EVALUATION OF ADAPTIVE ROUTING PROTOCOLS FOR PACKET RADIO NETWORKS

Sharada V. Vitalpur

ITT Aerospace/Optical Division
Fort Wayne, Indiana 46801

ABSTRACT

Current trends in C^3I call for automated and adaptive network control capabilities. Such advanced techniques are required to handle increasing traffic demands while ensuring a high degree of reliability and survivability of the network. In this paper, two separate adaptive algorithms for routing are developed and evaluated for use in Packet Radio Networks. The first algorithm dynamically assigns routes from source to destination based upon update information received from a node's neighbors and current link quality calculations. The second algorithm makes use of routing groups to forward packets. The routing group update algorithm is designed to keep routing groups current as the network configuration changes.

The two routing algorithms as well as other protocols for distributed control are being evaluated with a Packet Radio Network Simulation Model. This model is a dynamic simulation where the protocols and algorithms implemented allow the network to automatically adapt to any changes in its topology. The simulation sets up a fully distributed network and explicitly models the unique features of a specific combat net radio, the propagation, queuing and processing delays, and the protocols implemented. The simulation model is used as a basis for evaluating the performance of the routing algorithms as well as the overall network behavior.

EVALUATION OF ADAPTIVE ROUTING PROTOCOLS FOR PACKET RADIO NETWORKS

Sharada V. Vitalpur

ITT Aerospace/Optical Division
Fort Wayne, Indiana 46801

I. INTRODUCTION

The main motivation behind network and protocol design is to formulate a set of algorithms that, when implemented, not only optimize network performance but also allow the network to adapt to changes in its topology while ensuring a high degree of reliability and survivability of information transfer. While guidelines for these protocols are somewhat standardized, there is still a wide range of flexibility on the actual algorithm implementation. In this paper, we are interested in communication networks that cover wide geographical areas with interconnected nodes. In such a network, it is highly unlikely that all nodes are within direct communication range of each other. Instead, packets have to be forwarded from originating nodes through intermediate nodes to the end destination. This necessitates the implementation of a protocol to route packets from every source to destination node pair.

Routing a packet of data involves the question of not only selecting a set of nodes to forward the packet through but also of how to physically send the packet. A routing protocol addresses both of these issues. The first issue falls under the Network Layer of the Open Systems Interconnection (OSI) Seven Layer Reference Model while the second falls under the Data Link Layer. The OSI Seven Layer Reference Model is given in Fig. 1.

Most computer networks use shortest-path routing algorithms to route packets from source to destination. Two of the most commonly used algorithms are variations of Ford's algorithm (Ref.4) or Dijkstra's algorithm (Ref.5). Variants of Ford's algorithm are simple and easily implemented in a distributed environment. The major disadvantage in this class of algorithms is the looping problem. Routing tables may start to form loops for certain destinations when intermediate link costs to that destination increase. The class of variants of Dijkstra's algorithm provides a faster rate of convergence but has a high overhead in terms of memory requirements for actual implementation in a distributed packet radio network.

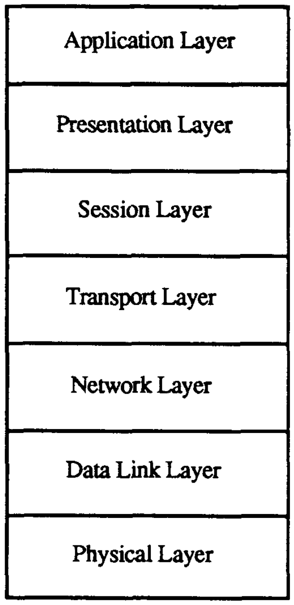

Fig. 1 OSI Seven Layer Reference Model

A packet radio not only has memory limitations; it is dynamic and frequently undergoes changes in topology and quality of links as operating conditions change. In the packet radio network (PRNET), a node can only receive one collision free packet at a time, and the relative network size is small. Based on these restrictions and shortcomings of both Ford's and Dijkstra's algorithms, we are proposing new routing and packet forwarding algorithms for the PRNET. New routing table update and packet forwarding algorithms have been designed for a PRNET under the SINCGARS Packet Overlay Program by SRI International for CECOM. An extension of the packet forwarding protocol called Dynamic Shortest Path Incremental Routing (DSPIR) has been developed at ITT Aerospace/Optical Division (ITT-A/OD) and is currently being evaluated.

A dynamic simulation model is used to evaluate network performance of PRNETs and their protocols. This paper will briefly describe the routing table update algorithm, the two packet forwarding protocols, the simulation model and the performance of DSPIR for a dynamic network.

II. NETWORK DESCRIPTION AND ASSUMPTIONS

The Packet Radio Network is a highly mobile network that must be designed to provide a very survivable and a reliable medium for communication in a tactical environment. Since nodes enter and leave the network almost at will, the protocols implemented in the PRNET must provide for distributed network control. Distributed network control permits local network reconfiguration without significantly affecting global connectivity or deteriorating network performance.

In this paper the radio assumed in the PRNET is a combat net radio employing spread spectrum techniques. The use of a frequency-hopping radio in the network provides several important capabilities. The most significant capability is multi-channel access in which more than one radio within the same communication range can be concurrently transmitting as long as they don't initiate transmission within a vulnerability window of one another.

The packet applique has the capability of decoding the packet header and the data separately. This allows the radio to break capture, i.e. stop reception of an incoming signal. Upon synchronization with a particular transmission, decoding the header, and determining that the packet is not intended for it, the radio can stop packet reception and be ready to initiate a transmission or receive another packet.

Utilizing these features of the PRNET, routing algorithms for the network have been designed and are presented in the following sections. In designing these routing algorithms, the network is modeled as an undirected graph of G = (N,E), where N is the set of nodes and E is the set of links. All links are assumed to be bidirectional and each node has a unique identifier. A Stop-Wait protocol is used at the Data Link Layer to forward a packet. For each data packet transmitted, a node receives either an active or passive acknowledgement from its neighbors (Ref.2). Periodic update packets are used in the network to update routing, distance, and neighbor tables.

III. OVERVIEW OF ROUTING ALGORITHM

The Routing Table Update (RTU) algorithm, designed and validated by SRI International under the SINCGARS Packet Overlay Program for CECOM (Ref.1), uses an update propagation technique similar to the one used by variants of Ford's algorithm but avoids the counting to infinity problem. The RTU algorithm makes use of the information contained in the periodic update packets transmitted by each node. Each node maintains four tables; a distance table, a neighbor table, a routing table and a routing group table. The routing group table (RGT) is used for packet forwarding using Sibling Duct Routing (SDR) and its contents will be described in a later section. A distance table at Node j is an $|N| x |N_j|$ matrix where N_j is the set of neighboring nodes of j. For every k in N_j, the table consists of the next node in shortest path from j to l through k, a flag specifying whether this path has increased since last update, quality of path to l through k and length of the shortest path. The link quality is a function of the number of transmissions reported by the neighbor, the number of correct receptions at node j and how current the information is.The neighbor table specifies for every known neighbor of j, the node id, forwarding delay, pacing delay, number of packets received correctly from neighbor since last update, running count of packet transmissions reported by neighbor, and quality of link with neighbor. The routing table at node j is a column vector of $|N|$ rows where N is the set of active nodes in the network having a path to j plus j itself. The row corresponding to k consists of the shortest path from j to k, the next node in shortest path from j to k, time entry was last updated, flag specifying whether the distance to j increased during last update and a flag specifying quality of radio link to k.

The network initializes itself in a distributed manner as nodes transmit and process periodic update messages. These periodic updates consists of the nodes' routing table and any changes to its other tables since its last update transmission. When a node, say j, is first powered up, it initializes its tables which entails creating empty tables with a self entry for node j itself. The node transmits an update message and lets its neighbors know of its presence. The periodic updates are broadcast to the network with a specific id in the packet header which indicates to the receiving nodes that it is an update message. A node decodes all update messages it receives. Paths to newly detected neighbors are denoted as bad until certain link quality is achieved. When a node detects a node failure, it changes its tables and accordingly sends an update message. A formal description of the RTU algorithm along with examples of operation are presented in (Ref.1).

This routing table update algorithm is used in SDR and DSPIR packet forwarding protocols. In the remainder of this section, we shall present Sibling Duct Routing (SDR) and Dynamic Shortest Path Incremental Routing (DSPIR).

Sibling Duct Routing

Due to the low data rate of the SINCGARS radio in the PRNET, packet retransmissions have to be minimized to achieve reasonable throughput and end to end delays. The SINCGARS radio provides less radio coverage for correctly decoding the packet than for achieving packet synchronization. Using broadcast techniques to forward packets in the network would cause too many nodes to synchronize onto transmissions not intended for them substantially reducing network throughput. Therefore, a forwarding protocol is designed that optimizes network performance while reducing the need for packet retransmissions caused by the dynamic nature of the network. This protocol maintains the number of repeaters per packet to a minimum.

This routing protocol is based on the idea of duct routing, which was first introduced by Shacham, etal. (Ref.3). In duct routing, a node that receives a packet not destined for it checks the header of the packet to see if the distance of the sending node to the destination is more than its distance to the destination. If its distance is smaller, the node schedules the packet for transmission in its transmit queue. If the node hears another node forwarding the packet before it transmits the packet, it drops the packet from its transmission queue. But, a problem arises when nodes having the same distance to the destination, and are neighbors of the source, can't hear each other. The probability of collisions and duplicate packets in the network increases thus, reducing network throughput. This is known as the hidden terminal problem and is shown in Fig. 2.

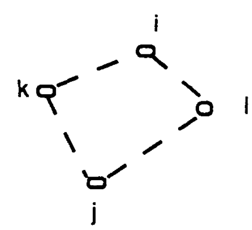

Fig. 2 To transmit packet from node *i* node *j*, node *i* transmits packet and both node *k* and *l* try to forward it since they can't hear each other.

SDR builds upon duct routing by specifying routing groups in addition to the destination and distance to destination in the packet header. A routing group (RG) is a set of nodes that can hear each other and are the same distance to the destination. A given destination can have one or more RGs. By requiring that the members of a particular routing group hear each other, the hidden terminal problem is eliminated and the number of repeaters per packet is reduced. Since there is a group of nodes that can transmit the packet, packet forwarding is guaranteed if at least one of the nodes in the group receives the packet. This minimizes the need for retransmitting the packet due to packet collisions and the intended receiver being busy at time of transmission.

Each node keeps track of the RGs it defines and those defined by its neighbors of which it is a member of. A RG to forward a packet from node i to node j is formed in the following manner [1]:

- o from routing table, the distance from node i to node j, D_{ij} is found.
- o using the distance table, all neighbors whose distance to j is $(D_{ij} - 1)$ are placed in set **B**
- o for each k in **B**, form set A_k such that:

  $$l \in A_K \text{ if } D_{kl} = 1 \text{ and } D_{lj} = D_{ij} - 1$$

- o routing group formed to route packet from node i to node j is the intersection of the A_k

  $$RG_{ij} = \cap A_k$$

For example in Fig. 3, to forward the packet from node i to node j, the RG formed at node i contains {a, b, c}. The RGs at node a, node b and node c are {e}, {f,d}, and {d} respectively. Note the RG at node a might as well have been {f} but not {e,f} since nodes e and f don't hear each other. The last RGs to forward the packet to node j are {h}, {h,g}, and {g} and are defined at nodes $e, f,$ and d.

The entries and changes to the RG table is also part of a node's periodic update message. The RG table is updated when the node detects a change in the topology which affects its routing or distance tables. The main advantage of SDR is that forwarding packets does not entirely depend on the information maintained at any one node or the decisions made by a central node thus making it highly attractive for distributed network management.

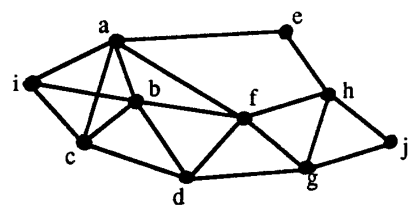

Fig. 3 General Network Topology

Dynamic Shortest Path Incremental Routing

The Dynamic Shortest Path Incremental Routing (DSPIR) protocol is based upon existing shortest-path routing algorithms. It is different in the way routes are updated and alternate routes are assigned. This makes it an adaptive algorithm which senses changes in network topology not only with respect to node additions and deletions but also with regard to link conditions and quality of links. Link quality as pertaining to DSPIR is defined below. The looping to infinity problem of variants of Dijkstra's algorithm is avoided by using the RTU algorithm previously described. The network dynamically adapts to changes in network conditions ie. in connectivity, node additions and deletions, interference and delay in the propagation of update information through the network. This protocol selects routes that maximize the link quality while minimizing the forwarding delay.

Link quality to a destination is updated every time an update is received from a neighbor that is a next hop to the given destination. The quality of the link is a function of: the number of transmissions that the neighbor reports it has made and the number of correct transmissions the node has received from neighbor since last update; the pacing delay for a particular neighbor; and how long since last update from neighbor. Based on whether the link quality is above or below a certain threshold, that link is declared as either good or bad. Since drastic changes in link conditions can occur in a very short time span due to changes in terrain or other operational conditions, the algorithm to determine link quality weighs the newly calculated value with the history accumulated over a period of time. This prevents drastic variations in the link quality calculations and makes the algorithm more robust.

When a node is started up, it initializes its tables using the RTU algorithm. Its tables are filled as updates from neighboring nodes are received. When a new shortest path is detected, it is kept as an alternate route until the quality of that route can be denoted as "good". Alternate routes are used to forward packets when the primary route fails to acknowledge a packet either actively or passively. A node may not acknowledge a packet if the node has crashed, moved out of communication range or its transmit queue is full. To forward a packet, the node first tries to send the packet through the primary route. If it doesn't hear the next node forwarding the packet and its pacing time (Ref. 2) expires, the node retransmits the packet. Pacing is a method of flow control used in the PRNET and will be described in more detail in a later section. If after M tries the packet is still not acknowledged, the node looks into its routing table for an alternate route and changes the packet header to forward the packet through the alternate route. The value of M needs to be optimized through simulation and analysis. An alternate route is declared as a primary route if:

- o the primary route hasn't been heard for T seconds where T is a node time-out period
- o the link quality to destinations using the primary route is below the set threshold.
- o the next node in the primary route has a pacing delay which exceeds a maximum allowable forwarding delay. This indicates that the node's transmission queue is large.

The last condition is only applicable when using adaptive pacing.

This protocol does not rely entirely on the routing decisions made at any one node and is highly adaptive to changes in the operating conditions. By slightly modifying this protocol, even the need for bidirectional links maybe overcome. The main advantage for this protocol is that the overhead for its implementation is very low yet it provides an efficient and reliable means for forwarding packets in a PRNET.

Comparison of SDR and DSPIR

Both routing algorithms are suitable for a packet radio network in terms of reducing the number of retransmissions, avoiding loops and reducing forwarding delay. But for SDR to be effective, the network should be fairly large. The members of a routing group have to be within communication distance of each other and a routing group will consist of more than one member if and only if there are two or more nodes with the same distance to the intended destination and can hear each other. For this reason, SDR can only achieve better network performance than other forwarding protocols when the network is densely populated.

Formation and maintenance of RGs and RG tables increase the complexity of the algorithm and delays the packet forwarding decision process. The maintenance of RGs for a network adds a significant amount of overhead, not only in terms of memory but also with respect to having to send additional organizational information to maintain and update RG tables. This additional overhead combined with the existing overhead of higher level protocols such as Telnet and Internet, will drastically reduce network throughput.

If the network is loosely populated and/or its size is small, SDR protocol collapses to incremental routing where each node decides the next node to forward the packet to based on shortest-path considerations. In essence, this is just a simpler form of DSPIR. DSPIR doesn't require any additional information to be maintained at each node or transmitted in the packet header. DSPIR has to make the added calculation of the link quality as previously described and of alternate routes which entails another search through the routing and distance table. The problem of having more than one node forward the packet is overcome by the fact that the intended next hop is already specified in the packet header and any other node receiving the packet will break radio capture after decoding the header and realizing that the packet is not intended for it. This frees up those nodes to either receive another packet or transmit one. In SDR since a group of nodes are specified to forward the packet, all those nodes have to receive and decode the packet when only one of those nodes will probably forward it.

Based on the expected network size for a PRNET in a tactical environment, the additional complexity and overhead, and memory constraints, it can be concluded that SDR is not the protocol that would optimize network performance. Therefore, SDR is not considered for further evaluation using the simulation model. DSPIR is modeled and evaluated using the simulation model. The simulation model, the protocols implemented and some simulation results are presented in the following sections.

IV. SIMULATION MODEL

The evaluation and analysis of network performance and protocol operation is important in the process of optimizing the protocols. This can be done analytically or by use of rapid prototyping. Due to the complexity of the network and its protocols, a combination of analytical modeling and rapid prototyping is used to accurately simulate a PRNET. The simulation model provides a platform to study and characterize network performance under different topology and operation constraints. The protocols can be evaluated and optimized in a pseudo-realistic situation.

A static network was first simulated and the performance of the network was evaluated using a shortest-path incremental routing and an adaptive pacing algorithm. The purpose of the pacing algorithm is to provide for network data flow control. If a node is heavily congested, there is a significant forwarding delay through that node and increases retransmissions which in turn affects network throughput. By using a pacing algorithm, transmissions are scheduled such that it permits packets downstream to leave the network and release network resources. A detailed description of the static topology simulation model, the protocols and parameters implemented and results of the analysis is presented in (Ref.2).

Using the static network simulation model as a guideline, the dynamic topology model has been implemented. The simulation model has been developed using SIMSCRIPT II.5 simulation software package by CACI. The simulation uses graphics and animation techniques to actually demonstrate network operation as the simulation executes. This model explicitly simulates the network configuration process which is the actual initialization and formation of the tables by means of periodic update messages. It also explicitly models the RTU algorithm, DSPIR packet forwarding algorithm, constant and adaptive pacing algorithm, and all queuing, processing, acknowledgement, and propagation delays. The radio synchronization and break capture capabilities are also implemented in the simulation. Network loading is modeled as a Poisson process with packets generated for random origin-destination pairs. At the start of the simulation, the initial network topology and protocol parameters are specified by the user into the model. Statistics for End To End (ETE) transmission delay, system throughput, number of packets lost, etc. are maintained throughout the simulation.

V. SIMULATION RESULTS

A general network topology is selected to evaluate the routing protocol. Since the network is dynamic as well, nodes enter and leave the network throughout the duration of the simulation. This allows us to evaluate how the protocols adapt to changing network conditions.

Network performance is evaluated using both adaptive and constant pacing algorithms. we are primarily interested in system throughput, ETE delays and the robustness of the network. The simulation results for the dynamic network is compared with that of a general static network. The system throughput and ETE delays are measured as a function of system loading. The results for the dynamic network using constant pacing is presented in Fig. 4, and adaptive pacing is presented in Fig. 5. Comparing these results to that of a general static topology network, Fig. 6, it can be seen that the system throughput for the dynamic network does not achieve the same maximum value as that of the static network. the additional overhead associated with periodic updates and delays in propagation of update information reduces the maximum achievable throughput level. However, the results show that the ETE delays are reduced for the dynamic network. The performance of the dynamic network is much better when using an adaptive pacing algorithm as opposed to a constant pacing value. The throughput curves also indicate a more flat curve indicating a broader region of stable operation.

In conclusion, the routing and pacing protocols used in the dynamic topology network make the network more robust and reduces the ETE delays associated with packet forwarding. The deterioration in

Simulation Results For Dynamic Network With Constant Pacing

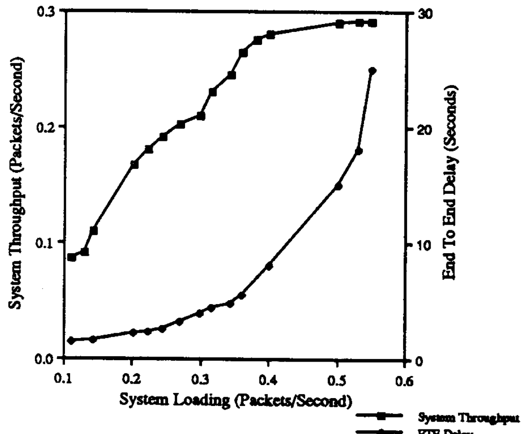

Figure 4.

Simulation Results For Dynamic Network With Adaptive Pacing

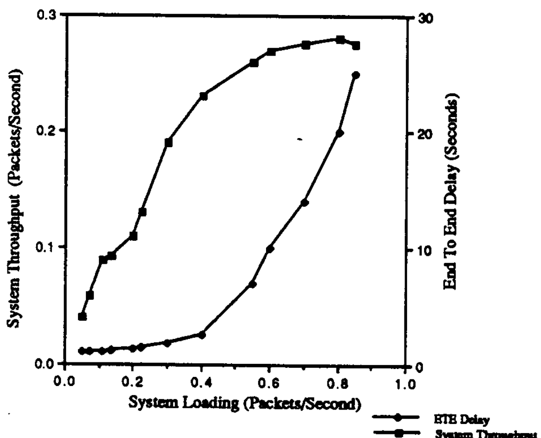

Figure 5.

Simulation Results for Static Network With Adaptive Pacing

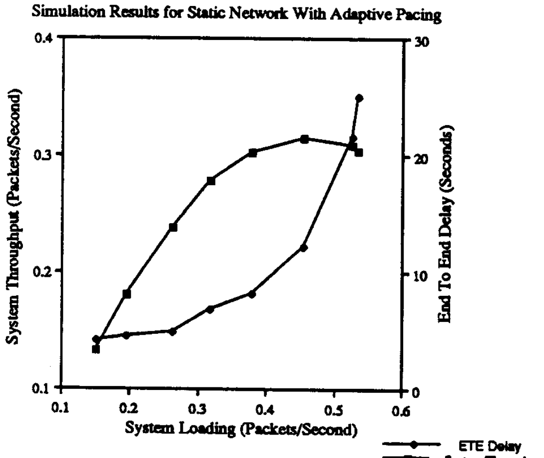

Figure 6.

system throughput performance can be attributed to the additional overhead involved in maintaining a distributed dynamic network.

VI. CONCLUSIONS

The routing table update algorithm presented can be used with either Sibling Duct Routing or Dynamic Shortest Path Incremental Routing which are protocols designed for packet forwarding. The added overhead, complexity and requirements for a densely populated network imply that SDR is not a viable alternative for packet forwarding in a PRNET. The DSPIR adaptively forwards the packet by making use of link quality and forwarding delay information. A dynamic simulation is developed and the protocols are modeled. Results of the simulation are presented and it is shown that the RTU algorithm and DSPIR packet forwarding protocol function as desired but due to the added overhead involved in transmitting update information, the network performance is not as good as that of a static network.

VII. ACKNOWLEDGEMENTS

The author would like to acknowledge and thank Mr. Mark. Soderberg for his efforts in the implementation and evaluation of the dynamic network simulation model.

VIII. REFERENCES

[1] SINCGARS Packet Switch Overlay, Design Plan, prepared by ITT-A/OD and SRI International for CECOM, Contract No. DAAB07-85-C-K581, 1986

[2] S. V. Vitalpur, "Protocol Analysis and Design for CNR Packet Radio Networks" TCC -88 Conference Proceedings, Ft.Wayne, IN., May 1988

[3] N. Shacham, E.J. Craighill and A. Poggio, "Speech Transport in Packet-Radio Networks with Mobile Nodes", IEEE Journal on Selected Areas in Communications, Vol.SAC-1, No. 6, December 1983.

[4] L. R. Ford, Jr. and D. R. Fulkerson, Flows in Networks, Princeton, NJ; Princeton University Press, 1962.

[5] E. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mithematik, Vol. 1, pp 269 271, 1959.