A SURVEY OF HIERARCHICAL ROUTING ALGORITHMS AND A NEW HIERARCHICAL HYBRID ADAPTIVE ROUTING ALGORITHM FOR LARGE SCALE COMPUTER COMMUNICATION NETWORKS

Fathy Amer and Yao-Nan Lien

Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210-1277
(614) 292-5236,{amer,lien}@cis.ohio-state.edu, cbosgd!osu-cis!{amer,lien}

ABSTRACT

This paper is concerned with a survey of present hierarchical routing algorithms for Large Scale Computer Communication Networks (LSCCN), with emphasis on a new proposed scheme, Hierarchical Hybrid Adaptive Routing Algorithm (HHARA).

Routing schemes of some currently implemented computer communication networks are reviewed and a short comparative survey for newly proposed hierarchical routing algorithms are given. Reliability, communication overheads, computation overheads are the major performance measures.

HHARA is proposed to reduce the size of routing database by dynamically organizing nodes into hierarchically structered clusters such that only partial information is stored and maintained in each site. Also, the responsibility of routing is shared by the routing hierarchy such that the algorithm can maintain the global routing optimality and the local adaptivity at the same time. In this way, the storage, maintenance, communication, and computation overheads can be reduced while the response time to local status changes is kept small. HHARA can be used with either datagram routing or virtual circuit routing. Due to its ability to adapt to the network changes, it is specially useful for dynamic networks such as large military computer communication networks.

This study leads to a conclusion that the hierarchical hybrid adaptive routing algorithm applied to a clustered network could be a convincing algorithm.

1. INTRODUCTION

Routing is one of the principal functions of a communication network in which a data unit (message or packet) is moved along a network path from the source node to the destination node. In the ISO-OSI protocol architecture, routing is the principal function of Layer 3, the Network Layer. A good routing algorithm is essential for the successful operation of a computer network and a poor routing algorithm could cause inefficient utilization of network resources and excessive delay for packets. This problem has been studied extensively since 1970's [11, 14]. The link cost definition and measurement, route generation, and packet (message) forwarding are the major problems in routing [2].

One of the major route generation strategies is *adaptive routing*, in which routes are determined according to the network status and are able to adapt to traffic variations and node or link failures. According to the place the adaptation is performed, they are divided into four basic schemes, centralized, isolated, distributed, and delta. [11].

For LSCCN, on the order of many hundreds or more, the size of routing database will grow quadratically. The storage required to contain this database at each node will be extremely costly. Also as a direct consequence of these large routing database, the cost of interchanging routing information among nodes will also grow and will represent a significant burden on the communication facilities. Furthermore, the time to adapt to the network changes, that includes the time to disseminate/collect the changes, the time to generate new routes, and the time to distribute new routes, may become too long to keep the adaptation in phase. All these considerations suggest that a routing scheme that can reduce the routing database without lose of optimality and adaptivity of the scheme. The problem is more serious in a dynamic network such as military computer communication networks, which must be very robust and very flexible [4].

To address above problems, researchers have proposed *hierarchical routing*, which divides a network into clusters and distributes the routing authority to different "levels" of nodes. Routes are generated by the routing hierarchy cooperatively. Each network node only maintains a subset of the complete database and uses that subset to perform adequate routing. Thus, the routing database maintenance overhead can be reduced, while the optimality of generated routes and the adaptivity to the lccal changes are preserved. Notice that packets are not necessary to be forward across the hierarchy in the hierarchical routing since such a routing scheme may be very poor. In this paper, a new hierarchical routing algorithm, Hierarchical Hybrid Adaptive Routing Algorithm, is proposed to reduce the database maintenance overhead. The method is achieved by organizing nodes in a network into multi-level structured clusters such that the distribution of routing responsibility may have a better match to the traffic flow locality.

The paper is organized as follows. Section 2 is concerned with the previous work for the currently implemented routing algorithms and newly proposed ones. Section 3 is concerned with the proposed Hierarchical Hybrid Adaptive Routing Algorithm. The routing algorithms are also qualitatively compared in Section 3. Finally, the conclusions and future work are presented in Section 4.

2. PREVIOUS WORK

This section briefly reviews the currently implemented computer networks and gives a short survey for the newly proposed hierarchical routing algorithms. A qualitative comparison of the routing algorithms in current networks is also presented.

2.1. Currently Implemented Computer Networks

ARPANET, TYMNET, SNA, DECNET, DATAPAC, and TELENET are reviewed in this section.

The original ARPANET routing algorithm was a distributed adaptive routing using estimated delay as the performance measure and a version of the backward-search shortest-path algorithm to find the shortest path between each node pair. The routing database update period was 2/3 second. The new ARPANET routing algorithm is also a distributed adaptive routing [12]. Each node in the network maintains a database describing the complete network topology and the delays on all links. Every 10 seconds, each node computes the average delay on each outgoing link and sends it to all other nodes using flooding. It is clear that the new ARPANET routing algorithm is still too costly in terms of storage, computation, and communication overheads for the networks larger than the current size.

Routing in TYMNET [15] is session-oriented and is centrally computed by a supervisor node. The supervisor node contains the complete network topology to compute the shortest paths. To prevent catastrophic failure of the network if a supervisor node should fail, TYMNET has a hierarchy of four supervisors nodes with only one being active at a time. If a supervisor node should fail, the existing routes are maintained while one of the dormant supervisor nodes takes control within 2 to 2.5 minutes.

There are two types of routes in IBM's System Network Architecture (SNA) [8] : explicit routes and virtual routes. An *explicit route* is the physical representation of a path and consists of an ordered sequence of links between the source and destination nodes. Route selection is performed locally at the source node (unlike TYMNET) at the beginning of a session. SNA routing algorithm is

CH2538-7/88/0000-0999 $1.00 © 1988 IEEE

IEEE International Conference On Communications '88, Philadelphia, PA, June 12-15, 1988

each node in the cluster prepares a section of the routing table for the nodes in the same cluster. It also receives a section from supervisors for all other nodes in the network.

Within the overall routing strategy established by the supervisor, further decisions could be delegated to the individual nodes or clusters which could react to the local status changes instantaneously. It is up to the node to make the final choice based on its own local information. Two kinds of information are available: (1) slightly outphased global information for the whole network, and (2) more accurate information for the cluster. These two kinds of information could be combined together for the best routing choice.

By this proposed routing procedure, the supervisor node could avoid the inefficiencies of looping while the distributed nodal or cluster portion permits instantaneous local adaptation. Thus, the proposed algorithm achieves hybrid routing in LSCCN and gives better performance than that achieved by either of the two limiting cases (centralized and distributed).

The underlining theme behind HHARA is that a strong locality on the network flow exists in some highly structured user communities such that it is better to have responsibility of local traffic routing on the local nodes to increase the adaptivity and to reduce the routing database maintenance overhead. Infrequent remote traffic is handled by higher level supervisor nodes that have better view on the network. One such example is the telephone switch system. It is obvious that most of phone calls to a local switch board are local. Another example is the network for army's command and control systems. The information flow in such a system usually matches the structure of the users (e.g. a troop,) which in turn matches the physical structure of the network topology. This spatial locality may not exist in some network like ARPANET although the temporal locality may exist.

3.1. Network Structure and Node Clustering Problem

Nodes in a large network are partitioned into many appropriate sized clusters. Each cluster contains up to N nodes (for any given $N \leq M$). All nodes belonging to a certain cluster have paths to all other nodes within that cluster through links belonging to that cluster. Each cluster has its own supervisor node, which corresponds to a centralized node in centralized or delta routing scheme. These supervisor nodes are again partitioned into some superclusters. The same procedure is applied repeatedly to form higher levels of superclusters until all clusters are in appropriate size. Nodes may dynamically leave or enter supervisor cluster or higher levels. The resulting partioning is called a *hierarchy*. Nodes are called clusters of level zero. The clusters that are formed the first time are called Clusters of Level one, CL(1), and the clusters formed at the nth application of the above procedure are called Clusters of Level n, CL(n). An example with 50 nodes is shown in Figure 1.

Each node in a cluster stores and maintains a local routing database such that it can adaptively generate routes to any node in the same cluster. The supervisor node at any level could generate routes between nodes within its descendents.

3.2. Overview of the Algorithm

Each node contains a routing table that records the outgoing links to forward packets to all other nodes. To reduce the routing table search time, it is divided into two sections, Cluster Section and Supervisor Section. The Cluster Section is always searched first since most of the incoming packets are expected to be local. Accordingly, the Supervisor Section will be searched last. The highest level supervisor nodes generate complete routing tables and distribute to their descendent nodes. Other nodes receive the routing tables from their parent nodes to fill the Supervisor Section. If there is a contradiction between the route generated by the Supervisor and the route generated by the node itself, the following tie-breaking rule is used: (1) if the cost of local generated route is lower than that of supervisor dictated route, take the local generated route; (2) otherwise, take the supervisor generated route. A supervisor node can force its subordinate nodes to take a particular route (e.g. to avoid looping) by lowering the route delay to zero.

Under some conditions, subordinate nodes may want to take the route generated by itself rather than the route generated by supervisor nodes even if its own route is more expensive. For example, the path generated by the subordinate node may be based on updated information, while the route generated by the supervisor nodes may be based on obsolete information. However, it is not an easy job to develop the "optimal" protocol to make the best decision. In HHARA, these conditions are simply ignored. The subordinate nodes will take the supervisor dictated route. It is conservative, but the update information will eventually reach the supervisor nodes and a better route can be dictated by the supervisor nodes later. More progressive protocols will be developed in the future.

The correct operation of the proposed algorithm assumes the existence of a link-level protocol that assures that [5]:

- Every node knows its neighbors inside the cluster and its supervisor(s).
- All packets forwarded over a link are received correctly and in the proper sequence.
- All update packets, link-failure information, and new neighbor information are processed one at a time and in the order in which the updates are received in each node.
- Each node keeps the routing information for those nodes that become unreachable.

Packet Forwarding

The proposed HHARA can be used with datagram routing or virtual circuit routing but with a slight modification during implementation. In the case of datagram routing, each packet is sent individually to its destination. Each node selects the next node to which the packet should be sent based on its Routing Table. In virtual circuit routing, the ROUTE-SETUP packet is forwarded in the same way as the packet forwarding in datagram routing. The routing is fixed for its session from source to destination. Details of session routing by the proposed algorithm will be done in the following future work.

3.3. Structure of Routing Database

Each node has its own limited database describing the topology and the link delays of the cluster it located. Using this local database each node independently calculates the best paths to all nodes in the cluster. In addition to the node routing database, there is a supervisor routing database, which maintains full routing information about its subordinates.

3.3.1. Delay and Routing Tables

Each node (either regular node or supervisor nodes) will have the following two tables:

(a). **HHARA Delay Table:**

which is a full $m \times m$ link delay table where m in the number of nodes in the cluster. Each entry (i,j) represents the delay on the link connecting node i to node j. Note that the entries (i,j) and (j,i) are usually different since the delay along the link from node i to j could be different from the delay of the link from node j to i. Table 1 illustrates the Delay Table at cluster 1.2, which has source nodes delay entries.

The cluster delay may differ from one cluster to another depending on the number of its nodes, and supervisor nodes. Clusters which have no supervisor nodes will have an infinity in the corresponding entries.

(b). **HHARA Hierarchy Routing Table:**

Each node has a routing table having two sections of routing entries:

Cluster Section:	routing entries for nodes inside the cluster.
Supervisor Section:	routing entries for all other nodes in the network which is dictated by the supervisor and are called supervisor routing entries.

Table 1 Qualitative comparison evaluation of HHARA with other schemes.

Comparison Points	Proposed Scheme HHARA	New ARPANET Routing Algorithm	Kamoun Scheme HRA
Database	routing information of the nodes in the same cluster and closest neighbor nodes	complete network routing information	complete routing information about the nodes close to the node and lesser information about the remote nodes
Clustering	uses optimal clustering procedures	no clustering	uses heuristic clustering procedures
Routing Information Updates	inside each cluster, neighbors and between each supervisor and its nodes	among all nodes	between neighbors and between special nodes in different clusters at different levels
Updating Period	supervisor update: as in new ARPANET; cluster update (parameter): 1/2 to 1/3 of supervisor update	periodically every 10 sec. or due to sharp or sudden change in traffic or network topology	every 128 msec., as in original ARPANET routing algorithm
Updating Packet Length	one update packet from each node to all cluster members and to supervisor	176 bits on an average for each node	16 bit word for each node, its size depends on the number of nodes in the network
Updating Packet Priority	highest priority	highest priority	highest priority
Routing Table	one entry per node in every node, divided into cluster section and supervisor section.	one entry per node in every node	a function of the clustering structure - one entry per node in each cluster and one entry per cluster inside its supercluster or other supercluster(s) for the remote nodes
Delay Tables	cluster one is for local nodes inside the cluster and supervisor one is for its descendents	for all nodes in the network; will increase linearly as the size of the network	for all neighboring nodes
Route Length	is optimal because it is guided	is quite long in large networks due to several intermediate links between source and destination	is much longer than the proposed one in higher level hierarchies
Communication Overheads	lowest, because there is no routing update information between clusters; routing information exchanged only between supervisors and its nodes in the corresponding cluster	high, due to more routing information required to be passed to all nodes and update packets occupying a very significant portion of the communication bandwidth	in between, because there is more routing information passed to the "exchange nodes" in different clusters at different levels
Computation Overheads, Memory inside each node	less, due to limited local routing database in each node	large, due to large routing information database for the whole network	in between, because each node maintains routing information for its neighbors and higher level "exchange nodes"
Processing Overheads	less, due to the limited local database in each node	large, due to the large database in each node	in between, due to medium database in each node
Routing Orientation	Topology	Topology	Shortest path
Partitioning Procedure	identified	no clustering	not addressed

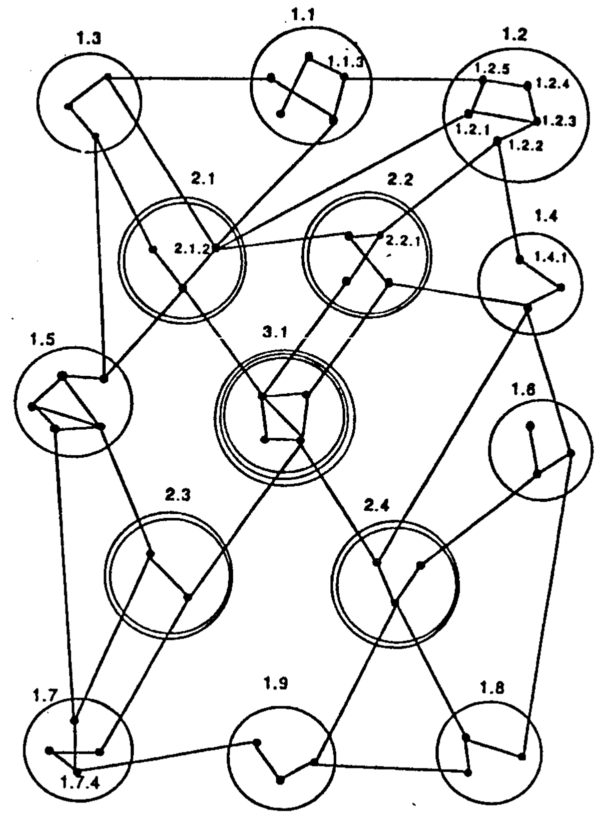

Figure 1 HHARA clustering example for a 50 node network.

more concerned with reliability than with packet delay and throughput.

DECNET and DATAPAC [7] networks are discussed together because their routing is essentially identical. Routing in these two networks is of distributed type and is datagram-oriented, with link costs being based only on the bandwidth of each line. Since the bandwidth for each line is fixed the routing will only change when there is a change in the network topology.

Routing strategies in TELENET [7] is session-oriented and consists of two stages: one or more Network Control Center generates a routing table for each node at network configuration time, and downloads this information through the network at the second phase. The routing table for a given node defines, for each destination, primary and secondary outbound links.

The qualitative comparisons for the above networks will be concentrated on link cost computation, route generation and update and load balancing [7].

Both ARPANET and TYMNET take into account packet delay in computing link costs. The network nodes are responsible for collecting this information and forwarding it to the adjacent nodes in the case of ARPANET, or the supervisor node in the case of TYMNET. The link cost function in DECNET and DATAPAC is a function of link capacity only. Since link capacity changes infrequently (only with topology changes) the processing overhead incurred is low. Routing in SNA is performed only at network definition time. Thus there is no processing overhead for link cost computation during normal operation.

In ARPANET, the link cost update packets are broadcast from each node to every other node every 10-60 seconds. This along with the resulting shortest path computation at each node, results in a fairly high level of processing overhead. With TYMNET the link cost update packets are transmitted only to the supervisor nodes every 16 or more seconds. Also, the route selection is session-oriented, consequently the processing overhead is lower than that in ARPANET. Route selection in TELENET involves higher processing overhead than TYMNET does because each node must select the least saturated outbound links and check for looping. Since SNA route generation is performed at network definition time, the only overhead incurred during normal operation is the activation and deactivation of virtual routes and explicit routes. This processing overhead is minimal. Routing changes in DECNET and DATAPAC only when link bandwidth changes, that is, when links change operational status. Since this is done infrequently, the resulting overhead is low.

Both ARPANET and TYMNET adapt to the saturation levels of their links. In TELENET, the selection of outbound links is performed locally based on the estimation of the saturation level of those outbound links. With DECNET, DATAPAC, and SNA networks, link loading information is not used in route selection. Hence there is no load balancing mechanism.

Details of comparison evaluation could be found in [7].

2.2. Newly Proposed Hierarchical Routing Algorithms

2.2.1. Kleinrock and Kamoun's Routing Scheme

Kleinrock and Kamoun [10] have proposed M-level hierarchical routing algorithm. Their main contribution is that, as the number of network nodes becomes very large, the additional average delay suffered by the packets is negligible compared to nonhierarchical routing case. The particular numerical examples show that the transition point where hierarchical routing becomes certainly better than a nonhierarchical one occurs for a network size between 100 and 200 nodes. They also developed approaches for the optimal hierarchical clustering to reduce the table length from N to $\ln N$, where N is the number of nodes.

Each node maintains a shortest-path table indicating the best path to nodes in the same cluster, and one entry per set of destinations for the remote nodes.

2.2.2. Baratz and Jaffe's Routing Scheme

Baratz and Jaffe [2] address the problems of route performance and network partition with a single concept. By keeping more information than Kleinrock and Kamoun, they develop a scheme that can determine optimal paths independent of the network partitioning. Due to the overhead at path determination time, their technique is only applicable to virtual circuit networks like TYMNET, or SNA, and not datagram networks like ARPANET [9].

2.2.3. Jaffe's Routing Scheme

In Jaffe's routing scheme [9], a set of nodes are designated as the *backbone nodes*, which carry more routing responsibility than other nodes, which are referred to as *internal nodes*. Nodes are also partitioned into clusters. Every cluster in the network must have at least one node in the backbone. For each cluster, it is generally good to have more than one node in the backbone for higher reliability. The backbone network has a backbone database to provide optimal routing between frequent communicators. The combination of a cluster database, and a backbone, allows reasonable routing to all nodes.

2.2.4. Chlamtac and Elazar's Error Bounded Routing Scheme

Chlamtac and Elazar [3] proposed a hierarchical routing algorithm with a bounded error, which is difference between an actual path and its optimal path. They found that there is a correlation between the clustering and the routing error in hierarchical networks such that an arbitrary network partitioning may lead to significant routing errors. The network designer could choose the number of cluster levels and the routing error size according to the given requirements.

2.2.5. Ramamoorthy and Tsai's Adaptive Hierarchical Routing Algorithm

Ramamoorthy and Tsai [13] proposed an adaptive hierarchical routing algorithm for Battle-field Information Distributed System. The system consists of a set of large, ground based, packet radio units. The topology of the network changes dynamically since network elements are mobile. However, the authors solved the problem by assuming fixed-locations. The basic idea of the proposed scheme is to extend the new ARPANET routing algorithm to hierarchical network. Each node keeps the complete topology information of the nodes in the first level cluster like the new ARPANET routing algorithm. However, each node will have summary information of other clusters.

2.2.6. Houstis and Leon's Routing Scheme

Houstis and Leon [6] have developed an adaptive routing strategy for large store-and-forward message switched networks where each message is transmitted regardless of length. The work is related to AUTODIN network of the Defense Communication System. The measure of performance is the average message delay, which includes transmission delay, queuing delay, propagation delay, and node processing delay. The algorithm used is a modification of adaptive algorithm used in ARPANET, but is working for message switching networks.

3. PROPOSED SCHEME

The Proposed hierarchical hybrid adaptive routing algorithm organizes the network nodes into a multi-level hierarchy and distributes the routing authority to the nodes in different levels. In this way, the proposed scheme is aimed to combine the strengths of centralized adaptive and distributed adaptive routing algorithms. The centralized adaptation is done by the supervisor nodes in different levels to achieve the global optimality, while the distributed adaptation is done by the lower level supervisors and nodes to retain the local adaptivity. The nodes and lower level supervisors report to higher level supervisors the current anticipated delays periodically and the critical situations such as node/link failure or sharp traffic variations immediately. Based on this information, each supervisor node updates its own routing information database. After that, it prepares the overall routing strategy and sends to all subordinate nodes or clusters. At the same time, each cluster performs its routing function in a distributed adaptive fashion (as in the new ARPANET routing algorithm) only for the routes to the cluster itself. Hence,

3.3.2. Routing Database update

In order to react to failures, congestion, or a planned or non-planned topology change, and traffic variations, the routing data-bases are updated automatically.

There are two kinds of updates: the cluster update and the supervisor update. The cluster (at any level) update is done periodically or as soon as a node (or a supervisor) detects sufficient changes in one of its outgoing links. It works exactly like the new ARPANET update protocol but for nodes inside the cluster [12]. The update information is carried by the small Routing-update packets, which are forward with the highest priority to all other nodes in the cluster.

The supervisor update is broadcast by the supervisor at any level periodically or when the following two conditions are detected: (1) sufficient changes in the subordinate nodes reported information, and (2) changes received from other supervisors. Supervisor node regulates the exchange of update information between the cluster supervisor and its nodes.

In case of link failure to supervisor, the node will send the routing information to the alternate one. For whole failure of supervisors, node will communicate (forward or get) its routing information with its neighbors in neighboring clusters. So each node knows routing information about neighboring nodes in neighboring clusters.

The following topology changes in dynamic computer networks are considered in the proposed scheme: (1) link failure and its repair, (2) node failure and its repair, (3) cluster failure and its repair, and (4) locations of nodes changes. Node failure may be considered as failure of all links associated with the node, and node repair may be viewed as repaired of links associated with the just activated node. In practice, most of update problems are only in the case of link failure and link repair. For cluster failure and repair, they usually require reclustering of the network. In this paper we will not consider cluster and node failure and repair.

3.4. Evaluation of The Proposed Routing Scheme: HHARA

In this section, the proposed scheme is compared with other hierarchical routing algorithms and the new ARPANET routing algorithm. The new ARPANET routing algorithm is chosen as an example of an efficient nonhierarchical adaptive routing algorithm. It has been used in a real network satisfactorily and is also widely referenced. Kamoun's scheme is chosen as the representative of hierarchical routing algorithm in a clustered network. Some of the algorithms surveyed above are not compared here because they either work for static networks only or because they are very similar to the ones being compared. A fixed routing algorithm, the new ARPANET routing algorithm, and HHARA are compared using simulation experiments on a 50-node simulated network. Details of simulation could be found in [1]. Table 1 illustrates the qualitative comparison between the proposed scheme, the new ARPANET routing algorithm and and the Kamoun's scheme for large hierarchical networks.

4. CONCLUSIONS AND FUTURE WORK

In this paper the importance of using hierarchical hybrid adaptive routing algorithm for large scale computer communication networks is addressed. It has the advantage of reducing the overhead of routing database maintenance while retaining the adaptivity to the network status changes.

The proposed algorithm is based on the modification of delta routing algorithm and follows the routing-update procedures of ARPANET. It makes use of local routing information as well as the global routing information. It appears to be a promising algorithm for large dynamic networks due to the distribution of routing responsibility to the network hierarchy that has better utilization of traffic flow locality.

The proposed HHARA can be used with datagram routing or virtual circuit routing. Due to its ability to adapt to the network changes, it is specially useful for the dynamic networks such as large military communication networks.

The hierarchical routing would be a very interresting area of research related to large scale computer communication networks. Most of the recent research in this area is limited to two-level hierarchical routing. The classification of hierarchical routing algorithms is at present a mandatory task, since the research in this area is relatively new.

Many aspects need to be studied. Among them, a comparative study of various hierarchical routing algorithms, under different network conditions seems to be the most useful for the designers and planners of future large scale computer communication networks.

References

1. Amer, Fathy and Yao-Nan Lien, "Performance Evaluation of A Hierarchical Hybrid Adaptive Routing Algorithm for Large Computer Communication Networks," *Proc. of Computer Networking Symposium*, Washington D.C., USA, April 11-13, 1988.
2. Baratz, A. and J. Jaffe, "Establishing Virtual Circuits in Large Computer Networks," *Proc. of INFOCOM 83*, San Diego, California, April 1983.
3. Chlamtac, Imrich and Avishai Elazar, "Hierarchical Routing with Bounded Routing Error," *Computer Network Symposium*, pp. 196-202, 1986.
4. Frankel, Michael S., "Telecommunications and Processing for Military Command and Control: Meeting User Needs in the Twenty-First Century, On Emerging Technologies to support Distributed Survivable Command and Control," *IEEE Comm. Mag.*, vol. 22, No. 7, pp. 18-25, July 1984.
5. Garcia-Luna-Aceves, J. Joaquin, "A New Minimum-hop Routing Algorithm," *Proceedings of IEEE INFOCOM'87*, pp. 170-180, San Francisco, March 1987.
6. Houstic, C. E. and B. J. Leon, "An Adaptive Routing Algorithm for Large Store-and-forward Computer Communication Networks," *Proceedings of National Telecomm Conf.*, pp. 28:5-1 - 28:5-6, Los Angeles, CA, Dec. 1977.
7. Hsieh, Wen-Ning and Israel Gitman, "Routing Strategies in Computer Networks," *IEEE Computer*, vol. 17, No. 6, pp. 46-56, June 1984.
8. IBM., "Systems Network Architecture: Concepts and Products," *IBM GC30-3072*, 1981.
9. Jaffe, J. M., "Hierarchical Clustering by Designing a Backbone," *Proc. of 4th Convention of IEEE*, pp. 2.2.1/1-4, Tel Aviv, Isreal, March 1985.
10. Kamoun, Farouk and Leonard Kleinrock, "Stochastic Performance Evaluation of Hierarchical Routing for Large Network," *Computer Networks*, vol. 3, pp. 337-353, 1979.
11. McQuillan, John M., "Routing Algorithms for Computer Network: A Survey," *Proc. Nat Telecomm. Conf. NTC 77, Conf. Rec.*, pp. 28:1-1 - 28:1-5, Cambridge, MA, Dec. 1977.
12. McQuillan, John M., Ira Richer, and Eric C. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Trans. of Comm.*, vol. Comm-28, no. 5, pp. 711-719, May 1980.
13. Ramamoorthy, C. V. and W. T. Tsai, "An Adaptive Hierarchical Routing Algorithm," *Proceedings of 7th COMPSAC*, pp. 93-104, 1983.
14. Schwartz, Mischa and Thomas E. Stern, "Routing Techniques Used in Computer Communication Networks," *IEEE Trans. on Comm.*, vol. 28 No. 4, pp. 539-552, April 1980.
15. Tymes, L., "Routing and Flow Control in TYMNET," *IEEE Trans. on Comm.*, April 1981.

IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS '88

DIGITAL TECHNOLOGY...

SPANNING THE UNIVERSE

PHILADELPHIA, PA

JUNE 12-15, 1988

WYNDHAM FRANKLIN PLAZA HOTEL

Conference Record **Volume 2 of 3**

Volume	Day	Sessions	Pages
1	Monday	1 - 17	1 - 568
2	Tuesday	19 - 36	569 - 1200
3	Wednesday	37 - 54	1201 - 1783

Sponsored by the IEEE Communications Society
and the
Philadelphia Section IEEE

IEEE COMSOC IEEE ICC GLOBECOM