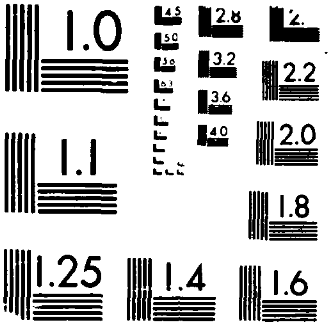
AD-A193 942 HIERARCHICAL ORGANIZATION FOR LARGE DYNAMIC RADIO 1/1
NETWORKS(U) SRI INTERNATIONAL MENLO PARK CA N SHACHAM
JAN 88 SRI-ECU-6975 ARO-20948.10-EL DAAG29-84-K-0042
UNCLASSIFIED F/G 25/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

ARO 20948.10-E

②

Final Report

January 1988

HIERARCHICAL ORGANIZATION FOR LARGE, DYNAMIC RADIO NETWORKS

By: NACHUM SHACHAM, *Program Director*
Information Sciences and Technology Center

Prepared for:

U.S. ARMY RESEARCH OFFICE
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-2211

Attention: DR. WILLIAM SANDER

CONTRACT DAAG29-84-K-0042

333 Ravenswood Avenue
Menlo Park, California 94025 U.S.A.
(415) 326-6200
Cable: SRI INTL MPK
TWX: 910-373-2046

SRI International

SRI International

Final Report *January 1988*

HIERARCHICAL ORGANIZATION FOR LARGE, DYNAMIC RADIO NETWORKS

By: NACHUM SHACHAM, *Program Director*
Information Sciences and Technology Center

Prepared for:

U.S. ARMY RESEARCH OFFICE
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-2211

Attention: DR. WILLIAM SANDER

QUALITY INSPECTED 4

CONTRACT DAAG29-84-K-0042

SRI Project 6975

Accesion For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved by:

MICHAEL S. FRANKEL, *Director*
Information Sciences and Technology Center

DONALD L. NIELSON, *Vice President*
Computer and Information Sciences Division

SRI International®

333 Ravenswood Avenue • Menlo Park, California 94025 • U.S.A.
(415) 326-6200 • Cable: SRI INTL MPK • TWX: 910-373-2046

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
N/A since Unclassified

2a. SECURITY CLASSIFICATION AUTHORITY
N/A since Unclassified

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A since Unclassified

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
ECU 6975

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ARO 20948.10-EL

6a. NAME OF PERFORMING ORGANIZATION
SRI International

6b. OFFICE SYMBOL *(if applicable)*

7a. NAME OF MONITORING ORGANIZATION
U.S. Army Research Office

6c. ADDRESS *(City, State, and ZIP Code)*
333 Ravenswood Avenue
Menlo Park, California 94025-3493

7b. ADDRESS *(City, State, and ZIP Code)*
P.O. Box 12211
Research Triangle Park, NC 27709-2211

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
U.S. Army Research Office

8b. OFFICE SYMBOL *(if applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DAAG29-84-K-0042

8c. ADDRESS *(City, State, and ZIP Code)*
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
	1L161102 BH57-03		

11. TITLE *(Include Security Classification)*
HIERARCHICAL ORGANIZATION FOR LARGE, DYNAMIC RADIO NETWORKS

12. PERSONAL AUTHOR(S)
Dr. Nachum Shacham

13a. TYPE OF REPORT
Final Report

13b. TIME COVERED
FROM 840301 TO 871231

14. DATE OF REPORT *(Year, Month, Day)*
880131

15. PAGE COUNT
24

16. SUPPLEMENTARY NOTATION
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy,

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
25	02	
25	05	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Packet Radio
Packet Switching
Network Architecture
Protocols
Network Control

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

As new technology is incorporated into weapons and command and control systems, the need for data communications and data processing at the tactical level will increase dramatically. Tactical ground forces are mobile and they operate in a highly stressed environment complete with noise, false messages and attrition of communication resources.

Packet radio network (PRNET) is a promising technology that can serve tactical data communication applications if its architecture and the functionality of its protocols is enhanced to allow it to adapt to frequently changing environment and incorporation of a large number of users. Developing such architectures and protocols for large, survivable PRNETs and evaluating their performance were the principal thrusts of this

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE *(Include Area Code)*

22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR
83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

16. SUPPLEMENTARY NOTATION (Continued)

or decision unless so designated by other documentation.

19. ABSTRACT (Continued)

research project. The main advances made in this effort include development of architectures and algorithms that fall in the following categories:

- Hierarchical architectures and routing for dynamic networks.
- Multichannel networks--architectures and protocols.
- Dynamic selection of radio FEC code rate to adapt to channel variations.
- Broadcast algorithms that provide for transport of messages to multiple destinations amidst topological changes.
- Self-organizing networks in which the nodes control their local connectivity.

The correctness of some algorithms was validated analytically; others were tested by simulations.

1 INTRODUCTION

As new technology is incorporated into weapons and command and control systems, the need for data communications and data processing at the tactical level will increase dramatically. Tactical ground forces are mobile and they operate in a highly stressed environment complete with noise, false messages and attrition of communication resources. Thus, the data communication networks that provide the information distribution functions for Army tactical command and control (C^2) must be able to operate in a highly dynamic environment. Such networks are also required to be able to serve a large number of users, allowing each one of them to communicate with other users of their choice.

The users' mobility implies that these networks must rely on radio channels as the transmission medium and, being data networks, they can use packet switching to utilize best the scarce radio spectrum [7]. However, existing packet radio network (PRNET) technology, as demonstrated so far, has limited functionality that allows for the incorporation of only a moderate number of nodes in a single network, and those protocols can support limited degree of mobility. More advanced algorithms, protocols, and network architectures are needed to allow tactical PRNETs to incorporate a large number of users and to continue providing service in a dynamic environment [27,28,26,4]. Developing such architectures and protocols for large, survivable PRNETs and evaluating their performance were the main thrusts of this research project.

Over the contract period we made significant advances in these areas. In particular, we have focused on the following subjects:

- Network architecture for:
    - Hierarchical networks structures that allow a large population of mobile radio nodes to operate as a network.
    - PRNETs that use multiple parallel channels to increase their throughput.
- Dynamic selection of transmission parameters to allow radio units to adapt to varying channel and network conditions.

- Broadcast algorithms that provide for transport of messages to multiple destinations amidst topological changes.
- Self-organizing networks in which the nodes control their local connectivity.

Papers describing the various results of this research project were published in refereed technical journals and in conferences proceedings. These papers are cited in the following sections and are also listed in the Appendix of this report.

The results of this project have proved significant, relevant, and beneficial in several ways:

- **Military experimental PRNETs:** Ideas and protocols in hierarchical architectures and self-organizing networks were incorporated into the experimental network developed under the Survivable Radio Networks (SURAN) DOD-sponsored (both DARPA and Army) program. The principal investigator of the project reported here also provides System Engineering and Technical direction (SETD) to the SURAN program. The interrelation between these two programs has proven beneficial to both.
- **Non-military PRNETs:** The study conducted in this project influenced the design of a packet radio network used for library automation by the University of California's Division of Library Automation [22,12].
- **Scientific research:** Many of the project-sponsored published papers from this project have been cited in other scientific papers and have provided a basis for at least two Ph.D. research theses that expanded on the models and architectures first developed under this project.

2 NETWORK ARCHITECTURES

2.1 HIERARCHICAL NETWORKS

Often a command and control system requires its communications networks to support a large number of users, distributed over a large geographical area, while allowing data exchange between any two points in the system. A PRNET architecture thus must be designed to be able to support a large number of nodes. Major issues in this area are the routing strategy and the data structures the nodes are required to store and exchange. The ability to find a path is a function of the information held at the nodes' routing tables. A "flat" architecture, where each network node (a radio unit) keeps routing information about every other node in the network [15], cannot handle large dynamic radio networks [7]. The size of the routing tables, which grows linearly with the number of nodes, and the volume of control traffic, which grows even faster, can easily consume a large portion of the network processing, memory, and channel resources, thereby reducing the level of service provided to user traffic.

"Hierarchical" architecture is a method for incorporating a large number of nodes in a network while keeping overhead low [8,14]. In this architecture the network is organized by a hierarchy of clusters, so that each node holds information sufficient for routing to any other node in the same cluster and, in the case of a two-level hierarchy, *to any other cluster* in the network. This arrangement greatly reduces the amount of routing data a node needs to store and update, and thereby lowers the volume of control traffic exchanged over the links. If further reduction is desired, say to incorporate a very large number of nodes, more levels in the hierarchy can be used.

The hierarchical architecture has two main disadvantages compared to the flat architecture [19]:

- The hierarchical network is slow to adapt to topological changes. If the clusters are node-disjoint, as originally suggested, mobile nodes keep changing their cluster affiliation, also called *address*, thereby causing messages which carry their old address to be discarded when they reach the destination's former cluster.
- The hierarchical routing tables result in origin-destination paths longer than those followed by the data in flat networks. In the worst case, the increase in path length can be up to a factor of three [5].

The reduction in network overhead provides a strong motivation for overcoming the deficiencies of hierarchical architecture in handling network dynamics. In this project we have investigated two mechanisms for overcoming this difficulty [19,28]:

- Parallel routing-table updates at the various levels of the hierarchy and distribution of the tables to the nodes at the lower levels. Nodes keep shortest-path and next-node in formation for each other node that belongs to its lowest-level cluster. Thus, each cluster is managed like a flat network. Our hierarchical technique calls for a nomination of a special node in each cluster at every level. Such a node, dubbed *cluster head*, establishes connections with the cluster heads of all bordering clusters, that is, all the clusters to which a message can be sent without having to pass through a third cluster. These clusterheads thus form a logical network, the topology of which they maintain. A clusterhead distributes its cluster-level data to all the nodes in its cluster. As a result, each node in the network has enough information to direct a packet to the next node on a route that leads either to that destination, if it is in the same cluster, or to the the destination's cluster.

  This two-tier routing approach results in somewhat longer routes than those followed by the packets if each node keeps direct distance information to every other cluster in the network. However, the major advantage of the two-tier routing is that it improves the routing data dissemination speed [5], thereby allowing the network to respond faster to topological changes. Furthermore, since two clusterheads retain their logical links as long as at least one physical radio link interconnects their clusters, the resulting intercluster network exhibits a lower degree of dynamics than the other scheme mentioned above, in which severance of a remote radio link may affect the distance to a cluster, thereby generating a large volume of control traffic.

- Overlapping clusters, i.e., node's affiliation with more that one cluster at a time. Each node participates in the routing information exchange in all the clusters to which it belongs and receives cluster-level routing data from all the corresponding cluster-heads. Thus when a node moves away from one of its clusters, thereby losing its ability to communicate directly with that cluster's members, it is likely to still retain connectivity with other clusters it belongs to. A packet carries all the addresses of its destination and thus can be routed to that destination in spite of the latter's mobility.

Overlapping clusters also enhance the network's stability when a cluster-head fails. If the clusters are disjoint, loss of a cluster head not only prevents the nodes of the affected cluster from getting cluster-level routing updates but also causes the entry for that cluster to be erased from the routing tables in the other clusters. The resulting effect is that the failure of a cluster-head in a disjoint cluster hierarchy removes the nodes in that cluster from the network. When the clusters overlap, however, a node whose cluster-head fails continues to receive routing services and data through the other cluster-heads with which it is affiliated.

Notice, however, that the size of the routing tables is larger when the clusters overlap than it is under disjoint clusters. Using a mathematical optimization model, it is possible to show that that increase is not very large. We have obtained lower bounds on the size of the routing tables and showed a scheme to organize the overlapping clusters such that the resulting routing tables are only about 22% larger than that lower bound [20].

2.1.1 Relevance and effects of this effort:

The work reported here helped to shape the hierarchical architecture of SURAN, whose selected structure for large scale-networks closely resembles the architectures developed here.

2.2 MULTICHANNEL OPERATION

Most of the research and development in packet radio so far has been directed toward networks whose nodes share a single channel. By this we mean that a node can receive any packet transmitted within the nodes reception range, provided that other packet transmissions do not interfere (collide). A typical node employs a transmitter/receiver (transceiver) radio that can operate on one channel at a time; however, modern hardware technology allows a radio to change its operating channel in milliseconds. The concept of a multichannel PRNET permits using more than one channel simultaneously within the same reception range, and consequently enables multiple packet transmissions within the same geographical area; it is analogous to broad-band local area networks. These channels can be either in the frequency domain (FDMA) or, in the case of spread-spectrum systems, in the code domain (CDMA). A combination of both types is also possible. It is desirable, but not essential, that the channels be orthogonal, inflicting little cross-over channel interference.

Multichannel operation has some advantages over operating with one channel:

- Allows for graceful expansion of network capacity when an increase in the allocation of a single wide-band channel cannot be accommodated.
- Even if the frequency allocation constraint could be relaxed to
- Allows for an increase in network capacity when an increase in the bandwidth of a single wideband channel is prohibited by the propagation media (e.g., frequency-selective fading).
- Permits a natural partitioning of user-groups in some circumstances.
- Reduces the mutual interference if the channels are sufficiently separated, although the same separation may pose a connectivity problem.
- Interference reduction is important to dense PRNETs because each node must generate a minimum level of traffic to establish solid connectivity with the node's neighbors. Given enough neighbors, the channel access requirements would exceed the available bandwidth. Hence, distributing the traffic among several channels reduces the congestion on each channel.

The advantages of the multichannel over the single-channel PRNET are acquired at the cost of higher complexity in organization and operation (sometimes in higher hardware cost as well). Protocols for routing, packet forwarding, broadcasting, and the like are all more complicated to design in a multichannel environment. In this project we have devised two basic architectural for organizing multichannel, multihop PRNETs operating with contention-based channel access protocols [24,23]:

- ***A single transceiver per node:*** In this architecture each transceiver selects a *quiescent* channel on which it listens when idle. A message to another node that is within hearing range is sent over the receiver's quiescent channel. The packet is successful if the receiving node is idle and if no other neighbor transmits on that channel simultaneously.

In this receiver-directed architecture, each node must know which quiescent channels its neighbors are using. This can be achieved by having the channels allocated at network initialization and relying upon network management tools to instruct the nodes during network operation.

- *Multiple transceivers per node:* Here each node is allowed to operate simultaneously over as many channels as it has transceivers. The extra transceivers allow the partition of the nodes into disjoint sets, each of which operates over different frequencies. Bridges among the sets are provided by interconnecting the digital ends of the radios in the multitransceiver nodes.

It is harder to gather and distribute routing information in a multichannel network than in a single-channel one. Our work resulted in the definition of a multichannel architecture and an extension to the multichannel operation of routing techniques used in single-channel PRNETs. Routing information may be broadcast by each node over all channels or just over one channel. The former technique provides more information to the nodes, and hence better routing, whereas the latter results in less channel overhead.

We evaluated the throughput of various multichannel schemes [23]. The purpose of this evaluation was to ascertain the gain from adding channels, to understand the effect of splitting a given channel into subchannels, and to compare the performance of a multichannel PRNET with either full or partial routing information. This evaluation was based on the model for a multihop PRNET that was first proposed by Kleinrock and Silvester [9]. This model assumes that the network's nodes are scattered over the plane according to a Poisson process with a given rate of nodes/unit-area. Routing is done through the node that provides the most forward progress toward the destination if the node knows the corresponding direction. Whenever partial routing information is used, a node that has to send a packet to a node on a different channel chooses a neighbor at random to which to send that packet. The network is assumed to be very large, and the model takes the performance of a single hop into account while considering the multihop interference as well.

We have ascertained the throughput per node per slot for a multichannel slotted ALOHA system that selects at random neighbors for cross-channel transmission. At high densities of nodes, throughput per node grows almost linearly with the number of channels available. The optimal value of throughput is almost identical for any number of channels. However, the density at which the optimum is achieved corresponds to approximately the same density of nodes in each quiescent channel. We have also shown that a larger increase in maximum throughput is possible when full routing is used than is achievable under partial information. In high densities, however, both schemes perform about equally.

2.2.1 Relevance and effects of this effort:

- Operating over multiple channels is now a high-priority implementation item for the SURAN program, with the architectures proposed in this effort as prime candidates.
- H.W. Chung, a Ph.D. student at Northwestern University, working under the guidance of Prof. S. Kumar, is now investigating multichannel PRNET architectures based on the work reported here.

3 DYNAMIC SELECTION OF ERROR CORRECTING CODES

The changing location of mobile nodes cause variation in the quality of the received signals because of different propagation conditions and different noise conditions. In military PRNETs, jamming is also an important contributor to the time-variations of channel quality. These variations in signal-to-noise ratio are seen by the nodes as time-varying error rates and, in extreme cases, as breakage or reestablishment of links [18].

The network-level protocols view these changes in signal quality as changes in the link and end-to-end throughput. These changes increase the level of congestion because of the network's reduced capacity and the increased overhead due to additional control traffic needed for updating nodes about these changes. If the changes are frequent, the delay in the dissemination of control information forces the network to operate under inaccurate information, causing suboptimal routing decisions which further aggravate the network congestion.

Thus, in order to stabilize performance, the nodes should be able to adapt their transmission parameters to the varying channel conditions, thereby presenting a more stable environment to the protocol layers above them. Our work has focused on adapting the rate of the forward error-correcting (FEC) codes to compensate for links variations.

Adaptive FEC can be achieved by sending information incrementally and by combining data from several packets at the receiver. In the FEC code class one can include the protocol under which the data are sent first, protected by error-detecting code; only upon detecting an error are the error correcting bits sent [11]. Another approach is to encode all the packets with FEC and combine the received signals according to their signal-to-noise ratio (SNR) [1].

Simplicity of operation in a PRNET dictates that a node discard erroneous packets that cannot be decoded. Under these constraints, the transmitter has to select a code for every transmission according to the channel conditions as it perceives them [25]. We devised two algorithms for adapting the FEC rate for a transmitter-receiver pair [17]. These algorithms monitor the results of packets transmissions and decide which is the best of two available codes for each packet. The basic elements of these schemes are:

- A link-quality measure, which indicates the state of the channel.
- Information extracted from the received packets, which is used to update the link-quality measure. The extracted information is the number of errors that a received packet contained – information that can be extracted only if the packet is successfully decoded. If the packet is not decoded successfully, this fact is also used to update the link-quality measure.

Two types of updating are considered: the first, parametric, assumes that the general shape of the possible distribution of the number of errors is known. Thus, the measure represents the probability that the channel is in the state for which one of the codes, say code number 1, is best suited. The error information is used to evaluate the conditional probability that the channel is in that state, given its a priori probability and the number of errors. The resulting conditional probability is then substituted for the link quality measure. If that measure is greater than some predetermined threshold, code 1 is used; otherwise the next packet is encoded by code 2.

The second updating scheme does not assume knowledge of the distribution of the errors and attempts to compare the two codes directly. For example, by knowing the number of errors in the packet encoded by code 1, the receiver can tell whether or not the other code could have been used to send that packet successfully. According to the answer, the link quality measure is updated by the amount of additional throughput achieved, or that could have been achieved, by the better of the two codes.

Evaluation of the performance of these two schemes has shown that both of the aforementioned adaptive schemes achieve higher throughput than the nonadaptive schemes for all "reasonable" rates of channel states change. The first scheme achieves higher throughput than the second scheme; this is not surprising, however, because the first scheme makes use of more information.

3.1 RELEVANCE AND EFFECTS OF THESE RESULTS:

- The SURAN program will integrate adaptive FEC rate selection in its protocols. The techniques discussed above are under consideration for these protocols. Also, the SINCGARS packet overlay, which is equipped with several FEC codes, can use these techniques for dynamic code selection.

- P. Feldmann, A Ph.D. student at the University of Southern California, working under the guidance of Professor Victor Li, has expanded on the results presented here and developed algorithms for adaptive code selection using the model developed in this project, but with additional assumptions on the knowledge at the transmitter [3].

4 BROADCAST AND ROUTING ALGORITHMS

Many communications networks make use of algorithms that allow a node to broadcast a message reliably to all other nodes in the network (e.g., [13,29]). We consider the case in which a node (the *originator*) sends a message reliably to a *subset* of the network nodes, which consists of all the nodes up to y hops away from the originator. In this section we present and validate a distributed algorithm that provides that service, and that has the following properties [10]:

- All the nodes within the defined region receive the message.
- No node outside the region participates in the protocol if there are no topological changes.
- The node that initiates the protocol is able to recognize when all the nodes within the span have been probed (termination feedback).
- The algorithm operates correctly amidst topological changes–in the sense that, after a finite sequence of topological changes the algorithm terminates properly.

The protocol execution starts when the initiator wishes to probe the nodes around itself up to y hops away. The protocol is executed in y steps. In the first step, all the 1-hop nodes (those connected to the initiator by a link) are probed; we say that a 1-hop probe has been performed. In the second step, it is the turn of the 2-hop nodes to be probed. This is followed by a succession of increasing-depth probes, during which a minimum-hop spanning tree (with the initiator as the root) is gradually built, until a y-hop probe is completed in the last step.

When the protocol starts a i-hop probe, it takes advantage of the minimum-hop spanning tree built by the previous steps to propagate the new probe-order uptree. Eventually, all the nodes at a distance of $i-1$ hops (they are the tree leaves) receive the probe order. Upon receipt of the probe order, an $i-1$ node probes its neighbors, except the neighbor from which that probe order was received. The probe is always answered by the neighbors. If one of the neighbors has not received any message before it sees a probe message, it is exactly at i hops from the initiator. In this case, the minimum-hop spanning tree is updated by linking this neighbor to the $i-1$ node that sent the probe.

Before starting the next step (an $i+1$-hop probe), the initiator has to know whether the i-hop probe is finished because it has to be sure that an $i+1$ hop node is probed *after* an i hop node. If this is not guaranteed, an i hop node might wrongly believe that it is at $i+1$ hops, which may cause some nodes within the y-hop span not to be probed. To provide the originator with this information, the protocol utilizes feedback in a manner similar to the propagation of information with feedback (PIF) [16]. An acknowledgment message is sent downtree when an $i-1$ hop node has finished its probe, that is, when all the neighbors except the one that sent the probe order have replied to its probe. This acknowledgment is then forwarded downtree only if all the uptree neighbors have sent an acknowledgment. When the initiator has received an acknowledgment from all its neighbors, the initiator is certain the i-hop probe has finished and it begins the next step.

The protocol has been extended to handle topological changes (TCs) that occur during its execution. TCs may be in the form of a breakdown of existing links or the appearance of new ones. TCs can take place in any order. The event of a node's going down or coming up is treated as a corresponding change in the links between the node and its neighbors. Two major problems may occur with changing topology networks:

- A TC may affect the minimum-hop spanning tree built by the protocol. For instance, a broken link on the tree disrupts the message flow, thereby preventing the protocol from terminating. The appearance of a new link may cause some nodes to be closer to the initiator. Then the tree is no longer a minimum-hop tree, resulting in an incorrect protocol execution.

- A broken link in the tree may isolate the whole subtree above the broken link because of the protocol's limited span. Even if there are still paths between the initiator and the nodes in the subtree, these paths may be longer than y hops. Therefore some nodes may dangle forever, believing that they are participating in the protocol. This may be costly in terms of memory resources devoted to the protocol.

To solve the first problem, we base our approach on reporting all the TCs to the initiator. Whenever the initiator gets a TC report, it restarts the protocol in order to build a new minimum-hop spanning tree, in which case we say that a new *cycle* has been started. However, not every kind of TC affects the minimum-hop tree. A TC involving nodes farther than y hops or farther than the current probe depth does not affect the tree. Also, a failing link that is not on the minimum-hop spanning tree does not affect it. In this latter case, the protocol is still correct because the nodes' distances to the initiator are unchanged and because all the protocol messages are forwarded on the tree. Therefore, our approach is based on reporting only the new links, such that the nodes at both ends are within the current probe depth and the link failures on the tree. However, it is not necessary to rebuild the minimum-hop spanning tree from scratch as long as there are no modifications in the subtree of height equal to the distance from the initiator to the closest node involved in a TC. Therefore the rebuilding of the spanning tree in a new cycle starts at that distance, thereby saving time and messages. To implement this solution, it is necessary to be able to differentiate obsolete messages and messages associated with the latest cycle. Thus, each message carries a number that is unique for each cycle: *the cycle number*. Whenever the initiator restarts the protocol, it starts a new cycle by defining a new cycle number, incrementing the previous cycle number by one.

The second problem is solved by a clearing process. When a tree link fails, the node above the failing link starts the propagation of a clearing order. This order will be propagated in the whole subtree above the failing link. When a node receives a clearing order, it resets and releases all the variables that were devoted to the protocol, and it forwards the clearing order.

Some issues are raised when these two solutions are implemented together, because they interact with each other. When a clearing is done in a subtree, the initiator may have already started a new cycle, and new messages may have already reached some nodes of the subtree that are above the failing link. Then the variables of these nodes must not be reset, although it is still necessary to propagate the clearing order so that all the subtree nodes have a chance to get it. However, it is not possible to rely on the tree to propagate the clearing order as the new cycle has destroyed the tree structure! Therefore a clearing order is forwarded to all the neighbors except the one from whom it was received. A clearing order is accepted only if it comes from the preferred neighbor, thus ensuring that the clearing process does not propagate to the whole network.

Upon receipt of the clearing order from the preferred neighbor, a node has to decide whether to forward the message and/or to reset its variables or not. To help make that decision, a clearing order carries the cycle number of the node which has detected the link failure *incremented by one*. It is forwarded if its number is greater than or equal to the node's current cycle number. If the clearing order number is *strictly greater* than the node's current cycle number, the node also resets its variables. This solution may not be sufficient when there are two TCs within a short time span. The clearing order associated with the first TC may not be forwarded by some nodes because their cycle numbers were incremented by two. Therefore, at the end of the protocol, the initiator propagates a final clearing order carrying the latest cycle number. As it is the highest cycle number, the propagation will be done correctly, and all the nodes with old cycle numbers will reset their variables. One might suggest doing this clearing process only once, at the end of the protocol. This approach would not work because a node that is above a link failure may become isolated and never be able to reset its variables. Thus, when a tree link fails, it is necessary to initiate a clearing process in the uptree direction.

5 TOPOLOGY CONTROL

A PRNET must invest resources, such as channel bandwidth, processing, and storage, to acquire and update the routing tables. This overhead is proportional not only to the size of the network (that is, the number of nodes in it) but also to the number of neighbors the nodes have (their *degree*). Such added overhead for neighbors can be discerned in many existing distributed routing schemes that require a node to keep and exchange only distance and next-node information about distant nodes. However, for each of its neighbors, a node must process and broadcast such additional details as link quality and congestion control parameters.

The overhead in PRNETs with a high degree of connectivity may thus be substantial, thereby reducing the network's efficiency. The locations of the radio units are usually determined by the users, hence the physical connectivity is imposed on the network level. This means that in networks with rich connectivity, it may be advantageous to limit the number of neighbors each node maintains full routing information about. Limiting a node, which has too many physical neighbors, to maintain only a subset of its adjacent links for routing, makes the routing algorithm use only a subset of the physical topology, denoted here as the network's *routing* graph. Clearly then, a protocol that reduces the physical topology while still providing a "reasonable" routing topology is needed for these situations, and this is the subject considered.

The objective of this part of the project was to design a topology-reduction protocol and then ascertain its performance. We thus want to limit the physical graph, which represents the network's connectivity, to a subgraph such that:

- The subgraph is connected
- The number of neighbors each node has in the subgraph is not larger than a predetermined bound.

The algorithm for selecting the subgraph is intended for use in dynamic networks where the communication resources are scarce. We are interested in a *distributed algorithm*, whereby each node observes its own physical neighbors and participates in reducing the topology by exchanging coordination messages with its neighbors. The algorithm should also satisfy any constraints imposed by the operating environment. For example, in a distributed network it is hard to guarantee that all nodes will begin operating at the same time. Thus, the algorithm must incorporate any changes and movement within the node population, accommodate late arrivals, and respond to the deletion of nodes from the network. Furthermore, since network protocols for data transfer and network management usually require bidirectional links, it is desirable that the property is achieved by the graph-restriction algorithm, that is, the resulting routing graph employs only bidirectional links.

It seems that an algorithm for solving the problem in its entirety should be a complex one. It is well known that finding a degree-constrained spanning tree is NP complete [6]. Now, although we are not looking for a spanning tree, but rather for a degree-constrained, general-connected subgraph, if we could solve the latter problem in polynomial time, we could solve the former in polynomial time as well, by finding any spanning tree of the degree constrained graph. Because of this inherent complexity, we have taken the approach of devising an algorithm that satisfies one of the constraints (nodal degree), hoping that the resulting routing graph turns out to be connected most of the time [21].

Because it is very unlikely that an efficient distributed algorithm that limits nodal degree while at the same time guaranteeing network connectivity will be found, we decided to examine the performance of a very simple algorithm. The basic function of the algorithm is to have each node select *at random* from its physical neighborhood a number of nodes, no more than its maximum allowed degree, and consider them to be its routing neighbors. This approach is appealing because it lends itself to a distributed implementation in which links are established by communications between neighbor nodes, and each node is responsible for remaining in its maximum routing degree.

However, even though the protocol guarantees that the nodal degree is kept within bounds, we expect that the routing network that results from the random selection will be globally connected with high probability. This is likely because random graphs, which are created by establishing a link between any two nodes with a probability p, have been shown to be connected with high probability for values of p above a certain threshold, and to be disconnected with a high probability below that same threshold [2]. Although the routing graph is created in a different way than the "classical" random graph, (for instance, it has to satisfy nodal degree constraint), and not every node pair may be connected by a link), we expect the same threshold behavior to carry over.

We now describe the procedure for restricting the physical network. We assume that before the algorithm starts, all the network nodes are turned on and each node is aware of its physical neighbors. This assumption is reasonable because the routing algorithm that the nodes participate in requires the nodes to periodically broadcast a packet declaring their presence.

Each node, then, holds a list of its routing neighbors, that is, those neighbors with which it has established a link that is considered to be a part the routing graph. Initially, this list is empty. To add a physical neighbor to its list of routing neighbors, node i considers the list of routing neighbors already acquired. If their number is less than its maximum degree, node i selects one of the remaining physical neighbors at random and initiates a handshake protocol that results in either a mutual agreement between the two nodes to include their link, or a rejection, which prevents the inclusion of that link. Once the handshake protocol terminates, nodes i and j either add each other to their routing neighbor's list, or record each other's refusal to establish a link. Node i continues in this manner, and only terminates when it either achieves the limit of routing degree or has checked all its physical neighbors. The protocol itself then terminates when the last of the nodes terminates.

While executing this protocol, the nodes participate in a routing-table update protocol [7]. In this protocol, only the links of the routing graph are considered, and the resulting internodal distances represent paths completely within that graph.

If this procedure results in a connected graph, the network can function according to the established routing tables. Otherwise, the nodes have to take corrective actions to make the routing graph connected. Thus, if the physical graph is connected but the routing graph is not, some nodes observe physical neighbors that are unreachable by the routing table. That is, only the nodes on the partition line can detect the partition. Those nodes will then attempt to establish links to the disconnected parts, thereby reducing the number of partitioned nodes in the network. Again, this can be done using a handshake protocol. There is, however, a potential difficulty in doing so: if one of the aforementioned nodes has already achieved its maximum degree, it must disconnect a previously established link in order to reconnect the isolated node set. Notice that the disconnected link in turn may cause another node set to be separated from the network, thereby creating a new problem, and possibly an oscillatory behavior of the protocol in which nodes keep partitioning the network in their attempts to reconnect it. However, we do not expect this to pose a problem for networks with rich connectivity.

We have simulated the operation of the algorithm described in the previous section. This simulation consisted of locating N nodes randomly on the unit square. The physical neighborhood was then specified by a number R, such that a node's physical neighborhood was the set of nodes within a circle of radius R around it.

The simulation results revealed an interesting behavior of the topology control algorithm: the connectivity of the logical network as a function of the maximum nodal degree exhibits a threshold effect. If we permit a maximum degree above the threshold to exist, the resulting routing graph is connected with a probability very close to 1. The actual threshold depends on the number of nodes in the network and on the average physical degree, but its value does not vary much—it remains between 3 and 7. This means that as long as the maximum permitted nodal degree is 7 or more, the resulting network is almost certain to be connected for networks of 120 nodes or less. For nodal degree values below the threshold, the simple random-selection algorithm would not be sufficient, and a different approach should probably be taken.

5.1 RELEVANCE AND EFFECTS OF THESE RESULTS:

This relatively new result has been presented to the SURAN community and has been implemented in its SURAP1 protocol. Its performance is expected to be demonstrated in Army testbeds within the next year.

6 CONCLUSION

This project, whose focus was the design of distributed algorithms for large, dynamic packet radio networks, has yielded several such algorithms and protocols. These algorithms fall in the following categories:

- Hierarchical architectures and routing for dynamic networks.
- Multichannel networks—architectures and protocols.
- Dynamic selection of radio FEC code rate to adapt to channel variations.
- Broadcast algorithms that provide for transport of messages to multiple destinations amidst topological changes.
- Self-organizing networks in which the nodes control their local connectivity.

The correctness of some algorithms was validated analytically; others were tested by simulations. All of the algorithms were designed for the environment typical to Army tactical terrestrial networks; that is, where the network has to operate under stress due to jamming and nodal movement and failure. The relevance of algorithms developed in this project to the Army is evidenced by their acceptance by the Survivable Networks (SURAN) program, which is sponsored jointly by DARPA and Army/CECOM. Ideas from this project have also influenced the design of a packet radio network for library automation built by the University of California.

The scientific value of the results of this project can be seen by:

- The papers detailing this project's results published in refereed journals and conferences proceedings.
- References to the these papers in the scientific literature.
- Ph.D. theses based on the models and results developed here.

APPENDIX: PAPERS PUBLISHED UNDER THIS PROJECT

[1] N. Shacham. Organization of Dynamic Radio Networks by Overlapping Clusters: Architecture Considerations and Optimization. In *Proceedings of Performance Symposium*, Paris, France, December 1984.

[2] N. Shacham. Hierarchical Routing in Large Dynamic Ground Radio Networks. In *Proc. of the 18th Hawaii International Conference on Systems Sciences*, Honolulu, Hawaii, January 1985.

[3] N. Shacham and J. Tornow. Future Directions in Packet Radio Technology. In *Proc. of IEEE INFOCOM'85*, Washington, D.C., 1985.

[4] J.J. Garcia-Luna and N. Shacham. Analysis of Routing Strategies for Packet Radio Networks. In *Proceeding of IEEE Infocom*, Washington, D.C., March 1985.

[5] N. Shacham and A. Livne. An Adaptive Hybrid ARQ Algorithm for Radio Channels. In *Proceedings of the International Conference on Communications*, Chicago, Illinois, June 1985.

[6] N. Shacham. Adaptive Link Level Protocols for Packet Radio Networks. In *Proceedings of INFOCOM'86*, Miami, Florida, March 1986.

[7] N. Shacham and P.J.B. King. The Performance of Multichannel, Multihop Packet Radio Networks. In *Proc. of ICC'86*, Toronto, Canada, 1986.

[8] N. Shacham. Dynamic Parameter Selection in Packet Radio Transmission. In *Proc. of MILCOM'86*, Monterey, California, 1986.

[9] N. Shacham and J. Westcott. Future Directions in Packet Radio Architectures and Protocols. *Proceedings of the IEEE*, January 1987.

[10] N. Shacham and P.J.B. King. Architectures and Performance of Multichannel, Multihop Packet Radio Networks. *IEEE Journal on Selected Areas in Communications*, July 1987.

[11] N. Shacham and R. Ogier. Network Control and Data Transport for C^3I Applications. In *Proceeding of IEEE MILCOM'87*, Washington, D.C., 1987.

[12] N. Shacham, E.B. Brownrigg, and C.A. Lynch. A Packet Radio Network for Library Automation. In *Proc. of IEEE MILCOM'87*, Washington, D.C., 1987.

[13] R. Kung and N. Shacham. A Distributed Algorithm for Reliable Message Delivery Over a Sub-Network. In *Proc. of Globecom'87*, Tokyo, Japan, 1987.

[14] M. Frankel, N. Shacham, and J. Mathis. Self-Organizing Networks. In *International Conference on Frontiers of Computing*, Amsterdam, the Netherlands 1987.

[15] N. Shacham and D. Beyer. A Distributed Protocol for Reducing Neighborhood Size in Radio Networks. In *Proc. of ICC'88*, Philadelphia, Pennsylvania, 1988.

REFERENCES

[1] D. Chase. Code Combining. In *Proc. of MILCOM*, Los Angeles, Ca, October 1984.

[2] M.S. Chen. Shortest Path in Communication Networks. In *Proc. of IEEE Globecom*, Houston, Tx., 1987.

[3] P.M. Feldman and V. Li. Delay Analysis for Adaptive Hybrid ARQ Protocols. In *Proc. of MILCOM'87*, Washington, D.C., 1987.

[4] M. Frankel, N. Shacham, and J. Mathis. Self-Organizing Networks. In *International Conference on Frontiers of Computing*, Amsterdam, 1987.

[5] J.J. Garcia-Luna and N. Shacham. Analysis of Routing Strategies for Packet Radio Networks. In *Proceeding of IEEE Infocom*, Washington, D.C., March 1985.

[6] M.R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Company, 1979.

[7] J. Jubin and J. Tornow. The DARPA Packet Radio Network Protocols. *Proceedings of the IEEE*, 75(1):21–32, January 1987.

[8] L. Kleinrock and F. Kamoun. Hierarchical Routing For Large Networks. *Computer Networks*, 1():155–174, January 1977.

[9] L. Kleinrock and J. Silvester. Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number. In *Proc of Hawaii Internt'l Conference On System Sciences*, Honolulu Hawaii, January 1979.

[10] R. Kung and N. Shacham. A Distributed Algorithm For Reliable Message Delivery Over a Sub-Network. In *Proc. of Globecom'87*, Tokyo, Japan, 1987.

[11] S. Lin and P.S. Yu. A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels. *IEEE Transactions on Communications*, COM-30(7):1701–1719, July 1982.

[12] C. Lynch and E.B. Brownrigg. *Packet Radio Networks: Protocols, Technologies and Applications.* Pergamon Press, 1987.

[13] J. McQuillan, E. Rosen, and Richer. The New ARPANET Routing Algorithm. *IEEE Transcations on Communications*, COM-28(5):711–719, May 1980.

[14] A. Nilsson, W. Chou, and C.J. Graff. A Packet Radio Communication System Architecture in a Mixed Traffic and Dynamic Environment. In *Proceedings of Computer Networking Symposium*, pages 51–66, Gaithersburg, Maryland, 1980.

[15] M. Schwartz. *Telecommunication Networks*. Addison Wesley, 1987.

[16] A. Segall. Distributed Network Protocols. *IEEE Trans. on Inf. Theory*, IT-29(1):23–35, January 1983.

[17] N. Shacham. Adaptive Link Level Protocols For Packet Radio Networks. In *Proceedings of INFOCOM'86*, Miami, Fl., March 1986.

[18] N. Shacham. Dynamic parameter selection in packet radio Transmission. In *Proc. of MILCOM'86*, Monterey, Ca., 1986.

[19] N. Shacham. Hierarchical Routing in Large Dynamic Ground Radio Networks. In *Proc. of the 18th Hawaii International Conference on Systems Sciences*, Honolulu, Hawaii, January 1985.

[20] N. Shacham. Organization of Dynamic Radio Networks By Overlapping Clusters: Architecture Considerations and Optimization. In *Proceedings of Performance Symposium*, Paris, France, December 1984.

[21] N. Shacham and D. Beyer. A Distributed Protocol for Reducing Neighborhood Size in Radio Networks. In *Proceedings of IEEE ICC'88*, Philadelphia, Pa., 1988.

[22] N. Shacham, E.B. Brownrigg, and C.A. Lynch. A Packet Radio Network for Library Automation. In *Proc. of IEEE MILCOM'87*, Washington, D.C,, 1987.

[23] N. Shacham and P.J.B. King. Architectures and Performance of Multichannel, Multihop Packet Radio Networks. *IEEE Journal on Selected Areas in Communications*, July 1987.

[24] N. Shacham and P.J.B. King. The Performance of Multichannel, Multihop Packet Radio Networks. In *Proc. of ICC'86*, Toronto, Canada, 1986.

[25] N. Shacham and A. Livne. An Adaptive Hybrid ARQ Algorithm for Radio Channels. In *Proceedings of the International Conference on Communications*, Chicago, Ill., June 1985.

[26] N. Shacham and R. Ogier. Network Control and Data Transport For C^3I Applications. In *Proceeding of IEEE MILCOM'87*, Washington, D.C., 1987.

[27] N. Shacham and J. Tornow. Future Directions in Packet Radio Technology. In *Proc. of IEEE INFOCOM'85*, Washington, D.C., 1985.

[28] N. Shacham and J. Westcott. Future Directions in Packet Radio Architectures and Protocols. *Proceedings of the IEEE*, January 1987.

[29] W.D. Tajibnapis. A Correctness Proof of a Topology Information Maintenance Protocol for a Distributed Computer Network. *Comm. of ACM*, 20(7):477–485, July 1977.

END
DATE
FILMED
7-88
DTIC