AN ABSTRACT OF THE THESIS OF

Rudhrakumar Venkatesan for the degree of Master of Science in Electrical & Computer Engineering presented on January 24, 2003.

Title: Implementation and Performance Analysis of a Scalable Routing Protocol

Redacted for privacy

Abstract approved ______________________

Suresh P. Singh

Ad Hoc networks are multi-hop wireless networks consisting of mobile hosts. They do not have any pre-existing network infrastructure and are characterized by constantly changing topology, limited battery power and bandwidth. Typical applications of such networks are battlefield networks, medical relief during natural calamities or disasters, conference room networking, and intra-vehicular communications. Routing packets in an ad-hoc network is a challenge because of the mobile nature of the nodes and the constantly changing topology. In ad hoc networks, each mobile node functions as a router, forwarding packets, establishing routes and helping each other in maintaining the network. A novel scalable routing protocol SLURP (Scalable Location Updated-based Routing Protocol) addresses these issues of ad hoc networks routing. The protocol is based on a location management strategy, which keeps the routing overhead to a minimum. In this thesis we compare the protocol against an existing set of multi-hop ad hoc network routing protocols that cover a range of design choices: DSDV, TORA, DSR, and AODV. We implemented SLURP in the network simulator ns-2, with the necessary wireless extensions. Experiments were run to simulate changes in network topology, number of active sources, link connectivity, and speed of motion. The difference in performance and scalability are illustrated.

Implementation and Performance Analysis of a Scalable Routing Protocol

by

Rudhrakumar Venkatesan

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented January 24, 2003
Commencement June 2003

Master of Science thesis of Rudhrakumar Venkatesan presented on January 24, 2003.

APPROVED:

Redacted for privacy

Major Professor, representing Electrical & Computer Engineering

Redacted for privacy

Head of the Department of Electrical & Computer Engineering

Redacted for privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Rudhrakumar Venkatesan, Author

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to all the people who have been a part of my successful graduation. I would always be grateful to their exceptional support and guidance in times of distress.

Dr.Suresh Singh, my major professor, was the most influential person who helped me in every possible way to finish my thesis. I am deeply indebted to Dr.Singh for being my mentor and supporting me throughout my graduate life and in my thesis. He gave me the freedom and liberty to choose my courses and charter my way through the entire graduate program.

I would like to thank Dr.Bella Bose, Dr.Alexandre Tenca and Dr.Roy Haggerty for being a part of my thesis defense committee and for their valuable suggestions for completing the thesis write up. My Thanks to Mr. Michael Woo, of Motorola Inc, in allowing me to use some of his research work in my thesis.

I was fortunate to have my dearest friend Suresh Rangarajulu by my side. He was always there for me when I needed him the most. His patience and perseverance was one of the key factors that helped me in finishing my thesis. I would also like to thank Shashidhar Lakkavalli who helped me overcome many an obstacle with valuable inputs and suggestions to solve them.

I had the pleasure and company of my wonderful friends, who were always there to help me. Especially, Arun Ganapathy, Saumya Venkataraman, Aparna Sambamoorthy who have all spent some exciting hours trying to make my thesis report look better. I would also like to thank Shin pravin malli, for providing me with his laptop to help in the thesis report.

I would like to thank the OSU writing center and the exciting people working there who helped me with my thesis report. I would also express my thanks to Ferne Simendinger, Graduate Coordinator, Department of ECE, for helping me in setting up the thesis defense. I also want to thank the Department of computer science, Portland State University for providing me with some of the computing resources, which helped me a great deal in running my simulations.

My family has always been supportive throughout my career. I take this opportunity to thank them for providing me with all the love and care. My father Mr K.Venkatesan, my mother Mrs.Mani Venkatesan, and my brother Prashanth have been my source of energy throughout my career.

Thank you all. It has been a pleasure.

TABLE OF CONTENTS

LIST OF FIGURES

DEDICATED

To my Gurus, beloved Friends and Family.

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF A SCALABLE ROUTING PROTOCOL

1. INTRODUCTION

Wireless devices are becoming increasingly popular; they became smaller, less expensive and more powerful. As a result, there is an increasing need for direct communication between devices. The necessary networking support for these mobile devices is currently provided by installing base stations and access points. But, providing such access points everywhere is not feasible due to the high cost of installation, infrequent usage, unfriendly terrain or territory and lower return of investment. This widespread availability of mobile wireless devices together with the limitations of the existing system has triggered research in a new area in which nodes with minimal or no communication infrastructure, organize themselves to form an ad hoc network.

These ad hoc networks consist of autonomous mobile nodes which not only act as a host but also as routers forwarding packets, maintaining routes and helping each other in a symbiotic fashion. Such a network can be envisioned as a collection of routers equipped with wireless transmitters/receivers, which are free to move about arbitrarily. The status of communication links between the routers is a function of their positions, transmission power, antenna patterns, co channel interference etc. The mobility of the routers and the variability of other connectivity factors result in a network with a potentially rapid and unpredictably changing topology.

Examples of such networks include a battle site network where soldiers with mobile communication devices need to communicate with each other in enemy territory, disaster relief scenario where information is exchanged for coordinating efforts, conference room networks, and intra-vehicular communication networks.

Several networking problems need to be solved in order to deploy ad hoc networks, and in this thesis, we focus on the problem of routing in ad hoc networks.

1.1. Routing Problems in Ad Hoc Networks

1.1.1. Conventional Routing Protocols

The simplest and easiest way to provide a routing solution to an ad hoc network is to treat each mobile node as a router and run a conventional routing protocol like BGP or OSPF in each of them. This would treat the entire network as a set of routers. Each node will act as intermediate routers with respect to the source and destination. Conventional routing protocols fall into two categories, as mentioned in [1], into distance vector-based algorithms and link state-based algorithms.

In distance vector-based routing algorithms, each node maintains a routing table listing the routes to all possible destinations it has ever heard about. Each router broadcasts this message across the network periodically. Every other router, which receives this message, computes the best next-hop for each of the destinations based on its current set of neighbors. Whenever a packet is presented to the router for forwarding, it simply does a look-up of the routing table and forwards it to the next hop router for the destination. These algorithms depend on the periodic updates to maintain routes. If a link goes down or is added frequently then the rate at which these routing updates are sent should increase in order for the algorithm to converge quickly. This increases the utilization of network bandwidth and processor time. An example of distance vector protocols includes RIP, used in parts of the Internet.

In link state-based routing algorithms, each node maintains a picture of the current topology of the network. It also monitors the various Type-of-service factors such as cost of each link, message efficiency, and congestion on each of its neighboring routers. It then periodically broadcasts this message to all other routers in the network. With this information each router can compute the best path for a destination. When a packet is handed to a router, it forwards it to the next router based on the best path for a destination. Link state is more efficient when the network topology changes, as it converges faster, but the routing overhead is very high. In addition, it requires more processor time, as each router has to compute the best path to each of the destinations separately. Examples of link state protocols include OSPF, which is used in parts of the Internet.

1.1.2. Problems with using conventional protocols in Ad hoc networks

Wireless transmission between two hosts is asymmetrical. If a Node A receives a packet from Node B, it does not necessarily mean that Node A will be able to sent packets to Node B. Radio and infrared propagation does not observe true bi-directional symmetry [9]. Assumptions like these in the routing table computations make several routes to be stale. Movement, fading, jamming or interference is all typical characteristics of wireless medium.

The wireless networks are highly dynamic. The wired networks are configured properly with only a certain number of routers between each network. Due to the mobile nature of the ad hoc networks, there are many suitable routers, which are neighbors and are constantly changing. All these routers sending routing updates severely clogs the networks and has a toll on the life of the mobile node as the information in the packets has to be processed with each routing update.

The periodic routing updates of the conventional protocol wastes a lot of bandwidth. Even when nothing is changed in the network topology, there is a huge routing overhead due to the broadcast nature of these updates. All nodes within the transmission range of the sending node will consume each other's bandwidth. Ideally these updates in the Ad hoc networks should be mobility based and at steady state there should be no updates.

Sending routing packets periodically consumes battery power [39]. For any mobile node, battery power is the single most important resource and every effort should be made to conserve it. Each packet transmitted consumes a portion of the power stored within the battery. Although receiving a packet does not consume power like sending a packet, it does not allow the node to go into standby mode and save power, as it has to receive packets often.

The conventional routing protocols are neither tuned nor designed for very fast topology changes that might happen in an ad hoc network. Topology changes in a wired network are limited to a link going down or coming up, new routers that are added to a network, existing routers being removed from a network, cost changes and congestion. However, in an ad hoc network, mobility triggers topology change. Routes become stale very fast, old links go down soon, and new links come up pretty fast when compared to wired networks. If the convergence time of the routing protocols is greater than the average route validity time, then those protocols are not of much use, because by the time we find a route it is already obsolete. This speed of convergence can be increased with increasing the periodic updates, which again consumes network bandwidth, processing power and ultimately the battery life upon which the mobile nodes survive.

Though some of the above-discussed issues can be addressed with some improvements to the conventional routing protocols, it assures us that ad hoc networks require a new approach to routing.

1.1.3. Ideal ad hoc network routing protocol

The ideal characteristics of an ad hoc networking protocol should accomplish the following:

- All nodes should equally participate in the work of maintaining the routes to the various nodes
- No node should be a bottleneck; The failure of a node does not jeopardize the network
- Provides loop free routes
- Provides multiple routes to alleviate congestion and link failure
- Establishes routes quickly before they become stale

- Minimizes communication and routing overhead by localizing topological changes to conserve bandwidth, increase life of battery and increase scalability
- Storage and communication costs should grow only as a small fraction of the total number of nodes

1.2. Problem statement

Routing packets in ad hoc networks is a challenge because of the constantly changing topology of the network triggered by node mobility. As a result the control information that needs to be exchanged to maintain routing information increases in these fast changing networks. Side effects of stale routes are increased end-to-end delay and increased probability of packet loss. Many different protocols have been proposed to solve the multi-hop routing problem in ad hoc networks, each based on different assumptions.

In addition to correctness, a good routing protocol for ad hoc networks should have a low communication overhead, by minimizing route setup and maintenance messages. The protocol should also be simple and efficient, since it must quickly converge to select a route before network changes make the route invalid. The routing protocol must be distributed since no centralized host is available and to make the protocol scalable. It must be loop-free and should have minimal memory overhead. It should take advantage of the technology available and utilize information about the current state of the network.

Scalable Location Update-based Routing Protocol (SLURP) [42] is one such protocol. In this thesis we analyze the protocol SLURP and provide a realistic and quantitative analysis comparing the performance of a variety of Ad hoc networking protocols. For this purpose, SLURP was implemented in the network simulator ns-2 [25]. Results of the detailed simulation are presented showing the relative performance of SLURP against four recently proposed ad hoc networking protocols: DSDV [35], DSR [21], AODV [17], TORA [15].

1.3. Overview of thesis

The first chapter of the thesis Introduces ad hoc networks and addresses the problem of routing in ad hoc networks. Chapter 2 presents a literature review of existing ad hoc networking protocols and their classification. Chapter 3 provides an overview of SLURP. Chapter 4 discusses the simulator ns-2 and the simulation parameters and criteria. The results of the detailed simulations and the comparative analysis of the protocol with the existing protocols are explained in Chapter 5. Finally, Chapter 6 presents our conclusions and future research directions.

2. LITERATURE REVIEW

2.1. Classification

There are a variety of protocols which were proposed to solve the problem of routing in ad hoc networks. We can easily classify them into five major categories as follows:

2.1.1. Non-hierarchical Minimum cost Routing protocols

These are proactive routing protocols, which maintain routing tables by periodic exchange of routing information. This implies routing overhead with each exchange and routes have to be computed according to the new information. Since these tables are possibly large, it utilizes up huge bandwidth and battery. Examples include Destination Sequenced Distance Vector (DSDV [35]), Wireless Routing Protocol (WRP [11]) and Global State Routing (GSR [5]). In WRP, routing nodes communicate the distance and second-to-last hop for each destination. WRP produces the number of cases in which a temporary routing loop can occur, which accounts for its fast convergence properties. In GSR, nodes exchange vectors of link states among their neighbors during routing information exchange. Based on the link state vectors, nodes maintain a global knowledge of the network topology and optimize their routing decisions locally. DSDV periodically advertises its view of the interconnection topology with other mobile nodes within the network to maintain up to date information about the status of the network.

2.1.2. Non-hierarchical On-Demand Protocols

These protocols typically find routes only when needed, unlike the protocols discussed above. Examples are, Dynamic Source Routing (DSR [21]), Temporally-Ordered Routing (TORA [15]), Ad Hoc on-demand Distance Vector (AODV [17]) and Signal Stability-based Adaptive Routing (SSA [44]). DSR, as the name suggests, uses Source-Routing rather than hop-by-hop routing. Each packet to be routed carries the entire ordered list of nodes through which the packet must pass in its header. The advantage of source routing is the fact that intermediate nodes do not need to maintain routing information to forward the packets to the next hop. This feature coupled with the on-demand nature of the protocol, eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols.

TORA is designed to minimize reaction to topological changes. A key concept in its design is that it decouples the generation of potentially far-reaching control message propagation from the rate of topological changes. Control messaging is typically localized to a very small set of nodes near the change without having to resort to a dynamic, hierarchical routing solution with its attendant complexity. The Ad-Hoc On-Demand Distance Vector (AODV) algorithm enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to establish and maintain an ad-hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. AODV also defines timely responses to link breakages. The operation of AODV is loop free, and by avoiding the Bellman-Ford, "counting to infinity" problem offers quick convergence when the ad-hoc network topology changes.

SSA protocol uses the signal strength and stability of the individual hosts as route selection criteria. It believes that selecting the nodes which exhibit the strongest signals for the maximum amount of time, leads to longer-lived routes and less route maintenance. The route initiates a route discovery broadcast on demand; it propagates to the destination; the destination chooses a route and returns a route reply.

2.1.3. Non-hierarchical protocols for large networks

The Zone Routing protocol (ZRP [7]) is designed for large networks. In ZRP each node has a zone that is defined as all nodes that are within 'n' hops of the node. Routing within a zone uses a protocol such as DSR. For sending a packet to a node outside the zone, the source sends a route request packet to nodes at the periphery of it's zone. These nodes query their zones or their peripheral nodes.

2.1.4. Hierarchical Routing protocols

These protocols rely on the construction and maintenance of a hierarchy in the ad hoc network. One set of protocol uses a clustering algorithm at the lowest level. Communication between nodes from two different clusters takes place via the cluster heads. Examples of some protocols in this category include SPINE [19] and Cluster Based Routing Protocol (CBRP [10]). A spine is a self-organizing network structure. SPINE requires only partial topology information at each spine node, consisting of the spine structure, dependents of each spine node, propagation of long-lived links, and snooped routing information from alongside routes. CBRP protocol divides the nodes of the ad hoc network into a number of overlapping or disjoint clusters in a distributed manner. A cluster head is elected for each cluster to maintain cluster membership information. Inter-cluster routes are discovered dynamically using the cluster membership information kept at each cluster head.

2.1.5. Location based protocols

Position based routing algorithms eliminate some of the limitations of topology-based routing by using additional information about the physical position of the participating nodes. Commonly, each node determines it's own position through the use of GPS or some other type of positioning device. A location service is used by the sender of a packet to determine the position of the destination. The routing decision at each node is then based on the destination's position contained in the packet and the position of the forwarding node's neighbors. Examples of such protocols include Distance Routing Effect Algorithm for Mobility (DREAM [2]), Location-Aided Routing (LAR [12]) and Grid Location Service (GLS [45]).

In GLS, each mobile node periodically updates a small set of other nodes (location servers) with its current location. A node sends its position updates to its location servers without knowing their actual identities, assisted by a pre-defined ordering of node identifiers and a pre-defined geographic hierarchy. Queries for a mobile node's location also use the pre-defined identifier ordering and spatial hierarchy to find a location server for that node.

LAR can be used as a sub-protocol for any protocol that employs route discovery (i.e. any on-demand protocol) in order to reduce route discovery by restricting the search area. This results in a significant reduction in the routing messages. DREAM is built around two observations. One, called the distance effect, uses the fact that the greater the distance separating the two nodes, the slower they appear to be moving with respect to each other. The second idea is that of triggering the sending of location updates by the moving nodes autonomously, based only on a node's mobility rate. Based on the routing tables, the directional algorithm sends messages in the recorded direction of the destination node, guaranteeing delivery by following the direction with a given probability.

2.2. Scalability Issues

Hierarchical protocols perform poorly in the presence of fast moving nodes (because the hierarchies need to be re-computed resulting in a high routing message cost) and in the presence of large numbers of nodes (because of message overhead for cluster maintenance tends to be high). The non-hierarchical minimum cost protocols scale even less because of the high cost of link state or distance vector exchange. Non-hierarchical on-demand protocols also perform poorly because of the basic need to broadcast route discovery packets. Caching helps alleviate this problem but still the scalability is limited. The location-based protocols like DREAM require nodes to predict the future of distant nodes. Unfortunately, the predictability of a node's future position is questionable in networks with random movement of nodes. Thus, there is a need for nodes to frequently refresh their location information via broadcasts. While ZRP purports to be designed for large networks, its performance does not scale well because its route discovery process is as bad as flooding.

2.3. *Ad Hoc Network Protocols Studied*

In this section, we briefly discuss the basic mechanism of the four protocols, viz., DSDV, DSR, TORA, AODV, that were compared against SLURP.

2.3.1. DSDV

DSDV broadcasts periodic routing updates to all its neighbors. It uses a hop-by-hop routing scheme in that; each node maintains a routing table with the information for the best next-hop for each of the other nodes about which it knows about. A route is selected either if the sequence number of the new route is higher than the old one or if the sequence numbers are the same but the metric (distance) is less. It also has a method to detect link-breakage. When any node decides that the link is broken, it broadcasts that information with a sequence number higher than the current sequence number and with an infinite metric. Each of the nodes, which receive these packets, updates its routing table accordingly unless it hears otherwise.

2.3.2. DSR

DSR is a source routing protocol. It consists of two basic mechanisms called Route discovery and Route maintenance. Whenever a node needs to obtain a route to a destination, it sends out a Route discovery packet broadcast. it is answered with a Route reply either from one of the intermediate nodes which has a recent enough information about the node or by the destination itself. Whenever a node discovers that the topology has changed and the older routes are unusable, then the Route maintenance part of the protocol sends a Route Error notification packet. The sender cannot use the previously established route. It either restarts the Route discovery mechanism or uses another route that might be available from its cache.

2.3.3. TORA

TORA is different from other protocols in that a logically separate copy of TORA is spawned separately for each destination. It is based on a link-reversal algorithm whose reaction is a temporally ordered sequence of diffused computations. Similar to Route discovery, it sends out a QUERY packet containing the address of the destination to which it wishes to obtain the route. This packet propagates through the network until the destination or till a node which already has the route to the destination. This node sends an UPDATE packet listing its height with respect to the destination. Every other node that receives this UPDATE packet sets its height greater than the advertised height of the neighbor, which sent the packet. This has the effect of creating a series of directed links from the original sender of the QUERY to the node that initially generated the UPDATE. When a node discovers the route to the destination is no longer valid it adjusts its height so that it is the local maximum with respect to its neighbors and broadcasts an UPDATE packet. If the node has no neighbors of finite height with respect to this destination, then the node instead attempts to discover a new route as described earlier. When a node detects a network partition, it generates a CLEAR packet that resets routing state and removes invalid routes from the network.

2.3.4. AODV

AODV combines the concepts of both DSDV and DSR. Similar to DSR, whenever a node needs a route to a destination, it broadcasts a ROUTE REQUEST packet to its neighbors together with the last known sequence number for that destination. This packet is flooded through the network until any intermediate node, which has a route for the destination, answers it or until it reaches the destination itself. The response is sent as a ROUTE REPLY packet. Each node participating in the route discovery process establishes a backward route to the source of the packet. Route Maintenance is either through link-layer detection schemes or by sending periodic HELLO packets. When a link breakage is detected, it is propagated to all the nodes that might use the link as part of the route.

3. OVERVIEW OF SLURP

The following is a discussion of SLURP's basic design. An insight into the genesis and factors considered in design as explained in SLURP [42] is provided in this section.

The routing information for source destination pairs tends to become stale very quickly in large networks as longer paths have a shorter time for failure (a path fails when any link on in breaks). This also happens in networks where nodes travel at high speeds because the time to path failure is shorter. In both these environments caches (for routing information) also have shortened lifetimes. This implies that finding or maintaining precise routing information can be expensive.

To develop a scalable routing protocol it is important to constantly maintain approximate location information about nodes in the network. It is also necessary to only find accurate routes to specific nodes when there are packets to be sent to them. This approach is similar to the one adopted in DREAM [2]. A node first determines the approximate location of the destination when it needs to send a packet. It then uses a simple geographic routing protocol to send the packet. The benefits of this two-step approach is that there is reduced cost as we need not maintain the routing information, while at the same time it provides the ability to find relatively inexpensive routes when required. SLURP is based on a combination of approximate geographical routing and a simple static mapping procedure to maintain the approximate location of the nodes. In the following sections we first discuss the algorithm used to find the geographical location of a node followed by an explanation of the approximate geographic routing protocol.

3.1. Location management

We assume that all nodes in the system are equipped with GPS [46](Global Positioning System) hardware that provides them with their current location. We further assume that the ad hoc network is in a rectangular region of dimension l x h. All nodes are equipped with radios and we assume that they are aware of the identity of their neighbors, provided by the MAC layer. Finally, we assume that all nodes have a universally unique ID (such as an NIC or IP address).

Figure 3.1 illustrates an ad hoc network with the area segmented into rectangular regions of dimension a x b. All of these home regions have well-defined IDs. This ID is derived by concatenating the x and y coordinates of the bottom left and upper-right corners. Next, we assume that there exists a static mapping f that maps a node's ID into a specific region (called its home region),

f(Node ID) ---> Region ID.

f is a many-to-one mapping. It is static and known to all nodes of the ad hoc network. In figure 3.1, node D is associated with home region 7. The function f needs to satisfy several properties including:

- f should be such that every region has the same node density. The reason being that the queries has to be evenly distributed through the network.
- Entry/departure of nodes from the network should be transparent to f.
- f should not depend on the shape or size of the geographical area covered by the ad hoc network. For example, the same f should work in different disaster areas.

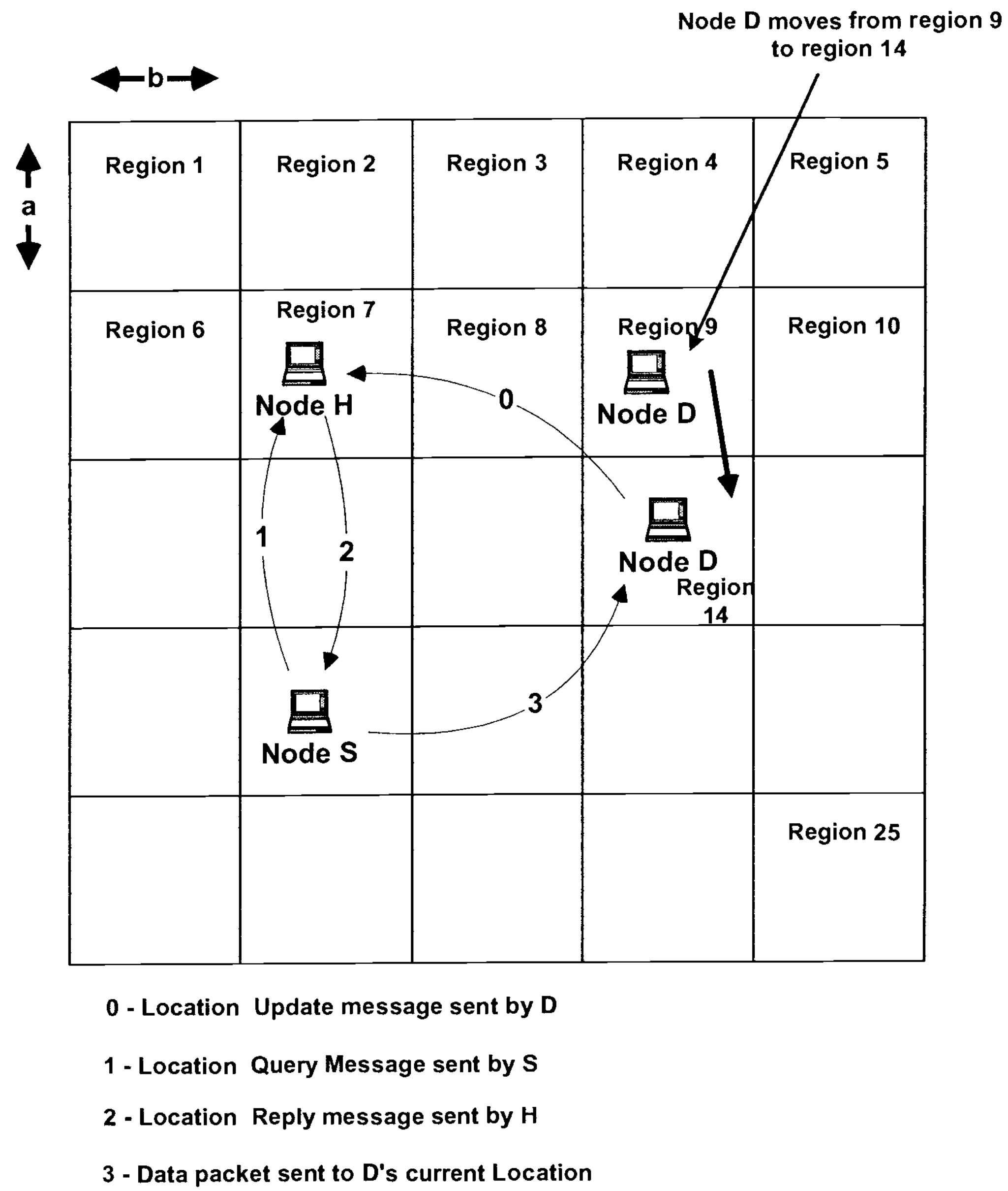

FIGURE 3.1 Location Management in Ad-Hoc Networks

An example of a function that satisfies these criteria is:

$$f(ID) = g(ID) \bmod k.$$

Where g (ID) = [0, 1, 2, . . . N] is a random number generating function that outputs a random number in some large range. The seed for the function is the ID. k is the number of home regions in the network. We assume that every node has a table containing home region addresses (the geographical coordinates) indexed by k. The function f provides us with an accurate mapping even if the network is made up of non-contiguous areas.

3.1.1. Location tracking algorithm

Given the above structure of the network, we use the following algorithm to maintain approximate location information about nodes in the network.

Step 1. Mobility triggered updates: A mobile node always informs the nodes currently present in its home region of its location. In other words the identity of the region it is located in (this is conceptually similar to MobileIP [3] wherein a roaming user informs it's current location to the home agent). This information is only updated when the node moves from one region to another. In Figure 3.1, node D sends a location update message to nodes in region 7 when it moves from region 9 into region 14.

Step 2. Update in home region: The location update message sent by a node to its home region is broadcast to all nodes in the home region. Thus, D's location update message will be broadcast to all nodes in region 7. Note that the update message is unicast until it reaches one or more nodes located in region 7, whereupon one of these nodes broadcasts it to all others in region 7.

Step 3. Locating a node: Suppose node S in Figure 3.1 needs to send packets to D, its location needs to be determined first. For this, S uses f to find the home region of D. It then sends a message to this region enquiring about D's current location. In our example, S sends a query message to region 7. The first node in region 7 to receive this message (say node H) responds with D's current location (i.e., region 14).

S then sends packets to D's current location. The next section explains the geographic routing algorithm used to achieve this. The same geographic routing algorithm is also used above in steps 1-3 to send location update/query messages.

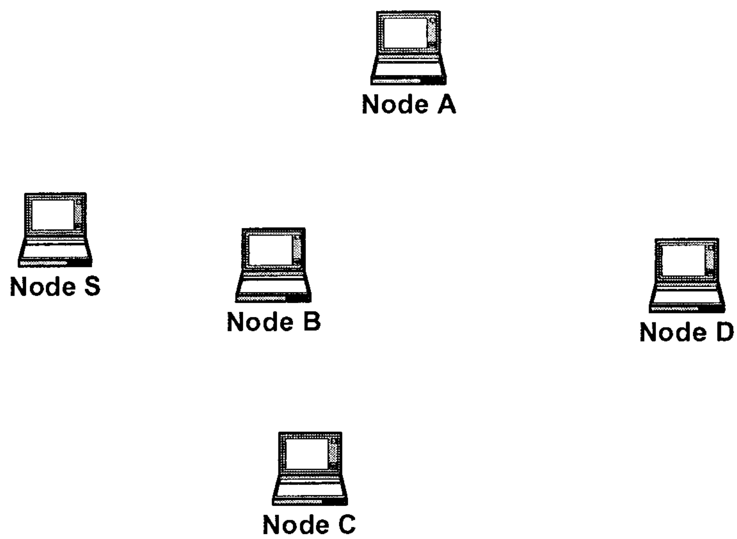

FIGURE 3.2 Example of MFR

3.2. Approximate geographic routing

For approximate geographic routing we use an algorithm based on MFR (Most Forward with fixed Radius) [8] without backwards progression. In this second part of SLURP, we deal with transmitting packets when the destination's region is known but we do not have topological routing tables. To explain MFR, consider Figure 3.2 where node S needs to send a packet to D. It has three neighbors, A, B and C. MFR selects the next hop based on closest physical distance to D, say A. A then repeats the same procedure. However, there arises a problem when there are no forward neighbors to a node. This is addressed in later sections. Initially during data transmission, the precise co-ordinates of a destination are not known. We only know the current region information. Thus to route the packet using MFR, the source determines the next hop approximately using the center of the destination's current region as the location.

Any node in the region, that receives the packet, can use the cached route of the destination to route the packet. If not it broadcasts a location discovery packet (as in DSR [21]) to nodes within the same region. The packet eventually reaches the destination and the destination in turn generates a response, which contains the route. MFR is used to send a data packet to the region, followed by source routing to get the packet to the destination. The Source-Routing protocol is very good since the regions tend to be very small.

3.3. Protocol details

This section contains the details of SLURP. Location management is the most complex aspect of SLURP. It is assumed that there are well-defined boundaries for each region which each node knows and every node can determine the center of the region and its own location in the region using GPS.

3.3.1. Location discovery

Every node maintains a location cache. It contains information about the node ID, co-ordinates, current region ID and best neighbor for routing to that node. If a node S wants to send a packet to D it first needs to find the location. So it sends a location_discovery packet. This packet contains the destination's ID, the source's ID, and current location and sequence number. The sequence number is required to ignore the earlier location_discovery packets. The location of the source is then updated by the neighbor's in their location cache. The neighbor who was the designated next hop then uses MFR to forward the packet. This process continues till the packet reaches some node in the destination's home region. This node then generates a location_reply packet containing the destination's region ID. The age of this information is not required to be verified as if a node is stationary then the information though high in age will still be correct.

3.3.2. Empty home region

What if there are no nodes in the home region of the destination? To manage this we define the nodes home region in two levels

Level 1- there is at least one node in the home region.

Level 2- home region is empty. In this case all neighboring regions are treated as home regions. In figure 3.1, if region 13 is empty, the home region for any node whose home region was region 13 now includes regions 7, 8, 9, 12, 14, 17, 18, 19.

This definition can be extended further. This increases the size of the home region thereby increasing the cost of broadcasting. In relation to the location_discovery package the level indicates the region to be queried (either the home region or the extended home region).

3.3.3. Maintaining home regions

The nodes in a region may change over time since the nodes are mobile. In SLURP, all nodes in the region keep the location information of every other node in the region. All nodes keep a node_list, which contains a list of all nodes located within the same region as them. This list is updated as new nodes arrive or leave. When a new mode moves into a region it sends a home_location_request to its neighbors requesting the (x, y) co-ordinates and the age of the location of all the nodes in the region. In order to avoid a storm of replies a formula with random waiting time is used. A node also updates its departure to the previous region when it leaves that region, thereby helping the other nodes in the previous region to modify their lists. When the last node leaves a region it generates a broadcast message in the eight regions surrounding the empty region. This broadcast message contains the location information for all nodes that regard the, now empty, region as their home region. When another node enters the empty region it collects the location information for nodes registered in the region from any nodes in the neighboring regions. This is done using a home_location_request_2 packet broadcast in the neighboring regions.

3.3.4. Location update

A location_update packet is sent to the home region when a nodes X leaves its current region and enters a new region. Since each node knows its home region a location_update packet is sent towards the center of the home region. Any node receiving the location_update packet in the home region updates its cache and broadcasts a location_update_broadcast packet with the new location information to all nodes in the home region. Each of these broadcasts contains a list of neighbors of the sender. Upon receipt the neighbor then compares this list with its own neighbors list. If there is at least one neighbor in the list, which is not on the received list, then it generates a new location_update_broadcast, which contains a list that is a union of the two lists. Every node broadcasts this list at most once.

3.3.5. Data packet delivery

A data packet contains the source ID, source location, destination ID and, destination location. Until the data packet reaches the current region of the destination MFR is used. After which a DSR-like source routing protocol is used for local delivery of the packet. Intermediate nodes update their cache with the location information contained in the packet. If there is a failure in delivery along the route, a message to that effect is transmitted back to the source and the source then rediscovers the location of the destination.

3.4. Optimizations

Several optimization techniques improve the performance of the protocol

1. There is always a trade-off between faster initial location discovery and accurate determination of the location. Hence if the cache information is recent enough, say 10 seconds or less, location replies are issued by intermediate nodes themselves. Similarly, if a node overhears a location discovery it sends an unsolicited location_reply packet. One possibility of this is that a reply storm may result. This

is in turn avoided by the random waiting period formula. The formula is a function of the age of the cached entry.

2. If there is no node in the direction of the destination, a node, say X generates an alternative_hop_ request packet to all its neighboring nodes. Any node, say Y that can generate the next hop forward to the destination sends a reply. The packet is then routed through the node Y. Snooping and dropping redundant replies as mentioned previously reduce the reply storm. Node X avoids the same procedure (request and reply) for the subsequent packets to the same destination in order to save resources by registering Y as a temporary next hop in its location cache (or a different data structure) until X finds its own next hop.

3. In large networks, a group of nodes may move together as a unit. If a node wants to send packets to a destination then instead of sending a location_discovery broadcast to the home region it first asks the other nodes in the group about the location. A limited hop location_discovery packet achieves this. This packet starts with propagation level 1 and increase gradually to a certain maximum. Propagation of this type of packet is limited only to the nodes whose distance from the source is smaller than or equal to the propagation level in the packet. The source uses a timeout mechanism to increment the propagation level.

4. MFR is used every time to decide on the forward next hop. This is fine in location discovery, reply, and update because their end-to-end service happens only infrequently. However, data transmission between a pair of end-to-end nodes typically consists of a sequence of packets. Since the speed of intermediate nodes is low compared to data transmission the next hop can be stored in a routing table till the time it becomes unavailable.

5. When there is a frequent on going session between a source and destination and the destination is moving fast then periodic location update from the fast moving destination is necessary. Since the intermediate hops are dynamically selected it is only enough to have the destination accurately mapped.

In LAR [12], two methods have been proposed to reduce the size of the area where route discovery packets are propagated. This method can be adopted in a high-speed destination small region scenario. Consider the illustration in figure 3.3. D is the location where the destination was at time t0. D1, D2, and D3 are three possible farthest locations of D computed by source S using (current time t - t0) * (speed of D at t0). Any node that satisfies the condition of MFR for at least one of the three points relays the data packet. In figure 3.3, node S selects A, B, and C as the set of next hops because A, B, and C satisfy the condition of MFR for D1, D2, and D3, respectively.

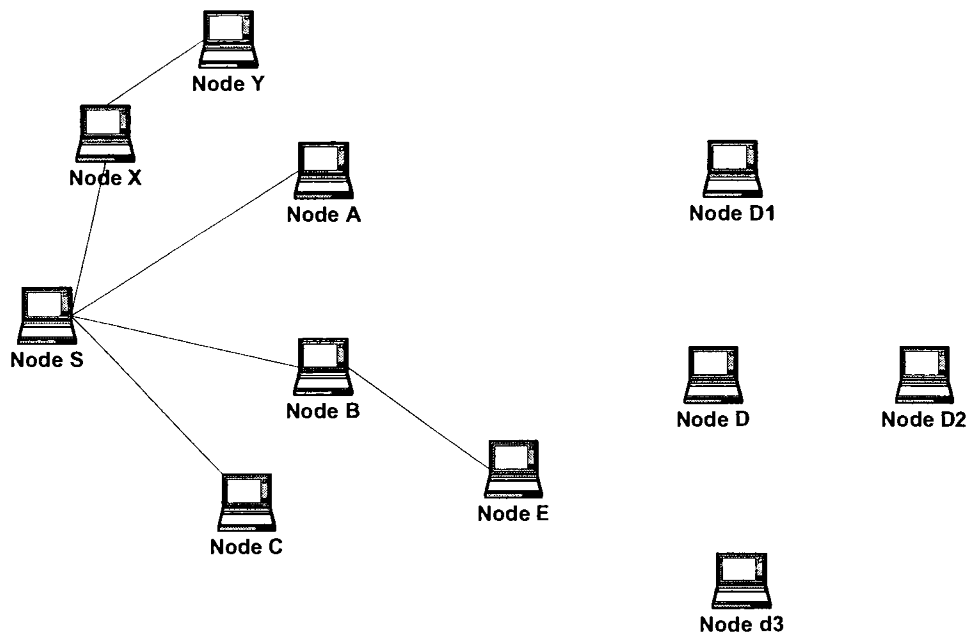

FIGURE 3.3: Selective flooding of data

6. If the ad hoc network moves into new geographic areas then the region boundaries need to be redefined and updates to be sent to all the nodes. A network management layer on top of the routing layer performs this function. The management layer performs the following tasks

- Periodically obtain location updates of all nodes and use this information to determine if the network's geographic spread is changing.
- Recomputed region boundaries (and perhaps create new regions).
- Inform all nodes of the new region boundaries.

This management function can be easily added to existing network management protocols. An example is defined and implemented in Ad hoc Network Management Protocol (ANMP) [6]), in which the network manager does collect geographic information about nodes. The functions specified above can be added to ANMP in a simple way.

4. SIMULATION ENVIRONMENT

4.1. Network Simulator (Ns)

Ns is a discrete event simulator targeted at networking research. Ns was originally developed by the University of California at Berkeley and the VINT project [6]. Ns also include substantial contributions from other researchers from Universities and industry, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. Ns provide support for network simulation for transport layer protocols such as TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Mobile node forms the core of Ns mobility model. These wireless Mobile nodes can move in a given topology, transmit/receive signals from/to wireless channels. Wireless support in Ns consists of a stack of LL, ARP, MAC, and IFQ. Wireless support in Ns allows simulations in ad hoc networks, wireless LANs, and sensor networks.

Ns as described in [25], is an object-oriented simulator, written in C++, with an OTcl interpreter as a front-end. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the OTcl interpreter (also called the interpreted hierarchy). The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. On one hand, detailed simulations of protocols requires a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks, run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important. Ns meets both of these needs with two languages, C++ and OTcl. C++ is fast to run but slower to change, making it suitable for detailed protocol implementation. OTcl runs much slower but can be changed very quickly (and interactively), making it ideal for simulation configuration. Ns (via tclcl) also provides glue to make objects and variables appear on both languages.

4.2. Methodology

The methodologies adopted in the thesis to test and compare the protocols are similar to the ones described in [4]. The constants that were used for the simulation of AODV, DSDV, DSR and TORA were chosen as specified by the authors of the protocols and from the analysis done as in [4]. The protocol constants for SLURP were chosen to provide optimal performance as specified in [42]. The goal of our simulation was to test the protocols' reaction to change in network topology and their ability to constantly deliver packets. The simulation consisted of constantly changing topologies under a variety of workloads, in which several nodes were trying to send packets to some destination. These simulations do not try to mirror a real world scenario in any way but rather tests the protocols under a range of conditions.

The number of nodes that participated in the simulation was fixed at 50 which where moving arbitrarily in a rectangular area of 1500 x 300m. A rectangular region was chosen to force longer routes while maintaining a high node density. The radio characteristics constants were chosen to approximate the Lucent Wave LAN [39] direct sequence spread spectrum. Identical movement scenarios and connection patterns were provided for each of the protocols to enable fair comparison between the protocols. Each simulation consisted of a scenario file, which describes the movement patterns and instance of motion of the various participating nodes. There is also a connection pattern file, which describes the exact sequence of packets originated by each node together with the time at which each change in motion or packet origination is to occur. We pre-generated 210 different pairs of scenario files with varying movement patterns and traffic loads, and then ran all five routing protocols against each of these scenario files.

Our protocol SLURP, due to its design is extremely scalable, but the inherent limitation of the simulator and the resources prevented us from running a simulation consisting of 1000's of nodes. So, we describe scalability in terms of the change in characteristics, when the number of active sources is increased and the nodes become mobile.

4.2.1. Movement Model

Nodes in the simulation move according to the "random waypoint" model as described in [29]. The movement scenario files were created based on the utilities provided by the Monarch Project at Rice University [43]. The nodes in our simulation start the simulation by being stationary for pause time seconds, then moves to a randomly selected destination at a uniform speed between 0m/s and 1m/s, waits there for pause time seconds and proceeds so on as previously described. This pattern is repeated for the entire simulation time of 900 seconds. Seven different pause times were used in the simulations: 0, 30, 60, 1120, 300, 600, 900 seconds. A pause time of 0 corresponds to continuous motion and a pause time of 900 corresponds to no motion at all. To average out the irregularities and the sensitivities caused due to the random nature of these scenarios, 10 different movement pattern files were used for each of the pause times. Each of the 5 protocols were run with the same set of movement pattern files.

4.2.2. Communication Model

The goal of the simulation is to compare and analyze the different protocols under different scenarios but same conditions. For this purpose, the traffic sources had to be a CBR (Constant Bit rate) sources. A TCP source was not used because TCP does not offer a confirming load. It sends packets based on its own perception of the network's ability to carry load. Because of this, the position of the node and the time at which it sends a packet will be different preventing a direct comparison between the protocols.

Varying the number of CBR sources was approximately equivalent to varying the sending rate as indicated by the experiments done by the Rice Monarch group. Hence, for these simulations we chose to fix the sending rate at 4 packets per second, and used three different communication patterns corresponding to 10, 20, and 30 sources. The packet size was fixed at 64 bytes, as a high packet size like 1024 bytes might cause congestion effects on the physical channel.

All communication patterns were peer-to-peer, and connections were started at times uniformly distributed between 0 and 180 seconds. The three communication patterns (10, 20, and 30 sources), taken in conjunction with the 70 movement patterns, provide a total of 210 different scenario files for each of the five routing protocols.

4.2.3. Scenario Characteristics

The characteristics of the scenarios under which the protocols were tested are presented in this section. In order to measure the challenge our scenarios placed on the routing protocols, we measured the lengths of the routes over which the protocols had to deliver packets, Number of unreachable destinations, number of route changes and the total number of link connectivity changes in each scenario.

When each data packet is originated, an internal mechanism (General Operations Director - GOD) of ns-2 calculates the shortest path between the packet's sender and its destination. The packet is labeled with this information, which is compared with the number of hops actually taken by the packet when received by the intended destination. The shortest path is calculated based on a nominal transmission range of 250m for each radio and does not account for congestion or interference that any particular packet might see.

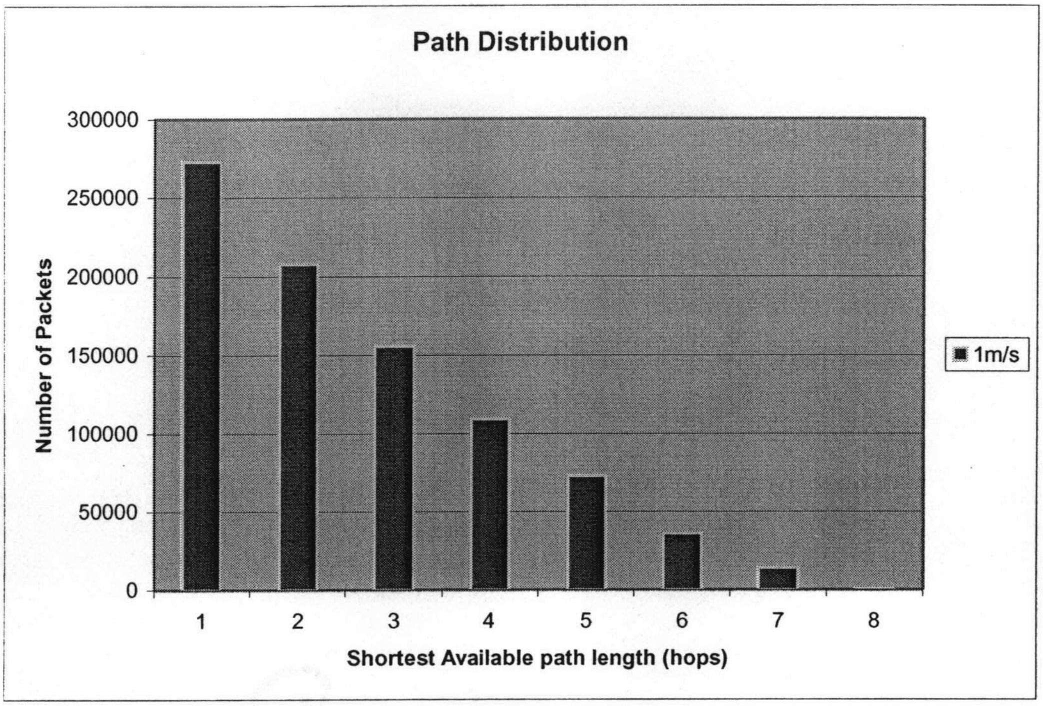

FIGURE 4.1 Distribution of the shortest path available to each application packet originated over all scenarios

Figure 4.1 shows the distribution of shortest path lengths for all packets over the 210 scenario files. The height of each bar represents the number of packets for which the destination was the given distance away when the packet was originated.

The average data packet in our simulations had to cross 2.62 hops to reach its destination, and the farthest reachable node to which the routing protocols had to deliver a packet was 8 hops away.

Table 4.1 shows the average number of link connectivity changes that occurred during each of the simulation runs for each value of pause time. We count one link connectivity change whenever a node goes into or out of direct communication range with another node. It also shows the number of destinations that are created randomly, which are unreachable with the given radio propagation model. Finally, it shows the number of route changes that are simulated.

Pause Time	Destination Unreachable	Route changes	Link changes
0	0	57587	8562
30	96	52625	8296
60	0	51558	7464
120	593	47064	7173
300	624	39222	5355
600	0	21013	2371
900	0	0	0

TABLE 4.1 Average number of Link changes, route changes, and destination unreachable.

4.2.4. Metrics

The metrics used in this paper are similar to the ones suggested by the Rice Monarch project. In comparing the protocols, we chose to evaluate them according to the following three metrics:

Packet delivery ratio: The ratio between the number of packets originated by the "application layer" CBR sources and the number of packets received by the CBR sink at the final destination.

Routing overhead: The total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each transmission of the packet (each hop) counts as one transmission.

Path optimality: The difference between the number of hops a packet took to reach its destination and the length of the shortest path that physically existed through the network when the packet originated.

Packet delivery ratio is an important indicator of the success of the protocol. It determines the maximum throughput that the network can support. This metric characterizes both the completeness and correctness of the routing protocol.

Routing overhead is a measure of the scalability of the protocol. It also determines the degree to which it will function in congested or low-bandwidth environments, and its efficiency in terms of the consumption of node battery power. Protocols that send large numbers of routing packets can also increase the probability of packet collisions and may delay data packets in network-interface transmission queues.

Path optimality measures the ability of the routing protocol to efficiently use network resources by selecting the shortest path from a source to a destination.

5. SIMULATION RESULTS

In this section we compare the simulations against the metrics as explained in the section 4.3. We first compare the various protocols against each other in terms of Packet efficiency, Routing efficiency, and Path optimality. We then take a closer look at the protocols to see how scalable they are in terms of the number of sources that are actively participating in the simulation. For all simulations, the number of sources was 10, 20 or 30 with each source sending 4 packets per second and all connections were peer-to-peer.

5.1. Comparison Summary

Figure 5.1 through 5.3 highlights the relative performance of the 5 protocols, when the traffic load was changed between 10, 20, 30 sources, all of them averaged out. Before comparison, it should be taken into account that, the ns-2 implementations of DSR, DSDV, TORA, and AODV are highly stable and contains all the optimizations specified in the protocols [4].

The packet delivery ratio of DSR, DSDV and AODV are all highly efficient. Almost all the packets are delivered when the nodes are immobile. TORA and SLURP due to the nature of the protocols, does not guarantee high packet efficiency. However, it should be noted that, many of the optimizations proposed for SLURP are not implemented, due to the inherent limitation of the simulator and because of the time constraints. This will be an area of future work. We are confident that, even when the basic protocol performs considerably well, the optimizations should make it more efficient than the other protocols. It was also noted that TORA suffered severely due to congestion, and many of the experiments had to be re-run to get better performance

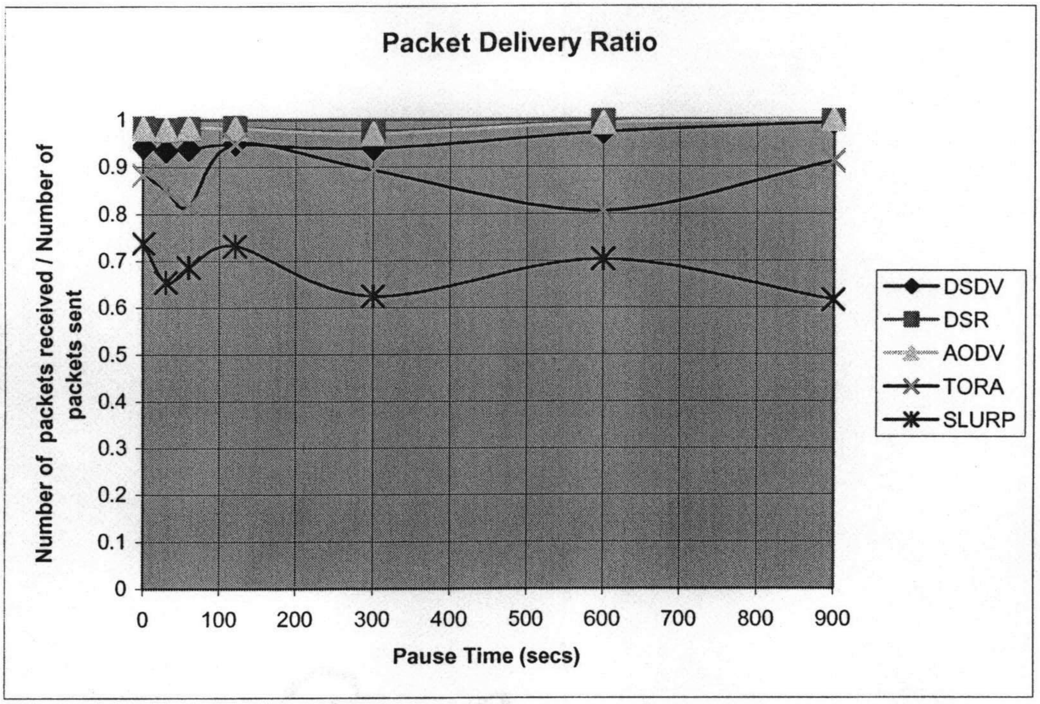

FIGURE 5.1 Comparison of Packet Delivery Ratio

In terms of Routing Overhead, SLURP out-performed DSDV and was comparable in performance to DSR and AODV. TORA suffered heavy packet loss due to the mobility of the nodes. DSDV was pretty much constant irrespective of the mobility, due to the periodic nature of the protocol. In DSR and AODV, the number of routing packets increases linearly with increase in mobility due to the on-demand nature of the protocol. SLURP, also suffers from mobility, though the rate of increase is not as much as DSR or DSDV.

SLURP and TORA does not guarantee optimal number of hops. As expected the number of packets which follow the optimal path is less when compared to DSR, DSDV and AODV, which advertises shortest paths. SLURP believes that the energy saved in minimizing the routing packets is more important than guaranteeing the optimal route, in a constantly changing environment.

Further analysis of the protocols is provided in the later sections.

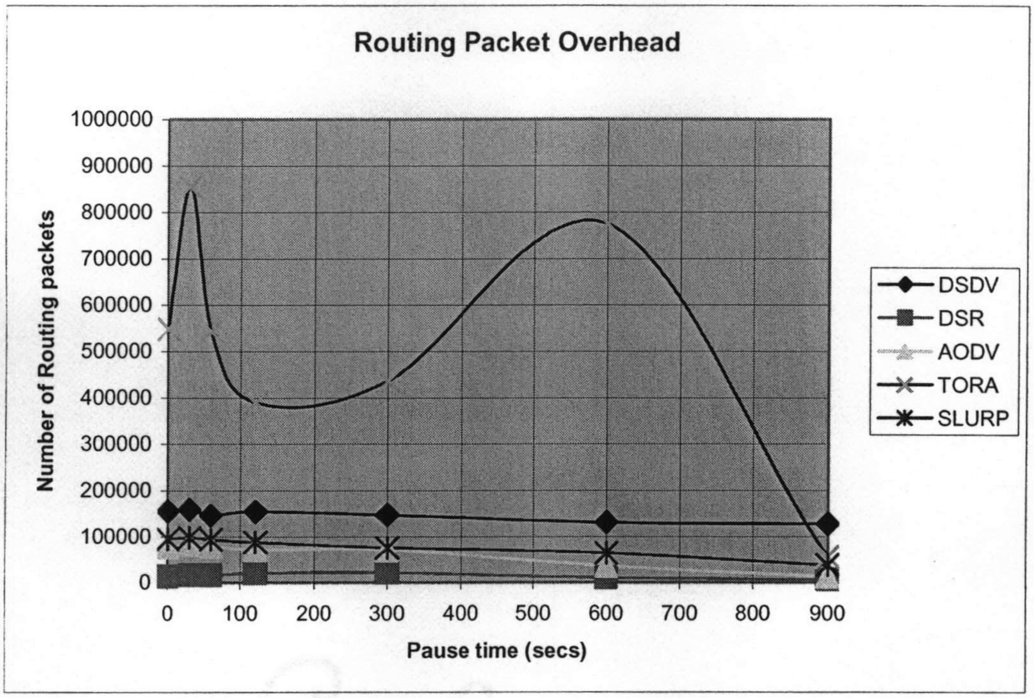

FIGURE 5.2 Comparison of Routing Packet Overhead

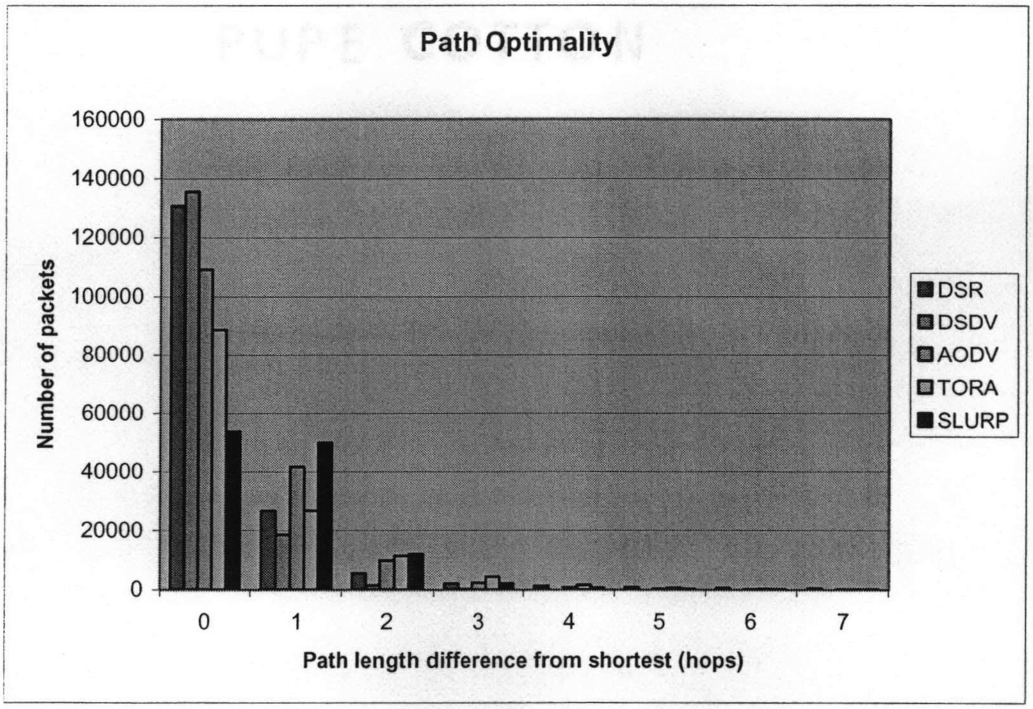

FIGURE 5.3 Path optimality

5.2. Packet Delivery Ratio Details

Figure 5.4 through 5.8 shows the variability of the various protocols' packet efficiency when the number of sources that are actively participating is increased from 10 to 20 to 30.

SLURP is extremely scalable. We can see from the graphs that neither the increase in the number of active sources nor the mobility of the nodes has a significant effect on the packet efficiency. This is an encouraging sign because, when all of the optimizations are implemented, this scalable feature is useful for deployment in large networks, consisting of 1000's of nodes. It should be worth mentioning that, the packet efficiency difference between the concurrent simulations was much less and the protocol was more stable than any other protocol tested. Even though the packet efficiency is not as good as DSR or DSDV, the optimizations will certainly improve the performance.

DSR and DSDV seem to scale very well. As mentioned earlier, these protocols suffer severely when the number of nodes in the simulation increases. AODV suffers a huge loss of performance due to mobility and due to the number of simultaneous routes it has to maintain. The characteristics of TORA are highly dynamic and depend upon the current state of the network. This means that it does not always converge properly. The authors of TORA might have to provide certain optimizations to improve the performance and make it more stable. It should be noteworthy that when the parameters tested against were the same the packet delivery ratio of TORA dropped to about 30% in some of our simulations and was as high as 90% in some others.

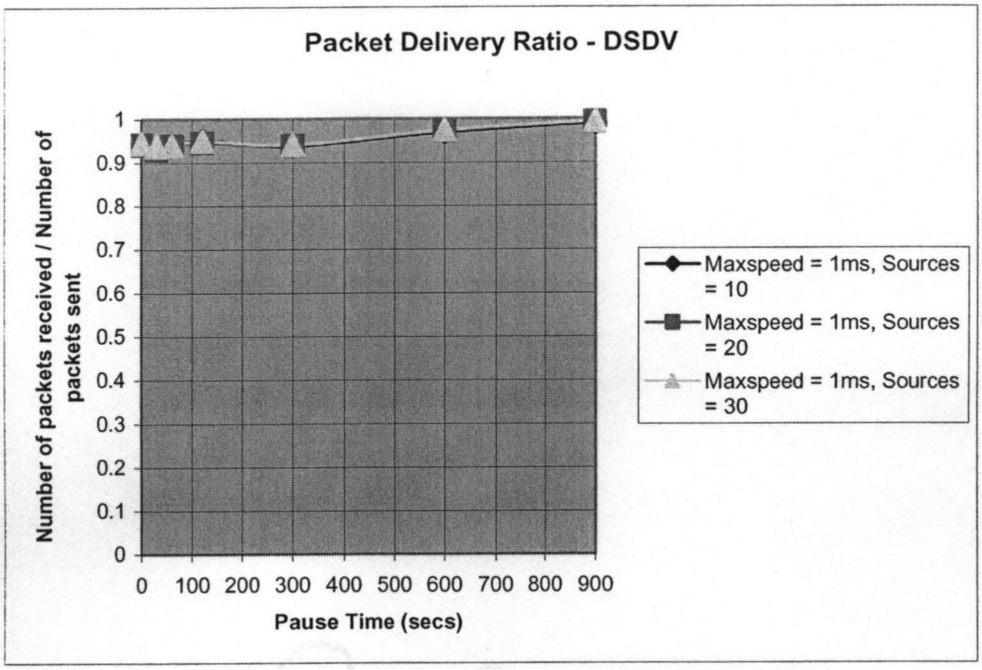

FIGURE 5.4 DSDV Packet Delivery ratio

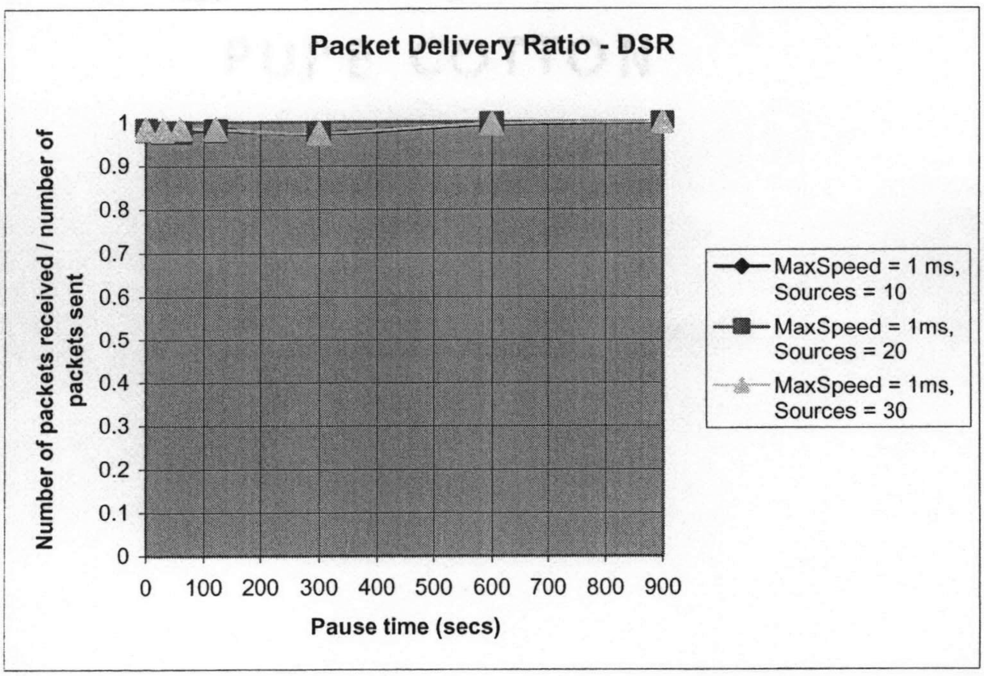

FIGURE 5.5 DSR Packet Delivery ratio

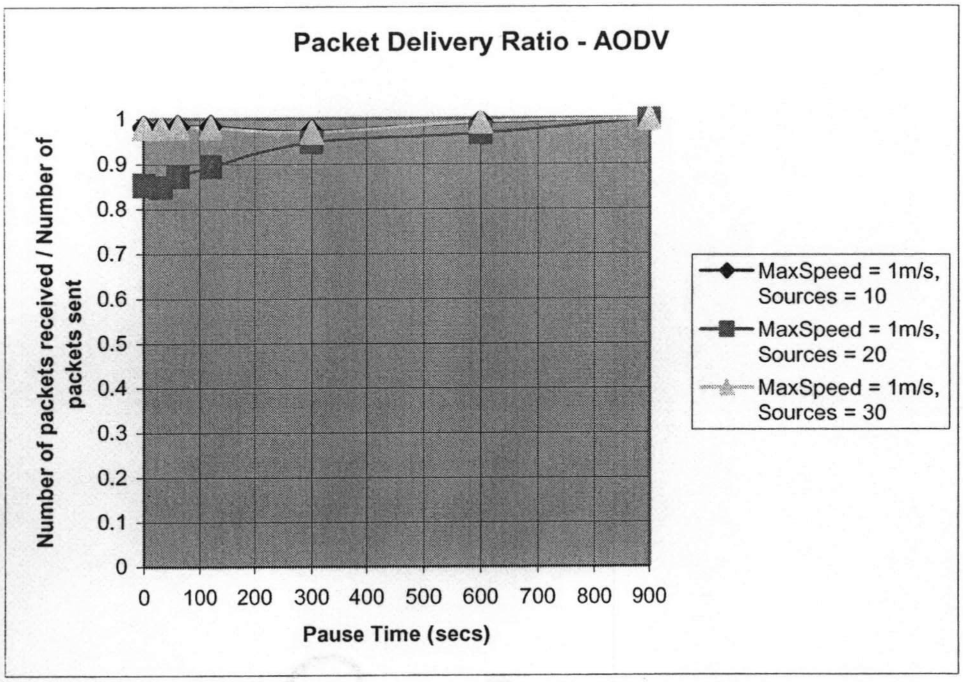

FIGURE 5.6 AODV Packet Delivery ratio

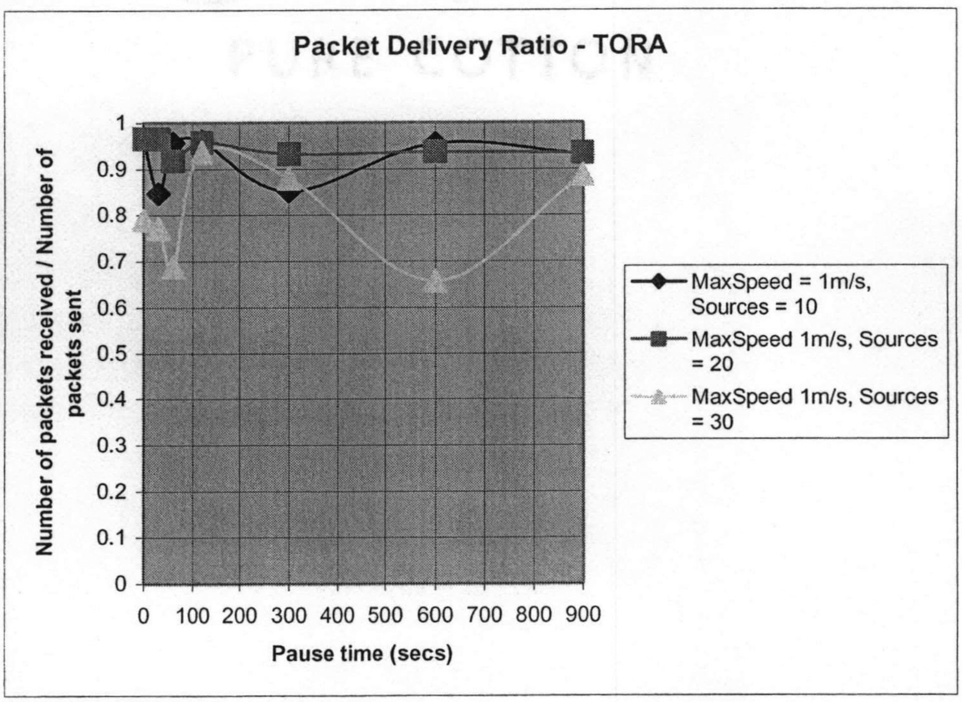

FIGURE 5.7 TORA Packet Delivery ratio

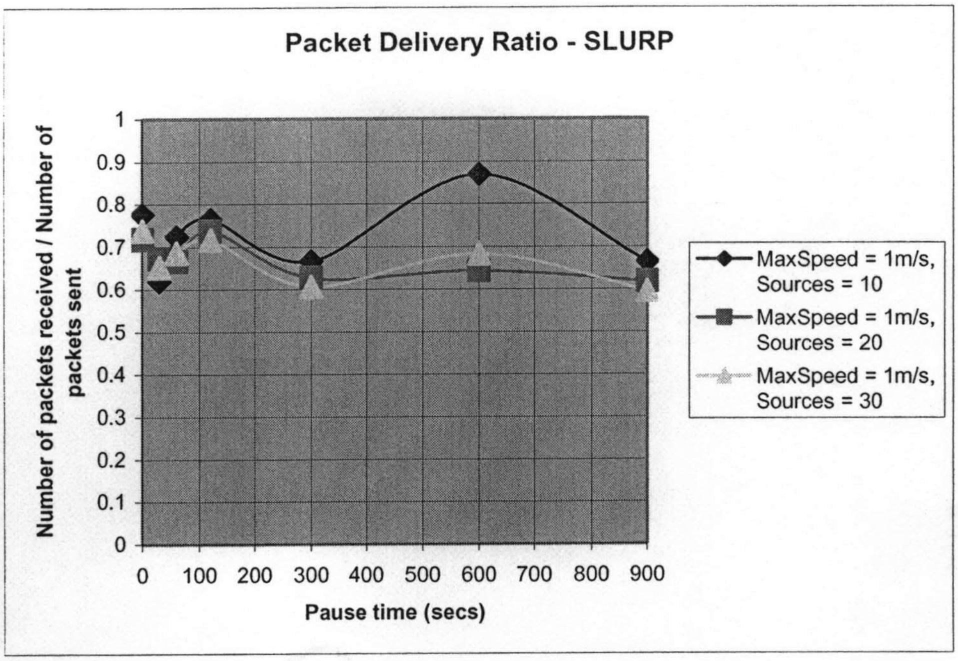

FIGURE 5.8 SLURP Packet Delivery ratio

5.3. Routing Packet Overhead Details

Figure 5.9 through 5.13 shows the routing packet overhead of the different protocols, when the number of active sources participating in the network was increased from 10 to 20 to 30.

The Routing packet overhead of SLURP is significantly lower when compared to TORA, DSDV, and AODV and comparable to DSR. Fig 17 shows the scaling effect of SLURP. SLURP, linearly scales depending upon the number of packets that needs to be delivered. For Example, at 10 active sources, the number of packets sent by the CBR agent is about 2100, and at 20 sources it is about 4200 packets and at 30 sources it is about 6300 packets. The graph shows an almost equivalent scale of increase in the number of routing packets.

DSDV has a constant overhead irrespective of the number of sources participating, and it is very high at about twice the number required by SLURP. DSR scales excellently well due to the on-demand nature of the protocol. In

AODV, mobility has a huge effect on the routing packets. At constant motion and high number of connections, the routing packets are significantly higher. Hence it does not scale properly. TORA is out of the ordinary. The number of routing packets required is in a different scale than all the other protocols. In certain cases it underwent a congestive collapse generating millions of packets.

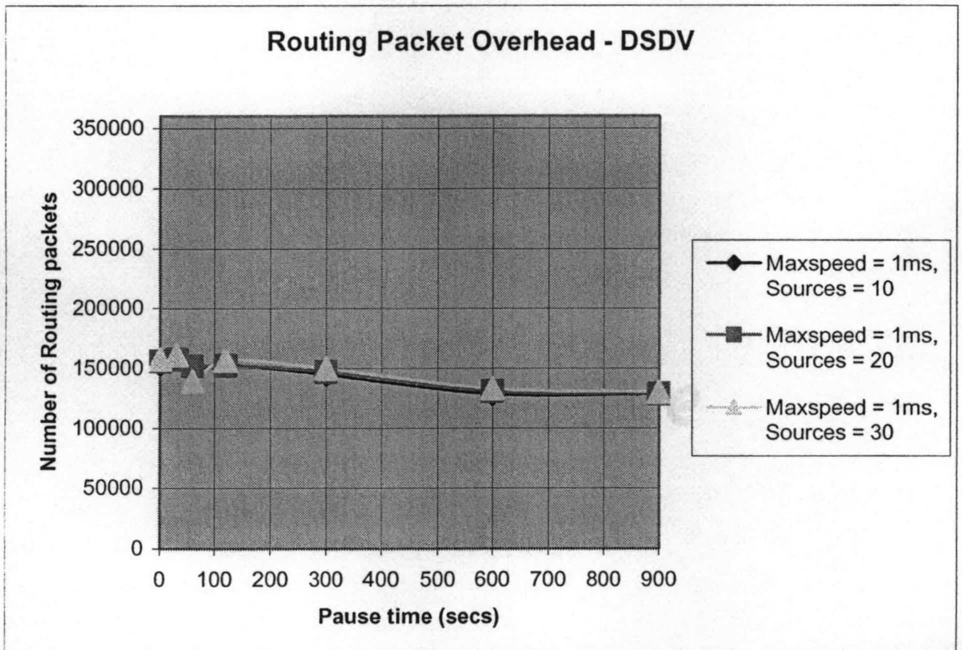

FIGURE 5.9 DSDV Routing Packet Overhead

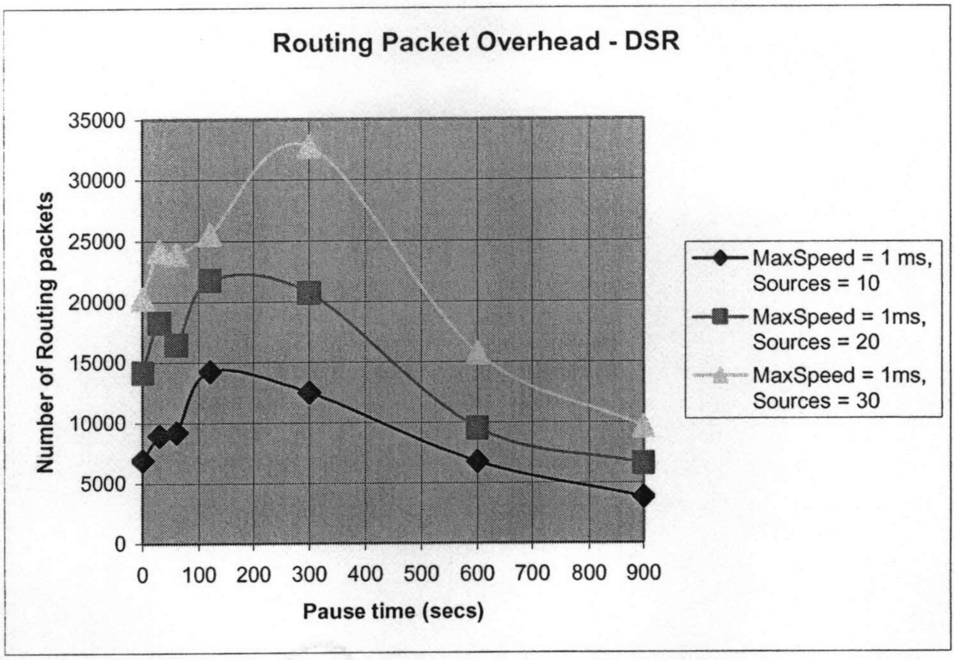

FIGURE 5.10 DSR Routing Packet Overhead

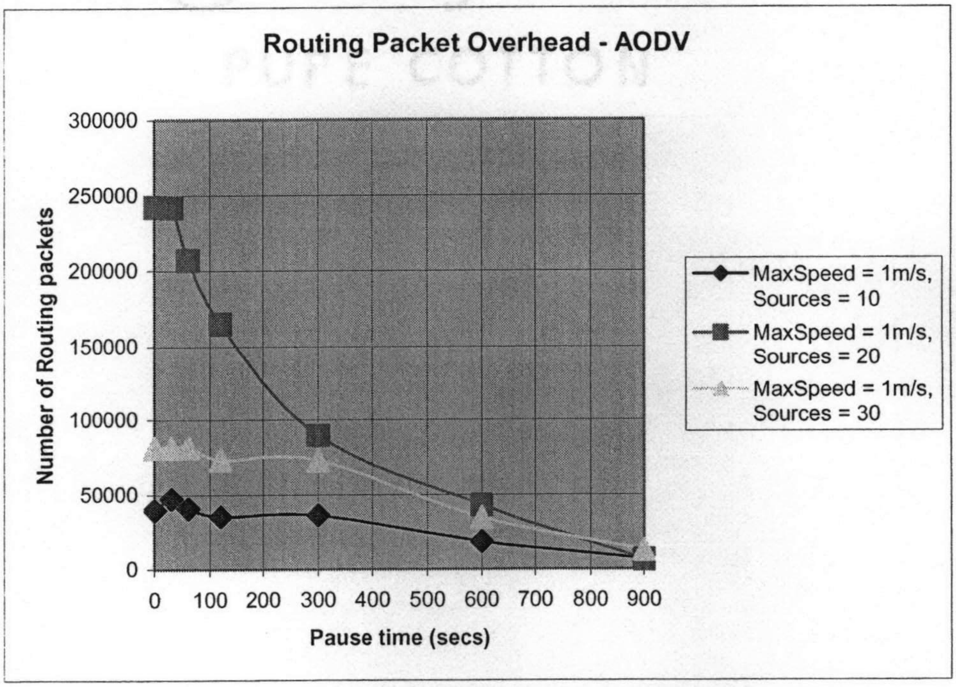

FIGURE 5.11 AODV Routing Packet Overhead

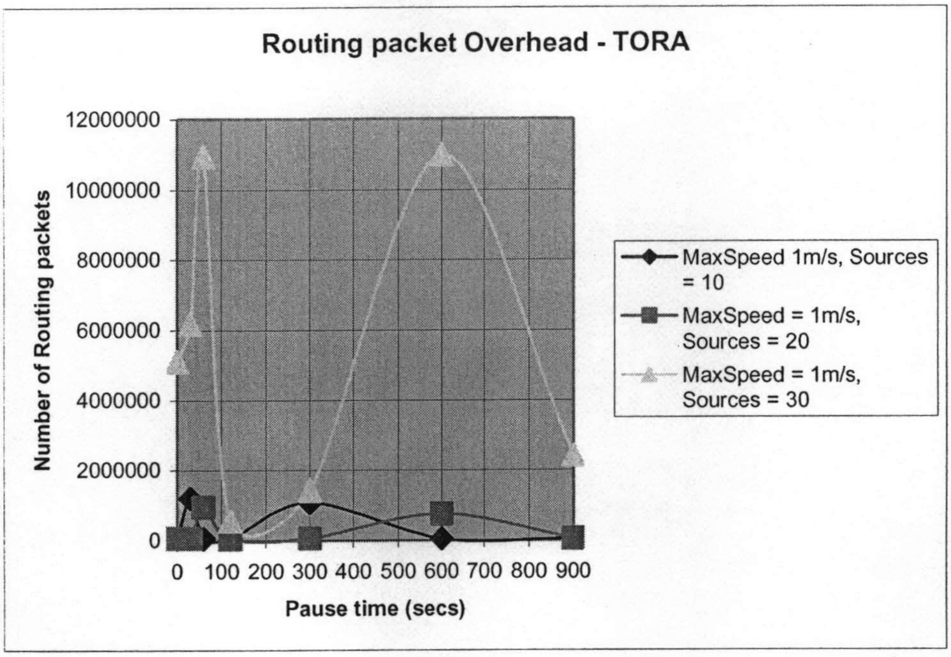

FIGURE 5.12 TORA Routing Packet Overhead

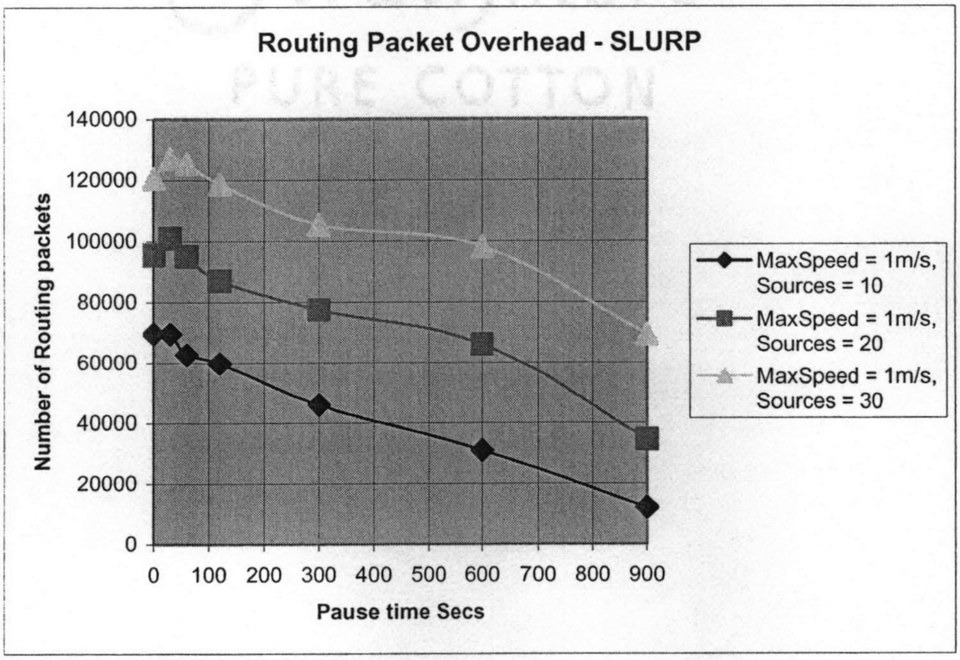

FIGURE 5.13 SLURP Routing Packet Overhead

6. CONCLUSION

A new routing protocol SLURP was presented in this thesis. SLURP scaled well for networks with large number of active sources and in networks were nodes are constantly moving. It performed as well as DSR in reducing the routing protocol overhead. It out-performed DSDV, AODV and TORA in the same category. It out-performed TORA in all categories of comparison. Without the optimizations in effect it did not have a very good packet delivery ratio, as one would expect from a good routing protocol. Nevertheless, it showed encouraging signs of good performance. The specific reason that makes SLURP so well behaved is the use of location tracking to maintain approximate location information for the nodes in the network.

Scalability is achieved by taking advantage of the following properties of SLURP: uses de-centralized node locations; does not operate using global broadcasting (as in DSR); does not use hierarchies thus eliminating the overhead of hierarchy management; does not depend on time (no table exchange as in DSDV); and finally, routing is made largely insensitive to significant topology changes since routes are found on demand using MFR. Several optimizations have been discussed in this thesis and we are confident that when considered and implemented, SLURP will definitely out perform most of the existing protocols.

There are numerous challenges that are facing the research of ad hoc networks. Many of these challenges are still un-addressed. Security is a significant challenge in position based routing mechanisms, since position information can be readily deduced. In a typical battlefield example, this means that when a soldier is caught, the enemy will know the whereabouts of the entire platoon. This is unacceptable. New location services and forwarding strategies have to give a lot of thought to achieve more secure networks.

Greedy forwarding, as explained in [45] showed that position-based packet forwarding techniques including MFR has been a topic of active research for the past several years and has always out-performed other forwarding strategies. There is still room for many changes and the two main issues that need more attention are the strategy employed to determine the next hop when a packet is forwarded, and the repair mechanism used when greedy packet forwarding fails. The service by the wireless hardware will most definitely determine the choice of the next hop. Inaccurate position information also needs to be more efficiently handled by greedy algorithms.

The future research in ad hoc networks should also include strategies to allow connectivity between the Internet and the ad hoc networks. A proposal is given in Terminodes and Grid projects [45, 46]. It proposed an approach in which there was a location-based approach at the local level and topology-based routing over long distances and for Internet integration.

BIBLIOGRAPHY

[1] David B. Johnson, "Routing in ad hoc networks of mobile hosts", in Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, pages 158–163, December 1994.

[2] S. Basagni, I. Chlamtac, V. Syrotiuk and B. WoodWard, "A distance routing effect algorithm for mobility (DREAM)", in Proceedings of 4th Annual Conference on Mobile Computing and Networking (Mobicom 1998), Dallas, TX (October 25–30, 1998).

[3] Gabriel Montenegro, Phil Roberts and Basavaraj Patil, "IP Routing for Wireless/Mobile Hosts (mobileip)", http://www.ietf.org/html.charters/mobileip-charter.html (March 08, 2001).

[4] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", in Proceedings of 4th Annual Conference on Mobile Computing and Networking (Mobicom 1998), Dallas, TX (October 25–30, 1998).

[5] T. -W. Chen and M. Gerla, "Global state routing: A new routing scheme for ad-hoc wireless networks", in Proceedings of IEEE International Communications Conference (ICC'98) (1998).

[6] W. Chen, N. Jain and S. Singh, "ANMP: Ad hoc network management protocol", IEEE Journal on Selected Areas in Communications 17(8) (August 1999) 1506–1531.

[7] Z. Haas and M. Perlman, "The zone routing protocol (ZRP) for ad hoc networks", draft-ietf-manet-zone-zrp-01.txt (August 1998).

[8] T.-C. Hou and V.O.K. Li, "Transmission range control in multihop packet radio networks", IEEE Transactions on Communications 34(1) (January 1986).

[9] Theodore S. Rappaport, "Wireless Communications: Principles and Practice", Prentice Hall, New Jersey, 1996.

[10] M. Jian, J. Li and Y.C. Tay, "Cluster based routing protocol (CBRP) functional specification", Internet Draft, draft-ietf-manet-cbrp-spec-00.txt.

[11] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks", in Proceedings of 6th Annual Conference on Mobile Computing and Networking (Mobicom 2000), Boston, MA (August 6–11, 2000).

[12] Y.B. Ko and N.H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks", in Proceedings of 4th Annual Conference on Mobile Computing and Networking (Mobicom 1998), Dallas, TX (October 25–30, 1998).

[13] J. Li, J. Jannotti, D.S.J. DeCouto, D.R. Karger and R. Morris, "A scalable location service for geographic ad hoc routing", in Proceedings of 6th Annual Conference on Mobile Computing and Networking (Mobicom 2000), Boston, MA (August 6–11, 2000).

[14] S. Murthy and J.J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks, Mobile Networks and Applications", Special Issue on Routing in Mobile Communications Networks (1996).

[15] V.D. Park and M.S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks", in Proceedings of IEEE INFOCOM' Kobe, Japan (1997).

[16] C.E. Perkins and P. Bhagwat, "Routing over multi-hop wireless network of mobile computers", in Mobile Computing, eds. T. Imielinski and H.F. Korth (Kluwer Academic, 1996) pp. 183–205.

[17] C. Perkins, "Ad hoc on demand distance vector (AODV) routing", Internet Draft, draft-ietf-manet-aodv-00.txt (November 1997).

[18] S. Singh, "A mathematical toolbox for analyzing the performance routing protocols in ad hoc networks", Manuscript.

[19] R. Sivakumar, B. Das and V. Bharghavan, "The clade vertebrata: Spines and routing in ad hoc networks", in: Proceedings of IEEE Symposium on Computers and Communications (1998).

[20] 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.

[21] Josh Broch, David B. Johnson, and David A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Internet-Draft, draft-ietf-manet-dsr-00.txt, March 1998. Work in progress.

[22] M. Scott Corson and Anthony Ephremides, "A distributed routing algorithm for mobile wireless networks", Wireless Networks, 1(1):61–81, February 1995.

[23] M. Scott Corson, S. Papademetriou, Philip Papadopoulos, Vincent D. Park, and Amir Qayyum, "An Internet MANET Encapsulation Protocol (IMEP) Specification", Internet-Draft, draft-ietf-manet-imep-spec-01.txt, August 1998. Work in progress.

[24] M. Scott Corson and Vincent D. Park, "An Internet MANET Encapsulation Protocol (IMEP) Specification", Internet-Draft, draft-ietf-manet-imep-spec-00.txt, November 1997. Work in progress.

[25] Kevin Fall and Kannan Varadhan, editors, "ns notes and documentation", The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1997. Available from http://www-mash.cs.berkeley.edu/ns/.

[26] Bernd Freisleben and Ralph Jansen, "Analysis of routing protocols for ad hoc networks of mobile computers", in Proceedings of the 15th IASTED International Conference on Applied Informatics, pages 133–136, Innsbruck, Austria, February 1997. IASTED-Acta Press.

[27] IEEE Computer Society LAN MAN Standards Committee, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-1997", The Institute of Electrical and Electronics Engineers, New York, New York, 1997.

[28] P. Bose, P. Morin, I. Stojmenovic and J. Urrutia, "Routing with guaranteed delivery in wireless ad hoc networks", in Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM'99) (August 1999).

[29] David B . Johnson and David A. Maltz, "Dynamic source routing in ad hoc wireless networks", in Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[30] Phil Karn, "MACA - A new channel access method for packet radio", in Proceedings of the 9th Computer Networking Conference, pages 134–140, September 1990.

[31] Barry M. Leiner, Robert J. Ruth, and Ambatipudi R. Sastry, "Goals and challenges of the DARPA GloMo program", IEEE Personal Communications, 3(6):34–43, December 1996.

[32] National Science Foundation, "Research priorities in wireless and mobile communications and networking: Report of a workshop held March 24–26, 1997, Airlie House", Virginia. Available at http://www.cise.nsf.gov/anir/ww.html .

[33] Vincent D. Park and M. Scott Corson, "Temporally-Ordered Routing Algorithm (TORA) version 1: Functional specification", Internet-Draft, draft-ietf-manet-tora-spec-00.txt, November 1997. Work in progress.

[34] Vincent D. Park and M. Scott Corson, "A performance comparison of TORA and Ideal Link State routing", in Proceedings of IEEE Symposium on Computers and Communication '98, June 1998.

[35] Charles E. Perkins and Pravin Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers" in Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications, pages 234–244, August 1994. A revised version of the paper is available from http://www.cs.umd.edu/projects/mcml/papers/Sigcomm94.ps.

[36] David C. Plummer, "An Ethernet address resolution protocol: Or converting network protocol addresses to 48.bit Ethernet addresses for transmission on Ethernet hardware", RFC 826, November 1982.

[37] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in Proceedings of 6th Annual Conference on Mobile Computing and Networking (Mobicom 2000), Boston, MA (August 6–11, 2000).

[38] Neil Siegel, Dave Hall, Clint Walker, and Rene Rubio, "The Tactical Internet Graybeard Panel briefings", U.S. Army Digitization Office. Available at http://www.ado.army.mil/Briefings/Tact%20Internet/index.htm, October 1997.

[39] Bruce Tuch, "Development of Wave LAN, an ISM band wireless LAN", AT&T Technical Journal, 72(4): 27–33, July/August 1993.

[40] Gary R. Wright and W. Richard Stevens, "TCP/IP Illustrated, Volume 2 : The Implementation", Addison-Wesley, Reading, Massachusetts, 1995.

[41] Seung-chul M. Woo and Suresh Singh, "Scalable Routing Protocol for Ad Hoc Networks", Wireless Networks 7, 513–529, 2001. 2001 Kluwer Academic Publishers. Manufactured in The Netherlands.

[42] M.Woo and Suresh Singh, "Scalable Location Update-based Routing Protocol for ad hoc networks", Technical Report TR00.01, Department of Electrical and Computer Engineering, Oregon State University, March 2000

[43] Rice University Monarch Project, www.rice.monarch.edu

[44] Rohit Dube, Cynthia D.Rais, Kuang-Yeh Wang, and Satish K.Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad hoc Mobile Networks", IEEE Personal communications, pages 36-45, February 1997.

[45] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris, "A Scalable Location Service for Geographic Ad Hoc Routing", in Proceedings 6th Annual Conference on Mobile Computing and Networking (Mobicom 2000), Boston, MA (August 6-11, 2000).

[46] Martin Mauve, and Jorg Widmer, "A Survey on Position-Based Routing in Mobile Ad Hoc Networks", IEEE Network, pages 30-39, November/December 2001.