NASA Technical Paper 2605

July 1986

Hybrid Routing Technique for a Fault-Tolerant, Integrated Information Network

Barry D. Meredith

NASA

NASA
Technical
Paper
2605

1986

Hybrid Routing Technique for a Fault-Tolerant, Integrated Information Network

Barry D. Meredith

Langley Research Center
Hampton, Virginia

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

Summary

The evolutionary growth of the Space Station and the diverse activities onboard are expected to require a hierarchy of integrated, local area networks capable of supporting data, voice, and video communications. In addition, fault-tolerant network operation is necessary to protect communications between critical systems attached to the net and to relieve the valuable human resources onboard the Space Station of time-critical data system repair tasks. A key issue for the design of the fault-tolerant, integrated network is the development of a robust routing algorithm which dynamically selects the optimum communication paths through the net. A routing technique is described that adapts to topological changes in the network to support fault-tolerant operation and system evolvability.

Introduction

The NASA Office of Aeronautics and Space Technology is devoting part of its resources to the development of technologies that will support the needs of an orbiting Space Station. One area of involvement for the Langley Research Center has to do with the development of .data system technologies which fit into the context of the Space Station as currently envisioned. The Space Station should be established in Earth orbit in the early 1990's and will initially consist of a minimum number of modules or compartments. There would then be a buildup phase where compartments are added and joined as per the functional activities onboard the station. In the mature phase, the Space Station will have reached its maximum structural size (but would allow for the replacement of modules) and will possess its maximum operational capability.

The evolutionary growth of the Space Station, along with the uncertainty of that growth, has implications on the design of the data system. Electronic systems also need to be added or taken away to meet changing requirements and to accommodate the continuing advances in electronics technology. One potential solution involves establishing a hierarchy of computer networks for the Space Station (ref. 1). Each module would contain its own local area network (LAN) (refs. 2 and 3), which would be joined to networks in other modules via gateways or nodes to form a Space Station-wide network. The Space Station network would be able to communicate with Earth-based networks over a telecommunications channel producing a global data communications and processing system.

Although there are a large variety of local area networks currently available (refs. 2 and 3), they possess some notable weaknesses in terms of their ability to address the diverse data communication requirements of the Space Station. Commercially available LAN's lack the performance characteristics necessary to accommodate high data rate applications such as the transfer of real-time, video data to and from points in the Space Station. Full motion video displayed at various work stations would augment a worker's ability to perform proximity operations onboard the Space Station. In addition to the performance limitations, most existing networks are employed for office automation tasks and provide little (if any) fault tolerance. The ability to detect and recover from network faults with minimum delay is particularly essential when critical systems, such as on-orbit control, are attached to the net and are dependent upon it for reliable data communications. It is also desirable to employ fault-tolerant networks even though the attached systems are not critical to mission safety. This would relieve the valuable human resources onboard the Space Station of time-critical data systems maintenance chores.

The development of technologies for an integrated, fault-tolerant information network capable of supporting data, voice, and video communications is the emphasis of a research program at the Langley Research Center. Since the network must provide alternate communication paths for fault-tolerant operation and be capable of evolving along with the Space Station, the development of a robust routing algorithm is a critical design issue. This paper presents a description of a routing technique which was developed to meet the diverse requirements of a Space Station information network.

Network Topology and Switching Considerations

Essential to the issue of fault tolerance is the selection of an appropriate topology for the network. Typical network topologies (e.g., ring and bus) offer only one or two possible routes between any message source and destination. Although this minimizes system complexity (e.g., routing decisions are trivial), very little fault tolerance is achieved. The Space Station network should provide many alternate paths between attached systems to achieve a sufficient degree of fault tolerance. The penalty for this feature is increased system complexity; however, information can then be routed around or away from faulty network elements so that normal operation is maintained (at somewhat reduced capacity) after a failure occurs. In addition, increased performance can be obtained since multiple paths can also support multiple, simultaneous communications.

The mesh topology (fig. 1) allows a high degree of connectivity to achieve fault tolerance and high performance. The nodes are intelligent interfaces which service their attached host systems by controlling the flow of information in the network. The maximum degree of connectivity is achieved when each node has a link to every other node in the net. The fully connected mesh is, however, impractical for large networks since the number of links grows as the square of the number of nodes. For the Space Station, a partially connected mesh (four to eight links per node) is feasible. In general, the partially connected mesh is an irregular topology; that is, no specific node interconnect scheme is implied. Therefore, the interconnections between nodes can be tailored to the specific distributed processing application so that related systems can be attached regardless of physical location. The braided mesh (fig. 2) is a special case of the mesh topology that offers a regular connection scheme. Here, each node is connected to its left and right neighbor and to the neighbors successor (four-port node). This regular topology simplifies the layout of the network around the habitation modules of the Space Station and also simplifies the procedure for adding (or deleting) a node to the network (fig. 3). In addition, regularity of interconnects, in general, tends to reduce the complexity of the routing problem. The routing algorithm subsequently described in this paper is applicable to the braided mesh as well as any irregular mesh topologies. The major drawback to the braided mesh is that the worst-case communications path between the two nodes at either extreme of the braid will include all the nodes in between those extremes. For a large network, this could result in excessive message delay and/or signal attenuation; however, further definition and evaluation of the node technology is necessary before definitive limitations to network size can be established.

Real-time, video communications onboard the Space Station will require a network whose data paths offer high throughput and minimum delay. This requirement for video cannot be obtained by using conventional packet-switching concepts (ref. 4). The delay associated with the store and forward process at each node is excessive for real-time applications. One solution is to establish a physical, high-throughput circuit between communicating sites on the net. This technique is known as circuit switching. The high-performance, integrated network will be configured to support circuit switching (fig. 4), and the network nodes will be responsible for establishing a host-to-host circuit prior to the start of communications. That circuit will remain intact until all data transfers between the two points are complete.

Although circuit switching meets the requirements for video data, interactive or "bursty" data communications are better suited to packet switching. This switching technique allows network links to be shared by host systems in a time division multiplexing fashion. Therefore, more efficient bandwidth utilization is realized when bursty data are accommodated with packet as opposed to circuit switching. One packet-switched method involves transporting packets through the network over virtual circuits (ref. 4). In this technique, the source to destination path is established prior to data transfer, as for physical circuits; however, there exists no direct connection through the intermediate nodes. Packets are stored at each node and are then forwarded over the appropriate logical output channel as a function of which input channel they employed on arrival. Input-to-output channel assignments are made at each node along the path during the virtual circuit establishment process. The routing algorithm for the integrated information network must accommodate both virtual and physical circuit establishment to support the diverse network communications.

Routing Requirements and Algorithm Classifications

The selection of a robust routing technique for the integrated information network is critical to achieving desired system features such as high-performance, fault-tolerant operation, evolvability, and minimum message delays. Several key requirements for routing in the network are listed in table I. Most all routing algorithms compute the "shortest path" through the network; however, the algorithm must also take into account the state of the network and the overhead associated with establishing the path to be truly effective. For example, an inoperative node or a dead link must be detected and subsequently avoided during path generation to ensure delivery of the data to the destination. In addition, the circuit setup time must be kept to a minimum so that the network can respond to service requests in a timely fashion. The routing scheme should include some mechanism for congestion control to minimize message delay. For packet-switched networks, this involves measuring the traffic flow at nodes in the network so that new routes can be steered away from heavily loaded points in the network. This procedure tends to smooth out or balance traffic around the network. When conventional circuit switching is employed, links are dedicated to a physical circuit and are unavailable for use in any other communications path until the circuit is disconnected. Therefore, the routing technique should be capable of bypassing the busy links and selecting the "next-best" path.

One of the primary attributes of local area networks in general is their ability to support data system evolvability. Therefore, a robust routing algorithm would automatically recognize the addition (or deletion) of a node or link to the system and adjust the subsequent routing decisions accordingly.

Most of the diverse algorithms which exist for routing data in a network can be placed into one of the four general classes (ref. 5) listed in table II. The first category is the nonadaptive which encompasses both fixed and random routing. In fixed routing, no attempt is made to adjust to changing network conditions. The routing table for each node, which specifies the best outgoing link toward every other node in the net, must be altered manually in order to respond to topological changes. Therefore, fixed algorithms are of little practical use. One example of random routing is a technique known as flooding. Here, messages arriving at a node in the network are forwarded over all the outgoing links. Although no routing decisions are necessary and the message is likely to arrive at the intended destination, this "scatter-gun" approach greatly increases the congestion on the network.

Implicit in the centralized algorithms, the second category, is the use of a single processor as the network routing center which dictates routing decisions to all the nodes in the network. This central authority possesses global knowledge of the state of the network and employs this knowledge to adapt the network to changes in topology and communications load. The concept of a global perspective offers the potential for near-optimum routing. However, the existence of a single route manager represents a performance bottleneck because of the overhead associated with exchanging large amounts of state and route information between the nodes and the central authority. In addition, centralized approaches are susceptible to a single-point failure at the route center if redundancy techniques are not employed at that point.

Both the isolated and distributed adaptive, the third and fourth categories, are classes of decentralized algorithms; that is, routing decisions are made by the individual nodes in the network. The major difference between the two is that in the isolated adaptive case, each node bases its perception of the network state solely upon local data, whereas for the distributed algorithms, neighboring nodes exchange information to achieve a broader network perspective. The information obtained by the collaborative method, in general, is more precise than that obtained by the independent operation. Therefore, the distributed approach can produce more optimum routing.

The algorithm described in this paper can best be classified as a hybrid approach to routing. In this scheme, the nodes operate in a decentralized, isolated fashion to establish communications paths through the network, and a central authority is introduced to provide evolvability and route optimization. This hybrid routing technique relieves the communications bottleneck associated with a totally centralized approach while maintaining a global view of the network condition.

Decentralized Route Selection

In the hybrid routing technique, the nodes in the network operate in a decentralized manner to establish physical and virtual circuits through the net. A source to destination communications path is formed as each node along the path refers to its local routing table to select the best output link or port toward the specific destination node.

This local route table (fig. 5) contains a list of the addresses of all the other nodes in the network, and for each address there is a ranking of the node output ports. (See ref. 6 and pp. 95–104 of ref. 2.) The ports are ranked according to the choice which will provide the shortest distance to the corresponding destination address. A port status table is also resident in each node to specify which ports are operational, and for circuit-switched communications, whether the ports are, at that time, supporting a physical circuit. For packet-switched applications, this table might specify the number of logical channels per port which were actively supporting virtual circuits so that heavily loaded ports could be avoided.

A local routing and port status table is presented in figure 6 for one node (address d) of a seven-node braided mesh network. For an example, consider the case where node d attempts to establish a circuit-switched connection with node b. By examining its routing table, node d recognizes that the shortest path to node b is over port 2. However, port 2 has been declared dead according to the status table. Therefore, node d selects the second highest priority path, over port 1 to node c. Node c will then refer to its local route table and complete the connection to node b via its port 1, if that resource is available.

In summary, this decentralized approach utilizes alternate route information so that paths are dynamically selected based on the availability of links. As a result, faulty links and busy circuits can be avoided. This technique alone, however, is insufficient in meeting all the routing requirements of the information network. The limitations of the algorithm can be attributed to the narrow, local perspective the individual nodes possess of the network world around them. For example, nodes adjacent to an inoperative destination node would be unaware of that fact. Each neighbor would perceive that its port to the failed node is down and would attempt, to no avail, to contact that destination over an alternate route. The procedure for determining that the path search process should, in this case, be aborted involves significant communications overhead. Another shortcoming of this algorithm as a stand-alone approach is its inability to support network evolution. The mechanism that would allow the nodes to adapt their operation to accommodate the addition or deletion of network elements is missing.

To overcome these limitations, a centralized authority is added to the routing algorithm which maintains global information about the state of the network. This authority takes the form of a dedicated processor attached to the network with the responsibility to optimize the decentralized routing operation and adapt the route tables to topological alterations.

Centralized Route Table Optimization

The information contained in the local route tables of the nodes must be updated on occasion so that the route selection process can be made responsive to changing network conditions. This updating is essential to achieving optimum route selection and adaptive operation. In the hybrid routing strategy, a central manager (CM) is added to the network for the sole purpose of optimizing the route tables employed by the nodes. The optimization process relies on the ability of the central manager to obtain a global view of the network condition. To achieve this global view, the CM periodically collects network state data from each node. Although these state data represent only the local network perspective held by the individual nodes, the CM uses this set of local data to deduce global knowledge of the state of the network. In other words, the nodes provide pieces of the network puzzle, and the central manager assembles those pieces to construct an overall picture. Knowledge of the network condition allows the CM to compute enhanced route tables which it then transfers to the appropriate nodes. In an operational version of the information network, the central manager would be configured as a fault-tolerant processor with multiple connections to the net (fig. 7). This configuration alleviates the repercussions associated with a single-point failure at the CM.

As an example of the CM-node interaction for updating route tables, consider the case where a node goes down or is actually removed from the network. The neighboring nodes which are (or were) physically attached to the inoperative (or missing) node fail to receive a response or an acknowledgment when attempting to communicate with that network element. Each of the neighbors logs a failure of the output port connecting itself to the dead node. The central manager collects these fault reports while polling the nodes for network state data. Since none of the surrounding nodes could establish communications with the node in question, the central manager deduces that the node must be missing or inoperative. The CM then deletes the node from its list of active nodes in the net and initiates the execution of a shortest-path algorithm. This algorithm will determine the shortest route between every node pair in the network and must also explore alternate paths so that they may be ranked and included in the revised route tables. A shortest-path algorithm which accomplishes all the above for any configuration of the mesh topology is described in the next section of this paper. Having completed the route computations, the CM transfers the updated tables, which now reflect the absence of the failed or removed node, to the active nodes in the net.

The network response to a dead or broken link begins with the detection of that fault by the two node neighbors connected by the link. Both nodes report the failure of their output ports associated with the faulty link to the central manager. If this is the only fault in the network at that time, the failure can be handled effectively by the link avoidance capability of the nodes and no action other than recording the fault need be taken by the CM. However, after a second element failure occurs, the potential exists for establishing a path which does not represent the best choice in terms of minimum distance because the best choice for an alternate route which avoids a faulty link will be influenced by subsequent faults along the path. Since the nodes which are responsible for route selection lack the broad perspective necessary to anticipate subsequent faults, the central manager must intercede after a second link failure occurs. The CM will recompute network routes and alter the affected table entries in the nodes.

The hybrid routing algorithm must also adapt to the addition of a new node to the network in support of system evolution. When the new node is in place, its preliminary function is to issue an introductory message to all the nodes to which it is directly attached. The neighboring nodes report the presence of the new node to the CM. From these reports, the central manager determines the relative position of the node in the net, assigns it a unique address, and recomputes route information which reflects its presence and the availability of new routes.

Shortest-Path Algorithm

After analyzing the network state data supplied by the nodes, the central manager may conclude that sufficient topological changes have occurred to warrant recomputing the route tables. For this purpose, the CM initiates execution of a centralized algorithm which can be accurately classified as a shortest-path algorithm. Most existing computer networks employ some form of shortest-path algorithm for routing and a great deal of attention has been devoted toward their development in the past. The fundamentals on which shortest-path algorithms are based reside in the graph theory discipline. (See ref. 7.) In general, the objective is to determine the shortest path from all nodes v in the net to a node r, the result of which forms a tree T with root r. For example, figure 8 presents a rooted tree T whose root is node 1. Each node v has a unique path to the root r, node 1, and that path represents the shortest path from node v to r. Therefore, for example, the shortest path from node 4 to 1 is through node 2 and from node 6 to 1 the path is through node 3.

The concept of a "shortest path" requires further clarification. For some applications, the shortest path is that which provides the shortest physical distance from a specific node to the destination for the given network configuration. This is true for circuit-switched applications. The distance is measured in hops where the number of hops is equal to the number of nodes along the path including the root node. For example, the distance from node 5 to node 1 in figure 8 is 2 hops. The shortest path in a packet-switched network will not necessarily be the shortest physical route but rather the path that offers minimum message delay. The selection of the minimum-delay path requires a mechanism whereby the selection process is biased in favor of routes with lighter traffic loads. This mechanism generally involves assigning a cost factor to each network link which is a measure of the traffic load on the link. For virtual circuits, the cost factor would be directly proportional to the number of active, logical communication channels supported by a link. The shortest-path algorithm considers the individual link costs when exploring potential point-to-point routes and selects the path that offers the minimum aggregate cost.

An algorithm, described by Schwartz and Stern in reference 8, is a candidate approach to route table generation for the hybrid routing technique. This shortest-path algorithm is applicable to any network topology and has been implemented in various forms in existing computer networks (e.g., TYMNET (ref. 9)). It is an iterative procedure by which the central manager can compute shortest source to destination paths and, with slight modification, can derive a ranking of alternate routes.

Before a specific description of this algorithm can be given, it is necessary to introduce and define pertinent variables. Let r once again represent the destination or root node and v represent all other nodes in the net. Every node v will be assigned a label to keep track of the shortest paths during each iteration of the process. The label has the form [n,D(v)] where D(v) is the current iterate for the shortest distance from node v to r and n identifies the next node along that path. The distance D(v) can be expressed as the total number of hops to r or the aggregate cost associated with a path from v to r, depending upon the application. Finally, L(i,j) is the length of the link joining nodes i and j. If no link exists between nodes i and j, L(i,j) is equal to ∞. If a link exists, L(i,j) is equal to either 1 hop or, when cost factors are employed, it is the cost of the link joining nodes i and j.

The step-by-step procedure for computing the shortest path from all the nodes in the network, v, to a common destination, r, is as follows:

(1) Initialization: Set the distance from r to itself, D(r), equal to 0; label all other nodes, v, with the label $[\cdot,\infty]$ which means that the path from node v to r is currently undefined

(2) For each node v, update the label as follows:

For each neighbor of v, denoted as w, examine the current distance from w to r,D(w)

For every w of v let

$D'(v) = D(w) + L(v,w)$; i.e., the distance from v to r via a path through w is the sum of the distances from w to r and w to v

If $D'(v) < D(v)$, then set $D(v) = D'(v)$ and n = w; i.e., if the current iterate for D(v) is less than the previous shortest distance, relabel node v as $[w,D'(v)]$

(3) Repeat step 2 for every node v until there are no further label changes

This procedure can best be illustrated via an example. Figure 9 presents a six-node, irregular mesh network and the step-by-step results of the shortest-distance computations with node 6 as the root or common destination. The labels [n,D(v)] for nodes v, 1 through 5, are shown as a result of each step and all distances to node 6 are expressed in hops. Initially, the labels are all of the form $[\cdot,5]$, where the distance of 5 hops designates the lack of a path to the destination (total number of nodes in the net minus one). The iterative process is complete after step 5 since these results are identical with those of step 4.

From the information contained in the final set of labels, a rooted tree can be constructed (fig. 10(a)) which depicts the shortest paths to the root. In addition, route table entries for nodes 1 through 5 are presented in figure 10(b) which specify the next node along the shortest path to destination node 6.

For the purpose of evaluating this algorithm, it was coded in Pascal to run on a Digital Equipment VAX-11/750 computer. Each network node takes its turn as the root for the program computations to obtain a complete set of shortest-path information for any network configuration. All routes for the six-node network shown in figure 9 were computed in 20 ms on a VAX-11/750.

Since the decentralized route selection process of the hybrid technique requires alternate path information, a minor modification to the shortest-path algorithm is required. The central manager must maintain labels associated with alternate routes to the destination in addition to the primary shortest-distance label. This process involves recording the results of each D(w) + L(v,w) computation at every step of the process. Upon completion of the process, the CM sorts the resulting path information for each node v to obtain a ranking of paths. From the previous example (fig. 9), the possible paths from node 2 to 6 are through nodes 1, 3, 4, and 5. The distances associated with those paths are

w = 1; D(1) + L(2,1) = 4 hops
w = 3; D(3) + L(2,3) = 3 hops
w = 4; D(4) + L(2,4) = 2 hops
w = 5; D(5) + L(2,5) = 2 hops

Therefore, entry 6 of the route table for node 2 would look like this:

Destination address	Next node address for path ranking of —			
	1	2	3	4
6	5	4	3	1

Concluding Remarks

A hybrid routing strategy has been presented which combines both decentralized and centralized routing concepts. In this technique, the network nodes make independent routing decisions to establish communication paths through the network, avoiding busy and failed links. A central authority is also present, in the form of a dedicated processor attached to the net with the responsibility to optimize the decentralized path selection process. It does so by adapting the route information resident in the nodes to changes in topology caused by faults and/or data system evolution. The central manager periodically collects local state data from the nodes, derives a global perspective of the network condition from these data, and computes updated route tables which reflect the current condition of the network.

An experimental emulation network will be employed to evaluate the hybrid routing technique, as well as other control algorithms, for the purpose of determining their applicability to an integrated, fault-tolerant information network for Space Station applications. Since this emulation network will consist of twisted-pair links connecting microprocessor-based electronic nodes (as opposed to optical fibers and electro-optic nodes), the net will not support the high data rates required for fully integrated communications. It will, however, provide a relatively inexpensive test bed for performing various network experiments and for evaluating candidate control algorithms. Therefore, a relative "measurement of merit" can be established for the candidate approaches in terms of their amenability to high performance and fault tolerance.

Of initial concern is the establishment of physical circuits (i.e., circuit switching) in the experimental network to support the emulation of real-time video communications. The continuous, high-throughput nature of video data presents the major challenge to the design of the network architecture. Therefore, the hybrid routing technique will initially be evaluated in terms of its effectiveness in selecting routes for physical source to destination circuits. Algorithm characteristics such as path setup time, "shortest-path" computation time, and speed of response to topological alterations must be investigated in an actual network environment.

NASA Langley Research Center
Hampton, VA 23665-5225
May 1, 1986

References

1. Hendricks, Herbert D.; and Murray, Nicholas D.: Wavelength Division Multiplexing for Future Space Station Data Systems. *Fiber Optics in Local Area Networks,* Volume 434 of *Proceedings of the International Society for Optical Engineering,* Charles R. Husbands, ed., 1983, pp. 41–49.
2. Thurber, Kenneth J.; and Freeman, Harvey A., eds.: *Tutorial—Local Computer Networks,* Second ed. Institute of Electrical and Electronics Engineers, Inc., c.1981

3. Meisner, Norman B.; and Rosenthal, Robert, eds.: *Proceedings of the Local Area Communications Network Symposium*, U.S. National Bureau of Standards, May 1979. (Available from NTIS as PB80 152 903.)
4. Davies, D. W.; Barber, D. L. A.; Price, W. L.; and Solomonides, C. M.: *Computer Networks and Their Protocols* John Wiley & Sons Ltd., c.1979, ch. 2.
5. McQuillan, John M.; and Walden, David C.: The Arpa Network Design Decisions. *Comput. Netw.*, vol. 1, no. 5, Aug. 1977, pp. 243–289.
6. Martin, James: *Telecommunications and the Computer*, Second ed. Prentice-Hall, Inc., c.1976.
7. Hsieh, Wen-Ning; and Gitman, Israel: Routing Strategies in Computer Networks. *Computer*, vol. 17, no. 6, June 1984, pp. 46–56.
8. Schwartz, Mischa; and Stern, Thomas E.: Routing Techniques Used in Computer Communication Networks. *IEEE Trans. Commun.*, vol. COM-28, no. 4, Apr. 1980, pp. 539–552.
9. Rajaraman, A.: Routing in TYMNET. *Eurocomp 78—Proceedings of the European Computing Congress 1978*, Online Conferences, Ltd., 1978, pp. 9–21.

TABLE I. INFORMATION NETWORK ROUTING REQUIREMENTS

Select optimum source to destination communication path
- Shortest path
- Detect and avoid faulty network elements
- Minimize path setup time

Congestion control
- Avoid busy circuits (circuit switching)
- Traffic balancing (packet switching)

Adapt to topological alterations

TABLE II. ROUTING ALGORITHM CLASSIFICATION

Nonadaptive
- No attempt to adjust to changing net conditions
- Fixed or random routing

Centralized adaptive
- Central authority dictates routing decisions
- More near-optimal routing
- Routing control center can represent performance bottleneck

Isolated adaptive
- Independent operation
- Adaptability via exclusive use of local node data

Distributed adaptive
- Utilize internode cooperation
- Nodes exchange information to arrive at routing decisions

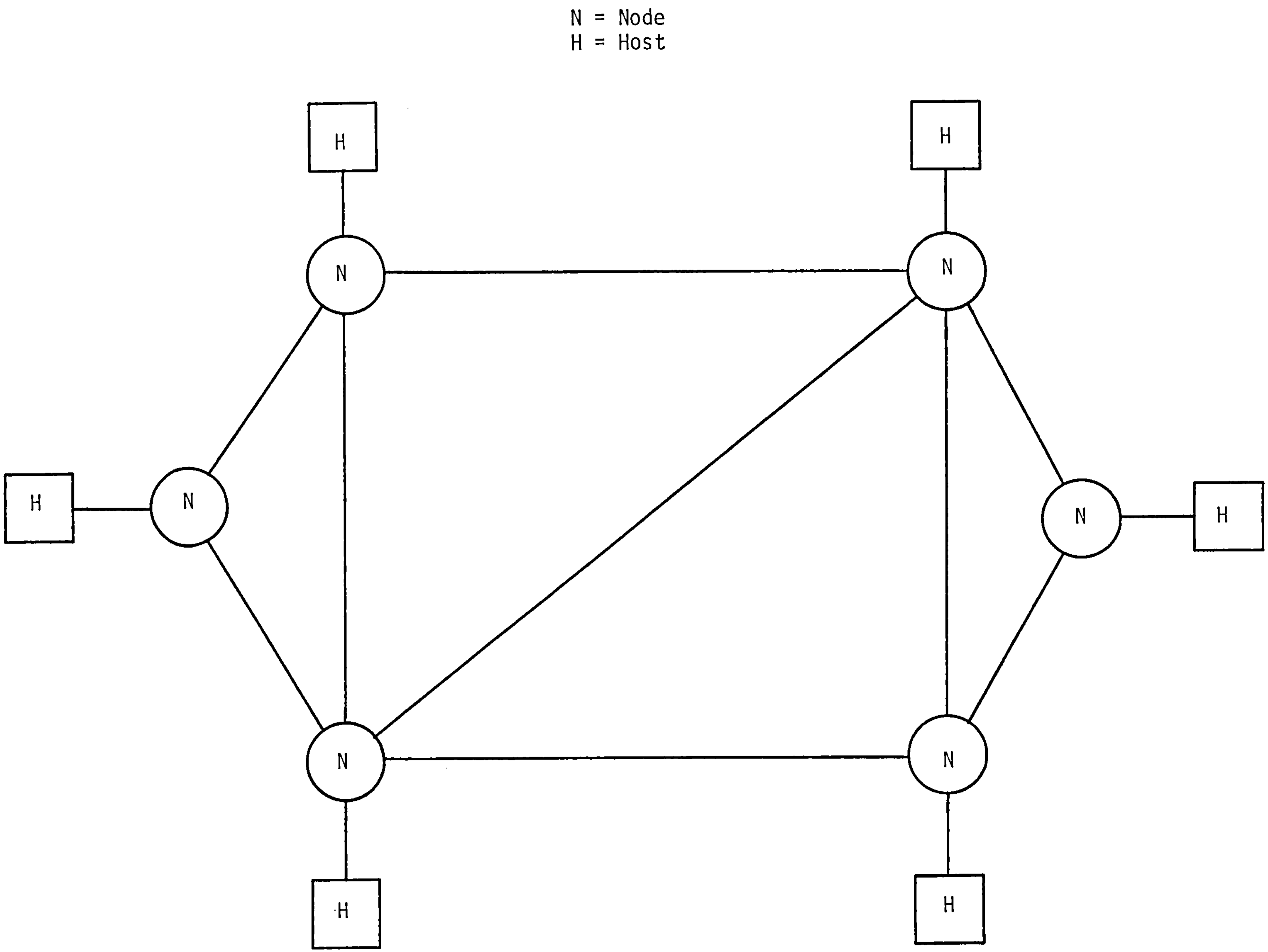


Figure 1. Mesh topology provides alternate paths for fault tolerance.

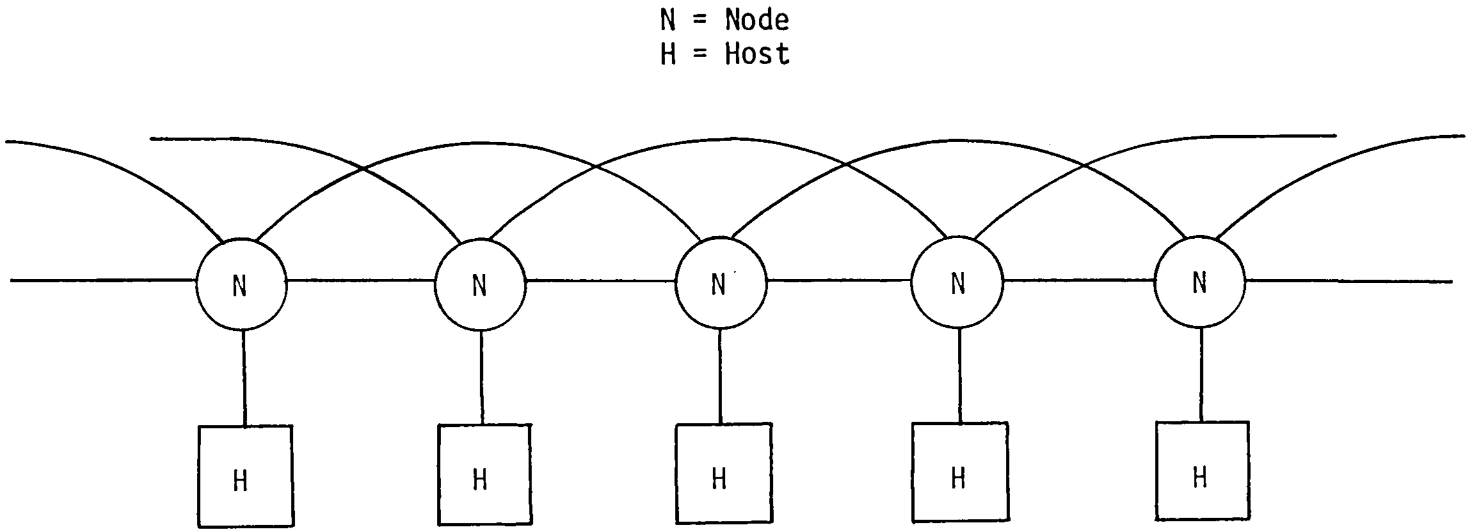


Figure 2. Braided mesh topology.

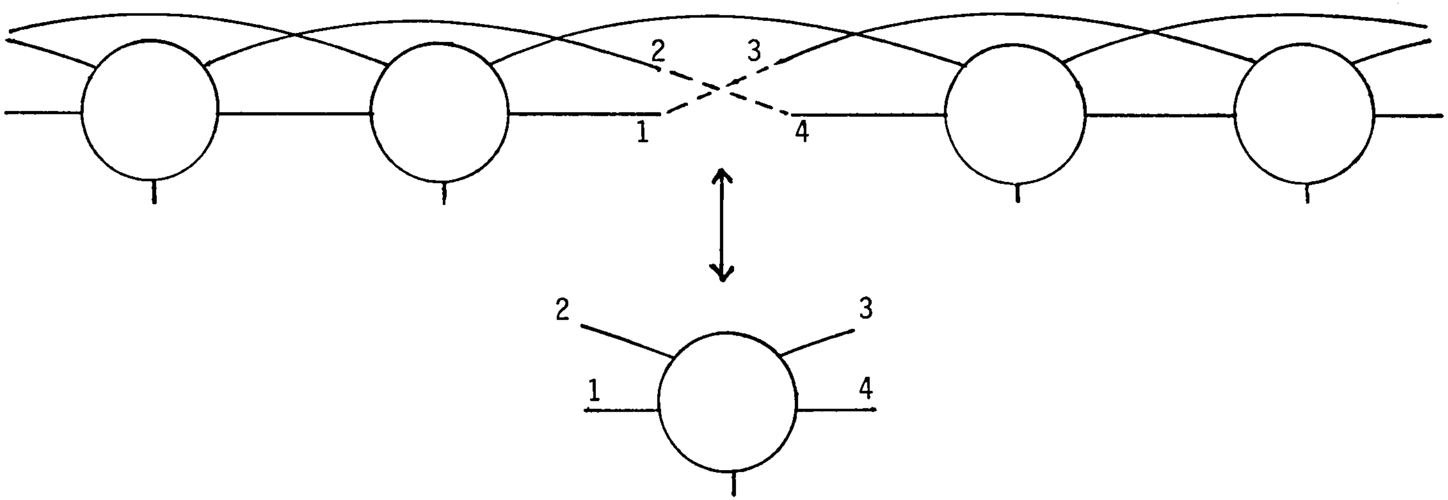


Figure 3. Regularity of braided mesh simplifies network modifications.

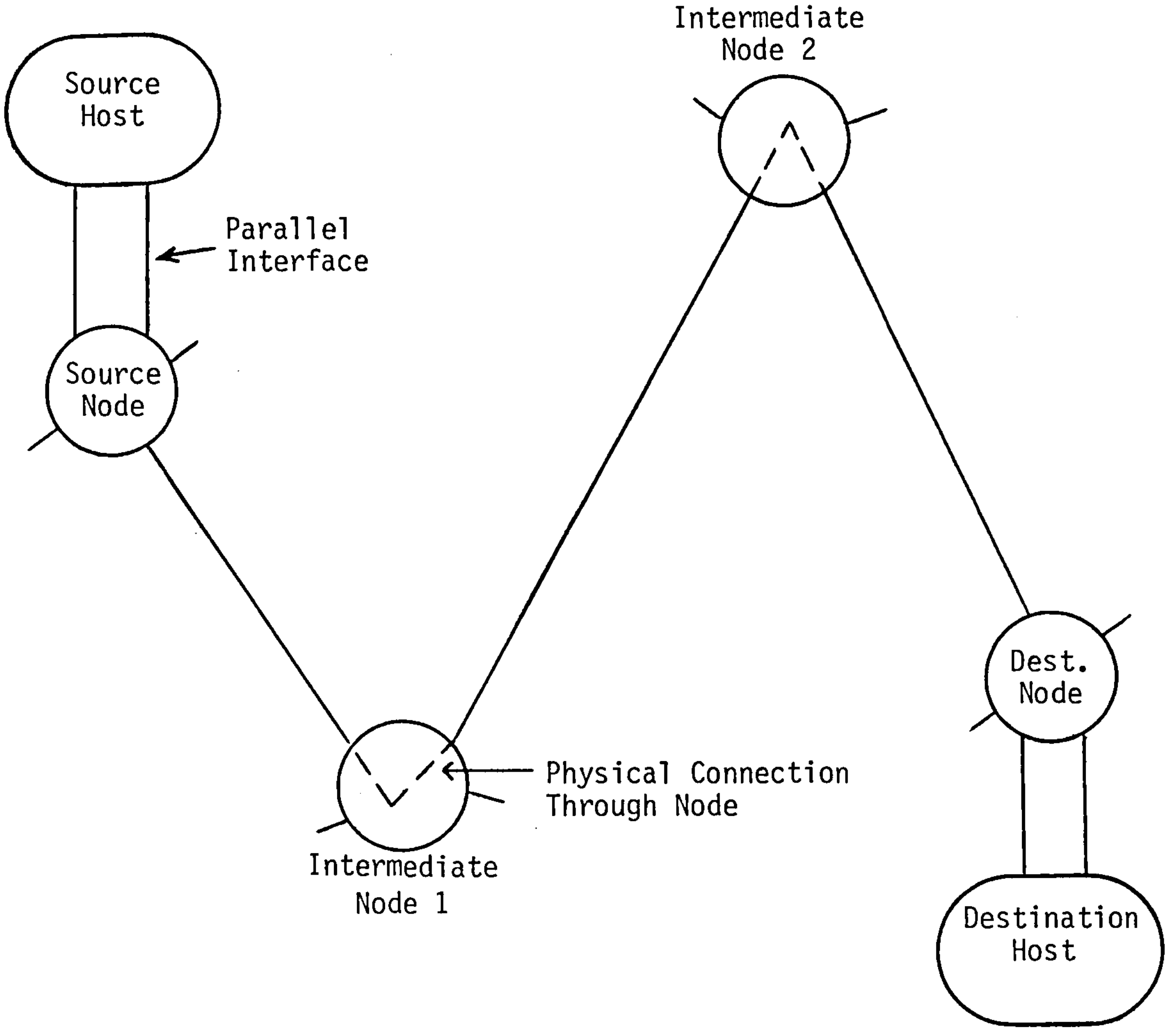


Figure 4. Circuit switching provides a physical path for host-to-host communications.

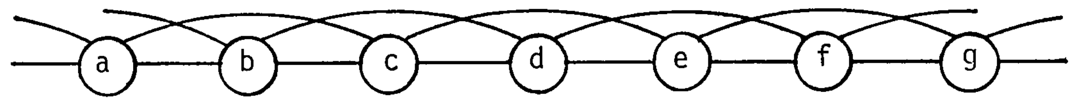
LOCAL ROUTING TABLE

DESTINATION ADDRESS	PORT RANKING			
	1ST	2D	3D	4TH
ADDRESS OF ALL OTHER NODES				

PORT STATUS TABLE

PORT	ALIVE	BUSY
1 2 3 4		

Figure 5. Table structure for route lookup and port status (four-port node).



Route table for node d

Destination	Port ranking			
	1st	2d	3d	4th
Node a	2	1	4	3
Node b	2	1	4	3
Node c	1	2	4	3
Node d	-	-	-	-
Node e	4	3	1	2
Node f	3	4	1	2
Node g	3	4	1	2

Port status table for node d

Port	Alive*	Busy
1	1	0
2	0	0
3	1	0
4	1	0

*1 denotes alive and busy.

Figure 6. Route and port status tables for node d of seven-node braided mesh network.

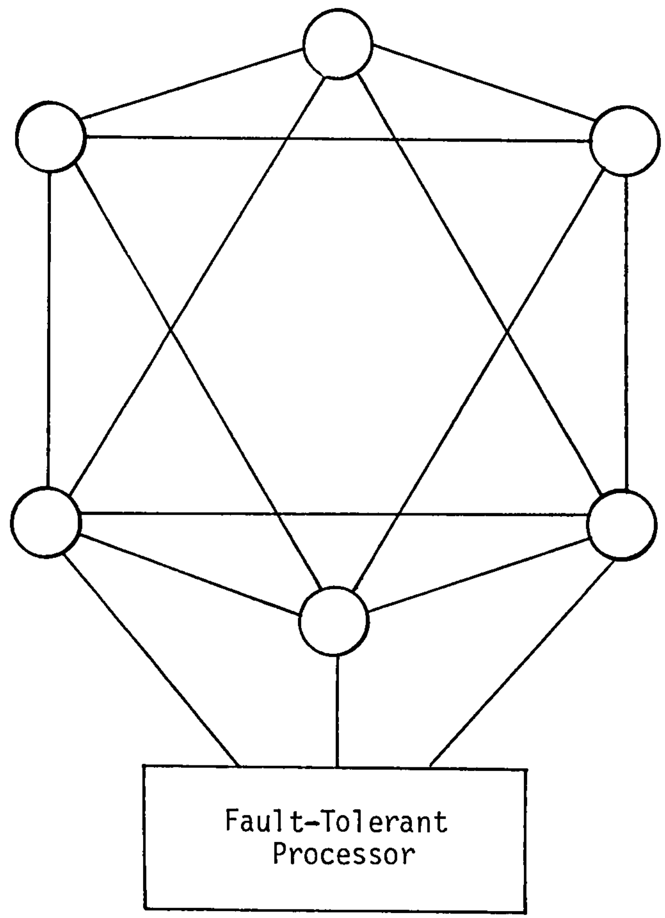
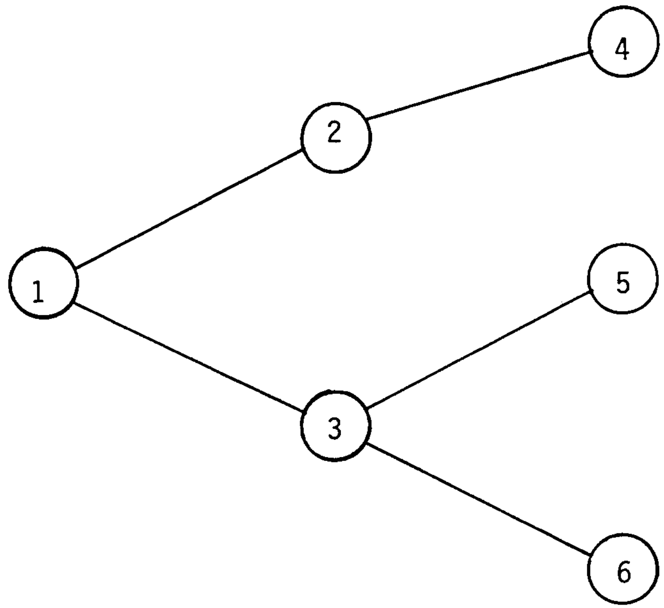
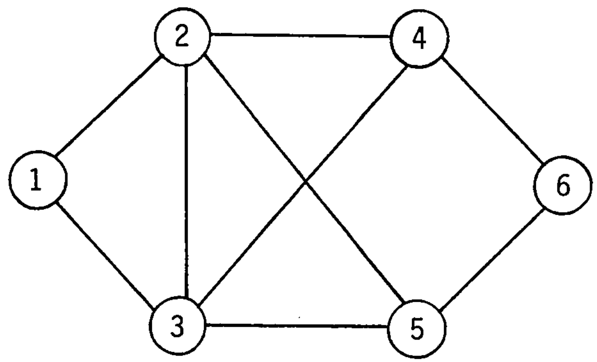


Figure 7. Fault-tolerant central manager with multiple connections to the network.



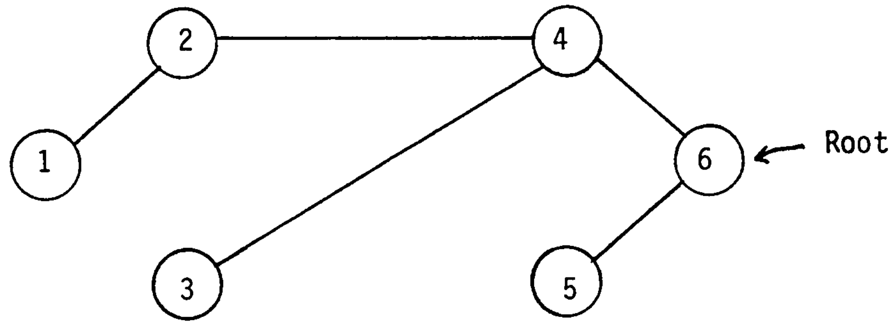
r = Node 1
v = Nodes 2, 3, 4, 5, and 6

Figure 8. Example of a rooted tree T.



Step:		Nodes, v = 1	2	3	4	5
1	D(6) = 0	[•,5]	[•,5]	[•,5]	[•,5]	[•,5]
2		[•,5]	[•,5]	[•,5]	[6,1]	[6,1]
3		[•,5]	[4,2]	[4,2]	[6,1]	[6,1]
4		[2,3]	[4,2]	[4,2]	[6,1]	[6,1]
5		[2,3]	[4,2]	[4,2]	[6,1]	[6,1]
Stop						

Figure 9. Example of shortest-path algorithm applied to six-node, irregular mesh network with node 6 as the common destination.



(a) Rooted tree resulting from shortest-distance computations of figure 9.

Route table for node	"Next node" entry for destination 6
1	2
2	4
3	4
4	6
5	6

(b) Next node along the shortest path to node 6 for route tables 1 through 5.

Figure 10. Results of shortest-path computations.

Standard Bibliographic Page

1. Report No. NASA TP-2605	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Hybrid Routing Technique for a Fault-Tolerant, Integrated Information Network		5. Report Date July 1986
		6. Performing Organization Code 506-58-13-02
7. Author(s) Barry D. Meredith		8. Performing Organization Report No. L-16034
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665-5225		10. Work Unit No.
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001		13. Type of Report and Period Covered Technical Paper
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

The evolutionary growth of the Space Station and the diverse activities onboard are expected to require a hierarchy of integrated, local area networks capable of supporting data, voice, and video communications. In addition, fault-tolerant network operation is necessary to protect communications between critical systems attached to the net and to relieve the valuable human resources onboard the Space Station of time-critical data system repair tasks. A key issue for the design of the fault-tolerant, integrated network is the development of a robust routing algorithm which dynamically selects the optimum communication paths through the net. A routing technique is described that adapts to topological changes in the network to support fault-tolerant operation and system evolvability.

17. Key Words (Suggested by Authors(s)) Integrated network Fault tolerance Mesh topology Routing algorithm Central manager		18. Distribution Statement Unclassified—Unlimited Subject Category 62	
19. Security Classif.(of this report) Unclassified	20. Security Classif.(of this page) Unclassified	21. No. of Pages 18	22. Price A01

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1986

National Aeronautics and
Space Administration
Code NIT-4

Washington, D.C.
20546-0001

Official Business
Penalty for Private Use, $300

BULK RATE
POSTAGE & FEES PAID
NASA
Permit No. G-27

POSTMASTER: **If Undeliverable (Section 158 Postal Manual) Do Not Return**