DESIGN, IMPLEMENTATION AND SIMULATION OF A UNICAST ROUTING PROTOCOL IN MOBILE ADHOC NETWORKS (MANETS)

By

Aisha Khalid Khan

Roomana Inayat Malik

Submitted to Faculty of Computer Science, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of BE Degree in Computer Software Engineering

May 2005

ABSTRACT

DESIGN, IMPLEMENTATION AND
SIMULATION OF UNICAST ROUTING
PROTOCOL IN MOBILE ADHOC
NETWORKS (MANETS)

Mobility-**A**daptive **O**n demand **R**outing **P**rotocol (MAORP) is intended for use by mobile nodes in an ad hoc network. It offers quick adaptation to dynamic link conditions, low processing and memory overhead, low network utilization, and determines unicast routes to destinations within the ad hoc network. Current approaches to multi-path routing make use of pre-computed routes determined during route discovery. These solutions, however, may not work under conditions of high mobility since the alternate paths are not actively maintained. Hence, precisely when needed, the routes are often broken. In this protocol, a solution is presented for the stale and the broken links. The protocol uses multi-path routing and keeps the list of available routes at the source as well as the destination. Update packets are broadcast along the network every few seconds (pre-computed based on heuristics). These packets measure the sum of signal strengths along all the paths, summed over individual links. The current recorded paths by the source or the destination may then be dynamically changed according to the cumulative signal strength obtained along the complete paths. The path with the highest cumulative signal strength is then used for data transmission. The performance of the network was measured using Packet Delivery Ratio as the metric. MAORP was found to perform better than DSDV and comparable to AODV, under conditions of high-mobility, with 70 nodes observed as the bench mark beyond which a drastic change in the characteristics was observed.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of any other award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible

ACKKNOWLEDGMENTS

We wish to thank Almighty Allah who gave us the strength and determination to complete this project. We gratefully acknowledge the continuous guidance and motivation provided to us by our project president Lt. Col. Naveed Khattak. We would also like to thank Lecturer Ahmad Raza Shahid, without whose personal supervision, advice and help, timely completion of this project would have been impossible. Our very special thanks are extended to the Head of Department Lt Col. Raja Iqbal for his ever extended moral and technical support.

We deeply treasure the unparallel support and forbearance that we received from our friends for their critical reviews and useful suggestions that helped us in completion of this project. We are also deeply indebted to our families for their never ending patience and support for our mental peace and to our parents for the strength that they gave us through their prayers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

1 Introduction

1.1 Overview

Mobile ad hoc networks (MANETs) are infrastructure-less wireless networks of mobile nodes that communicate with each other on a peer to peer basis. 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 have been designed, and many of them also implemented, to make routing decisions at each node.

Major issues in MANETs have always been high mobility resulting in frequent link breakages, packet drops and dynamic topology changes, low channel bandwidth and limited battery power. Maximum throughput has always been the primary goal that every routing protocol aims to achieve. MAORP works on the achievement of the same goal. MAORP protocol enables dynamic, self starting, multi-path routing between participating mobile nodes wishing to establish and maintain an ad-hoc network. It allows paths that have unique nodes (for a given destination, source pair) to be built so that no node appears in more than one path other than the source and the destination.

Thus each path is independent from the other one. One node appearing in only one path frees a node from broadcasting of packets and keeping track of its neighbor nodes or upstream and downstream nodes. As a result very little routing information is stored at each intermediate node. Using the signal strength along any link as the criteria for its selection as well as the cumulative signal strength as the path selection criteria means that at any time the path selected for the data transmission is the one that is most stable and has the minimum chance of breaking. As a result the packet delivery ratio is expected to increase. The control overhead is also expected to reduce owing to the periodic sending of update packets which carry only small information instead of sending large control packets. Also at anytime if a better route is detected by the update packets that is not listed in the list of alternate routes maintained by the source and the destination, that path would be added as the primary path and data would be delivered from that path onwards. If the destination detects that it has not been receiving data packets at the rate it had been receiving earlier it sends an alert packet to the source from the second alternate route in the list as well as information of how many data packets it has received so far. This enables the broken routes or any intermediate node failure to be detected quickly. Also since all the paths are independent of one another therefore they fail independently of each other. If the destination does not receive any packet from the source in the timeout period it would believe that the source has migrated and would delete the alternate route table for that source. The destination would then be free to accept request from any other source on the network.

Algorithms are intended to be validated through simulation using NS-2 [1] and their efficiency, scalability and other performance related properties are to be studied.

Implemented in this thesis is Mobility Adaptive On-demand Routing Protocol (MAORP), a routing protocol that uses the signal strength along any link as the criteria for its selection as well as the cumulative signal strength as the path selection criteria, ensuring that at any time the path selected for the data transmission is the one that is most stable and has the minimum chance of breaking.

Some of the primary contributions of the work are (a) Design of MAORP, (b) Coding of the algorithm in NS-2 version 2.27, (c) Simulations using a wide variety of mobility scenarios and traffic patterns and (d) Detailed comparison of MAORP, AODV [2] and DSDV [3].

1.2 Wireless Networks

In the last years, the popularity of wireless networks increased. Wireless networks offer many advantages in the form of availability and mobility to the users. They are built in environments where the installation of wires is not possible or not wished. They require less infrastructure than wired networks and can be set up faster, e.g., in an emergency mission. Furthermore, they provide mobility to the users by freeing them of dangling cables. There exist three types of wireless networks (a) mobile ad-hoc networks, (b) cellular networks and (c) multi-hop cellular networks.

1.2.1 Mobile Ad-hoc Networks

A mobile ad-hoc network (MANET), as in Figure 1-1, consists of a collection of mobile nodes which have the possibility to connect to a wireless medium and form a dynamic network with wireless links.

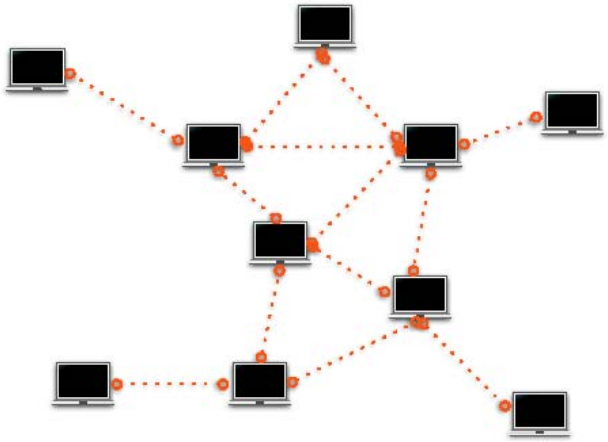

Figure 1-1: Mobile Ad-hoc Network

Since the nodes are mobile, the links between them are not permanent. The network topology may change rapidly and unpredictably in time. New nodes can join the network, and other nodes may leave the network. The expected size of a MANET is larger than the transmission range of the nodes, because of this fact it is necessary to route the traffic through a multi-hop path for giving the nodes the ability to communicate with each other. There exist neither fixed routers nor fixed locations for the routers nor centralized administration. The lack of any fixed infrastructure is compensated by the routing ability of every mobile node. They all act as mobile routers and for this they need the capability to discover and maintain routes to every node in the network and to route the packets accordingly.

Possible applications of MANET are in scenarios with little or no communication infrastructure such as emergency relief, military operations, or situation where people wish to simply share information, e.g., at a conference.

1.2.2 Cellular Networks

Cellular networks or infrastructure networks are based on a wired back-bone which connects the base-stations. The base-station nodes have at least one network interface for the wired network and one or more wireless network interfaces to provide communication to the mobile nodes. A pictorial demonstration can be seen in Figure 1-2.

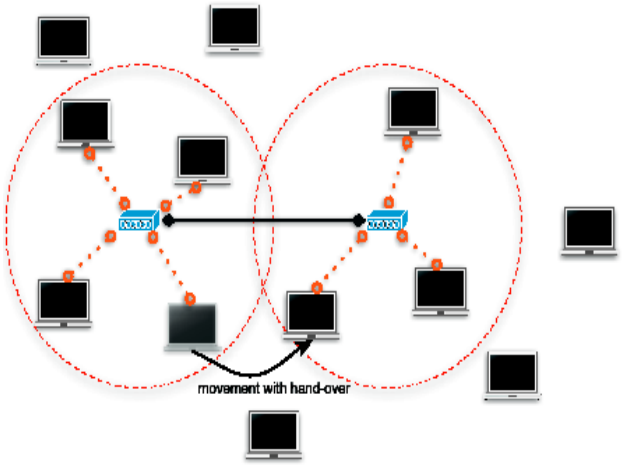

Figure 1-2: Cellular Network

The communication of the mobile node is only possible over a one-hop link to base-station. Direct links between nodes or multi-hop links to the base-station are not possible. The size of a cellular network is limited by the transmission range of the base-stations. If the node is out of the transmission range of the base-stations, no communication is possible. Inside the area covered by the base-stations it may move without losing connection and if it leaves the transmission range of the current base-station, a hand-over to a another base-station will let the node communicate seamlessly.

1.2.3 Multi-hop Cellular Network

In multi-hop cellular networks the two concepts described before are combined. On one hand there is a cellular network; on the other hand there are mobile nodes with additional routing facilities.

With this approach it is possible to have multiple hops between a mobile node and a base-station. The idea is to benefit from existing infrastructure and to gain more efficiency out of it, to cover wider areas with less fixed antennas and base-stations and to reduce power consumption due to shorter hop distances. This can be seen in Figure 1-3.

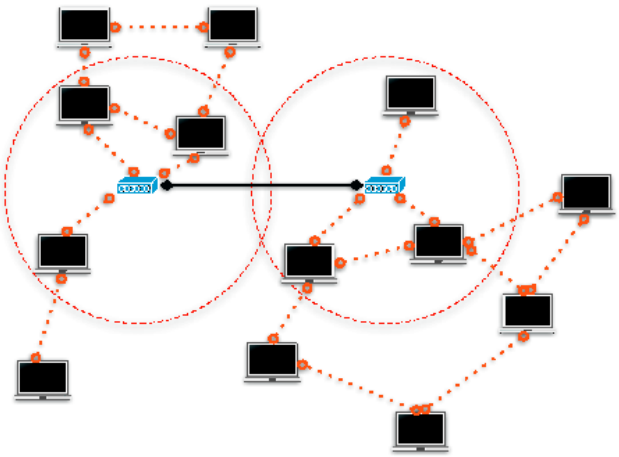

Figure 1-3: Multi-hop Cellular Network

1.3 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. 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.

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 nodes 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; (a) They can be deployed easily in several situations (nodes could possibly drop into place by hand or by an airplane), (b) They can be deployed quickly and hopefully with economies of scale, cheaply as well and (c) They can lead to decreased dependence on prior or fixed infrastructure or provide alternative infrastructure in areas where current infrastructures 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.

1.4 Wireless Routing Protocol

Several unicast routing protocols have been developed for MANETs that have their own unique characteristic strengths and weaknesses. A routing protocol for a mobile ad-hoc environment is in urgent need of (a) Loop-freeness, (b) Multi-hop paths, (c) Self-starting, (d) Dynamic maintenance of the network topology, (e) Fast convergence, (f) Minimal Routing overhead, (g) Economical consumption of resources, e.g., memory and bandwidth, (h) Minimized and local effect of link breakage and (i) Scalability with large numbers of nodes

Different concepts for mobile ad-hoc routing are established in order to achieve the above capabilities. There are different interdependences between the wished capabilities. 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. Because of this, there exists no concept that is optimal in all aspects. Each approach has to make a compromise on the different capabilities. A detailed description of all these protocols is beyond the scope of this thesis. Described here in detail, however, are all protocols that are relevant to the work.

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 congestion. 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.

1.5 Classification of Routing Protocols

1.5.1 Position Based Protocols

This concept makes use of location information. The routing can be based on the location information either to flood route requests or to forward the data packets. The basic components of position based routing are, (a) positioning service to determine the physical position of the node, e.g., Global Positioning System (GPS), (b) location service to determine the position of the destination, e.g. DREAM [4] and (c) forwarding strategy, i.e. selection of the next node.

GPSR [5], GRID [6] and LAR [7] 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.

1.5.2 Topology Based or Non Position Based Protocols

The topology based protocols do not make use of additional location information. They utilize network topology information to make a routing decision.

In proactive or table-driven protocols, the nodes in the network maintain a table of routes to every destination. They periodically exchange messages to keep the routing

table up-to-date. At all times the routes to all destinations are ready to use. The maintenance of routes to all destinations, even if they are not used, consumes a lot of bandwidth and network resources. It can even end in increasing delays because of queues filled up with control packets and more packet collisions due to more network traffic. As a result, proactive protocols do not scale in the frequency of topology change. Therefore they are only appropriate for low mobility networks.

Representatives of proactive protocols are Destination-Sequenced Distance Vector Routing (DSDV) and Optimized Link State Routing (OLSR) [8].

Reactive (or on-demand) protocols acquire only routing information upon request. They are designed to overcome the wasted effort in maintaining unused routes. Routes are searched on demand. When a node requires a new route to a destination, it starts a route discovery process. This process ends once a valid route is found or all possible routes are checked. The nodes are not forced to maintain unused routes, on the other hand the latency for sending data packets will considerably increase. A long delay before data transmission can arise because the transmission has to wait until a valid route to the destination is acquired. As reactive routing protocols flood the network to discover the wished route, they are not optimal in terms of bandwidth utilization, but scale well in highly dynamic networks. Thus this strategy is suitable for high mobility networks.

Exponents of this strategy are Temporally-Ordered Routing Algorithm (TORA) [9], Dynamic Source Routing (DSR) [10] and AODV.

1.6 Problem to be Solved

In contrast to wired networks, routing in mobile ad hoc networks is challenged by a complicated interaction of three fundamental difficulties. First is contention. The nature

of mobile computing devices demands wireless communication. The nature of wireless communication results in significant contention for the shared medium (the wireless channel). Second is congestion. Another aspect of wireless communication is decreased bandwidth which results in much higher congestion when compared to a similar wired network configuration. The links between wireless nodes can support less data traffic than is attainable with wired connections. Finally, and most importantly, is the unique set of challenges created by mobility. Node mobility in MANETs makes communication links break and these breaks may occur at a rapid rate. This changing network topology is the key challenge that MANET routing protocols must overcome. Several existing MANET routing protocols have been proposed that deal with this mobility problem in different ways. These protocols and their mechanisms are described in Chapter 3. Any attempt to provide effective routing mechanisms in MANETs must deal with the changing network topology created by mobility.

In addition to mobility, contention, and congestion, MANET protocols must deal with other significant issues. Mobile computing devices are often battery powered and therefore have limited power and lifetime. They may also be constrained by limited memory or processing capabilities. These additional factors combine with the above three key challenges to make routing in Mobile Ad Hoc Networks extremely difficult.

Our research goals are aimed at improving the effectiveness and scalability of routing in MANETs, especially in VERY demanding network mobility conditions. More specifically, this research enables MANET routing protocols to adapt their operation based on the current network mobility conditions present. The use of signal strength information, when incorporated into adaptive protocols, promises dramatic improvements.

1.7 Problem Statement

"To design, implement and simulate a uni-cast routing protocol for mobile ad-hoc networks that achieves a good packet delivery ratio (PDR) in any network topology, and using the results for analysis with other standard routing protocols for a regressive assessment of the work."

1.8 Project Area and Motivation

Development of a uni-cast routing protocol means working at the network layer which is the third layer of Open System Interconnection (OSI) model. Any new idea that is generated is first simulated to see how much it is successful. Where it can be improved and what should be changed to make it better. Mobile ad-hoc networks are currently in the research phase. New ideas are developed and then simulations are performed to check out the performance.

The aim was to do something innovative, generate new ideas and test them in a field that in the coming years would revolutionize the world. Seeing the importance of simulations and amount of learning one gets from using good simulators provided the motivation to make a new routing protocol and then simulate it using the most famous simulator used in networking which is 'the network simulator NS-2'. Making a new routing protocol from scratch meant first of all a thorough study of mobile ad-hoc networks, the various routing protocols developed for it; their merits and demerits, to add to studying NS-2 in detail itself. This added to great and advanced level of programming required in the making of a protocol seemed an excellent opportunity for using the knowledge gained so far and learning many new things that would prove to be very useful in the future.

Chapter 2

2 Research Objectives

2.1 Project Goals and Objectives

The goals/ objectives of the project were: (a) design a uni-cast routing protocol for MANETs which adjusts to the changing topology of the network, is scalable and has a low routing overhead, (b) implement it in C++, (c) simulate it in the network simulator (NS-2) and (d) compare its performance with other standard uni-cast routing protocols.

2.2 Deliverables

The deliverables of the project are: (a) compiled C++ code of MAORP, (b) mobility and traffic scenarios in OTcl, (c) awk script and (d) trace graph.

3 Literature Review

This section discusses the various protocols that have been studied as part of the research done.

3.1 AODV

Ad hoc on demand Distance Vector Routing 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.

When a source needs a route to a destination, it initiates a route discovery process as in Figure 3-1 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), as depicted by Figure 3-1, 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).

Figure 3-1: Route Request

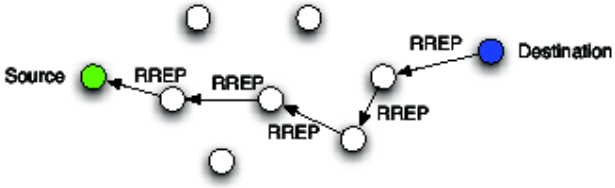

Figure 3-2: Route Reply

For the path maintenance part, disconnection is detected by periodic exchange of hello messages, Figure. When a route failure is detected, route error (RERR) packet, as shown in Figure 3-3, 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.

Figure 3-3: Periodic HELLO Messages

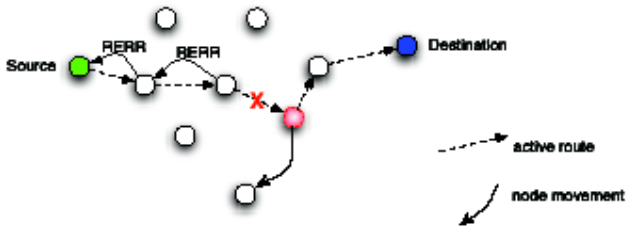

Figure 3-4: Router Error

3.2 Ad-hoc On Demand Distance Vector with Backup Routing (AODV-BR)

In AODV-BR [11], the authors propose a scheme to calculate alternate paths, Figure 3-5, such that when a link failure occurs, the intermediate node searches for an alternate path to circumvent the broken link. The basic assumption made in this protocol is that all the nodes are in promiscuous mode and that they can overhear every transmission within their range.

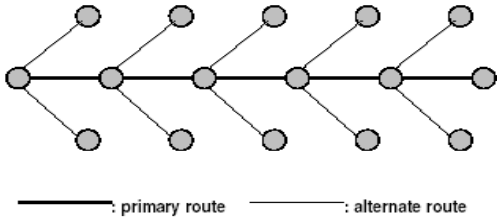

Figure 3-5: Multiple Routes Forming a Fish Bone Structure

This protocol, however, has a number of limitations. First, it assumes that several nodes are within transmission range of each other. Also, constant mobility of the nodes is not

taken into account. The protocol assumes that a node that offers the alternate route around a broken link does not move away and remains within range of the two nodes between whom the link has broken. Moreover, the utilization of promiscuous mode greatly increases the power consumption of each node. It can also be considered as a multi-path routing scheme as multiple routes, depicted by Figure 3-6 are constructed.

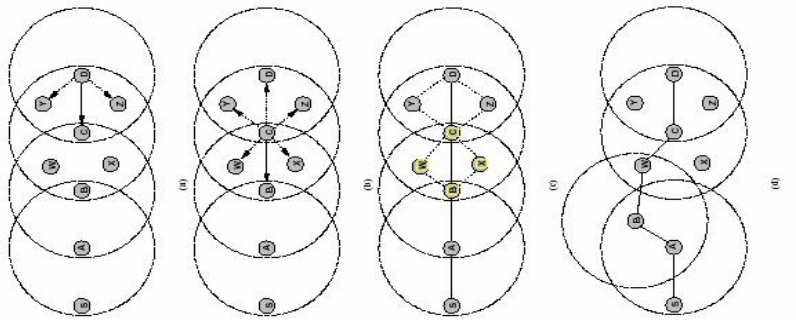

Figure 3-6: Multiple Routing Construction and its Use

3.3 Dynamic Source Routing (DSR)

DSR [9] 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.

3.4 Multi-path Dynamic Source Routing (MDSR)

MDSR [12] protocol proposed by A. Nasipuri and S.R. Das is the multi-path extension to DSR. The basic idea is that when multiple flooded query messages arrive at the destination, apart from replying the query with the shortest route (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 MAORP 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 multi-path 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.

3.5 Split Multi-path Routing (SMR)

SMR [13] proposed by Lee and Gerla is another disjoint multi-path protocol using source routing. SMR is similar to multi-path 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 multi-path routing.

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, Figure 3-7. Based on the shortest path chosen, destination computes disjoint routes and RREP packets are sent back via them.

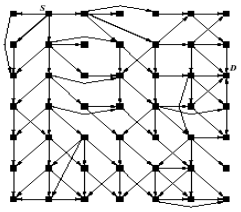

Figure 3-7: Request Propagation

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.

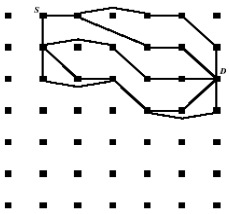

Figure 3-8: Available Paths

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: (a) SMR-1, which initiates the route recovery process when any route of the session is broken, and (b) SMR-2, which initiates the route recovery process only when both routes of the session are broken.

When RREP for the first discovered route is 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.

3.6 Ad-hoc On Demand Multi-path Distance Vector Routing (AOMDV)

AOMDV [14] proposes multi-path 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.” AOMDV uses a unique way of implementing multi-path 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.

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.

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.

3.7 Caching and Multi-path Routing Protocol

CHAMP [15] 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 multi-path 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.

3.8 Temporary Ordered Routing Algorithm (TORA)

TORA is a distributed loop-free routing protocol that is based on diffusing computations. Here, multiple routes are computed mainly to alleviate congestion on links. TORA, however, requires reliable, in-order delivery of control messages.

Information on a per link basis, whereas, our solution uses the signal strength information accumulated over an entire path.

3.9 Signal Stability Based (SSA)

SSA [16] routing protocol is a routing protocol that also selects paths using the signal strength metric. However, the signal strength information is utilized in a different way than what is presented in this protocol. The signal strength criteria in SSA allow the protocol to differentiate between strong and weak channels. SSA is a distributed protocol that uses the signal information on a per link basis, whereas, the solution presented here uses the signal strength information accumulated over an entire path.

4 Design

4.1 Assumptions

The protocol as described here is designed mainly for mobile ad hoc networks of up to about one hundred nodes, and is designed to work well with even very high rates of mobility. It is assumed in this document that all nodes wishing to communicate with other nodes within the ad hoc network are willing to participate fully in the protocols of the network. In particular, each node participating in the ad hoc network SHOULD also be willing to forward packets for other nodes in the network. The diameter of an ad hoc network is the minimum number of hops necessary for a packet to reach from any node located at one extreme edge of the ad hoc network to another node located at the opposite extreme. Packets may be lost or corrupted in transmission on the wireless network. It is assumed that a node receiving a corrupted packet can detect the error and discard the packet. Nodes within the ad hoc network MAY move at any time without notice, and MAY even move continuously, but it is assumed that the speed with which nodes move is moderate with respect to the packet transmission latency and wireless transmission range of the particular underlying network hardware in use. In particular, the protocol can support very rapid rates of arbitrary node mobility, but it is assumed that nodes do not continuously move so rapidly as to make the flooding of every individual data packet the only possible routing protocol. Wireless communication ability between any pair of nodes may at times not work equally well in both directions, due for example to differing antenna or propagation patterns or sources of interference around the two nodes. That is, wireless communications between each pair of nodes will in many cases be able to operate bidirectional, but at times the wireless link

between two nodes may be only unidirectional, allowing one node to successfully send packets to the other while no communication is possible in the reverse direction. Bi-directional links are assumed in the protocol.

4.2 Overview

The message types defined by the protocol are RREQ, RREP, RERR, ALERT, HELLO and PUSH. These message types are received via UDP, and normal IP header processing applies. So, for instance, the requesting node is expected to use its IP address as the Originator IP address for the messages. For broadcast messages, the IP limited broadcast address (255.255.255.255) is used. This means that such messages are not blindly forwarded.

The main feature of this protocol is the formation of independent paths and using the signal strength as the path selection criteria. The reason for preferring the use of the signal strength metric over the usual hop count metric is that the hop count of a route is not sufficient to determine the quality and stability of the path. A very weak link, even if on a low hop count route, could lead to a significant number of dropped packets, leading to the re-initiation of the route discovery process. This results in an increased number of control packets in the network, and thus, may lead to a reduced packet delivery ratio. In contrast, using the signal strength metric provides information about both the quality and reliability of the path. Choosing paths on the basis of signal strength yields more reliable routes with very little chances of link breaking.

Moreover the destination, on detection of broken links, reports the source of the problem using the second alternate route from the selected path vector maintained by the destination; like the source.

4.3 Route Discovery

4.3.1 Route Request

When the source has data to send to some node in the network (called the destination node for that source) but no route information of the destination node the source initiates a route discovery procedure. In this route discovery procedure the source broadcasts route request packets (RREQ) to the downstream nodes in the network. The RREQ packet initially contains the IP address of the source, the IP address of the destination of that source and IP address of the first neighbor of the source. Each intermediate node of the source forwards the packet to its downstream node after inserting its IP in the packet. When some downstream node (other than the neighbor of the source itself) receives the RREQ it considers if this is the first packet it is processing. If it is the first one, it calculates the signal strength and broadcasts the packet to its downstream nodes. If this is not the first one, it sees if this packet has come from a different source neighbor as compared to the one it processed earlier for the same source. If yes, then it discards the packet because the node itself can be available in one path only. If not, then it calculates the signal strength again and if the signal strength is greater than the previous one it has processed, it forwards the packet. Figure 4-1 gives the algorithm for processing RREQs at an intermediate node.

```
function proc_req(MP : mobility prediction, SourceIP, Dest_IP, Source_neighborIP)
    {MP is normalized with legal values ranging from 0 to 1}
    if x = 1 then
        if MP > 0.0001 then forward the packet
        else discard the packet
    else if Source_neighborIP = Route_table_neighborIP
        then
            if MP > Route_table_MP then forward the packet
            else discard the packet
    else discard the packet
```

Figure 4-1: Algorithm for Processing a RREQ
at an Intermediate Node

When the packet reaches the destination, the destination again calculates the signal strength which gives the cumulative signal strength of the entire path. It keeps a vector of the paths that are from different source neighbor nodes and which have signal strength greater than the threshold. Thus in this manner what the destination has at the end is a vector of independent paths which are the most stable ones in the network till now. Since the links are assumed to be bi-directional, the calculation of the most stable paths is done by the destination itself rather than the source. Figure 4-2 gives the algorithm for processing a RREQ at the destination.

```
function proc_req_dest(MP: mobility prediction, SourceIP, Dest_IP, Source_neighborIP
(MP is normalized with legal values ranging from 0 to 1)
if source_neighborIP = Route_list_neighborIP
 then
     if ((MP > Route_list_MP) and (MP > .0001)) then select path
     else
         discard path
else
   if (MP > .0001) then select path
   else
       discard path
```

Figure 4-2: Algorithm for Processing a RREQ
at the Destination Node

4.3.2 Route Replies

Once the best paths in terms of being the most stable ones are available to the destination it now has to send the route replies back to the source. The destination sends each path in the vector through the route available in the path enclosed within a route reply packet containing all the node IPs of that particular route selected. Each node receives the reply, checks for its next hop (to which it will forward the data packet) and last hop (from which it will receive the data packet) and then forwards the reply packet to its previous hop. In this manner the packet moves on from one node in the path to the other node (its upstream for that path) till it reaches the source. The source now contains the same vector that was available to the destination node. The source extracts the signal strength information of all the vectors, sorts to arrange the strength in descending order and selects the path whose signal strength is always the first one in the sorted list of the signal strengths. That path is then selected as the primary path and data is immediately sent on that path. Figure 4-3 gives the algorithm for processing a RREP at

the intermediate nodes, and Figure 4-4 shows the algorithm for processing a RREP at the source node.

```
function proc_reply(nodeIP, next_hopIP, prev_hopIP)

        route_table next_hop = next_hopIP

        route_table prev_hop = prev_hopIP
```

Figure 4-3: Algorithm for Processing a RREP
at an Intermediate Node

```
function proc_reply_src(path)

        int i = 0;

        route_table[i] = path

        i++;

        signal strengths sorted in descending order in sig_str[] array

        sig_str[0] == highest strength

        forward data on route table[0]
```

Figure 4-4: Algorithm for Processing a RREP
at the Source Node

As a result of the above mechanism, data packets always travel along the most stable path. Whenever the signal strength of the current primary path becomes lower than one of its alternate paths, the primary path is switched. At all times, the best available path is the primary path. In this way, the source switches routes to a better alternative when it sees the primary path growing weaker. Since the route is switched before the primary path is broken, fewer data packets are dropped and the end-to-end delay is also minimized. To prevent path oscillations, a heuristic mechanism is adopted. Here, the source node switches from its current primary path to an alternate path only if the

difference in the corresponding path stabilities is greater than some predefined threshold.

4.4 Route Maintenance

There are two situations in which route maintenance must be done for the protocol.

4.4.1 Route Breakages

Route breakage occurs whenever some link used to send data breaks or some node fails. In either case the immediate upstream node detects the failure of the node/ link and sends a route error message (RERR) back to the source. Since the source node maintains a vector of available paths thus it shifts to the second best path immediately, deleting the primary path from its route table. This is where the maintenance of alternate paths comes into benefit. Instead of initiating another route discovery and discovering another new path, which wastes a lot of time and decreases the packet delivery ratio of the protocol, the source immediately shifts to a newer route which would now be the most stable one in the network Figure 4-5 gives the algorithm for route maintenance at an intermediate node.

```
function route_main_inter ( Packet *rerr, upstream_nodeIP )

        send (rerr, route_table next_hop)
```

Figure 4-5: Algorithm for Route Maintenance
at an Intermediate Node

To further fasten up the process of the source sending the data packets with out any delay to the destination in case of the primary path failing, the destination sends an ALERT packet to the source through the now most stable route. When the destination is receiving the data packets normally from the source, the destination checks out the rate

at which it are receiving the packets. Incase of primary path failing, the destination discovers that it has not received any packet for a time that is double the rate at which it had been receiving earlier. The destination knows about all the stable paths. It thus sends an ALERT packet back to the source through the second available path. Whichever of the two packets reach the source first (RERR or ALERT), the source is informed that the primary path must now be changed. Figure 4-6 gives the algorithm for route maintenance at the destination node, and Figure 4-7 gives the algorithm for route maintenance at the source node.

```
function route_main_dest ( )
        ALERT * al;
        al->err = 1
        al->src_IP = route_table_srcIP
        send (al, route_table nexthop)
```

Figure 4-6: Algorithm for Route Maintenance at Destination Node

```
function route_main_src( )
        delete route_list[0]
        shift array to left
        forward data on route_table[0]
```

Figure 4-7: Algorithm for Route Maintenance at the Source Node

The reason the second mechanism (using ALERT packets) is used is to extend the protocol to deal with the scalability issue as well. If there is only one source in the

network then the queue for the data packets at each node is available only for that source. The node is to process the data packets for that source only. In that case the RERR message will travel much faster to the source and there is no need for any other mechanism of reporting the source. But when there are more sources in the network, some node may be part of the route of many other sources. In that case processing data packets from the queues of so many sources will take a lot of time and the source will be informed after a long period during which a lot of data packets may be lost, reducing the packet delivery ratio further. Thus ALERT packets are a solution to this problem. Any node that will receive this ALERT packet will forward it before forwarding any data/ control packet for any other source. ALERT packet itself will contain IP of the source, and a bit that will be set by the destination. In this way forwarding such a small packet will not take any time and the source will be informed quickly.

4.4.2 Route Refreshment

Mobile nodes in an ad-hoc network are constantly moving. The speed of the nodes in the network may vary a lot. Sometimes they may be static and at other time they may be moving fast, in and out of the network. To predict the topology of the network at any given time is thus impossible. Most of the routing protocols devised for mobile ad-hoc networks maintain different routing information at the intermediate node, that thus report to the source about any available route to the destination at any given time.

To ensure that the alternate paths stored at each source node remain up-to date with the changes in the network topology, a separate mechanism is needed. The source node periodically sends a special update message, called PUSH, to the destination along each of its alternate paths. As the PUSH packets propagate through the alternate paths, every node along that path updates the packet with a mobility prediction metric (MP). The

MP is a measure of the relative signal strength with which a node receives a packet from its upstream node. Equation 4.1 gives the value of MP as calculated by a node.

$$MP = \frac{P_{AB} - P_{min}}{P_{min}}$$

Equation 4-1

where P_{AB} is the power of the signal from node A as received by node B and P_{min} is the minimum threshold power with which the signal must be received for it to be considered as a valid transmission. Thus, the MP is a normalized representation of the signal strength. The source initializes the MP to one and as the packet traverses through the path, each node multiplies its MP with the value in the PUSH. Figure 4-8: Algorithm for Processing a PUSH Packet at the Source and Figure 4-9: Algorithm for Processing a PUSH Packet at an Intermediate Node give the algorithms for processing of PUSH packets.

```
function push_at_src()

        {MP is normalized with legal values ranging from 0 to 1}

        MP = 1
        Send_push_down(MP, srcIP)
```

Figure 4-8: Algorithm for Processing a PUSH Packet at the Source

```
function send_push_down(MP: mobility prediction, node_IP)
        {MP is normalized with legal values ranging from 0 to 1}
        if (nodeIP = route_table prev_hop) then MP *= (P – Pmin) / Pmin
                forward (MP, nodeIP)
        else
                if ( MP > .0001) then forward (MP, nodeIP)
```

Figure 4-9: Algorithm for Processing a PUSH Packet at an Intermediate Node

Hence when the PUSH packet reaches the destination, the value of the MP in the PUSH is a cumulative product of the MPs of all links along that path. The path MP is given by Equation 4.2.

$$MP_{path} = \prod_{i \in path} MP_i$$

Equation 4-2

This product gives a measure of the relative stability of the path because links with higher signal strength are less likely to break. As the value of the MP increases, so does the stability of the path.

The destination updates the values of the signal strength for all the paths it has maintained and the MP value is then unicast back to the source through the same path. In this way the source is always updated for any change in the signal strength of the path. When the source finds out that the signal strength of its primary path it had been holding earlier and the one it has now received from the PUSH packet is different and sees that the newer value is very small than the earlier one, it immediately comes to know that the primary path is becoming unstable and thus shifts to the path that now has

the best signal strength value. This mechanism in itself reduces the chances of any data loss due to link breakages at the primary path a lot.Figure 4-10: Algo for Processing a PUSH Packet at the Destination gives the algorithm for push packets processing at the destination node.

```
function push_at_dest (MP, nodeIP)

        if (nodeIP = path_list_node) then path_str = MP
                update path_list

        else
                if((MP > .0001 ) and (sourc_neig != route_table_src) )
                        then insert (new path_list)

        send_src_updates (MP, path_list)

function send_src_updates (MP, path_list)

        old_path_list = path_list
```

Figure 4-10: Algo for Processing a PUSH Packet at the Destination

4.5 Example

In order to illustrate the algorithm, an example of how route discovery and route maintenance will take place is presented in the network of Figure 4-11.

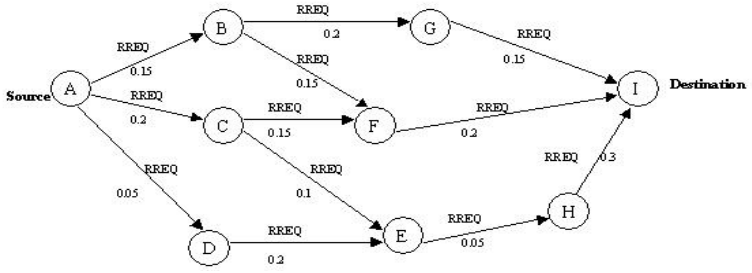

Figure 4-11: Route Discovery/Maintenance Example

The source node A broadcasts a RREQ packet to all its neighboring nodes. Node B receives the packet, records the value of the signal strength from A to itself, multiplies it

with the value of MP in the packet and broadcasts the packet. The packet reaches nodes F and G. Now both the nodes F and G have just one upstream node .Consider the case of node G first. It receives the packet from B and will thus only calculate the MP in the packet with the value of the signal strength from node B to node G (as obtained from the MAC layer by node G). The packet is then forwarded to the downstream nodes by node G. As a result of the packet forwarded by node G, the RREQ packet reaches the destination node I via the path A-B-G-I and the cumulative signal strength of the entire path is recorded at the destination node. The destination node takes the MP value and multiples this value from the signal strength it receives for the last hop through which it received the packet. As a result the destination gets the cumulative signal strength of the entire path.

Now consider the case of node F. Node F receives the packet broadcast by node B. It will also receive the packet that is broadcast by node C. Let us assume that the packet reaching node F via C arrives at F before the packet coming from B. The packet that traverses the path A-C-F has an MP value of 0.03. The node F broadcasts this packet and the packet reaches destination I. At some later time, the node F receives the packet coming from the path A-B-F but it has an MP value of 0.0225, which is less than the previous value of MP recorded by the node (i.e. when the packet came from the path A-C-F), so the node F discards this packet, also since F has processed packet from B so it will not process any packet from C; rule is that a node can be part of just one path only. Let’s say now that the node E receives the RREQ from D before it receives the RREQ from C. The recorded MP value at this stage is 0.01 at node E. The node forwards the RREQ which reaches H and finally reaches I. The RREQ reaching E via C is discarded because this although path has a higher cumulative MP value (0.02) than the one

previously recorded (0.01) but one packet for a different neighbor of the same source has already been forwarded. So the node discards this packet. The destination node I wait for some timeout period, and then do the calculations. Lets say that the during this interval, I receives all the four RREQ packets that have been discussed above. The situation now is that the node I has the paths and the corresponding MP values as (a) Path A-B-G-I : MP = 0.0045 (b) A-C-F-I : MP = 0.006 (c) A-D-E-H-I : MP = 0.0001

Each RREQ packet has information about the immediate neighbor of the source node. This information is in the form of the IP address of the neighbor nodes incorporated in the RREQ packet. The destination thus knows that nodes B, C and D are the neighbor nodes of the source. No two routes should have the same neighbor of the source as this would lead to paths that are not independent. Path A-C-F-I has the highest cumulative signal strength which is .006, followed by path A-B-G-I.

Also, the destination sorts the paths in decreasing order of their MP values. As a result, the list stored at the destination is (a) Path A-C-F-I : MP = 0.006 and (b) Path A-B-G-I : MP = 0.0045.

This path information is sent back to the source each through its own route. Thus first one will follow through its path A-C-F-I and the second one will follow through its own path A-B-G-I. Each intermediate node when forwarding the reply packet will store information about its upstream node as well as its downstream node. Thus for example node C in the first route will store in its routing table that node A is its upstream node (through which it will receive its data or any other packet) and node F is its downstream node to which it must forward / unicast any packet for source A.

Path A-C-F-I will be selected as the primary path by the source and the destination. Path A-B-G-I will be the second option. Incase primary path fails; no new route discovery will take place. Rather the second path will be adopted by the source and be made the primary path. In situations, when no alternate routes are available in case the primary route has been broken, the route discovery process is re-initiated.

Once the route discovery process is done, the source starts sending data on the primary path. As mentioned in the previous section on route maintenance; the source node will periodically generate PUSH packets after about 2 seconds which will be broadcasted on the network so that the signal strength can be calculated. The source node will shift from the primary path to some other path if the difference in the signal strength of any path with the primary path is 1.5 times. PUSH packet is initiated by the source with an MP value of 1. As it moves along the network the MP value is constantly updated by the mechanism described above. If some intermediate node receives a PUSH packet that is already marked as last hop for that node for that particular source than only MP value is placed in the packet. But if the upstream node is a different one and if the MP value is greater than the threshold then IP of the node is also inserted in the packet and broadcasted.

Finally it reaches the destination. Once at the destination the new values are compared with the older ones and destination updates the information. Also some new routes may be discovered which have good cumulative signal strength. The destination sends such path information to the source too. Any changes made by the destination are then sent back to the source. If the primary path signal strength is lower than some other path (1.5 times) then the other path is taken as the primary path. In this way both the source and the destination remain consistent with the path vector information and the signal

strengths of all the paths in the network. This mechanism thus deals with the mobility of the network where any node moving out (reduction in the total signal strength of the path if that node is part of some path) or moving in (increment in the total signal strength of the path if that node is part of some path) will be detected and the necessary steps will be taken.

4.6 Format of Protocol Packets

The type field in each packet is numbered in the order in which the packet is appearing.

4.6.1 Route Request Packet (RREQ)

The format of the RREQ packet is shown in Figure 4-12.

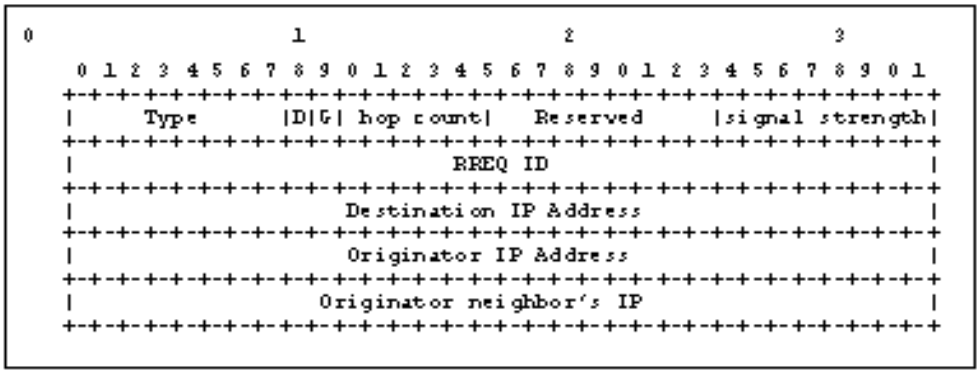

Figure 4-12: RREQ Format

The entries in the RREQ packet are elaborated in Table 4-1.

Table 4-1 Entries in RREQ packet

G	Gratuitous RREP flag; indicates whether a gratuitous RREP should be unicast to node specified in Destination IP Address field
D	Destination only flag; indicates only the destination may respond to this RREQ
Reserved	Sent as 0; ignored on reception.
Hop Count	The number of hops from the Originator IP Address to any node
Signal Strength	Initialized value of the signal strength for the path to 1
RREQ ID	A number uniquely identifying the particular RREQ
Destination IP	The IP address of the destination for the route
Originator IP	The IP address of the node which originated the Route Request
Originator's neighbor IP	The IP of the neighbor of the source

4.6.2 Request Reply Packet (RREP)

The format of the RREQ packet is shown in Figure 4-13.

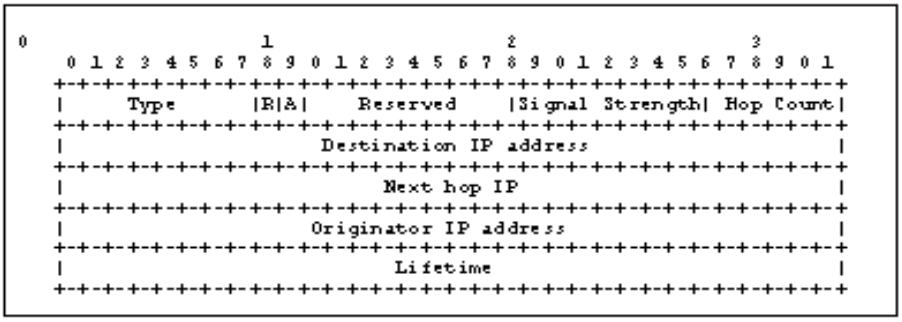

Figure 4-13: RREP Format

The entries in the RREP packet are elaborated in Table 4-2 .

Table 4-2 Entries in RREP packet

R	Repair flag; used for multicast
A	Acknowledgment required
Reserved	Sent as 0; ignored on reception.
Signal Strength	The cumulative signal strength of the entire path from the destination to the source
Hop Count	The number of hops from the Originator IP Address to the Destination IP Address
Destination IP address	The IP address of the destination
Originator IP address	The IP address of the node which originated the RREQ
Next Hop IP	IP of the next hop to which a packet has to be forwarded
Life Time	The time in milliseconds for which nodes receiving the RREP consider the route to be valid

4.6.3 Route Error Packet (RERR)

The format of the RERR packet is shown in Figure 4-14.

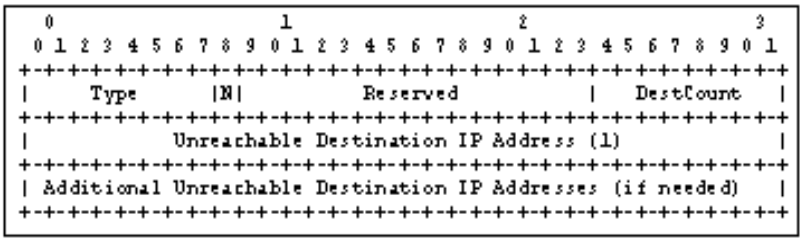

Figure 4-14: RERR Format

The entries in the RERR packet are elaborated in Table 4-3.

Table 4-3 Entries in RERR packet

N	No delete flag; set when a node has performed a local repair of a link
Dest Count	Number of unreachable destinations
Reserved	Sent as 0; ignored on reception.
Unreachable Destination IP Address	IP address of the unreachable destination

4.6.4 Route Reply Acknowledgment (RREP – ACK)

The format of the RREP-ACK packet is shown in Figure 4-15.

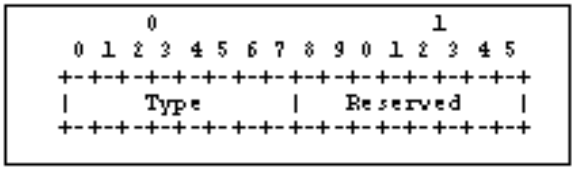

Figure 4-15: RREP - ACK Format

The entries in the RREP-ACK packet are elaborated in Table 4-4.

Table 4-4 Entries in RREP-ACK

Reserved	Sent as 0; ignored on reception

4.6.5 ALERT

The format of the ALERT packet is shown in Figure 4-16.

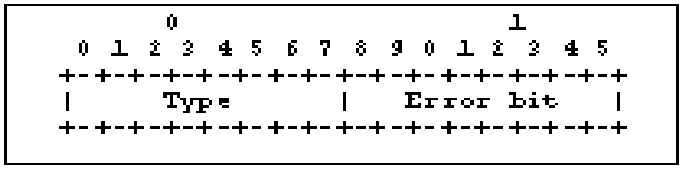

Figure 4-16: ALERT Format

The entries in the ALERT packet are elaborated in Table 4-5.

Table 4-5 Entries in ALERT packet

Error bit	Sent as 1; primary path is failed

4.6.6 HELLO

The format of the HELLO packet is shown in Figure 4-17: HELLO Format.

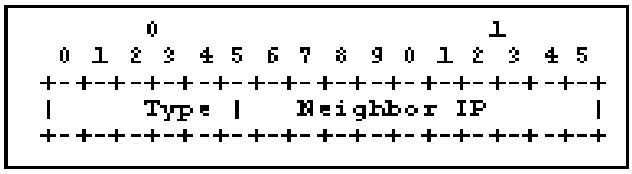

Figure 4-17: HELLO Format

The entries in the HELLO packet are elaborated in Table 4-6.

Table 4-6 Entries in HELLO packet

Neighbor IP	IP of the neighbor in the table

4.6.7 PUSH

The format of the PUSH packet is shown in Figure 4-18.

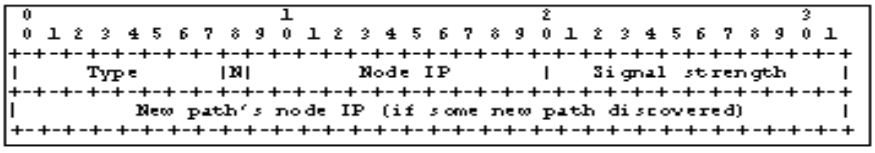

Figure 4-18: PUSH Format

The entries in the PUSH packet are elaborated in Table 4-7.

Table 4-7 Entries in PUSH packet

N	No delete flag; set when a node has performed a local repair of a link
Node IP	IP of the current holder of the packet
Signal Strength	Cumulative signal strength till now
New path's node IP	If some new route discovered; insert IP of the new node holding the packet

4.7 Conclusion

Computation of the most stable paths, based on the signal strength, involves only a marginal increase in computation at the source nodes. Signal strength is used as the path selection metric as opposed to the hop count, because previous experimental results have demonstrated that the use of weak links can lead to routing path oscillations and numerous dropped data packets.

5 Testing and Simulation

5.1 Random Waypoint Mobility (RW) Model

The Random Waypoint model, as depicted by Figure 5-1, is most commonly used mobility model in research community. In the current network simulator (NS-2) distribution, the implementation of this mobility model is like this: at every instant, a node randomly chooses a destination and moves towards it with a velocity chosen uniformly randomly from [0,Vmax], where Vmax is the maximum allowable velocity for every mobile node. After reaching the destination, the node stops for a duration defined by the 'pause time' parameter. After this duration, it again chooses a random destination and repeats the whole process again until the simulation ends. In this framework, the RW model acts as the 'baseline' mobility model to evaluate the protocols in Ad Hoc Network.

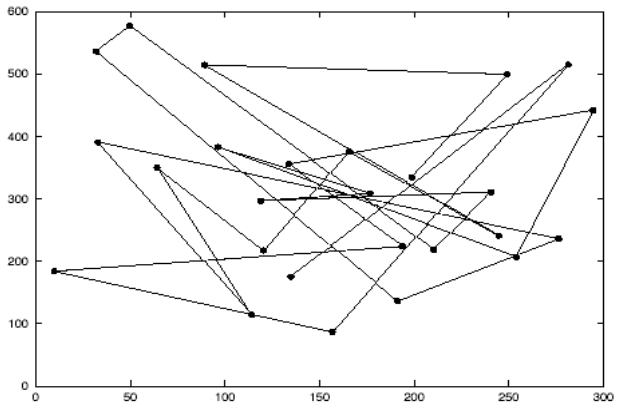

Figure 5-1: Traveling Pattern of the Node in Random Way Point Mobility Model

5.2 Implementation of Simulation Model

This section describes the network simulator 2 (NS-2), its tracing mechanism and especially the wireless model in NS-2

In order to test the protocol, an implementation in a network simulator is chosen. The alternative of an implementation in a real system (e.g., Linux) and testing it as experimentation would use too much resources and finally be too expensive. Furthermore, the implementation in a simulator offers more flexibility and variations, i.e., scenarios with much more nodes can be tested and adapted for the initial parameter tuning. An implementation in real systems can be considered, if the verification with the help of the simulation is successful. A network simulator for the verification of the cooperation schemes should fulfill the requirements: (a) Simulation scenarios with 50 and more nodes, (b) Physical Layer model with Radio Propagation, (c) MAC Layer and Link Layer models, (d) Mobility of the nodes and (e) Enhanced tracing functionality.

There exist quite a number of network simulators today. Not all of them have a good reputation within the research community, and of those which have, most are expensive. Therefore, NS-2 is chosen, because it is open source software, freely available and it is widely used in the research community. Besides, NS-2 meets perfectly the requirements

5.2.1 Implementation with Network Simulator II

NS-2 is a discrete event driven simulator. The source code and the documentation [17] are currently maintained by the Virtual Internet Test bed (VINT) at the Information Sciences Institute (ISI) of the University of Southern California (USC). The goal of NS-2 is to support networking research and education. It provides an environment for

protocol design, traffic studies and protocol comparison. Its license model enables the sharing of code, protocols, models, and ensures that the work is given back to the community. It allows easy comparison of similar protocols. This collaborative environment and the big number of users should also increase the confidence in the results because more people look at the models in more situations than by using a closed source simulator.

5.2.2 Structure of NS-2

In NS-2 real world objects are modeled by objects in the simulation and programmed to react as much as possible as their correspondents in the real world would react. In the concept of event driven simulation, physical activities are translated to events. The events are stored in a queue. They are processed in the order of their scheduled occurrences. The time in the simulation progresses as the events are processed. Each event happens in an instant of simulated time, but takes an arbitrary amount of real time. NS-2 is built using object oriented methods in C++ and Otcl. The developers of NS-2 tried to combine fast iteration time with good run-time performance. This results in a mixed coding framework in C++ and Otcl, Figure 5-2, C++ serves as system programming language in which all time consuming components, e.g., packet processing and routing algorithms, are implemented. OTcl is used as the configuration language for the simulation scenarios. It allows the quick setup of different simulation scenarios and an interactive simulation mode. OTcl and C++ share linked class hierarchies and the additional library TclcL offers sharing of functions and variables. Objects in C++ are compiled and then made available to the OTcl interpreter through an OTcl linkage (TclcL) which maps methods and member variables of the C++ object to methods and variables of the linked OTcl object. This system architecture facilitates the

usage of NS-2 and its existing components, but it makes the development of new components complicated and time-consuming.

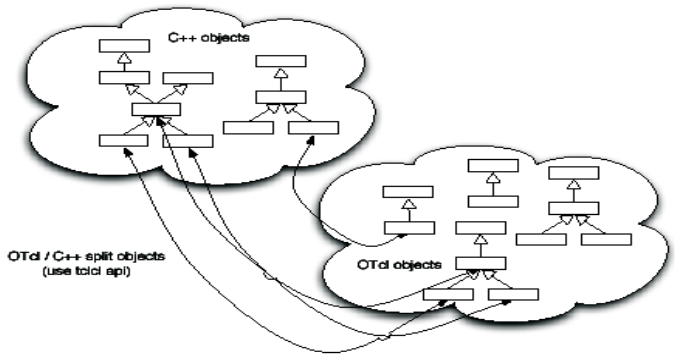

Figure 5-2: Duality of C++ and Otcl in NS-2

5.2.3 Internal Packet Representation

The internal packet representation of NS-2 is quite different from a packet in the real world. The packet in the simulator contains all headers that the simulator supports, e.g., UDP, TCP, MAC, IP etc., and not only the headers of the real world packet. Furthermore, a packet in the simulator has a common header which contains important simulation information, e.g., the simulated packet size (size of the real world packet), the packet type, the flow direction, a unique packet ID and a time-stamp. Besides, the packet headers of NS-2 do not necessary correspond to the protocol headers defined in RFCs, e.g., header checksums are normally left out.

5.2.4 Simulation Process

The figure shows the simplified process for a simulation. The user, Figure 5-3, has to set the different components, e.g. event scheduler objects, network components and setup module libraries, up in the simulation environment. This is done by a simulation

script in OTcl. The script is processed by ns2 and delivers trace files that the user analyzes with the Network Animator (NAM) or custom scripts.

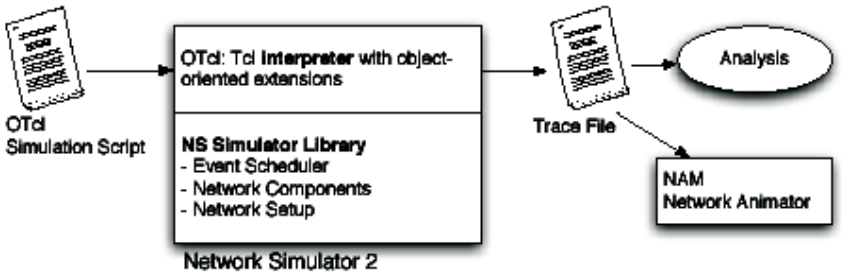

Figure 5-3: User View of NS-2

5.2.5 Wireless Model in NS-2

The wireless model in NS-2 is contributed from CMU's Monarch project (Wireless extension to NS-2). Various modules were added to ns2 to simulate node mobility and wireless networking, including (a) Mobile Node, (b) Base station Node, (c) Ad-hoc Routing Agents (DSR, DSDV, TORA, AODV, AODV+), (d) MAC 802.11, (e) Radio Propagation Model and (f) Channel.

5.2.6 Tracing

NS-2 offers tracing of all packets in the simulation. Furthermore, NS-2 enables the tracing of variables in C++ or OTcl and supports the monitoring of queues and flows (see [17] for detailed information). In this thesis only the packet tracing ability is used. There exist three different trace file formats (old, new wireless and NAM) for packet tracing. [18] gives a good overview of them.

5.3 Simulation Model

Used in the thesis is a detailed simulation model based on NS-2. The Monarch research group in CMU developed support for simulating multi-hop wireless networks complete with physical, data link and MAC layer models on NS-2. IEEE 802.11 [19] is used as the MAC layer. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN [20]. WaveLAN is a shared-media radio with a nominal bit-rate of 2 Mb/sec and a nominal radio range of 250 meters. The random waypoint model is used to model mobility. Here, each node starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0 and max. speed taken as 10 m/s). Once the destination is reached, another random destination is targeted after a pause. The mobile hosts are placed randomly within an 800mx800m area. Traffic sources are CBR (continuous bit-rate)/User Datagram Protocol (UDP). Real time communication is all UDP, that is why, UDP is simulated on the transport layer. The source-destination pairs (sessions) are spread randomly over the network. Only 512 byte data packets are used. Simulations are run for 300 simulated seconds.

Different mobility scenarios and traffic patterns are used for the various simulations done. MAORP is compared with AODV and DSDV to have an idea about how the protocol performs when judged against a reactive protocol and a proactive protocol.

5.3.1 Metrics

To judge the efficiency and effectiveness of MAORP, the metrics used for the analysis are (a) Packet Delivery Ratio which is the measured ratio between the number of data packets delivered to the destinations and the number of packets generated by all traffic sources and (b) Normalized Routing Load which is the ratio of the number of control

packets propagated by every node in the network and the number of data packets received by the destination node. This value hence represents the protocol's efficiency. Mobility and Sc

5.4 Mobility

5.4.1 Simulation Environment

For generating the results to measure the performance of the three protocols with changing mobility conditions, environment in NS-2 was set up as given in Table 5-1.

Table 5-1 Simulation Environment A

Simulation Parameters	**Values**
Num of nodes	50
Num of connections	47
Max speed	10 m/s
Min speed	0 m/s
Pause time	0 sec, 2 sec, 10 sec, 20 sec, 50 sec, 70 sec, 100 sec, 150 sec
Packet size	512 bytes
Sending rate :	1 packet/sec (CBR)
Transport layer	UDP
Mac layer	IEEE802.11
Antenna type	Omni antenna
Communication range	250 m
Bandwidth	2 Mbps

5.4.2 Packet Delivery Ration vs. Mobility

Figure 5-4 shows the performance of the three protocols with varying mobility conditions. A lesser pause time means greater mobility. MAORP and AODV perform extremely well, and give a packet delivery ratio of more than 90% for different mobility conditions. DSDV, being a reactive protocol, gives a much lower packet delivery ratio, which deteriorates even further when mobility increases.

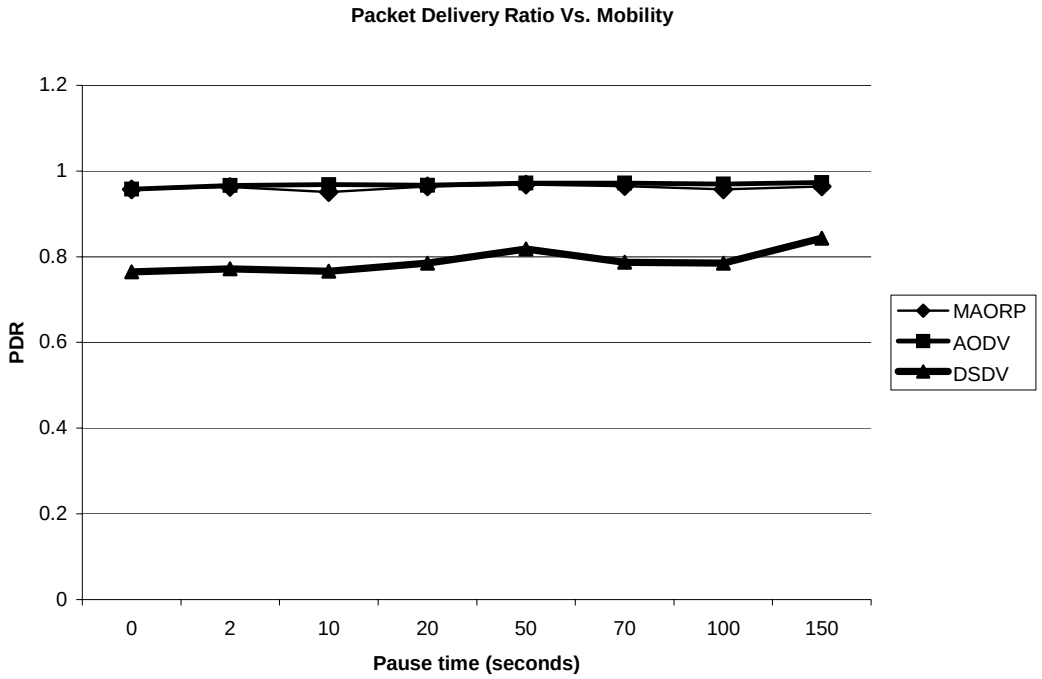

Figure 5-4: PDR VS Mobility

Closely observing the graph shows that the plots of both AODV and MAORP are nearly straight lines. In the case of AODV, the PDR is almost constant with mobility because the intermediate nodes store routes that are changed according to the sequence numbers. Hence time by time nodes get information of routes that are fresh. Also the timer for the generation of control packets is constant whether mobility is high or low. Thus with the intermediate nodes getting new routes information constantly in any situation they immediately inform the source node of any new route to the destination whenever the source node initiates a new route discovery. As a result the PDR is more or less constant for any situation.

The plot of PDR vs. pause time for MAORP is also almost a constant line. In MAORP the source and the destination nodes maintain a vector of available paths. Using signal strength already in itself provides paths that are stable and have very little chances of breaking. And under high mobility, if link breakages are frequent, then maintaining many alternate routes by the source prevents it from initiating another route discovery and a new route is available immediately. Thus data packets can be sent without any delay. This coupled with the generation of PUSH packets every 2 seconds for any mobility situation provides a near to constant PDR.

5.4.3 Normalized Routing Load VS Mobility

Figure 5-5 shows that at low mobility, DSDV has the least amount of NRL.

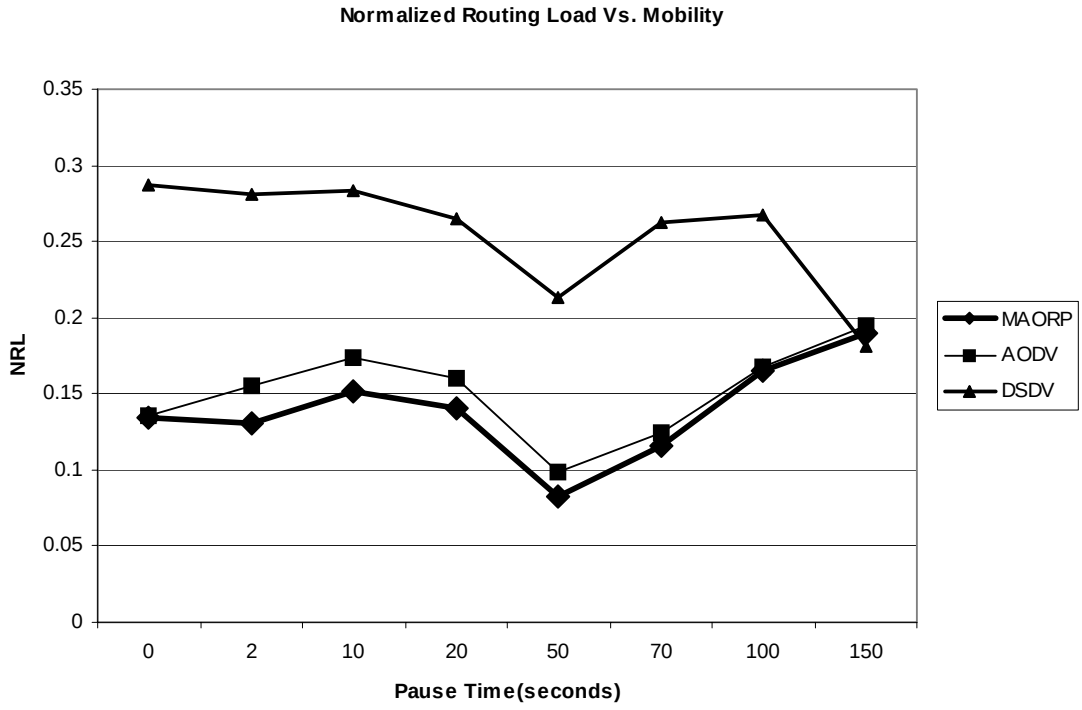

Figure 5-5: NRL VS Mobility

This is expected because MAORP and AODV both generate more control packets during route discovery process. However as the mobility increases, NRL of DSDV also starts increasing. MAORP and AODV both perform much better than DSDV in this regard as mobility increases. MAORP performs slightly better than AODV in this case because MAORP utilizes more stable paths as compared to AODV. The chances of link breakages are more in case of AODV than in MAORP. Once a link is broken, AODV initiates a new Route Discovery cycle, so more control packets are generated. But in MAORP, maintaining stable alternate paths makes a new route immediately available to the source incase of broken links, thus reducing the routing load greatly. Moreover, the size of MAORP push packets is much smaller than that of AODV.

5.5 Scalability

5.5.1 Simulation Environment

To find out how scalable the three protocols are, the simulation environment was set up as shown in Table 5-2.

Table 5-2 Simulation Environment B

Simulation Parameters	**Values**
Num of nodes	30, 50, 70, 100
Number of Connections	27, 47, 67, 97
Max speed	10 m/s
Min speed	0 m/s
Pause time	0 sec, 150 sec
Packet size	512 bytes
Sending rate :	1 packet/sec (CBR)
Transport layer	UDP
Mac layer	IEEE802.11
Antenna type	Omni antenna
Communication range	250 m
Bandwidth	2 Mbps

5.5.2 Packet Delivery Ratio vs. Number of Nodes

First we check the scalability of the three protocols under high mobility. Figure 5-6 shows that when mobility is extremely high (pause time is zero) and the number of nodes in the network is kept below 70, MAORP and AODV have a very consistent PDR, which is also greater than DSDV; but it starts dropping as the number of nodes becomes greater than 70. However, MAORP performs better than AODV as the number of nodes increases beyond 70. This shows that MAORP is more scalable to larger networks as compared to AODV. As the number of nodes reaches 100, DSDV gives the best PDR, followed by MAORP and then AODV.

With DSDV there is a good PDR with increasing number of nodes because more nodes mean more routes to other nodes in the network and thus a route is available every time for the source to send data to the destination. With AODV the PDR decreases with an increase in the number of nodes because this places a lot of load on the intermediate nodes. So many nodes mean that the intermediate nodes must at all times be holding the routing information for these many nodes as well. With such huge routing tables, intermediate nodes are over burdened and thus processing so many packets for so many sources whose data is queued in the buffers takes time, reducing PDR considerably.

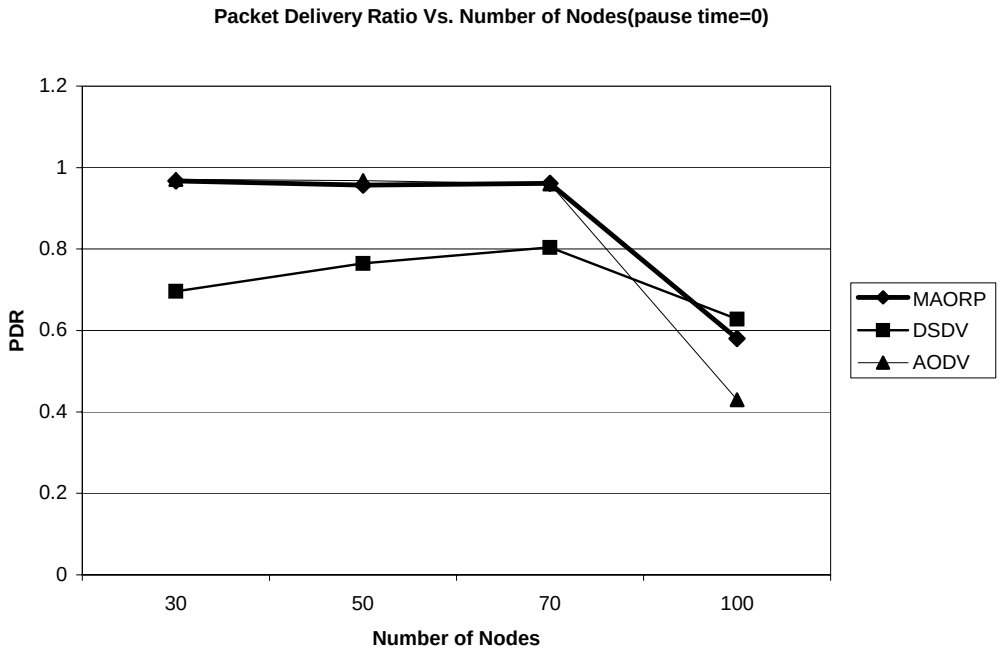

Figure 5-6 PDR VS Number of Nodes (p=0)

Under low mobility, MAORP performs slightly better than AODV when the number of nodes is less than 70 as depicted by Figure 5-7. As the number of nodes increases beyond 70, AODV starts performing a little better than MAORP. Again, DSDV is most scalable to a larger network when the network is not too mobile.

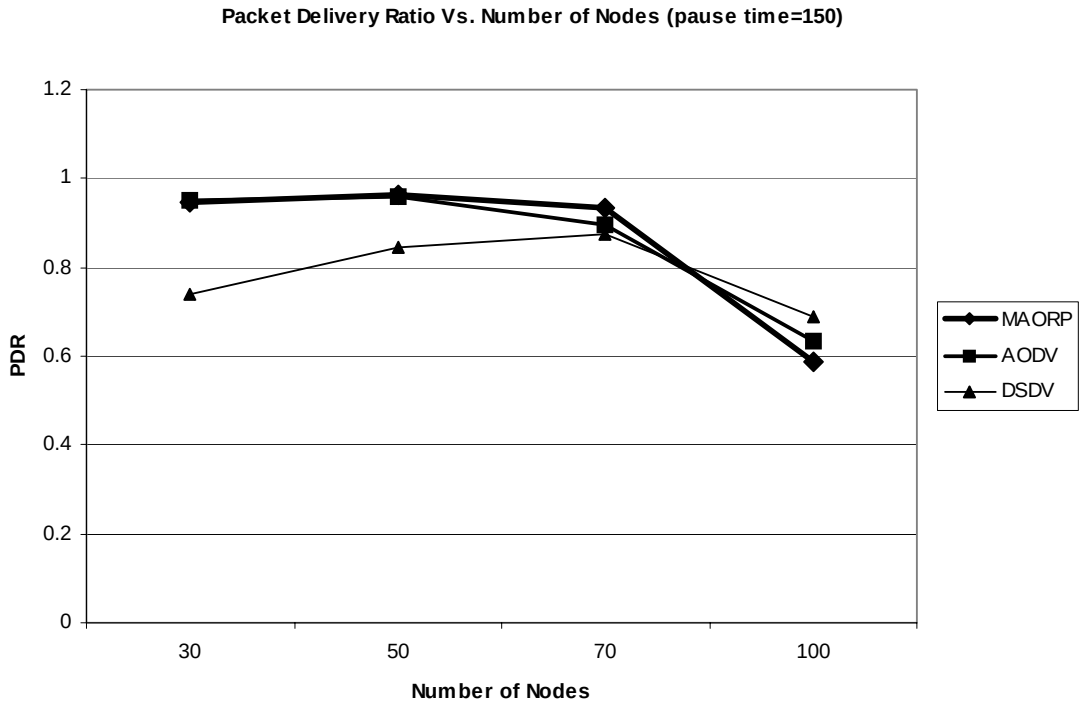

Figure 5-7 PDR VS Number of Nodes(p=150)

The fact that DSDV is pro-active means that as number of nodes increases, so does the information in the routing tables and thus with low mobility the routing tables need not be updated frequently. As a result a good PDR is achieved. AODV and MAORP both show a nearly equal plot at low mobility because since the mobility is very low thus few link breakages will take place and best paths would be used all the times. With the intermediate nodes free from the issues of route maintenance and less flow of RERR messages, no matter how many nodes be there, the PDR would not decrease too much. Also in MAORP, more nodes means more number of alternate stable paths at the source readily available. This stabilizes PDR under low mobility.

5.5.3 Normalized Routing Load vs. Number of Nodes

Figure 5-8 shows that DSDV has an NRL that is greater than both AODV and MAORP when the number of nodes is less than 70, and pause time is taken as zero, that is, very high mobility. But as the number of nodes increases beyond 70, the NRL of AODV increases at a much greater rate than that of MAORP. MAORP thus has a lower routing load than AODV for larger networks, under high mobility conditions.

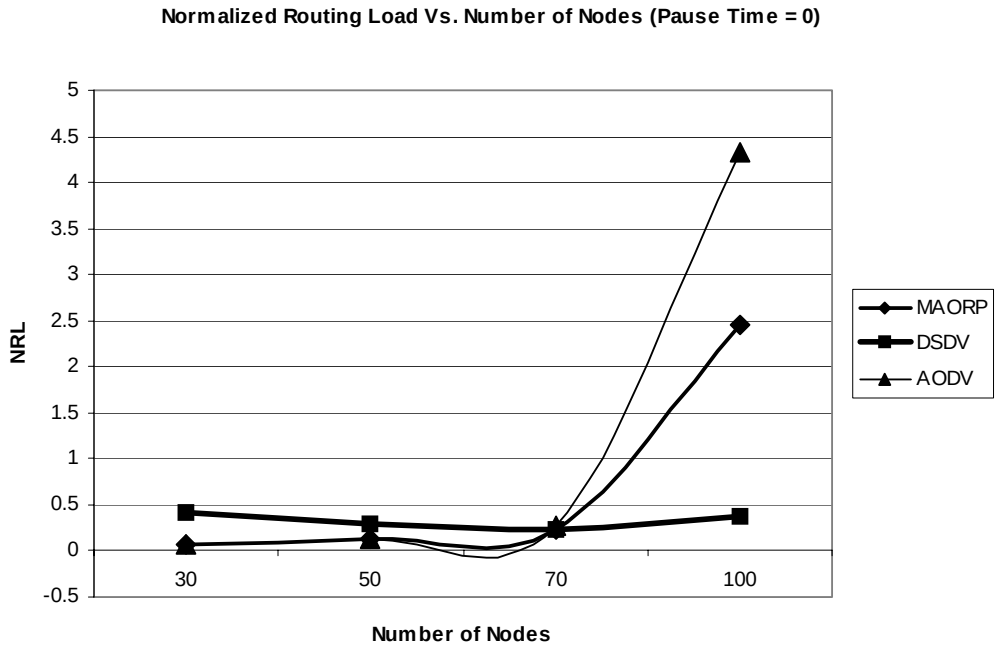

Figure 5-8 NRL VS Number of Nodes (p=0)

With the increase in the number of nodes under very high mobility the NRL of AODV increases greatly. This is because more route breakages and thus more route discovery will take place. As the number of nodes increases, greater number of link breakages would lead to extensive number of RREQs being flooded in the network and more RREPs by all the intermediate and destination to be sent back to the sources. MAORP

lies between AODV and DSDV. With MAORP, although the control overhead will increase too, but since the size of MAORP control packets is much smaller than AODV and since in MAORP the intermediate nodes do not store any routes and are thus less over burdened, hence NRL does not exceed that much.

Again, as shown in Figure 5-9, DSDV gives a consistent performance as far as the NRL is concerned under low mobility (pause time is 150). DSDV in-fact performs better than both MAORP and DSDV as the number of nodes becomes greater than 50. DSDV is thus suited for larger networks which are not highly mobile. The over all performance in terms of NRL of both AODV and MAORP is similar under low mobility conditions.

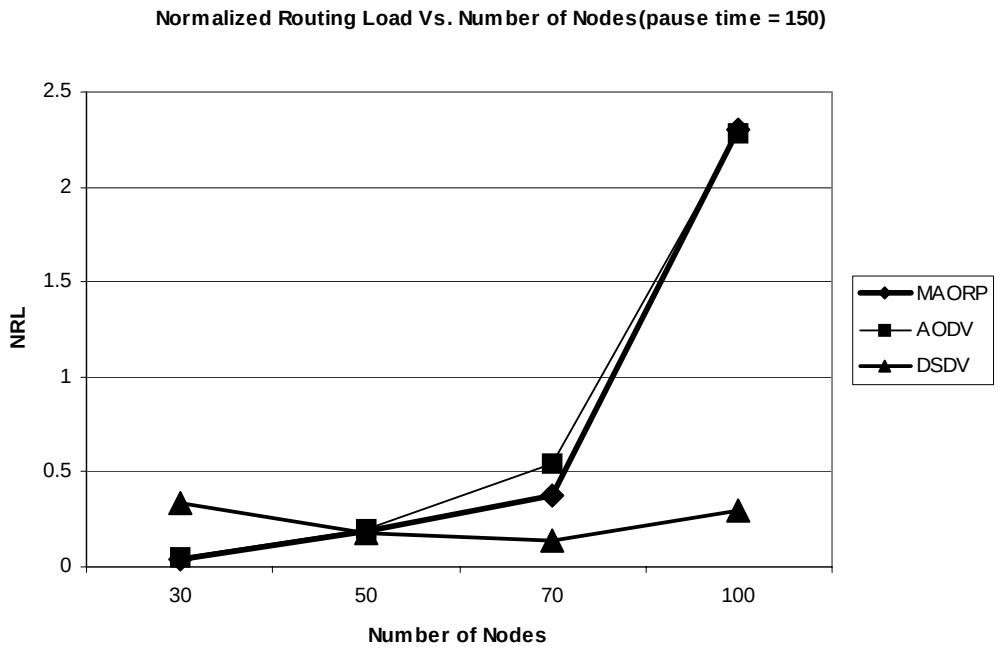

Figure 5-9 NRL VS Number of Nodes(p=150)

Under low mobility, DSDV has an NRL which remains constant with the increasing number of nodes. In case of MAORP and AODV, NRL increases as the number of

nodes increase under low mobility. Since the timers for the generation of control packets of both the protocols are not dependant on mobility, thus most of the times the control packets would be generated when not needed, increasing the routing load.

6 Conclusion and Future Work

6.1 Conclusion

The analysis presented in Chapter 5 leads us to conclude that the performance of MAORP is better than DSDV and comparable to AODV, under conditions of high-mobility, with 70 nodes observed as the bench mark beyond which a drastic change in the characteristics was observed. This drastic change can be attributed to the traffic patterns which were kept constant during all the simulation runs.

With changing mobility, MAORP gives a very consistent Packet Delivery Ratio which remains above 90% for all the simulations performed as part of this project. In this regard, it performs very similar to AODV which also gives an almost constant PDR under changing mobility conditions. This consistent behavior of MAORP can be attributed to the fact that the use of signal strength as the path selection metric provides such stable routes that have minimum chances of breaking. Furthermore, in case of link failures, alternate stable paths are readily available reducing the probability of the initiation of another route discovery cycle. Also, the PUSH packets sent after every 2 seconds ensure that the protocol adapts to the dynamic network topology, and the source and the destination maintain paths that are highly stable at all times. All these factors also help in reducing the routing overhead in MAORP. The analysis shows that MAORP has a much lower normalized routing load than both AODV and DSDV, which indicates its efficiency. AODV has a higher NRL than AODV because in AODV, a link breakage results in the commencement of a new route discovery cycle which again floods the network with RREQs this resulting in a higher routing overhead.

MAORP also scales well to larger networks of mobile nodes and performs even better than AODV as the number of nodes increases beyond 70. However, the general trend observed is that as the number of nodes becomes greater than 70, the performance of MAORP and AODV starts deteriorating and a point comes when DSDV starts performing better than both of these reactive protocols.

Thus, it can be concluded that MAORP gives a highly acceptable performance under all mobility conditions when the number of nodes is less than 70-75. Beyond that, it is DSDV that outperforms the other two protocols. So, MAORP can be used very efficiently in a network that has a rapidly changing topology. The advantage that MAORP has over AODV is that it has a lower routing overhead than AODV, and also it outperforms AODV as the number of nodes increases. MAORP thus provides a viable solution for a more reliable communication system in ad hoc networks.

6.2 Future Work

There is always a room for improvement in every thing. Same goes for MAORP. There still is a need to conduct further analyses to study why the performance starts deteriorating after a certain number of nodes. Since, this protocol promises to provide routes with the most stable links, it should give a PDR higher than AODV. This requires even further studies of the said protocol. The analysis done in this thesis is for CBR/UDP traffic only. The analysis should be done for TCP only and mixed traffic scenarios to get a better idea of how the protocol performs under varying traffic conditions.

The protocol can be extended to include battery power (BP) as an additional metric for the selection of the paths. In addition to appending the MP value in the RREQ and

PUSH packets, the nodes would also append their respective battery power. The RREQ packet would have an additional field for storing the battery power. Three kinds of nodes would be identified: The sending nodes, the receiving nodes and the forwarding (intermediate) nodes.

It is considered that the power consumed by the forwarding node (Pin) is higher than the power consumed by the sending node (Ps), which is higher than the power consumed by the receiving node (Pr).

As the source would send the packet, it would append its battery power value, in the RREQ packet. The value appended will be the current battery power of the node minus Ps. Any intermediate node which might receive this RREQ would add its current battery power to the value of battery power stored in the packet. The current battery power of an intermediate node would be its actual power minus Pin. As a result when a RREQ would reach the destination, it would contain the cumulative MP value of the path, and also the cumulative battery power of all the nodes in the path. The destination would then check the MP and BP values for each path. There could be three different cases. If MP and BP of one particular path are higher than the corresponding values of all other paths, then that path is the primary path. If MP value of a path is the highest, and BP value is lower than the BP value of some other path, that path will be selected as the primary path. If MP value is highest/good enough, but the BP value is lower than a certain threshold, that path will not be included in the path vector.

Same procedure will be used in the Route Maintenance Stage.

APPENDIX A -- FLOW CHARTS

Included in this appendix are the flow charts that explain the entire flow of the algorithm of MAORP. The figures should be viewed in this order: Figure A-1, Figure A-2, Figure A-3 and Figure A-4.

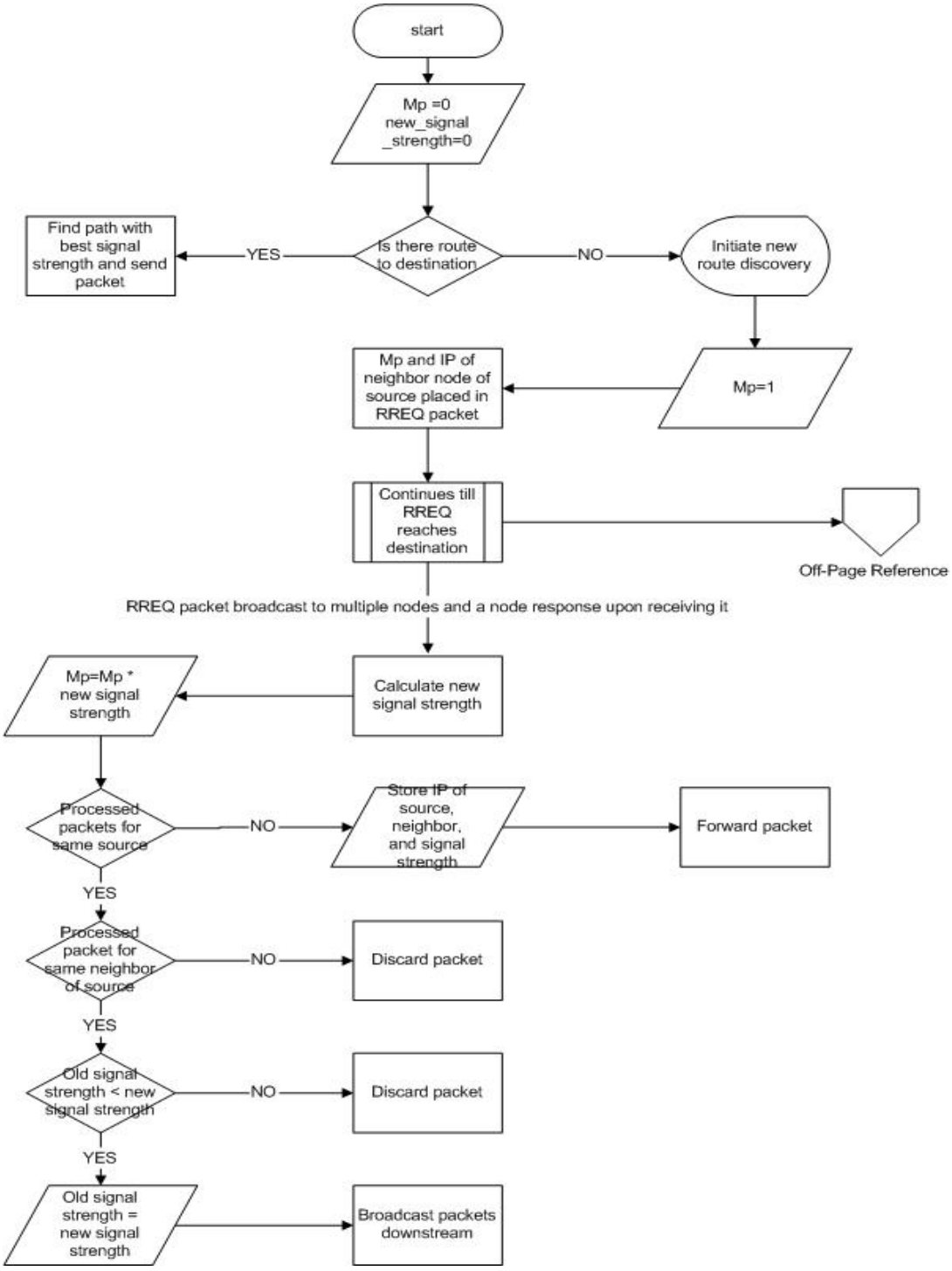

Figure A-1 Flow chart (1/4)

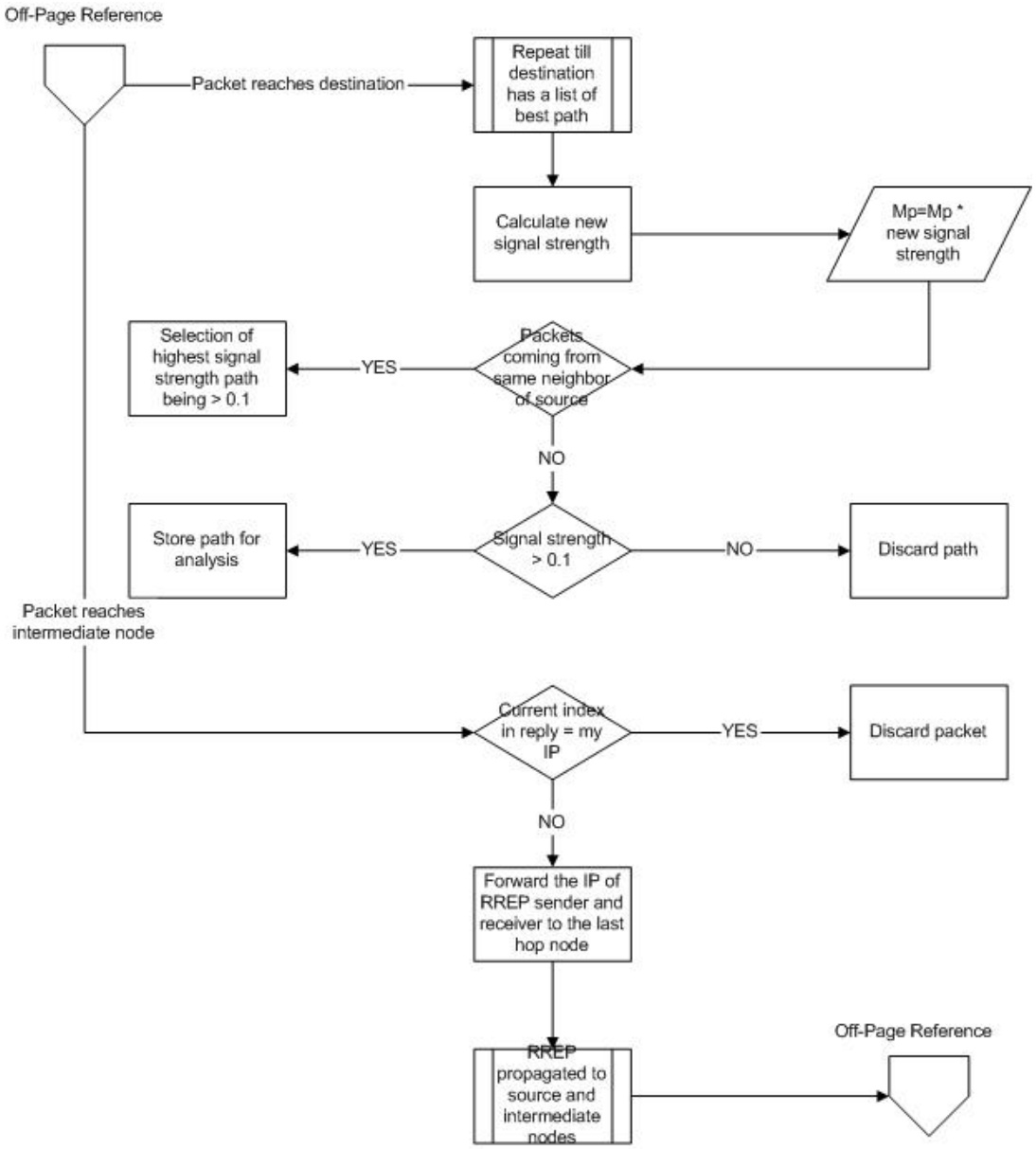

Figure A-2 Flow Chart (2/4)

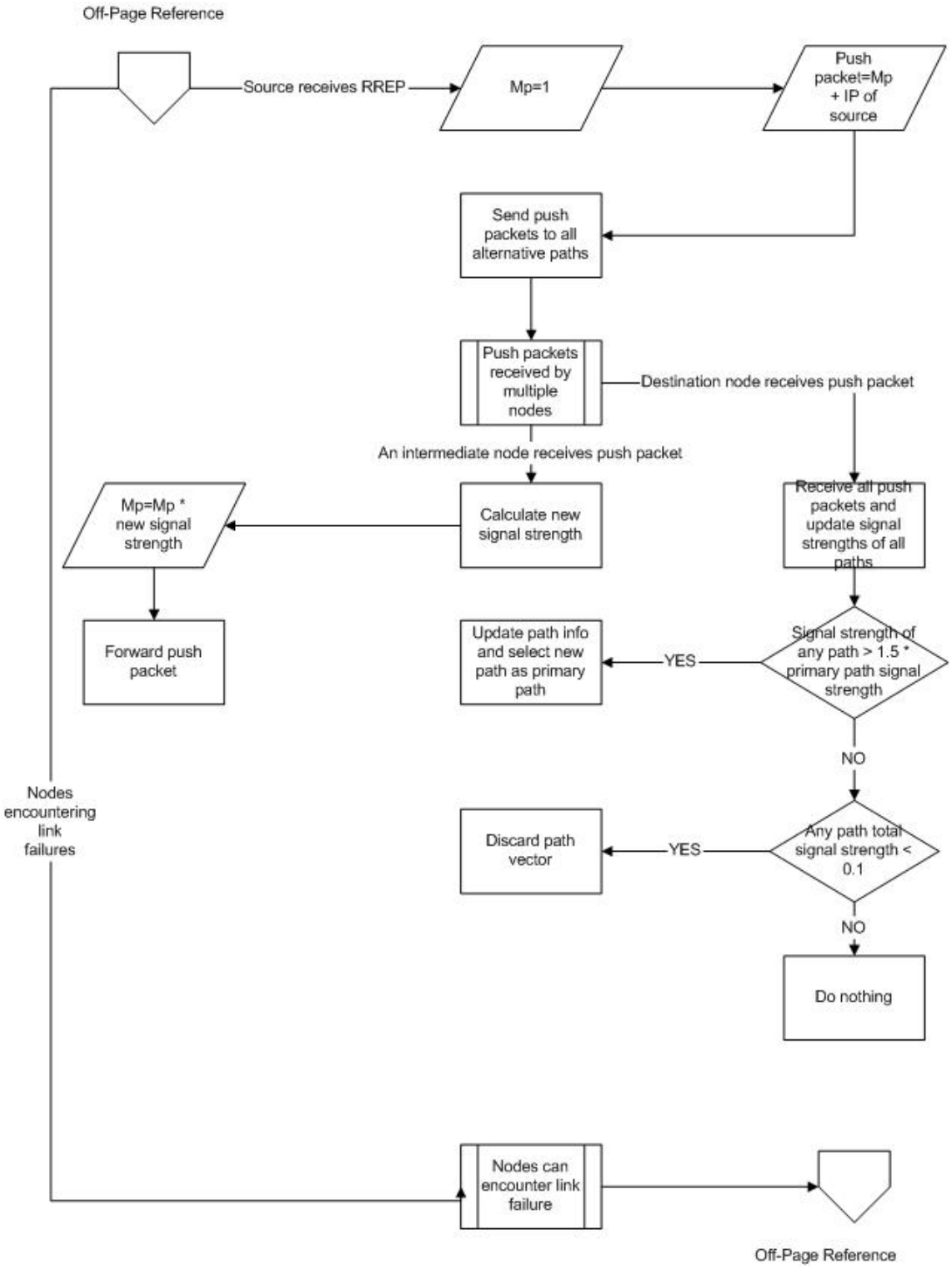

Figure A-3 Flow Chart (3/4)

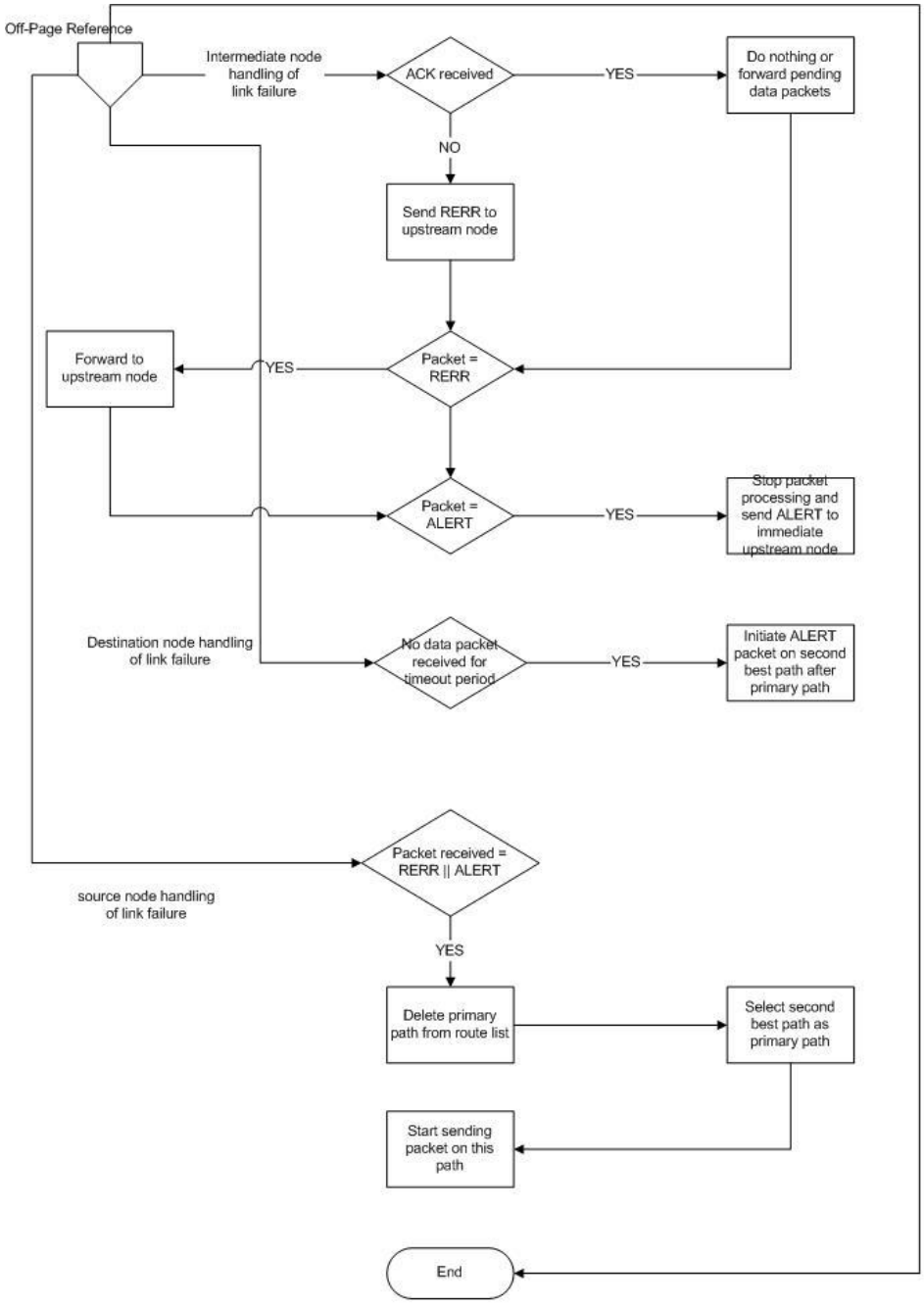

Figure A-4 Flow Chart (4/4)

REFERENCES

[1] "NS-2 Distribution" *http://www.isi.edu/nsnam/ns/*

[2] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing", in Proceedings of the ACM Conference, pp. 154-190, 1992.

[3] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), pp. 234–244, 1994.

[4] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), pp. 142–184, 1996.

[5] Brad Karp and H. T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks (2000)", in Proceedings of IEEE ICCCN'99, Boston, MA, pp. 54-70, Oct. 1999.

[6] Wen-Hwa Liao, Yu-Chee Tseng and Jang-Ping Sheu, "A Fully Location Aware Routing Protocol for Mobile Ad hoc Networks, (2001)".

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

[8] Thomas lausen, Philippe Jacquet, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum and Laurent Viennot, "Internet Draft, Optimized Link State Routing Protocol(1999)".

[9] Vincent D. Park and M. Scott Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks (1998)".

[10] David B. Johnson, David A. Maltz and Yih-Chun Hu, "Internet Draft, The Dynamic Source Routing Protocol (2000)".

[11] Sung Ju Lee, "Routing and Multicasting Strategies in Wireless Mobile Adhoc Networks" Ph.D. dissertation, Univ. of California, Dept. of Computer Science, 2000.

[12] A. Nasipuri and S.R. Das, "On-Demand Multi-path Routing for Mobile Ad Hoc Networks," in Proceedings of IEEE ICCCN'99, Boston, MA, pp. 64-70, Oct. 1999.

[13] S.-J. Lee and M. Gerla, “Split Multi-path Routing with Maximally Disjoint Paths in Ad hoc Networks,” in Proceedings of IEEE ICC 2001, Helsinki, Finland, pp. 3201-3205, June 2001.

[14] Mahesh K. Marina and Samir R. Das “On-demand multi-path distance vector routing in ad hoc networks (2001)”.

[15] Alvin Valera, Winston K.G. Seah and S.V. Rao, “A Highly-Resilient and Energy-Efficient Routing Protocol for Mobile Ad hoc Networks,” in Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002).

[16] Rohit Dube, Cynthia D. Rais, Kuang-yeh Wang and Satish K. Tripathi “Signal Stability based Adaptive Routing for Adhoc Mobile Networks (1998)”.

[17] “NS-2 notes and documentation” *http://www.isi.edu/nsnam/ns/ns-documentation.html*

[18] “NS-2 trace formats” *http://www.k-lug.org/~griswold/NS2/ns2-trace-formats.html*

[19] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, ISO/IEC 8802-11; ANSI/IEEE Std 802.11*, Aug.1999.

[20] *Development of WaveLAN, an ISM Band Wireless LAN, AT&T Technical Journal*, pp. 27-37, July/Aug. 1993.

BIBLIGRAPHY

Doe, John B. *Conceptual Planning: A Guide to a Better Planet,* 3d ed. Reading, MA: SmithJones, 1996.

Smith, Chris. *Theory and the Art of Communications Design.* State of the University Press, 1997

INDEX