NETWORK ROUTING TECHNIQUES AND THEIR RELEVANCE TO PACKET RADIO NETWORKS

James Geier   Martin DeSimio, WB8MPF   Byron Welsh, KD8WG

Air Force Institute of Technology
Department of Electrical and Computer Engineering
Wright-Patterson AFB, OH 45433

Abstract

With packet radio networks, the distance between source and destinat.ion nodes typically necessitates one or more nodes to relay data to the final destination. Thus, some form of routing must. take place. This paper explains several current network routing algorithms and shows their relevance to packet, radio networks. In addition, current research at. AFIT concerning the development of an automatic routing algorithm for Air Force Logistics Command's (AFLC) HF packet radio network is explained.

1 INTRODUCTION

Most routing algorithms store node address information in tables, which show the next node to receive a packet. These routing algorithms may be static or dynamic. If the algorithms are static (nonadaptive) the table entries do not change during normal operation of the network. Dynamic (adaptive) routing algorithms periodically update the tables to reflect. changes in the network's topology or utilization or both [6].

Since packet radio networks typically have changeable connectivity, the routing mechanism must be capable of updating routing tables. This paper examines three basic types of adaptive routing algorithms: centralized, distributed, and isolated [7]. The final section describes current research of an automatic routing algorithm for AFLC's HF packet. radio network.

2 CENTRALIZED ROUTING

A centralized routing algorithm requires a routing control center (RCC) to make routing decisions based on information gained from each node within the network. Each node monitors connectivity and delay metrics among neighboring nodes and periodically sends this information to the RCC. The RCC calculates the best route (normally in terms of least delay) and sends each node new routing table information depending* on the most recently measured state of the network.

In most cases, a source node needing to send data packets can notify the RCC of the source and destination. The RCC will respond with a special call request packet called a needle packet, that contains the route which is the most efficient circuit. The route is specified as an ordered set of nodes. The source node then sends the needle packet through the network to establish the circuit, and then data packets can follow.

Although centralized routing offers a solution to adaptive routing, this technique has disadvantages worth discussing. For one, centralized routing requires a large amount of overhead due to routing information sent between nodes and the RCC. As a result, centralized routing may not be suitable for some networks operating with limited bandwidth, such as HF packet. radio. Also, large networks having many nodes will require the RCC to perform lengthy calculations to determine optimum routes. Hence, the "optimum" table entries may not be valid if the network topology changes rapidly.

3 DISTRIBUTED ROUTING

With distributed routing, each node distributes routing metrics (connectivity information, node delays) throughout the network, enabling other nodes to update routing tables. Distributed routing has proven to be very robust with ARPANET (Advanced Research Project Agency Network). Within ARPANET, each node periodically measures the delay to each node within one transmission hop and puts this information into a status packet. Nodes within one transmission hop are known as neighbors. The node then transmits the status packet to each neighbor, which records the status information. Each neighbor repeats the delay measuring process, formulates a status packet containing local delay information as well as delay information from incoming status packets and sends this status packet. to each of its neighbors. By having all nodes follow this process, each node will eventually have an overall "picture" of the network in terms of node-to- node delays. Each of the nodes can then determine the route of least delay by referring to the status information received from other nodes.

Distributed routing requires a significant amount of overhead with packet radio networks having highly mobile nodes. Each node must send status information often enough to account for rapid changes in topology due to node movement. Hence, the network must have a considerable amount of bandwidth available or suffer from rather low throughput.

Jubin and Tornow explain that. the DARPA (Defense Advanced Research Project. Agency) packet radio network (not ARPANET) applies distributed routing techniques by having each node maintain a tier table [2]. The tier table specifies nodes that are one hop away (tier 1), two hops away (tier 2), three hops away (tier 3), and so on. The tier table is arranged in a matrix format. The tiers represent. the rows, whereas the tier 1 entries head off the columns. For example, node X could have a tier table as shown in Figure 1. Here, nodes A, B, and C are neighbors of node X; nodes D and E are neighbors of node A; node N and J are neighbors of node D; node F is a neighbor of node B, and so on. If node X has a packet needing transmission to node F, node X would choose to send the packet to node F via node B because the transmission would take only two hops in comparison to three hops if sent via node C. The tier table gives information regarding connectivity among the nodes; therefore, software at the source node may use the tables to make effective routing decisions.

The nodes update their tier tables in the following manner. Every 7.5 seconds, each node transmits a Packet. Radio Organization Packet (PROP) that announces the node's existence and includes a copy of its tier table. A node receiving a PROP simply includes the information in its own tier table. After a period of time, all nodes in the network will have complete connectivity information for the entire network.

Very high frequency (VHF) packet networks established by amateur radio operators also employ distributed routing techniques. Amateurs use an approach similar to the DARPA packet radio network described above. Each node in the network periodically broadcasts information concerning the nodes it has connectivity to. Included in the broadcasts is not only node topology information, but also a relative measure of route quality within the network. The route quality between two nodes is calculated based on the type of link between the nodes in question. Factors that affect node quality include the baud rate supported by a link as well as whether it is a radio or a direct hardware link. Overall link quality is calculated by multiplying the qualities of each link traversed in the path from the source to the destination. A node will choose the route having the highest, quality which insures that the route used will support the highest baud rate possible as well as the route with the least. number of hops. Nodes are also automatically deleted and added to the network via the same broadcasts [5]

4 ISOLATED ROUTING

Isolated routing allows a node to make routing decisions without. reference to an RCC or other nodes. Three classical approaches of isolated routing are flooding, hot. potato and backward learning. [7]

4.1 Flooding

With the flooding technique, a node with a packet. to send initializes a hop counter in the packet, header to a count of zero and transmits the packet, out every outgoing link. With packet radio networks, this means that. the node simply broadcasts the packet. Each node within the network examines incoming packets, and if the packet, has not reached its destination and the hop counter has not reached a predetermined limit, the node increments the counter and rebroadcasts the packet. The hop counter maintains the stability of the flooding routing algorithm. Without the hop counter, nodes would continue to retransmit@ packets indefinitely. Since every node follows the flooding process, the packet. will eventually reach its destination.

Flooding is effective as a routing technique with networks having rapidly changing connectivity. This reasoning stems from the fact that flooding does not require routing updates as the network changes. Even though flooding has overhead in the form of redundant, data packet transmissions, the amount of overhead saved due to the absence of routing table updates outweighs the amount of overhead flooding generates in retransmissions for systems with dynamic topologies. However, with static topologies the opposite condition occurs; the amount of overhead due to redundant transmissions is greater than routing table updates because static topologies do not require as many routing table updates.

Flooding, because of its nature, offers the shortest end-to-end packet delay when compared to other routing techniques. Since control packets normally require expedient. delivery, even packet radio networks having static topologies use flooding to disseminate urgent, control information, regardless of the fact that it will generate higher overhead.

4.2 Hot Potato

The hot. potato routing method bases routing decisions on the current availability of transmit queues within the node. Pure hot potato will put an incoming packet in the outgoing queue having the least number of packets waiting for transmission. Thus, the current. node treats the packet, as a "hot potato" by getting rid of it as soon as possible, without regard to where the packet goes next. Eventually, the packet. will reach its destination.

Static routing, which uses routing table entries that do not change, can effectively utilize hot. potato by putting a packet, in the second best (or lesser) choice output queue if the best choice output. queue has more than a certain number of packets waiting for transmission. As a result, congestion will be kept to a minimum on the best, choice lines.

4.3 Backward Learning

The backward learning approach assumes data packet headers contain source node addresses and hop counters. The source node initializes the hop counter to zero before transmitting the packet, and intermediate nodes increment the hop counter by one before retransmitting the packet. For the purpose of updating routing tables, each node constantly monitors the incoming packets by noting the packet's original source address, hop count, and address of the immediately preceding node. With this information, a node can make educated decisions on which node to send outgoing packets to for delivery to a specific location. A node transmits a packet. to the neighbor where packets came from with the least hop count and originated from the desired destination node.

The main advantage of the backward learning technique is less overhead because the data packets themselves carry the routing information; bandwidth-consuming routing control packets do not need transmission. Of course, the backward learning algorithm assumes network conditions are approximately the same in both directions. Also, the routing tables will only be accurate if data are sent between the nodes frequently because the data packets are responsible for carrying the routing information.

The choice of routing algorithm for a packet radio network depends greatly on the mobility of the nodes and the available system bandwidth. If the network incorporates highly mobile nodes, the flooding technique is desirable because the changing topology, when compared to a static one, will not cause the routing algorithm to generate additional overhead. If the network is somewhat stationary, then a distributed routing method works effectively because periodic status transmissions can enable the updating of tables as node availability changes due to changes in radio wave propagation. On the other hand, a network wit.11 highly mobile nodes and limited bandwidth may benefit from a backward learning technique if there are frequent data transmissions between nodes. Whichever routing algorithm is chosen, it should be the most efficient in terms of least delay.

5 CURRENT RESEARCH AT THE AIR FORCE INSTITUTE OF TECHNOLOGY [1]

The Air Force Institute of Technology (AFIT) is continuing in the development of an HF packet. radio network for Air Force Logistics Command (AFLC) for the transmission of text messages during wartime contingencies and natural disasters. This network will eventually consist, of a packet radio station (node) at each of AFLC's Air Logistics Centers and several portable units. AFLC expects to use the packet. radio system during wartime if other message systems fail because HF packet radio architecture, due to relatively low operating frequencies, lends itself well to post-nuclear attack. In addition, the ease of mobility with packet. radio nodes makes the packet radio system ideal for supporting communications during the response to natural disasters [3,4].

Each node in AFLC's packet radio network consists of an Advanced Electronics Applications, Inc. PK-232 Multi-Mode Data Controller, which acts as a terminal node controller (TNC). In addition, each TNC connects to an AN/URC-119 (HF) broadcast radio that prepares the data for transmission through the atmosphere. A software interface, written at AFIT, interfaces with the TNC. The interface allows an operator at a node location to send text messages to other node sites. The TNC controls the HF radio and accepts commands from the operator via the software interface [4].

The current network software insists that. the operator at a node location in AFLC's packet, radio network must specify other nodes that comprise a circuit. between the source and destination before sending a message. The source TNC then connects to the destination via the chosen nodes, and the nodes along the chosen path relay the message to the destination. In order to make wise decisions on the selection of routing nodes, the operator must manually, by phone calls to node sites or trial-and-error transmissions, determine t he availability of nodes. System operators can keep a log of previous successful routes and store these in existing routing tables; however, the availability of nodes fluctuate as conditions for radio propagation change. Thus, routing tables may be inaccurate. Due to the slow response time of the human operator in choosing relay nodes, this method of manual routing selection degrades network performance [3].

An AFIT research objective is to develop and implement a method that will free the system operator Gom selecting nodes that relay messages in-route to the destination and allow the system adaptive to changes in node availability. The proposed method will be in the form of a software algorithm that will automatically monitor the availability of nodes in the network and determine the best route that a particular message should follow to the final destination. Thus, the operator will only need to specify the destination node to successfully send data to another node.

The general class of routing chosen for the automatic routing algorithm has a distributed form similar to the algorithms used by the DARPA and VHF amateur radio packet. networks. Centralized routing was not chosen because of the amount of overhead generated, and even though isolated routing does not generate very much overhead, it. was not acceptable in this case because of the dependence of data transmissions to carry rou ting information.

5.1 Routing Table Structure

Each node will initially establish, and continually update, a tree-like routing table. The general form of the routing table is shown in Figure 2. The node at level 0 (resident node) identifies t.he node that contains the routing table, and one or more branches may form below this level. The level 1 entries are immediate neighboring nodes, which are one hop or radio link away from the resident! node. The resident. node will have as many level 1 neighbors as there are nodes having direct connectivity with the resident, node. Neighbors that. are two hops away from the resident. node are identified as level 2 neighbors, which are shown directly connected to respective level 1 neighbors. Levels 3 and higher may also form depending on the size and connectivity of the network. For instance, a resident, node will have a routing table entry for level 10 if it. has connectivity with another node 10 hops away.

An example of the routing table organization is shown in Figure 3. This figure illustrates a simple connection of nodes A, B, C, D, and E and a depiction of the routing table for nodes A and C. Notice that. the structures of the routing tables match the network's topology. In the routing table at. node A, nodes B and C are shown as level one neighbors. The table shows that. node A has direct. connectivity with nodes B and C. The routing table at. each node also represents the connections between node B and nodes C, D, and E, which are two hops away from node A. The same is true for the connections between node C and nodes B and D, which are also two hops away from node A. In addition, the table correctly denotes the links among nodes B and D with node E. Hence, the routing table gives a clear picture of the organization of connections among network nodes.

5.2 Route Determination

Using the routing table structure above, a node can choose the circuit that provides the connection that. offers the least number of hops between the source and destination nodes. For example, if node A in Figure 3 wishes to send a packet to node E, then node A can send the packet three possible ways. It could use circuit A-C-B-E, AC-D-E, or A-B-E. It is obvious that circuit A-B-E would be the best selection because this circuit will require only two hops versus three hops for the other choices.

A general rule for deciding which circuit to choose is as follows. A node needing to send a packet. to a certain destination begins by scanning each level of the routing table starting with level 1 and proceeding to higher-numbered levels until the destination node is found. The resulting level number will indicate the shortest number of hops to the destination. To determine the route, the algorithm starts at the destination node found in the table and follows the path back to the resident. node. The nodes that the algorithm must traverse to get back to the resident node comprise the desired circuit. If the algorithm finds more than one node on the same level that identify as the destination, then the algorithm will choose one at random and then determine the circuit,, as explained above, by following the path back to the resident node.

If for some reason the network can not support a connection by way of the chosen circuit., then the algorithm will choose another circuit. based on the same routing table level as before if another destination node entry resides at that level or drop down through the truth table to higher-numbered levels until another destination node entry is found. In addition, the algorithm will initiate a probing that will determine which part of the circuit. is causing the original circuit, to be faulty. The algorithm will accomplish the probing similar to the method technicians use to manually troubleshoot a faulty circuit. First, the resident node will fabricate and send a probe packet that is addressed to an intermediate node at or near the center of the faulty circuit. If this node is available and still connected to the circuit., it will send an immediate response to the sender. If the resident node receives a response from the intermediate node, the resident node can assume the circuit. is operational up to the that point in the circuit, and if the resident node does not receive a response, the resident node can assume the problem lies somewhere between the resident and the intermediate node. The resident node will continue probing in the appropriate direction until a faulty node pair is found.

As an example of the probing action, refer to Figure 3. Node A will assume has found circuit. A-B-D is faulty if no acknowledgment heard from node D. Node A will then send a test probe to the midway point, which is node B. If an acknowledgment, is not heard within a specified time period, then the problem must lie between nodes A and B. With this example, two possible causes of the discontinuity are that either node B is inoperable or nodes A and B have lost direct. connectivity with each other. Because of the sporadic propagation of HF radio signals and the reliable nature of nodal hardware, the most probable cause is that the two nodes have simply lost connectivity due to a change in radio propagation. Thus, the algorithm at the resident node should delete from the routing table any A-B or B-A node pair and the connections that fall directly below the faulty node pairs. This will avoid the selection of other circuits that contain the faulty node pair. Once the proper deletions have been made, the algorithm will broadcast a status packet that identifies the changes made to the table.

5.3 Updating Routing Tables

Nodes establish and periodically update routing tables through the use of status packets. Status packets contain the sender's identification and a copy of the sender's routing table. Initially, a node broadcasts a status packet which its neighbors receive. Nodes do not retransmit status packets. When a node receives a status packet, the node determines which node sent the packet? and then does one of two things. If the origin of the status packet, is not currently a level 1 table entry, the (resident) node will append the origin's table to the existing table, with the origin as a level 1 entry. If the sending node is already a level 1 table entry, then the resident node will replace the existing table's entries (associated with the sender) with the sender's table included in the status packet. If the routing table update causes the table to change, then the node will broadcast a status packet..

An important. function of the routing algorithm, with regard to table updates, is to determine when a level 1 neighbor has become disconnected from the resident node. If the neighbors of an inactive node are not warned, then they may attempt. to send data packets through the "dead" node. The solution to this problem is to have each node periodically send status packets if no information packets have been sent over a certain period of time. Thus, the resident node can monitor nodes currently listed as level 1 neighbors, and if the resident node does not receive any activity from a certain neighbor in the specified period of time, then it will delete from its routing table that level 1 entry and all connecting branches and nodes listed below it. In addition, the resident. node will broadcast, a stat us packet, to reflect the change of connectivity. This will reduce the chance of a node choosing a circuit. that includes the "dead" node.

The amount. of time to allow between periodic status transmissions is a function of the connectivity variance among nodes. Nodes should send status packets often enough to account, for changes of connectivity due to variations in HF radio propagation. HF propagation is sporadic, but it. tends to change over a period of hours, not. minutes or seconds. Thus, the transmission of status packets for the determination of connectivity changes should occur at least once every hour if no information packets are being sent.. Notice that this mechanism alone will not. allow quick modification of the tables due to nodes that. become inoperable.

With this method of updating the routing tables, a resident node is capable of gathering enough information to make routing decisions, but the tables will have a great deal of redundancy, mainly in the form of looping. Consider again the example illustrated in

Figure 3. When node C broadcasts its status packet, it effectively sends its routing table to nodes A, B, and D. In effect, node A will replace its level 1 entry corresponding to node C with the table contained in the status packet from node C. As a result, the table at node A will appear as in Figure 4(a). The redundant information here includes the new parts of the table that include node A, the resident uode. For instance, node A will never need the circuit A-C-A-B-E, A-C-A-B-D or A-C-B-A because they will loop data packets through the sending node (node A) one time before reaching the destination. This looping introduces useless overhead in the network.

Routinely, a node inherently sends redundant. looping information when sending a status packet!. The reason for this is that the status packet contains a copy of the routing table of the sender which normally includes the receiving node's level 1 entries. This causes the looping redundancy in Figure 4(a). In fact, if nothing is done, continual status transmissions will not only cause looping in the tables, but it will make the routing tables grow in length without bound.

5.4 Pruning

A method to reduce the looping and endless routing table growth is to "prune" the table after using the status packet to update table entries. The objective in pruning is to delete any parts of the routing table that represent, circuits with a loop through particular nodes (not. only the resident, node). By eliminating all circuits that cause this looping, the resulting pruned table will not grow indefinitely, and it will provide just. enough redundancy to accommodate valid alternate routes.

The rules of pruning are straightforward. After receiving a status packet and incorporating the neighbor's routing table as part of the existing table, the resident node searches for repeating node entries in every circuit. If the resident, node finds a repeating node, it deletes that node plus any branches that fall below and connected to the repeating node. For example, the resident node (node A) of Figure 4(a) will not find any repeating nodes in circuits A-B-E, A-B-C, A-B-D, A-C-B-D, A-C-B-E, A-C-D-B, or A-C-D-E. However, it will find repeating nodes in circuits A-C-A-B-E, A-C-A-B-D, and A-C-B-A. After deleting the repeating nodes and all connecting nodes below, the routing table at node A will appear as shown in Figure 4(b). This routing table illustrates the desired result: no looping redundancy.

After pruning, the resident node will immediately broadcast, a status packet if the new routing table structure is different! than before receiving the last. status packet.. This ensures other nodes get the updates as soon as possible.

5.5 Adaptability

In order to discuss the adaptability of this algorithm, it is important to first outline what causes connectivity to change and then treat each cause separately. Network connectivity may vary if a node is added to the network, if a node is deleted from the network, or if the radio propagation between two nodes changes.

If a node is added to the network, a technician will initialize the node, and the algorithm will clear its routing table, enter the proper resident node identification in the table, and immediately broadcast a status packet. This status packet, will only indicate the node sending the status because the rest. of the table will be empty. After reception of the status packet, each of its neighbors will immediately broadcast status packets as well so the new node can learn the connectivity among other nodes and other higher level neighbors can learn about the new node. Eventually, the new node will know the connectivity of all other nodes in the network, and the other nodes will know about the new node. Therefore, the algorithm will support the addition of a node.

If a node becomes inoperable, its level 1 neighbors will discover this because the neighbors will have been listening for transmissions from the "dead" node. After a period of time, if a node has not received any information packets or periodic status transmissions from a specific level 1 neighbor, it will delete that neighbor from its routing table and send status packets to reflect! the changes so other nodes can adjust. their routing tables. Because a node may not transmit, but still be operable, each node must periodically broadcast a beacon so all neighboring nodes will sense the presence of a possibly idle node. Thus, the algorithm will accommodate the removal of a node.

As radio propagation changes between a pair of nodes, the connectivity between that pair of nodes may be lost. This disconnection may possibly affect, every node's routing table. If the loss of connectivity is between, say, nodes A and B, then node A will not hear anything from node B, and node B will not hear anything from node A. Therefore, after a period of time, node A will delete from its table node B, and node B will delete from its table node A. Of course, this will initiate the immediate transmission of status packets, which will reflect the changes. Thus, the algorithm will successfully update routing tables as connectivity changes.

6 CONCLUSION

Designing a routing algorithm for a packet radio network is challenging, mainly because of changing connectivity due to node mobilitv and variances in radio propagation. Hopefully, this paper will have given networkers a basis of knowledge concerning routing techniques as they relate to packet radio networks.

AFIT's routing algorithm is currently being written in the Turbo C programming language and will be incorporated into the existing user interface. To handle the transmission of status packets, the software will operate the TNC in a connected mode to use the TNC's error recovery mechanisms. AFIT will begin testing the revised user interface with AFLC's packet. radio network in October 1990.

Bibliography

1. Geier, James T. "Automatic/Adaptive Routing Algorithm for the Air Force Logistics Command Packet Radio Network." MS thesis draft, AFIT/GE/ENG/90S. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, July 1990.

2. Jubin, J. and J.D. Tornow. "The DARPA Packet Radio Network Protocols," Proceedings of the IEEE, 75: 21-32 (January 1987).

3. Katsampes, TSgt, NCOIC Operations and Plans. Telephone interview. Air Force Logistics Command, SCSXP, Wright-Patterson AFB, OH, 13 March, 1990.

4. Marsh, Steven L. "Integration of the Air Force Logistics Command Packet Radio Network ." MS Thesis, AFIT/GE/ENG/89D-29. School of Engineering, Air Force Institute of Technology (AU), Wright- Patterson AFB OH, December 1989.

5. *NET/ROM for the TNC-2. Users'* Manual. Software 2000, Inc. Arroyo Grande, CA, 1987.

6. Schwartz, M. *Telecommunication Networks.* Addison-Wesley, 1987.

7. Tanenbaum, A.S. *Computer Networks* (Second Edition). Prentice-Hall, 1988.

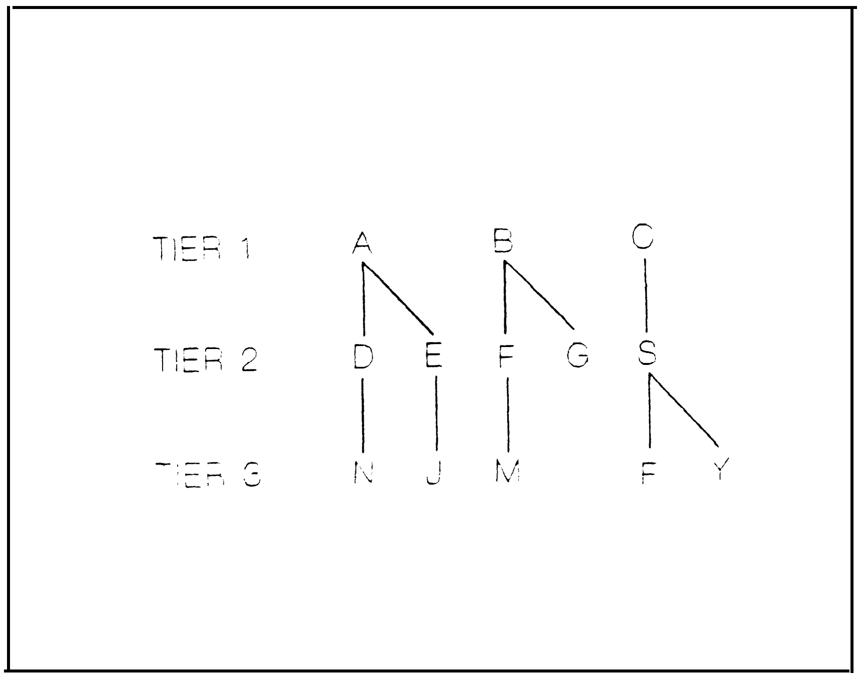

Figure 1. An Example of a Tier Table.

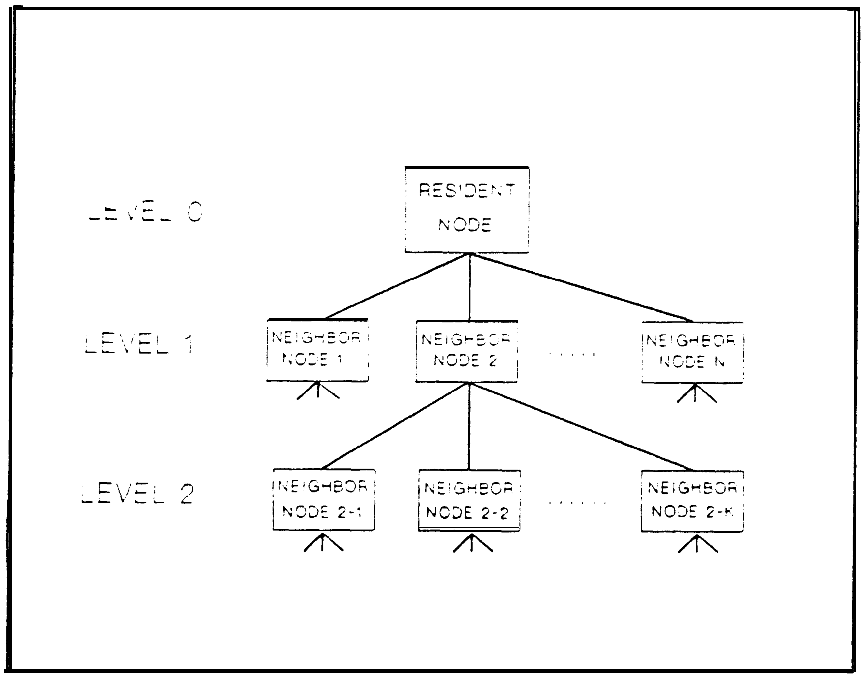

Figure 2. Generalized Routing Table Structure.

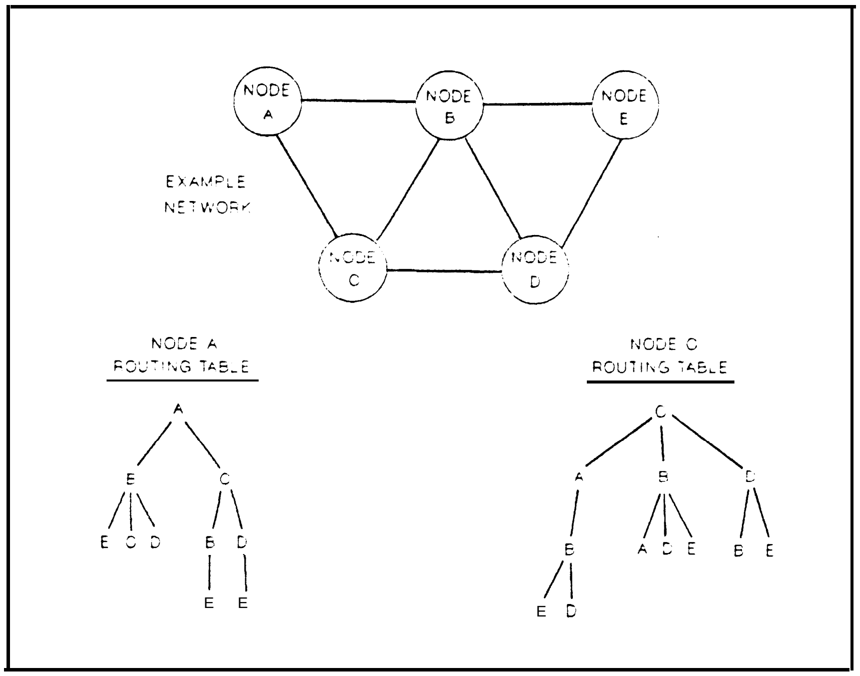

Figure 3. Example of Routing Table Entries.

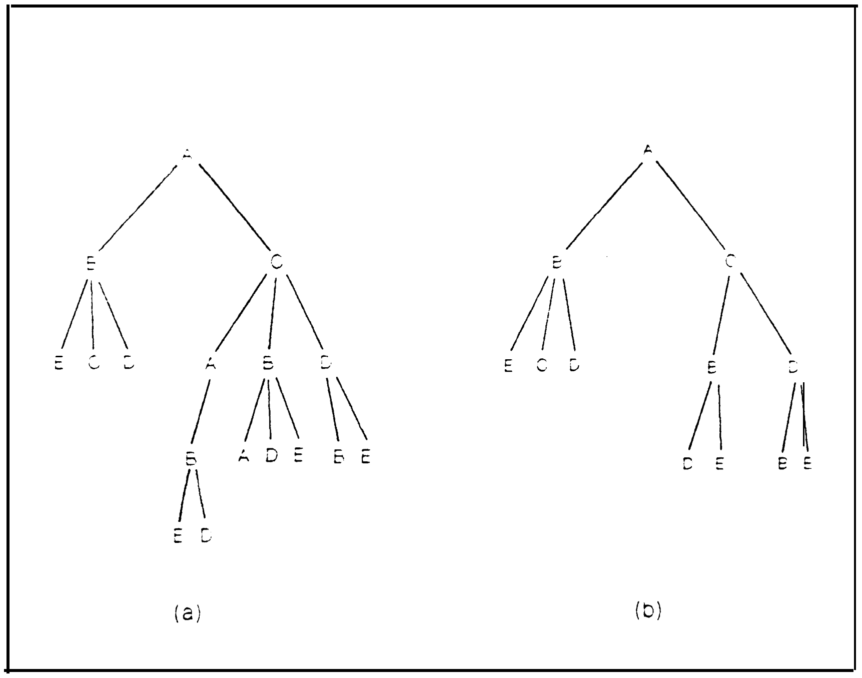

Figure 4. Routing Tables (a) before pruning and (b) after pruning.