An Evaluation of Local Path ID Swapping in Computer Networks

GEORGE MARKOWSKY AND FRANKLIN H. MOSS

Abstract—**This paper analyzes a method for identifying end-to-end connections in computer networks which is designed to provide reductions in the sizes of the packet headers and routing tables stored in the nodes. The method, known as Local Path ID Swapping, uses a shortened connection identifier, called the LPID, in the message headers and routing tables. In general, the LPID field is swapped in the message header from node to node along the path of the route. Some analytical results are presented for evaluating the important tradeoffs involved in LPID swapping. Most notable is the tradeoff between the size of the LPID field and the number of connections which can be defined in the network.**

I. INTRODUCTION

THE problem of routing in computer networks consists of steering messages from one end user participating in a session, through a series of links and nodes in the network, to the other participating end user. In the "next link" method considered here, each message traveling in the network contains information in a message header which, working in conjunction with routing tables stored at the nodes, steers the message from node to node until it reaches its destination node. The technique finally provides for delivery of the message to its destination end user.

Two major issues in routing are: 1) determination of the most appropriate message header/routing table mechanism for the network under consideration, and 2) definition of the paths to be used for end user communication. This paper is concerned primarily with issue 1) where the basic choice is to be made is between source-independent and source-dependent routing. In source-independent routing, the next link for a message arriving at a given node is determined by the destination node of that message and possibly a routing indicator usually small) such as described in [1]. In source-dependent routing, the next link for a message is a function of the origin and destination node of that message and possibly a routing indicator—a form of source-dependent routing is used in TYMNET [2].

In this paper we are concerned with *source-dependent routing,* whose primary advantage is to allow next links to be changed independently for each (origin, destination, routing indicator) triplet at each node. This triplet may be used to identify an individual *session* between two end users, in which case the routing indicator distinguishes among different sessions between the same origin/destination node pair. On the other hand, the triplet may be used to identify one of several *routes* between an origin/destination node pair, in which case there must be a separate field in the message header used to distinguish among different sessions using the same route (seen only at route ends). In general, we shall refer to the triplet as defining a single *connection* in the network, which may correspond to either a session or a route.

The advantage of independent changeability of next links which characterizes source-dependent routing allows flexible algorithms for the dynamic establishment/disestablishment of end-to-end connections. One such algorithm has been implemented in TYMNET [2], where connections are identified with individual sessions and centralized path definition is used. Another such algorithm has been described in [3], where connections are identified with routes and distributed path definition is used.

The primary disadvantage of source-dependent routing with respect to source-independent routing is that message header size and routing table size, both number of rows and size of each row, may be significantly increased. We first discuss the question of number of rows in the routing tables. Let N = number of network nodes and M = number of routing indicators. Assume for the moment that *all* connections are simultaneously defined in the network. For source-dependent routing, the number of rows in the routing table is $M\binom{N}{2}$, whereas for source-independent routing, the number is $M(N-1)$. The explosive growth of the number of rows with network size for source-dependent routing is abated by going from simultaneous definition of all connections to dynamic establishment/disestablishment of connections as required for communications as in [2] and [3]. It will be shown in Section III of this paper that in the dynamic environment, the number of rows in a source-dependent routing table does not present a practical problem.

We now turn to the question of size of the message header and size of each routing table row, which go hand in hand. The size of these factors depend on the type of message header/routing table mechanism implemented. One approach, which we refer to as "global path ID (GPID)," identifies each connection by its entire (origin, destination, routing indicator) name. If there are N nodes in the network, then the message header must contain $\log_2 N$ more bits for source-dependent than for source-independent routing. Each row of the routing tables is also increased by the same number of bits. In many cases, these are extremely critical factors in the design and operation of large networks. Since future networks may provide packetized voice service [4], it is extremely crucial to keep total message size small. Also, it is desirable to keep

Paper approved by the Editor for Computer Communication of the IEEE Communications Society for publication without oral presentation. Manuscript received April 2, 1980; revised September 16, 1980.

The authors are with the Department of Computer Sciences, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

routing table size small in order to reduce the on-line memory capacity dedicated to communications.

An implementation of source-dependent routing is analyzed in the paper which eases the problem of message header/routing table row size. Referred to here as "Local Path ID (LPID) Swapping," it is essentially equivalent to a method called "path number (PN)" described in [5]. Also, LPID swapping is similar to a technique called "logical record number" presently implemented in TYMNET [2].

LPID swapping will now be described. Briefly, it involves replacing the three fields for the GPID (origin, destination, routing indicator) in the message header by shortened identifier, the LPID, which is, in general, swapped from node to node along the path of a connection. LPID swapping is tailored to exploit the following observations.

1) In order to determine the proper next link for a message traveling on a connection, a node need not contain information relating *explicitly* to either the origin, the destination, or the routing indicator of the message.

2) At any given time, a node need only distinguish between connections which actually originate, terminate, or pass through it at that time, where by "distinguish" we mean associate the proper next link with the proper connection.

3) In larger networks, the number of connections which originate, terminate, or pass through a node at any time will, in practice, be a small percentage of the total set of connections which can be defined in the network. The reasoning is as follows. First, even if all possible $M\binom{N}{2}$ connections are defined at once, it will generally be true that a given node will lie on only a small number of such connections. The second justification to be given applies when connections are dynamically established/disestablished in response to end user requirements for communication as in [2] and [3]. Since only a small percentage of potential end users will be in communication at any time, this further reduces the number of connections which originate, terminate or pass through a node.

Observations 1)–3) above imply that connections may be identified by a "local path ID (LPID)," where under some circumstances the number of bits required to specify the LPID may be significantly less than the number required to specify the GPID. The principle behind LPID swapping is now introduced by way of a very simple example.

Example 1

Consider the situation depicted in Fig. 1, in which the connections $A.B.1$, $E.D.1$, $E.D.2$, and $C.F.3$ are the only connections passing through the node i at a given time. If GPID were used, then the routing table at node i would appear as in Table I. Each message header would carry the entire connection identifier (origin, destination, routing indicator) comprised by $\log(2N + M)$ bits.

The LPID swapping approach is now illustrated. The LPID's which are associated with these connections *locally at node i* are as indicated by the 0's and 1's in Fig. 1. Now, the inbound portion of a connection is distinghised by its *inbound LPID* and its *inbound link*. For example, both $A.B.1$ and $E.D.2$ arrive with LPID = 1, but they arrive on different links. The outbound portion of a connection is specified by *outbound LPID*

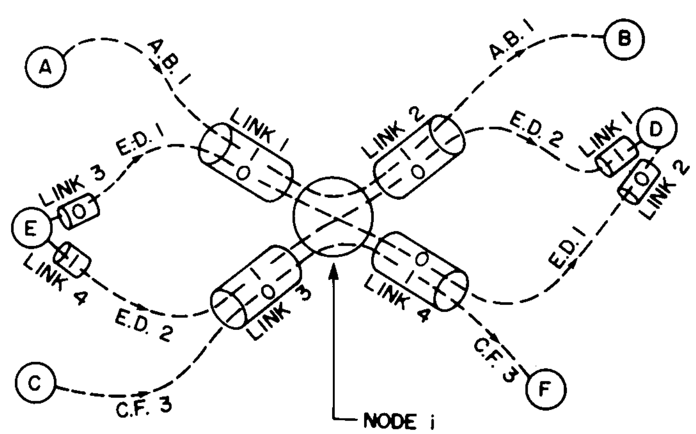


Fig. 1. Illustration of LPID Swapping.

TABLE I
GPID ROUTING TABLE AT NODE i

$A.B.1$	→	LINK 2
$E.D.2$	→	LINK 2
$E.D.1$	→	LINK 4
$C.F.3$	→	LINK 4

as well as *outbound link.* Note that the outbound LPID may be different from the inbound LPID, that is, *LPID swapping* may occur.

A conceptual form of the routing table which represents these four connections at node i is presented in Table II. At connection ends (e.g., nodes E and D), the routing tables have the form presented in Table III (for $E.D.1$ and $E.D.2$).

In this example, only one bit for LPID need be carried in a message header arriving at node i in order to distinguish between the four connections passing through it. This is independent of the total number of nodes in the network. Before the message is forwarded on the indicated outbound link, the LPID bit is swapped in some cases. □

In brief, every "LPID connection" in the network is identified by a sequence of pairs {(link, LPID)$_i$} where a given (link, LPID) pair appears in the sequence for at most one connection.

This paper presents an analysis which can be used to determine the savings which result by using LPID swapping in a network. Expressions are presented which allow the network designer to arrive at an appropriate number of bits for the LPID field in the message headers, and to estimate the amount of storage which should be allocated in the main memory of network nodes for routing tables. The expressions are useful at the early stages of networking planning and design.

The paper proceeds as follows. In Section II, pertinent questions regarding LPID swapping are presented. Section III presents three theorems which provide answers to the questions of Section II, where for simplicity it is assumed that all node pairs are equally likely to require connections between them, independent of distance or number of end users at each node. In Section IV, analytical and numerical examples are presented of particular sample networks to illustrate the theorems of Section III. An approach is presented in Section V to account for the fact that not all node pairs in a network are equally likely to require connections between them, thus providing a practical generalization of the results of Section III. Conclusions are presented in Section VI.

TABLE II
LPID ROUTING TABLE AT NODE i

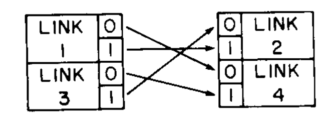
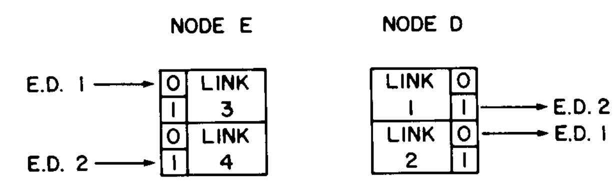


TABLE III
ROUTING TABLES AT E AND D



II. THE PROBLEM

One central issue in LPID swapping is to arrive at the appropriate number of bits (say α) to be used for the LPID field. It is desirable to keep α small because this provides the desired reduction in message header and routing table row size. On the other hand, α must be sufficiently large to accommodate the connectivity needs of the network end users a large percentage of the time.

There are many factors which enter into the determination of the appropriate value of α for a particular network. First, it must be decided whether the LPID connections are to be used to represent routes (which multiplex many sessions) or individual sessions. Second, it turns out that the appropriate value of α is a function of the network topology as well as the connectivity patterns required by the end users. Also, since α is a logical capacity which determines the number of connections which may pass through any node, it may be desirable to choose its value in recognition of the physical capacity of a node (e.g., buffer capacity, bandwidth, memory available for tables, etc.) to handle traffic on those connections.

This paper presents some analytical results intended to give a "seat of the pants" feel for the tradeoffs involved in LPID swapping. Among the questions confronted are the following.

• Given a value of α, what is the total simultaneous "connection capacity" of any node pair and of the network as a whole?

• Given the need for simultaneous connections between every node pair in a network, what savings are to be expected by using LPID swapping over GPID?

• Given the need for simultaneous connections between only some fraction η of node pairs (i.e., a dynamic connection establishment/disestablishment capability), what is the appropriate value of α?

III. BASIC RESULTS

This section presents three theorems which answer the questions about LPID swapping posed in the previous section. Conceptually, it is useful to think of LPID's as being resources available on links, where each link has a "connection capacity" of 2^α (recall that α is the number of bits available for specifying the LPID field in the message header). Also, in order to state theorems which are not explicitly dependent on particular network topologies and specific path definitions for connections, the notion of average minimum path (AMP) in a network is used freely. In some cases, theorems will be stated in terms of bounds rather than equalities. It is also assumed in this section that all node pairs are equally likely to require connections between them, independent of distance or number of users at each node.

First, some notation is presented. Throughout this paper, all logarithms are to the base 2 unless otherwise noted. Let N be the number of nodes in a network and let $\mathcal{N}$ denote the collection of nodes. Also, let L denote the number of links. Let p_{ij} denote a single path between node i and node j and let $\{p_{ij}\}$ be a collection of paths for all $i, j \in \mathcal{N}$. The number of links in p_{ij} is denoted by $d(p_{ij})$. Let p_{ij}^m denote a connection between i and j which minimizes $d(p_{ij})$. Next, the average minimum path of the network topology T is defined as follows:

$$\text{AMP}(T) \triangleq \frac{\sum_{i,j \in \mathcal{N}} d(p_{ij}{}^m)}{\binom{N}{2}} \tag{1}$$

For simplicity, it assumed that every LPID path between i and j traverses p_{ij}^m. From the practical path definition point of view, this is a reasonable assumption to make in the absence of any detailed knowledge regarding network resources and traffic requirements.

We now define two figures of merit for evaluating LPID swapping as a function of α and T (the network topology):

$c(\alpha, T)$ ≜ the number of different LPID connections which may be specified per node pair, i.e., the "LPID connection capacity" per node pair;

$C(\alpha, T)$ ≜ the total number of different LPID connections which may be specified per network, i.e., the "LPID connection capacity" of a network.

Note that $c(\alpha, T)$ is equivalent to the number of routing indicators M which may be used to identify paths.

Theorem 1: For a given value of α and a network topology T,

$$c(\alpha, T) \leqslant \frac{2^\alpha \times L}{\binom{N}{2} \times \text{AMP}(T)} \tag{2}$$

$$C(\alpha, T) \leqslant \frac{2^\alpha \times L}{\text{AMP}(T)} \tag{3}$$

Proof: The number of LPID's used in the tables throughout the network by any set $\{p_{ij}\}$ is $\sum_{ij} d(p_{ij})$. The total number of LPID's available throughout the network for connection specification is $2^\alpha \times L$ since there are 2^α per link. Therefore, the total number of different LPID connections which may be specified per node pair is

$$c(\alpha, T) \leqslant \frac{2^\alpha \times L}{\sum_{i,j,\in N} d(p_{ij})}. \tag{4}$$

But since $d(p_{ij}) \geqslant d(p_{ij}^m)$,

$$\frac{\sum_{i,j,\in N} d(p_{ij})}{\binom{N}{2}} \geqslant \frac{\sum_{i,j\in N} d(p_{ij}{}^m)}{\binom{N}{2}} = \text{AMP}(T). \tag{5}$$

Inequality (2) follows directly from (4) and (5).

Finally, inequality (3) follows directly from definition of $C(\alpha, T)$. □

The following observations are made with regard to Theorem 1.

1) For a fixed topology, the upper bound on connection capacity per node pair is a linear function of the path capacity of each link. Specifically,

$$c(\alpha, T) \leqslant f(T) \times 2^\alpha \tag{6}$$

where

$$f(T) = \frac{L}{\binom{N}{2}\text{AMP}(T)}.$$

2) Writing (2) in terms of average degree $\bar{d}$, where $\bar{d} = 2L/N$, we obtain

$$c(\alpha, T) \leqslant \left(\frac{2^\alpha \times \bar{d}}{\text{AMP}(T)}\right) \bigg/ (N-1). \tag{7}$$

It has been empirically observed (see Section IV-D) that for computer networks, the value of $\bar{d}$ is relatively insensitive to the number of nodes. Typically, $\bar{d} = 2.5$. Therefore, for an appropriate class of networks with fixed value of AMP and a fixed value of α, the upper bound on connection capacity per node pair drops off as the inverse of the number of nodes,

$$c(\alpha, T) \leqslant \frac{\frac{2.5}{\text{AMP}(T)} 2^\alpha}{(N-1)}. \tag{8}$$

3) Theorem 1 may be modified to account for more realistic communication patterns in a network by replacing average minimum path by weighted minimum path. This is discussed in Section V.

The next theorem tells us what is the best one can expect to save (in terms of number of bits in the header required for routing information) under the following conditions:

1) LPID swapping is used instead of GPID,

2) it is desired to specify one or more connections simultaneously between every node pair in the network, and

3) the number of nodes tends to infinity.

First, we define

$n \triangleq$ number of bits required to specify c connections between every node pair in a network when LPID swapping is used

$m \triangleq$ number of bits required to specify c connections between every node pair in a network when GPID is used.

Theorem 2: For a network where the degrees are bounded by d, $2 \leqslant d < \infty$,

$$\lim_{N\to\infty} \frac{n}{m} \geqslant 0.5. \tag{9}$$

Proof: From expression (2) of Theorem 1,

$$n \geqslant \log c + \log \binom{N}{2} + \log(\text{AMP}) - \log L. \tag{10}$$

Since no more than d^k nodes can have distance exactly k from a given node, it can be readily seen that AMP is bounded below by $\lambda \log_d N$ ($\lambda > 0$). Therefore,

$$\log(\text{AMP}) \geqslant \log \lambda + \log\log_d N \quad \text{for some } \lambda \tag{11}$$

and thus

$$\log(\text{AMP}) \geqslant \log\lambda + \log\log N - \log\log d. \tag{12}$$

Also, $L \leqslant (d/2)N$ so that

$$-\log L \geqslant -\log d - \log N + 1. \tag{13}$$

Substituting (12) and (13) into (10) and simplifying, we obtain

$$n \geqslant \log\left(\frac{c\lambda}{d}\right) + \log(N-1) + \log\log N - \log\log d. \tag{14}$$

Now

$$m = \log\left[c\binom{N}{2}\right]$$

$$= \log c + \log N + \log(N-1) - 1. \tag{15}$$

Therefore, (14) and (15) imply

$$\frac{n}{m} \geqslant \frac{\log\left(\frac{c\lambda}{d}\right) + \log(N-1) + \log\log N - \log\log d}{\log c + \log N + \log(N-1)}, \tag{16}$$

and taking the limit as $N \to \infty$, we obtain

$$\lim_{N\to\infty} \frac{n}{m} \geqslant 0.5. \qquad \square \tag{17}$$

In brief, Theorem 2 tells us that if it is desired to specify one or more connections simultaneously between every node pair in large networks, then the relative savings realized by LPID Swapping is not dramatic (half at best). However, for large networks, the absolute savings (in terms of number of bits) could be significant.

As pointed out in Section I, LPID swapping is tailored to produce substantial savings in a dynamic connection establishment/disestablishment environment in which at any time only a fraction of the total number of possible connections are defined. To get some handle on this, we approach the problem from a different point of view. Let us consider homogeneous networks (all nodes of equal degree) of N nodes. We now define

$\eta \triangleq$ expected proportion of node pairs which simultaneously require at least one connection;

$\mu \triangleq$ expected proportion of routing indicators (M) simultaneously used by each node pair which require at least one connection;

$\rho \triangleq$ the expected number of connections on which a node lies at a given time.

Theorem 3:

$$\rho = \frac{\eta\mu M(N-1)(\text{AMP}(T)+1)}{2}. \tag{18}$$

Proof: Assume, as before, that all connections traverse the minimum hop path. Then the "average connection" in the network contains AMP(T) + 1 nodes. Therefore, the expectation that a given node will be on a given connection is $(\text{AMP}(T)+1)/N$.

The total number of connections defined is $\eta\binom{N}{2}\mu M$. Therefore, the expected number of connections on which a node will lie is

$$\rho = \eta\mu\binom{N}{2}M\frac{\text{AMP}(T)+1}{N}$$

$$= \frac{\eta\mu M(N-1)(\text{AMP}(T)+1)}{2} \qquad \square$$

The following observations are made with regard to Theorem 3.

1) The number of bits required for the LPID field to specify the set of connections implied by the parameters η, μ, N, M is simply $\lceil(\log\rho)\rceil$, i.e.,

$$\alpha = \left\lceil\left(\log\left(\frac{\eta\mu M(N-1)(\text{AMP}(T)+1)}{2}\right)\right)\right\rceil. \tag{19}$$

2) Consider the following example:

$\eta = 10^{-1}$

$\mu = 10^{-2}$

$M = 256$

$N = 256.$

Assume that AMP(T) = 3.5. Then using (19), $\rho = 146.88$ and α turns out to be 8 bits. If GPID is used, then the number of bits required is log

$$\left\lceil 256\binom{256}{2}\right\rceil = 23.$$

In this case, the savings ratio is about one to three.

3) Using Theorem 3, we can obtain an idea of the expected number of rows in the routing table at each node when connections are dynamically established/disestablished. This is simply equal to ρ, which for the example of 2) above is 147. We can now define an estimate of total expected routing table size at each node for that example. Each row contains the entries (link, LPID)$_{\text{in}}$ and (link, LPID)$_{\text{out}}$. If we assume 4 bits for the link field, each row contains three (8 bit) bytes. Therefore, total table size is 441 bytes. In practical terms, this should not be a problem even for small communication nodes.

IV. EXAMPLES

We will first present some sample networks which can be handled analytically and then we will present a numerical treatment of some existing networks. The emphasis will be on the determination of AMP(T) and $C(\alpha, T)$ for various network topologies. We shall also breifly discuss how the figure of merit $C(\alpha, T)$ may enter into the question of network topological design.

A. String Networks

The first case we will discuss is that in which the nodes are connected together like beads on a string (see Fig. 2). It is easy to see that the average degree is $2 - 2/N$. Furthermore, we have

$$\text{AMP}(T) = \left(1\bigg/\binom{N}{2}\right)\sum_{j=1}^{N-1}\sum_{t=1}^{N-j} t$$

$$= \left(1\bigg/\binom{N}{2}\right)\sum_{j=1}^{N-1}(N^2 - (2j-1)N$$

$$+ j^2 - j)/2 \tag{20}$$

$$= (N+1)/3.$$

Thus,

$$C(\alpha, T) = 3 \times ((N-1)/(N+1)) \times 2^\alpha. \tag{21}$$

This figure for the network connection capacity might seem low, but a little reflection shows that it is a reasonable one since the expected distance between two nodes which might


Fig. 2. An N-node string network.

want to communicate is roughly $N/3$. If we knew that only neighbors were interested in communicating to one another, then $C(\alpha, T)$ would be substantially increased. This issue will be discussed more thoroughly in the next section.

B. Loop Networks

Suppose that our network T is in the form of a loop (see Fig. 3).

We have two cases to consider: 1) N even; 2) N odd. In the first case,

$$\text{AMP}(T) = N\left(2\sum_{j=1}^{N/2} j - N/2\right)\bigg/\left(2\binom{N}{2}\right) = \frac{N^2}{4(N-1)}. \tag{22}$$

The formula is derived as follows. The numerator consists of two factors, the second of which sums the distances of nodes from a given node, i.e., there are two nodes at distance 1, 2, etc., but only one node at distance $N/2$. The first factor in the numerator represents the fact that we construct the sum for each node. The 2 in the denominator is necessary because the distance between each pair of nodes is summed twice. Finally, the $(N/2)$ is just the averaging factor.

In N is odd, we get,

$$\text{AMP}(T) = N\left(2\sum_{j=1}^{(N-1)/2} j\right)\bigg/\left(2\binom{N}{2}\right) = (N+1)/4. \tag{23}$$

In both cases, we see that AMP(T) is roughly $N/4$. Thus, $C(\alpha, T)$ is roughly $4 \times 2^\alpha$ or $2^{\alpha+2}$. □

C. Star Networks

Suppose that T is in the form of a star (see Fig. 4). Here we have

$$\text{AMP}(T) = \left((N-1) + \left(\binom{N}{2} - (N-1)\right)2\right)\bigg/\binom{N}{2} = 2 - 2/N. \tag{24}$$

Thus,

$$C(\alpha, T) = (N/2)2^\alpha = N2^{\alpha-1}. \quad \square(25)$$

Note how much higher the capacity of the Star Networks are than the capacities of the String or Loop Networks for a given value of N. Notice that the three classes of networks have essentially the same number of links for the same number of nodes, so that AMP and hence $C(\alpha, T)$ actually capture some of the flavor of the topology of a network. These examples quantify to some extent the general principle that the closer points are in general, the fewer are the resources necessary to support a given amount of communication. The advantage of using $C(\alpha, T)$ is that we have a quantitative measure for this

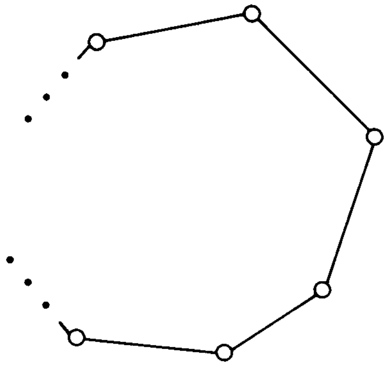


Fig. 3. An N-node loop network.

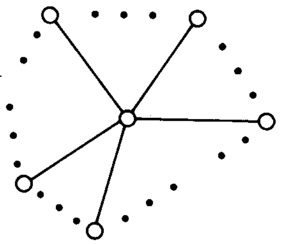


Fig. 4. An N-node star network.

general notion. We will now turn our attention to several specific networks in order to illustrate the application of $C(\alpha, T)$ and to provide some values of key network parameters to support some of the claims made earlier.

D. Real Networks

We now present numerical analyses of ARPANET, two early proposed versions of TELENET and TYMNET (see Figs. 5, 6, and 7).

ARPANET has 57 nodes and 71 edges and thus an average degree of almost exactly 2.5 (actually 2.49+). Furthermore, AMP (ARPANET) $\cong$ 5.24 and thus $C(\alpha, \text{ARPANET}) \cong 13.55 \times 2^\alpha$.

The proposed version of TELENET without the satellite links has 18 nodes, 22 edges, and an average degree of about 2.44. AMP in this case is about 3.12, which yields a capacity of roughly $7.05 \times 2^\alpha$. Adding the satellite links gives us 19 nodes, 26 edges, an average degree of about 2.74, and an AMP of about 2.82. This yields a capacity of roughly $9.22 \times 2^\alpha$. Thus, adding the satellite links increases capacity by about 30 percent.

TYMNET has 42 nodes, 48 edges, an average degree of about 2.30, and an AMP of about 3.16. This yields a capacity of roughly $15.19 \times 2^\alpha$.

We now make an interesting observations on the figures presented above. It was noted in the proof of Theorem 2 that with the degrees of all nodes bounded by d, AMP is bounded by a multiple of $\log_d N$.

With this in mind, it is interesting to compare AMP and $\log_{\bar{d}} N$ for the networks above ($\bar{d}$ is the average degree). These values appear in Table IV.

Note that in these cases $\log_{\bar{d}} N$ gives a good approximation of AMP. In the cases of TELENET, the fit is excellent. The $\log_{\bar{d}} N$ value basically arises if we think of the network resem-

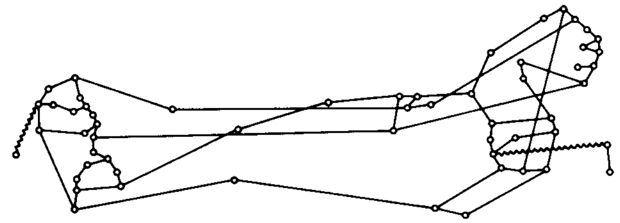


Fig. 5. ARPANET.

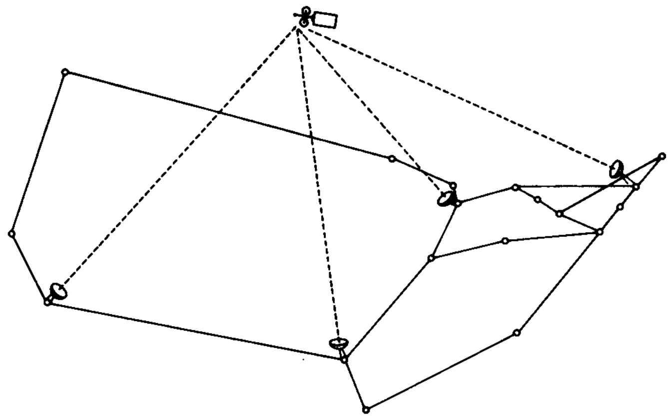


Fig. 6. Two early versions of TELENET, 1977.

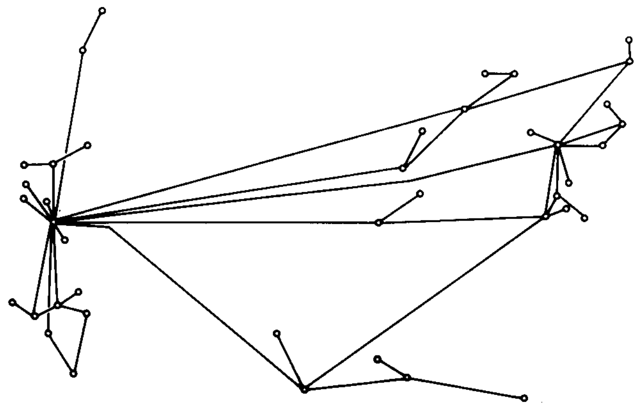


Fig. 7. TYMNET, 1972.

TABLE IV

Network	AMP	$\log_{\bar{d}} N$
ARPANET	5.24	4.43
TELENET (no satellite)	3.12	3.24
TELENET (satellite)	2.82	2.92
TYMNET	3.16	4.50

bling a $\bar{d}$-ary tree from each of its nodes. If AMP exceeds the log, it indicates that the network has many loops and strings on it (as in ARPANET). On the other hand, if AMP is less than the log, it indicates that the network contains many stars or clusters (as in TYMNET). Thus, again, we see that AMP can be used to quantify some interesting topological information. Since $\bar{d}$ seems roughly to be 2.5 for real networks, AMP is roughly $\log_{2.5} N \cong 0.76 \log N$.

V. WEIGHTED MEAN PATH AND PORT CAPACITY

In working with AMP, we have assumed that any two nodes are just as likely to communicate with one another as any other pair. In this section, we discuss a model which allows us to consider the effects of a nonuniform distribution of communication. The basic concept here is that of *weighted minimal path* (WMP), which means that we average together the lengths of the minimal paths according to their frequency of use. More formally, given a probability distribution $\mathcal{P}$ over pairs of nodes in a network T, i.e., $\mathcal{P}$ is a nonnegative, real-valued function with $\sum_{i,j,\epsilon N} \mathcal{P}(i,j) = 1$ and $\mathcal{P}(i,i) = 0$ for all i, we define

$$\text{WMP}(T, \mathcal{P}) = \sum_{i,j \in N} \mathcal{P}(i,j) d(p_{ij}{}^m). \tag{26}$$

We get AMP from WMP by letting $\mathcal{P}$ be the uniform distribution, i.e., $\mathcal{P}(i,j) = 2/N(N-1)$ for all $i \neq j$. Clearly, the value of WMP depends greatly on being able to get good values for $\mathcal{P}$.

In order to get an estimate for $\mathcal{P}$, we focus our attention on the number of end user "ports" available at a given node for communication to other nodes. It is assumed that every connection between nodes involves exactly one such port at each node. If we do not know of any reason why a given node is any more or less likely to communicate with any other given node, it seems reasonable to proceed as follows. If node i can support twice as many end users ports as node j, it seems reasonable to assume that it is twice as likely to require a connection to some node $k (k \neq i, j)$ as node j is. This suggests the following approach. First, let π_i denote the number of such ports available at node i. Then the "port-derived connection distribution" is

$$\mathcal{P}(i,j) \triangleq \frac{\pi_i \pi_j}{\pi_k \pi_l} \mathcal{P}(k,l) \qquad (i \neq j, \quad k \neq l). \tag{27}$$

If we further require that $\Sigma \mathcal{P}(i,j) = 1$, we get

$$\mathcal{P}(i,j) = \frac{\pi_i \pi_j}{\left(\sum_{i \neq j} \pi_i\right)^2 - \sum_{i \neq j} \pi_i^2} \qquad (i \neq j).$$

The reader will note that if $\pi_i = \pi_j$ for all i, j, $\mathcal{P}(i,j) = 2/N(N-1)$ which is just the uniform distribution. The port-derived connection distribution is the most sensible one to use in the absence of any more specific information about the distribution of communication within a network.

For completeness, we conclude this section by showing how the distribution of ports in a network effects the number of possible connections in the absence of any other constraints. In this case, Theorem 4 shows how much simpler the problem of determining connection capacity based on port distribution alone is than that of determining connection capacity based on LPID constraints.

Theorem 4: Let T be a network with at least two nodes such that node $i \epsilon N$ has π_i ports. Assume that there are no LPID constraints on the number C of simultaneous connections possible in the network. Then the following are true.

1) If there exists $i_0 \epsilon N$ such that

$$\pi_{i_0} > \sum_{i \neq i_0} \pi_i,$$

then

$$C = \sum_{i \neq i_0} \pi_i.$$

2) If for all

$$j \in N, \ \pi_j \leqslant \sum_{i \neq j} \pi_i,$$

then

$$C = \left\lfloor \sum \pi_i / 2 \right\rfloor.$$

Proof:

a) Clearly,

$$C \geqslant \sum_{i \neq i_0} \pi_i$$

since we can simply connect each port in each $i \neq i_0$ to a port in i_0. If C were greater than

$$\sum_{i \neq i_0} \pi_i,$$

we would have a connection beginning and ending at i_0 which is impossible.

b) Clearly, $C \leqslant \lfloor \Sigma \pi_i / 2 \rfloor$ since each connection uses up exactly two ports. If $C < \lfloor \Sigma \pi_i / 2 \rfloor$, choose a largest possible set of simultaneous connections and observe that either there exist nodes i_1 and i_2, each of which has an unused port, or there exists a node i_3 with two unused ports. In the first case, it is clear how to get an additional connection which contradicts the assumption that we had constructed the largest possible set of connections. If the first case does not hold, then all ports of $j \neq i_3$ are used up. Either we can find distinct nodes $i_4, i_5 \neq i_3$ having a connection between them or all connections are between i_3 and some other node. This last case cannot happen since it would imply that

$$\pi_{i_3} > \pi_{i_3} - 2 \geqslant \sum_{i \neq i_3} \pi_i.$$

Thus, we have a connection between i_4 and i_5. Note that we can replace that connection by two others: one between i_4 and i_3 and one between i_5 and i_3. This contradicts the maximality of C. □

The reader will note the simple state of affairs when we have only port constraints. If we call C the *port-constrained connection capacity* of a network, Theorem 4 enables us to calculate it quite easily.

VI. CONCLUSIONS

In this paper we have presented an evaluation of the LPID Swapping method for connection (route or session) identification in computer networks. It has been shown that this method can lead to substantial reductions in the sizes of message headers and routing tables, particularly in networks in which connections are dynamically established/disestablished as required for end user communication.

The notion of average minimum path (AMP) has proved to be very useful in providing simple expressions which are independent of particular network topology. Also, it has been shown that for a representative set of computer networks, AMP can be approximated quite closely by a simple function of average degree and number of nodes only. Therefore, the analysis presented is particularly appropriate for application to the initial phases of network planning and design.

REFERENCES

[1] V. Ahuja, "Routing and flow control in systems network architecture," *IBM Syst. J.*, vol. 18, no. 2, pp. 298–314, 1979.

[2] J. Rinde, "TYMNET I: An alternative to packet technology," in *Proc. 3rd ICCC*, Toronto, Ont., Canada, Aug. 1976, pp. 268–273.

[3] F. H. Moss and J. Jaffe, "Distributed dynamic explicit route generation for computer networks," IBM Res. Rep., in preparation.

[4] H. Frank and I. Gitman, "Study shows packet switching best for voice traffic, too," *Data Commun.*, pp. 43–62, Mar. 1979.

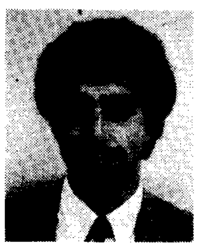
[5] M. Schwartz and T. E. Stern, "Routing techniques used in computer communication networks," *IEEE Trans. Commun.*, vol. COM-28, pp. 539–552, Apr. 1980.

★

George Markowsky received the B.A. degree in mathematics from Columbia University, New York, NY, and the M.A. and Ph.D. degrees in mathematics from Harvard University, Cambridge, MA.

He joined IBM in 1974 and has been at the Watson Research Center since that time. At present he is Technical Assistant to the Director of the Computer Sciences Department and Manager of Special Projects, a group which combines consulting work together with small-scale explorations of promising new directions for research. He has worked and is working in a number of areas including circuit design and testing, three-dimensional modeling, networking, algorithm design, applications of computer science to biology, coding theory, discrete mathematics, and theoretical computer sciences.

★



Franklin H. Moss received the B.S.E. degree in aerospace and mechanical sciences from Princeton University, Princeton, NJ, in 1971, and the S.M. and Ph.D. degree in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge, in 1972 and 1977, respectively.

He is presently Manager of the Network Architecture and Protocols Group at the IBM T. J. Watson Research Center, Yorktown Heights, NY. This group develops advanced network control features for IBM's Systems Network Architecture (SNA). He joined IBM in 1977 in a postdoctoral position at the Israel Scientific Center, where he served as Leader of a project investigating on-line computer control of greenhouse environment. His current research interests encompass various areas of data communications networks including distributed dynamic routing, congestion control, very large networks, and network interconnection.