An Improved Model for the Dynamic Routing Effect Algorithm for Mobility Protocol

by

Karthik Ramakrishnan

A thesis

presented to the University of Waterloo

in the fulfilment of the

thesis requirement for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada 2004

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Karthik Ramakrishnan

Abstract

An ad-hoc network is a packet radio network in which individual mobile nodes perform routing functions. Typically, an ad-hoc networking concept allows users wanting to communicate with each other while forming a temporary network, without any form of centralized administration. Each node participating in the network performs both the host and router function, and willing to forward packets for other nodes. For this purpose a routing protocol is needed.

A novel approach utilizes the uniqueness of such a network i.e. distance, location and speed of the nodes, introducing a Distance Routing Effect Algorithm for Mobility (DREAM). The protocol uses the *distance effect* and the *mobility rate* as a means to assure routing accuracy. When data needs to be exchanged between two nodes, the directional algorithm sends messages in the recorded direction of the destination node, guaranteeing the delivery by following the direction. The improved algorithm suggested within this thesis project includes an additional parameter, direction of travel, as a means of determining the location of a destination node. When data needs to be exchanged between two nodes, the directional algorithm sends messages in the recorded direction of the destination node, guaranteeing the delivery by following the direction. The end result is an enhancement to the delivery ratio, of the sent to the received packet. This also allows the reduction in the number of control packets that need to be distributed, reducing the overall control overhead of the Improved Dream protocol.

Acknowledgements

My deepest gratitude goes to my supervisor, Dr. George Freeman, for having accepted me as a graduate student and providing continuous support through my graduate studies.

I would like to thank Professor Sagar Naik and Professor Murat Uysal for reading this thesis and providing their insightful comments and suggestions.

I would like to acknowledge Professor Violet Syrotiuk, University of Texas Dallas, and Associate Professor Tracy Camp, Colorado School of Mines, for sharing their experience with implementation of the DREAM protocol, as well as their valuable assistance.

I would like to thank Sayee Mahalingam for his continuous support and encouragement of the last three years.

This thesis is dedicated to my parents from whom I have learnt so much and owe my gratitude for making it all possible.

Contents

List of Illustrations

1. Introduction

Wireless communication between mobile users is becoming more popular than ever before. This has been fed by the growing technological advances in laptop computers and wireless data communication devices, such as wireless modems and wireless LANs. Conceptually, two different kinds of wireless networks exist, but the difference between them may not be as obvious as it may seem. The first kind and most often used today is a wireless network built on top of a “wired” network and thus creates a *reliable infrastructured wireless network.*

The wireless nodes connected to the wired network and able to act as bridges in a network of this kind are called base-stations. The major issue in such a network is related to the concept of handoff, where one base station tries to hand off a connection to another seamlessly, without any noticeable delay or packet loss. Another practical problem in networks based on cellular infrastructure is that it is limited to places where there exists such a cellular network infrastructure.

The other kind of network is one where there is no infrastructure in place except for the participating mobile nodes. This is referred to as an *infrastructureless network* or more commonly an ad-hoc network. The term ad-hoc translates to “improvised” or “not organized” and refers to the dynamic nature of such a network. All or some nodes within an ad-hoc network are expected to be able to route data-packets for other nodes in the network who want to reach nodes beyond their own transmission range. This is called *peer level multi hopping* and is the base for ad-hoc networks that constructs the interconnecting infrastructure for the mobile nodes.

This form of networking is limited in range by the individual nodes’ transmission ranges and is typically smaller compared to the range of cellular systems. This is not to imply that the cellular infrastructure approach is superior to the ad-hoc network approach. Ad-hoc networks have several advantages compared to traditional cellular systems. These advantages include:

- On demand setup
- Fault tolerance
- Unconstrained connectivity

Ad hoc networks do not rely on any pre-established infrastructure and can therefore be deployed in places lacking traditional infrastructure. This is useful in disaster recovery situations and places with non-existing or damaged communication infrastructure where rapid deployment of a communication network is needed. Given the dynamic nature of the ad hoc network, routing protocols used in ordinary wired networks are not well suited for this kind of an environment. They are usually built on periodic updates of the routes and create a large overhead in a relatively empty network and also cause slow convergence to changes in the topology. Currently, there does not exist any standard for a routing protocol for ad hoc networks, instead this is a work in progress. Many protocols are in the process of evaluation. This thesis attempts to study one of the many proposed routing protocols and attempts at making some performance enhancing improvements on the protocol design.

1.1. iDREAM – An Introduction

Ad hoc networking protocols can be broadly classified as either proactive or reactive. Proactive protocols maintain up to date route information for all nodes within the network. When data needs to be sent to a destination node, the sender node most usually has the route path information, generally the next hop to it, and can be used immediately. On the other hand, reactive protocols obtain a route to the destination node only when a message needs to be sent in an “on-demand” fashion, i.e., the transmission of a message is preceded by a route discovery phase.

Regardless of whether a protocol is proactive or reactive, current routing protocols for ad hoc networks are required to store route information similar to routing protocols for static networks, essentially as a sequence of nodes. In proactive protocols, this information is generally in the form of a next hop table lookup at each node along the route. In a reactive protocol the result of a route discovery control message is the route to be used as an explicit sequence of nodes in order to reach the destination.

The aim of this masters thesis paper is to discuss the performance and benefits of a *location based* routing protocol which uses the location information stored within the routing table of each node, for all other nodes within the network. The location information refers to the geographic coordinates that can be obtained from and by the use of the *Global Position System* (GPS) [1]. The location based protocol specifically considered here is the Distance Routing Effect Algorithm for Mobility or *DREAM*. The DREAM protocol can be considered proactive in the sense that a mechanism is defined for the dissemination and updating of location information. When the sender node S needs to send a message to the destination node *D*, it uses the location information for *D* to obtain *D*’s direction, and transmits the message to all its one hop neighbors in the direction of *D*. The subsequent nodes repeat the same procedure until the destination node is reached. This effectively results in using a reactive approach, as individual nodes in the path determine the next hop in an on-demand manner.

In the DREAM algorithm, each node participates in the transmission of control messages containing the current location of a particular node to all other nodes within the network, in the form of *Location Update* messages.. The frequency of such updates is determined by the distance factor and mobility rate of each node. The enhancement proposed within this thesis introduces the direction of travel information of the particular node in addition to the location and time information, within the location update message. This allows the sender node *S* to calculate the direction of the destination node *D* with a greater accuracy. This would also ensure that a lesser number of next-hop neighbors are chosen when a data packet is sent, effectively reducing the overhead caused by the collaborative transmission mechanism inherent to an ad hoc network.

2. Background

2.1. Wireless Ad-Hoc Networks

2.1.1. General

In areas in where there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use, wireless mobile users may still be able to communicate through the formation of an ad hoc network. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within direct wireless transmission range of each other. Each node participates in an ad hoc routing protocol that allows it to discover “multi-hop” paths through the network to any other node. The idea of ad hoc networking is sometimes also called *infrastructureless networking.* Figure 2-1 Local Ad-Hoc Network shows a simple ad hoc network with three nodes. The outermost nodes are not within transmitter range of each other. However the middle node can be used to forward packets between the outermost nodes. The middle node acts as a router and the three nodes form an ad hoc network.

Figure 2-1 Local Ad-Hoc Network

Ad hoc networks are also capable of handling topology changes and malfunctions in nodes. It is fixed through network reconfiguration. For instance, if a node leaves the network and causes link breakages, affected nodes can easily request new routes. Although there are incremental delays, the network continuous to remain operational.

Wireless ad hoc networks take advantage of the inherent nature of the wireless communication medium. In a wired network, the physical cabling is done a priori, restricting the connection topology of the nodes. Provided two mobile nodes are within transmission range of each other, this restriction is easily overcome within the wireless domain, forming an instantaneous communication link.

2.1.2. Characteristics

The distinguishing characteristic of mobile ad hoc networks is the dynamic topology as a result of the nodes changing their physical location by moving around. This feature favors routing protocols that dynamically discover routes over conventional routing algorithms like distance vector and link state. Another significant characteristic is that every node typical of such a network, has very limited CPU capacity, storage capacity, battery life and bandwidth. This highlights the fact that given the limited power usage, the transmission range is limited in its reach.

2.2. Routing

Given that all packets in the network have to traverse several nodes before reaching the destination node, a routing protocol is essential for the existence of an ad-hoc network. The routing protocol has two main functions, selection of routes for the various source-destination pairs and the delivery of messages to the intended destination. The second function is conceptually straightforward, using a variety of protocols and data structures (routing tables). This thesis is based on applying and evaluating a protocol for the former purpose in order to make the latter possible.

2.2.1. Conventional Protocols

Given the large number of protocols that already exist within the conventional networking realm, it makes sense to apply the same to ad-hoc networks as well. These include the well known link state and distance vector type protocols. The main drawback to using such protocols is that they are designed to be used in a static topology, and experience adverse problems when trying to converge to a steady state in an ad-hoc network with a dynamically changing topology.

Link state and distance vector could work well for an ad-hoc network with low mobility, i.e. a network with a more static topology. The issue that still remains is that both link-state and distance-vector are dependant on periodic control messages. As the number of network nodes can be large, the potential number of destinations is also large. This requires large and frequent exchange of data among the network nodes. This in consideration of the fact that all updates in a wireless interconnected ad hoc network are transmitted over the air, are thus costly in resources such as bandwidth, battery power and CPU usage. Because both link-state and distance-vector try to maintain routes to all reachable destinations, it is necessary to maintain these routes and wastes resources for the same reasons as above.

Another characteristic for conventional protocols are that they assume bi-directional links e.g. that the transmission between two hosts works equally well in both directions. In the wireless radio environment this is not always the case.

Because many of the ad-hoc routing protocols have a traditional routing protocol as the underlying algorithm, it is beneficial to understand the basic operation for conventional protocols like distance vector, link state and source routing.

2.2.2. Link State

In Link state routing, each node maintains a view of the complete topology with a cost of each link. To keep these costs consistent; each node periodically broadcasts the link costs of its outgoing links to all other nodes using flooding (i.e. it distributes the update packet to all nodes within the network without restriction). As each node receives this information, it updates its view of the network and applies a shortest path algorithm to chose the next hop for each destination i.e. the path which results with the lowest cost, after the cost associated with each link within the path have been summed.

Some link costs in a node view can be incorrect because of long propagation delays, partitioned networks, etc. Such inconsistent network topology views can lead to the formation of routing loops. These loops are however short lived; because they disappear in the time it takes a message to traverse the diameter of the network.

2.2.3. Distance Vector

In distance vector, each node only monitors the cost of it outgoing link, but instead of broadcasting this information to all nodes, it periodically broadcasts to each of its neighbors as estimate of the shortest distance to every other node in the network. The receiving nodes then use this information to recalculate the routing tables, by using a shortest path algorithm.

Compared to link state, distance vector is more computationally efficient, easier to implement and requires much less storage space. However, it is well known that distance vector can cause the formation of both short lived and long lived routing loops. The primary cause for this is that nodes choose their next hops in a completely distributed manner based on information that could be stale.

2.2.4. Source Routing

In source routing, each packet contains the complete route to the destination, i.e. the source node originating the packet specifies the complete path the packet must take through the network. The advantage of this approach lies in the removal of the occurrence of any routing loops. Given that the source specifies the routing path, this method is referred to as source based routing. The added overhead in this approach are the larger packets as they contain the complete path information.

2.2.5. Flooding

A common approach to distributing routing or control information is to utilize a broadcast method, whereby the source nodes sends packets to all nodes within the network. Flooding is a common broadcast implementation used within the wireless environment. The source node sends the information to all nodes who are its direct neighbors i.e. in the wireless world, all nodes within transmission range. The neighbors in turn forward these information packets to all nodes within their reach. In this manner, the packets 'flood' the entire network. The information packets are numbered in sequence to prevent stale packets and loops.

3. Ad-Hoc Routing Protocols and Classification

This chapter is focused on capturing the elemental characteristics of an ad-hoc routing protocol, as well as to capture the essence of the various routing protocols that exist in this area. The protocols are listed under a broad classification perspective, in order to capture their underlying philosophy.

3.1. Desirable Properties

Routing protocols for mobile ad-hoc networks need to meet certain criteria in order to be considered suitable for the environment that they are functioning under. These criteria or metrics [2] are mentioned here:

3.1.1. Distributed Operations

The protocol should be distributed in nature and not dependant on any centralized control function or node. The criterion applies to both static and mobile environments. However, in an ad hoc environment the distinguishing factor is that the nodes may enter and leave the network randomly, as well as resulting in a partitioned network due to mobility.

3.1.2. Loop Free

It is desirable that the protocol provides loop-free routes and has fail-safe mechanisms to address loop conditions. This essentially avoids any waste of precious CPU and bandwidth consumption.

3.1.3. Demand-Based Operation

The protocol must be reactive, in order to minimize the control overhead in the network, and thus conserving precious network and node resources. The protocol should only react when needed and should broadcast control information periodically.

3.1.4. Unidirectional Link Support

Most routing algorithms assume bidirectional links and do not function well under unidirectional situations. The wireless environment often cause the presence of unidirectional links, and the ability to make use of them is valuable.

3.1.5. Security

Given the nature of the wireless environment, it may be relatively simple to snoop network traffic, replay transmissions, manipulate packet headers, and redirect routing messages, within a wireless network without appropriate security provisions. Various means of authentication and encryption methods have been discussed [3,4].

3.1.6. Power Convervation

To reduce the number of reactions to topological changes and congestion multiple routes can be used. If a particular route becomes invalid, alternate routes can be used without resorting to expensive route discovery routines.

3.1.7. Quality of Service Support

Depending on the type of application it may be required to provide Quality of Service considerations within the routing protocol. In such situation, e.g. real time traffic support, it maybe be necessary to incorporate the same into the routing protocol.

It is not to say that all routing protocols have all of the above desired properties today. Current protocols are still an exercise in determining innovative ways to find paths to the destination node and maintain the appropriate routing tables in an efficient manner. However, most of the protocols are works in progress, looking for ways to improve their efficiency and extend their functionality.

The remainder of the chapter will now concentrate on different routing protocols and analyze them from a theoretical perspective.

3.2. MANET Protocols

MANET (Mobile Ad-Hoc Networks) is a working group within the IETF (Internet Engineering Task Force), working to develop a peer-to-peer mobile routing capability in a purely mobile, wireless domain. The purpose of this working group is to standardize IP routing protocol functionality suitable for wireless routing application within both static and dynamic topologies. Currently, the group is working under a revised plan, targeting the promotion of a number of core routing protocol specifications to EXPERIMENTAL RFC status (i.e., AODV, DSR, OLSR and TBRPF).

Currently there are seven routing protocol drafts, under:

- AODV – Ad Hoc On Demand Distance Vector [5]
- ZRP – Zone Routing Protocol [6]
- TORA/IMEP – Temporally Ordered Routing Algorithm / Internet MANET Encapsulation Protocol [7,8,9]
- DSR – Dynamic Source Routing [10, 11]
- CBRP – Cluster Based Routing Protocol [12]
- CEDAR – Core Extraction Distributed Ad Hoc Routing [13]
- AMRoute – Ad-Hoc Multicast Routing Protocol [14]
- OLSR – Optimized Link State Routing Protocol [15]

Of the various protocols in the MANET list, I have chosen to discuss the theoretical aspects of a few of the above mentioned protocols. The intention here is to give some background information for an understanding of the various protocols, their underlying philosophies and scope. The DREAM protocol highlighted under this thesis is a location based routing protocol, unlike the more conventionally based protocols discussed herewith. The simulation studies performed here compare the relative performance of the original DREAM protocol, the enhancement to DREAM implemented in this paper and the DSR protocol. Discussion of the DREAM protocol is conducted in the next chapter.

There are several classification methods and taxonomies used to group Mobile Ad-hoc Network routing protocols. However, the broadest classification may group most of the protocols as being either *proactive* or *reactive* in nature.

Precomputed routing is also called *proactive* or *table-driven routing* [16]. In such a method, the routes to all destinations are computed *a priori*. In order to compute routes in advance, nodes need to store the entire or partial information of the network topology. In order to keep the information current and up-to-date, nodes need to update their information periodically or whenever the link state or network topology undergoes changes. The advantage of a precomputed route driven environment is that when information needs to be sent to a destination, the route is readily available, resulting in low latencies and reaction times. Most of the current routing protocols utilize shortest path algorithms, modified to fit into the mobile environment. This includes the *DSDV* [17] and more recently the WRP (Wireless Routing Protocol) [18]. The disadvantage to this approach is that information about routes may never be used. The other big disadvantage is that the dissemination of routing information consumes a lot of network bandwidth – precious in the wireless environment, in addition to device resources such memory, to maintain possibly large routing tables, and energy in transmission of control packets. The overheads mentioned become significant in scenarios where the mobility rates of the nodes are high, and the protocols fail to keep routing tables current and hence become inefficient or fail.

On-demand routing is also referred to as *reactive* routing. In this method, the route to a destination may not exist in advance, and is computed only when needed. When a source needs to send packets to a destination, it first finds a route or several routes to a destination. This process is referred to as the *route discovery* phase. Once the route(s) are discovered, the source is now in a position to utilize this information to send packets along the computed path. During the transmission of packets, the route may get broken as the nodes along the path may move away or go down. The process of detection of such route breakage and rebuilding is called *route maintenance.* The major advantage of *reactive* routing is that the precious bandwidth of such a wireless network is conserved, as limited amount of routing information is exchanged, and routes are maintained solely to those nodes to which the source needs to send data traffic. On-demand routing also obviates the need to disseminate routing information on a periodic basis or flooding the system each time a link failure is detected. The primary issue with on-demand routing is of course the added latency at the beginning of the transmission due to the route discovery phase. Furthermore, there is no guarantee that the route obtained is usable, as in the meanwhile some of the nodes in the route may have moved out of transmission range. Again, the problem becomes more pronounced when the mobility rate is high, as the route discovery mechanism is not able to keep up with the variations of the speed of the nodes. This basic idea of reactive protocols is used by protocols such as the Dynamic Source Routing [10] protocol, Temporally Ordered Routing Algorithm[7], and the Ad Hoc On-Demand Distance Vector [5] routing protocol.

3.2.1. Destination Sequenced Distance Vector Routing (DSDV)

The Destination-Sequenced Distance-Vector (DSDV) Routing Algorithm is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements to make it suitable for wireless schemes.

Every mobile node maintains a routing table that lists all available destinations, the number of hops to reach the destination and the sequence number assigned by the destination node. The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The nodes periodically transmit their routing tables to their immediate neighbors. A node also transmits its routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and event-driven.

The routing table updates can be sent in two ways: - a "full dump" or an incremental update. A full dump sends the full routing table to the neighbors and could span many packets whereas in an incremental update only those entries from the routing table are sent that has a metric change since the last update and it must fit in a packet. If there is space in the incremental update packet then those entries may be included whose sequence numbers have changed. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast-changing network, incremental packets can grow big so full dumps will be more frequent.

Because DSDV is dependant on periodic broadcasts it needs some time to converge before a route can be used. This convergence time can probably be considered negligible in a static wired network, where the topology is not changing so frequently. In an ad-hoc network on the other hand, when the topology is expected to be highly dynamic, this convergence time results in a lot of dropped packets before a invalid route is detected. The periodic broadcasts also add a large amount of overhead into the network.

3.2.2. Ad-Hoc On-Demand Disctance Vector Routing (AODV)

The Ad-Hoc On-Demand Distance Vector Routing [5] (AODV) protocol enables multihop routing between participating nodes wishing to form an ad-hoc network. The AODV protocol builds on the DSDV protocol previously described. The main difference being that AODV is reactive, unlike DSDV being proactive. Hence AODV would be considered an improvement on DSDV, as it minimizes the number of required broadcasts by creating routes on an on-demand basis and does not require maintenance of routes to destinations which are not actively used in communication.

The features of this protocol include freedom from loops, along with the fact that link breakages result in notifications being sent to the affected set of nodes. The use of destination sequence numbers ensures that a route used always remains fresh. The algorithm utilizes different messages to maintain and discover routes. When a node needs to determine the route to another node, a Route Request (RREQ) message is broadcasted to all its neighbors. The RREQ message traverses the network until it reaches the destination node or a node with a fresh route to the destination. Either of the nodes responds with a Route Reply (RREP) message back to the originating node by means of a unicast message.

The algorithm also utilizes a Hello message (a form of special RREP), where a node periodically broadcasts such a message to its immediate neighbors (those within transmission range). This enables each node to update its neighbors as to its continued presence. Also the neighbors use this message to continue to mark routes using this node as being valid. In the absence of hello messages from a particular node, the neighbor can assume that the node has either moved away or gone, identify the link as being broken, and notify the affected set of nodes by sending a link failure notification message to the set of nodes.

AODV keeps the track of following information:

- Destination IP address
- Destination sequence number
- Hop count: How many hops a packet has traversed.
- Next hop: Next to be forwarded host
- Lifetime: Duration for which this route is considered to be valid.
- Active neighbor list: Neighbors which use this route entry.
- Request buffer: A request should only be processed once.

Figure 3-1 Propagation of Route Request Packet RREQ

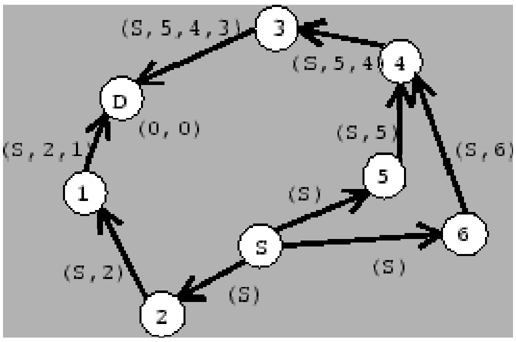
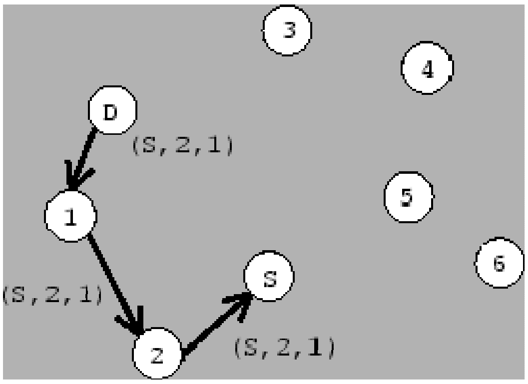


Figure 3-2 Path traversed by Route Reply Packet, RREP



Route Discovery Mechanism:

A node initiates a *path discovery* process (Figure 3-1 Propagation of Route Request Packet RREQ) to locate another node when the source desires to send a message to some destination node and does not have a route readily available. The broadcast may also be initiated when the route to the destination expires. The originating node expects a route reply RREP message in response to the RREQ packet. If the RREP message is not received within a given interval of time, the node reissues a RREQ packet.

The RREQ packet is forwarded by intermediate nodes until either the destination node is reached or an intermediate node has recent information for a path to the destination. While processing and forwarding the RREQ message, the intermediate nodes record the address of the neighbor from which the packet is received, thereby establishing a reverse path. These reverse paths are used when the RREP needs to make its way back to the originating node (Figure 3-2 Path traversed by Route Reply Packet, RREP).

Route Maintenance:

Route maintenance is accomplished when a node detects that a route to a neighbor is no longer valid, removes the route entry and sends a link failure message, informing those nodes that are actively using the route that the path is no longer valid. The active neighbor list is maintained to keep track of the neighbors using a particular link actively. This process is repeated by the other nodes in response.

The advantage with AODV is the greatly reduced number of routing messages in the network, as compared to the conventional routing protocols such as link state or distance vector. This is achieved by using the reactive approach.

3.2.3. Dynamic Source Routing (DSR)

Dynamic Source Routing Protocol [10,11] is a source-routed on-demand routing protocol. Every node maintains a route cache containing the source routes that it is aware of. The node updates the entries in the route cache if there is a better route, as it learns about new routes.

DSR requires that each packet keeps its route information, thus eliminating the need for every node in the network to do periodic route discovery advertisements. DSR performs a route discovery and takes required actions for maintaining that route. DSR depends on the support of the MAC layer (the MAC layer should inform the routing protocol about link failures). The two basic operations of DSR are route discovery and route maintenance.

Route Discovery:

The route discovery phase is used when a mobile node needs to send information to a particular destination node. The source node X first consults its internal source route cache to determine if it already has a route to the destination node. If an unexpired route exits, it will use that as the route to be used for all packets. However, if no such route exits, node X requests a route by broadcasting a Route Request (RREQ) packet. The RREQ packet contains information about the destination node, the source node and a unique identification number. Every node receiving the RREQ packet searches through its own route cache to see if it has a route to the destination. If no route is found, the intermediate node forwards the RREQ packet further, after adding its own address to the route record of the packet. To limit the number of route requests propagated, a node processes a route request packet only if it has not already seen the packet and its address is not present in the route record of the packet.

A route reply (RREP) is generated when either the destination node itself is reached, or an intermediate node containing route information of the destination. The selected return route may either be a list reversal of the route record within the packet, or using another existing route in the destination node's table. Thus the route may be considered unidirectional or bidirectional. DSR nodes stay awake and listen to everything that is of importance to their routing tables in promiscuous mode, so that route discovery may speed up.

Route Maintenance:

Route maintenance is the mechanism by which a sender detects if the network topology has changed and can no longer use the route to a particular destination. A failed link is determined either actively by monitoring acknowledgements or passively by running in promiscuous mode, overhearing that a packet is forwarded by a neighboring node.

When route maintenance detects a problem with a route in use, a route error packet is sent back to the source node. When this error packet is received, the error in the hop information is removed from its host's route cache, and all routes that that contain this hop are truncated at this point.

DSR uses the key advantage of source routing. Intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward. There is also no need for periodic routing advertisements messages, which leads to reduced network bandwidth utilization, particularly during period where little or no host movement taking place. Battery power is also conserved on the mobile hosts, both by not having to send the advertisements as well as receiving them, and a host could then go into a sleep mode if required.

This protocol has the advantage of learning routes by scanning for information on packets that it is handling. A route from A to C through B, implies that A has learnt the route to C, but also implicitly learns the route to B. The source route also means that B learns the route to A and C, and C learns the route to both A and B. This form of active learning is very good and reduces the overhead in the network.

However each packet carries the slight overhead containing the source route of the packet. This source route grows when the packet has to go through more hops to reach the destination. So the packets will be slightly bigger, because of the overhead.

Running the interfaces in promiscuous mode is a serious security threat. Since the address filtering on the interface is turned off, and all packets are scanned for information. A potential intruder could listen to all packets, and scan them for useful information such as security passwords or credit card numbers. The security aspect has to be dealt with by the application in this case by ensuring the data is encrypted prior to transmission. The routing protocols are prime targets for impersonation attacks and must therefore also be encrypted.

DSR also has the support for unidirectional links by the use of piggybacking the source route a new request. This can increase the performance in scenarios where we have a lot of unidirectional links. However, the MAC layer protocol must also support this.

3.2.4. Temporally-Ordered Routing Algorithm (TORA)

Temporally Ordered Routing Algorithm [7,8] is a distributed routing protocol. The basic underlying algorithm is one in a family referred to as link reversal algorithms. TORA is designed to minimize reaction to topological changes. A key concept in its design is that control messages are typically localized to a very small set of nodes. It guarantees that all routes are loop-free and typically provides multiple routes for any source destination pair. It provides only the routing mechanism and depends on the Internet MANET Encapsulation Protocol (IMEP) [9] for other underlying functions.

TORA has three basic functions: Route creation, route maintenance, route deletion. Each node keeps the following values :

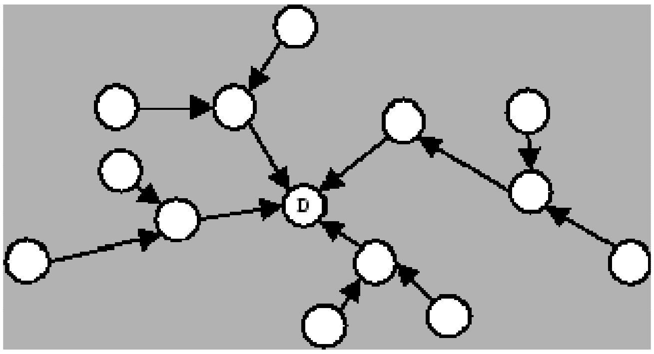
- The old unique ID of the node that defined the new reference level.

- A clock tag set to the time of the link failure, where nodes should have synchronized clocks with an external time source such as the Global Positioning System (GPS).
- A reflection indicator bit
- A propagation ordering parameter, height.
- The current unique ID of the node itself.

The creation of routes basically assigns different directions to links in an undirected network or portion of the network, building a directed acyclic graph (DAG) rooted at the destination. The first three elements collectively represent the reference level. ``A new reference level is defined each time a node loses its last downstream link due to a link failure". Last two values define an offset with respect to the reference level, which were the first three values**.**

TORA associates a height with each node in the network. All messages in the network travel downstream, from a node with a higher height to a node with a lower height. Routes are created using QUERY (QRY) and UPDATE (UPD) packets. When a node requires a route to a destination, it sets the height of the destination to 0 and all other node's height set to undefined, NULL. It then broadcasts a QUERY packet with the id of the destination node (Figure 3-3 Propagation of Route Request Packet, RREQ).

Figure 3-3 Propagation of Route Request Packet, RREQ



The QRY packet propagates through the network until it reaches a node with a route to the destination or the destination itself. Any node receiving this QRY packet, responds with UPD message containing its ID only if that node's height is a non-NULL value. Any node receiving an UPD packet sets its height to one more than that of the node that generated the UPD. A node with higher height is considered upstream and a node with lower height downstream. UPD packet floods until it completes the route information. This way a Directed Acyclic Graph (DAG) is constructed from source to destination. The propagation may end when an intermediate route knows the rest of the route. This results in multiple routes being known for a given destination.

Maintaining routes refers to reacting to topological changes in the network in a manner such that routes to the destination are re-established within a finite time. Upon detection of a network partition, all links in the portion of the network that has become partitioned from the destination are marked as undirected to remove invalid routes. The removal of routes is done by Clear (CLR) messages.

The protocols underlying link reversal algorithm will react to link changes through a simple localized single pass of the distributed algorithm. This prevents CLR packets from traveling too far into the network. The comparison studies initiated by the CMU Monarch project have shown that the overheads in TORA are quiet large due to the use of IMEP. The graph is rooted at the destination, which has the lowest height. However, the source initiating the QRY does not necessarily have the highest height. This leads to situations, where multiple routes may exist between the destination and source, but only a single route may be found. The reason for this is that the height is initially based on the distance in number of hops from the destination.

3.2.5. Location Aided Routing (LAR)

Location Aided Routing (LAR) [19] protocol is an on-demand based protocol in which routes to destinations are determined only when explicitly needed to route packets. It uses location information to limit the route query flooding area. Every mobile host node is assumed to know its location and the global time, which can be provided by a Global Positioning System (GPS).

LAR defines the concept of *expected zone* and *request zone*. In the situation that a node S does not have a route the destination node D, the node S initiates a route discovery process. If S has knowledge of the location L of D at a time t_0 and the current time being t_1, then the *expected zone* of the node D is the region that node S expects to contain node D at time t_1. For instance, if node S is also aware of the speed v of node D, then S may assume that the expected zone is the circular region of radius $v(t_1 - t_0)$ centered at the location L.

The *request zone* is used to limit the route query flooding. A node forwards the route query message only if it belongs to the request zone. The request zone is the smallest rectangular region, which includes the expected zone of D and current location of the source S.

When a node wants to send a message to node D, it broadcasts a *route query* message, which is only forwarded by nodes in the request zone. After node D finally receives the request message, it sends back a route reply message to the source S, using the reverse path which is recorded in the head of the route query packet. The route from S to D is established when the source node S receives the route reply packet. The authors [19] of LAR propose two methods by which the source and destination nodes may determine the request and forwarding zone for a route request packet.

Method I :

The first method used a rectangular request zone. In this method a neighbor of S determines if it is within the forwarding zone by using the location of the source S and the expected zone of the destination D. The expected zone is a circular area determined by the most recent location information on D, (X_D, Y_D), the time of this location information (t_0), the average velocity of D (V_{avg}), and the current time (t_1). This information creates a circle with Radius R = V_{avg})x (t_1 - t_0), centered at (X_D,Y_D). The request zone is the rectangular area covered by the source node S in one corner and the circle containing D in the other corner. The source node S includes the four corners of the rectangle with the route discovery messages. When a node receives the route request message, it checks to see if it is within the specified rectangle, and discards the packet if it is not. Node D responds with the route reply packet when it receives the route request packet from S. However, in this case, D also includes its current location and current time within the route reply packet. When node S receives this route reply message, it records the relevant information as conveyed by D, and uses this information for future route request and route query zones.

Method II :

Unlike method I, where the node S explicitly defines the request zone in the route request message, in method II an intermediate mobile node determines it is within the request zone. In order to facilitate this process, the source node S includes its distance from a previously known location of D, as well as the location of D that was used to calculate this distance. When an intermediate node receives this request message, it calculates its distance from the last know location of D as recorded within the packet. If this intermediate node is closer or not much farther from D than S, then it forwards the route request message.

Both the LAR methods of LAR provide the facility of increasing or decreasing the request zone, via the inclusion of an adjustment factor. If the route reply packet is not received within the route request timeout period, then a second route request packet is flooded within the entire ad hoc network. If the route reply is still not received, the destination node D is considered unreachable.

3.3. Summary Of Routing Protocols

As can be seen from the above discussion, several routing protocols exist within the ad-hoc mobile routing domain.

DSDV and ZRP were the only proactive protocol discussed. AODV is a reactive, on-demand version of DSDV. Authors of AODV, who were also authors of DSDV, added multicast capability to AODV. Reactive approach of AODV is similar to DSR's. They both have route discovery mode which uses messaging to find new routes. DSR uses source routing; the route is contained within each packet. Thus, DSR learns more routes than AODV. DSR supports unidirectional links due to its vast knowledge on the topology. TORA runs on top of IMEP, and suffers for its internal instability and IMEP's frequent HELLO messages generating too much control overhead in the network.

The main protocol studied as part of this thesis work is the DREAM protocol, which uses the physical location information of a node as a basis for the routing protocol structure. LAR was the only routing protocol discussed which lies within this domain of utilizing location information.

4. Improved Distance Routing Effect Algorithm for Mobility

As discussed in the previous sections, Mobile Ad Hoc Network protocols can be broadly classified as either proactive or reactive. In both cases, each node builds a routing table, similar to a static network, representing a topology of the network and sequence of next hops that would enable information to traverse the network to the desired destination. In the case of proactive protocols, the sequence of nodes is not explicit, rather a next hop reference to be used for a particular destination. Reactive protocols resort to a route discovery mechanism, which results in a sequence of nodes to be explicitly followed in order to reach a particular destination. Regardless of the protocol class, these determined routes become defunct when a node moves out of its position and is no longer in the routing path to a destination. Given the mobility of the nodes, an intrinsic nature of an ad hoc mobile environment, these scenarios become highly probable, and nodes have to resort to repopulating their routing tables. Increased mobility result in rendering these protocols more inefficient, with constant control and route discovery packets flooding the network, increased overheads and lost transmission of packets.

The Distance Routing Effect Algorithm for Mobility (DREAM) protocol is essentially a location based protocol. This implies that each node (within the network) contains the *location information* for every other node within the network, as an entry against each node. This location information may be obtained from GPS (Global Positioning System) [1], which enables a mobile node to know its physical location. In real life scenarios however, the position information provided by GPS has a margin of error, which is calculated as the difference between the GPS calculated coordinates and the real coordinated. However, within this discussion, it is assumed that all mobile nodes know their current location precisely.

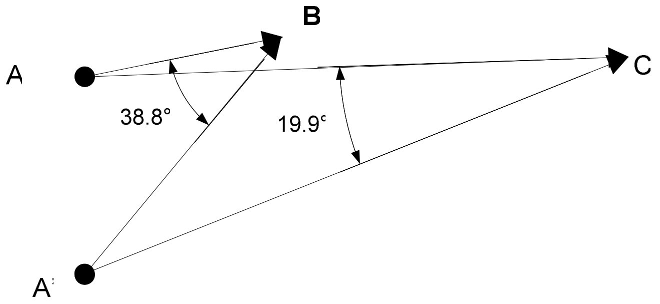
DREAM may be considered part proactive and part reactive in nature. The nodes within a DREAM environment have a means of disseminating and collectively updating the location table entries for each other, behaving as a proactive protocol. When an information packet needs to be transported form node A to node B, node A looks up the location of B from within its tables and forwards the packet to nodes “in the direction” of B, as the next hop node. These intermediate nodes in turn perform a lookup and forward the packet “in the direction” of B. This results in the protocol mechanism reflecting a reactive nature.

As a proactive protocol, each DREAM node disseminates and updates other nodes within the network with its current location information. The frequency of generation and distribution of information within the *location packets* is determined by two phenomena addressed by the DREAM protocol, the *Distance Effect* and *Mobility Effect*.

Distance Effect :

The distance effect may be conceptually compared to the *parallax* phenomena. The parallax phenomena maybe summarized as the “apparent change in position of distant objects, due to the actual change in position of the observer”. In practicality, this results in the fact that further the distance between two points, the slower they seem to move with respect to each other.

Figure 4-1 Distance Effect



As can be seen from the figure above, Node A moves from position A to position A'. There are two nodes B and C, who are stationary with respect to A, where node B is closer to node A than node C. As is evident from the illustration, node A has moved a greater angular distance with respect to node B (38.8 deg) as compared to the farther node C (19.9 deg). This results in the fact that, for the same distance traversed and same speed, node A "appears" to be moving more slowly from C's perspective, as compared from B's perspective.

With the above information in mind, it can be realized that nodes that are farther apart, need to update each other with their location information less frequently as compared to nodes which are closer. Therefore, when a node distributes a location information packet, it can now specify an *age* for such a control packet. The age may be in terms of distance, i.e., the control packet is not propagated into network beyond a certain distance, or in terms of time, i.e., the packet is not propagated within the network after a certain timeout period.

Mobility Effect :

The mobility effect addresses the question of how often a node should generate and disseminate location information packets. A node essentially updates other nodes within the network with its location information. Ideally, every time the location of the node changes, it should generate and distribute a location packet. However, as an optimum method, each node generates a location update packet at a periodic interval. This periodic interval is governed as a function of the mobility rate of the node itself i.e. the faster a node travels, the more frequently it distributes location update messages. This effectively allows each node to optimize the route dissemination frequency, thus transmitting route information only when needed, without sacrificing the route accuracy.

While addressing the distance and mobility effects within the protocol behavior, the DREAM protocol effectively reduces the amount of control packet overhead which can become quiet excessive in proactive protocols. Similarly, it also overcomes the initial delays of the route discovery phase as experienced by reactive protocols.

4.1. Model For DREAM

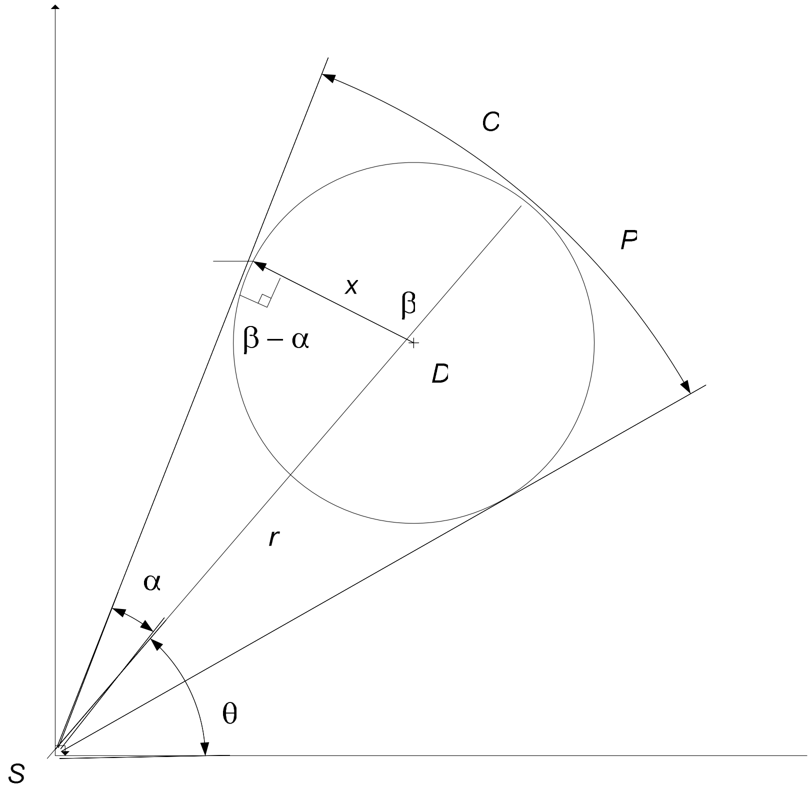
The model for DREAM defines a method of determining a probabilistic guarantee of finding a destination node in a given direction. Prior to this, the location information dissemination (discussed in further detail is section 5.2.1) mechanism ensures that each node has relatively fresh location information tables. When a source node S wants to send information packets to a destination node D, it retrieves the location information of D stored within it location tables. Using this location information as a reference, S determines those nodes amongst its neighbors who are "in the direction" of D, and forwards the message packet to them. On receipt of this information packet, the intermediate neighboring nodes in turn perform a lookup into their location tables to retrieve the location entry for the destination D. The intermediate nodes in turn forward the message packet to those nodes, amongst it neighbors who are in the direction of D, similar to S. This process continues until the destination D is eventually reached. This method of selecting neighbors within a given direction range, results in a certain probabilistic guarantee of p, $0 < p < 1$, that destination B will be reached.

Each location update packet, and therefore the associated location entry for a given node represented by a location packet, contains the location, the time of sending the update message and the velocity of an individual node. Given the information of D within the location table of S as entry LT(D) = t_0, as detailed in figure below, it is now easily possible to calculate the distance D_r (from node S to D) and the angle D_θ.

When node S needs to send information packets to the destination node D at some later time t_1, where $t_1 > t_0$, S needs to choose its neighbors to which it can forward the packet. Neighbors A are chosen by S such that, A_θ i.e. the direction vector of A, lies within the range $[\theta+\alpha, \theta-\alpha]$. The value of α must be chosen in such a manner that the probability of finding the destination D is the sector *C* is maximized. The sector *C* is centered about the line segment connecting *S* and *D* and defined by $[\theta+\alpha, \theta-\alpha]$.

Within the time interval t_1 - t_0, the maximum distance node D can travel at velocity *v* can be calculated as $x = v(t_1 - t_0)$. If a circle *P* is drawn with the radius as *x*, centered on the position of node D at time t_0, the circle borders the confines of the new position of node D at time t_1. This implies that node D cannot be anywhere outside of circle *P* after the time interval t_1 - t_0. Given that the direction of travel of node D is not specifically known, D can move in any direction β uniformly chosen between o and $2\tilde{\pi}$ Therefore the optimum or minimum value of α needs to be chosen such that, the maximum distance *x* that D can travel within t_1 - t_0 at velocity *v* is within the sector *C*. The value of α needs to be at a minimum essentially because next hop neighbors are chosen such that they are within the sector determined by α. A smaller value of α results in a smaller sector area, resulting in fewer number of next hop nodes bring present within the sector. This further implies that fewer next hop nodes are transmitted the message to forward to the destination. This effectively results in a lower overall network bandwidth and resource utilization i.e. improved efficiency.

Figure 4-2 Graphical Description of DREAM



The value of α is clearly dependant on the speed v of D. Therefore, if either the average or maximum speed of the node D is known, then it is straightforward to calculate the value of α which guarantees that D will lie within the direction $[\theta+\alpha, \theta-\alpha]$,

$$\alpha = \arcsin \frac{v(t_1 - t_0)}{r}$$

It is evident, that if the distance x traveled by D is greater than the distance r i.e. the distance between S and D, then D could be anywhere around S. In this case, α would = π.

If v is not known and only a probability density function of $f(v)$ is available, we need to find an ά such that the probability of finding D in the direction range [ɵ+ά, ɵ-ά] is greater than or equal to p, for a given p, $0 < p \leq 1$. More formally, we need to determine ά such that,

$$P\,(\,x \leq (t_1 - t_0)v\,) \quad \geq \quad p$$

In this case, since geometrically,

$$\frac{x}{\sin \alpha} = \frac{r}{\sin(\beta - \alpha)}$$

and, since

$\beta - \alpha = \pi/2$, the above equation become $x = r \sin \alpha$

we need to find ά so that,

$$P\,(\,x \leq (t_1 - t_0)v) \;=\; P\,(\,r \sin \acute{\alpha} \leq (t_1 - t_0)v\,)$$

$$= P\left(\mathrm{v} \geq \frac{r \sin \acute{\alpha}}{(\,t_1 - t_0\,)}\right)$$

$$= \int_{\frac{r\sin\alpha}{t1-to}}^{\infty} f(v)\,dv$$

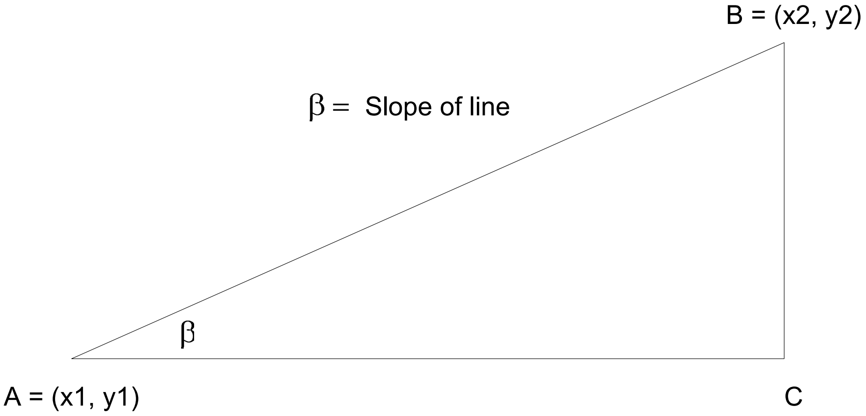
4.2. Model For Improved DREAM

The previous section defines the basic mechanics of the working of the DREAM protocol. The base protocol mechanics discusses a means by which the destination nodes current location is calculated within a circle centered on the last known location of the node (as updated within the location tables from location information packets received from the destination node). The model for improved dream includes the direction of travel of the destination node, in addition to the location, the time of sending the update message and the velocity of an individual node.

The location table entry within each node now contains the speed, location, time and direction of travel for every node within the network. When a node needs to send packets to a particular destination node, it calculates the correct location of the destination with the above information. The direction of the travel of the destination now allows estimation of the current location of a node with greater accuracy than the original model of Dream.

When a source node S wants to send information packets to a destination node D, it retrieves the location information of D stored within it location tables. This location information of the destination node is adjusted, given the direction of travel of the destination node.

Figure 4-3 Calculating Direction of Travel

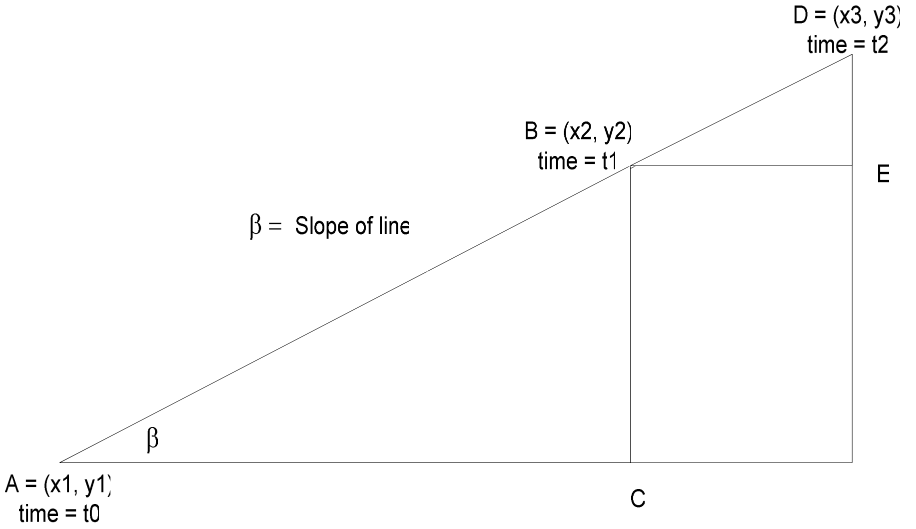


A node disseminating a location packet calculates it direction of travel by keeping a record of its location over successive intervals of time. If at time t_0 the location of a node is (x_1, y_1) and at time t_1 (when it has to send a location update packet) the location is (x_2, y_2), the direction of travel can be represented by the slope of the line joining the two location coordinates (Figure 4-3 Calculating Direction of Travel). Therefore the direction of travel is calculated as;

$$\beta = \frac{y2 - y1}{x2 - x1}$$

When node S needs to send information packets to the destination node D at some later time t_2, where $t_2 > t_1$, S needs to choose its neighbors to which it can forward the packet. Neighbors A are chosen by S such that, A_θ i.e. the direction vector of A, lies within the range [$\theta+\alpha$, $\theta-\alpha$], as shown in the previous figure. However, before calculating the neighboring nodes, node S first adjusts the location information of D, by calculating the most accurate position coordinates of D (Figure 4-4 Adjustment to Determine New Location Coordinates).

Figure 4-4 Adjustment to Determine New Location Coordinates



The distance between position B at time t_1 and position D at time t_2 can be calculated using the $x' = v(t_2 - t_1)$. Similarly,

$$\cos(\beta) = \frac{BE}{BD}$$

Where BD = $x' = v(t_2 - t_1)$ and

$$BE = v(t2 - t1)\cos(\beta)$$

Therefore,

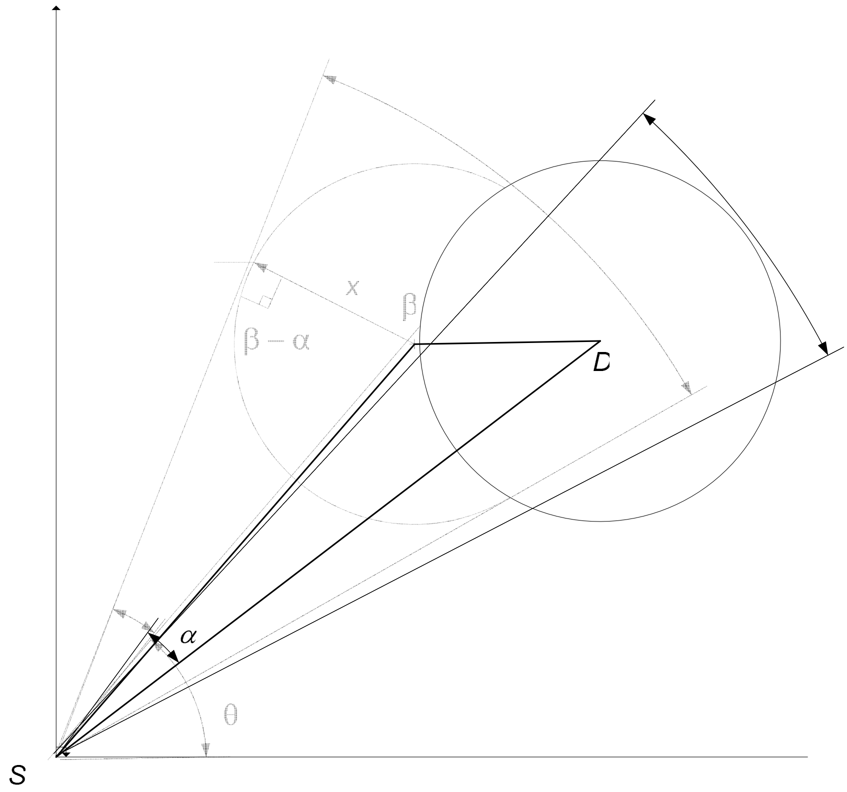
$$x_3 = x_2 + v(t_2 - t_1)\cos(\square$$

Similarly,

$$y_3 = y_2 + v(t_2 - t_1)\cos(\beta)$$

With this new location information for node D, node S can now determine the neighbors for node D as per the original model for the Dream protocol. We can now modify the diagram as per figure, to be a more accurate means of determining the location of D, (Figure 4-5 Representation of Improved DREAM).

Figure 4-5 Representation of Improved DREAM



Where the new location is given by D. The calculation for the angle α and β are carried as normal, according to the original protocol calculation means. However, given that we have a most accurate location of destination D, the angle α can now be made smaller. This results in a smaller sector of neighbors chosen to forward the packet. Fewer neighbors implies fewer packets are introduced into the network resulting in a reduced overall transmission overhead within the network.

As discussed above, the value of alpha can now be reduced within the algorithm of the packets. We can now determine the effect of this reduced value of α on the probability of a packet being delivered to the destination node D. From the previous discussion, the probability of finding the node D was given as;

$$P\ (\ x \le (t_1 - t_0)v) \ = \int_{\frac{r\sin\alpha}{t1-to}}^{\infty} f(v)dv$$

However, the new value of $\alpha^{\backepsilon}$, where $\alpha^{\backepsilon} < \alpha$, implies that

$$\alpha^{\backepsilon} < \alpha$$

hence, $\sin(\alpha^{\backepsilon}) < \sin(\alpha)$

and $r\sin(\alpha^{\backepsilon}) < r\sin(\alpha)$

$$\int_{\frac{r\sin\alpha}{t1-to}}^{\infty} f(v)dv \quad < \quad \int_{\frac{r\sin\alpha'}{t1-to}}^{\infty} f(v)dv$$

Because the left side probability function is now integrated over a larger interval, given that the lower integral has a smaller value, the probability of finding the destination node D with the new location information and smaller alpha, is higher.

Therefore:

$$\boldsymbol{P'(\ x \le (t_1 - t_0)v) \quad > \quad P\ (\ x \le (t_1 - t_0)v)}$$

5. Implementation Environment

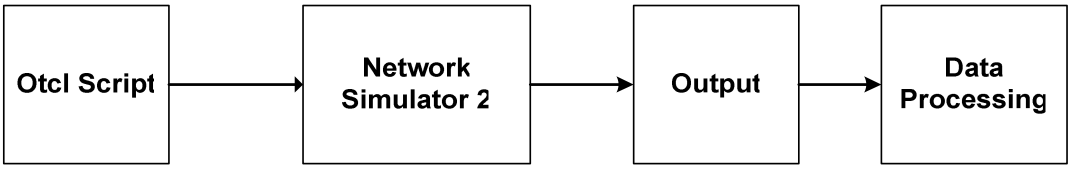
The following chapter discusses the implementation details of the DREAM and Improved DREAM protocols.

5.1. Network Simulator

The network simulator is a discrete event simulator and is the result of on-going research that is administered by researchers at Berkley. It provides considerable support for simulation of TCP, routing and multicast protocols.

The simulator is written in C++, accompanying an OTCL script language based on Tcl/Tk. The researcher defines the network components such as nodes, links, protocols and traffic using the OTCL script i.e NS uses OTCL as the interface to the user. This script is then used with ns, the simulator, to conduct the desired simulation, and as a result outputs traces at different selective layers. The data within the trace output files is then used to calculate the various metrics such as delays, throughput, overheads etc. An overview of the simulation is shown in the figure below:

Figure 5-1 NS-2 Simulation Overview



The current version of the Network Simulator does not contain support for a mobile wireless environment. The Network Simulator is only intended for wired networks with stationary links.

5.1.1. Mobility Extension

However, there are two mobility extensions for ns, which are :

- Wireless mobility Extension released by the CMY Monarch Project
- Mobility support, mobile IP and wireless channel support developed by C.Perkins at Sun Microsystems

The ns group at Berkley intends to incorporate both extensions to the official release of ns2 in the future. For the purpose of this thesis, the CMU Monarch extension for mobility support within ns-2 has been chosen, primarily because this extension is specifically designed for Mobile Ad-Hoc Networks. The version of the extension that has been used within the thesis provides the following features:

Mobile Node:

This is the basic object with added functionalities, which can make movements as well as receive and transmit on a channel. Mobility features include node movement, periodic position updates, maintenance of topology boundary etc. These aspects and behavior of the node are implemented in C++. Plumbing of network components like classifiers, dmux, LL, MAC, channel within MobileNode have been implemented in OTCL.

Each mobile node is attached to a routing agent for calculating routes to other nodes in the network. Packets sent from the application are received by the routing agent. The agent then determines a routing path for the packet and stamps it. It sends the packet down to the link layer. The link layer uses ARP to determine the hardware addresses of neighboring nodes and maps IP addresses to their correct interfaces. The packet is then sent to the interface queue, and stays there until a signal from MAC is received. It leaves the IFQ and waits for MAC to send it when the channel is available. The packet is copied to all interfaces at the time at which the first bit of the packet would begin arriving at the interface in a real physical system. Each network interface stamps the packet with its own properties, and invokes the propagation model. Note that the propagation model is invoked at the received part. The propagation model uses transmit and receive stamps to determine the power that the interface will receive the packet. The receiving network interface is left to decide whether the packet is received successfully or not. If successful, the packet is passed to MAC layer. If MAC layer receives this packet as error-free and collision-free, it passes the packet to node's entry point. The packet then reaches a demultiplexer, which decides whether the packet should be forwarded again or if it has reached its destination node. If the arrival point is the destination node, the packet is sent to the demultiplexer, which then decides the application to which it should be delivered. If the packet is forwarded, this operation is repeated [20].

MAC 802.11

An implementation of the IEEE 802.11 Media Access Control (MAC) [21] protocol was was included in the extension. The MAC layer handles collision detection, fragmentation, acknowledgements, as well as to detect transmission errors. 802.11 is a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol and avoid collisions by checking the channel before using it. If the channel is free, it can start sending, if it is not, it waits random amount of time before re-sending. For each retry, an exponential backoff algorithm is used. In a wireless medium it cannot be assumed that all stations hear each other. If a station seizes the medium as available, it may not necessarily be so. This problem is known as hidden terminal problem and to overcome these problems, the collision avoidance mechanism and positive acknowledgement scheme is used together. Positive acknowledgement requires peers to retransmit data and acknowledge to each other until both are successful.

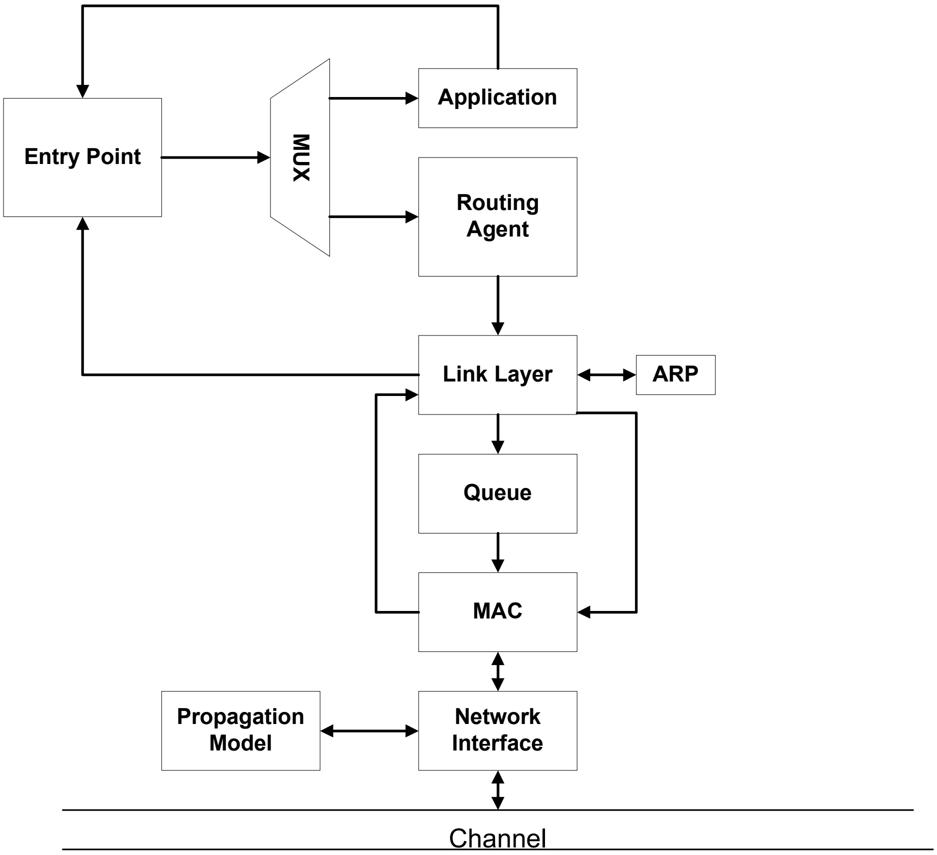
In NS, ARP or the Address Resolution Protocol is also [22] implemented. It translates IP addresses to hardware MAC address before the packets sent down to

MAC. The antenna gain and receiver sensibility parameters are available in NS. There are different antennas available for simulations.

Shared Media

The channel implementation is based on a shared media model (Figure 5-3 Simulation Overview). This means that all mobile nodes have one or more network interfaces that are connected to a channel. A channel represents a particular radio frequency with a particular modulation and coding scheme. Channels are orthogonal, that is, packets sent on one channel do not interfere with transmission and reception in adjacent or any other channels. A packet is received if the transmission range is within the radio propagation model calculation, and if bit errors allow it.

Figure 5-2 A Mobile Node



5.1.2. Simulation Overview

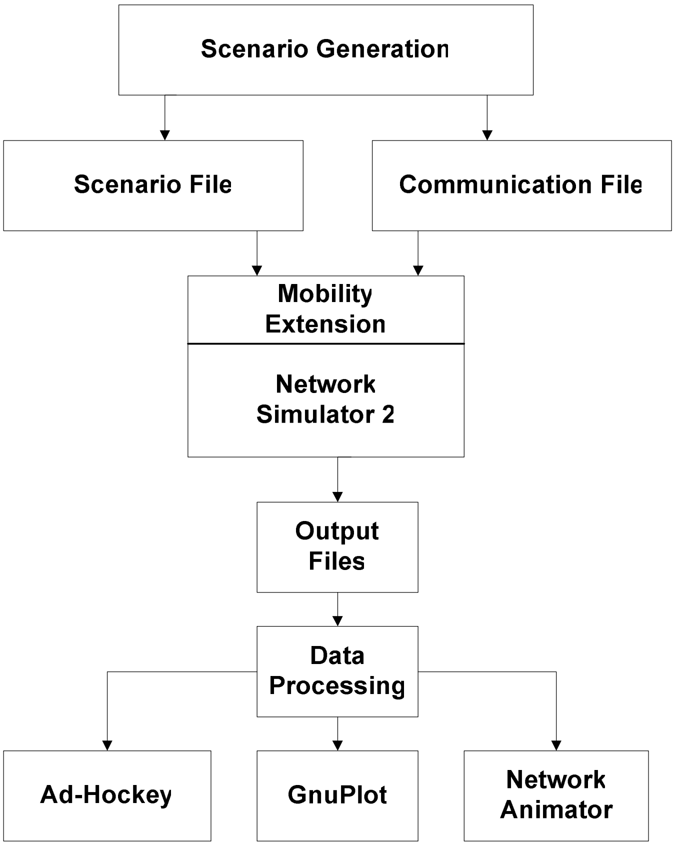
A simulation run in ns typically, is depicted in the figure below. Any simulation consists basically of two input files to ns:

- Scenario File : the scenario file describes the basic movement pattern for each node within the network.
- Connection File : the connection file describes the traffic model and connections between nodes within the network.

The above files are generated in a completely randomized format from a script which generates the connection and movement patterns.

These files are then used for the simulation as a feed for the behavior of the nodes. The simulation output results in the generation of a trace file. Prior to the simulation, the modules and files specify the parameters which are that are of interest and are to traced within the trace files. The data and parameter values are then analyzed post simulation and data that we need to measure is extracted. The trace file data can be used in plots with for instance the GnuPlot. It can also be used to visualize the simulation run with either the Network Animator or the Ad-Hockey simulator [32].

Figure 5-3 Simulation Overview



5.2. Design

5.2.1. Location Information Disemmination

The DREAM protocol as discussed in the Chapter 4 , is a proactive protocol, i.e. each node performs route update procedures, whereby they disseminate routing information depending on the methodology of the underlying table driven protocol. DREAM being a location based proactive protocol, each node requires to be aware of its own location (represented by coordinates) with respect to some predefined positioning system (e.g. the Global Positioning System as in [1].Each node broadcasts it location and other information into the network constantly with the intention that other nodes within the network can obtain location information about all other nodes. Each node maintains a *Location Table*, with an entry for every node within the network. A typical location table entry for each node includes:

Table 5-1 Node Entry

Name	Description
Time	Time of generation of the location packet
X coordinate	The X coordinate of the node at the time of sending the location packet
Y Coordinate	The Y coordinate of the node at the time of sending the location packet
Speed	The speed of the node while sending the location packet
Direction	The direction of travel of the node while sending the location packet (applicable to iDream)

Since the DREAM protocol is based on a proactive approach, with constant route update and maintenance, the overhead introduces by the location update packets can be very expensive. A model approach is required to reduce the expense in the dissemination of location information. This is achieved by incorporating the two concepts, introduced in Chapter 5, of the Distance and Mobility effects.

The Distance Effect phenomenon suggests that the further apart two nodes are, the less often their location table entries need to be updated. Intuitively, when two nodes are moving at the same speed, a closer node appears to be changing position (or moving) more rapidly than a node moving at the same speed but further away.

To realize the distance effect, each *control packet* (containing the location information of a particular node in the form of location coordinates) is assigned a lifetime for the duration of which it is propagated or disseminated within the network. The lifetime is based either on the geographical distance that a packet has traveled from the originating source or as a time duration since the time of origination from the source node. In this particular implementation, the lifetime is associated with the geographical distance of the control packet from the sender. Also, the distance effect implies that closer nodes need to be updated more frequently than nodes further away from the source. Therefore, a majority of the packets have "short" life time (they travel a short distance), i.e., the control packets with a short lifetime are generated at a higher frequency, and "die" after traversing the network a short distance from their sender.

The source node also generates control packets with a "long" life time which travel farther through the network, enabling the most distant nodes to update their location tables with the node location entry. The generation of long location packets is based on a timer and interspersed between short location packets.

As mentioned earlier, each location control packet is forwarded based on the life time of the packet. In the current implementation, the lifetime of the packet is a reference to the geographical distance of the packet from its originating source. When a control packet is received by an intermediate node A, the node determines how far the packet has traveled by calculating the distance between itself and the sender of the packet. If the distance is greater than the lifetime associated with the packet, then packet is then no longer forwarded.

The Mobility Effect phenomenon suggests that a node needs to generate location update messages in order to update the rest of the nodes within the network as to its location based on the speed of travel of the node. In other words, the frequency with which a node broadcasts control packets is a function of the node’s mobility, i.e., the more mobile a node (i.e., the greater the speed of a node) the more often it must disseminate its location information. The fact that most of the packets are short lived packets clearly implies the fact that the nodes closest to the originating node are those in need of its location as it changes its location most dramatically compared to the closer nodes. Nodes farther away need to be updated less often.

As a result of the above mechanisms, the further away a particular destination and the slower the rate of movement of the updating node, the less often a copy of the control packet will be sent. This procedure results in effectively minimizing the total number of control packets in the network, while maintaining the same probability of error per route. The dissemination method also reflects the distance effect and this maintains the same probability of routing accuracy while distributing control packets proportionately to distance and rate of movement.

The dissemination method described above has the following properties:

- When no movement occurs no bandwidth is wasted on control packets since control packets will be initiated only nodes on the move.
- The frequency of update location information can be optimally gauged since the decision of the update frequency inherently lies with the node itself and independent of the protocol or network environment.
- The total number of control packets can be minimized since the aging of control packets captures the relative distance between the moving node and the location table updating node.
- The previous point also implies that there is consequent conservation of energy on the node, a sparse commodity on mobile nodes.
- The aging and removal of control packets prevents the formation of routing loops in the network. Hence, the protocol is essentially loop free.

6. Simulation Study

The overall goal of the simulation study is to study the performance of the protocol, with respect to the protocols ability to react to network and topology changes, and being able to successfully deliver data packets to their destinations. In order to measure this ability, the basic methodology is to apply to the simulated network a variety of workloads, in effect testing with each data packet originated by some sender whether he routing protocol can at that time route to the destination for that packet. The workload and network conditions are specific to the movement and scenario files as generated by the NS network simulator simulation platform and not reflective of a real-life topology or load condition.

The simulation environments created for this particular thesis are comparable to the ones used in [24]. The paper compares the environment as created in [25,26] as a way of validating the particular choices of the environment. The comparison and validation conducted in [24] is summarized in the following discussion.

The simulation area chosen is rectangular. Although a square simulation area allows nodes to move more freely, it results in fewer numbers of hops between senders and receivers as compared to a rectanglar simulation area of the same size.

The movement model selected for the mobility scenario is the random Way-point model. With this mobility model, there is a complex relationship between node speed and pause time. For example, a scenario with fast MNs and long pause times actually produces a more stable network than a scenario with slower Ns and shorter pause times. Genrally, longpause times (i.e., over 20 seconds) produce a stable network(i.e., fewlink changes per MN) even at high speeds [27]. Thus, for the purpose of this simulation environment it has been chosen to keep the pause times short and to vary speed along the x-axis in all of the simulations.

Within the simulations, the speed of an mobile node, between the MN's current location and its next destination is chosen from a uniform distribution between *avg* ± 10% meters per second(m/s), where *avg* is set to 0, 1, 5, 10, 15, and 20. For example, when the speed is set to 20 m/s, all nodes have speeds between 18 and 22 m/s. The narrow range of speeds prevents the creation of a stable "backbone" consisting of a few slowly moving mobile nodes.

The chosen communication model is similar to the communication model used in [25] and [26]. Specifically, there are 20 CBR (constant bit rate) sources sending 64 byte packets at a rate of 4 packets per second to 20 receivers. One difference between the communication models is that [26] randomly spreads the traffic among all mobile nodes, while [25] and the current simulations create peer-to-peer traffic patterns. Peer-to-peer traffic stresses the network protocols since traffic is concentrated in specific areas of the network while the risk of unnecessary contention in the transmission of packets is avoided in this particular simulation.

Table 6-1 Simulation Parameters

	[1]	[2]	Herein
Simulator	NS2	NS2	NS2
Simulation Time	900s	250s	250s
Simulation Area	1500x300m	1000x1000m	300x600m
Number of MNs	50	50	50
Transmission Range	250m	250m	250m

Movement Model	Random Waypoint	Random Waypoint	Random Waypoint
Maximum Speed	1 and 20m/s	0-20m/s	0-22m/s
Pause Time	0,30,60,120, 300, 600,900	1s	10s +/- 10%
CBR Sources	10,20 or 30	15	20
Data Payload	64 bytes	64 bytes	64 bytes
Packet Rate	4 packet/s	4 packets/s	4 packets/s
Traffic Pattern	Peer-to-peer	Random	Peer-to-peer

The following describes each of the simulation parameters and their significance in the environment:

Simulator : Describes the type of simulator used

Simulation Time: Details the time length for a simulation for a fixed set of parameters

Number of MNs : Describes the total number of mobile nodes present in the simulation environment

Transmission Range: The distance upto which a mobile node can transmit. A mobile node within this distance is considered an immediate and reachable neighbour

Movement Model : The type of model used to generate and represent the movement of each node

Maximum Speed : The maximum speed that a node may reach within the simulation

Pause Time : The time in between direction changes for a node

CBR Sources : The number of data sources sending Constant Bit Rate traffic

Data Payload : The size of each data packet

6.1. Simulation Results

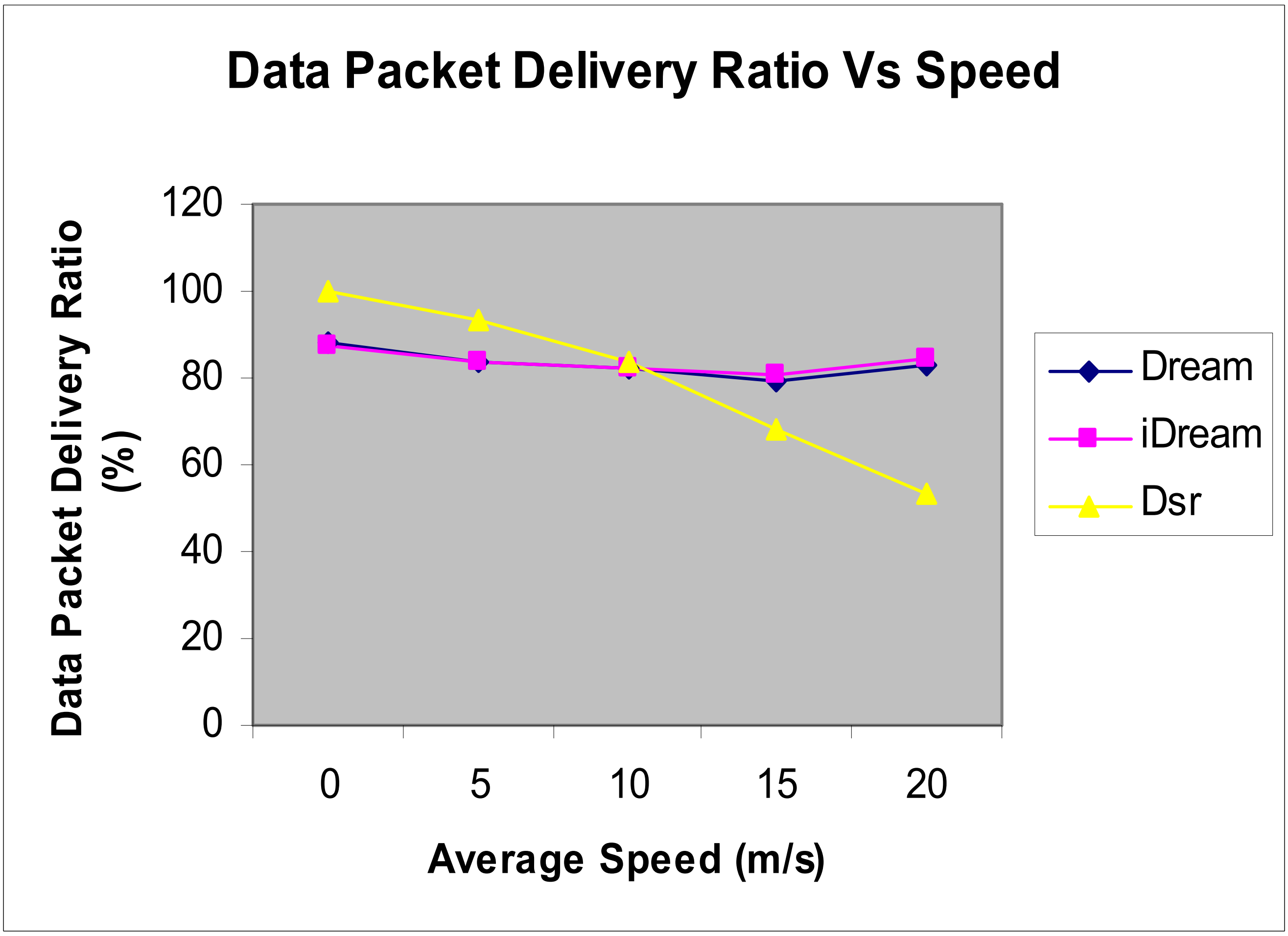
In the simulation study and comparison study conducted herein, between DSR, DREAM and i-Dream, the following performance metrics have been considered:

- Protocol Overhead
- Network-wide data load
- End-to-end delay, and
- Data packet delivery ratio

6.1.1. Packet Delivery Ratio

The data packet delivery ratio is the ratio of the number of data packets delivered to the destination nodes divided by the number of data packets transmitted by the source nodes. The simulation for this particular scenario has been conducted under different speeds and the packet delivery ratio determined under each speed variation. Figure 6-1 below illustrates the data packet delivery ratio versus speed.

Figure 6-1 Data Packet Delivery Ration vs. Speed



- From the above figure it can bee seen that at zero speed, the data packet delivery ratio for DSR is 100%, whereas the data packet delivery ratio's of Dream and iDream are approximately 88%.

- 100% delivery ratio is not reached by Dream and iDream due to contention and congestion within the network caused by the flooding nature of the protocol.

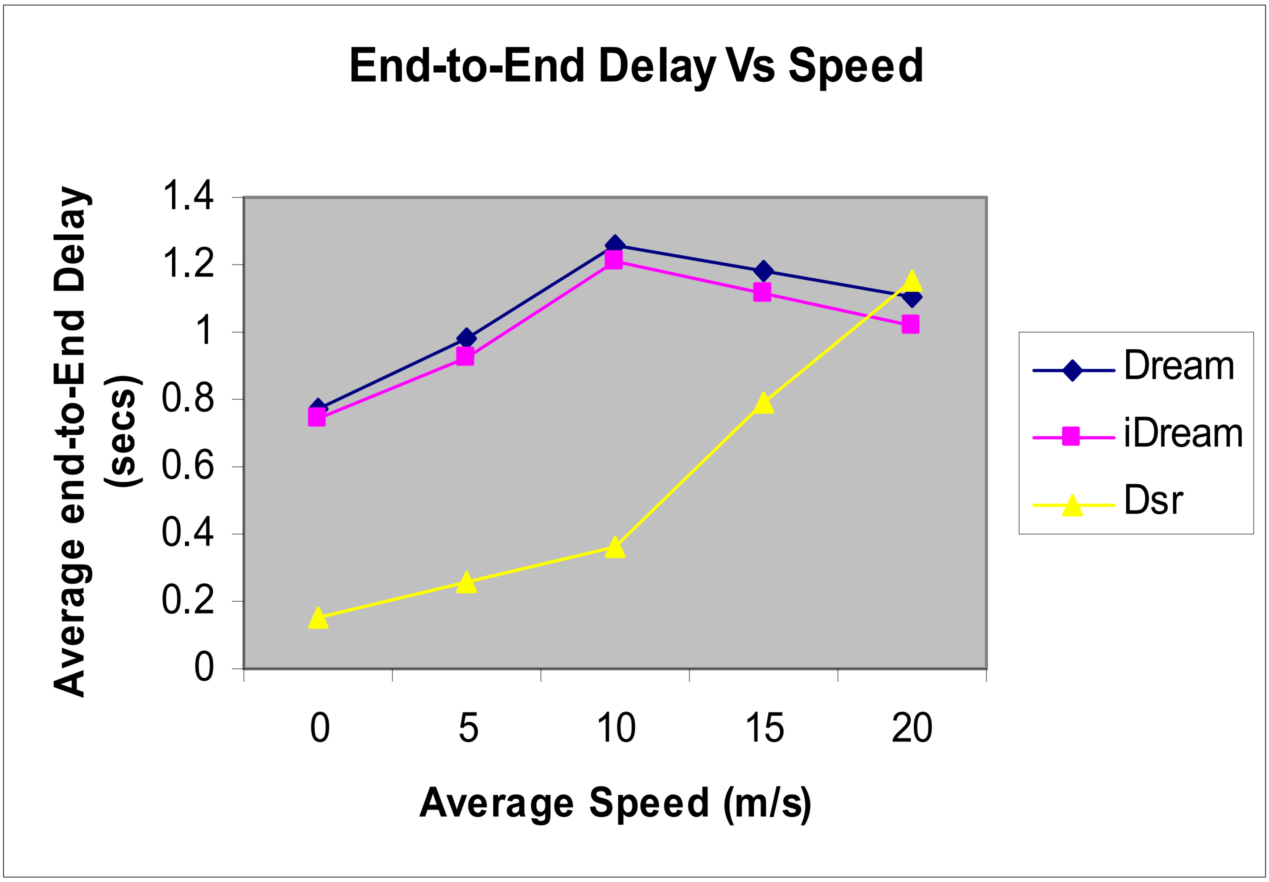
- Contention and congestion result in constant packet delivery ratio's for Dream and iDream as speed increases.

- It can be seen that the nature of Dream and iDream result in a higher packet delivery ratio of the protocol, as compared to DSR at higher speeds.
- The higher accuracy of determining the location of the destination node, result in iDream having a slightly higher delivery ratio as compared to Dream.
- The delivery ratio of DSR decreases considerably at higher speeds as it is much more difficult to find a usable route to a destination.

6.1.2. End-to-End Delay

The average end-to-end delay is calculated from the time taken for a data packet to arrive at the destination for every data packet transmitted. In the simulation studies performed, the average end-to-end delay has been calculated at different speeds. below illustrates the average end-to-end for the three protocols under study at various speeds.

Figure 6-2 Data Packet Delivery Ration vs. Speed



- At zero speed, DSR has an extremely low end-to-end delay. This can be attributed to the fact that the route discovery procedure needs to be performed only once.

- Dream and iDream have a high end-to-end delay even at zero speed, due to flooding, resulting in contention and congestion.

- It can be noted from the above figure, that all three protocols have a higher end-to-end delay with increasing speed.

    - This can be attributed to the fact that as speed increases, i.e., node mobility increases, route paths between nodes change more frequently. At zero speed, location information, for both Dream and iDream protocols, do not change. However, due to congestion and contention as speed increases, data packets may not reach the destination, or may reach the destination after the allowable timeout for receiving an ack for the data packet.

    - Similarly, with increased speed, more route requests need to be performed with the DSR protocol, resulting in higher end-to-end delays.

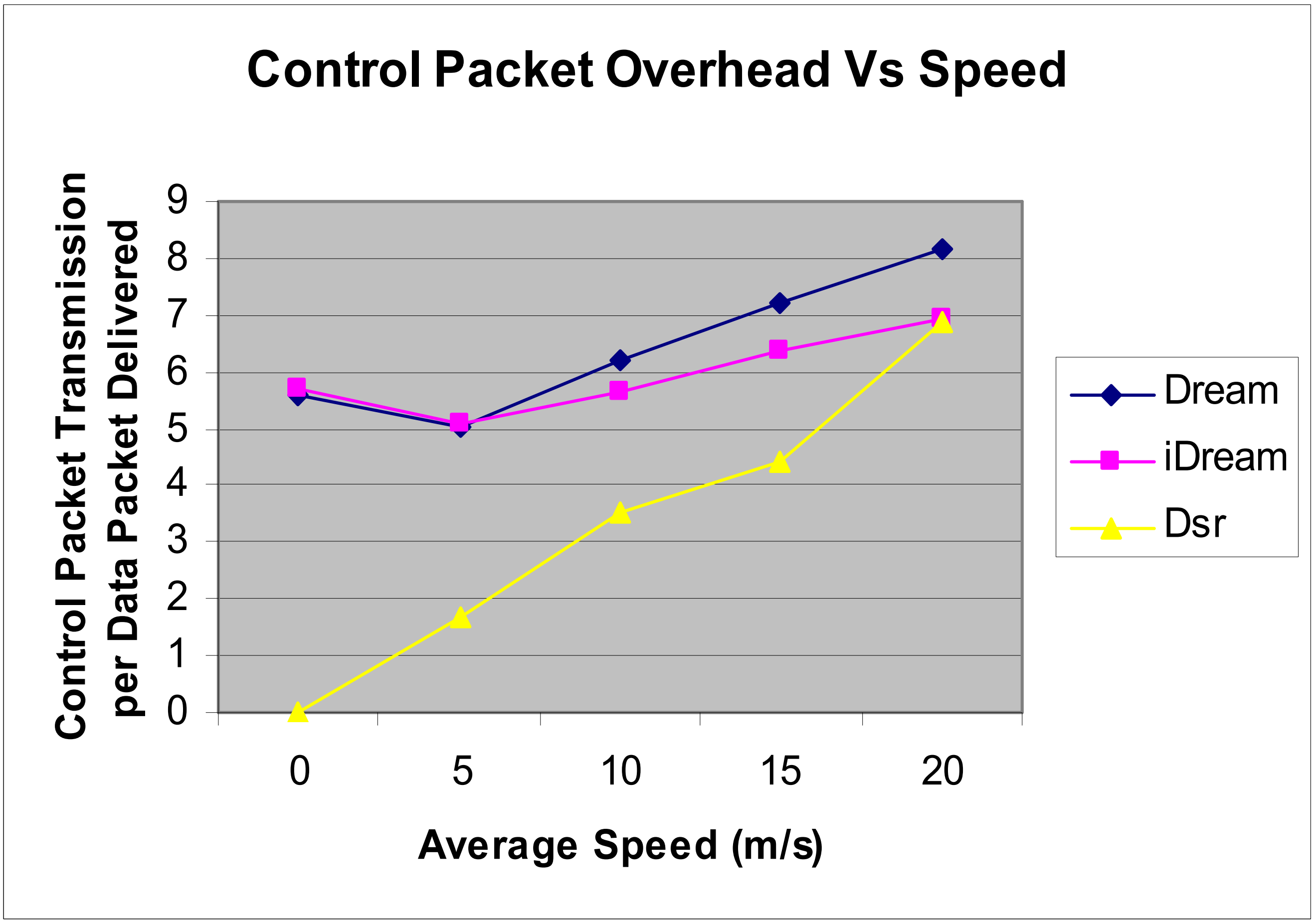
- At higher speeds, Dream and iDream perform significantly better than the DSR protocol. This is a result of the fact that route dissemination, both long and short lived, result is a constant update of the network with regards to routing paths. This conveys that Dream and iDream adapt well within a high mobility environment.

- It can be seen that iDream has a slightly and consistently lower end-to-end delay as compared to the Dream protocol. This can be seen as an enhancement provided by iDream over Dream due to the inclusion of direction information within the routing packets.

6.1.3. Control Packet Overhead

The overheads associated with each protocol are calculated by determining the number of control packets generated for every data packet. This determination is made for varying degrees of speed during the simulation. Determining the control packet overhead helps understand the power requirements for each protocol. As discussed previously, power is a scare commodity in a mobile environment and efficiency of a protocol needs to be studied with respect to its power requirements.

Figure 6-3 Data Packet Delivery Ration vs. Speed



- Dream and iDream constantly transmit small packets as part of the location dissemination process. Both the protocols have a high control packet overhead, being proactive protocols. In addition, each data packet transmitted is associated with an ACK packet, further increasing the control packet overheads.

- At zero speeds, DSR has no control packet overheads, as there are no route request or route reply packets generated. In comparison, due to the above mentioned reasons, Dream and iDream have a significantly higher control packet overhead.

- The control packet overheads of DSR increase significantly as speed increases, as result of more route errors and route recovery packets being transmitted at higher speeds.

- With increasing speeds, both Dream and iDream have higher overheads. This is accounted for by location information dissemination occurring more frequently at higher speeds.

- Also, iDream has a lower control packet overhead as compared to Dream. This is attributed to the intelligence of iDream with the direction information, as more packets are delivered more easily to the destination, without resorting to the recovery procedure.

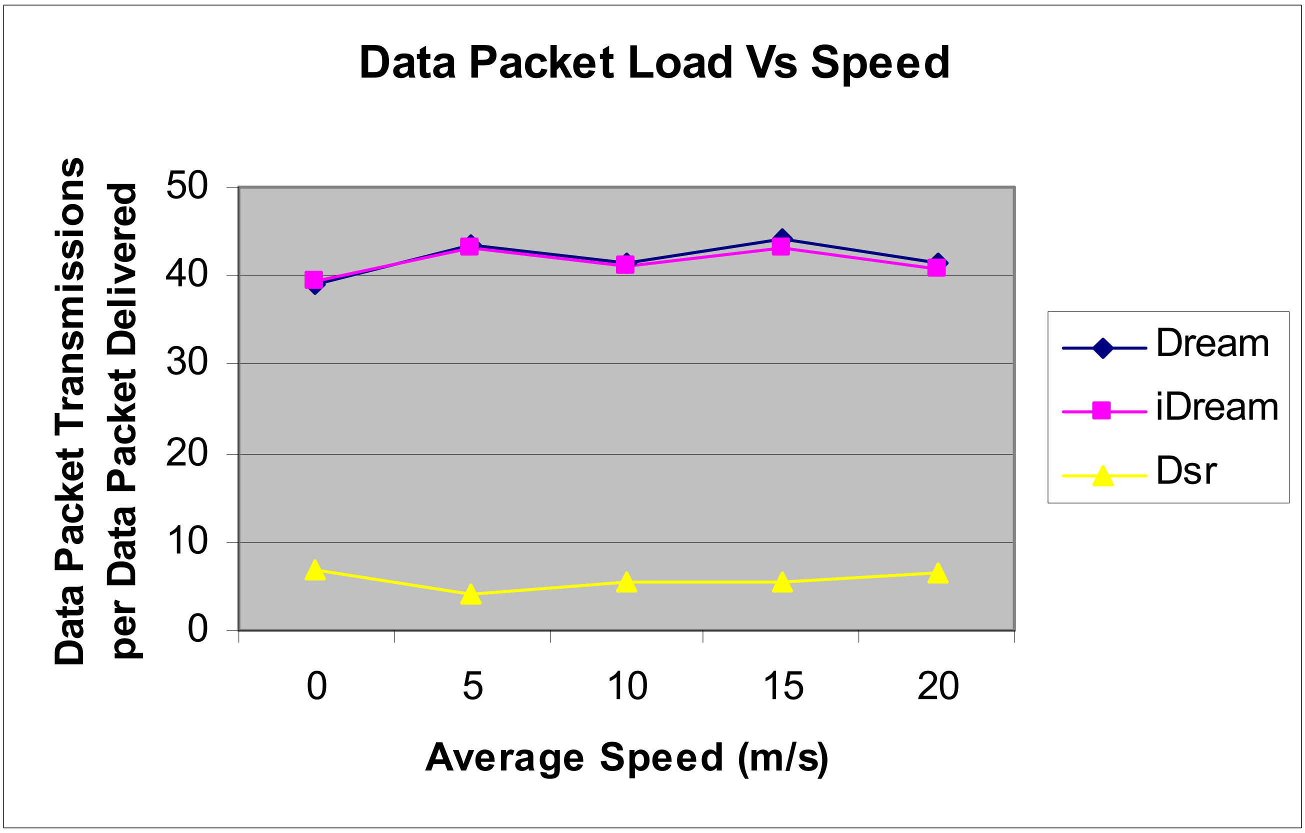
6.1.4. Data Packet Load

Data packet load represents the total load offered by the introduction of data packets into the network or the number of data packets introduced into the network per data packet generated by the source. The offered load is calculated for various speeds or varying degrees of node mobility.

- The Dream and iDream processes forward data packets to all nodes within the sector, identified in the direction of the destination node i.e. both protocols do not try a unicast method of trying to deliver data packets. The above results in multiple data packets being generated for each data packet issued from the source node. This results in Dream and iDream generating a high data packet load within the network as compared to DSR.

- As speed increases, the data packet load introduced by Dream and iDream remain nearly constant due to the flooding nature of the protocols.

Figure 6-4 Data Packet Load Vs. Speed



- At higher speeds, iDream tends to have a slightly lower data load as compared to Dream due to the efficiencies introduced by the protocol.

- DSR, being a unicast protocol i.e. the same data packet being transmitted and forwarded within the network, has a lower and constant data packet overhead.

7. Conclusions and Future Direction

The area of ad-hoc networking has received growing attention from researchers with the advent of powerful mobile computing devices, and the ability to implement the technology. A variety of ad-hoc routing protocols have been discussed, with particular focus on location based routing protocols. The focus of this study, within the location based protocols, has been the Distance Routing Effect Algorithm for Mobility (DREAM). An attempt has been made to enhance the DREAM protocol, with the proposal of the Improved DREAM protocol.

A few conclusions can be made from the study conducted on the Dream and iDream protocols. The Dynamic Source Routing (DSR) protocol has also been studied as a comparison to traditional source routing based protocols.

- The iDream protocol introduced within this thesis, studies the concept of "direction of motion" within the Dream protocol. The introduction of this vector enhanced the ability of the location based protocol to determine the location of a destination with greater accuracy, and therefore brought about greater efficiencies to the original DREAM protocol.
- Results from simulations conducted showed that iDream introduced a slight improvement on the Dream protocol, in each case studied. Particularly, improvement was pronounced at higher speeds, indicating that the iDream protocol is more efficient at higher speeds. Therefore, iDream may be better suited in a high mobility environment.
- The end-to-end delays introduced by iDream was also studied as a part of the theses and found to be lower than the Dream protocol. End-to-end delay signifies the time taken for a data packet to reach its destination, once generated by the source. The lower end-to-end delays for the iDream

protocol indicates that data packets reach the destination faster at higher speeds as compared to the Dream protocol.

- Control and data packet overheads were also studied, as the number of these packets represent the overall efficiency of the protocol. For both the Dream and iDream protocols the overhead is found to be similar given that the underlying algorithms of the protocols remain the same.
- The data packet overhead may be controlled by reducing the forwarding zone of the Dream and iDream protocols i.e. reducing the sector angle α̣̃However, if no nodes are found within the forwarding zone, the current logic resorts to a flooding algorithm i.e. forwarding to all nodes within the vicinity. However, as an improvement, it should be possible to increase the forwarding zone first, before resorting to flooding.
- It is seen that at higher speeds, both Dream and iDream protocols perform better or equal to the DSR protocols. Therefore, Dream and iDream protocols may be better suited in a high mobility environment.

In addition to the above conclusions, there are areas where research may be conducted to further understand the nature and application of the iDream protocol.

- In the current implementation, when data is received by the destination, it may be beneficial for the data and ack packets to record the exact nodes in all the hops. Once this route has been determined, the source node can specify this path information to the next data packets and limit the multicasting of data packets to too many nodes. This would improve the data packet overload within the network.

- In addition, the data and ack packets may used to carry location information as well i.e. perform control packet functions as well, thereby improving the efficiency of the overall protocol.

- It should be considered to develop a hybrid protocol of DSR and iDream, where DSR is used in a low mobility environment and the nodes switch to iDream in a high mobility environment, given the efficiencies of iDream in such an environment

- Also, as mentioned previously, an improvement may also be made in selection of the forwarding region. When a suitable next hop node is not found in the forwarding zone, the source nodes shoud first attempt to expand the forwarding zone before trying to flood the network.

- Finally, most simulations have been conducted in a simulation and controlled environment. A practical real-life implementation and study would be beneficial to understand the performance and applicability of the protocols.

Given the high interest within the area of MANET routing protocols, there are many issues to be researched and resolved before reaching a stage of being commercially or practically viable. This thesis paper attempted to understand MANET routing protocols and provide an insight into a few of the existing routing protocols while introducing some efficiencies.

References

[1] KAPLAN, E. D., Ed. *Understanding GPS: principles and applications.* Artech House, Boston, MA,1996.

[2] Scott Corson and Joseph Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”. Internet-Draft, draft-ietf-manet-dsr-00.txt. March 1988 Work in progress.

[3] Stephen Kent and Randall Atkinson, “Security Architechture for the Internet Protocol”, Internet Draft, draft-ietf-ipsec-arch-sec-07.txt July 1988 Work in progress.

[4] Mobile Ad Hoc Networks (MANET). URL: http://www.ietf.org/html-charters/manet-charter.html

[5] Charles E. Perkin, ‘Ad Hoc On Demand Distance Vector (AODV) Routing”. Internet Draft, draft-ietf-manet-aodv-01.txt August 1988

[6] Zygmunt J. Haas and Marc R. Pearlman, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, Internet Draft, draft-ietf-manet-zone-zrp-o1.txt August 1988

[7] M.Scott Corson, S. Papademetriou, Phillip Papadapolous, Vincent D. Park and Amir Qayyum, “An Internt MANET Encapsulation Protocol (IMEP) Specification”. Internet Draft, draft-ietf-manet-imep-spec01.txt August 1988

[8] Vincent D. Park and M.Scott Corson, “Temporally-Ordered Routing Algorithm (TORA) Version 1: Functional Specification”. Internet Draft draft-ietf-manet-tora-spec-01.txt August 1988

[9] Vincent D. Park and M.Scott Corson, “A Performance comparison of the the Temproally-Ordered Routing Algorithm and Ideal Link State routing”. In Proceedings of IEEE Symposium on Computers and Communciation ’98 June 1998.

[10] 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.

[11] David B. Johnson and David A. Maltz, “Protocols for Adpative Wireless and Mobile Computing”. In IEEE Personal Communications 3(1), February 1996

[12] MingLiang Jiang, Jinyang Li and Yong Chiang Tay, “ Cluster Based Routing Protocol (CBRP) Functional Specification. Internet Draft, draft-ietf-manet-cbrp-spec-00.txt August 1998

[13] Raghupathy Sivakumar, Prasun Sinha and Vaduvur Bhargavan, “Core Extraction Distributed Ad-Hoc Routing {CEDAR) Specification”. Internet Draft, draft-ietf-manet-cedar-spec-00.txt October 1988

[14] Bommaiah, McAuley and Talpade “AdHoc Multicast Routing Protocol” Internet Draft, draft-talpade-manet-amroute-00.txt August 1998

[15] Philippe Jacquet, Pual Muhlethaler and Amir Qayyum, “Optimised Link State Routing Protocol” Internet Draft drat-ietf-manet-olsr-00.txt November 1998

[16] Elizabeth Royer and C-K Toh " A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks " , IEEE Personal Communications Magazine, April 1999

[17] C. E. Perkins and P. Bhagwat, \Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Computer Communications Review, October 1994.

[18] S. Murthy and J. J. Garcia-Luna-Aceves, \An E_cient Routing Protocol for Wireless Networks," ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks, October 1996.

[19] Y.Ko and N.H.V aidya, “Location-aided routing (LAR) in mobile ad hoc networks,” in Proceedings of the ACM/IEEE InternationalConference on Mobile Computing and Networking (Mobicom), 1998,

[20] Kevin Fall and Kannan Varadhan, *NS2 Network Simulator Reference Manual*, 2000

[21] IEEE Computer Society LAN MAN Standards Commitee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std 802.11-1998, The Institute of Electrical and Electronics Engineers, New York, 1998

[22] D. Plummer, *An Ethernet Address Resolution Protocol - or - Converting Network Protocol Address to 48 bit Ethernet Address for Transmission on Ethernet Hardware*, RFC826, MIT, 1982.

[23] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In Proceedings of the Fourth Annual International Conference on Mobile Computing and Networking (MobiCom'98), ACM, Dallas, TX, October 1998.

[24] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks'" Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), 2002.

[25] J.Broch, D.Maltz, D.Johnson, Y.Hu, and J.Jetc heva, “Multi-hop wireless ad hoc network routing protocols,” in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom), 1998.

[26] P.Johansson, T.Larsson, N.Hedman, B.Mielczarek, and M.Degermark, “Routing protocols for mobile ad-hoc networks - a comparative performance analysis,” in Proceedings of theACM/IEEE International Conference on Mobile Computing andNetworking (Mobicom), 1999.

[27] J.Boleng, “Normalizing mobility characteristics and enabling adaptive protocols for ad hoc networks,” in Proceedings of the 11th Local and Metropolitan Area Networks Workshop, March 2001.

[28] Larsson, Tony; Hedman, Nicklas, “Routing protocols in wireless ad-hoc networks: a simulation study” Lulea University of technology, 1998.

[29] The CMU Monarch Project. ‘The CMU Monarchs Projects Wireless and Mobility Extensions to ns’ URL: http://www.monarch.cs.cmu.edu/

[30] Goodman, David J., Wireless Personal Communications Systems, Addison-Wesley Longman Inc., August 1997.

[31] Tanenbaum, Andrew S., Computer Networks, Prentice Hall Inc, USA, 1996.

[32] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/