ISSN (online): 2250-141X
Vol. 2 Issue 1, February 2012

PROTOCOL ROUTING IN AD-HOC SENSOR NETWORKS

V. Thrimurthulu[1], L. Rangiah[2], L Mihira priya[3]

1. Assoc. prof, Dept of ECE, chilkur balaji institute of Technology, Hyderabad
2. Professor, Dept of ECE, TRR College of Engineering , Hyderabad
3. Asst. prof.,Dept of electronics and Physics, St. joseph degree and PG college, hyd

Abstract

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions. This paper presents a protocol for routing in ad hoc networks that uses dynamic source routing. The protocol adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently. Based on results from a packet-level simulation of mobile hosts operating in an ad hoc network, the protocol performs well over a variety of environmental conditions such as host density and movement rates. For all but the highest rates of host movement simulated, the overhead of the protocol is quite low. In all cases, the difference in length between the routes used and the optimal route lengths is negligible, and in most cases, route lengths are on average within a factor of 1.01 of optimal.

1. INTRODUCTION

Mobile hosts and wireless networking hardware are becoming widely available, and extensive work has been done recently in integrating these elements into traditional networks such as the Internet. However, mobile users will want to communicate in situations in which no fixed wired infrastructure such as this is available, either because it may not be economically practical or physically possible to provide the necessary infrastructure or because the expediency of the situation does not permit its installation.

For example, in the network illustrated in Figure 1, mobile host *C* is not within the range of host *A*'s wireless transmitter (indicated by the circle around *A*) and host *A* is not within the range of host *C*'s wireless transmitter. If *A* and *C* wish to exchange packets, they may in this case enlist the services of host *B* to forward packets for them, since *B* is within the overlap between *A*'s range and *C*'s range. Indeed, the routing problem in a real ad hoc network may be more complicated than this example suggests, due to the inherent non-uniform propagation characteristics of wireless transmissions and due to the possibility that any or all of the hosts involved may move at any time.

This paper describes the design and performance of a routing protocol for ad hoc networks that instead uses dynamic source routing of packets between hosts that want to communicate. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which to forward the packet; the sender explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination host. Source routing has been used

© Copyright – IJST 2011

in a number of contexts for routing in wired networks, using either statically defined or dynamically constructed source routes, and has been used with statically configured routes in the Tucson Amateur Packet Radio (TAPR) work for routing in a wireless network. The protocol presented here is explicitly designed for use in the wireless environment of an ad hoc network. There are no periodic router advertisements in the protocol. Instead, when a host needs a route to another host, it dynamically determines one based on cached information and on the results of a route discovery protocol.

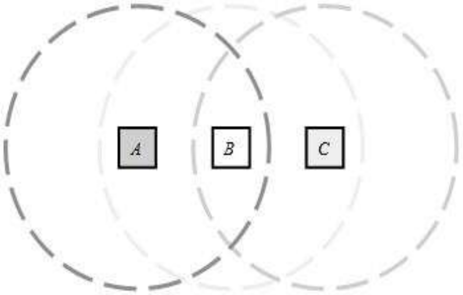


Figure 1 *A simple ad hoc network of three wireless mobile hosts*

2. SOURCE ROUTING

2.1. Overview

To send a packet to another host, the sender constructs a source route in the packet's header, giving the address of each host in the network through which the packet should be forwarded in order to reach the destination host. The sender then transmits the packet over its wireless network interface to the first hop identified in the source route. When a host receives a packet, if this host is not the final destination of the packet, it simply transmits the packet to the next hop identified in the source route in the packet's header. Once the packet reaches its final destination, the packet is delivered to the network layer software on that host.

Each mobile host participating in the ad hoc network maintains a route cache in which it caches source routes that it has learned. When one host sends a packet to another host, the sender first checks its route cache for a source route to the destination. If a route is found, the sender uses this route to transmit the packet. If no route is found, the sender may attempt to discover one using the route discovery protocol. While waiting for the route discovery to complete, the host may continue normal processing and may send and receive packets with other hosts. The host may buffer the original packet in order to transmit it once the route is learned from route discovery, or it may discard the packet, relying on higher-layer protocol software to retransmit the packet if needed. Each entry in the route cache has associated with it an expiration period, after which the entry is deleted from the cache.

2.2. Route Discovery

Route discovery allows any host in the ad hoc network to dynamically discover a route to any other host in the ad hoc network, whether directly reachable within wireless transmission range or reachable through one or more intermediate network hops through other hosts. A host initiating a route discovery broadcasts a route request packet which may be received by those hosts within wireless transmission range of it. The route request packet identifies the host, referred to as the target of the route discovery, for which the route is requested. If the route discovery is successful the initiating host receives a route reply packet listing a sequence of network hops through which it may reach the target.

When any host receives a route request packet, it processes the request according to the following steps:

1. If the pair initiator address, request id

© Copyright – IJST 2011

for this route request is found in this host’s list of recently seen requests, then discard the route request packet and do not process it further.

2. Otherwise, if this host’s address is already listed in the route record in the request, then discard the route request packet and do not process it further.

3. Otherwise, if the target of the request matches this host’s own address, then the route record in the packet contains the route by which the request reached this host from the initiator of the route request. Return a copy of this route in a route reply packet to the initiator.

4. Otherwise, append this host’s own address to the route record in the route request packet, and re-broadcast the request.

In order to return the route reply packet to the initiator of the route discovery, the target host must have a route to the initiator. If the target has an entry for this destination in its route cache, then it may send the route reply packet using this route in the same way as is used in sending any other packet. Otherwise, the target may reverse the route in the route record from the route request packet, and use this route to send the route reply packet. This, however, requires the wireless network communication between each of these pairs of hosts to work equally well in both directions, which may not be true in some environments or with some MAC-level protocols.

2.3. Route Maintenance

Conventional routing protocols integrate route discovery with route maintenance by continuously sending periodic routing updates. If the status of a link or router changes, the periodic updates will eventually reflect the changes to all other routers, presumably resulting in the computation of new routes. However, using route discovery, there are no periodic messages of any kind from any of the mobile hosts.

If the wireless network does not support such lower-level acknowledgements, an equivalent acknowl- edgement signal may be available in many environments. After sending a packet to the next hop mobile host, the sender may be able to hear that host transmitting the packet again, on its way further along the path, if it can operate its wireless network interface in promiscuous mode. For example, in Figure 1, host *A* may be able to hear *B*’s transmission of the packet on to *C*. This type of acknowledgement is known as a 0. In addition, existing transport or application level replies or acknowledgements from the original destination could also be used as an acknowledgement that the route (or that hop of the route) is still working. As a last resort, a bit in the packet header could be included to allow a host transmitting a packet to request an explicit acknowledgement from the next-hop receiver. If no other acknowledgement signal has been received in some time from the next hop on some route, the host could use this bit to inexpensively probe the status of this hop on the route.

Route maintenance can also be performed using end-to-end acknowledgements rather than the hop-by- hop acknowledgements described above, if the particular wireless network interfaces or the environment in which they are used are such that wireless transmissions between two hosts do not work equally well in both directions. As long as some route exists by which the two end hosts can communicate (perhaps different routes in each direction), route maintenance is possible. In this case, existing transport or application level replies or acknowledgements from the original destination, or explicitly requested network level acknowledgements, may be used to

© Copyright – IJST 2011

ISSN (online): 2250-141X
Vol. 2 Issue 1, February 2012

indicate the status of this host's route to the other host. With hop-by-hop acknowledgements, the particular hop in error is indicated in the route error packet, but with end-to-end acknowledgements, the sender may only assume that the last hop of the route to this destination is in error.

3. Optimizations

A number of optimizations are possible to the basic operation of route discovery and route maintenance as described in Section 3.2, that can reduce the number of overhead packets and can improve the average efficiency of the routes used on data packets. This section discusses some of those optimizations.

3.1. Full Use of the Route Cache

The data in a host's route cache may be stored in any format, but the active routes in its cache in effect form a tree of routes, rooted at this host, to other hosts in the ad hoc network. For example, Figure 2 shows an ad hoc network of five mobile hosts, in which mobile host *A* has earlier completed a route discovery for mobile host *D* and has cached the route to *D* through *B* and *C*. Since hosts *B* and *C* are on the route to *D*, host *A* also learns the route to both of these hosts from its route discovery for *D*. If *A* later performs a route discovery and learns the route to *E* through *B* and *C*, it can represent this in its route cache with the addition of the single new hop from *C* to *E*. If *A* then learns it can reach *C* in a single hop (without needing to go through *B*), *A* can use this new route to *C* to also shorten the routes to *D* and *E* in its route cache.

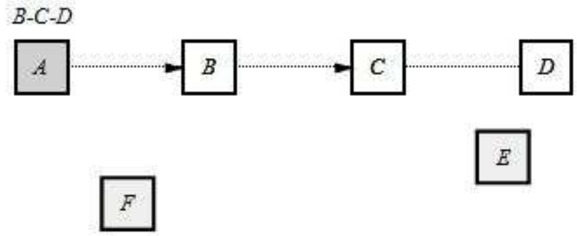
A host can add entries to its route cache any time it learns a new route. In particular, when a host forwards a data packet as an intermediate hop on the route in that packet, the forwarding host is able to observe the entire route in the packet. Thus, for example, when host *B* forwards packets from *A* to *D*, *B* can add the route information from that packet to its own route cache. If a host forwards a route reply packet, it can also add the route information from the route record being returned in that route reply, to its own route cache. Finally, since all wireless network transmissions are inherently broadcast, a host may be able configure its network interface into promiscuous receive mode, and can then add to its route cache the route information from any data or route reply packet it can overhear.

A host may use its route cache to avoid propagating a route request packet received from another host. In particular, suppose a host receives a route request packet for which it is not the target and is not already listed in the route record in the packet, and for which the pair initiator address, request id is not found in its list of recently seen requests; if the host has a route cache entry for the target of the request, it may append this cached route to the accumulated route record in the packet, and may return this route in a route reply packet to the initiator without propagating (re-broadcasting) the route request. Thus, for example, if mobile host *F* needs to send a packet to mobile host *D*, it will initiate a route discovery and broadcast a route request packet. If this broadcast is received by *A*, *A* can simply return a route reply packet to *F* containing the complete route to *D* consisting of the sequence of hops *A*, *B*, *C*, and *D*.

A particular problem can occur, however, when several mobile hosts receive the initiator's broadcast of the route request packet, and all reply based on routes found in their route caches. In Figure 2, for example, if both *A* and *B* receive *F*'s route request broadcast, they will both be able to reply from their route caches, and will both send their replies at about the same time since they both received the broadcast at about the same time. Particularly when more than the two mobile hosts in this example are involved, these simultaneous replies from the mobile

© Copyright – IJST 2011

hosts receiving the broadcast may create packet collisions among some or all of these replies and may cause local congestion in the wireless network. In addition, it will often be the case that the different replies will indicate routes of different lengths. For example, *A*'s reply will indicate a route to *D* that is one hop longer than that in *B*'s reply.



We avoid the problems of many simultaneous replies and attempt to eliminate replies indicating routes longer than the shortest reply, by causing each mobile host to delay slightly before replying from its cache. Before replying from its route cache, a host performs the following actions:

1. Pick a delay period d= H X (h-1+ r), where h is the length in number of network hops for the route to be returned in this host's reply, r is a random number between 0 and 1, and H is a small constant delay to be introduced per hop.

2. Delay transmitting the route reply from this host for a period of d .

3. Within this delay period, promiscuously receive all packets at this host. If a packet is received by this host during the delay period addressed to the target of this route discovery (the target is the final destination address for the packet, through any sequence of intermediate hops), and if the length of the route on this packet is less than , then cancel the delay and do not transmit the route reply from this host; this host may infer that the initiator of this route discovery has already received a route reply, giving an equal or better route.

As a last optimization involving full use of the route cache, we have added the ability for the initiator of a route request to specify in the request packet, the maximum number of hops over which the packet may be propagated. If another host near the initiator has a cache entry for the target of a route request, the propagation of many redundant copies of the route request can be avoided if the initiator can explicitly limit the request's propagation when it is originally sent. Currently, we use this ability during route discovery as follows:

1. To perform a route discovery, initially send the route request with a hop limit of one. We refer to this as a non-propagating route request.

2. If no route reply is received from this route request after a small timeout period, send a new route request with the hop limit set to a predefined "maximum" value for which it is assumed that all useful routes in the ad hoc network are less than this limit.

This procedure uses the hop limit on the route request packet to inexpensively check if the target is currently within wireless transmitter range of the initiator or if another host within range has a route cache entry for this target. Since the initial request is limited to one network hop, the timeout period before sending the propagating request can be quite small. This mechanism could also be used to implement an "expanding ring" search for the target, in which the hop limit is gradually increased in subsequent retransmissions of the route request for this target.

3.2. Piggybacking on Route Discoveries

When one host needs to send a packet to

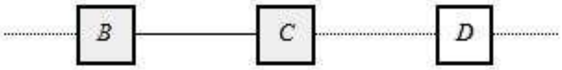
© Copyright – IJST 2011

another, if the sender does not have a route cached to the destination host, it must initiate a separate route discovery, either buffering the original packet until the route reply is returned, or discarding it and relying on a higher-layer protocol to retransmit it if needed. The delay for route discovery and the total number of packets transmitted can be reduced by allowing data to be piggybacked on route request packets. Since some route requests may be propagated widely within the ad hoc network, though, the amount of data piggybacked must be limited. If the route request is received by some host and is replied to based on the host's route cache without propagating the request , the piggybacked data would be lost when the host discards the route request. In this case, before discarding the packet, the host must construct a new packet containing the piggybacked data from the route request packet, setting the route in this packet from the route being returned in the route reply. The route should appear as if the new packet had been sent by the initiator of the route discovery and had been forwarded normally to this host: the first portion of the route is taken from the accumulated route record in the route request packet, and the remainder of the route is taken from this host's route cache. The sender address in the packet should also be set to the initiator of the route discovery.

3.3. Reflecting Shorter Routes

While two hosts are communicating with each other using cached routes, it is desirable for the hosts to begin using shorter routes if the hosts move sufficiently closer together. In many cases, the basic route maintenance procedure is sufficient to accomplish this, since if one of the hosts moves enough to allow the route to be shortened, it will likely also move out of transmission range of the first hop on the existing route.

An improvement to this method of reflecting shorter routes is possible if hosts operate their network interfaces in promiscuous receive mode. Suppose somewhere in the forwarding of a packet, mobile host *B* transmits a packet to *C*, with *D* being the next hop after *C* in the route in the packet, as illustrated in Figure 3. If *D* receives this packet, it can examine the packet header to see that the packet reached it from *B* in one hop rather than two as intended by the route in the packet. In this case, *D* may infer that route may be shortened to exclude the intermediate hop through *C*. *D* then sends an unsolicited route reply packet to the original sender of the packet, informing it that it can now reach *D* in one hop from *B*. As with other route reply packets, other hosts which also receive this route reply may also incorporate this change into their route caches.



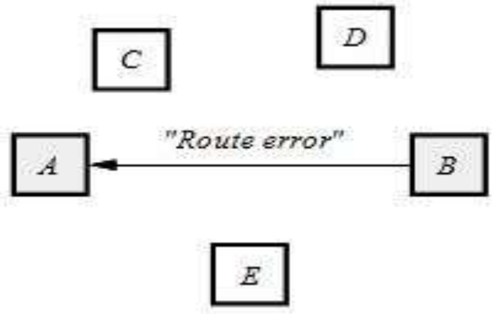
3.4. Improved Handling of Errors

One common error condition that must be handled in an ad hoc network is the case in which the network effectively becomes partitioned. That is, two hosts that wish to communicate are not within transmission range of each other, and there are not enough other mobile hosts between them to form a sequence of hops through which they can forward packets. If a new route discovery were to be initiated for each packet sent by a host in this situation, a large number of unproductive route request packets would be propagated throughout the subset of the ad hoc network reachable from this host. In order to reduce the overhead from such route discoveries, we use exponential back off to limit the rate at which new route discoveries may be initiated from any host for the same target. If the host attempts to send additional data packets to this same host more frequently than this limit, the

© Copyright – IJST 2011

subsequent packets may be buffered until a route reply is received, but they do not initiate a new route discovery until the minimum allowable interval between new route discoveries for this target has been reached. This limitation on the maximum rate of route discoveries for the same target is similar to the mechanism required by Internet hosts to limit the rate at which ARP requests are sent to any single IP address [3].

An additional optimization possible to improve the handling of errors is to use promiscuous receiving mode to allow hosts to eavesdrop on route error packets being sent to other hosts. For example, Figure 4 shows the return of a route error packet to mobile host *A* from host *B*. If hosts *C*, *D*, and *E* are operating in promiscuous receive mode, they will be able to receive the route error packet. Since a route error packet names both ends of the route hop causing the error, any host receiving the error packet can update its route cache to reflect the fact that the two hosts indicated in the packet can no longer directly communicate. A host receiving a route error packet can simply search its route cache for any routes using this hop, and for each such route found, the route is truncated at this hop. All hosts on the route before this hop are still reachable on this route, but subsequent hosts are not.



A last optimization to improve the handling of errors is to support the caching of “negative” information in a host’s route cache. Suppose, in Figure 4, that none of these optimizations for handling errors are in use. When *A* receives *B*’s route error packet, it may initiate a route discovery in order to find a new route to the same target. However, if hosts *C*, *D*, or *E* have an entry in their route cache for this target, they will likely reply to *A* from their cache with a cached copy of the same route that *A* just removed from its cache. If instead, *A* could enter into its cache when it receives *B*’s route error packet, an indication that this hop is not currently working (rather than simply removing this hop from any routes currently in its cache), then *A* could ignore future replies from *C*, *D*, and *E* that include this hop from their caches. A short expiration period must be placed on this negative cached information, since while this entry is in its cache, *A* will otherwise refuse to allow this hop in any route entries in its cache, even if this hop begins working again.

We have not currently included this caching of negative information in our simulation, due to the difficulty of picking a suitable expiration period, and since it appears to not be necessary in most cases, if hosts also promiscuously receive route error packets. For example, in Figure 4, if *C*, *D*, and *E* also receive *B*’s route error packet, they will have removed this hop from their caches before *A*’s new route discovery is initiated, thus avoiding the problem.

4. Results

We executed 20 runs of the simulator for each of a number of different movement rates and numbers of mobile hosts in the simulated ad hoc network. Each run simulated 4000 seconds of execution, with each mobile host moving and communicating. The movement rate of the mobile hosts was determined by the pause time described above, with pause times ranging from 0 to 4000. With a pause time of 0, all hosts are constantly in motion, whereas with a pause time of 4000,

© Copyright – IJST 2011

hosts do not move during the run from their initial randomly chosen locations. The ad hoc network in each run consisted of 6, 12, 18, or 24 mobile hosts. We present here the average over the 20 runs of the simulator for each of these cases; standard deviation for all cases was within 7% and in general was 2% or less of the average value for each case.

Figure 5 shows the total number of packet transmissions performed relative to the optimal number of transmissions for the data packets sent during the simulation. The optimal number of transmissions is the number of hops for each data packet needed to get from the sender of a packet to the intended receiver, if perfect routing decisions are made for each packet and if no transmission errors occur. The total number of packets actually transmitted includes the number of hops for each data packet based on the source route used by the sender, plus all packet transmissions used for route request, route reply, and route error packets. This ratio shows the work efficiency of the protocol: a value of 1.0 indicates a perfectly efficient protocol with no overhead packets present.

For all but the shortest pause times in the simulated environment, the total number of packets trans- mitted by the protocol is very close to optimal, and falls to an overhead of about 1% for pause times greater than 1000 with 24 mobile hosts, as shown in Figure 5. For very short pause times, representing very frequent host movement, the protocol overhead is higher, reaching a maximum ratio of 2.6 for a pause time of 0, representing all hosts in constant motion. In such situations, source routes become invalid quickly after they are discovered, and additional overhead is spent dis- covering new routes. However, because the route maintenance procedure can quickly detect when a route in use is no longer working, nearly all data packets can be successfully delivered even in periods of such extreme host movement.

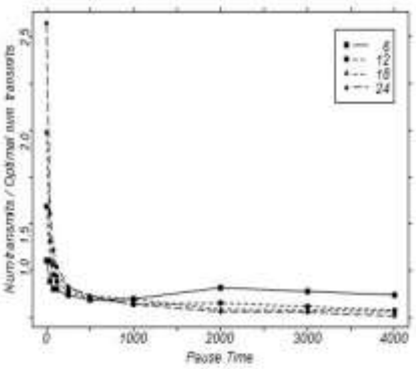


Figure 5 *Average total number of transmissions performed relative to optimal (20 runs)*

The performance results for protocol overhead presented here are affected by the occurrence of discon- nected components of the mobile hosts within the area of the ad hoc network. When placing a number of mobile hosts at random locations within the simulation area, there is a chance that some groups of hosts will be unable to communicate with other groups of hosts since all hosts in each group are out of wireless transmission range of all hosts in other groups. This effectively forms a partition of the ad hoc network. For example, Figure 6 illustrates a typical placement of 24 mobile hosts, which happened to form two disconnected components.

For each data packet sent with the receiver outside the sender's disconnected component, the basic protocol would initiate a route discovery, although we have included an optimization to limit the rate of new discoveries using an exponential backoff. The remaining extra route discoveries still performed in such situations show in increased protocol overhead, such as in the higher overhead values for 6 and 12 hosts shown in Figure 5, although the number of such extra route discoveries due disconnected components is greatly reduced by this optimization.

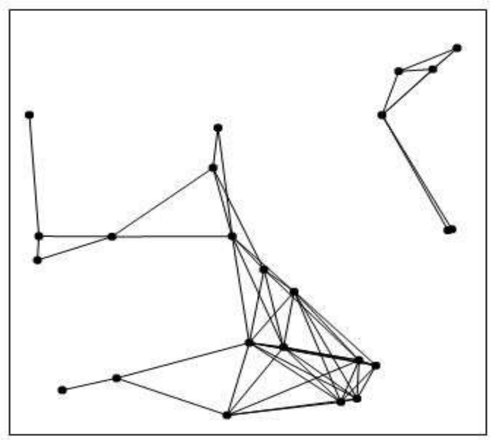


Figure 6 *Example of disconnected clusters with 24 hosts*

5. Conclusion

This paper has presented a protocol for routing packets between wireless mobile hosts in an ad hoc network. Our protocol uses dynamic source routing which adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently. Based on results from a packet-level simulation of mobile hosts operating in an ad hoc network, the protocol performs well over a variety of environmental conditions such as host density and movement rates. For all but the highest rates of host movement simulated, the overhead of the protocol is quite low, falling to just 1% of total data packets transmitted for moderate movement rates in a network of 24 mobile hosts. In all cases, the difference in length between the routes used and the optimal route lengths is negligible, and in most cases, route lengths are on average within a factor of 1.02 of optimal.

We are currently expanding our simulations to incorporate some additional optimizations and to quantify the effects of the individual optimizations on the behavior and performance of the protocol. We are also continuing to study other routing protocols for use in ad hoc networks, including those based on distance vector or link state routing, as well as the interconnection of an ad hoc network with a wide-area network such as the Internet, reachable by some but not all of the ad hoc network nodes.

References

[1] David F. Bantz and Frédéric J. Bauchot. Wireless LAN design alternatives. *IEEE Network*, 8(2):43–53, March/April 1994.

[2] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN’s. In *Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications*, pages 212–225, August 1994.

[3] Robert T. Braden, editor. Requirements for Internet hosts — communication layers. Internet Request For Comments RFC 1122, October 1989.

[4] Roy C. Dixon and Daniel A. Pitt. Addressing, bridging, and source routing. *IEEE Network*, 2(1):25–32, January 1988.

[5] Deborah Estrin, Daniel Zappala, Tony Li, Yakov Rekhter, and Kannan Varadhan. Source Demand Routing: Packet format and forwarding specification (version 1). Internet Draft, January 1995. Work in progress.

[6] Daniel M. Frank. Transmission of IP datagrams over NET/ROM networks. In *ARRL Amateur Radio 7th Computer Networking Conference*, pages 65–70, October 1988.

[7] Bdale Garbee. Thoughts on the issues of address resolution and routing in amateur packet radio TCP/IP networks. In *ARRL Amateur Radio 6th Computer Networking Conference*, pages 56–58, August 1987.

[8] James Geier, Martin DeSimio, and

Byron Welsh. Network routing techniques and their relevance to packet radio networks. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 105–117, September 1990.

[9] C. Hedrick. Routing Information Protocol. Internet Request For Comments RFC 1058, June 1988.

[10] International Standards Organization. Intermediate system to intermediate system intra-domain routing exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473). ISO DP 10589, February 1990.

[11] John Jubin and Janet D. Tornow. The DARPA packet radio network protocols. *Proceedings of the IEEE*, 75(1):21–32, January 1987.

[12] M. Frans Kaashoek, Robbert van Renesse, Hans van Staveren, and Andrew S. Tanenbaum. FLIP: An internetwork protocol for supporting distributed systems. *ACM Transactions on Computer Systems*, 11(1):73–106, February 1993.

[13] Phil Karn. MACA — A new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, September 1990.

[14] Philip R. Karn, Harold E. Price, and Robert J. Diersing. Packet radio in the amateur service. *IEEE Journal on Selected Areas in Communications*, SAC-3(3):431–439, May 1985.

[15] Gregory S. Lauer. Packet-radio routing. In *Routing in Communications Networks*, edited by Martha E. SteenStrup, chapter 11, pages 55–76. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

© Copyright – IJST 2011