Spatial Multipath Location Aided Routing

Soumendra Nanda

Dartmouth College

Follow this and additional works at: https://digitalcommons.dartmouth.edu/masters_theses

Part of the Computer Sciences Commons

Recommended Citation


https://digitalcommons.dartmouth.edu/masters_theses/6

This Thesis (Master's) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Master's Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
SPATIAL MULTIPATH LOCATION AIDED ROUTING

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of
Master of Science
in
Computer Science
by
Soumendra Nanda
DARTMOUTH COLLEGE
Hanover, New Hampshire

Examining Committee
(chair) Robert S. Gray
Daniela Rus
Brad Karp
Susan McGrath

Carol Folt
Dean of Graduate Studies

Dartmouth College
Department of Computer Science
Technical Report TR2005-533
Abstract

Mobile ad-hoc networks (MANETs) are infrastructure-free networks of mobile nodes that communicate with each other wirelessly. There are several routing schemes that have been proposed and several of these have been already extensively simulated or implemented as well. The primary applications of such networks have been in disaster relief operations, military use, conferencing and environment sensing. There are several ad hoc routing algorithms at present that utilize position information (usually in two dimensional terms) to make routing decisions at each node. Our goal is to utilize three-dimensional (3D) position information to provide more reliable as well as efficient routing for certain applications. We thus describe extensions to various location aware routing algorithms to work in 3D. We propose a new hierarchical, zone-based 3D routing algorithm, based on GRID by Liao, Tseng and Sheu. Our new algorithm called "Hyper-GRID" is a hybrid algorithm that uses multipath routing (alternate path caching) in 3D. We propose replacing LAR with Multipath LAR (MLAR) in GRID. We have implemented MLAR and are validating MLAR through simulation using ns-2 and studying its efficiency, scalability and other properties. We use a random waypoint mobility model and compare our MLAR approach versus LAR, AODV and AOMDV in both 2D and 3D for a range of traffic and mobility scenarios. Our simulation results demonstrate the performance benefits of MLAR over LAR and AODV in most mobility situations. AOMDV delivers more packets than MLAR consistently, but does so at the cost of more frequent flooding of control packets and thus higher bandwidth usage than MLAR.
# Table of Contents

**Chapter 1: Introduction** ................................................................. 1
  1.1 Overview ......................................................................................... 1
  1.2 MANET ................................................................. .......................... 2
  1.3 Wireless Routing Protocol Basics .................................................. 4
  1.4 Classification of Routing Protocols .................................................. 5
  1.5 Geographic Routing ................................................................. 6
  1.6 Thesis Organization ................................................................. 7

**Chapter 2: Research Objectives** ........................................................ 9
  2.1 Why use 3D position information? ................................................. 11
  2.2 AODV ................................................................. .......................... 15
  2.3 Multipath Routing ................................................................. 15

**Chapter 3: The GRID Protocol** .......................................................... 17
  3.1 Why GRID? ..................................................................................... 18

**Chapter 4: Implementation Details** .................................................... 25
  4.1 Simulator Information ................................................................. 26
  4.2 Our 3D Extensions to ns ............................................................... 26
  4.3 Implementation of Multipath LAR (MLAR) ...................................... 28
  4.4 Route Error Handling ................................................................. 31
  4.5 Geographic Location Information .................................................. 32

**Chapter 5: Performance Analysis** ....................................................... 34
  5.1 Testing Methodology ................................................................. 34
  5.1.1 Mobility Patterns ................................................................. 34
  5.1.2 Traffic Patterns ................................................................. 35
  5.2 Simulation Parameters ............................................................... 36
  5.3 Simulation Results in 2D ............................................................. 38
Acknowledgements

I would like to thank Bob Gray for believing in me and expertly guiding me right from day one of my scholastic career at Dartmouth College. It has been a privilege for me to work with such a talented individual who has on several occasions, gone the extra mile to help me. I thank my committee members: Daniela Rus, Susan McGrath and Brad Karp for their helpful suggestions as well as my PhD co-advisor David Kotz.

I would like to acknowledge the help I received from Tracy Camp and Jeff Bolding at the Colorado School of Mines for ns-2 simulations of LAR, Rachit Chawla and Mahesh Marina for their AOMDV code and Yu-Chee Tseng for the original GRID code. I would also like to thank Yu-Chee Tseng, Brad Karp and Ivan Stojmenovic for giving me permission to use their illustrations in this thesis.

I thank my wonderful girlfriend, Cindy Torres, for her unwavering support, motivation and invaluable help in finishing this thesis. Last but not least I would like to thank my family, especially my parents Kamales and Aparna Nanda, for teaching me to believe in myself and follow my dreams, my eldest sister Sarmila, for teaching me the importance of organization and my elder brother Soumitra for helping me select Dartmouth.

I dedicate this thesis in memory of my late grandfather, Shyamadas Bhattacharya who sadly passed away the day before I defended my thesis.
Chapter 1: Introduction

1.1 Overview

Mobile ad hoc networks (MANETs) are infrastructure free networks of mobile nodes that communicate with each other wirelessly. There are several routing schemes that have been proposed and several of these have been extensively simulated or completely implemented as well. The primary applications of such networks have been in disaster relief operations, military use, conferencing and environment sensing. Unlike conventional wireless networks one may find in offices, universities, communities or homes there is no central entity that controls how, when and where, packets are delivered to each recipient. All communication takes place in an ad hoc manner, which means on the fly and all the nodes in the network participate in relaying packets or messages to each other whenever it is possible for each node to do so.

There are several ad hoc routing algorithms at present that utilize position information (usually in two dimensional terms) to make routing decisions at each node. Our goal is to utilize three-dimensional (3D) position information to provide more efficient and reliable routing for various 3D scenarios such as urban rescue or ocean sensor networks to name a few. We thus describe extensions primarily to the GRID and Location Aware Routing algorithms (LAR) to work in 3D. We propose a new hierarchical, zone based 3D routing algorithm, based on GRID [LTS] by Liao, Tseng and Sheu. The new algorithm called “Hyper-GRID” is a hybrid algorithm that uses multipath routing in 3D. We intend to validate our algorithms through simulation using ns-2 and study their efficiency,
scalability and other performance related properties. We also aimed to create at least one realistic 3D mobility model to test our new algorithm and compare it with other algorithms extended to work in 3D.

We implement in this thesis Multipath Location Aided Routing (MLAR), a routing protocol that serves as a fundamental base to implement a complete version of HyperGRID. We compare the performance of MLAR against three comparable protocols in both 2D and 3D. We also propose a potential method of implementing GPSR in 3D. Some of the primary contributions of this work are:

1. Development of MLAR, a multi-path/ alternate path routing version of LAR
2. Detailed comparison of single and multipath versions of LAR and AODV in both 2D and 3D which provides guidance as to whether we can expect significant performance differences in 3D and as to whether single or multi-path algorithms should be used in a particular scenario.
3. Extensions to ns-2 version 2.26 to allow implementation and testing of 3D algorithms and mobility models.
4. Porting of LAR and MLAR code to the latest version of ns-2 so that all comparisons can be done on the same common platform.

1.2 MANET

The leading authority on Mobile ad-hoc networks or MANETs, as they are popularly abbreviated, is the Internet Engineering Task Force (IETF) working group whose goal is to standardize IP-level routing protocol functionality for wireless applications within both
static and dynamic topologies. As stated on the group’s website, the fundamental design issues are that the wireless link interfaces have some unique routing interface characteristics and the fact that node topologies within a wireless routing region may experience increased dynamics due to motion or other factors [MANET WEBSITE].

Nodes in a MANET are assumed to be mobile and communicate with other nodes wirelessly. The nodes in a MANET can be just about anything from micro-sensor equipped motes to Personal Digital Assistants (PDAs) to laptops or even computer systems embedded in vehicles. If one node needs to send a message to another node, it often has to send the message through multiple hops or intermediate nodes which themselves may be moving, thus causing frequent disconnections in the communication network. Radio interference, node movements, environmental factors, battery life and signal power all create a dynamic and challenging situation in which to send messages.

A wireless routing protocol in a MANET is the methodology or algorithm by which routes are created often with the help of routing tables in intermediate nodes in order to enable nodes to send packets to each other in a manner that is as efficient, reliable and error free as possible.

MANETS will prove popular in new and exciting applications in the near future for three basic reasons.

1. They can be deployed easily in several situations (nodes could possibly dropped into place by hand or by an airplane).

2. They can be deployed quickly and hopefully with economies of scale, cheaply as well.
3. They can lead to decreased dependence on prior or fixed infrastructure or provide alternative infrastructure in areas where current infrastructure fail.

Current interest in ubiquitous computing has given rise to the possibility that there will be thousands of devices, if not more, which will be networked wirelessly in the future homes of tomorrow. These devices would then form MANETs on their own for different durations of time in complex applications.

1.3 Wireless Routing Protocol Basics

There are several unicast routing algorithms that have been developed for MANETs that have their own unique characteristic strengths and weaknesses. A detailed description of all these protocols is beyond the scope of this thesis. We do describe in detail, however, all protocols that we felt were relevant to this work. Different algorithms may have benefits in different topologies and motion scenarios and for different application scales. For example, one protocol may work very well for 10 nodes in a small area but may work poorly (cause excessive delay or fail to deliver or drop most packets) for 100 nodes in a large area or in certain mobility conditions.

The simplest wireless routing protocol is called flooding and as the name implies, a message is sent by a node to all its neighbors who send it out to all their neighbors and so on until it reaches the desired destination. This is one method known to guarantee delivery of packets provided at least one path exists between any two nodes. It has a great drawback, however, in that it wastes a lot of the limited bandwidth available, and if all
nodes were to flood all other nodes, there would be too much interference, causing what is known as the Broadcast Storm problem [Storm]. Ideally, flooding should be avoided as much as possible or only done when absolutely necessary, such as in instances of very high mobility or to set up initial routes.

1.4 Classification of Routing Protocols

Most protocols can be classified in several ways. Some are classified as reactive or on-demand while others are proactive. In general, a proactive protocol finds routes in advance while a reactive protocol finds routes to the destination only when it absolutely must. For example, Ad hoc On demand Distance Vector routing (AODV) [AODV] is an on-demand protocol since no protocol information is transmitted before an application decides to send data and no data is sent until a route is formed, whereas Destination Sequenced Distance Vector protocol (DSDV) [DSDV] is a more proactive protocol in which routes are discovered and stored even before they are needed.

Proactive protocols generally generate much more traffic than on-demand protocols. A third general category is a hybrid algorithm that effectively combines multiple characteristics in a unique and meaningful way. For example, the Zone Routing Protocol (ZRP) [ZRP] is a hybrid protocol that combines local proactive routing with a globally reactive routing strategy.
1.5 Geographic Routing

Another possible way of characterizing MANET routing protocols is whether they utilize position information or not. AODV for instance does not use position information whereas protocols like GPSR [GPSR], GRID [LTS] and LAR [LAR] do use position information. GPSR, GRID and LAR and can be considered position based or geographic routing protocols since the position of each node is used as the basis for most routing decisions. It is assumed that individual nodes are aware of their own positions in absolute or relative terms as well as their velocity and the direction in which they are moving. This category is very relevant to this thesis since the protocols we propose lie in this category. At present there are already over thirty such position based protocols, as can be seen in the taxonomy of position based protocols by Ivan Stojmenovic and others [s1] and in [s2]. The following table, Table 1 - 1 reproduced with permission from [s1] shows some of these position-based protocols and indicates whether they are loop free, scalable, can always provide guaranteed delivery, etc. In spite of having 30 different approaches, the question that intrigued us, is whether there is room for further improvements and optimizations.
Table 2 - 1 A taxonomy of position based routing algorithms for wireless networks [S1]
(Reproduced with permission from Ivan Stojmenovic)

1.6 Thesis Organization

This chapter provides an introduction to the reader about the general domain this thesis pertains to, namely, wireless mobile ad hoc networking. Chapter 2 explains our research
objectives and motivations for using 3D scenarios and multipath routing. Chapter 3
describes the GRID and Hyper GRID protocols in detail. Chapter 4 explains
implementation details of the MLAR component of our Hyper Grid protocol and our
simulation system. Chapter 5 presents our results, insights and conclusions. Chapter 6
gives an overview of related work by other authors in ad hoc routing, specifically in
geographic routing and multipath routing. Chapter 7 suggests directions for future work
and Chapter 8 presents our conclusions from this study.
Chapter 2: Research Objectives

The basic advantage of using position information for wireless routing is to improve network scalability by reducing overall routing overhead. Location information can be used to reduce propagation of control packets, to perform controlled flooding, to maintain routes in mobility conditions and to make simplified packet forwarding decisions.

We hope to use position information and specifically positional information in three dimensions to find efficient routing solutions for several applications. Most traditional location aware routing algorithms in the literature make use of only 2D information, such as (x, y) planar coordinates.

Possibly at the time these algorithms were conceived the target applications did not require more information or were limited to 2D for reasons of computational or even notational complexity. It is much easier to conceptualize and represent most geometric problems in 2D instead of 3D. Some algorithms such as GPSR [GPSR] in its present form acknowledge that they will fail if the nodes all do not lie in almost the same plane. Karp does mention extending the GPSR algorithm to 3D as potential future work. It was this very fact that motivated our primary interest in building a 3D protocol that could provide the equivalent 2D performance of GPSR in 3D or provide a comparable point of referenc. In addition there has been at least one attempt [Kosuke] to extend GPSR to 3D and they provide some statistical analysis of how successful they feel there approach
could be by considering how frequently dead ends occur in 3D for a limited fixed size 3D space with a given single mobility model. They basically suggest that in case of a dead end or 3D void where no greedy choice is possible simply try another node that is least further away from the destination than the current dead end. This method of extending GPSR to 3D does not seem to guarantee delivery of packets if a path is actually does exist. In chapter 6, we describe their simple approach as well as present our own unique ideas on how to implement GPSR in 3D by considering a 3D perimeter approach, which we are still in the process of evaluating analytically.

Greedy algorithms are simple algorithms that select the next hop alternative based on a greedy strategy locally such as which hop is the geographically nearest next hop for a position based algorithm. Purely greedy based algorithms would work in 3D with little or no modification. However, 3D void regions where no greedy next hops are possible are not easy to deal with in a reliable or scalable manner. Some algorithms like LAR can be extended to 3D quite easily as explained in the next chapter, and are generally not hurt by 3D void regions too much. However, we decided to use this thesis to see if there is a significant change in their complexity, efficiency, space requirements, and other characteristics of 3D versus traditional 2D ad hoc routing protocols.

We also wish to contribute realistic 3D mobility models, which can be used to model future simulations. Most current mobility models used in network simulators like ns-2 and GlomoSim utilize only 2D information. Fortunately, ns-2 has support for nodal positions in 3D with the Cartesian z-axis values zeroed out by default and is relatively
easy to modify. One could argue that no one would need to use the z-axis since one does
not imagine mobile nodes as typically in the air or flying. The next section will explain
our reasoning via examples where we feel it is useful to consider using all three
dimensions.

2.1 Why use 3D position information?

In the real world every solid object structure occupies volume and has three dimensions.
We believe that any of the current location aware routing algorithms enumerated below
and in [s1], [s2] may fail or prove to be inefficient in certain 3D scenarios. Most greedy
decision based algorithms are likely to perform as well as they would in a 2D
environment and may be the only effective ones in cases of high mobility. They may not
be very efficient or have other drawbacks in other scenarios, however. Algorithms that
flood route requests globally would work in 3D in an analogous manner to 2D, but with
equivalently high bandwidth requirements determined by the frequency of flooded
packets.

Since radio waves are by nature inherently omni directional (unless one uses directional
antennas) some may consider or argue that the extra information in the 3rd dimension is
trivial or unimportant. We argue, however that there are several scenarios where the 3rd
dimension can provide crucial information for efficient routing and reliable delivery of
packets. The following scenarios are just a few illustrations of where we hope our work
will prove most useful:
1) Ocean Sensor Networks

The model we imagine here is a collection of ocean buoys containing sensors submerged at different points and different depths and drifting together as well as apart from each other at variable speeds. The sensors could be sensing temperature, pressure, oil leaks, radiation or even motion. We assume the sensors are aware of their position in 3D. We are certain that the 3D nature of this configuration, as well as its dynamic nature will prove a strong test for our theories and algorithms. Obviously any real solution used here has to be conservative in its use of power and resources. We intend to focus our attention on this model as it represents a generic case involving 3D positions and 3D mobility. While we do our testing on a random waypoint mobility model which is not exactly identical to the ocean sensors model as described above, but our model does provide a good generic worst case and has some similarities.

2) Urban Roof Top Networks

Various networks of this type are emerging and several small-scale solutions are even commercially available off the shelf (e.g. Nokia Rooftop Solutions). The idea is to have several nodes capable of transmitting and receiving wirelessly in a metropolitan scenario. They need not all be on the roof, but instead can be anywhere in a building or outside it. Such networks can provide alternative options when conventional infrastructure fails during a war or natural disaster, like a tornado or an earthquake.
It is interesting to note that the last scenario described the possibility of having exact positional information a priori and that most of the nodes (not all) are static or stationary. Thus it is possible to build a database containing the exact 2D position from GPS technology or by relative positioning and actually measure the elevation of most nodes to a very high degree of accuracy unlike other scenarios, given the limits of the technology applicable for the applications we are considering. Since in a metropolitan area, most buildings are at different elevations and different devices and antennas may have varying radio ranges, this is an interesting 3D scenario that presents its own challenges along with the effects of physical obstructions offered by the various materials as well as the layout of the building.

3) Military / Disaster Recovery Operations

In current military operations, soldiers typically have voice communication only, which makes it difficult to access needed information and coordinate mission activities. Ideally, each soldier would have a portable computing device through which they could query military databases, access maps of the surrounding terrain, view the positions of their fellow soldiers, and send complex observations to the mission planners at headquarters [Gray]. We consider a platoon or a battalion (say 100 to 1000 strong) of modern day soldiers armed with a wireless and GPS enabled personal digital assistant to guide them in the field. The PDA may be updated with mission critical orders or maps of the surrounding area that need to be downloaded at a moment’s notice. One group may
exchange data with other groups collected from sensors or notify of troop movements or
notify Head Quarters about damage inflicted or received. Alternatively, it could be a team
of firefighters trying to work through a multi-level industrial complex or commercial
complex of skyscrapers after a fire or natural disaster who need to update their
information about the structure, exit routes, maps etc.

In the case of the platoon, we imagine possible use of a helicopter or some other means to
help in synchronization and command-and-control from a higher elevation than the
troops. Some of the troops could be on flat terrain and some on a hillside or on a
mountaintop. Thus the positional information could be vital. Since the platoon is walking
it may be possible for the algorithm or sensors to use relative positional information
accurately or predict positions based on estimates of walking speed and group orders.

An alternate scenario is of a platoon inside a large high rise building structure and at
different floors combing the building looking for terrorists. We would like to point out
that GPS information via satellites is generally not available indoors, so there are obvious
engineering challenges in implementing this scenario. Another potential scenario is an
army trying to sweep an entire city on foot and this would involve motion through streets,
inside buildings and camping in areas on top of buildings and other strategic positions,
like water or telecom towers etc., covering large areas, elevations and structures.
2.2 AODV

Ad hoc On demand Distance Vector Routing (AODV) [AODV] introduced by Perkins and Royer in 1999 is an on-demand, reactive routing protocol and thus builds routes only when nodes require them. AODV builds routes using a route request / route reply query cycle and is a single path non position based algorithm. AODV is often considered a benchmark by which other protocols are compared in the literature and is discussed in much greater detail in Chapter 6. We compare the performance of our approach directly against that of AODV and it multipath equivalent Ad hoc On demand Multipath Distance Vector Routing (AOMDV).

2.3 Multipath Routing

The use of multipath routing for ad hoc networks is not new and has been studied by several authors as extensions to existing protocols as well as for entirely new ones. If multipath routes are stored but only one path is used at a time, multipath routing is generally called alternate path routing, where as if more than one path is used at the same time it is referred to as simultaneous or disjoint multipath routing. Stojmenovic et al [Location] show via simulation that, while multipath routing may increase routing overhead while finding multiple routes, they have the potential for helping in network traffic load balancing, if data is sent simultaneously along multiple paths. In simulation studies on Ad hoc On demand Multipath Distance Vector Routing (AOMDV) [AOMDV], where data is sent via just a single path at a time, the authors stated that AOMDV, a multipath variant of AODV, improved the packet delivery ratios for
CBR/UDP traffic by up to 40% and significantly reduced the packet delivery latency often by a factor of almost two. They also stated that routing overhead in their method was improved by 30% since less route discovery phases were required versus AODV. They do note, however that at higher mobility the performance difference between AODV and AOMDV is much lower. In our experiments we find AOMDV does better than AODV in terms of delivery ratio in most scenarios at a cost of increased flooding. The AOMDV protocol is explained in more detail in Chapter 6.

The original Dynamic Source Routing (DSR) protocol includes the optimization of using an alternate cached path when a path fails as an optimization, but did not explore it. Some other authors have proposed multipath DSR and alternate path DSR protocols and evaluated their performance via simulation. Thus we were curious to study how well a position based algorithm which we call MLAR for Multipath LAR routes using a multipath route caching strategy versus other state of the art non-position based algorithms AODV (single path) and AOMDV (multi-path), as well as the position based algorithm LAR (single-path) in both 3D and 2D.
In this chapter we explain the GRID protocol in detail to allow the reader to get a clearer idea of how and why we felt GRID should be modified for multipath routing in three dimensions. GRID [LTS] is a location aware ad hoc routing protocol that tries to exploit location information in its route discovery, packet relay and route maintenance phases. GRID divides the geographic region in consideration into a number of logical "grids" each being a square of the same size (Figure 3-1). Routing is performed in a grid-by-grid manner through grid leaders which are appointed within each logical grid. Route discovery and maintenance within a grid is done proactively and route discovery between different grids is done reactively. It is thus a hybrid protocol since it combines position based routing with a zone based hierarchical method of routing. Any leader in a grid square or box is thus exactly one hop away from any other leader in an adjacent box. Position based routing is useful for nodes to know which geographical grid they are in.
and which areas they can communicate with, and to identify where they wish to send packets to as well as what grid zones they should not send packets to. The hierarchical nature is due to the fact that only the grid leaders talk to each other and all nodes within a grid talk only to their grid leader. Thus it is a two level hierarchy.

Since grid leaders alone are responsible for route discovery the number of control packets required is effectively reduced. The grid nature allows for scoped flooding (Figure 3–2) preventing situations such as the broadcast storm problem [Storm] as shown in the diagrams below with S as the source and D as the destination.

3.1 Why GRID?

The primary focus of our work is to develop a new hybrid GRID based, multipath routing algorithm called “Hyper-GRID” for use in a 3D environment. We believe such an algorithm will be able to combine the best qualities of GRID and provide good performance, which we hope to prove analytically and by simulation. There are several algorithms at present that use location aware routing techniques. The key reasons we selected GRID as a starting point are clarified below.
As described by [LTS] in the GRID algorithm, the geographic area is partitioned into a number of squares called grids. In each grid, one mobile host (if any) will be elected as the leader of the grid. Routing is then performed in a grid-by-grid manner through grid
leaders; non-leaders have no such responsibility. Grids with no leaders are bypassed. The size \( d \) of each grid depends on the transmission radius \( r \). Several options are proposed in Figure 3–3 below, with the general idea of one leader being able to communicate directly with leaders in neighboring grids and all nodes within each grid being connected to their leaders.

Figure 3 - 3 Some of the possible relationships between \( d \) (length of each square) and \( r \) (radio transmission range)

(Reproduced with permission from [LTS])
Given that our primary application domain is to route efficiently in 3D space, GRID can be extended to 3D quite simply by using cubical grids instead of squares by adding a Cartesian z-axis (assuming the ideal condition that we have perfect location information in 3D). Scoped flooding is possible as well, by filtering out cubical regions based on their x, y and z coordinates (e.g. Consider someone A in the east wing of the basement wants to send a message to someone B who was last seen in the west wing of the third floor and heading towards the second floor, then A should make sure that the initial Route Request Packets do not go beyond the third floor of the west wing and thus limit the number of packets broadcast or flooded and how far they can travel in the network. Thus a packet that travels to the fourth floor will immediately be dropped by any node that knows it’s on the fourth floor). This idea of scoped flooding is inherited from LAR.

The initial results from simulation of the algorithm by the authors [LTS] indicate low performance overheads and it seems to scale well. The algorithm routes within zones in a proactive manner and between zones in a reactive manner (similar to the Zone Routing Protocol). Thus the hierarchical nature helps to provide scalability. Considering the future potential of ubiquitous computing applications, which may involve thousands of ad hoc devices talking to each other at a time, scalability will prove to be very important.

One of the GRID protocol’s best feature is that it uses flooding for route discovery only in certain limited regions of interest and thus can mitigate the "broadcast storm" problem [Storm] in a manner quite similar to LAR. The 3D equivalent requires the use of cubical partitions inscribed within spherical radio ranges originating at the center of each cube.
The zonal nature of this algorithm also provides potential for experimenting with various other inter-zonal and intra-zonal routing techniques, possibly more efficient ones that may be discovered in the future and can potentially be plugged in easily.

There is potential for optimizing the protocol presently used for route discovery, since it uses a variant of LAR by Ko and Vaidya [LAR] and may lead to loops, according to [s1]. We have implemented LAR in a loop free manner for our simulations, however since we use source routing. Another potential area of improvement is mechanisms to deal with empty zones as nodes move in and out of the various GRID zones and leave some of them empty. Routes are broken if the zones they initially passed through become empty and new routes must be rediscovered to continue routing of packets in the queue.

It may be possible to overcome this latter problem by using a multipath routing strategy as proposed in [Location] and [Terminode]. This was an avenue we wished to investigate for creating our own hybrid variation called "Hyper-GRID". We planned to implement it and compare its performance with a 3D non-multipath GRID algorithm (a greedy algorithm), an ideal shortest path algorithm and perhaps AODV in a 3D cubical random waypoint mobility model. We did not however implement a full blown version of Hyper-Grid for this study but implemented MLAR the 3D multipath routing component to replace the role of LAR in GRID. We intend to study the characteristics properties of our new and modified protocols such as delivery ratio, relative bandwidth utilization, end-to-end delay, hop count, etc in a directly comparable manner via simulation on a common and uniform platform.
Simultaneously, we consider simple ways to extend other popular geographic forwarding based algorithms to 3D. Some non-position based algorithms that can use mobility prediction schemes like [ODMRP] just require an additional dimension, i.e., the Z axis parameter to be included in the header and a minor modification to the trajectory prediction equations. Others like GPSR, require a little more thought and do not have obvious or efficient solutions. One of our initial goals was to extend GPSR to function in 3D to serve as a performance comparison point for our simulations. We were unable to do so completely in time for this study, however, we have made some progress towards that end.

For our simulation and performance studies in this thesis we implement a multipath variant of LAR for 2D and 3D scenarios in the ns-2 simulator. LAR can be considered as being an instance of GRID in which the size of each grid square or cube is so small as to hold just one node, making the node its own leader and follower. Since LAR is the core of the intra-zonal routing protocol in GRID we can hypothesize that the benefits of multipath LAR will translate to equivalent performance benefits in GRID or HyperGRID. We thus call our version of Multipath LAR as MLAR and simulate it in ns-2 and compare its performance directly with LAR, AODV and AOMDV on a common platform. While the word multipath here may indicate that we are using multiple paths simultaneously, we chose to use only one path at a time and save our other discovered paths as alternate paths for use on failure of the primary path. Using multiple
simultaneous paths is an approach explored in several other algorithms and has its own benefits and drawbacks as discussed in Chapter 6.
Chapter 4: Implementation Details

It is generally feasible to implement most wireless ad hoc protocols for use in the real world. Testing such an implementation with real hardware is quite hard, however in terms of the manpower, time and resources required to validate and experiment with the protocol and measure its characteristics in desired mobility scenarios. External conditions also can affect the measured performance characteristics. The preferred alternative is to model the system in a detailed simulator and then plug in various ad hoc protocols in a wide range of scenarios and measure their performance for various patterns of mobility and traffic. Simulation is not without its drawbacks obviously as even a single real world factor, such as the weather, humidity, real-world traffic patterns, human behavior, radio interference from other devices, physical obstacles, or material properties, might not be modeled perfectly and thus could produce entirely different performance characteristics from the ones discovered during actual use. Some basic assumptions that we have made in our simulations are:

1. We assume radio ranges of all nodes to be approximately equal and symmetric.
2. We assume a free space propagation model for radio transmission
3. We do not force link failures to occur by selectively turning nodes off. Most link failures in our simulations occur due to mobility or contention.
4. We do not check if the network is partitioned before we send packets and do not count packets that were undelivered because of an unreachable network configuration separately in our statistics
Relaxing these assumptions will be an important topic for future work, but they are appropriate for our initial development and comparison.

4.1 Simulator Information

We decided to implement our protocols in the popular network simulator ns-2. It is possible to simulate a mobile multi-hop ad hoc wireless network in ns-2 using a simulated 802.11 MAC layer. We selected ns-2 so that we could compare our approach with the other protocols on a single common and pre-validated simulation platform. Ns-2 Version 2.26 was the most recent version of the ns-2 at the time this work was started and served as a common platform for all the protocols that we wished to compare. With permission, we were able to use contributed code from several other authors for our study. We received a copy of the AODV code with the basic installation of ns-2 version 2.26. We received a compatible version of AOMDV from Rachit Chawla as well as an older version from Mahesh Marina that did not work on version 2.26 but worked on earlier versions. We did not modify the AODV or AOMDV code or any of their timeout values or parameters, which the contributing authors selected during their own evaluations. We received a copy of LAR for a much earlier and incompatible version of ns-2 by Jeff Boleng from Tracy Camp and her project team at the Toilers group at the Colorado School of Mines [Camp]. We modified their code to work with our version of ns-2 and to duplicate their performance results.
4.2 Our 3D Extensions to ns-2

We created 3D versions of the four routing protocols since they worked only in 2D by default. This required some changes to the core simulator and some additional tools were needed for creating traffic patterns and 3D movement patterns. AODV and AOMDV needed minimum changes since they are position independent. For LAR, the changes were quite simple and logical. Circular radio ranges were extended to spheres, the box method was extended to a cubical model, all distance calculations were extended to 3D and a free space propagation model was used since the two-ray ground model was found unsuitable. The 802.11 MAC layer was used with each node having a range of approximately 250 meters. The size of the simulation area is 670x670 meters for 2D and 670x670x670 meters for 3D. Some runs were done with nodes having radio ranges of 100 and 670 meters respectively as well to validate our protocols. The exact transmission range and simulation area is unimportant. We picked these values only to have a certain ratio between the radio range and the area dimensions. Several authors prefer using simulation areas where the length is longer than the breadth of the simulation area (as well as the transmission range) by a factor of two to guarantee that nodes make several hops across the area. We found that since our radio range was 250 meters, our selected area often had routes as long as 7 or 8 hops.

To create 3D movement patterns we modified the 2D random waypoint model to create 3D scenarios using the setdest tool from the cmu-tools directory provided with ns-2. Some recent work [Camp2] indicates the benefits of using the steady state waypoint model over the standard waypoint model. We attempted to create a 3D equivalent of the
tool \textit{mobigen-ss}, which creates a steady state waypoint mobility scenario in time for this study, but were unsuccessful and are working with the authors of the 2D version to create a 3D one for future use.

4.3 Implementation of Multipath LAR (MLAR)

For a detailed description of how LAR works please refer to the LAR section in Chapter 6. In order to create the multipath variant of LAR, which we refer to as MLAR, we started with the code base for LAR which we first extended to work in 3D as well as with the newest version of ns-2. LAR is basically a source routing algorithm like DSR with the entire hop-by-hop routing path in the header of each packet. The advantage it has over DSR is that it is location aware and tries to find routes with minimal flooding using the information available about the source and destination positions. As mentioned earlier, by multi-path we mean caching of alternate paths between the source and destination and not the use of simultaneous multiple paths between the source and destination which can lead to out of order packet delivery problems.

For MLAR we simply cached the two most recently received routes, similar to the optimization described in DSR mentioned earlier in this thesis. The reasoning for this was that in cases of high mobility, the most recently received route is more likely to be more successful. In the original LAR code we received, the most recently received route was always used. Thus in LAR the path used in the most recently received route reply would be the path used for the next data packet to be sent. Of the two routes in the MLAR
cache, the shorter one was selected as the primary route if it was the newer route. If both were entered in the cache at approximately the same time (the interval between two successively received paths to the same destination was less than a low threshold value), the shorter route was initially preferred. We were able to get significantly better performance, however, by selecting the most recently received path even if it was longer. The reasoning is quite simple: The most recently received path is likely to be the path most likely to succeed since mobility could cause paths to break, even if the older path in our cache was one or two hops shorter or had a shorter recorded round trip delay time for the route request and reply cycle.

In both LAR and MLAR routes never expire even if they are not used for extended periods of time, unless a route transmission error is detected. Since the packet header contains the entire source route, all the alternate paths are checked easily as being loop free. Initially we did not check the two paths for degree of link or node disjointedness, except for checking if the route is loop free or if one route is a sub-route of another, or if one route is identical to the route already in the cache. Even without considering path disjointedness, this naïve approach gave us a miniscule improvement (0.5%) in the data packet delivery ratio of MLAR over LAR in both 2D and 3D. Some other studies like AOMDV and Multipath DSR (M-DSR) make certain that the paths stored are ‘link disjoint’ and have no common hop between them or are ‘node disjoint’ and thus have no common nodes in their paths. AOMDV does this by deciding in a distributed manner at each node along the route if the path is link disjoint or node disjoint whereas other approaches like M-DSR let the destination examine the route request packets (RREQs) it
has received and the paths within them before sending route replies (RREPs) back to the source with the most disjoint paths. We thus modified our approach in MLAR to allow a node to accept a second route to a destination if and only if it was link disjoint with the first cached path. This is done quite simply by checking if the same link is in both paths, i.e., if the path in both routes contains the same two nodes in the same order, the nodes are ignored. This is done at all nodes whenever a routing table entry is updated, on receiving any data or control packet since in MLAR and LAR the entire source route is available in every packet.

During the simulation of MLAR, if the source route path in a data packet fails, the second path is tried if it exists. In LAR the packet would have been put into a queue at the node before the transmission failure and eventually dropped after a timeout if a new route to the destination was not discovered before the timeout. In LAR an error packet would also be sent back to the source to let it know of the broken path so that it can initiate a new route request and reply cycle. A path fails whenever the MAC layer reports back a transmission failure in reaching the next hop after a certain threshold number of resending attempts. In MLAR, if the second path also fails in the same manner from the source, a new route request cycle is initiated. If the second failure in MLAR is at an intermediate node, the node sends an error packet back to the source by the reverse route or broadcasts an error message back to the source so the source and all nodes that used that old path can invalidate their caches at least beyond the breakpoint where the failure occurred. The naive risk here is if the second path that we attempt to use is stale, then we will keep trying to use it until we get our first error packet or unless the first hop
transmission from the source is unsuccessful. This risk can partly be minimized by having routes expire after a reasonable or adaptive value of a timeout period as suggested by Das and Marina [MARINA] and several other authors as an optimization for DSR. We have not implemented this optimization for this study. However, even without this optimization we find MLAR performs consistently better than LAR in terms of delivery ratio by as much as 25% in some cases, similar to the way in which AOMDV outperforms AODV by exploiting the spatial disjointedness in the alternate paths and avoiding reinitiating request and reply cycles. We do note that AOMDV floods more packets throughout the network than AODV, and MLAR and LAR flood significantly far fewer packets into the network than both AODV and AOMDV by at least a factor of four times fewer packets.

We did not consider using three or more cached routes for our initial study since most studies indicate that the gains in caching three routes rather than two are very low, and that four or more paths generally provide insignificant improvements in highly mobile scenarios[Location], [SMR]. Again we would like to note that there is no benefit from using an alternate cached path if the cached path is outdated or inaccurate.

**4.4 Route Error Handling**

In MLAR if a node detects a link failure, it tries to retransmit the data packet using an alternate path from its own route cache by updating the packet header with the new alternate path. In either case, it sends a route error packet to the source to let it know of the broken link for future transmissions. One optimization that was not implemented was
having the node that detects an error inform the source of the new alternate route to the final destination if the alternate route was successful. In a highly mobile scenario, however, this could lead to the phenomenon of cache contamination if the information is wrong.

In the absence of route errors or transmission failures, MLAR should behave almost identically to LAR. The benefit from using a cached alternate route should not be overlooked, however. In order to do intermediate route repair, an intermediate route, looks into its cache and tries to find an alternate path to the destination and use it. In LAR, chances are that the path stored at the intermediate node is likely to be the same as the path in the source route in the header. Having two or more paths saved in MLAR, on the other hand provides an alternative for salvaging the packet. However, if the alternate path selected is stale and no longer available, however an error packet will be ultimately generated. On receiving an error packet which is generally flooded (route errors are only flooded when the unicast route back to the source fails at any point) to ensure delivery to the source, the source can try an alternate path if it has one or try and seek a new route via a route request cycle. Link disjoint and node disjoint paths ensure that routes fail independently of each other in most cases. The route error packet contains the addresses of the hosts at both ends of the hop in error and when it is traversing back, all routes in the route caches of all intermediate nodes containing the failed link will be removed from the caches and a new route discovery is initiated by the source if the route is still needed.
4.5 Geographic Location Information

A lot of geographic routing protocols assume the presence of Geographic Location Services (GLS) that allow each node to know the position of every other node. There have been a few attempts to implement such services, but most have very high overheads since information needs to be propagated throughout the network for every single significant movement. Some simulations make use of global knowledge of positions through hooks in the simulator code and state that they assume they know the exact position of destination nodes through an assumed perfect GLS that works separately.

Our approach for LAR and MLAR as in [Camp] uses previous knowledge of the position of a node if available. If not, it floods route discovery packets in an incremental and scoped manner until the destination is found (if the network is connected and it is reachable) or until the timer expires. Assuming the presence of a GLS would definitely decrease our routing overheads and improve our results if we were to consider the routing overheads for a GLS as an isolated entity. It would be interesting to see how MLAR and LAR would both perform when combined with a more advanced GLS integrated into the simulation as future work.
Chapter 5: Performance Analysis

5.1 Testing Methodology

We tested four protocols: LAR, MLAR, AODV and AOMDV using ns-2 for both 2D and 3D scenarios. We used the same traffic models and mobility pattern for each protocol and repeated our simulations five times for each scenario or combination to obtain an average and unbiased result or data point.

5.1.1 Mobility Patterns

We used a random waypoint mobility model which works as follows: A node is selected at a random time and it selects a random direction and a random speed chosen uniformly between zero and a pre-selected max speed and travels for a random duration. On reaching the destination it pauses for a random interval chosen uniformly between zero and a pre-selected pause time or rest period. If the pause time is selected as zero the node never pauses. We consider different values for max speed from zero to fifty meters per second (m/s) and vary the maximum pause time between 0 and 300 seconds. Our entire simulation lasts for 800 seconds. Thus each traffic pattern has two main parameters, the maximum speed and the maximum pause time. We have experimented with maximum speeds of 5, 10, 20 and 50 m/s and have used pause times of 0, 25, 50, 150 and 300 seconds respectively in our simulations. We allow the simulation to run for 50 seconds before we start sending data packets rather than send packets from the first second when nodes are all starting from their stationary positions. While the random waypoint model
does not correspond perfectly with the real-world movement patterns one would associate with the applications like ocean sensors and disaster recovery operations we described earlier, we consider this model an appropriate starting point since it represents the most general movement case (as no environmental effects are causing groups of nodes to move along the same trajectories, etc).

5.1.2 Traffic Patterns

We generate three types of traffic loads which we generalize as representing low, medium and high peer to peer bidirectional traffic loads. A node is selected at random at a random time and it selects a destination node at random. A random number of packets to be sent are selected uniformly from the interval $[5, 50]$ with each data packet payload being of a fixed size of 64 bytes. Each packet is sent from the source to the destination after a inter packet delivery delay which is chosen from an exponential distribution with a variable pre-determined mean which we can vary between what we consider low, high and medium values to control the traffic rate and volume. The destination also repeats the process and sends the same number of packets back to the source after a random delay.

Some studies such as [Camp][LAR] chose to have the same source nodes sending data to the same destination nodes after a fixed interval repeatedly throughout the duration of the simulation. We felt our traffic model provided a more randomized model of network traffic and a more general case and free of any potential bias caused by constant reuse of certain repeated traffic patterns and their effects on the cached routes.
5.2 Simulation Parameters

In this section we present the results of our simulations for scenarios in 2D followed by scenarios in 3D. The main results we present are the packet delivery ratio, latency and a relative measure of the bandwidth utilized by each protocol. We always use 50 nodes in each scenario with an approximate radio range of 250 m/s and a theoretical bandwidth limit of 2 Mb/s. The simulation area for 2D is a square flat ground of 670 by 670 meters while for 3D we use a bounded cubical space with each side being of 670 meters. The scale on the Y axis for several of the following graphs has been modified to best represent the data being presented and make certain crossover points distinctly visible.

Each data point on any graph is the result of an average of five runs for that combination of mobility pattern and traffic pattern. For any given combination of rate and volume, we created five traffic patterns with five different random seeds. Similarly for each combination of maximum speed and pause time parameters, we created five mobility patterns with a different set of five random seeds. We then paired the five traffic and five mobility patterns with each other, and ran a simulation with 50 nodes for 800 seconds, with no data being sent for the first 50 seconds. We ran the five combinations for the same parameters of traffic volume, traffic rate, maximum speed and maximum pause time and then average our results to arrive at a data point.

We define our Low Traffic Scenarios as having up to 500 bidirectional pairs of source destination connections with an inter packet delivery interval of 10 seconds selected from an exponential distribution. Thus by low traffic we mean few connections and packets sent at a low frequency. Each data packet always had 64 bytes as the data payload.
For Medium Traffic Scenarios we have a maximum of 1000 connections but the equivalent mean delay between packets is only 0.5 seconds or on average 2 packets per second.

Similarly, we define our High Traffic Scenario as having a maximum of 2000 connections with an equivalent mean delay of only 0.1 or on average 10 packets per second. It is also possible for any node to send packets to multiple recipients at the same time. We would like to point out that our traffic models are graded as low, medium and high in our relative terms and it is debatable whether the titles are appropriate. There is no doubt that these three cases are not exhaustive in their coverage of potential traffic patterns but each does produce its own distinct performance as can be observed from the following graphs. The same traffic patterns were used for both the 2D and 3D scenarios.

We also point out that the scripts used for analysis of results for LAR and MLAR in 2D and 3D were identical, but not the same as the scripts used for AODV and AOMDV in 2D and 3D. However, both provide approximately equivalent results in most of the parameters we measured from counters used during the simulation or by parsing the trace files from each simulation run. The reason we could not use the same scripts was because the trace formats used for the two protocols were extremely different.
5.3 Simulation Results in 2D

As is visible from the following graphs (Figure 5-1 to Figure 5-12) on delivery ratio versus mobility in various traffic scenarios in 2D, AOMDV performs better than AODV and all the other protocols consistently, and MLAR does better than LAR in most cases by similar margins. In some cases, the performance of MLAR and AODV is almost identical. At lower speeds or in very high pause time scenarios (fast moving nodes that move less frequently), the performance of all four protocols seems to converge. In low-traffic conditions where paths are reused less frequently, LAR and MLAR are at a disadvantage to AODV and AOMDV since stale routes may still remain in the cache after long periods due to the sparse and slow traffic patterns and lack of automatic expiration of such routes. It is the presence of multiple paths in the routing tables of MLAR and AOMDV that allow them to do better than LAR and AODV respectively.
Figure 5 - 1 Packet Delivery Ratio at a maximum speed of 5 m/s for a Low Traffic Scenario

Figure 5 - 2 Packet Delivery Ratio at a maximum speed of 10 m/s for a Low Traffic Scenario
Figure 5 - 3 Packet Delivery Ratio at a maximum speed of 20 m/s for a Low Traffic Scenario

Figure 5 - 4 Packet Delivery Ratio at a maximum speed of 50 m/s for a Low Traffic Scenario
Figure 5 - 5 Packet Delivery Ratio at a maximum speed of 5 m/s for a Medium Traffic Scenario

Figure 5 - 6 Packet Delivery Ratio at a maximum speed of 10 m/s for a Medium Traffic scenario
Figure 5 - 7 Packet Delivery Ratio at a maximum speed of 20 m/s for a Medium Traffic Scenario

Figure 5 - 8 Packet Delivery Ratio at a maximum speed of 50 m/s for a Medium Traffic Scenario
Figure 5 - 9 Packet Delivery Ratio at a maximum speed of 5 m/s for a High Traffic Scenario

Figure 5 - 10 Packet Delivery Ratio at a maximum speed of 10 m/s for a High Traffic Scenario
5.3.1 Delivery Ratio versus Varying Speed

Here we can see the difference between the three traffic scenarios as we vary the maximum speed for nodes that never pause. At lower speeds with the exception of the
low traffic scenario, the difference between the four protocols is negligible. However, at high speeds like 20 and 50 m/s we find that AOMDV does much better while the performance of MLAR and LAR is neck to neck and AODV does the worst. In the low traffic pattern, cache buildup and infrequent reuse is probably why AODV does better than LAR and MLAR. The performance of AOMDV is a little surprising since it seems unusually high even at 50 m/s, which indicates that the increased flooding is not saturating the network. AODV drops packets frequently and generally does the worst. In any case we feel we have achieved our first goal of MLAR improving on the performance of LAR in terms of delivery ratio. The performance gap between MLAR and LAR is more pronounced in the low traffic scenario (Figure 5-13) and exists to a lesser extent in the medium (Figure 5-14) and high traffic scenarios (Figure 5-15).

![Figure 5 - 13 Packet Delivery Ratio vs. Node Mobility in a Low Traffic Scenario](image-url)
5.3.2 Average End to End Delay

We have demonstrated that MLAR does better than LAR and in most cases AODV in terms of delivery ratio for most 2D scenarios. We now take a look at average end to end delay in LAR and MLAR as calculated by our scripts. Whenever a node receives a data packet it notes in a cumulative sum the exact time it received the packet minus the time it was originally sent and finally at the end of the simulation, we divide the sum of these
transit times by the total number of packets received to get an average end to end latency. In most cases we find the latencies are well below 0.2 seconds (84.5% of the data packets have a latency under 0.05 seconds, and 94.7% have a latency under 0.2 seconds). A few packets, however, wait in a node’s queue for completion of an abnormally long route discovery (e.g., a route appears after a node moves), and the waiting time dramatically increases the latency for these packets. These outliers can dramatically affect the average per-packet latency as seen in the graphs. Thus in a low traffic and low mobility scenario (5 m/s with zero pause time) we find that LAR and MLAR deliver more packets within 0.1 seconds than AODV or AOMDV but take longer on average to deliver all their packets as seen in Figure 5-16 below.

![Graph showing average end to end delay characteristics in a low traffic low mobility scenario](image)

**Figure 5 - 16 Average End to End Delay Characteristics in a Low Traffic Low Mobility Scenario**
5.3.3 Control Overhead in 2D

During our analysis we kept several counters to measure the total number of control packets such as route requests, route replies, route errors, number of packets flooded, etc. AOMDV allows for more RREQ and RREP packets in the network in order to build multiple paths to each destination for each node. AODV allows only for a single RREP packet, for the first RREQ the destination node received to be sent back via the reverse route it arrived in. As a result we see AOMDV has 10% more RREQs sent than AODV and about 9 times as many RREP packets sent overall as seen consistently in most scenarios. The numbers do not vary significantly with changes in mobility from 5 m/s all the way to 50 m/s. Consider the following data from a high traffic (10 packets per second) and very high mobility scenario (50m/s with no pausing).

![Figure 5 - 17 Control Packets Received for a High Mobility and High Traffic scenario in 2D](image-url)
<table>
<thead>
<tr>
<th></th>
<th>Flooded RREQs Sent</th>
<th>RREP sent</th>
<th>Route Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV 2D</td>
<td>54741</td>
<td>1882</td>
<td>56623</td>
<td></td>
</tr>
<tr>
<td>AOMDV 2D</td>
<td>64137</td>
<td>14910</td>
<td>79047</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 - 1 Control Packets Sent for a High Mobility and High Traffic Scenario in 2D

<table>
<thead>
<tr>
<th></th>
<th>Flooded RREQs received</th>
<th>RREP received</th>
<th>Route Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV 2D</td>
<td>802730</td>
<td>2958</td>
<td>805688</td>
<td></td>
</tr>
<tr>
<td>AOMDV 2D</td>
<td>904878</td>
<td>25387</td>
<td>930265</td>
<td></td>
</tr>
<tr>
<td>LAR 2D</td>
<td>89553</td>
<td>11099</td>
<td>100652</td>
<td></td>
</tr>
<tr>
<td>MLAR 2D</td>
<td>170465</td>
<td>23034</td>
<td>193499</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 - 2 Control Packets Received for a High Mobility and High Traffic Scenario in 2D

It is easy to observe that the total number of control packets injected into the network in AODV and AOMDV is almost ten times that of LAR and five times that of MLAR and the most important figure is the number of flooded RREQs. In case of LAR, 51882 or more than half of those RREQ packets received were single hop route requests and similarly for MLAR, 107724 of the 170465 received RREQs were single hop route requests. The rest were flooded but even if the total number is considered it is significantly lower than AODV or AOMDV. Scoped flooding is another factor why LAR and MLAR flood significantly less that AODV and AOMDV. For this scenario LAR generated on average 1308 Error packets while MLAR correspondingly generated 3094 Error packets indicating that the cached primary and alternate routes failed often. The cached routes were successful on a number of occasions, however, which explains the significantly improved delivery ratio of MLAR over LAR and of AOMDV over AODV. The higher number of failures is also the reason why MLAR sends more RREQs than
LAR, as RREQs are generated at the source when it receives a RERR. We noticed that these relative characteristics were observed in practically all the scenarios and was very consistent and practically independent of the mobility parameters. An almost identical observation is made in our 3D results where we again see that AODV and AOMDV flood more than LAR and MLAR by a factor of ten even for low mobility and low traffic conditions.

5.4 Simulation Results in 3D

In the following graphs (Figure 5-18 to 5-29) we examine the performance of our four algorithms in 3D scenarios. The trends are very similar to those in the 2D scenarios with one notable difference. The performance gap between MLAR and LAR is consistently a little higher in 3D than in 2D. We believe this is potentially due to difference in node density in the 2D and 3D scenario. In both 2D and 3D scenarios, we have 50 nodes, but in the 3D scenario, the density of nodes is lower and the maximum diagonal distance between opposite corners is significantly larger. In the very first graph with slow moving nodes in low traffic, the performance of MLAR almost matches that of AOMDV. In most cases AOMDV does the best and MLAR comes next and almost always does better than AODV and LAR. In some cases LAR does better than AODV especially as the traffic load tends to increase in high mobility or low pause time scenarios. Again the results for AOMDV are higher than one would expect, especially in the extremely high mobility scenarios at 50 m/s but its use of bandwidth is also significantly higher due to the larger number of flooded requests and route replies in the network as well as route
maintenance packets. The performance of LAR, MLAR and AOMDV seem to converge when the pause time is very high (300 seconds) in almost all medium and high traffic scenarios which indicates that the routing performance is close to the optimal value it can obtain. Please note the change of scale for some of the graphs where the packet delivery ratio values were very close to each other.

Figure 5 - 18 Packet Delivery Ratio at a maximum speed of 5 m/s for a Low Traffic Scenario

Figure 5 - 19 Packet Delivery Ratio at a maximum speed of 10 m/s for our Low Traffic Scenario
Figure 5 - 20 Packet Delivery Ratio at a maximum speed of 20 m/s for a Low Traffic Scenario

Figure 5 - 21 Packet Delivery Ratio at a maximum speed of 50 m/s for a Low Traffic Scenario
Figure 5 - 22 Packet Delivery Ratio at a maximum speed of 5 m/s for a Medium Traffic Scenario

Figure 5 - 23 Packet Delivery Ratio at a maximum speed of 10 m/s for a Medium Traffic Scenario
Figure 5 - 24 Packet Delivery Ratio at a maximum speed of 20 m/s for a Medium Traffic Scenario

Figure 5 - 25 Packet Delivery Ratio at a maximum speed of 50 m/s for a Medium Traffic Scenario
Figure 5 - 26 Packet Delivery Ratio at a maximum speed of 5 m/s for a High Traffic Scenario

Figure 5 - 27 Packet Delivery Ratio at a maximum speed of 10 m/s for a High Traffic Scenario
Figure 5 - 28 Packet Delivery Ratio at a maximum speed of 20 m/s for a High Traffic Scenario

Figure 5 - 29 Packet Delivery Ratio at a maximum speed of 50 m/s for a High Traffic Scenario
5.4.1 Delivery Ratio versus Varying Speed

The results and trends in Figure 5-30, 31 and 32 are quite similar to those we observed in the 2D cases. AODV performance seems to deteriorate with increasing traffic and increasing mobility. AOMDV does the best consistently and MLAR follows with LAR in third place in most scenarios followed by AODV. The main exception is like 2D, LAR does worse than AODV only in the low traffic scenarios. Again we believe this is due to the fact that routes are rarely being reused and old routes are not being expired. AOMDV and MLAR perform better than their single path counterparts due to the use of multiple paths in their route tables.

![Figure 5 - 30 Packet Delivery Ratio vs. Node Mobility in a Low Traffic Scenario](image-url)
5.4.2 Average End to End Delay in 3D

We present the delay characteristics for a high mobility (50 m/s with zero pause time) and high traffic scenario in Figure 5-32. We find that at very high mobility scenarios MLAR is delivering more packets than LAR but several of these packets are getting delayed significantly at intermediate nodes. We thus conclude that as mobility increases the delay characteristics of MLAR degrades significantly and we are looking into ways to reduce this effect since packets that arrive too late may not be of much use to many
applications. The underlying cause of the delay is unclear currently, but we expect that MLAR could be modified to have similar delay characteristics as AOMDV.

![Figure 5 - 33 Average End to End Delay in a High Traffic and High Mobility Scenario](image)

### 5.4.3 Control Overhead in 3D

As mentioned in the section on control overhead in 2D, during our simulations we kept several counters to measure the total number of control packets such as route requests, route replies, route errors, number of packets flooded, etc. Once again, we observe in Table 5-3 that AOMDV has 10% more RREQs sent than AODV and about 4 times as many RREQ packets sent overall as seen consistently in most scenarios. The numbers do not vary significantly with changes in mobility from 5 m/s all the way to 50 m/s. Consider the following data from a low traffic low mobility scenario.

<table>
<thead>
<tr>
<th></th>
<th>Flooded RREQ sent</th>
<th>RREP sent</th>
<th>Route Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>67364</td>
<td>3585</td>
<td>70949</td>
</tr>
<tr>
<td>AOMDV</td>
<td>68965</td>
<td>10110</td>
<td>79075</td>
</tr>
</tbody>
</table>

**Table 5 - 3 Control Packets Sent for a low mobility and low traffic scenario in 3D**
When we measure the number of RREQ packets that were received since they were flooded we get a better idea about the benefits of our approach and its scalability. We had not measured the number of RREQs and RREPs sent for LAR and MLAR but we did measure the number of control packets received and they are less than AODV or AOMDV by a factor of 10 for the same scenario. The following numbers in Figure 5-34 which compare all four protocols in terms of control packets received at all nodes as extracted from the trace files do not include error packets in any protocol or Hello packets in AODV and AOMDV which would effectively increase the difference in favor of LAR and MLAR in term of control overhead and bandwidth usage.

![Figure 5 - 34 Control Packets Received for a Low Mobility and Low Traffic Scenario in 3D](image-url)
We also measured the number of error packets generated in LAR and MLAR and found for the scenario above, the average number of error packets for LAR was 1103 and for MLAR it was 1346. The difference between these numbers and the ones presented in the high mobility case earlier are simply because in high mobility, more routes are likely to fail. The higher number of error packets for MLAR can be interpreted as being due to the number of times MLAR may have tried an alternate path that was stale and thus produced an additional error packet. Another point to be noted is that out of the 61624 RREQ packets received in MLAR, 33257 of them were not flooded throughout the network but only traveled one hop from the source. The lower number of RREQs received for LAR and MLAR also can be attributed to the use of scoped flooding since several of the RREQ packets were not retransmitted by nodes which knew they were not in the expected and defined request zone and dropped the packets whereas AODV and AOMDV would simply retransmit the packet to all available neighbors. While we do observe that the number of control packets generated by AODV and AOMDV is significantly higher, the effect on bandwidth used is slightly lower since the size of LAR and MLAR control packets are slightly bigger than those for AODV and AOMDV since they include a few additional bytes of information to store the entire source route in each

<table>
<thead>
<tr>
<th></th>
<th>Flooded RREQ received</th>
<th>RREP received</th>
<th>Route Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>57456</td>
<td>6585</td>
<td>581161</td>
</tr>
<tr>
<td>AOMDV</td>
<td>607368</td>
<td>26284</td>
<td>633652</td>
</tr>
<tr>
<td>LAR 3D</td>
<td>45219</td>
<td>5828</td>
<td>51047</td>
</tr>
<tr>
<td>MLAR 3D</td>
<td>61624</td>
<td>10275</td>
<td>71899</td>
</tr>
</tbody>
</table>

Table 5-4 Control Packets Received for a Low Mobility and Low Traffic Scenario in 3D
header. The additional number of bytes appended for the source route depends on the
length of the route which is variable from packet to packet.

We thus conclude that the control overhead for AODV and AOMDV is at least five to ten
times higher than LAR or MLAR in terms of total number of packets generated, flooded
and received by the nodes in the network as observed from our simulations in both 2D as
well as 3D.
Chapter 6: Related Work

This section discusses several protocols which are used in the study or related to those used. The reader is directed to the referenced publications for more in depth information about these protocols. The survey papers referenced [s1], [s2] earlier explain most of these protocols very lucidly and provide broad comparison and the individual papers provide greater depth. We are not aware of other studies that empirically compare LAR with AODV or AOMDV as directly as we do on the same common simulator platform with identical traffic and mobility patterns, but there are earlier studies of AODV vs. AOMDV and LAR vs. DSR. Our MLAR approach is similar to Backup Routing other which is an alternate multipath routing variant of DSR with a few major implementation differences and uses the link disjoint method of keeping alternate paths independent of each. The original DSR paper mentions the possibility of using alternate routes in the cache but didn’t explore it. LAR in turn is basically a source routing variant of DSR that uses geographic information to limit the number of control packets generated and propagated.

6.1 DSR

The Dynamic Source Routing (DSR) protocol [DSR] is an on-demand source routing protocol which has Route Discovery and Route Maintenance phases. Each mobile host
participates by maintaining a route cache for source routes that it has learned. When one host wants a route to the destination but no such information is available in its route cache, it will initiate a route discovery by flooding a route request (RREQ) packet throughout the network. A route record will be encapsulated in the header of each route request packet in which the specific sequence of hops that the packet passes through are recorded. Any intermediate node contributes to the route discovery by appending its own address to the route record. Once route request packet reaches the destination, a route reply (RREP) packet will simply reverse the route in the route record from the route request packet and traverse back upstream through this route. Route maintenance procedure monitors the operation of the routes and when a routing failure is encountered i.e. a node fails to deliver data packets to next hop, a route error packet will be sent back to the source. The route error packet contains the addresses of the hosts at both ends of the hop in error and when it is traversing back, all routes in the route caches of all intermediate nodes containing the failed link will be removed from the caches and a new route discovery is initiated if the route is still needed.

DSR is resistant to the presence of routing loops by using source routing. Upon receiving a route request packet, any intermediate node may detect a loop by comparing its own address with the sequence hop list in the header of the packet. A route reply can be sent back early to stop flooding of query message if a fresh route to the destination exists in the route cache of any intermediate node. Also, routes to the destination can be learned and recorded by intermediate nodes while relaying the route reply packets as well as observing the paths of other packets that pass through that node.
6.2 LAR

Location Aided Routing [LAR] by Ko and Vaidya was one of the first protocols to describe how location information could be used to reduce the routing overhead in wireless ad hoc networks. It is very similar in operation to DSR. In order to initiate a new route, DSR would have flooded the entire network with a route request. LAR instead floods packets in specified request zone or forwarding zone. The authors presented two location-aided routing (LAR) protocols which are often referred to as LAR Scheme 1 or LAR Box and LAR Scheme 2 or LAR Step. These protocols limit the search for a route to the so-called request zone, determined based on the expected location of the destination node at the time of route discovery. The request zone for Box is explained in detail in chapter 3 on GRID. In the Step mode we forward route requests only to nodes closer to the target than itself.

Figure 6 - 1 LAR Box Method
(Reproduced from [LAR], copyright IEEE 2000)
LAR in effect in a source routing protocol based on DSR which used geographic information to use a limited request zone in order to reduce the total number of control packets blindly flooded. Every packet sent in LAR contains information on the position of the last hop and its measured velocity. Simulation results indicated that using location information resulted in significantly lower routing overhead, as compared to an algorithm that does not use location information like DSR mentioned above, due to reduced flooding. GRID and thus Hyper-GRID in turn also use LAR. LAR was implemented in ns-2 by Camp et al [Camp] and we have utilized their code extensively in building our 3D multipath location aware protocol called MLAR. Adoption of LAR in the real world is obviously limited by availability of GPS information which is needed for nodes to be aware of their own position as well the availability of Geographic Location Servers (GLS) which will enable sources to know the position of the destination. We did most of our testing in the Step mode and found results using the Box mode to be very similar.
One interesting point to note is that in 3D the step mode can come across a void region where a greedy step hop is not possible. Here the recovery is simply to flood the RREQ from the source when the timeout for receiving a RREP expires.

### 6.2.1 Cache Effects in LAR and DSR

Many authors have noted that both LAR and DSR being source routing protocols are susceptible to stale caches due to frequent topology changes. Marina and Das [MARINA] proposed three techniques to improve cache correctness namely wider error notification, route expiration mechanism with adaptive timeout selection and the use of negative caches. In simulation results they found that application of all three methods individually as well as simultaneously provided significant improvements over DSR.

### 6.3 AODV

Ad hoc On demand Distance Vector Routing [AODV] introduced by Perkins and Royer in 1999 is an on-demand, reactive routing protocol and thus builds routes only when nodes require them. AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source
node in their route tables. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes. Since the protocol functions independent of any position information it should function in any 3D scenario as efficiently as it would in 2D.

When a source needs a route to a destination, it initiates a route discovery process by flooding route request (RREQ) packets throughout the network which search for a path to the destination. The RREQ packet can be uniquely identified by a sequence number so that duplicate RREQs can be recognized and discarded. Upon receiving non-duplicate RREQ, an intermediate node records the previous hop and checks whether there is a valid and fresh route entry to the destination existing in its own local route table. If this is the case, the node sends back route reply (RREP) to the source, otherwise it rebroadcasts the RREQ. As the RREP traverses though the route selected, each node along the path sets up a forward pointer, updates corresponding timeout information and records the latest destination sequence number (for checking the freshness of the route).

For the path maintenance part, disconnection is detected by periodic exchange of hello messages. When a route failure is detected, route error (RERR) packet is sent back to all sources to erase route entries using the failed link. A route discovery procedure is initiated if the route is still needed. AODV is often considered to be the benchmark by which other ad hoc routing protocols are measured.
6.4 GPSR

Greedy Perimeter Stateless Perimeter Routing [GPSR] is a routing protocol that uses the positions of routers and a packets source and destination to make routing decisions. GPSR uses greedy forwarding to forward packets to nodes that are always progressively closer to the destination. In regions of the network where such a greedy path does not exist (i.e., the only path requires that one move temporarily farther away from the destination), GPSR recovers by forwarding in perimeter mode, in which a packet traverses successively closer faces of a planar sub graph of the full radio network connectivity graph, until reaching a node closer to the destination, where greedy forwarding resumes. GPSR is considered to be a very efficient algorithm because it requires very little control information apart from nodal position information and very little state information to be stored at each node. It does require the implementation of a Geographic Location Service, however, and it is not clear how the perimeter mode would function equivalently in 3D hence our desire to develop a 3D geographic routing protocol.

The authors ran GPSR and DSR in simulations in an older version of ns-2 and compared the results. In general, in terms of packet delivery success rate, overhead, and hop-count over shortest path, GPSR performs significantly better than DSR. GPSR keeps state proportional to the number of its neighbors while both traffic sources and intermediate
DSR routers cache state proportional to the product of the number of routes learned and route length in hops.

GPSR's benefits all stem from geographic routing's use of only immediate-neighbor information in making forwarding decisions. Routing protocols that rely on end-to-end state concerning the path between a forwarding router and a packet's destination, as do source routed, distance vector, and link state algorithms, face a scaling challenge as network diameter in hops and mobility increase because the product of these two factors determines the rate that end-to-end paths change. Hierarchy and caching have proven successful in scaling these algorithms. GPSR shows that geography can be effective as well.

**6.4.1 GPSR in 3D**

A robust 3D version of GPSR is still open for future work. The simplest solution is to use the greedy method till as far as possible and then use either flooding or a 3D version of the perimeter mode that temporarily violates the greedy principle. Flooding could be used till one of the nodes is found to be nearer to the destination than the current one and greedy forwarding resumes. There is no known approach to constructing a GG or RNG in three dimensions, although we propose a possible alternative below. Greedy mode in 3D works the exactly same way in 3D as it does in 2D. Calculating distances in 2D and 3D are equivalent.
Kosuke et al [Kosuke] suggest that on reaching a void region in 3D where no greedy choice is possible, make the least worst greedy choice and proceed from there. In our opinion this does not provide any guarantee that packets will be delivered, whereas GPSR in 2D guarantees that if a route exists it will keep switching between greedy and perimeter mode until it is found.

We decided to try and see if we could find a 3D equivalent to perimeter forwarding. The key to perimeter forwarding is the ability to see where an edge of a face crosses vector $x$-D where $x$ is the point at which a packet enters perimeter mode and D is the destination. We can use this point of intersection/cross-over to study our progress as we go from one face to the next one and thus see whether we are moving across the void/around it and closer to our destination.

In a 3D dead end situation at node $x$, we have node $x$, destination $D$ and any one neighbor of $x$ considered in the same plane. Which neighbor should one pick? There is no perfect choice here since all are further away from $D$ than $x$ (which is why we are in perimeter mode).

Let us assume we take a node $y$ (which is a neighbor to the node at the dead end) that is the least further away from $D$ and move to it, and mark the packet as entering perimeter mode. If we still need to use the perimeter for next hop, (i.e. no node or neighbor of $y$ besides the previously traversed hop is closer to $D$ than current node $y$) then we can proceed to next node $z$ which has the lowest projection distance. By projection distance I
mean drop a geometric perpendicular from z to the line formed by x-D. Assume the perpendicular intersects line x-D at point p. then see if distance p-D is less than previous similar distance for previous hop or not. We believe that this can help get us across the void and closer to the destination till we can go into greedy mode again.

Basically we are trying to exploit the fact that any 3 points we consider at a time will define their own plane. So for making a perimeter forwarding decision, the node making the forward packet/ or point of entry into perimeter mode, any one neighbor it may forward to and the destination will be in the same plane. If these are the only considerations for routing in perimeter mode, Euclidean distances can easily be calculated. If we need 4 or more points then there is no guarantee they will be on the same plane. However any 3 at a time can be used to form a surface/plane and the goal would be to move around the 3D void which is an intersection of 2 spheres. Again we do note that at present we cannot guarantee delivery of packets in the manner GPSR does or argue fully the correctness of this method. Further work is required in formalizing our approach and eventually testing it, possibly via simulation in ns-2.
**Figure 6 - 3 Perimeter Forwarding Example**

D is the destination; x is the node where the packet enters perimeter mode; forwarding hops are solid arrows [GPSR]

**Figure 6 - 4 A 2D Geometric Routing Void**

Node x's void with respect to destination D in 2D. [GPSR]
6.5 Multipath Dynamic Source Routing

Multipath Dynamic Source Routing (MDSR) protocol proposed by A. Nasipuri and S.R. Das [NAS1] is the multipath extension to DSR. The initial paper proposed the idea and some more extensive performance results using the MaRS simulator were provided in [NAS2]. The basic idea is that when multiple flooded query messages arrive at the destination, apart from replying the query with the shortest route i.e. the primary route, the destination will also compute those source routes that are link-wise disjoint from the primary route. Disjoint routes are chosen so that a link failure in one route does not affect the others. When a route failure occurs in the primary route, alternate route will be used until a new route discovery initiated when all routes break down. The authors explored two variants, one where the source gets multiple routes and another where all intermediate nodes on the primary route get multiple alternate routes.

First, alternate routes are only assigned to the source, then failure in intermediate link sends error packet back to use alternate routes causing a temporary loss of route for data packets. Improvement can be applied by equipping all intermediate nodes with a disjoint alternate route. Destinations need to replies to each intermediate node in the primary route with an alternate disjoint route to it. When an intermediate node encounters a transmission failure to the next hop, it may use alternate path to destination immediately instead of sending back error packet to source. Thus, only loss of both routes in a node generates an error packet back to the source. Intermediate node with alternate route to
destination will stop the error packet and modifies source route on all later data packets to
direct to its alternate route. The procedure continues until no alternate route along the
primary route is available at all, a route discovery initiated.

The advantage of MDSR like MLAR is that it provides alternate paths for all
intermediate nodes along the primary route.

The main disadvantage of MDSR is that this scheme will result in more route reply
message flooding in the network, overhead for intermediate nodes’ cache storing and
computation overhead for the destination, particularly for the computing of alternate path
of all the intermediate nodes.

The authors also found that multipath routing decreased the routing load but increased
end to end delay as alternate routes tend to be longer than primary routes in their analytic
results. They conclude that in a real network, a lower routing load would mean less
interference and potentially lower end to end delay as well. The authors also found that
the benefits of having more than 2 routes were minimal if any.

6.6 Split Multipath Routing

Split Multipath Routing (SMR) proposed by Lee and Gerla [SMR] is another disjoint
multipath protocol using source routing. SMR is similar to multipath DSR except that the
former uses a modified flooding algorithm and the data traffic is split among the multiple
paths simultaneously to balance the transmission throughout the network and avoid congestion. They also found empirically in their simulation work that two is the optimal number of disjoint routes for multipath routing.

### 6.6.1 Route Discovery

During the route discovery phase, RREQ are flooded on demand and duplicate packets through different routes containing entire path of the route reach the destination. Based on the shortest path chosen, destination computed out disjoint routes and RREP packets are sent back via them. Different from the protocols mentioned before, intermediate nodes are not allowed to send RREQ back, otherwise the RREQ cant reach destination and disjoint routes are not available. Instead of dropping duplicate RREQs which mostly generates overlapped paths, intermediate nodes forward the duplicate copies from different incoming links to destination and two routes (one is shortest delay route) that are maximally disjoint can be chosen.

### 6.6.2 Route Maintenance

During route maintenance phase, RERR packet containing route to the source and nodes of the broken link will be sent back. The source removes every entry in the table using this disconnection hop and uses the remaining route to deliver data packet. When the
source is informed of a route disconnection, it may use one of the two policies in rediscovering routes:

**SMR-1**: Initiates the route recovery process when any route of the session is broken.

**SMR-2**: Initiates the route recovery process only when both routes of the session are broken.

When RREP for the first discovered route received, source uses it to deliver data packet in the buffer. Arrival of later RREP will cause source to split traffic transmitting on both routes. **SMR-2** scheme was found to be more efficient than **SMR-1** and both performed better than single path DSR in their simulations. However the extra re-sequencing burden will be placed on destination, as a result of the out of order delivery caused by distributed traffic transmission. The defect can be made up by applying simple reordering buffers in hosts.

### 6.7 AOMDV

Ad hoc On Demand Multipath Distance Vector Routing [AOMDV] proposes multipath extensions to the routing protocol AODV. The protocol computes multiple loop-free and link-disjoint and node-disjoint paths. The authors state that performance comparison of AOMDV with AODV using ns-2 simulations shows that “AOMDV is able to effectively cope with mobility-induced route failures. In particular, it reduces the packet loss by up to 40% and achieves a remarkable improvement in the end-to-end delay, often more than
a factor of two. AOMDV also reduces routing overhead by about 30% by reducing the frequency of route discovery operations.” We received ns-2 code from the author Mahesh Marina as well as a newer version from Rachit Chawla who had ported the code to a newer version of ns-2 which we used for our studies. AOMDV uses a unique way of implementing multipath routing and ensuring that routes are loop free.

Each hop incrementally decides if the previous hops create a loop free path via a distributed algorithm without the use of source routing. Routing decisions are made in a hop by hop manner. Disjoint paths have the desirable property that they are more likely to fail independently. Thus they have a better utility. There are two types of disjoint paths as mentioned earlier: node disjoint and link disjoint. Node disjoint paths do not have any nodes in common, except for the source and the destination. In contrast, link disjoint paths do not have any common links, but may have common nodes.

6.7.1 Path Discovery

For the route discovery phase, it is quite the same as which is in the AODV. And only some minute changes needed here. To guarantee loop freedom, multiple next-hop routes are accepted and maintained as obtained by multiple route advertisements, but the protocol only allows accepting alternate routes with lower hop-counts. To guarantee link-disjointedness, several changes are needed.
At the intermediate nodes, duplicate copies of RREQ are not immediately discarded. Each copy is examined to see if it provides a new node-disjoint path to the source. If it does provide a new path, the AOMDV route update rule is invoked to check if a reverse path can be set up. At the destination, to get link-disjoint paths, the destination node adopts a “looser” reply policy. It replies up to k copies of RREQ.

6.7.2 Path Maintenance

Route maintenance is almost exactly the same as AODV. Periodic Hello messages help keep local one hop table entries fresh and updated. The only difference with respect to AODV is that only when all the routes fail a new route discovery is initiated if the route is still needed.

6.8 MESH

Mesh is a scheme proposed in [MESH], which like MLAR uses geographic information to reduce the blind flooding of control packets. It uses a combination of a multi eye strategy to confine the route discovery region for route request and route reply packets and a special spiral hopping multipath strategy to provide online route maintenance. Their simulation results using their own Java simulator indicate modest improvements over LAR and DSR on average in terms of packet delivery in a reachable graph. With respect to LAR, it reduces the flooding region by a small amount through use of
intermediate eye nodes to decide smaller regions of interest for flooding packets relative to LAR. This could lead to the risk of not finding all disjoint alternate paths which could potentially be outside the smaller scope in consideration.

However a large number of control beacons need to be flooded throughout the network periodically and all destinations need to flood EyeInfo control packets. The eye nodes are nodes which are receiving information about the location or reach ability of other nodes and know their own position and velocity. They periodically rebroadcast the information they collect and in effect work as a simple GLS used to limit the scope of flooded route request packets. However it could be more efficient to have a query reply system instead of flooding position information or eyenode-destination connectivity information throughout the network.

### 6.9 CHAMP

CHAMP stand for Caching and Multipath Routing protocol [CHAMP]. CHAMP uses simultaneous multi-path routing along with data packet caching to provide an energy efficient and robust protocol. CHAMP allows nodes to cache data packets that they sent recently. Thus whenever an error message is broadcast for a broken route to a destination, an upstream node which has a cached copy of the data packet that failed can re transmit it with a new route if it has an alternate route in its own routing table. When forwarding data packets, each node forwards the packet to the least used next hop neighbor. This
spreads packets over all routes in round robin fashion and helps to decongest routes that
may get overloaded otherwise. Using such a multipath technique is certain to lead to out
of order receipt of packets at the destination. In simulation results published, CHAMP
performs significantly better (by as much as 30%) than AODV and DSR in terms of
packet delivery, routing overhead and energy efficiency but the authors do note that
further validation was needed to verify the protocols scalability and performance in low
mobility scenarios as well as the large number of out of order delivered packets.
Chapter 7: Future Work

We have demonstrated the effectiveness and benefits of using MLAR in lieu of LAR in terms of increased delivery ratio for most traffic and mobility scenarios in both 2D and 3D without affecting the original scalability of LAR. In terms of future work, we would definitely consider implementing GRID using MLAR and thus create a working implementation of Hyper-GRID for further evaluation via simulation. It is worth trying to optimize the timeout values and other parameters used in MLAR to see if other pre-assigned or adaptive values are more suitable for particular scenarios. It may be interesting to evaluate the performance of an MLAR version that uses simultaneous paths (as in Split Multipath Routing). The performance of MLAR could potentially benefit by implementing some of the techniques suggested to improve the performance of DSR, such as having routes expire periodically or the use of negative caches.

The mobility model we have used to test MLAR does not correspond directly with those we described in Chapter 1 as our potential real world applications. Thus there is scope for the creation of more realistic 3D mobility models and their evaluation via simulation as well as experimentation with a wider range of traffic patterns, node densities, and transmission ranges.

Some of the open research questions are how nodes can actually find out their own location information in different scenarios such as sensors in the oceans or within a room and to what limits of accuracy can they do so. It may be interesting to analyze how many
link failures are due to the physical partitioning of the network and how to best overcome such situations without introducing too much delay. Further analysis is necessary to understand the tradeoffs of different approaches towards link disjointedness, such as using link, node or spatial path separation.

We would also like to formalize our solution for implementing 3D GPSR and evaluate it via simulation and compare its performance directly against MLAR and Hyper GRID in ns-2 in 3D for identical mobility and traffic scenarios.
Chapter 8: Conclusion

Our initial goal was to build a position based routing algorithm that could route packets in a scalable and effective manner in three dimensions. By applying an alternate path caching strategy (using link disjoint paths) to the original LAR protocol and extending it to work in 3D we have effectively realized this goal through the development of MLAR, a multipath version of LAR. We have directly compared the performance of four routing algorithms: AODV, AOMDV, LAR and MLAR, side by side in both 2D and 3D on a common simulation platform under a range of mobility and traffic conditions. We have demonstrated clearly the significant benefits of MLAR over LAR as well as AODV in terms of routing performance in both 2D and 3D. Only AOMDV consistently performs better than MLAR in terms of overall packet delivery at the cost of higher bandwidth usage due to significantly higher control overheads. We have also described our initial attempts to implement GPSR in 3D. We also demonstrated how to extend ns-2 to implement and test 3D ad hoc routing protocols.
References:

[S1] Position Based Routing Algorithms For Ad Hoc Networks: A Taxonomy (2002), Silvia Giordano, Ljubica Blazevic and Ivan Stojmenovic


[LAR] Location-Aided Routing (LAR) in Mobile Ad Hoc Networks (1998), Young-Bae Ko and Nitin Vaidya.


[Kosuke] The proposal of the seamless location aware services and geographical routing protocols (2001), Kosuke Yamazaki and Kaoru Sezaki


[APS] Ad Hoc Positioning System (APS), Dragos Niculescu and Badri Nath


[AODV] Ad-hoc on-demand distance vector routing, C. Perkins and E. Royer


[GEDIR] GEDIR: Loop-free location based routing in wireless networks, I. Stojmenovic and X. Tin

[Compass Routing] Compass Routing on Geometric Networks (1999), Evangelos Kranakis, Harvinder Singh and Jorge Urrutia


