_w)\right)}{\bar{N}_m + N_{ch}\rho} \tag{25}$$

From Eq. (25) it is clear that $\bar{S}_c \leqslant \bar{S}_m$, and equality holds only when $P_w = 1$ or $\alpha = 1$. This shows that performance of the cut-through system, in terms of storage requirement, is never worse than the message switching system.

The ratio ϕ is a decreasing function of N_{ch}, the number of channels. For $N_{ch} \to \infty$ we have the limiting value of ϕ given below:

$$\lim_{N_{ch}\to\infty} \phi = 1 - \frac{\bar{n}_h - 1}{\bar{n}_h}(1-\alpha) = \frac{1 + (\bar{n}_h - 1)\,\alpha}{\bar{n}_h} \tag{26}$$

which is independent of the utilization factor ρ. Fig. 10 shows the behavior of ϕ as a function of N_{ch} for different values of ρ. Note that the true limiting value should be

$$\frac{2 + (\bar{n}_h - 1)\,\alpha}{2\bar{n}_h}$$

which can be found by considering the fact that in the limit ($N_{ch} \to \infty$) there are no waiting messages in the

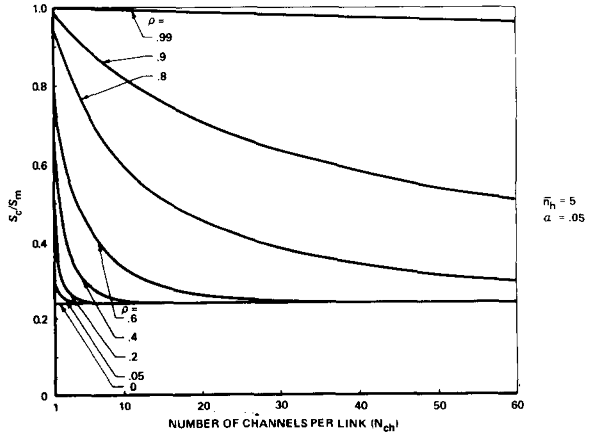


Fig. 10. Comparison of Storage Requirement in the Cut-Through System and the Message Switching System.

system ($P_w = 0$). In the case of cut-through switching, all of the messages are cutting through; and in the case of message switching, all of the messages in the system are in transmission. The discrepancy is the result of our approximation in calculating $\overline{S}_c$ (see Appendix 3). For our tandem network example, the limiting value found by Eq. (26) is 0.24, which should be compared with the true value of 0.22. Fig. 10 shows that the saving in storage is high when the network traffic light. We should point out that, while the storage requirement of both switching techniques is high when the network channels are noisy, the degradation in performance is higher for the cut-through system than the message switching system.

8. Conclusions

In this paper we proposed and analyzed a new switching system, named "Virtual Cut-Through". In this switching system a message is buffered in an intermediate node only if it must (due to busy channels); hence the delay due to unnessary buffering in front of an idle channel is avoided. If a message cuts through all of the intermediate nodes, the system becomes similar to circuit switching. On the other hand, if a message encounters busy channels at all of the intermediate nodes, the outcome is the same as message switching. Therefore, cut-through switching is essentially a hybrid mixture of circuit switching and message switching techniques which dynamically adjusts its behavior between these two as a function of load. It takes advantage of the good properties of message switching, by using multiple links along a path in a demand access fashion; meanwhile, it exhibits the nice properties of circuit switching, by not incurring unnecessary intermediate node delay for store-and-forward.

Our analysis showed that cut-through switching is superior to message switching from at least three points of view: network delay, traffic gain and buffer storage requirement. The analysis in Section 4 showed that in general the network delay for the cut-through system is less than the message switching system, and that only when the channel error rate is too high may the message switching delay become less than the cut-through switching delay (Figs. 6a to 6c). In Section 5 we showed that at the same network delay, cut-through switching can carry more throughput than message switching (Figs. 7a to 7c), and in Section 7 we demonstrated that storage requirements for cut-through switching is less than message switching (Fig. 10).

When a message can cut through all of the intermediate nodes the network path becomes virtually identical to a physical connection between two remote stations. This property is desirable for certain applications. In [14] an algorithm has been developed which maximizes the throughput while guaranteeing a probability of one that the network looks like a physical connection.

Our study in this paper showed that cut-through switching is a good solution to the switching problem in a communication network. In [14] it is shown that in many cases the performance of cut-through switching is even better than circuit switching.

As we pointed out in the opening remarks of the paper, because of physical and practical implications, the analytic results presented in this paper should be viewed as upper/lower bound values. While our assumptions in general affect the two systems equally, in some cases they tend to favor cut-through switching more than message switching. For example in a computer network there are various kinds of control information which should be passed around (control packets). We have completely ingnored this type of traffic in our analysis. One simple way to consider this traffic is to note that there are not data bits in a control packet, i.e., the entire packet consists of addressing and control information; for such packet $\alpha = 1$. We can then use $\alpha = 1$ for the fraction of the traffic that consists of control packets.

Appendix 1. *Proof of Theorem 1*

Whenever a message enters a free node it can make a cut. This event occurs with probability $(1 - P_w)$. Due to the independence assumption, the number of cuts has a binomial distribution and we have:

$$\Pr[\tilde{n}_c = k \mid \tilde{n}_h = n, \rho] = \binom{n-1}{k} (1 - P_w)^k P_w^{n-k-1} \quad 0 \leqslant k \leqslant n-1$$

The conditional generating function [12] of the number of cut-throughs may be written as

$$C_{n,\rho}(z) = \mathrm{E}[z^{\tilde{n}_c}|\tilde{n}_h = n, \rho] = \sum_{k=0}^{n-1} z^k \binom{n-1}{k} (1 - P_w)^k P_w^{n-k-1}$$

So,

$$C_{n,\rho}(z) = [P_w + z(1 - P_w)]^{n-1}$$

We may now find the conditional mean number of cut-throughs:

$$\bar{n}_c(n, \rho) = \mathrm{E}[\tilde{n}_c|\tilde{n}_h = n, \rho] = \frac{dC_{n,\rho}(z)}{dz}\bigg|_{z=1} = (n - 1)(1 - P_w)$$

The mean number of cut-throughs is therefore

$$\bar{n}_c = \sum_n \mathrm{E}[\tilde{n}_c|\tilde{n}_h = n, \rho] \Pr[\tilde{n}_h = n] = (\bar{n}_h - 1)(1 - P_w)$$

The unconditional generating function of the number of cut-throughs is simply

$$C_\rho(z) = \sum_n C_{n,\rho}(z) \Pr[\tilde{n}_h = n] = \sum_n [P_w + z(1 - P_w)]^{n-1} \Pr[\tilde{n}_h = n]$$

or

$$C_\rho(z) = \frac{N[P_w + z(1 - P_w)])}{P_w + z(1 - P_w)}$$

where $N[z] = \mathrm{E}[z^{\tilde{n}_h}]$ is the generating function of the number of hops.

Appendix 2. *Proof of Theorem 2*

For each node at which a cut-through is made, a nodal service time is saved. However, this service time is conditioned on the event that the waiting time is zero. So we have

$$T_m - T_c = \bar{n}_c \,\mathrm{E}[\tilde{s}|\tilde{w} = 0]$$

where $\tilde{s}$ is the total delay in a node and $\tilde{w}$ is the waiting time in a node. Considering the fact that $\mathrm{E}[\tilde{s}|\tilde{w} = 0] = t_m$, and that a message can be sent out only after its header is received, the previous equation is changed to

$$T_m - T_c = \bar{n}_c(t_m - t_h)$$

where $(t_m - t_h)$ is the saving in delay at each node when the cut-through method is used. Using the value of $\bar{n}_c$ from Eq. (1), we get

$$T_c = T_m - (\bar{n}_h - 1)(1 - P_w)(1 - \alpha)\, t_m$$

Appendix 3. *Proof of Theorem 5*

Let $\bar{S}_m^{(\rho)}$ be the average value of the total amount of storage required on a path in message switching. At any instant a message in transmission occupies two buffers. All other messages occupy only one buffer, and so we have

$$\bar{S}_n^{(\rho)} = [\bar{N}_m^{(\rho)} + \bar{n}_h(N_{ch}\rho)]\, l_m \tag{A3.1}$$

where

$$\bar{N}_m^{(\rho)} = \mathrm{E}[\text{number of messages on a path for message switching}] = T_m\lambda$$

and $(N_{ch\rho})$ is the average number of messages which are being transmitted at each node. The expected number of buffers required at each node is therefore $\bar{S}_m = \bar{S}_m^{(p)}/\bar{n}_h$. Using Eq. (A3.1) we get

$$\bar{S}_m = (\bar{N}_m + N_{ch}\rho)\, l_m \qquad \text{(A3.2)}$$

where $\bar{N}_m$ is the average number of messages at each node. This proves the first part of Theorem 5.

To prove Eq. (24) we must describe the operations of the cut-through system in more detail. Let

$$\delta = \text{ceiling } \{1/\alpha\}$$

$$= \text{ceiling } \{(\text{msg length})/(\text{header length})\}$$

(ceiling{x} = smallest integer larger than or equal to x). Consider a message which is being transmitted along a path. Every time that the message cuts through a node, it is allocated a header buffer, i.e., the total amount of storage allocated to it is increased by one header buffer. This process continues until the number of the consecutive cuts exceeds δ, after which, for each new header buffer that the message is allocated, it releases one. Fig. 11a shows this phenomenon for the case $\delta = 4$. The solid lines show the amount of storage allocated to the message while it is being transmitted along the path, Fig. 11b. At time t_1 the message starts transmission from node 1, the source node. Because we have assumed propagation delay is negligible, at the same time a header buffer is allocated to the message at node 2. After t_h seconds the header is received completely at node 2 (because it can make a cut through this node) and it starts transmission towards node 3 (while using the header buffer at node 2). At node 3 another header buffer is allocated to it instantaneously. This process continues and the amount of the allocated storage increases up to node 5; however, from node 6 on (at time t_3), the tail of the message starts leaving nodes 2, 3, ...; for each new header buffer it occupies, it releases one, hence the total amount of storage allocated remains unchanged. At time t_4 the message arrives in node 8 and a header buffer is allocated to it (so, there is no change in the total amount of buffer allocated to this message on the path). But, after t_h seconds (at

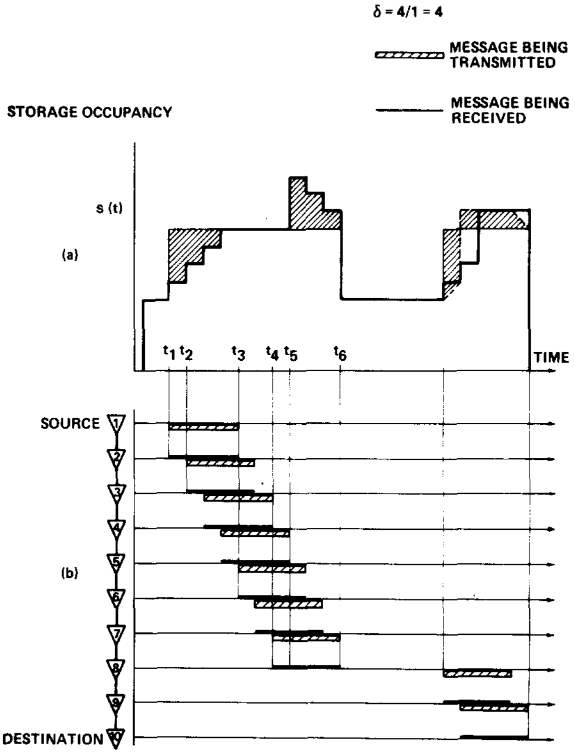


Fig. 11. Storage Occupancy in the Cut-Through System.

time t_5) when the header is received, it is recognized that the selected outgoing link is busy. So the message is blocked and (the remainder of) a message buffer is allocated to it. As the tail of the message leaves the intermediate nodes, header buffers are released. When the message is received in full at node 8 (at time t_6), the copy of the message present at node 1 (the source node) is dropped and its buffer is released. The right part of Fig. 11a shows the situation where the number of consecutive cuts is less than δ.

Let $s(t)$ be the amount of storage allocated to a message at time t. If we approximate the step function $s(t)$, drawn in solid lines in Fig. 11a, by the broken lines shown in the same figure, we observe a very useful property. The shaded areas in this figure are equal. The interesting fact is that when the number of consecutive cuts is larger than δ, there is no approximation involved (left part of Fig. 11a). By using this property and the assumptions stated above, it can be shown that

$$\overline{S}_c^{(p)} = \{\bar{n}_h \bar{N}_m + N_{ch}\rho[\bar{n}_h - 2(\bar{n}_h - 1)(1 - \alpha)(1 - P_w)]\}/m \tag{A3.3}$$

where $\overline{S}_c^{(p)}$ is the expected number of buffers required on a path for cut-through switching. The expected number of required buffers at a node is, therefore, $\overline{S}_c = \overline{S}_c^{(p)}/\bar{n}_h$, or

$$\overline{S}_c = \left\{\bar{N}_m + N_{ch}\rho\left[1 - 2\frac{\bar{n}_h - 1}{\bar{n}_h}(1 - \alpha)(1 - P_w)\right]\right\}/m \tag{A3.4}$$

(For a detailed derivation of Eqs. (A3.3) and (A3.4) the interested reader is referred to [14].)

References

[1] Jackson, P.E. and C.D. Stubbs. "A Study of Multi-access Computer Communications," *AFIPS Conference Proceedings,* SJCC, 1969, Vol. 34, pp. 491–504.

[2] Fuchs, E. and P.E. Jackson. "Estimates of Distributions of Random Variables for Certain Computer Communication Traffic Models," *Communications of the ACM,* Vol. 13 No. 12, December 1970, pp. 752–757.

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

[4] Kleinrock, L. and W.E. Naylor. "On Measured Behavior of the ARPA Network," *AFIPS Conference Proceedings,* NCC, Chicago, May 1974, Vol. 43, pp. 767–780.

[5] Kleinrock, L. *Communication Nets: Stochastic Message Flow And Delay,* McGraw-Hill, New York, 1964 (also Dover, 1972).

[6] Kamoun, F. "Design Considerations for Large Computer Communication Networks," Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7642, April 1976.

[7] Frank, H., I.T. Frisch and W. Chou. "Topological Considerations in the Design of the ARPA Network," *AFIPS Conference Proceedings,* SJCC, Atlanta City, New Jersey, 1970, pp. 581–587.

[8] Cole, G.C. "Computer Network Measurements: Techniques and Experiments," School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG- 7165, 1971.

[9] Fultz, G.L. "Adaptive Routing Techniques for Message Switching Computer-Communication Networks," School of Engineering and Applied Science, University of California, Los Angles, UCLA-ENG-7252, July 1972.

[10] Little, J. "A Proof Of The Queueing Formula L = λW," *Operations Research,* Vol. 9, No. 2, March 1961, pp. 383–387.

[11] Jackson J.R. "Networks Of Waiting Lines," *Operations Research,* Vol. 5, 1957, pp. 518–521.

[12] Kleinrock, L. *Queueing Systems, Vol. 1: Theory,* Wiley-Interscience, New York, 1975.

[13] Danthine, A.A.S. and L. Echenauer. "Influence on The Node Behavior of The Node-to-Node Protocol," Inter-Network Working Group (INWG) Protocol Note No. 22, March 1975, pp. 87–92.

[14] Kermani, P. "Switching and Flow Control Techniques In Computer Communication Networks," Ph. D. Dissertation, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7802. (Also published as Ph. D. Dissertation, December 1977).

[15] Lam, S.S. "Store-and-forward Buffer Requirements in a Packet Switching Network," *IEEE Transactions on Communications,* Vol. Com-24, No. 4, April 1975, pp. 394–403.