AT&T Technical Journal
Vol. 64, No. 7, September 1985
Printed in U.S.A.

Regular Mesh Topologies in Local and Metropolitan Area Networks

By N. F. MAXEMCHUK*

(Manuscript received January 14, 1985)

The throughput per user in loop and bus configured local area networks decreases linearly with the number of users. These networks cannot be extended to a metropolitan area with many users. A class of mesh networks is described that increases the throughput of conventional local area networks by decreasing the fraction of the network capacity needed to transmit information between a source and a destination. These networks have multiple paths that increase the reliability of the networks, and have point-to-point links that can cover a metropolitan area. In general, mesh networks require complex store-and-forward nodes that also route messages, control the flow of data entering the network, resequence packets at the destination, and recover packets with errors. However, there are characteristics of the local or metropolitan area that allow these functions to be simplified. As a result of these simplifications, loop access protocols are extended to mesh networks and the need to store and forward data is eliminated. A file transfer protocol that does not require packet resequencing is described. Three mesh networks are studied, and the desirable characteristics of networks are determined. One network, the Manhattan street network, has many of the desirable characteristics.

I. INTRODUCTION

Loop topologies[1] and random access strategies[2] were first applied to local data networks in the late 1960's. In that era,

- Low-bit-rate terminals were connected to large central computers,
- Computers and terminals were shared by a few computer experts,

* AT&T Bell Laboratories.

- Large-scale integrated circuits did not exist, and
- High-bit-rate transmission facilities were not readily available across public right of ways.

As a result, these networks trade reliability, total throughput, and the distance the network can span[3] for simple access and transmission strategies. Today, for comparison,

- Simple terminals are evolving into personal computers with bit mapped, rather than character, displays,
- Computer usage is becoming universal,
- Very-Large-Scale Integration (VLSI) is becoming commonplace, and
- The increased deployment of optical fibers and CATV systems makes it possible to obtain high-bit-rate communications over wider areas.

Personal computers use larger bandwidths than simple terminals to communicate with centralized support facilities and distribute processing. The increasing use of these devices and the increased distances that high-bit-rate networks can span increase the throughput required of the interconnecting network. In loop and bus systems the total throughput is constant. The average capacity available to each user decreases linearly as the number of users increases. Therefore, to support more users with greater individual requirements, alternative topologies must be considered. The complexity of the devices being connected to networks and advances in VLSI make more complex network interfaces feasible. This increases the class of networks and access strategies that can be considered.

A large number of users, dispersed over a large area, can be accommodated by interconnecting conventional local area networks with gateways. Schlatter and Massey have analyzed this type of network.[4] Their system consists of loops interconnected by switching elements, as proposed by Pierce.[5] Messages use a smaller fraction of the total network capacity than they would if the system were a single loop. Therefore, the maximum throughput of the system increases. Users who communicate the most often are placed on the same loop, which minimizes the interference between subgroups of users. The main disadvantage with this approach is that the gateways are different from the access units and are complex store-and-forward elements.

Yemini[6] and Saadawi and Schwartz[7] are investigating a tree topology. In this network, users are at the leaves of the tree and the nodes of the tree are switching points. Depending on the location of the destination, the switches direct messages toward the root of the tree or toward the leaves. In Yemini's system, the switches establish separate broadcast networks, and in Saadawi's, the switches store

packets until the desired path is available. In these systems, messages only use a portion of the network capacity. Locating users who communicate frequently—who are near one another in the tree hierarchy—minimizes the interference between subgroups of users. The advantage of this approach over gateways is that there is only one type of element in the network. The disadvantage is that the network either stores and forwards packets or retains the distance constraints of broadcast networks.

To a certain extent these alternatives remove the throughput constraints of loop and bus systems. However, they still have single points of failure. To make networks more reliable, there must be multiple paths between each source and destination. By adding paths appropriately, the average and maximum distance between nodes decreases, messages use a smaller fraction of the network bandwidth, and the throughput increases. Multiple paths also make it possible to avoid heavily used segments of the network to equalize the load. Mesh networks, like loop networks, have point-to-point communication channels between nodes. This results in less expensive line drivers and receivers than multidrop broadcast systems and is compatible with current optical fiber transmission capabilities.

In general, mesh-configured networks and some of the local network alternatives require complex store-and-forward nodes. A queue of messages is maintained because packets arriving on several of the incoming links may be destined for the same outgoing link. In addition, store-and-forward networks must do routing, flow control, packet resequencing, and error control. Long-distance networks, such as the ARPA network,[8] perform these functions, but their interfaces are more complex than personal computers. Therefore, these networks are not a reasonable interconnection alternative for personal computers. There are, however, characteristics of the local or metropolitan area environment that make simpler mesh networks possible.

Local or metropolitan area networks differ from general long-distance networks in that the

- Physical location of the nodes does not dictate the topology of the network to as great an extent,
- Error rates are much lower, and
- Communications lines are less expensive.

In the local environment, it is not always necessary to connect the closest nodes together. Occasionally, connecting nodes that are further apart can make the topology of the network regular, and simplify tasks such as routing. The lower error rates make it more likely that a packet will traverse the entire network without error. Therefore, error control protocols can operate on an end-to-end, rather than on a link-by-link basis, and networks can have unidirectional links. When

messages are not stored at the intermediate nodes for possible retransmission, the class of possible access and transmission strategies increases. In general, there is also a trade-off between the system complexity and communications efficiency. The cost of communications lines, and the additional access strategies and network options that are possible, result in a different network solution in local networks than in long-distance networks. The result is that local networks are significantly less complex.

There is a description in Section III of several regular networks with simple routing strategies. End-to-end error control protocols make it possible to extend the slotted system and register-insertion techniques developed for loop systems to mesh networks. This is shown in Section IV. With these techniques, flow control on mesh networks can be done by throttling the sources, as on loop networks. A trade-off exists between the buffering in these systems and the efficiency with which the communication lines are used. One attempt to take advantage of this trade-off is the Floodnet system.[9] There is a description in Section IV of how this trade-off is applied to mesh networks with slotted system and register-insertion interfaces. The result is that for certain topologies mesh networks without buffering are reasonable. Finally, there is a description in Section V of several file-transfer protocols that do not resequence packets.

II. EXAMPLE

Before discussing the implementation of mesh networks, we show that these networks provide a potential to increase the throughput of conventional local area networks. In this example, two-connected networks, with as few as 64 nodes, increase the throughput of bus configured networks by a factor of 20 to 30. This comparison assumes that the same rate communication lines are used in both the mesh and random access networks. A factor of two increase in throughput is obtained because there is twice as much capacity emanating from each node. However, the major portion of the increase occurs because messages in the mesh network use only a fraction of the total network capacity. Greater increases are obtained in larger networks.

Two traffic distributions are considered, a uniform distribution and a skewed distribution. In the uniform case, each node sends an equal amount of data to each of the other nodes. The skewed distribution corresponds to what might occur in a network of personal computers and file servers. The network is divided into communities of interest, each consisting of a file server and seven personal computers. A personal computer directs 80 percent of its traffic to its own file server and 20 percent to the other file servers. The computer receives an equal amount of traffic from the file servers.

For each traffic distribution, the throughput for six network topologies is investigated. The first two networks are the conventional broadcast bus and loop configured networks. The throughput of the bus network is calculated assuming that the link utilization can approach one, and is an upper bound on the achievable throughput. In the loop network, the packets only use the links between the source and destination. In this network, and in the remaining networks, the throughput is determined by increasing the traffic levels from the sources until the utilization on any link equals one. The remaining four networks are two-connected networks with two links arriving at, and two links emanating from, each node. The first of these networks is a conventional bidirectional loop. For the skewed distribution, the file server is in the middle of the seven personal computers it is servicing. The next two networks are regular arrays called the modified shuffle exchange and the Manhattan street network. These networks are described in Section III. The Manhattan street networks with 16, 32, 48, and 64 nodes are 4×4, 8×4, 8×6, and 8×8 arrays, respectively. For the skewed distribution, the seven personal computers and the file server in a community of interest are arranged in a 4×2 array on the network. This is shown for the 16-node network

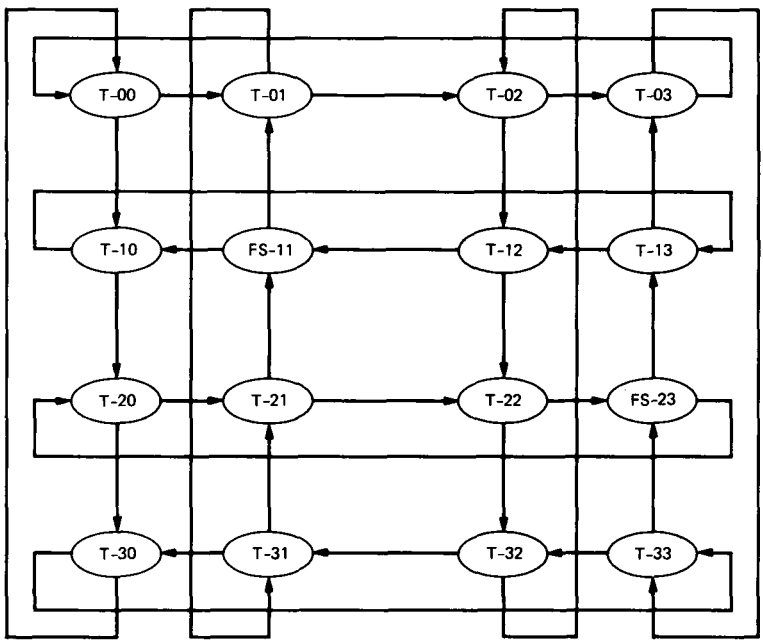

Fig. 1—A 16-node Manhattan street network with two communities of seven personal computers (T) and a file server (FS).

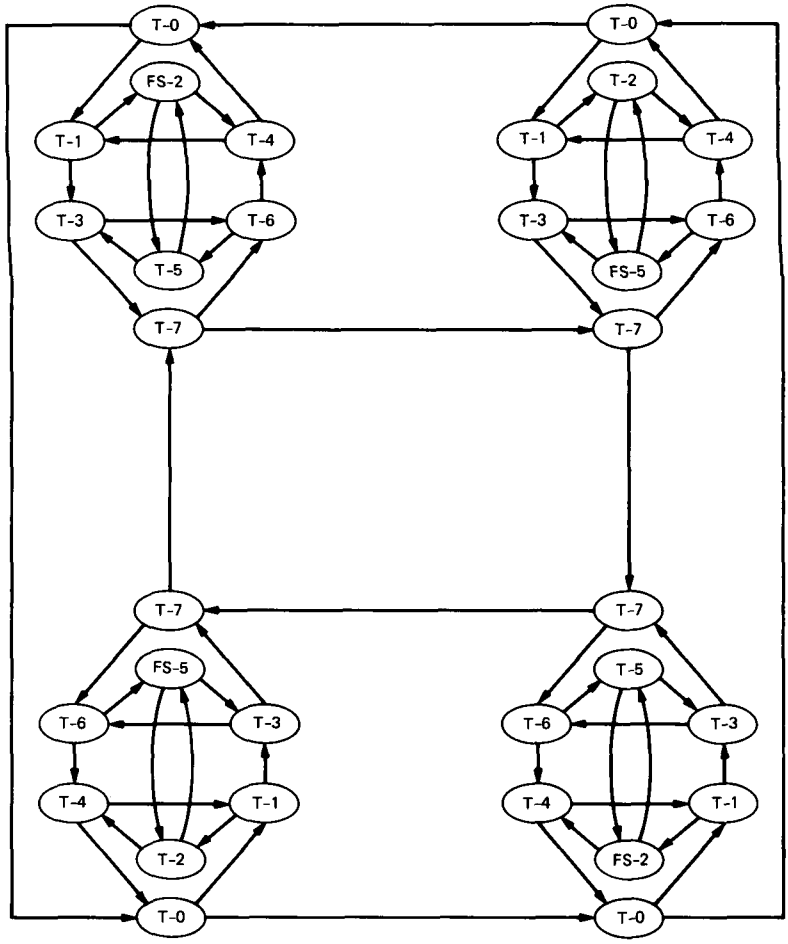

Fig. 2—A 32-node hierarchical network with four communities of seven personal computers (T) and a file server (FS) interconnected by shuffle-exchange networks. The four shuffle-exchange networks are connected by a bidirectional loop.

in Fig. 1. The final network is a hierarchical shuffle exchange, consisting of shuffle-exchange networks with the eight devices in a community of interest, interconnected by a bidirectional loop. Figure 2 shows a 32-node, hierarchical network consisting of four 8-node shuffle-exchange networks.

Traffic on the two-connected networks is placed on the shortest path between the source and destination. If there are several paths of equal length between a source and a destination, the path with the smallest flow is selected. Traffic with the shortest distance between a source and destination is assigned to the network first. Once a source destination requirement is assigned a path, the path is not changed if

Table I—The average megabits per user in networks with 10-Mb/s channels and the improvement over a broadcast network

	16 Nodes		32 Nodes		48 Nodes		64 Nodes	
Network	Mb/Usr	Imprv.	Mb/Usr	Imprv.	Mb/Usr	Imprv.	Mb/Usr	Imprv.
			Uniform Requirements					
BDCST	0.63	—	0.31	—	0.21	—	0.16	—
Loop	1.25	2.00	0.63	2.00	0.42	2.00	0.31	2.00
BDL	4.69	7.50	2.42	7.75	1.63	7.83	1.23	7.87
S-X	5.77	9.23	4.03	12.88			3.09	19.76
MSN	6.52	10.43	4.56	14.59	4.31	20.70	3.94	25.20
HS-X	4.69	7.50	1.96	6.28	1.41	6.75	1.13	7.20
			Skewed Requirements					
BDCST	0.36	—	0.18	—	0.12	—	0.09	—
Loop	0.71	2.00	0.36	2.00	0.24	2.00	0.18	2.00
BDL	2.63	7.37	2.08	11.67	1.80	15.11	1.59	17.82
S-X	1.61	4.52	1.01	5.68			0.81	9.10
MSN	2.63	7.37	2.17	12.17	2.12	17.80	2.12	23.76
HS-X	2.78	7.78	2.78	15.56	2.66	22.34	2.63	29.47

BDCST = Broadcast, BDL = Bidirectional, MSN = Manhattan street exchange, HS-X = Hierarchical street exchange, S-X = street exchange.

a link on the path becomes saturated, and the requirements are not split if two equally good paths exist. This procedure does not lead to the optimum throughput, but gives a reasonably good idea of what can be achieved.

The results of this investigation are presented in Table I. For each network, the average bit rate a user obtains in a network with 10-megabit-per-second transmission links, and the improvement this represents over a broadcast network, is presented. For the conventional broadcast and loop network, the fraction of the capacity a user obtains decreases linearly with the number of users, as expected. The loop system provides about twice as much throughput per user as the broadcast network because, on the average, a packet transmitted on this network uses only half of the network capacity. The two-connected networks obtain a factor of two increase in throughput because there is twice as much capacity emanating from each node, and an additional increase because the networks use a smaller fraction of the network capacity to transfer a packet between the sources and destinations.

The bidirectional loop, the Manhattan street network, and the hierarchical shuffle exchange respond well to the skewed requirements. These networks are capable of allowing complete connectivity while preventing users in different communities of interest from interfering with one another. This characteristic is extremely important in designing large networks. The shuffle-exchange and Manhattan street networks also respond well to a large group of users with uniform transmission requirements. This occurs because the average distance between users does not increase as rapidly in these networks as in the

other networks. The average distance between users in the Manhattan street network is greater than that in the shuffle-exchange network; however, the throughput of the Manhattan street network is greater. The Manhattan street network can support a larger throughput because there are more equal-length shortest paths, and bottlenecks can be avoided.

III. TOPOLOGY

In this section, three two-connected networks are described, the bidirectional loop, the modified shuffle exchange, and the Manhattan street network. These networks have two independent paths between any node, and they can survive a single loop or node failure. While these networks are not optimal, they show what measures can be used to compare topologies, and what network characteristics are desirable. Lower bounds on two measures, the average and maximum shortest path between nodes, are derived and compared with these three topologies.

3.1 Bidirectional loops

In a bidirectional loop with N nodes, labeled 0 to $N - 1$, node i is connected to nodes $(i - 1) \bmod N$, and $(i + 1) \bmod N$. This is the only two-connected network with bidirectional paths between all of the nodes. If the transmission protocols require a response each time a packet of data is transferred between two intermediate nodes on a path, this is the only possible two-connected network. This network was initially considered as a mechanism to make loop networks more reliable.

This network has many of the topological advantages of loop systems. It

- Is defined for any number of nodes,
- Makes geographical sense,
- Has a simple rule for expanding the network by one node at a time, and
- Has a simple routing rule.

When a node is added to the network, the two existing nodes closest to this node are disconnected from one another and connected to the new node. Even if the network covers a large geographical area, there are not many long wires. Shortest-path routing in this network is straightforward. The nodes in the network are sequentially numbered from 0 to $N - 1$. The distance from a source node s to a destination node d is $(d - s) \bmod N$ on the incremental path and $(s - d) \bmod N$ on the decremental path. At the source, the shorter of these two paths is selected. Once a packet in this system starts on a path, it remains

on that path. Therefore, a complete routing decision is made at the source, and the system is implemented as two separate loop systems.

A disadvantage of this network is that the node addresses and the value of N changes whenever a node is added to the system. Either an addressing and routing scheme that does not use this information must be found, or this information must be distributed each time the network is changed. Another disadvantage of this network is that the throughput is not as great as that in the shuffle-exchange and Manhattan street networks.

3.2 Modified shuffle exchange

The modified shuffle-exchange network is based on the shuffle-exchange multistage switch. The network is defined for N nodes, where N is constrained to be a power of two. Node i is connected to nodes $2*i$ mod N and $(2*i + 1)$ mod N. This results in self-loops at nodes 0 and $N - 1$, which are not used to transmit packets. They also make the network less reliable in that a single link failure can disconnect a node from the network. In the modified network, the self-loops are removed and nodes 0 and $N - 1$ are connected to one another, as shown in Fig. 3. When the shuffle-exchange network is part of a hierarchical structure, as in the previous section, the self-loops are replaced by connections to the higher-level network.

Routing in this network is straightforward. Initially, ignore the two paths that were added to the modified network. Represent the address of node i by $M = \log_2 N$ bits, and label the paths to nodes $2*i$ mod N and node $(2*i + 1)$ mod N as 0 and 1, respectively. When a packet is transmitted from node i, the address of the new node has the low-order $M - 1$ bits of node i's address in the high-order $M - 1$ bits. The low-order bit of the new address is 0 or 1, depending on the path selected. To find the shortest path between a source and destination, match as many of the high-order bits of the destination address with the low-order bits of the source address as possible. To get to the destination, shift the low-order bits of the destination address that are not included in this match into the address and determine the path that must be selected.

For instance, assume that the source address is 11011 and the destination address is 11001. The first two bits of the destination address match the last two bits of the source address. The bits 001 must be shifted into the address to get to the destination, and the distance to the destination is 3. To get to the destination, first path 0 is taken to node 10110, then path 0 is taken to node 01100, and finally path 1 is taken to node 11001.

If none of the high-order bits of the destination match the low-order bits of the source, then the distance to the destination is $\log_2 N$ steps.

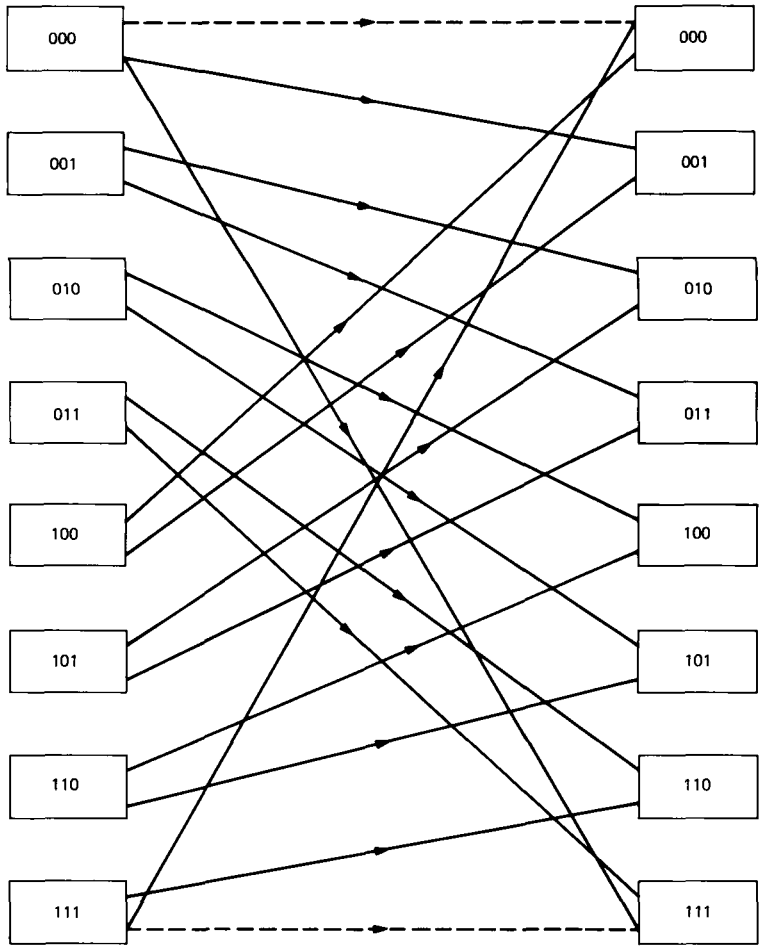

Fig. 3—An 8-node modified shuffle-exchange network.

This is the maximum distance between any source and any destination. In Section 3.4, this will be shown to be the minimum maximum distance between nodes for a two-connected network of this size.

The two paths that are added to the modified network make it possible to go from the all-one address to the all-zero address, and vice versa, in a single step. This shortens the average distance between nodes. These paths cannot change the maximum distance between nodes since this is already a minimum. The effect of these paths on the distance between nodes is shown in Table II. The network S-X is the shuffle exchange with two self loops and MS-X is the modified network. It is evident that the additional paths do not provide a great decrease in the average minimum distance between nodes. They should

Table II—The average and maximum shortest distance between nodes in the shuffle-exchange network and the modified shuffle-exchange network

Nodes	Average		Maximum
	S-X	MS-X	
4	1.50	1.33	2
8	2.11	1.96	3
16	2.83	2.73	4
32	3.65	3.58	5
64	4.53	4.49	6
128	5.46	5.44	7
256	6.42	6.40	8

be included to allow an alternate path when failures occur, but unless a simple routing rule is found to use them under normal operation, they should not be used.

There are several problems with this type of a network. The first problem is that the physical layout of this network does not make sense geographically. If half of the nodes in the network are in one area and half of the nodes are in another remote area, then half of the connections must be between the two remote areas. Therefore, the network can only be used in a small area where the length of the interconnections do not make a difference. Shuffle-exchange networks in physically disjoint areas must be interconnected by a hierarchical network that can make sense geographically, as in the example in Section II. The second problem is that the network is only defined if the number of nodes equals 2^i. At present, no way has been found to add one node at a time—changing a small number of connections—and move from a network with 2^i nodes to a network with 2^{i+1} nodes. The third problem is that the alternate paths between a source and a destination are not good. If the preferred path is blocked or inoperable, the alternate paths are much longer.

3.3 Manhattan street network

The Manhattan street network is based on a grid of alternatingly directed streets and avenues, as shown in Fig. 1. The nodes exist on the corner of a street and an avenue. The rationale for this type of a network is that routing from a particular street and avenue to a destination should be straightforward. As in a city with this layout, any destination street and avenue can be found without asking directions, even when some roads are blocked. In addition, it should be possible to lay out the network to make sense geographically.

The principal difference between a grid connecting corners with streets and a grid connecting nodes with wires is that the physical constraints associated with a two-dimensional surface can be violated more easily with wires. For instance, in the example in Section II, the file servers and terminals forming a community of interest are in the same neighborhood. Assume that the file servers in this system are in the same room and that the personal computers in the same community of interest are in the same physical area. By connecting the file server to the region of the network with the terminals, rather than basing the connections strictly on the physical location of devices, the file server appears to be in the same neighborhood as the terminals. This reduces the interference between terminals in different communities of interest.

The difference in physical constraints also allows the extremes of the grid to be connected. These connections form the grid on the surface of a torus instead of a flat surface. The advantage of this cyclic surface is that there are no corners. Therefore, the maximum distance from a source to a destination is not the distance between two corners of the grid, but the distance between the center and one of the corners. The graph can also be flipped so that the links leaving the center node are always pointed in the same direction. This allows the same routing decision function to be used at every node.

Consider a network with r rows and c columns. The current node has coordinates (i_s, j_s), and the destination node has coordinates (i_d, j_d). The current node is considered to be at location (0, 0), and the relative location of the destination (i, j), is expressed as

$$i = \left\{[1 - 2(j_s \bmod 2)](i_d - i_s) + \frac{r}{2} - 1\right\} \bmod r - \left(\frac{r}{2} - 1\right)$$

$$j = \left\{[1 - 2^*(i_s \bmod 2)](j_d - j_s) + \frac{c}{2} - 1\right\} \bmod c - \left(\frac{c}{2} - 1\right).$$

The current node is now in the center of the network. The value of i is between $-(r/2 - 1)$ and $r/2$, and j is between $-(c/2 - 1)$ and $c/2$. The factors $1 - 2^*(j_s \bmod 2)$ and $1 - 2^*(i_s \bmod 2)$ guarantee that the links leaving the current node point toward increasing i and j. The routing decision now depends only on the relative location of the destination and not on the current node.

The routing preference from the central node to outlying nodes for a 12 × 12, 12 × 14, and 14 × 14 Manhattan street network is shown in Fig. 4. In this network, the two links emanating from the central node are directed upwards and to the right. The routing preference is the shortest distance from the central node to the destination when the link to the right is taken, minus the shortest path to the destination

	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
6	0	4	0	4	0	4	0	0	0	0	0	0
5	4	4	4	4	4	4	0	0	0	0	0	0
4	0	4	4	4	4	4	4	0	0	0	0	0
3	4	4	4	4	4	4	0	0	0	0	0	0
2	0	4	4	4	4	4	4	0	0	0	0	0
1	4	4	4	4	4	4	0	-4	0	-4	0	0
0	0	0	4	0	4	0	-4	-4	-4	-4	-4	-4
-1	0	0	0	0	0	-4	-4	-4	-4	-4	-4	0
-2	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4
-3	0	0	0	0	0	-4	-4	-4	-4	-4	-4	0
-4	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4
-5	0	0	0	0	0	0	-4	0	-4	0	-4	0

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
6	2	0	4	0	4	0	4	0	0	0	0	0	0	0
5	2	4	4	4	4	4	4	0	0	0	0	0	0	2
4	2	2	4	4	4	4	4	4	0	0	0	0	0	0
3	2	4	4	4	4	4	4	0	0	0	0	0	0	2
2	2	2	4	4	4	4	4	4	0	0	0	0	0	0
1	2	4	4	4	4	4	4	0	-4	0	-4	0	-2	2
0	-2	2	0	4	0	4	0	-4	-4	-4	-4	-4	-4	-2
-1	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2	-2
-2	-2	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2
-3	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2	-2
-4	-2	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2
-5	0	0	0	0	0	0	0	-4	0	-4	0	-4	0	-2

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
7	2	2	2	2	2	2	2	-2	0	-2	0	-2	0	0
6	2	2	4	2	4	2	4	2	0	0	0	0	0	0
5	2	4	4	4	4	4	4	0	0	0	0	0	0	2
4	2	2	4	4	4	4	4	4	0	0	0	0	0	0
3	2	4	4	4	4	4	4	0	0	0	0	0	0	2
2	2	2	4	4	4	4	4	4	0	0	0	0	0	0
1	2	4	4	4	4	4	4	0	-4	0	-4	0	-2	2
0	-2	2	0	4	0	4	0	-4	-4	-4	-4	-4	-4	-2
-1	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2	-2
-2	-2	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2
-3	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2	-2
-4	-2	0	0	0	0	0	0	-4	-4	-4	-4	-4	-4	-2
-5	0	0	0	0	0	0	-2	-4	-2	-4	-2	-4	-2	-2
-6	0	0	2	0	2	0	2	-2	-2	-2	-2	-2	-2	-2

Fig. 4—Routing preference in a 12 × 12, 12 × 14, and 14 × 14 Manhattan street network.

when the upwards-directed link is taken. Therefore, a negative number implies that the right link leads to the shortest path to the destination, and a positive number implies that the upwards link yields the shortest path to the destination. The magnitude of the number shows how much longer the distance to the destination would be if a packet were forced to take a less desirable path. A zero implies that the distance to the destination is the same along either path. The figures show that to get to half of the nodes either path can be taken, to get to a quarter of the nodes the left path should be taken, and to get to the other quarter of the nodes the right path should be taken. The figures also

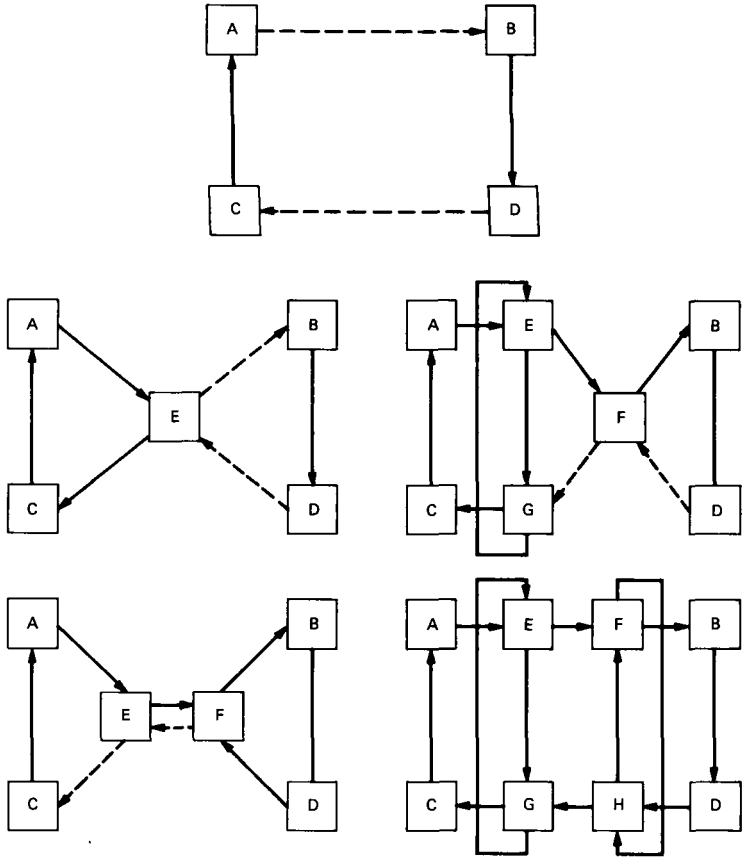

Fig. 5—Adding nodes E, F, G, and H one at a time to the basic rectangular structure consisting of nodes A, B, C, and D in a Manhattan street network.

show that if a packet is forced to take the wrong path, the increase in path length to the destination is never more than four.

One problem with the shuffle-exchange network is the difficulty in changing the number of nodes in the network. Figures 5 and 6 show how nodes may be added to the Manhattan street network. Figure 5 shows how two columns are added to the basic square structure within the Manhattan street network. The dotted lines show the links that will be broken when the next node is added. Figure 6 shows how the procedure is continued to add nodes to partially full columns. Each time a new node is added, two links are broken and connected to the new node. This is no greater than the number of links that must be broken in the bidirectional loop. Eventually this procedure leads to a network with two additional rows or columns, and the pattern of alternatingly directed rows and columns is preserved.

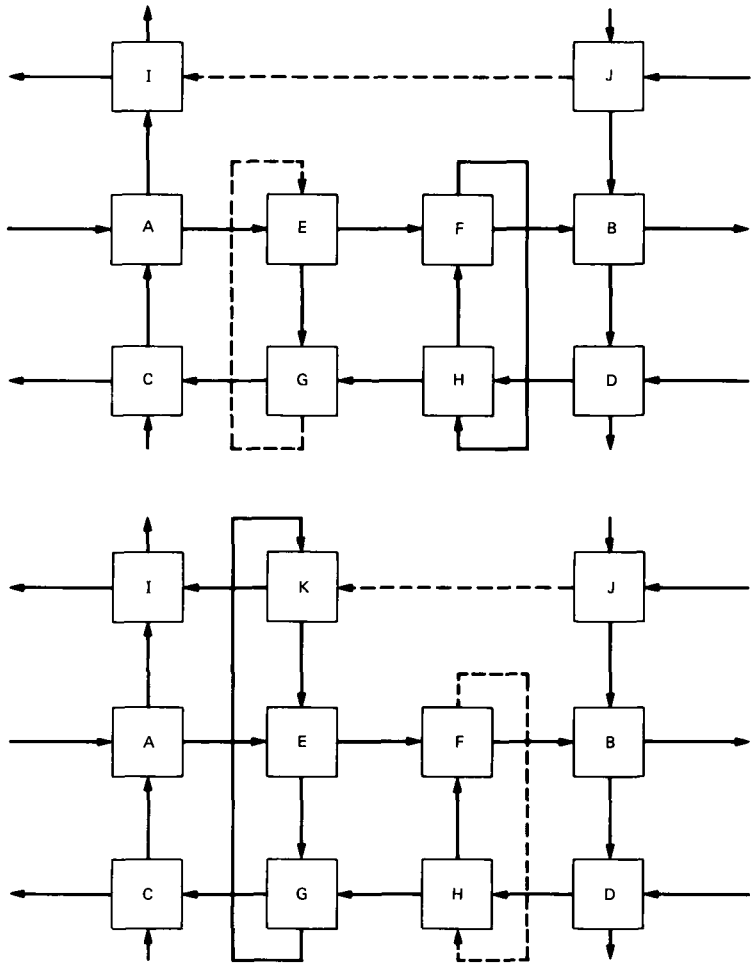

Fig. 6—Adding a node K to a partially full column in a Manhattan street network.

When adding a new node both the physical position of the node and the topology of the network must be considered. It is desirable to connect the node to the nearest existing nodes, but it is also desirable to start as few new rows or columns as possible, and to keep the number of rows and columns equal. When rows and columns are kept approximately equal, the average and maximum shortest paths between nodes increase as shown in Table III.

In the shuffle-exchange network it is occasionally better to establish a hierarchy of networks rather than make a single network larger. Hierarchical structures are also useful in Manhattan street networks. They are used to

Table III—Distances between nodes in $2i$ by $2j$ Manhattan street networks

Nodes	Rows	Columns	Shortest Paths Between Nodes	
			Average	Maximum
4	2	2	1.33	2
8	2	4	2.00	3
16	4	4	2.93	5
24	4	6	3.30	5
36	6	6	3.71	6
48	6	8	4.34	7
64	8	8	5.02	9
80	8	10	5.42	9
100	10	10	5.84	10
120	10	12	6.42	11
144	12	12	7.02	13
168	12	14	7.45	13
196	14	14	7.89	14
224	14	16	8.45	15
256	16	16	9.02	17

- Decrease the number of paths between physically distant sections of the network,
- Eliminate long paths between communities of interest, and
- Prevent traffic between communities of interest from affecting communications in other communities of interest.

The two-connected strategy can be maintained, as in Fig. 7. However, this will make routing more complex. An alternative is to connect one or more of the nodes in a local area to a higher-level network, as shown in Fig. 8. By using this approach, routing decisions in a local area are not affected by network changes in other areas, and addresses in different local areas are assigned independently. A hierarchical addressing and routing structure, similar to that used in the telephone system, can be used. For example, the address within the local area corresponds to a phone number, and the address of the local network on the higher-level network corresponds to the area code. When sending a packet within the local area an area code is not required.

3.4 An optimal two-connected network

Certain characteristics of the "best" networks are difficult to quantify. For instance, it should be possible to add nodes without making major reconfigurations, create geographically dispersed networks without adding excessive numbers of long links, and establish communities of interest. Other characteristics, such as the average and maximum number of links between nodes, can be compared and bounded.

Consider the class of two-connected networks. From a particular node, at most two nodes can be reached in one step, four additional

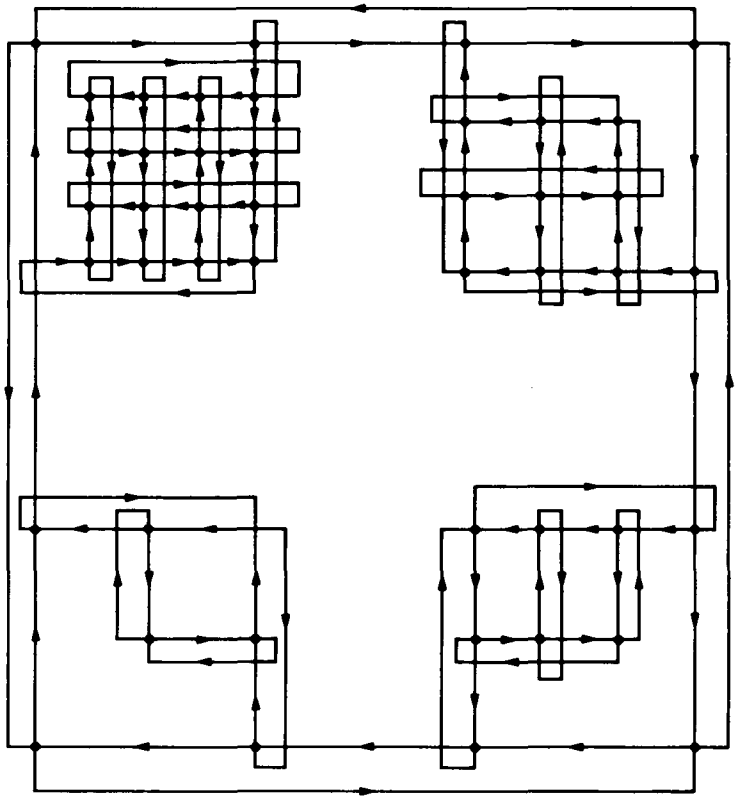

Fig. 7—A hierarchical Manhattan street network in which all of the nodes are two-connected.

nodes in two steps, and so on. The destination nodes form a binary tree. If, at any level in the tree, a destination node recurs, the number of new nodes that can be reached in future levels is reduced by the descendants of that node. Therefore, the maximum number of nodes that can be reached in m steps is

$$\sum_{i=1}^{i=m} 2^i = 2^{m+1} - 2.$$

If every node in a two-connected network can reach this number of new nodes in each m steps, then the network has the smallest average and maximum distance between nodes. In general, networks with these characteristics do not exist. However, this is a lower bound on these distance characteristics.

In the shuffle-exchange network with 2^j nodes, each node must reach $2^j - 1$ nodes, and the maximum minimum distance between

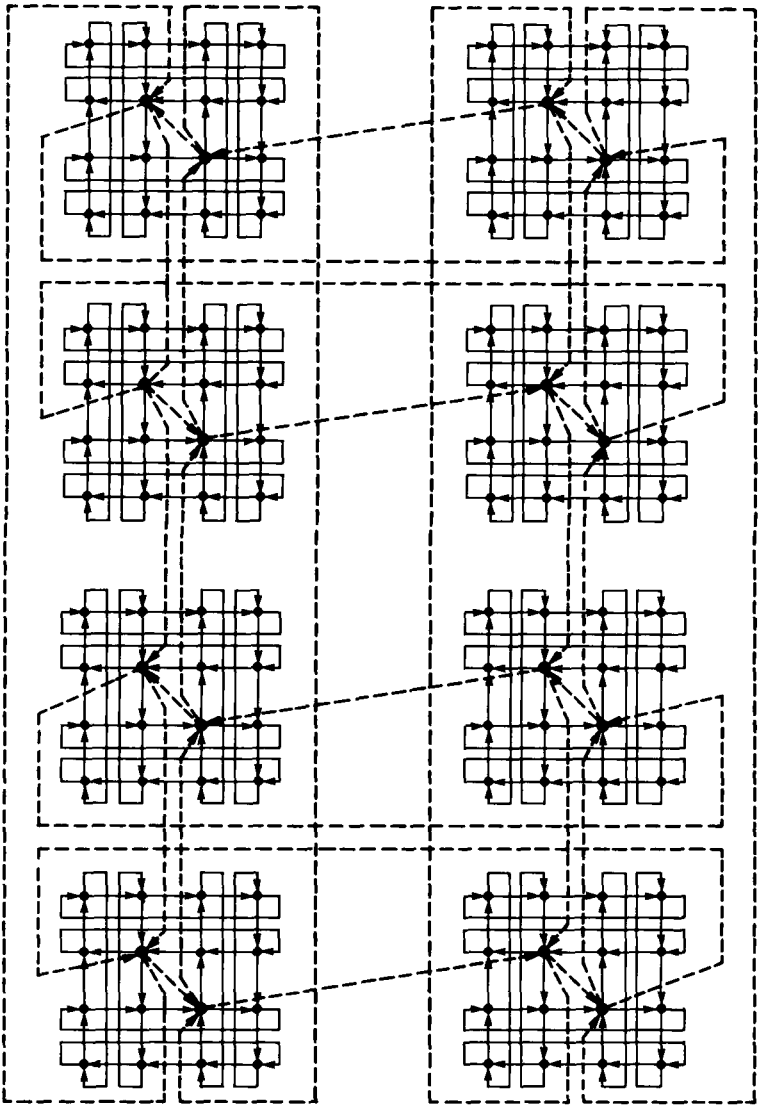

Fig. 8—A hierarchical Manhattan street network in which the nodes connected to the hierarchical network are four-connected.

nodes is j. The number of nodes that are reached in $j-1$ steps in the optimum network is 2^j-2. Therefore, if 2^j-1 nodes must be reached on the optimum network, the minimum distance to the furthest node is j, and the largest minimum distance between nodes in the shuffle-exchange network is less than or equal to that in any network with 2^j nodes.

A comparison of the average and maximum distance between nodes

Table IV—A comparison of the distances between nodes in several networks

Net	Distance	Number of Nodes		
		16	64	256
Opt	Avg	2.53	4.19	6.06
	Max	4	6	8
S-X	Avg	2.73	4.49	6.40
	Max	4	6	8
MSN	Avg	2.93	5.02	9.02
	Max	5	9	17
BDL	Avg	4.26	16.25	64.25
	Max	8	32	128

for the optimum, shuffle-exchange, Manhattan street network, and the bidirectional loop is shown in Table IV. In both the optimum network and the shuffle-exchange network, the maximum distance between nodes varies as the log of the number of nodes. In the Manhattan street network, the maximum distance between nodes varies as the square root of the number of nodes, and, in the bidirectional loop, this distance varies linearly with the number of nodes. The same relationship is also noted between the number of nodes in these networks and the average distance between nodes. The average distance between nodes shows what fraction of the network resources is used to transfer a packet and provides an indication of the relative throughput of the networks. Although, as shown in Section II, there are other factors that also affect the throughput.

IV. IMPLEMENTATION

In a mesh network, as in a loop network, the communications lines are point-to-point links with a single transmitter and a single receiver. Transmission on these links is much simpler than in a broadcast network with a shared communication channel. The access protocols are simpler because there is only one source, and it is not necessary to multiplex users on the communication channel or resolve collisons. The receiver is simpler because the distance between the source and destination is constant, and the signal strength does not change by a large amount from packet to packet. Regenerating the signal to eliminate distance constraints is simpler because signals only propagate in one direction. Timing recovery is simpler because the source can transmit continuously and bit synchronization does not have to be reestablished at the beginning of each packet. In addition, the communication line does not have many taps and is compatible with the current generation of fiber-optic equipment.

In a two-connected network there are two links and a local source inputting data to a node, and two links and a local sink removing data

from the node. Occasionally, multiple inputs try to transmit data to the same outgoing link. One way to resolve this problem is to queue packets waiting for a link. The network now assumes the complexity of a store-and-forward network. Not only must potentially large packet queues be maintained, but adaptive-routing, flow-control, deadlock-avoidance, and packet-resequencing issues must be addressed.

In this section, the slotted-system and register-insertion techniques, developed for loop communication systems, will be extended to mesh networks with equal in and out degrees. The general strategy guarantees that every packet arriving on an incoming link, and not destined for the node, will be transmitted on one of the outgoing links. Therefore, it is not necessary to maintain a packet queue for the links emanating from the node. The requirement that packets passing through the node take one of the outgoing paths results in longer paths when the shortest path to the destination is busy. However, it is possible to design networks to reduce the effects of incorrect paths. For instance, in the Manhattan street network the path to only half of the destinations is increased if a packet is forced to take one path rather than the other. In addition, if a packet is forced to take a less desirable path, the distance to the destination is increased by at most four.

The storage between the local source and the network is also limited. Packets from the local source are only transmitted when one of the outgoing links is not being used by an incoming link. It is assumed that either the local source can be throttled when the network is busy, or that the source provides data at a low rate relative to the network transmission rate. In the latter case, when a packet is lost, it must be recovered by a higher-level protocol. If the network delivers packets faster than the local sink can accept them, packets are either transmitted on one of the outgoing links or discarded. In the former case, the network is used for storage. Since new packets cannot enter the network when it is recirculating old packets, this transmission strategy acts as a flow-control mechanism. The assumptions on the local source and sink are implicit in all loop-configured systems without infinite storage.

The packets of data in a slotted system are fixed size. A node continuously transmits bits on each of the links emanating from the node, and periodically transmits a start-of-slot indication. The start-of-slot indication is followed by a packet of data or an empty slot. In the interval between the start-of-slot transmissions, at most one packet of data is received on each of the incoming links. The packets that are received between start-of-slot transmissions are forwarded after the start of slot is transmitted. These packets are switched to one of the outgoing links or the local sink before data from the local

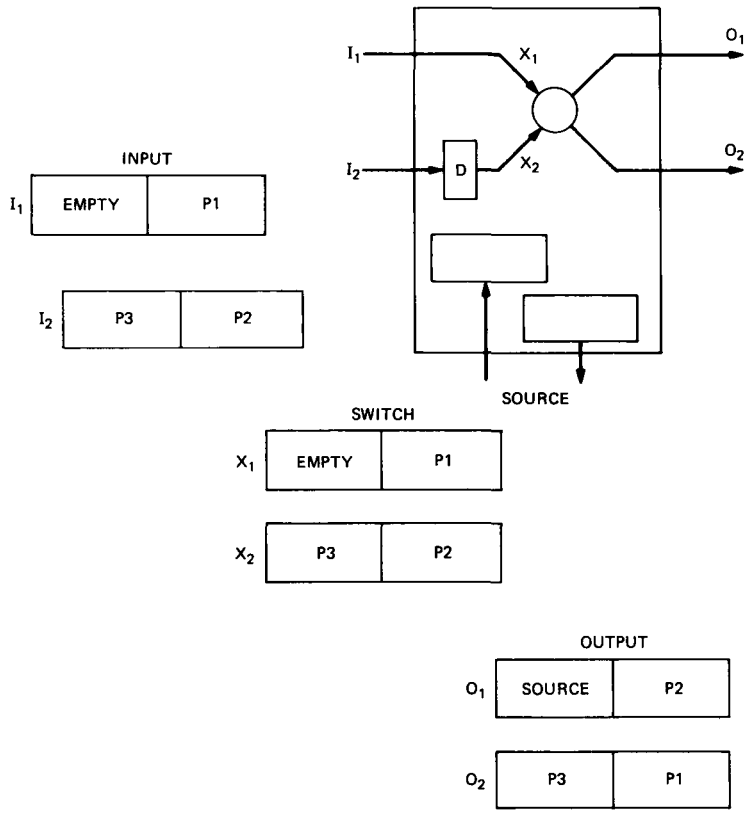

Fig. 9—Extension of slotted-loop systems to a mesh network.

source are given access to the slot. Since there are the same number of links arriving and leaving from each node, and the local source can be throttled, a queue of packets will not accumulate. The operation of a slotted system without a packet queue is shown in Fig. 9.

The interface for a register-insertion loop is shown in Fig. 10. Packets in this system are variable in size, but constrained to be less than the storage register W_l. The local source is only allowed to transmit when register W_l is empty. Since a packet from the local source is less than W_l, all data received from the loop while the local source is transmitting can be stored in W_l. When register W_l is not empty, bits from this register are transmitted on the loop. Therefore, the length of this register remains the same when bits are being received, and decreases when bits are not received. As long as this register is not empty, gaps between arriving packets are removed.

The register-insertion technique can be applied to a mesh network in which the in degree and out degree of the nodes are the same by

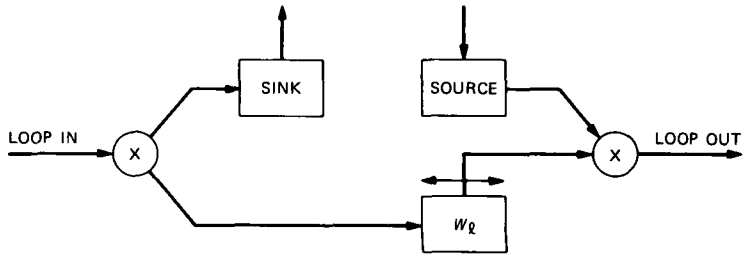

Fig. 10—A register-insertion access unit in a loop system.

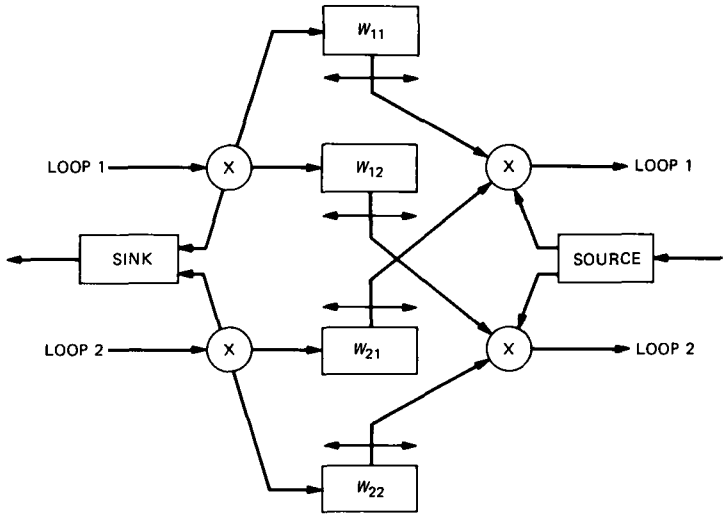

Fig. 11—Extension of register-insertion systems to a mesh network.

making the node appear as if several loops are passing through it. This is shown in Fig. 11. Registers W_{11} and W_{22} correspond to register W_l for loops 1 and 2, respectively. In addition to the local sink, register W_{12} appears to be a sink for loop 1. And, in addition to the local source, register W_{12} appears to be a source of data for loop 2. Therefore, register W_{12} allows messages on loop 1 to transfer to loop 2. As in a loop system, if buffer W_{12} is full the packet must continue around loop 1. Buffer W_{21} serves the same purpose for packets crossing from loop 2 to loop 1.

The register-insertion technique allows variable-length packets. When the system is busy, each node eliminates the null space between incoming packets to efficiently use the outgoing links. The slotted-system technique uses fixed-size packets. If there is less than a packet of data, the packet is partially empty. Therefore, the register-insertion

technique uses the channel more efficiently. In a slotted system, the packets from the incoming links are aligned at the switching point. A packet only traverses a longer path if more than one incoming packet requires the same outgoing channel. In a register-insertion system, a packet can only transfer from one loop to the other if the crossover buffer is empty. It is possible that both packets on the through loops would rather be on the other loop, but cannot cross over because the buffers are full. Therefore, the register-insertion technique uses individual links more efficiently, but the slotted system takes a shorter path between the source and destination. The technique that provides the greater throughput depends on both the message-length distribution and the network topology.

A small amount of buffering can be included in either the slotted or register-insertion system to reduce the probability of a packet taking a longer path. In the slotted system, fixed-size packet buffers are inserted at the output channels. The probability that a packet must take a longer path is the probability that two arriving packets must take the same path and the buffer for that path is full. Without buffering, one packet must take the longer path whenever two arriving packets want to take the same path. In the register-insertion system, the additional storage is inserted at the crossover point. A packet cannot cross over if, when it is received, there are fewer bits available in the crossover buffer than there are in a maximum-size packet. In the original system, a packet cannot cross over if, when it arrives, the crossover buffer is not empty. Decreasing the probability that a packet takes a longer path decreases the fraction of the network resources that a packet uses, and increases the throughput of the system. The trade-off between buffering and system throughput remains to be investigated.

V. FILE TRANSFER

A file transfer consists of several packets being transmitted from a source to the same destination. In a system in which packets do not take the same path, it is possible that packets are not received in the same order that they are transmitted. Packets may be resequenced at the receiver, however, it is preferable to avoid this task.

One possible solution to this problem is to transmit one packet at a time and wait for an acknowledgment. This reduces the file-transfer rate. However, because of the small delays at each node, this solution is not as bad in mesh networks as it is in store-and-forward networks. For instance, in a slotted system the average delay per node is half a slot time, and the average round trip delay equals the average number of hops between nodes, $\bar{L}$, times the slot time. Therefore, there is an average of $\bar{L}$ slots between each packet in the file, and the file-transfer

rate equals the channel rate divided by $\bar{L} + 1$. Higher file-transfer rates can be achieved by end-to-end protocols that take advantage of the delay characteristics of the system, or node protocols that take advantage of specific hardware structure.

Because of the small amount of delay at each node, it is unlikely that packets that take routes that have approximately the same length will arrive out of sequence. This probability can be reduced by allowing a small number of slots between packets in the same file transfer. A simple file-transfer protocol, which takes advantage of this characteristic, operates like the window protocols of store-and-forward networks and the go-back-N protocols of satellite systems. This protocol labels packets in a file transfer with a sequence number and a retry number. At the beginning of a file transfer, the transmitter and receiver start with the same sequence and retry number. The sequence number is the order of the packets. The receiver

- Increments its retry number when a packet with the expected retry number and a larger sequence number is received,
- Sends a positive acknowledgment if a packet has a sequence less than or equal to the expected number,
- Sends a negative acknowledgment with its retry number and expected sequence number if the packet has a larger sequence number than it expects, and
- Commits a packet if it has the expected sequence number.

The transmitter

- Stops saving a packet for retransmission when it receives a positive acknowledgment for a packet with a sequence number greater than or equal to the expected number,
- Adopts a new retry number and starts retransmitting from a negatively acknowledged sequence number if a negative acknowledgment with a larger retry number is received, and
- Periodically retransmits the last packet in a file transfer until it receives an acknowledgment.

The transmitter initially transmits packets in the file transfer in every available slot. However, when it receives negative acknowledgments, it increases the number of slots between subsequent packets.

Since this protocol only accepts packets in the correct order, packets do not have to be resequenced at the receiver. In a mesh network, several packets can be in transit between the source and destination. If the receiver misses a packet, it must send a negative acknowledgment for every packet with a larger sequence number than expected to be certain that the trasmitter receives the negative acknowledgment. The retry number is included so that the transmitter only backs up and starts retransmitting when it receives the first negative acknowledgment to an outstanding packet.

The transmitter adaptively changes the number of slots between packets according to network load and the rate of the receiver. When the network is lightly loaded, all packets follow the best path to the receiver, and arrive in sequence. If the receiver can accept packets at this rate, there are no negative acknowledgments, except for infrequent transmission errors, and the file-transfer rate equals the channel rate. When the network is heavily loaded, the packets follow different paths, are received out of sequence, and the file-transfer rate decreases. If the receiver cannot accept packets as fast as the transmitter can deliver them, the buffer in the interface unit is full when the packets arrive. In the systems described, these packets are directed to one of the output links at the node, and recirculate in the network until the buffer is available. These packets arrive out of sequence, negative acknowledgments are transmitted, and the transmitter slows down.

Additional improvements are possible by taking the structure of the nodes into account. In a slotted system, subsequent packets in a file transfer can be marked. At a node, packets in a file transfer, which follow immediately behind one another, can be directed along the same path. The end-to-end protocol can be implemented with empty slots only occurring when the source cannot deliver packets quickly enough, or when the channel at the source node is busy with traffic passing through the node. This will improve the file-transfer rate on moderately used channels. If this modification is used, the transmitter and receiver must negotiate the number of packets in a continuous sequence to be certain that the receiver can accept them at the channel rate.

In a register-insertion system, if the loop paths define a Hamiltonian circuit, packets in a file transfer can be constrained to follow these loops. Since a packet can never be denied access to this loop, all packets follow the same path and will be received in sequence. This allows file transfers to occur at the channel rate without resequencing, but requires file transfers to use a larger fraction of the network resources.

In both the register-insertion and slotted systems, it is possible to use a higher-level protocol to set up a limited number of virtual circuits along which file transfers can occur efficiently at the channel rate without resequencing. In the slotted system, the higher-level protocol is used to assign an input at a node to an output. When a file-transfer packet arrives at a node input, it is given first priority to the assigned output. Therefore, all packets in the file transfer follow the same path and do not have to be resequenced. The function of the higher-level protocol is to make the assigned paths for a file transfer use as few links as possible. The problem with this approach is that the preferred paths must be established at the beginning of each file transfer and

disabled at the end of the transfer. In addition, a file transfer may be temporarily blocked by previously assigned paths, creating a need for a scheduler. In the register-insertion system, the preferred paths are established as the paths through the node. This has the same problems as the slotted system.

VI. CONCLUSION

Mesh networks increase the throughput of conventional local area networks by decreasing the fraction of the network capacity needed to transmit information between a source and a destination. These networks have multiple paths between each source and destination, thus increasing the reliability of local networks. The networks consist of point-to-point links, and can be extended to cover a metropolitan, rather than a local, area.

In general, mesh networks require complex store-and-forward nodes that also route messages, control the flow of data entering the network, resequence packets at the destination, and recover packets with errors. There are characteristics of the local or metropolitan area that allow these functions to be simplified. In the local environment, regular network topologies can be selected in which routing is straightforward. The lower error rates make it reasonable to recover errors on an end-to-end basis. This allows loop-access protocols to be extended to mesh networks, eliminating the need for buffering and additional flow-control protocols. Extensions for the slotted system and register-insertion techniques used in loop systems have been shown. Buffering can be included in these systems to improve the channel utilization; however, channels are less expensive in the local environment. The small node delays in these systems also make it reasonable to implement file-transfer protocols that do not require packet resequencing.

Three mesh networks have been studied, and the desirable characteristics of networks have been determined. Networks should have regular structures with simple routing rules, and should not have single points of failure. By minimizing the average and maximum distance between nodes, the fraction of the network resources used to transmit a packet decreases, and the throughput increases. This can be done by packing topologies with these characteristics noted, and by locating communities of terminals that communicate frequently in the same area of the network. Equal-length alternate paths between sources and destinations reduce the probability of bottlenecks and the need for buffering within a node. Networks will change, and it must be possible to add or delete nodes without changing a large number of connections. If the network covers a large area, it must be possible to limit the connections between nodes in different areas. Of the networks studied, the Manhattan street network has all of these characteristics.

REFERENCES

1. E. H. Steward, "A Loop Transmission System," Conf. Rec. Intern. Conf. Commun., San Francisco, June 1970, pp. 36-1-9.
2. N. Abramson, "The ALOHA-System—Another Alternative for Computer Communications," University of Hawaii Tech. Rep. B70-1, April 1970, AD707853.
3. R. M. Metcalf and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Commun. ACM, *19* (July 1976), pp. 395–404.
4. M. Schlatter and J. L. Massey, "Capacity of Interconnection Ring Communication Systems With Unique Loop-Free Routing," IEEE Trans Inform. Theory, *IT-29*, No. 5 (September 1983), pp. 774–8.
5. J. R. Pierce, "How Far Can Loops Go," IEEE Trans. Commun., *COM-20*, No. 3 (June 1972), pp. 527–30.
6. Y. Yemini, "Tinkernet: Or Is There Life Between LANs and PBXs," Proc. ICC'83, *3*, Boston, June 1984, pp. 1501–5.
7. T. N. Saadawi and M. Schwartz, "Distributed Switching for Data Transmission Over Two-Way CATV," Proc. ICC'84, Amsterdam, May 1984, pp. 1409–13.
8. D. C. Walden, "Experiences in Building, Operating and Using the ARPA Network," Second USA-Japan Computer Conference, Tokyo, August 1975.
9. C. Petitpierre, "Meshed Local Computer Networks," IEEE Commun. Mag., *22*, No. 4 (August 1984), pp. 36–40.

AUTHOR

Nicholas F. Maxemchuk, B.S.E.E., 1968, City College of New York; M.S.E.E., 1970, Ph.D., 1975, University of Pennsylvania; RCA David Sarnoff Research Center 1968–1976; AT&T Bell Laboratories, 1976—. Mr. Maxemchuk is presently Head of the Distributed Systems Research Department. Since joining AT&T Bell Laboratories, he has done research on computer-communication networks, virtual and speech editing, and picture processing. From 1980 to the present, he has been on the adjunct faculty of the University of Pennsylvania, where he teaches a course on computer communications networks. From 1980 to 1985, he was the Associate Editor, then the Editor of Data Communications for the IEEE Transactions on Communications. Member, Eta Kappa Nu, Tau Beta Pi.