*1988 Computer Networking Symposium*, April 1988, pp. 266-275.

# Performance Evaluation of A Hierarchical Hybrid Adaptive Routing Algorithm for Large Scale Computer Communication Networks

*Fathy Amer, Yao-Nan Lien, and Ahmed Ghieth*

Department of Computer and Information Science

The Ohio State University

Columbus, OH 43210-1277

(614) 292-5236,{amer,lien,ghieth}@cis.ohio-state.edu

## ABSTRACT

A new Hierarchical Hybrid Adaptive Routing Algorithm (HHARA) is presented for dynamic Large Scale Computer Communication Networks (LSCCN). The performance of the proposed algorithm is evaluated and compared to non-hierarchical routing algorithms by simulation experiments on a 50 node network model. The major evaluation criteria are reliability, communication overhead, computation overhead, and average packet delay.

A fixed routing algorithm, the new ARPANET routing algorithm, and HHARA are compared using simulation experiments. The average packet delay of HHARA is found shorter than that of ARPANET routing algorithm in general due to the reduction of generated number of update packets in HHARA. Further, it maintains the global routing capability and the local adaptivity at the same time. HHARA performs much better when the network is unreliable or dynamically changed. As a comparison, the ratio of lost packets in fixed routing algorithm is very significant although the average packet delay of delivered packets is lower. The storage, the communication and computation overheads for routing database maintenance are smaller. Finally, the reliability is much higher which is specially useful for military communication network.

This simulation study leads to a conclusion that HHARA makes a good balance between the reduction of routing database maintenance overhead, and the global routing capability as well as the local adaptivity to the network changes.

## 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 [21,28,26,31]. The link cost definition and measurement, route generation, and packet (message) forwarding are the major problems in routing.

The *link cost functions* used in routing algorithms for computer networks are either (1) a function of the capacity of the channel that constitutes the link, or (2) a function of the channel utilization or equivalently, average packet delay. The current commonly used definition of link cost is the average packet delay experienced in the routing over the given route, computed over a period of time.

*Route generation* is the function to determine the path that a packet is forwarded from its source to the destination. The performance criteria, the least cost routing algorithm, the places to execute the algorithm and the time to execute the algorithm are some of the mandatory decisions to be taken into consideration in route generation.

The main objectives of a routing algorithm design are correctness, computational simplicity, adaptivity to the network changes, stability , and fairness [29]. Based on Dijkstra's algorithm, a number of shortest-path algorithms have been developed for packet routing. The shortest-path problem satisfies the the principle of optimality, so the route from a node to any destination is *path insensitive*. That is to say, each node needs only to know which neighboring node to forward a packet to a destination.

Most routing algorithms have nodes in the network to route packets according to a precalculated *routing table* to forward packets. Depending on whether the table is updated by the algorithm itself to adapt to the network changes or not, these algorithms can be classified into *fixed* and *adaptive*. The generated routes are not changed in a *fixed routing*, which is more attractive to use at the design phase and a relative stable network. The routes may be changed according to the network status in an adaptive routing, which is better in network operation as the network status changes with time. The major goal of *adaptive routing* policies is to sense status changes in the network and to route packets based on more current information. As a consequence, they adjust to load fluctuations and node or link failures.

The major design issues for adaptation and updating are : Objective identification, the place to perform the adaptation, type of adaptation, Updating frequency, and techniques to reduce routing overhead. Valuable surveys on various types of routing algorithms and the major design issues for adaptation and updating could be found in [24,27].

According to the place the adaptation is performed, they are divided into basic schemes, centralized, isolated, distributed, and delta.

In *centralized routing*, routing tables are generated by a central controller. *Centralized routing is able to generate optimal routes, to avoid* inconsistent route generations that might creat endless loops. On the other hand, it is vulnerable to the central controller crash. Also, the status acquisition and the routing table distribution present a substantial communication overhead.

In *isolated routing*, each node adapts to the network changes based only on the its own local information. In *distributed routing*, each node collects and exchanges network information with its adjacent neighbors to make routing decisions.

*Delta routing* is a hybrid algorithm that splits the routing responsibility between a central node and each individual node. In this way, it combines centralized routing with isolated routing to take advantage of global network planning capability of centralized routing and the adaptivity of isolated routing [24].

In general, centralized routing schemes are more efficient in the long term, giving stable traffic flows. Distributed routing schemes tend to be efficient only in the environment local to that node making decisions. On the other hand, distributed techniques allow a node to respond very rapidly to the change in traffic or resources in its own immediate environment.

for the cluster. These two kinds of information could be combined together for the best advantage.

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.

Using such a hierarchical routing, the overhead of routing database maintenance can be reduced while the requirement of routing requests can be fulfilled with only a few exceptions. Consequently, the proposed hierarchical hybrid scheme would achieve better performance than centralized or distributed routing schemes in other hierarchical routing algorithms.

### 3.1. Network Structure and Node Clustering Problem

LSCCNs are implemented in practice by intelligent devices (controllers, channels, concentrators, .....etc.) that are connected together through some kind of a communication medium (telephone lines, direct connection, microwave links, .....,etc.). These devices are called *nodes*. The communication line connecting two nodes is called a *link*. A network is a collection of interconnected nodes such that there is a single link connecting any two nodes and there are no links connecting a node to itself. Given any two nodes $n_i$ and $n_j$ in the network, a path exists, possibly through other units, that connects $n_i$ to $n_j$ over a series of links. An example with 50 nodes is shown in Figure 1.

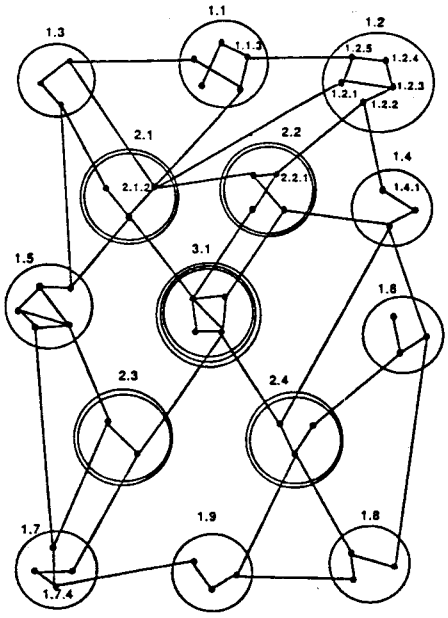



Figure 1. A 50 node clustered LSCCN for performing HHARA.

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 $n$th application of the above procedure are called Clusters of Level n, CL(n).

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 links to the nodes in the same cluster and descendent clusters are in the Cluster Section; and those to other nodes are in the 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 supervisor node in the proposed scheme plays the role of central node in centralized routing. In general, experience has shown that centralized routing schemes are more efficient in the long term, given stable traffic flows [24]. This is because a single entity with a global overview could prepare a consistent strategy by taking into account the global network status. The decisions made at the nodes in distributed routing schemes tends to be efficient only in the environment local to the node making the decision. This routing decision may not be globally optimal, and may generate endless loops. Looping has been a problem with the ARPANET which uses a distributed routing strategy. The looping problem is at its worst when the network can least afford it, i.e., in periods of heavy load.

On the other side, the distributed techniques allow a node to respond very rapidly to changes in network topology and traffic in its own immediate environment. In the proposed algorithm, each node follows the supervisor routing strategy unless changes occur inside its cluster. So each node is free to respond to the instantaneous flow of packets in its local cluster environment. The updating procedures follow two kinds of separated updates, one for the entire cluster-neighbor update and

### 3.4. How does the proposed algorithm solve the problems of LSCCN?

Due to the inevitable propagation delay and the random nature of the network configuration, there are three problems need to be taken into consideration in the updating process [12].

(a) The problem of "aging." Since delay information is exchanged periodically between nodes, it takes time for measured delay to propagate through the network. Thus the choice of the route to distant nodes are based on data that several update periods old. In a large network, aging will adversely affect routing decisions no matter how short the update period is.

(b) The measured delay at update may be unrepresentative in some fashion and the correction of the delay table information made accordingly may be very nonoptimal over the next update period. That could cause a deciding node to misjudge the distant nodes.

(c) In the network, there exist reliable and unreliable routes for the packet to be forwarded. This means, some routes consisting of links where the delay changes more slowly, on the average, and others where the delay fluctuate all over the place.

The proposed algorithm will "smooth out" these problems as follows.

(a). For the aging problem: the network is dynamically partitioned into hierarchically structured 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. Then from each cluster it is sent one piece of information that represents the cluster instead of sending information from every one of these cluster nodes to all other nodes.

(b) For the problem of "unrepresentative delay measure", the proposed algorithm uses the following representative value of delay instead of the current delay (or its average ):

(average delay of all packets across the link + the delay of the last period) / 2

(c) For the problem of uneven link reliability, the proposed algorithm adjust the representative delay value with different weights proportional to their variance. Thus, unreliable links are penalized with their large variances. In this way, biasing against unreliable links is achieved by magnifying the frequently changing delay effect.

In virtual circuit routing, the proposed algorithm follows the strategy used to try to route long messages through direct routes and where required, let the short messages take less direct routes. This is desirable for operational networks because the message delay is distributed to all messages, while average delay per message tends to a minimum [12].

## 4. PERFORMANCE EVALUATION

A qualitative comparison with the new ARPANET routing algorithm and Kamoun's routing scheme is done in [1]. In this Section, the HHARA is evaluated by simulation experiments based on a 50 nodes simulated network. The performance is compared to a fixed routing algorithm, the new ARPANET routing algorithm. It is simulated on a Pyramid 98X and a set of SUN-3/50 workstations runing BSD4.2 UNIX operating system at the department of Computer and Information Science, the Ohio State University. The motivation for the development of these simulation experiments is fourfold:

(a) To test the major feature of the new proposed scheme HHARA for large network.

(b) to study its implementation guidelines.

(c) To study its performance compared with other routing algorithms.

(c) To study the dynamic characteristics of the algorithm when applied to a large dynamic network and to suggest the mandatory future work to improve the algorithm.

### 4.1. Performance Measurement

Message delay is probably the single most important performance measure in computer network. Fortunately, average message delay is a natural metric for a computer network performance measure and reflects the following network phenomena in its measured value for the given network [ 9]: (a) message delay due to formation of queues within the nodes, (b) nodal processing delays, (c) nodal storage blockage, (d) propagation time delay, (e) packet retransmission due to link errors, (f) adaptability of the routing algorithm to varying traffic loads and link and node failures, and (g) packet looping caused by momentary errors in route determination by the routing algorithm.

The average packet delay may not be easy to estimate accurately in the design phase. Following criteria are very useful in performance evaluation. [25, 23].

(a) *Reliability* with attributes: ability ro recover and reconfigure, speed of response, and distributed control.

(b) *Communication Overhead* with attributes: number of control packets, and size of the control packet.

(c) *Computation overhead* with attributes: memory size, and computational complexity.

Reliability of routing algorithm is a measurement which depends on whether the algorithms can gracefully recover when the network topology changes, and the performance of its recovery procedures. One of the major concerns for the reliability of routing algorithms is that the algorithm shall not collapse as the failure of the one or more nodes in the network, thus distributed routing control algorithms are much preferred. Communication overhead mainly comes from the number of the packets sent by the nodes either for periodic update or aperiodic update. The goal is to minimize the communication overhead. Computational aspects are usually emphasized in the routing algorithm design, and memory size and computational complexity is usually minimized. Recent development of VLSI technology has also made available fast parallel processing units at very low cost, the computational complexity is no longer considered a great burden as long as it is a polynomial algorithm such as Dijkstra's and Warshall's Algorithms. Therefore, reliability and communication overheads are the central issues in routing algorithm. A similar conclusion has been made in [8]. Some comparison study on non-hierarchical routing algorithms using these criteria has been done in [2, 30].

### 4.2. Network Model

The communication network consists of the switching computers (nodes) and the communication links. It is a packet-switching store-and-forward network. An incoming message entering to a node is divided into fixed length packets, which are routed to the destination by the network. A packet entering a node is routed to an outgoing link according to the routing algorithm. It is placed in a queue associated with the particular link if the selected link is busy. When the link becomes available, it is transmitted to the next node in its path. The above process is repeated until the packet is delivered to its destination.

The characteristics of the network model are as follows.

(a) Pairs of nodes $(i, j)$ are connected by a bidirectional full duplex communication link.

(b) Traffic flow from node $i$ to Node $j$ is different from traffic from node $j$ to node $i$. So does the packet delay.

(c) Each node has infinite storage capacity.

Due to the system limitation, the simulated network only consists of 50 nodes and 96 bidirectional links. The network is clustered within one supervisor cluster. Each first level cluster consists of 9 nodes. Links capacities have four possible values 56000, 4800, and 2400 bit per second. The interconnecting links between neighboring clusters in chosen with low bit rate, 2400 bps to test the effect of the algorithm locality. By this way, the network is organized into a two level hierarchy. The larger scaler simulation may be done in the future if the resources are available.

generated packets since it has no ability to adapt to the topology changes. Comparing to the other two algorithms, it is about 100% higher. This ratio will be too high in a dynamic network. If the delay due to the lost packets were counted, it will performs much worse.

Figure 3 to Figure 5 show the correlation between network performance (average packet delay, ratio of lost packets, and throughput) and the Mean Interarrival Time. In Figure 3 as the network becomes loaded, the average packet delay is going up although HHARA has the lowest rate, especially when the cluster adaptation period is small. Figure 4 shows HHARA has the lowest ratio of lost packets and FIX has the highest. The rate of increasing the ratio of lost packets in ARPA in higher than HHARA especially when the cluster adaptation period is small. Figure 5 presents the maximum throughput for HHARA than others by 100% under traffic changes and it is more when the network is heavily loaded.

Figure 6 to Figure 8 show the correlation between the network performance and the MTBF. It is clear that HHARA is better than others when the network is unreliable. In Figure 6, the average packet delay does not increase monotonically when the network reliability is decreased. This is due to that the packet delay of lost packet during link failures does not count in the simulation. And, this is clear from Figure 7 as MTBF decreases the ratio of lost packets is increasing. As the MTBF reaches less than 1800 seconds, the ratio of lost packets goes up more than 40% in ARPA and in HHARA is still less than 20%, especially when the cluster adaptation period is small. Figure 8 indicates that HHARA achieves the highest throughput especially when the network is unreliable. As MTBF in going less than 1800 seconds, the throughput is decreasing in ARPA and FIX by a rate higher than HHARA. Also this figure shows HHARA has the highest throughput in all conditions of traffic variations and still has better throughput when the network is loaded.

Due to the small scale of the simulation, there is no significant difference in the average hop counts observed in the simulation. The average hop count of three algorithms are close in the simulation.

As the cluster adaptation period goes up, the average packet delay and the ratio of lost packets are increasing. It is found that the lower cluster adaptation period provides better performance for the whole network from the point of lower average packet delay and ratio of lost packets, especially when the network is unreliable.

Global adaptation period could reduce the average packet delay when it is shorter although the shorter the period induces more update overhead. However, since only supervisor nodes broadcast update messages during the global adaptation, this overhead is well under control. Observed from the current result of the simulation, it seems to be beneficial to further reduce the global adaptation period when the network is unreliable. There may have a strong correlation between the global adaptation period and cluster adaptation period. This correlation depends on many factors such as the traffic pattern, global adaptation period, and network topology. Further simulation study is needed to exploit this correlation and to determine the 'best' value under various working conditions.

## 5. CONCLUSIONS AND FUTURE WORK

A new hierarchical hybrid adaptive routing algorithm based on the modification of delta routing is proposed and evaluated in this paper. The algorithm performs quite satisfactorily as compared to other routing algorithms for large dynamic networks. This is achieved by a careful balance between the reduction of routing database maintenance overhead, and the global route optimality as well as the local adaptivity to the network changes. The routing database maintained in each routing decision site is smaller; the communication overhead of the routing database is reduced, as a consequence; and the unwanted looping is avoided due to the good cooperation of routing hierarchy. The study shows that the performance of the proposed scheme depends mainly on the clustering structures and traffic flow locality. Additional simulation experiments are desirable in the near future to compare the proposed scheme with other hierarchical routing schemes. The operability of the proposed scheme is mandatory right now for special purpose computer networks such as large military computer communication networks. Further, session oriented routing using the proposed scheme needs more details in the design.

## References


1. Amer, Fathy and Yao-Nan Lien, "A Comparative Survey of Routing Algorithms for Computer Commmunication Networks," *Tech. Report OSU-CISRC-9/87-TR27*, Dept. of Comp. and Info. Sci., Ohio State Univ., Sep. 1987.
2. Amer, F., M. El Dessouky, H. Farahat, A. El Moghazy, and A. El Abassi, "Routing Algorithms in Large Scale Computer Communication Networks State-of-the Art," *The 21st Annual Conf. in Statistics, Computer Science and Operation Research*, vol. 3 Computer Science, Cairo, Egypt, Dec. 1986.
3. Baratz, A. and J. Jaffe, "Establishing Virtual Circuits in Large Computer Networks," *Proc. of INFOCOM 83*, San Diego, California, April 1983.
4. Cerf, Vinton G., "Military Requirements for Packet Switched Networks and their Implications for Protocol Standardization," *Computer Networks, The Int'l Journal of Distributed Informatique*, vol. 7, Paris, Oct., 1983.
5. Chlamtac, Imrich and Avishai Elazar, "Hierarchical Routing with Bounded Routing Error," *Computer Network Symposium*, pp. 196-202, 1986.
6. Dessouky, Mahmoud El, "Simulation and Statistical Tools," *Internal Report CNET (Center National D'etudes Des Telecommunications)*, pp. 1-18, Paris, June 1980.
7. 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.
8. Friedmon, D. H., "Communication Complexity of Distributed Shortest Path Algorithms," *MIT Report, #LIDS-TH-886*, 1979.
9. Fultz, G. L., "Adaptive Routing Techniques for Message Switching Computer Communication Networks," *Ph. D. Dissertation, School of Engineering and Applied Science, UCLA, UCLA-Eng 7252*, July 1972.
10. Garcia-Luna-Aceves, J. Joaquin, "A New Minimum-hop Routing Algorithm," *Proceedings of IEEE INFOCOM'87*, pp. 170-180, San Francisco, March 1987.
11. Hilal, Wael and Ming T. Liu, "Guided-Adaptive Routing Techniques for Packet Switching Networks," *IEEE Proc. of Comp. Network Symp.*, pp. 65-74, Dec. 1981.
12. 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.
13. 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.
14. Kamoun,, "Design Consideration for Large Computer Communication Networks," *Ph. D. Dissertation, School of Engineering and Applied Science, University of California Los Angeles*, Los Angeles, California, March 1976.
15. Kamoun, Farouk and Leonard Kleinrock, "Stochastic Performance Evaluation of Hierarchical Routing for Large Network," *Computer Networks*, vol. 3, pp. 337-353, 1979.
16. Kerr, I. H., G. R. A. Gomberg, W. L. Price, and C. M. Solomonides, "A simulation Study of Routing and Flow Control Problems in a Hierarchically Connected Packet Switching Network," *Proc. Int'l Compu. Comm. Conf.*, p. 495, Toronto, Aug. 1976.

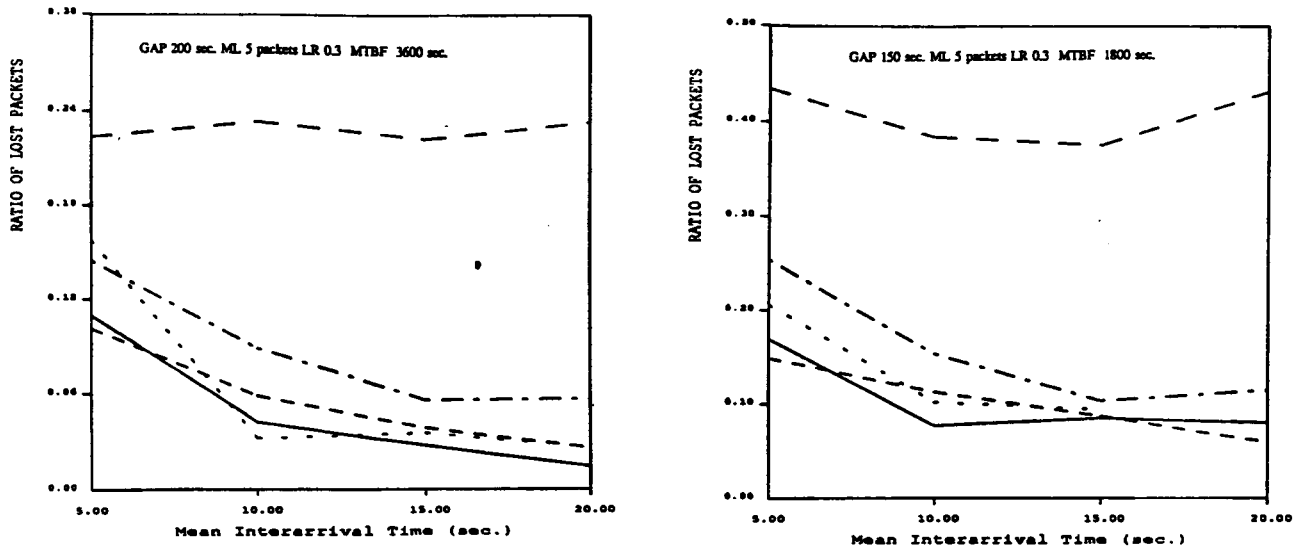



Figure 4. Simulation Results: Ratio of Lost Packets vs. Mean Interarrival Time.

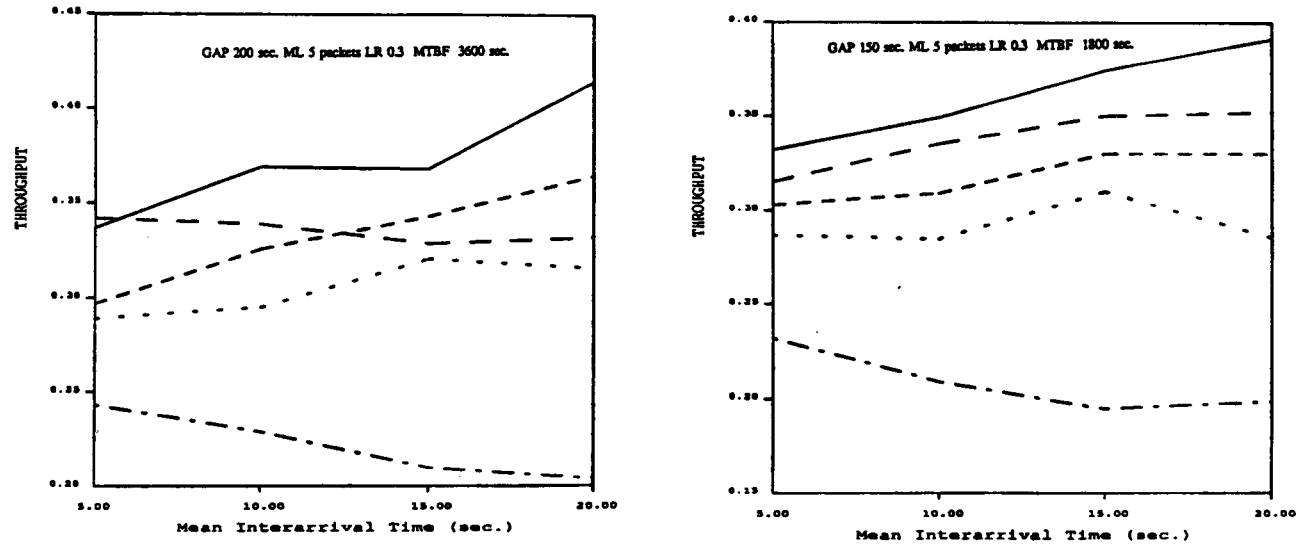



Figure 5. Simulation Results: Throughput vs. Mean Interarrival Time.

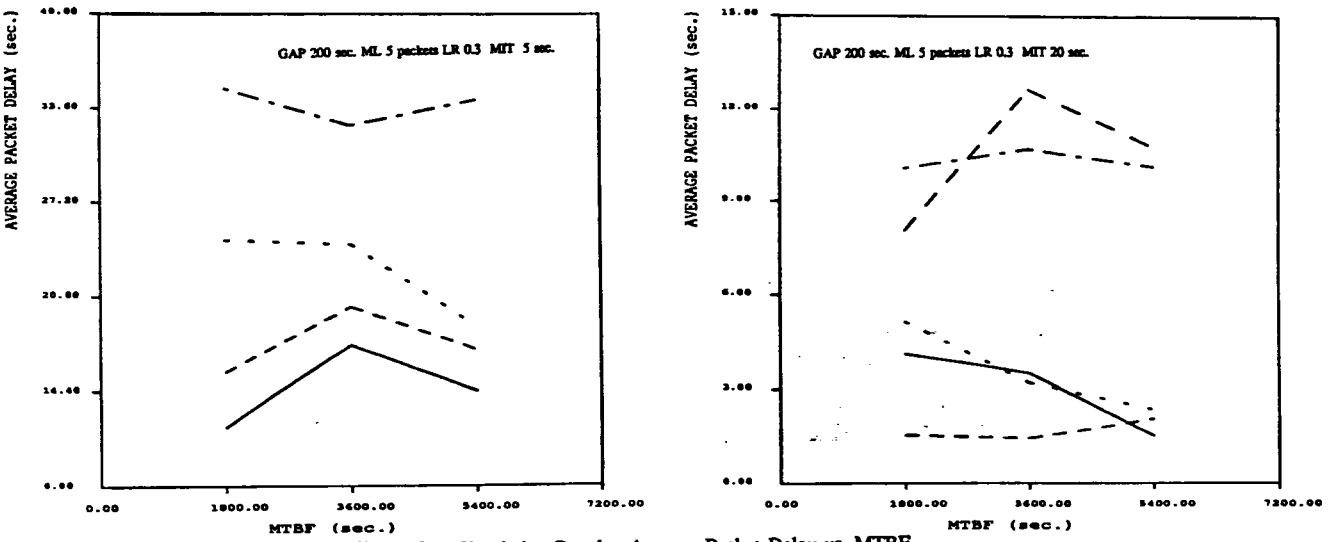



Figure 6. Simulation Results: Average Packet Delay vs. MTBF.

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 there is a need for a routing scheme that can reduce the routing database without loss 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, 7].

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 it to perform adequate routing. Thus, the routing database maintenance overhead can be reduced, while the optimality of generated routes and the adaptivity to the local changes are preserved. Notice that packets are not necessary to be forwarded across the hierarchy since such a scheme may be very poor. Nodes in HHARA are organized into multi-level structured clusters such that the distribution of routing responsibility may have a better match to the traffic flow locality.

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 in a dynamic network.

Since it is almost impossible to evaluate the performance of such a complicated algorithm under various loading conditions analytically, a simulation experiment is conducted to evaluate HHARA and to compare to a fixed routing algorithm and the new ARPANET algorithm in the experiment.

The paper is organized as follows. Section 2 gives a brief review on the previous work on hierarchical adaptive routing algorithms. HHARA is described in Section 3. The comparison evaluation of the proposed scheme and other routing algorithms is evaluated qualitatively and is also quantitatively evaluated using simulation experiments in Section 4. Finally, conclusion and future work are presented in Section 5.

## 2. PREVIOUS WORK

This section briefly reviews the newly proposed hierarchical routing algorithms. Detailed surveys can be found in [ 1]. The earliest work in hierarchical routing is possibly made by [16]. A two-level hierarchy containing five local areas, each of nine ordinary nodes and one super-node, 50 nodes in all. Fultz [ 9] proposed to cluster the network to employ hierarchical routing algorithm. Kleinrock and Kamoun [17, 15] 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 $N$ entries to $e \cdot ln\ N$ entries, where $N$ is the number of nodes. Baratz and Jaffe [3] 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. This technique is only applicable to virtual circuit networks like TYMNET, or SNA, and not datagram networks like ARPANET. Hilal and Liu [11] considered the design of two types of adaptive routing techniques, Periodic Queue-Exchange and Asynchronous-Traffic Splitting. Both techniques use more than one route between each source-destination pair to avoid routing conflicts and to evenly distribute the traffic on the available links, especially the critical links of the network. In Jaffe's routing scheme [13], 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. Within each cluster, each node maintains full topology about its own cluster. In addition, each node maintains limited information about remote clusters. This consists of two pieces. First, each node maintains information about nodes close to itself (radius database.) Second, each node has complete information about the network backbone. A radius database provides optimal routing between frequent communicators. The combination of a cluster database, and a backbone, allows reasonable routing to all nodes. Chlamtac and Elazar [5] 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. Ramamoorthy and Tsai [23] 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. Houstis and Leon [12] 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 problem of clustering in hierarchical routing remains open. In general, the clustering problem depends on the network topology, connectivity, the traffic, the routing algorithm, as well as the geographical location of the nodes. A clustering structure that minimizes the length of the routing tables is devised by [14]. However there is still no procedure that can give the general optimal clustering structures and takes the advantages of geographical locations of the nodes. The problem is even more complicated in a multi-level hierarchical clustering. Therefore, heuristic clustering algorithms may be used. Due to the time constraints in a dynamic computer network, even high order heuristic clustering algorithms are not usable.

## 3. PROPOSED HIERARCHICAL HYBRID ADAPTIVE ROUTING ALGORITHM

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, 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

another for supervisor-cluster update. The updating frequency is an engineering parameter determined by the network designer.

The correct operation of the proposed algorithm assumes the existence of a link-level protocol that assures that [ 10]:

- Every node knows its neighbors inside the cluster and its supervisor(s).
- All packets forwarded over a link are received corectly 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.

Virtual circuit routing is initiated via a "route-setup" procedure. If a user or a node wishes to communicate, it sends a ROUTE-SETUP packet along the desired communication path. 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.

Table 1 HHARA Delay Table at cluster 1.2.

| Destination | Source Nodes | | | | |
|---|---|---|---|---|---|
| | 1.2.1 | | | | |
| | 0 | 150 | 300 | 220 | 205 |
| 1.2.2 | 105 | 0 | 116 | 710 | 231 |
| 1.2.3 | 156 | 257 | 0 | 311 | 578 |
| 1.2.4 | 510 | 125 | 321 | 0 | 912 |
| 1.2.5 | 781 | 196 | 236 | 421 | 0 |

(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 2, the Routing Table at node 1.2.3, illustrates an example of HHARA Routing Table at each node. The first section of the routing table is generated by the node itself. The second section is generated by the supervisor node and sent to all its subordinate nodes. The highest level cluster hierarchy routing table will have empty Supervisor Section because it has no supervisor.

Table 2 HHARA Routing Table at node 1.2.3.

| Routing Entries for nodes | Destination Nodes | Next Nodes | Alternate Nodes | Hop Field (Path Length) |
|---|---|---|---|---|
| Inside the cluster | 1.2.1 | 1.2.4 | 1.2.1 | – |
| | 1.2.2 | 1.2.2 | 1.2.1 | – |
| | 1.2.3 | 0 | 0 | – |
| | 1.2.4 | 1.2.1 | 1.2.4 | – |
| | 1.2.5 | 1.2.1 | 1.2.4 | – |
| Outside the cluster | 1.1.3 | 1.2.1 | 1.2.4 | – |
| | 1.4.1 | 1.2.2 | 1.2.1 | – |
| | 1.3.1 | 1.2.4 | 1.2.1 | – |
| | 1.5.2 | 1.2.1 | 1.2.4 | – |
| | 1.5.3 | 1.2.2 | 1.2.1 | – |
| | 1.6.1 | 1.2.1 | 1.2.2 | – |
| | 2.2.1 | 1.2.2 | 1.2.1 | – |
| | 3.1.1 | 1.2.3 | 1.2.1 | – |
| | – | – | – | – |
| | – | – | – | – |
| | – | – | – | – |
| | – | – | – | – |
| | – | – | – | – |

#### 3.3.2. Routing Database update

In order to react to failures, congestion, or a planned or nonplanned topology change, and traffic variations, the routing databases are updated automatically.

There are two kinds of updates: the cluster update and the supervisor-cluster 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 [22, 18]. 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-cluster 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.

### 4.3. Queueing Network Model

Internal structure of a node is shown in Figure 2. Part of its function is to receive the messages/packets from the communication links, store there and make the routing decisions based on the routing algorithm. When an outgoing link selected is not free, then the message/packet is placed in the outgoing queue of the selected link. The queueing is done on a First-in-first-out (FIFO) priority discipline. When the outgoing link becomes free, the message is transmitted to the next node. Usually there is some networking overhead consumed in each node that a packet is forwarded. For example, error detection, that takes place. Propagation delay is taken into consideration and it is changing during simulation runs.

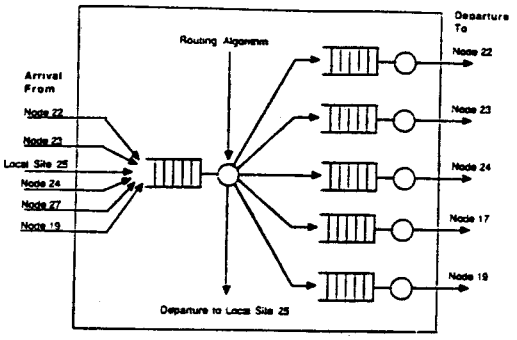



Figure 2. Structure of supervisor node 25/second level cluster 2.1.

The network is assumed to be partitioned into many appropriate sized clusters. In the model, we have 50 node, 192 link, and the network is clustered into five first-level clusters and one second-level cluster which is the supervisor cluster. By this way, the network is organized into hierarchically structured clusters such that only partial information is stored and maintained in each site.

**Traffic characteristics**

The network traffic characteristic are modeled as follows.

(a) All external arrivals are i.i.d.

(b) The external arrivals to a node follow a Poisson distribution.

(c) All message classes have the same exponential message length distribution.

(d) Each incoming packet has single destination.

The third assumption implies that the service time distribution at each link is exponential distribution. When a packet is routed from one node to another, its length remains unchanged.

### 4.4. Simulation

SMPL simulation package is used in the simulation. It is a simple, portable simulation language designed for discrete-event simulation systems [19]. HHARA, a fixed routing, and New ARPANET routing algorithm are simulated. Fixed network configuration parameters and simulation parameters are shown in Table 3.

*Global Adaptation Period* is the period for which all nodes in the network update their routing databases. In new ARPANET routing algorithm, this period has a fixed value 10 seconds and it can be extended up to 60 seconds if there is no significantly changes in routing updates. This period is used also in the proposed scheme for the centralized portion only (see section 3.3.2). During sudden changes in topology or traffic the new ARPANET routing algorithm and the centralized portion of the proposed scheme update their routing databases. *Cluster Adaptation Period* is the period for which each cluster updates its nodes' routing databases through distributed adaptive fashion. It is fixed for all clusters and its value depends on the local current conditions of the cluster. It might happen updating also for sudden changes as above. *Local Message Ratio* is the ratio for each node to generate messages to its vicinity nodes inside the cluster, and the rest of its generation to farther away nodes in the network.

Table 3 Simulation Parameters.

| Simulation Parameters | |
|---|---|
| Number of Nodes | 50 |
| Number of Links | 192 |
| Packet Size | 1000 bits |
| Mean Message Length (ML) | 5 packets |
| Global Adaptation Period (GAP) | 50, 100, 150, 200 seconds |
| Cluster Adaptation Period (CAP) | 5, 10, 20, 30 seconds |
| Mean Message Interarrival Time (MIT) | 5, 10, 15, 20 seconds |
| Local Message Ratio (LR) | 0.3 0.6 0.9 |
| Link Mean Time Between Failure (MTBF) | 1800, 3600, 5400 seconds |
| Link Mean Time To Repair | 1800 seconds |
| Link Capacity | 56000, 4800, 2400 bits |
| CPU Speed | 1 MIPS |

Following values are observed in the simulation: average packet delay, average hop count, network throughput, network utilization factor, and ratio of lost packets.

Network utilization factor is defined as the ratio of the rate at which "work" enters the network to the maximum rate (capacity) at which the network can perform this work. By another meaning:

$U$ = (Average arrival rate of packets) × (Average packet service time)

In this preliminary study, only node utilization is considered. The link utilization, which is more important, will be studied in the future. The throughput factor is defined as the ratio of the delivered packets to the total number of packets handled by the network [20]. This factor is directly influenced by the network utilization factor. The throughput factor decreases with increasing the utilization factor.

Due to the implementation difficulty, the delay time caused by the lost packets is not counted in the simulation. Although this may distort the simulation result, it is yet serious enough to distort the relative performance of simulated algorithms. In the original design of HHARA, the update information reported to the supervisors have to be aggregated inside the clusters before reported to the supervisors to reduce the necessary update traffic. However, each node reports the update information directly to its cluster supervisor since a reliable information aggregation mechanism is yet to be designed. By doing this, the reliability of updating algorithm is much higher in the face of failures. It is our observation that this does not induce much more overhead.

In order to observe the sensitivity of the network to a particular parameter, the message interarrival time, the global and local adaptation periods, the mean time between failure, and the local message ratio at all nodes and links are the same and are input parameters. Due to the time constraints, the mean message length is fixed in all runs.

### 4.5. Simulation Results

The new ARPANET routing algorithm and the fixed algorithm used in the simulation are referred to as ARPA and FIX respectively in this section. Due to inevitable errors in the simulation, the values of average packet delay may have a distortion up to 1 second as far as we can tell. The relative performance among different routing algorithms shows here clarify better evaluation than the absolute values in the quantitative comparison.

As we can see from Figure 3 to Figure 8, the average packet delay of HHARA is shorter than ARPA in reliable networks. As we expected, HHARA may perform 30% better than ARPA when the network is very unreliable. Also, the ratio of lost packets is lower in HHARA. HHARA will perform better since it collects topology and traffic information more frequently than ARPA.

In the figures, the average packet delay of FIX looks close to that of ARPA and its throughput looks to be the highest among all algorithms. However, the ratio of lost packets may be as high as 50% of

17. Kleinrock, Leonard and Farouk Kamoun, "Hierarchical Routing for Large Networks : Performance Evaluation and Optimization," *Computer Networks*, vol. 1, pp. 155-174, 1977.

18. Leiner, Barry, Jon Postel, Robert Cole, and David Mills, "The DARPA Internet Protocol Suite," *IEEE INFOCOM Proc.*, pp. 28-36, March 1985.

19. MacDougall, M. H., "SMPL - A Simpe Portable Simulation Language," *Amdahl, Technical Report*, April 1980.

20. McCoy,Jr., Caldwell, "Improvements in Routing for Packet Switched Networks," *Ph.D. Dessertation*, School of Eng. and Applied Science, george Washinton University, Feb. 1975.

21. 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.

22. 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.

23. Ramamoorthy, C. V. and W. T. Tsai, "An Adaptive Hierarchical Routing Algorithm," *Proceedings of 7th COMPSAC*, pp. 93-104, 1983.

24. Rudin, H., "On Routing and Delta Routing: A taxonomy and Performance Comparison of Techniques for Packet-Networks," *IEEE Trans. on Comm.*, vol. Comm-14, No. 1, pp. 43-59, Jan. 1976.

25. Schwartz,, "Routing and Flow Control in Data Networks," *IBM Research Report RC 8353 (# 36329)*, July 1980.

26. 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.

27. Stallings,, William, *Data and Computer Communications*. Macmillan Publishing Company, New York, 1985.

28. Stassinopoulos,, "Optimal Dynamic Routing in Multidestination Networks," *IEEE Trans. on Comm.*, vol. Comm-35, No. 4, pp. 472-474, April 1987.

29. Tanenbaum, A. S., *Computer Networks*. Prentice-Hall Inc., 1981.

30. Tsai, W. T., "Routing Techniques for Dynamic Computer Networks," *MS Report*, Computer Science Division, Department of EECS, University of California, Berkeley, Aug. 1982.

31. Zhang, Lixia, "Designing a New Architecture for Packet Switching Communication Networks," *IEEE Comm.*, vol. 25, No. 9, pp. 5-12, Sep. 1987.

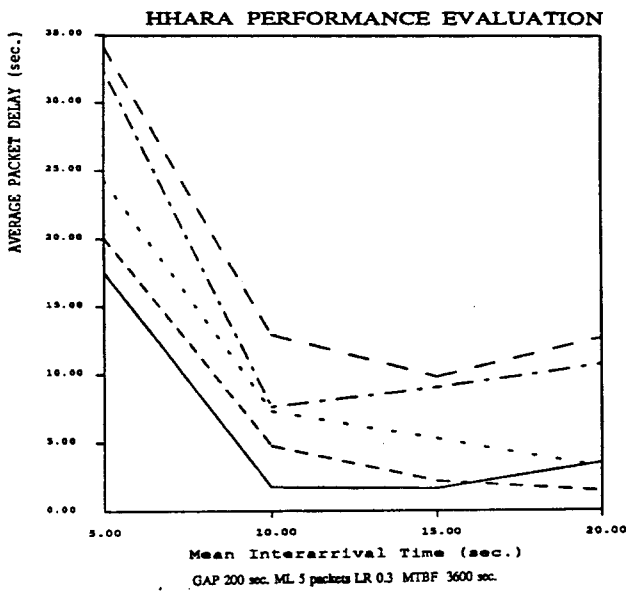
**LEGEND**

| | |
|---|---|
| GAP: | Global Adaptation Period (seconds) |
| CAP: | Cluster Adaptation Period (seconds) |
| MIT: | Mean Interarrival Time (seconds) |
| MTBF: | Mean Time Between Failure (seconds) |
| LR: | Ratio of local traffic |
| ML: | Mean Message Length (packets) |
| cap10: | HHARA with 10 second CAP |
| cap20: | HHARA with 20 second CAP |
| cap30: | HHARA with 30 second CAP |

HHARAcap30 FIX

HHARAcap20 ARPA

HHARAcap10



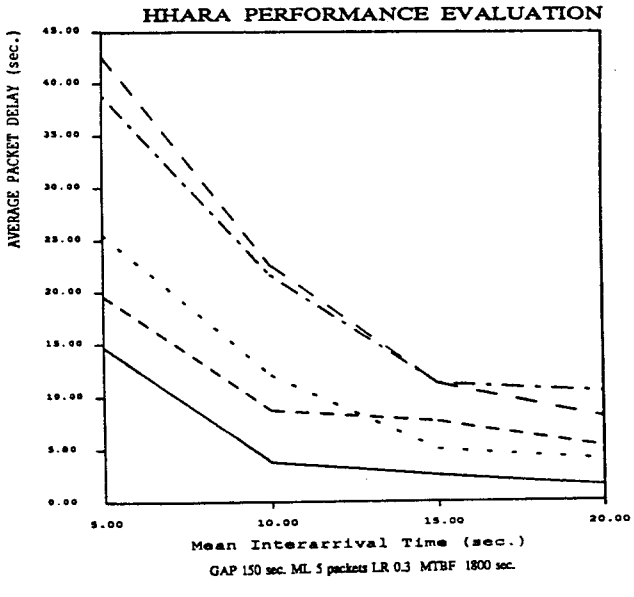




Figure 3. Simulation Results: Average Packet Delay vs. Mean Interarrival Time.

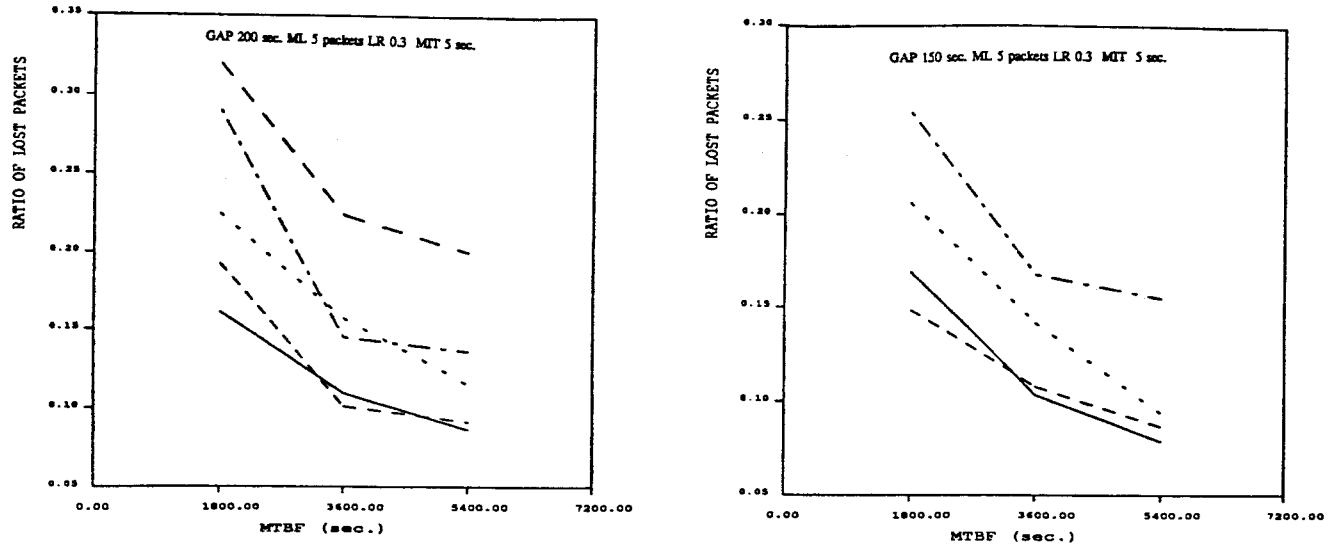



Figure 7. Simulation Results: Ratio of Lost Packets vs. MTBF.

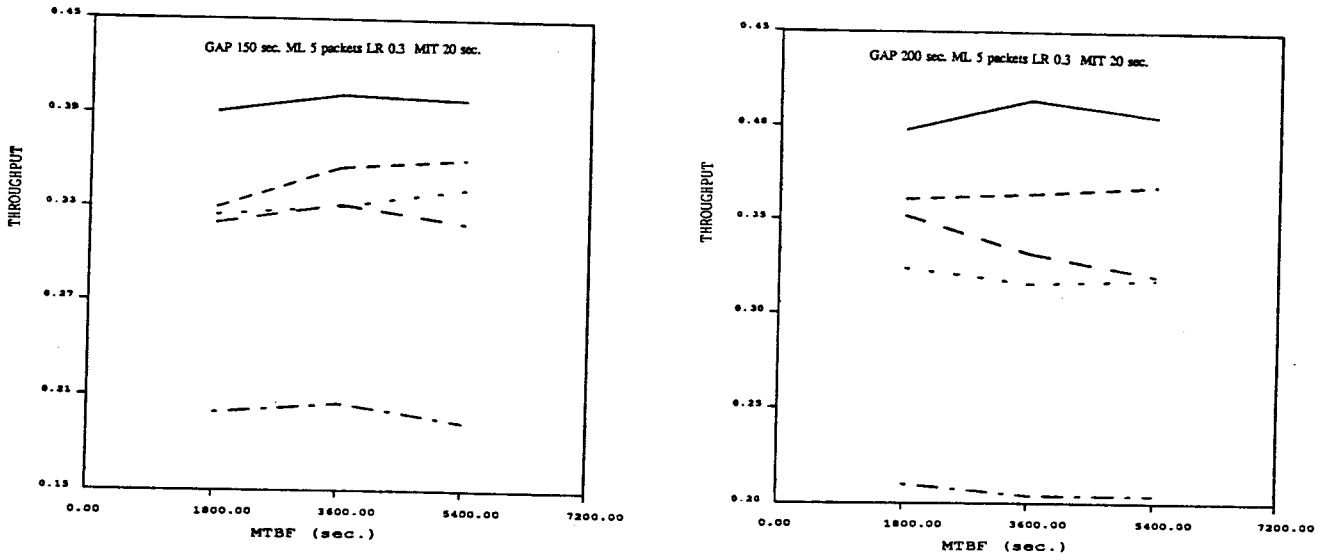



Figure 8. Simulation Results: Throughput vs. MTBF.