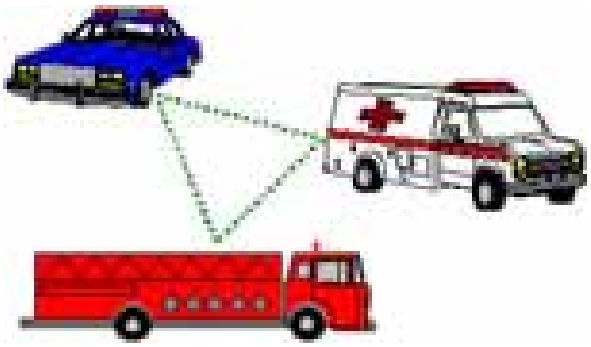
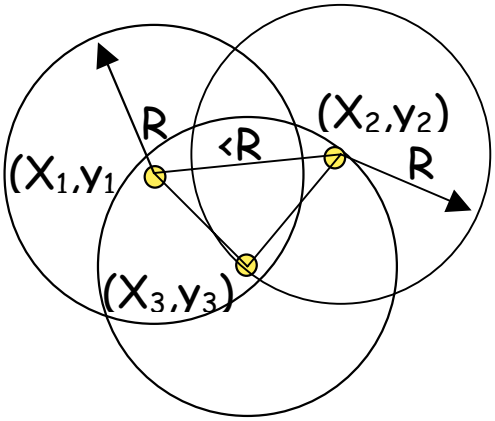
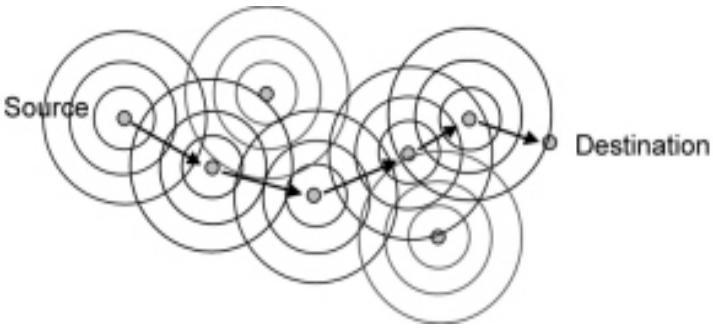
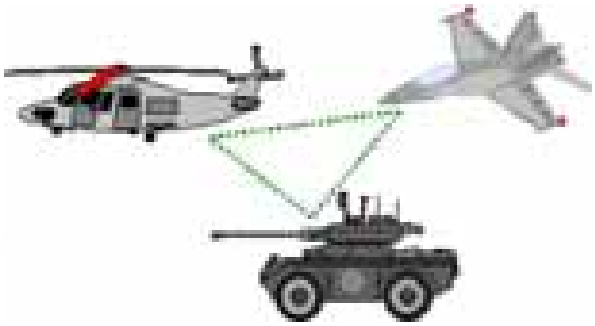
Wilfred Githuka Gikaru

“Mobility-based Routing Overhead Management In Reconfigurable Wireless Ad Hoc Networks”









Cuvillier Verlag Göttingen

"Mobility-based Routing Overhead Management In Reconfigurable Wireless Ad Hoc Networks"

Dissertation

zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

GIKARU, Wilfred Githuka
geboren am 26. Oktober 1966 in Kenia

Gutachter: Prof. Dr. rer. nat. habil. h. c. Alexander Schill (Technische Universität Dresden)
Prof. Dr. –Ing habil Klaus Karbitzsch (Technische Universität Dresden)
Dr. Miguel Sanchez (Technische Universität in Valencia, Spain)

Faculty Dean:
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Dresden im November 2004

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.
1. Aufl. - Göttingen : Cuvillier, 2004
Zugl.: (TU) Dresden, Univ., Diss., 2004
ISBN 3-86537-278-3

Gedruckt mit Unterstützung des Deutschen Akademischen Austauschdienstes (DAAD)

Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

1. Auflage, 2004
Gedruckt auf säurefreiem Papier

ISBN 3-86537-278-3

MOBILITY BASED ROUTING OVERHEAD MANAGEMENT IN RECONFIGURABLE WIRELESS AD HOC NETWORKS

Dissertation

Submitted at the Dresden University of Technology's faculty of computer science

In fulfillment of the requirements for the degree of Doctor of Engineering (Dr. -Eng)

and

Defended by M.Sc –Eng. GIKARU, Wilfred Githuka

Born on 26th October 1966

Main Advisor: Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Dresden in November 2004

DRESDEN UNIVERSITY OF TECHNOLOGY
Department of Computer Science
Chair of Computer networks

This Dissertation was submitted on

30th August 2004

And approved by

Reviewers:

Prof. Dr.rer. nat. habil. Dr. h. c. Alexander Schill (Main Supervisor)
Prof. Dr. –Ing habil Klaus Karbitzsch (Second Supervisor)
Dr. Miguel Sanchez (Guest from Polytechnic University of Valencia, Spain)

Other Commission members
Prof. Dr. –Ing Christian Hechberger
Dr. rer. Nat habil. Boris Flach

Commission Chairman
Prof. Dr.-Ing. Klaus Meissner

Faculty Dean:
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Defended on 9th November 2004

Acknowledgement

Working towards a doctorate is both intellectually and personally a great challenge. Although is comes with frustrations, doubts and regrets, the process is very rewarding. Experiences and discoveries that I encountered both academically and socially have enriched my life. It is very difficult to get to the finishing line alone. Support that I have received throughout the research has been of great help to me. I therefore recognize and thank the special people who have made it work. This work is dedicated to them.

First and foremost I would like to thank my mother Mrs. Mary Gacaki Gikaru who has always been there for me through out my life. Her encouragements and prayers have been my driving power. My two brothers David Gaceru Gikaru and Eliod Mbuthia Gikaru have also been very supportive. This includes Hilda who gave me a special present at a very crucial moment. The present was my beautiful daughter BriAnna Mary Gacaki Githuka. The girl gave me more reasons to fight to the very end.

I want to thank my academic advisor Professor Dr. Alexander Schill, who has guided me through the whole process since I enrolled for the program. His critics, advices, guidance and working atmosphere have made great contribution to my success. I look forward to working with him in future. My second advisor Professor Dr. Klaus Kabitzsch has also played a great role in the program. Special thanks also go to Kevin Goslar and all the other colleagues in the chair of computer networks for the much help they have accorded me through out my working at the department. Steffen Goebel and Worku Alemu have been of great help in my work especially with their programming skills. Thanks to Sven Buchholz who played a role in my initial settling and finding directions in my research. My other two external reviewers, Dr. Miguel Sanchez from Spain and Dr. Markus Borschbach from Muenster have been very supportive especially with their helpful remarks and commitments that allowed me to graduate on a tight time schedule. Thanks also go the members of the Graduate College, which I was a member. I cannot also forget to thank the graduation committee members for their great work in convening the defense out of their schedules. The list is long and cannot be completed on this dissertation.

Outside the academic circles, I wish to give special thanks to Maria Noel dos Santos from Venezuela and Andrew Fatufe from Nigeria who have been encouraging me when I felt the load being heavy on my shoulders. I cannot forget to thank My Pastor Paul Beduerftig, his wife JoAnn, Pastor Bill and other church members for their prayers and word of encouragements throughout my stay in Dresden.

Lastly and not least are very special thanks to the DAAD (Deutsche Akademischer Austausch Dienst) for their sponsorship of the whole doctorate program including the language course.

ABSTRACT OF THE DISSERTATION

Mobility-Based Routing Overhead Management in Reconfigurable Wireless Ad Hoc Networks

Dissertation topic by Gikaru Wilfred Githuka
Main Advisor: Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Routing Overheads are the non-data message packets whose roles are establishment and maintenance of routes for data packets as well as neighbourhood discovery and maintenance. They have to be broadcasted in the network either through flooding or other techniques that can ensure that a path exists before data packets can be sent to various destinations. They can be sent reactively or periodically to neighbours so as to keep nodes updated on their neighbourhoods. While we cannot do without these overhead packets, they occupy much of the limited wireless bandwidth available in wireless networks. In a reconfigurable wireless ad hoc network scenario, these packets have more negative effects, as links need to be confirmed more frequently than in traditional networks mainly because of the unpredictable behaviour of the ad hoc networks. We therefore need suitable algorithms that will manage these overheads so as to allow data packet to have more access to the wireless medium, save node energy for longer life of the network, increased efficiency, and scalability.

Various protocols have been suggested in the research area. They mostly address routing overheads for suitability of particular protocols leading to lack of standardisation and inapplicability to other protocol classes. In this dissertation ways of ensuring that the routing overheads are kept low are investigated. The issue is addressed both at node and network levels with a common goal of improving efficiency and performance of ad hoc networks without dedicating ourselves to a particular class of routing protocol.

At node level, a method hereby referred to as “link availability forecast”, that minimises routing overheads used for maintenance of neighbourhood, is derived. The targeted packets are packets that are broadcasted periodically (e.g. hello messages). The basic idea in this method is collection of mobility parameters from the neighbours and predictions or forecasts of these parameters in future. Using these parameters in simple calculations helps in identifying link availabilities between nodes participating in maintenance of network’s backbone.

At the network level, various approaches have been suggested. The first approach is the cone flooding method that broadcasts route request messages through a predetermined cone shaped region. This region is determined through computation using last known mobility parameters of the destination. Another approach is what is hereby referred as “destination search reverse zone method”. In this method, a node will keep routes to destinations for a long time and use these routes for tracing the destination. The destination will then initiate route search in a reverse manner, whereby the source selects the best route for next delivery. A modification to this method is for the source node to determine the zone of route search and define the boundaries within which the packet should be broadcasted. The later method has been used for simulation purposes.

The protocol used for verification of the improvements offered by the schemes was the AODV. The link availability forecast scheme was implemented on the AODV and labelled AODV_LA while the network level implementation was labelled AODV_RO. A combination of the two schemes was labelled AODV_LARO.

Simulation results showed that the both the "link availability forecast" and the network level scheme reduce the routing overhead at node and network levels, leading to improvement in efficiency, scalability and networks performance. The schemes also result to savings in energy on the nodes resulting to increase in the life of the network. An overhead reduction of over 90% on overheads due to hello messages and over 20% on the overall using the link availability scheme (AODV_LA) was achieved. The network level scheme (AODV_RO) offered a reduction of between 19 and 26% on the overall overheads. A combination of the two schemes (AODV_LARO) resulted in a reduction of between 35 and 38 %. The scenario used in the simulations was a node density of 60 to 80 nodes in an area of 1500 by 1000 m. A varying average node mobility used was between 2 and 16m/s while packet transmission rate choose was between 2 and 14 packets per second. There was a substantial improvement in network performance in terms of throughput, delivery ratio and delay for the chosen variables. There was also an improvement in the efficiency and scalability at high node density, mobility and transmission rates.

LIST OF DISSERTATION RELATED PUBLICATIONS AND SCIENTIFIC TALKS

Publications in International conferences:

W. Gikaru and A. Schill, "Location-based routing overhead reduction in manets*",* *Western simulation MultiConference (WMC'04),* San Diego, California, the USA. January 2004.

W. Gikaru and A. Schill, "Link Availability forecast in highly dynamic MANETs*",* *The Internationally Association of Science and Technology for development (IASTED), Technical Committee on Telecommunications*, Banff, Canada. July 2002.

Scientific talks:

"From Web Services to Grid Technology" as part of the dissertation at the Dresden University of Technology, November 2004

"A mobility-based Routing Overhead management for reconfigurable wireless ad-hoc-networks" at *Institute of mathematics and computer science, University of Münster,* Germany, February 2004.

*"*Mobile Wireless ad hoc networks*"* at *the graduate college, Department of Computer science, Dresden University of Technology*, February 2003.

*"*Adaptive protocols for Wireless Sensor Networks*"* at *the Institute of applied Computer Science, Department of Computer science Dresden University of Technology*, December 2001.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

1. INTRODUCTION

1.1 Reconfigurable Wireless Ad Hoc Networks

Among many possible ways of grouping, networks can be split into two main categories. The wired networks which use physical links or connections through wires, and the wireless networks, which make use of wireless links. The second category can further be sub-divided into infrastuctured, whereby there is a pre-existing infrastructure, and infrastructureless where the infrastructure is formed spontaneously when required "on the fly". The second sub-category is also referred to as ad hoc network.

Ad hoc networks are therefore self-organizing, rapidly deployable, and require no fixed infrastructure. They comprise of wireless nodes that can be deployed anywhere and must cooperate to dynamically establish communications using limited network management. Nodes in an ad hoc network may be highly mobile or stationary and may vary widely in terms of their capabilities and use.

One of the main objectives in designing recent network architectures is to achieve increased flexibility, mobility and ease of management relative to wired networks. In relation to infrastructure networks, ad hoc networks can be termed as "peer-to-peer" in the sense that all nodes have equal roles in terms of topology management i.e. the roles of base station and router are played by all nodes. These kinds of networks are highly dynamic and require adaptive control schemes that can respond to the high mobility and network changes without administrative intervention. The result of such schemes is what is referred to as "Reconfigurable Wireless Networks" (RWN)[1]. A more descriptive name to such a network with reference to this work would be "Reconfigurable Wireless Ad hoc Network (RWAN)".

Nodes in a RWAN dynamically join and leave the network frequently, often without warning and possibly without disruption of other nodes' communication. These nodes can be highly mobile and thus can rapidly change their constellation in presence or absence of a link. The main features in such networks are increased mobility, large number of nodes, and a large network span. For realization of such networks, certain requirements have to be met. These include robust routing and mobility management algorithms. Such algorithms are meant to increase the network reliability and availability. Adaptive algorithms and protocols for adequate adjustments to frequently changing radio propagation network and traffic conditions are also necessary. Low overhead algorithms and protocols that allow conservation of available resources and reduce congestion in multiple routes between source and destinations would allow longer survivability of the network while increasing its efficiency of information delivery. Algorithms that would reduce susceptibility to single point network failures and ease congestion around high level nodes thus increasing routing efficiency are also needed. In this dissertation, management of overheads due to routing, based on mobility parameters, has been addressed. Since routing packets are dispatched either with a single hop (only to the neighbors) or with multiple hops (to other destinations outside a node's transmission range), overhead

reduction has been considered at two levels. These are: node level (single hop packets) and Network level (multi-hop packets). The issue of routing overhead has been discussed in more details in chapters two and three.

1.2 Problem statement and Motivation

The issue being addressed in this dissertation is the problems associated with routing overheads in reconfigurable ad hoc routing protocols. Since routing overheads form the backbone of the connectivity and maintenance of an ad hoc network, they cannot be done without. However, better management of the overheads would lead to savings in the network's energy and bandwidth. This would also effectively increase the lifespan, efficiency, scalability and possibly the overall performance of both the protocol and the network.

In order to design an efficient algorithm or method of handling and managing these crucial overheads, it is important to understand a number of features associated with these overheads. The features that were analyzed leading to the problem statement of this dissertation are: The different forms of overheads that exist, how they are generated, how they are propagated, how they are used, where the different types are needed, what effects they have on different parts of the network among other important features. These features are addressed in more details in chapter two. The analysis hinted the points that while these overheads are crucial to the running of the network and the protocol, all the nodes in the network do not always need them. It is therefore necessary to design schemes that will allow the nodes to generate the overheads only when they are needed and direct them to the areas where they are required. This would allow the protocol to handle information packets more efficiently, use available resources more economically and increase both scalability and lifetime of both the protocol and the network without compromising their performances.

Although it has been suggested in literature that a "one fit all" kind of solution to the routing protocol problems is not feasible, a solution that handles one particular problem and overlooks others may not be ideal. The reason for such an argument is that real life situations, especially the current target application area of ad hoc networks (see chapter 5), are very dynamic in nature and require dynamic protocols that would self configure to cope with the changes experienced by the scenarios.
Another question to ask when solving routing problems would be, which routing protocol is most appropriate for the tasks at hand. In the view of this work, a full analysis of the existing types of routing mechanisms would be required before answering such a question. It is evident that the ideas contributed in this fast moving research area, have not been fully investigated and utilized. It is for this reason that this dissertation details designing of schemes that would take advantage of features of the existing protocols and add features that would benefit a large cross section of protocols.

This dissertation addresses two types of overheads that were found common in most of the traditional protocols. These are the neighborhood discovery and maintenance broadcasts (specifically the hello messages), and the route discovery messages (specifically the route request messages).

When hello messages are broadcasted periodically, this leads to poor knowledge of the neighborhood if the network is highly dynamic. A highly dynamic network here refers to a network that changes dynamically in terms of topology, traffic load, node density, among other network parameters. When these dynamics increase, the configuration of the network changes more frequently due to a more frequent break and make of links between nodes. These periodic updates of node identities lead to poor and inaccurate identification of its neighborhood. The scheme designed in this dissertation aims at minimizing sending these messages by sending them only when a node discovers that it is at a risk of loosing a link on an active route.

In case of the route request messages, the conventional method of broadcasting them is flooding. While this method ensures optimum selection of route, these messages are processed by all nodes in the network most of who should never have participated in the route creation. This leads to waste of crucial resources like bandwidth and energy. The scheme hereby proposed ensures that route requests are sent only to the regions where the route is likely to be found. While this scheme introduces other types of packets, the overall effect is reduction of the routing messages and avoiding areas where the messages would otherwise lead to waste of resources.

The understanding that better management of the overheads can lead to savings in reconfigurable wireless ad hoc network resources motivated us to this research.

1.3 Objectives

Ad hoc networks emerged to increase flexibility in the mobile computer field. These networks are considered suitable for applications where there is no pre-existing infrastructure or an existing one has been disabled due to some reasons. Such a network may be static, or dynamic. There has been tremendous success in the static network area. However the dynamic network continue experiencing challenges which have not been easy to alleviate. Notable challenges range from network performance to protocol efficiency that includes low overheads. These challenges are discussed in more details in chapter 2. They are unique in wireless networks unlike in wired network. The limited wireless channel and the mobile nature of nodes are probably the basic challenges. The mobility of nodes may result in continuous and unpredictable link breakages between participating nodes, which may lead to network fragmentation. Such link breakages are normally associated with increase in network overheads as nodes try to repair the broken links or establish new links. If these breakages can be foreseen, such undesired consequences can be avoided.

Our main objective in this research is to identify various features in existing routing protocols and exploiting underutilized features that can be used in reduction of overheads generated by RWAN routing protocols during normal network operations. Routing Overheads (RO) that are normally generated at two levels will be handled separately. At node level, the overheads generated only affect the node and its neighbors and has no direct impact on the entire network. While this has an indirect impact, it would however have undesired consequences if it happens more frequently on busy routes or many parts of the network. The packets involved in this level are mainly neighbor discovery and maintenance packets, which are normally broadcasted

with hop counts limited to node's neighbors. This problem is addressed with the "link availability forecast" scheme. At Network level, overhead packets are generated either through flooding or broadcasted possibly with multiple hops. The packets are mainly route discovery packets, maintenance packets, error packets, or acknowledgement messages. Since these packets pass through a great portion of the network, they have more impact on the operation of the entire network than individual nodes. These are addressed with the network level routing overhead management schemes. Reliability improvement methods on the suggested schemes have also been addressed in chapter three.

To achieve the main objective in this research, merits of certain routing mechanisms have been extracted. This has been done with the motivation that a better understanding of relative merits serves as a cornerstone for development of more effective routing protocols for mobile ad hoc networks that adapts to network dynamics and other parameters. It has been identified in this research that neighborhood discovery algorithms based on nodes' history of movement that will allow routing protocols to make intelligent decisions about routing and network level routing overhead reduction schemes would greatly benefit the research for more efficient routing protocols. Such schemes will allow the ad hoc networks to experience reduced overheads both at each node's neighborhood and network level since the nodes will be sending overhead packets only when it is vital and to regions requiring them. This would also result to an increase in delivery ratio since packets will be sent with relatively higher certainty of delivery. The overall throughput is expected to increase because time will be saved though reduction of the number of route requests and reduction of congestion of packets. A combination of parameters necessary for deciding the most appropriate route to take will be investigated and an algorithm on how to effectively use such parameters derived. In the "link availability" scheme, parameters that are needed to establish whether a link will be available between two communicating nodes before they can start transmitting information packets are derived. These parameters help in the forecast of the reliability of the entire path that the packets will follow so as to guarantee improvement in their delivery. Parameters needed in the case of network level RO reduction are also derived.

1.4 Related work

The idea of link and path availability is not absolutely new to the field. It has however not received full exploitation and detailed analysis that it deserves. Moreover the objective of the concept has not been RO reduction but knowledge of neighborhood for correct route selection. Researchers have made suggestions in the past on ways of predicting availability of a link between communicating nodes. An example is the use of mobility prediction used in the multicast ad hoc routing protocol [3]. In this paper, further analysis on the effects of unpredicted motion parameter change (e.g. speed and direction) should have been considered in more details. In this dissertation, motion parameters have been considered in terms of epochs, minimizing the error that could arise from unpredicted change in motion characteristics. The author of the referred work [3] mentions that there are other possible methods for future research. Modifications have hereby been suggested in order to take care of an extended number of parameters for consideration in link availability forecast. Other similar work done in this area include that on probabilistic link predictions. In the paper [4],

the author concludes with the remarks that "the expressions for link availability provide the basis for a novel routing metric". These remarks actually support the need for our research. The author gives an evaluation, exposing the need for knowledge of link availability forecast.

On route availability, there have been suggestions on independent metrics that should be used in determining the validity of a path. A good example is the analysis done by Bruce [5] in the paper on path availability. All the suggested methods seem to favor a particular feature e.g. one method may favor shortest path, another may favor most economical path in terms of power consumption etc. Harmonization of multiple parameters for a more comprehensive conclusion of the best route has been done in this research and presented in chapter three.

On Network level reduction of routing overheads, certain algorithms have been suggested in the location based protocol area. The Location Aided Routing (LAR) protocol [6] uses an approach similar to our first scheme (link availability forecast) but differs in the restriction of the expected zone. The definition of R (radius of expected region) in LAR also differs since it is based on the time differences and node movement while the "R" in this research is based on specific locations known from historical behavior of the node (reactively passed on by the destination) plus its radius. Greedy Perimeter Stateless Routing (GPSR) [7] is another geo-protocol with features that offer reduction of overheads. This protocol makes forwarding decisions using only information about a router's immediate neighbor in the network topology or the region's perimeter. The schemes hereby developed use information obtained from the destination node and information gathered by the "DestSearch" packet along an old route. The Optimized Link State Routing (OLSR) Protocol [8] uses the multipoint relay (MPR) method for reduction of message overheads as compared to flooding method by limiting flooding to MPRs only. The scheme in this thesis reduces flooding effects by limiting flooding to a predetermined region.

1.5 Scope and limitations

The Scope of coverage in this research is mainly dictated by time and the availability of resources for testing the algorithms designed. As mentioned in section 1.2, there are numerous types of overheads that contribute to the overall routing overheads. Only two types (hello and route request) are however investigated in details and tested in this dissertation. Other vital overheads like the negotiation packets have been left as a future research issue. However the results of the research gives the evidence needed to support the solutions proposed. It is worth mentioning at this point that the results are based on the network's mac and routing layer and little reflection at the application layer.

This research details the routing overhead. While we feel the importance of optimization and/or preservation of optimal routes this topic is outside the scope and objectives of this particular research. We however suggested ways way of ensuring that established route do not exceed reasonable lengths.

The tool used for testing the schemes is the ns2 simulation tool which is a freeware tool used by many universities for research purposes only. It is however a new tool and weak in some real world application features. It however offers a degree

of reliability on the features that has already been configured for. The fact that it is used across the digital divide leads to lack of proper documentation making it rather difficult to add features that are not modeled in the tool. This at times leads to frustrations and time consumption. An issue of particular interest in this research would have been the energy issue. While it is sufficient to believe that less processing of packets results in less energy consumption, it would have been clearer if this were demonstrated through simulations. This would have required modification of the tool, which would require more detailed study of the tool and thus more time. Another obstacle experienced by the simulation tool is that it is not transparent across all the layers. This means that it is not possible to track all the information (data) packets as the move along the routes. This lead to difficulties in monitoring the packets held up in queues on particular nodes and the once dropped by the queues. The result was low delivery ratios when the number of nodes in the network was increased. The delivery ration observed in the experiments was thereby used only for comparison purposes.

A research on suitability of simulation tool for various implementations is required to save researchers time by using tools that require minimum or no modifications. Moreover, only a few ad hoc routing protocols have been fully implemented in the tool. This makes it difficult to have a full comparison of the upcoming schemes with the existing ones. The research area is however relatively new and no tool so far incorporates all necessary features at the time of doing these simulations.

1.6 Organization of the dissertation

The rest of the dissertation is organized as follows: Chapter two gives a general overview of Ad hoc routing protocols and a review of the topic area. It starts with a brief overview of ad hoc routing protocols, extending to their features, challenges, requirements and characteristics. The chapter then looks into the basic classification and analysis of the protocols, with more details given on the classes more relevant to this research. A comparison of some current routing protocols is done analytically. The problems being experienced currently and needing attention are highlighted. An overview of the topic under research is also given in this chapter followed by other issues in the area. At the end of the chapter, performance evaluation is done on selected protocols for justification of the problem status.

In chapter three, the main ideas developed in the dissertation for handling the problem are discussed in details. The link availability forecast (node level RO reduction scheme) and Network level RO reduction schemes are analyzed. In this chapter, the basic theory and ideas leading to the development of the schemes are discussed. Integration of the scheme into the routing scenario is also done. Route reliability and topology maintenance strategies are also explained.

Chapter four gives an evaluation of the proposed schemes on a typical protocol through simulations and analysis of results generated. Simulation results and justifications of the results are given towards the end of the chapter followed by conclusions drawn from the experiments.

In chapter five, implementation in common ad hoc routing protocols with features of interest to this topic is discussed followed by application classes and areas.

Chapter six gives the dissertation's summary, conclusions and focus on future work. This is followed immediately by a list of references.

The appendix and references are given towards the end of the dissertation. The first appendix gives a list of abbreviations used in this dissertation followed by a description of the simulation tool used in our experiments (ns-2) in the second appendix. Appendix 3 gives tcl implementations used in the configuration of the simulation. This is followed by more simulation results tables and relevant program codes.

CHAPTER TWO

2.0 GENERAL OVERVIEW AND PROBLEM AREA

Meeting the major requirements in designing Reconfigurable Wireless Ad hoc (RWAdhoc) routing protocols is a challenging task. Many proposals have been suggested on how best to design a universal protocol [9], [10], [11] but no consensus has been reached so far. The challenge in designing a routing protocol for the ad-hoc communication environment stems from the fact that, on one hand, to determine the packet route, at least the reachability information of the destination nodes that neighbors need to know in order to successfully forward packets towards the destinations. On the other hand this topology may change quite often thus invalidating the information. In this chapter, an overview of most predominant routing protocols is given. The features and characteristics of the protocols as well as mobility patterns of the RWAdhoc nodes are also given. These features and characteristics form the basis of the classification. Routing overheads in RWAdhoc Nets and other issues in ad hoc routing are also discussed. This chapter gives a comparison two routing protocols selected from the most common routing protocols. The comparison is based on analytical and experimental results followed by conclusions that lead to motivation of this research work.

2.1 Features and Characteristics of RWAdhoc routing protocols

Certain critical network features determine the design of a routing protocol for the efficiency and effectiveness of an RWAdhoc network. These features are categorized into qualitative and quantitative features.

2.1.1 Qualitative and Quantitative features

One qualitative feature hereby considered is the knowledge of a node's location. It is determined whether the algorithm used needs to have local or global knowledge of the network. It is then determined which neighbour discovery scheme is suitable for the protocol in use. Another feature to consider is the effect of topology changes. Here it is determined whether the routing algorithm needs complete restructuring or only incremental changes. The most suitable topology maintenance schemes is then identified. Adaptation to radio communication environment is another feature to consider. A decision is made on which among the available communication schemes would be the most appropriate. Normally, selection is made from fading, shadowing and multi-user interface on links schemes. Power consciousness is also an important feature especially in energy critical environments like sensor networks and battery driven devices like in emergency or other hostile situations. Here, methods of controlling power consumption e.g. having uniform distribution of participation and/or implementation of node sleeping mechanisms are considered. Link orientation is another feature to consider. Here a determination of whether the routing algorithm performs efficiently on unidirectional links e.g. if bi-directional links become unidirectional is done.

Quantitative features are mainly time dependent. One feature to look into is the networks settling time i.e. the time required for a collection of mobile devices to automatically organise themselves and transmit the first packet reliably. Other quantitative features are the networks join/departure time, i.e. the time required for an entering node or node group to become integrated into the RWAdhoc network and the time the network needs to reorganize itself after a node leaves the network. Network recovery time is also as important. This is the time required for the network to resume operation after a collapse due to fragmentation, traffic overload or other failures. Time independent quantitative features include memory requirements and network scalability. These involve storage requirements for storing tables and determination of the number of nodes that the RWAdhoc network can scale to and reliably preserve communication respectively.

2.1.2 Mobility patterns of Reconfigurable wireless Ad Hoc Nodes

Mobility patterns experienced by different RWAdhoc nodes may be different. These patterns are highly dictated by the environment. People waiting in an airport lounge may have different patterns from drivers of city taxis. Similarly children playing have different patterns from military troops in battlefields. Some patterns are more aggressive than others and require real time features, e.g. video and audio transmissions. Some patterns however share characteristics, for example, military troops and law maintenance policing. Speeds may vary from one application to another. Mobility predictability also varies from one pattern to another. It is therefore important to study the mobility patterns in order to predict future state of the network topology, predict routing reconstruction, eliminate transmission of unnecessary control packets, reduce routing overheads, minimise disruptions and improve routing performance.

2.1.3 Major requirements and challenges in RWAdhoc routing protocols

One of the most important requirements to be considered when qualifying any protocol is its efficiency. With reference to RWAdhoc networks, a number of issues have to be considered. First, we have to consider the networks' ability to deliver sufficient amount of information within a pre-determined period of time (throughput). We also have to consider the ratio of information delivered at the destination to the information dispatched at the source (delivery ratio). Another item to check is the relationship between the number of information packets delivered at various destinations to the total amount of packets generated in the network (bandwidth use). Checking of the overall performance of the network relative to other known networks may also be required.

Various factors contribute to the efficiency of a RWAdhoc network. These include, but not limited to, the algorithm used in the networks routing protocol, the physical characteristics of the network, topological features of the network, among others factors. In this dissertation, we consider the routing protocols factor.
For a protocol to be considered efficient, it should have the following characteristics:

- It should adaptively self-tune its parameters for increased relative work speed and regional workload (self-configurable).

- It should use resources smartly, such that it can create a longer or an intelligent route for providing good power usage distribution, considering delay and other possible constraints.
- It should use less control messages and should be both intelligently proactive and at the same time reactive.
- It should have mechanisms for fair distribution of load and optimizing routes.
- It should be aware of the transmission protocol running on top of it, or vice versa.
- It should handle real-time traffic, with robust and quick route-maintaining techniques, which could include the hand-over technique similar to the once used for cellular networks.
- It should use a special addressing which is suitable for separation and merging of ad-hoc networks.
- It should provide quality of service, which should be embedded in the routing protocol.
- While doing all above jobs, it should be fast.

The design of a routing protocol that meets the above factors is a big challenge, especially in RWAdhoc networking scenario. This is due to various limitations found in wireless unlike in wired networks. These limitations include limited wireless transmission range, battery constraint of wireless nodes, and security issues among other limitations. The limitations are even more complex when the nodes in the wireless are mobile. This adds other challenges like packet loss due transmission errors and mobility, hidden terminal problem, potential frequent and unpredicted network partitioning, to mention a few. The Algorithms designed in this research address the first five characteristics with strong emphasis to the third issue: namely low overheads.

2.2 Classification

RWAdhoc networks can be classified using various parameters derived from the features and characteristics discussed above. Various methods of classification have been suggested in the literature [12], [13], [14], [15]. One way is to classify them according to the network topology that they form i.e. hierarchical whereby the network is organized in form of a tree vs. flat whereby the nodes are independent of one another. This classification is typically physical. When we consider the routing protocols that configure and maintain these networks, classification can be done according to the way they establish routes to the destinations. Here we can group them into proactive (sometimes called table driven or periodic) where the nodes keep a record of the whole network through periodic updates of routing tables or reactive (sometimes called on-demand) whereby routes are created on demand i.e. when a node wishes to send a packet to the destination. We can also group them as hop-by-hop where routers choose on next hop forwarding advertising only the path that they are using. Source routing is an alternative to hop-by-hop routing whereby, routes are determined by the originator of the packets.

Another way to differentiate the protocols is grouping them accordingly for symmetric and asymmetric networks. Symmetric networks are ones where all the nodes in the network have equal capabilities and share similar responsibilities. In the

asymmetric networks the capabilities of different nodes such as transmission ranges, processing capacities, speed of movement etc., and their responsibilities such as the ability to route etc. vary from node to node. Networks can also be classified according to the kind of traffic that the nodes are expected to carry. Traffic can be best effort data or real time data for multimedia applications such as voice and video.

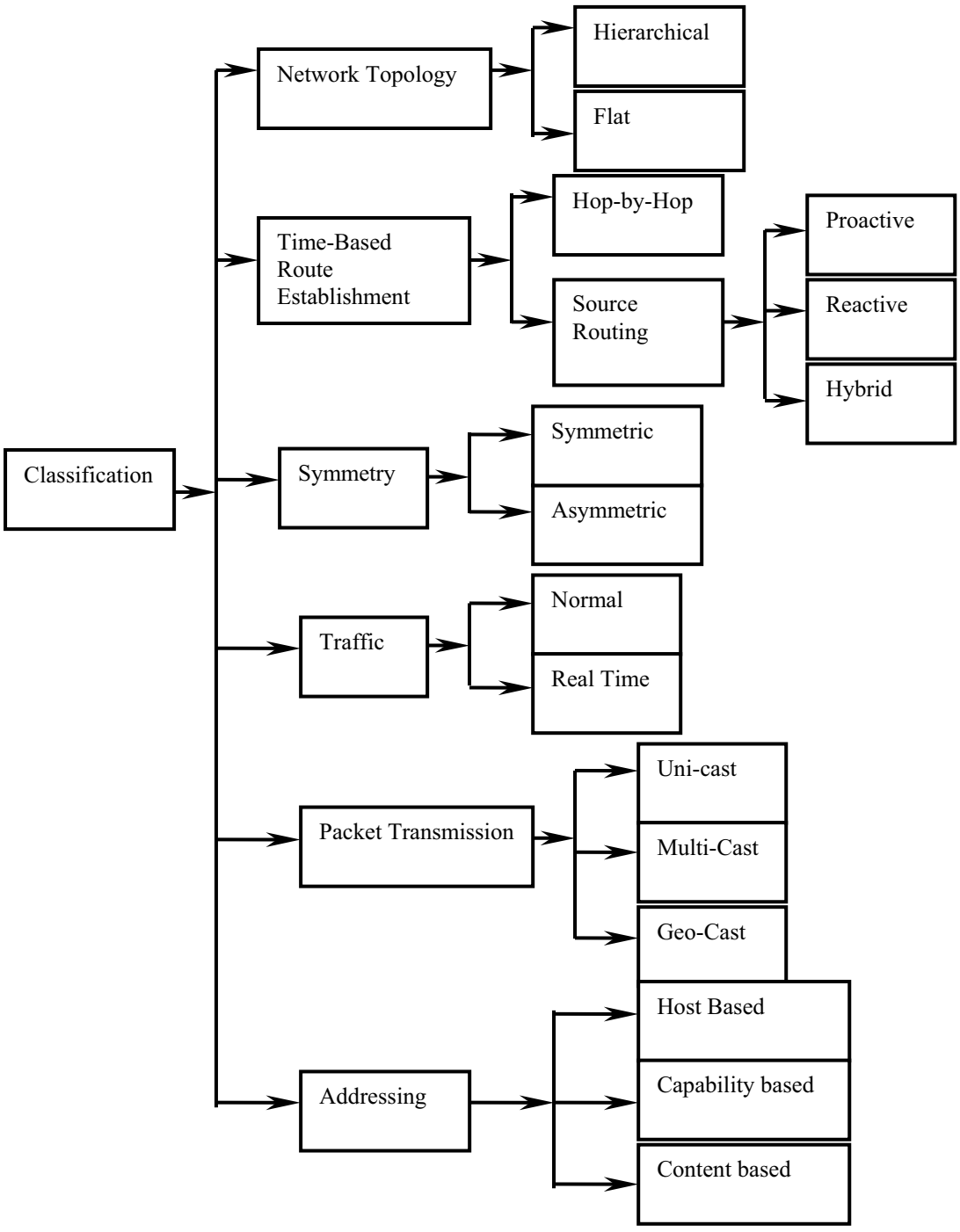


Figure 2.1 *General classifications of RWAd hoc routing protocols*

The schemes and protocols used at different layers in the nodes are typically modeled to suite the traffic being carried. Routing methods employed in the various networks may be different. Broadly classifying the routing may be unicast, multicast, or geocast. The addressing schemes may vary. Addressing schemes may be Host based or content based or even capability based. There may be other metrics such as bit rate, time constraints and reliability requirements base on which the networks can be differentiated. Figure 2.1 shows a summary of this general classification.

In this chapter, one of the classifications, which is more relevant to our goals is considered. This is the time dependant route establishment classification. An overview of the most common protocols is given, where by strengths and weaknesses are given. Suggestions to improvements are also given. A summary of classification of some of the common routing protocols is given later in the chapter (Figure 2.6).

2.2.1 Proactive Routing Protocols.

Proactive routing protocols learn the relevant topology before forwarding route requests. They therefore need to have knowledge of the entire network beforehand. Each node in the network maintains a vector for each destination and forwarding information for the destination. This information is periodically broadcasted to the neighbors. Following are examples of some common proactive routing protocols.

2.2.1.1 Destination Sequence Distance Vector (DSDV)

Destination Sequence Distance Vector [16] is a table-driven (proactive) distance vector protocol based on "*Bellman-Ford*" Routing Algorithm (An algorithm used for solving the shortest path problem i.e. shortest path between two points). This protocol sends packets periodically or when triggered by a change in the topology. The periodic nature of this protocol makes it suffer from its lateness in updating of the routing table especially during high mobility. These updates are either incremental updates or the whole routing table updates. Sequence numbers are used for loops' avoidance and for achievement of shortest routes. The route with greatest sequence number or lowest merit (for routes with similar sequence numbers) is preferred. When a route between two hosts is broken, hosts send route updates to their neighbors with infinite metrics and with uneven sequence numbers (sequence numbers for unbroken routes are normally even). This mechanism again helps in avoiding loops in the network. Two routing tables are used in this protocol: a forwarding table and a broadcast table. The forwarding table maintains the complete list of addresses of all other nodes in the network. The broadcast table contains the setting time data for each destination node used to determine the time for the update advertisements. The routing- and update-packets between nodes are based on these tables. Along with the nodes' address, the routing table also contains the address of the next hop, route metric, destination sequence number, etc. DSDV is bi-directional in nature, which unavoidably has the unidirectional links problem in RWAdhoc networks.

Another problem with this protocol is the accumulation of packets on the interface queues during high network speeds. This leads to packets dropping. Similar dropping may occur due to protocol control message and MAC control message collision.

2.2.1.2 Global State Routing (GSR)

Global State Routing (GSR) [17] is somehow similar to DSDV described above. It takes the idea of link state routing [18], [19] and improves on avoiding flooding of route messages. In this algorithm, each node maintains a neighbor list, a topology table, a "next hop table" and a "distance table". "Neighbor list" of a node contains the list of its neighbors (all nodes that can be heard by a node are assumed to be its neighbors). For each destination node, the "topology table" contains the link state information as reported by the destination and the timestamp of the information. For each destination, the "next hop table" contains the shortest distance to each destination node.

The routing messages are generated when a link changes as in link state protocols. On receiving a routing message, the node updates its topology table if the sequence number of the routing message is newer than the sequence number stored in the table. After this, the node reconstructs its "routing table" and broadcasts the information to its neighbors.

An improvement of this protocol is the Fisheye State Routing (FSR) protocol [20]. In this protocol the update message size is smaller since it does not contain information about all nodes. It exchanges information about closer nodes more frequently than about farther nodes. This means that the accuracy of a neighbors' information decreases with increase in distance from the node. The protocol scales well to large networks as the overhead is controlled in this scheme.

2.2.1.3 Cluster-head Gateway Switch Routing Protocol (CGSR)

Cluster head Gateway Switch Routing Protocol (CGSR) [21] uses the basics of the DSDV described above. Here, mobile nodes are aggregated into clusters and cluster-heads are elected. All nodes that are in the communication range of the cluster–head belong to its cluster. A gateway node is a node that is in the communication range of two or more cluster heads. In a dynamic network, cluster head scheme can cause performance degradation due to the frequent cluster-head elections, so CGSR uses a Least Cluster Change (LCC) algorithm. In LCC, a cluster head change occur only if a change in the network causes two cluster-heads to come into one cluster or one of the nodes move out of range of all cluster heads.

2.2.1.4 The Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) [22] is a table-based distance vector routing protocol. This protocol models a network as an undirected graph represented as *G (V, E),* where *V* is the number of nodes and E is the number of links connecting the nodes. It can therefore be considered as distributed in nature. Each node in the network maintains a

Distance Table, a Routing Table, a Link Cost Table, and a Message Transmission List. The Distance Table of a node *i* contains the distance of each destination node *j* via each neighbor N of *i*. It also contains the downstream neighbor N through which the path is realized. The Routing Table contains the destination's identifier, the distance of each destination node *j* from node *i*, the predecessor of *j* (p^i_j), and the successor of the node *i* (s^i_j) on the chosen shortest path. It also contains the tag to identify if the entry is a simple path, loop or invalid. Storing the predecessor and the successor in the table is essential in detecting loops and avoid counting-to-infinity problems. The protocol therefore ensures loop freedom but does not allow multiple roots to destinations. The "link-cost table" contains cost of link of each neighbor of the node and the number of timeouts since an error free message was received from that neighbor. The Message Retransmission List (MRL) contains the information to let a node know which of the neighbors has not acknowledged its update message and retransmits the update message to that neighbor.

Nodes exchange routing tables with their neighbors using update messages periodically as well as on link changes. The nodes present on the response list of update message (formed using MRL) are required to acknowledge the receipt of update messages. If there is no change in routing table since the last update, the node is required to send an idle Hello message to ensure connectivity. On receiving an update message the node modifies its distance table and looks for better paths using new information. Any new path so found is relayed back to the original nodes so that they can update their tables. The node also updates its routing table if the new path is better than the existing path. On receiving an ACK, the node updates its MRL. A unique feature of this algorithm is that is checks the consistency of all its neighbors every time it detects in link of any of its neighbors. This further eliminates looping situations in a better way and also has fast convergence. It is important to note here that the protocol only supports bi-directional links since a link will only be up after confirmation by a positive acknowledgement message. The issues of QoS and security are not addressed in this protocol.

2.2.1.5 Optimized Link State Routing Protocol (OLSR)

OLSR [8] is a proactive routing protocol for mobile RWAdhoc networks. The protocol inherits the stability of a link state algorithm and has the advantage of having routes immediately available when needed due to its proactive nature. OLSR is an optimization of the pure link state protocol, tailored for mobile ad hoc networks. Firstly, it reduces the size of the control messages: rather than declaring all links, a node declares only a subset of links with its neighbors, namely the links to those nodes that are its Multi-Point Relay (MPR) selectors. MRP is a set of selected nodes in a node's symmetric neighborhood that may retransmit its messages. Secondly, OLSR minimizes flooding of control traffic by using only MPRs to diffuse its messages. This technique significantly reduces the number of retransmissions in a flooding or broadcast procedure.

OLSR may optimize the reactivity to topological changes by reducing the time interval for periodic control message transmission. Since OLSR keeps the routes for all destinations in the network, the protocol is beneficial to traffic patterns where a large subset of nodes are communicating with another large subset of nodes and where the

(source, destination) pairs are changing over time. The protocol is particularly suited for large and dense networks, as the optimization done using the MPRs works well in this context. OLSR is designed to work in a completely distributed manner and thus does not depend on any central entity. It does not require reliable transmission for control messages: each node sends control messages periodically and can therefore sustain an occasional loss of some such messages. Such losses occur frequently in radio networks due to collisions or other transmission problems.

This protocol does not require sequenced delivery of messages. Each control message contains a sequence number that is incremented for each message. Thus the recipient of a control message can easily identify which information is newer which helps in avoiding loops.

OLSR provides support for protocol extensions such as sleep mode operation (for power saving), multicast-routing, support for uni-directional links, auto-configuration/address assignment etc. Such extensions may be introduced as additions to the protocol without breaking backwards compatibility with earlier versions. The protocol performs hop-by-hop routing, i.e. each node uses its most recent local information to route a packet. Hence for OLSR to be able to route packets, the frequency of control messages should be tuned to the speed of the mobile nodes such that their movements can be tracked by their neighborhood. This may result to overheads' increase and subsequent loss in performance. OLSR has an advantage that it does not require any changes to the format of IP packets. This protocol would benefit from an intelligent way of sending the hello messages.

2.2.2 Reactive Routing Protocols

Unlike proactive protocols, reactive protocols become active when forwarding requests arrive or when a need to send a packet to a destination arises. In this case a source has to establish a route to the destination. These protocols maintain a route as long as it is in use or is frequently used. If the network topology changes, the route has to be repaired. Following are examples of the most common and current reactive routing protocols.

2.2.2.2 Temporary Ordered Routing Algorithm (TORA)

Temporary Ordered Routing Algorithm [23] is a reactive type of protocol. It is a distributed source initiated routing protocol based on link reversal algorithm. This protocol is designed to minimize the reaction to topological changes. Its messages are localized to a small set of nodes near the occurrence of topological change. Its operation can be categorized into four main sets: routes creation, routes maintenance, routes erasing, and routes optimization. These are associated with four messages: query (QRY), update (UPD), clear (CLR) and optimization (OPT). A host launches route creation with the QRY broadcast containing the node ID of the destination. When QRY packets reach a node, which has a height value of the destination (height here refers to level numbers assigned to nodes so that data flows in a water flow analogy), a UPD is sent as a response with the height of the node attached to the packet. The offset of the packet is incremented

in the receiving node and sent to the neighbors. In this way a direct centric graph is constructed from the source to the destination. When there are changes in the topology, a new reference level is generated and forwarded to the network. When routes are no longer valid, they are erased using the CLR message. One disadvantage of routes created by this protocol is that the routes are not optimal. They are however guaranteed to be loop-free.

The periodic nature of updates of TORA involving multiple routes updates leads to large control messages overheads when the speed of the network's topology change is high. The protocol would therefore be improved by an intelligent way of sending the updates.

2.2.2.3 Signal Stability-based Adaptive / Signal Stability Routing (SSA/R) protocol

Signal Stability-based adaptive Routing protocol (SSA alt SSR) [24] is an on-demand routing protocol that selects routes based on the signal strength between nodes and a nodes' location stability. This route selection criterion has the effect of choosing routes that have stronger connectivity. SSR comprises of two corporative protocols: the Dynamic Routing Protocol (DRP) and the Static Routing Protocol (SRP).

The DRP maintains the Signal Stability Table (SST) and the Routing Table (RT). The SST stores the signal strength of neighboring nodes obtained by periodic beacons from the link layer of each neighboring node. Signal strength is either recorded as a strong or weak channel. A problem here may arise due to lack of guarantee that a signal will remain strong over the entire transmission since the strength is not shown. Another problem is that this information is only received during information transmission leading to a poor forecast if no information message is transmitted for a long duration of time. A solution to this would probably be incorporation of hello messages loaded with the signal strength. All transmissions are received by DRP and processed. After updating the appropriate table entries, the DRP passes the packet to the SRP. The SRP passes the packet up the stack if it is the intended receiver. If not, it looks up the destination in the RT and forwards the packet. If there is no entry for the destination in the RT, it initiates a route search process to find a route. The route request packets are forward to the next hop only if they are received over strong channels and have not been previously processed (to avoid looping). The destination chooses the first arriving route-search packet to send back as it is highly likely that the packet arrived over the shortest and/or least congested path. The DRP reverses the selection route and sends a route-reply message back to the initiator of the route-request. The DRP of the nodes along the path updates their RTs accordingly. Route-search packets arriving at the destination have necessarily arrived on the path of the shortest length because the packets arriving over a weak channel are dropped at intermediate nodes. It may however not be the route of strongest stability. If the source times out before receiving a reply, then it changes the route quality field (PREF field) in the header to indicate that weak channels are acceptable. This is because these may be the only links over which the packet can be propagated. When a link failure is detected within the network, the intermediate nodes send an error message to the source indicating which channel has failed. The source then sends an error message to

notify all nodes of the broken link and initiates a new route-search process to find a new path to the destination. This is done for route maintenance. There are therefore no periodic broadcasts in this protocol. This protocol seeks to reduce the duration of discovered routes. It offers a saving of up to about 60% on the number of route reconstructions according to Rohit Dube [12] over protocols without location stability.

This protocol would benefit from inclusion of a link stability parameter, included in our scheme, which includes the signal strength over time. This parameter needs to be reactive in nature so as to give more accurate link stability. The protocol does not address power saving issue. The periodic way of sending the signal stability beacons causes inaccuracy that can be improved by sending the beacons more intelligently. Further savings in broadcasting of packets can be done by reducing the zones of broadcasts.

2.2.2.4 On-Demand Multicast Routing Protocol (ODMR)

On-Demand Multicast Routing Protocol (ODMRP) [25] is a mesh-based protocol that uses a forwarding group concept. It uses a soft state approach to maintain multicast group membership. Here unlike the other reactive protocols, no explicit control message is required to leave the group. In this scheme, group membership and multicast route establishment are done on an on-demand basis. Join-query packets are broadcasted to the entire network when there is no multicast group to which a multicast source wants to send packets. This packet is also used for updating of routes whereby it is periodically broadcasted. It would be desirable to avoid periodic broadcasting packets that might not be of immediate use since these packets might take up bandwidth that would otherwise be used by information packets.

When an intermediate node receives a join query, it checks for any duplication and stores the packets ID and sequence numbers in its cache. The routing table is updated for reverse route learning. This message is then re-broadcasted if necessary. The message should be a non- duplicate and have a time to live (TTL) greater than zero.

On reaching the multicast destination, a “join Reply” message is broadcasted to the receivers’ neighbors. A node receiving the join reply packet sets its group forwarding flag (GF_FLAG) if its ID is the next hop ID on the packet. It then broadcasts its own Join Table built upon matched entries. Next hop information is extracted from the routing table. Each forwarding group member propagates a Join Reply until it reaches the multicast source via the selected shortest path. This forms a mesh of nodes called a “forwarding group”. A multicast source can then use these routes to multicast packets to receivers. A multicast source will periodically broadcast the join query packets as long as they have data packets to send. This helps in refreshing the forwarding groups and routes. This kind of periodic broadcast may cause congestion if a certain route is busy. Multicast data packets are forwarded only if they are not duplicates and the FG_FLAG for the multicast group is not expired. This helps in reducing traffic overhead and avoiding of stale routes. When a multicast source wants to leave a group, it does not need to send explicit control packets. It simply stops broadcasting the join query messages. Similarly, when a receiver no longer wants to receive from a particular multicast group, is stops

sending join reply to that group. Here, nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no join Tables received) before they timeout.

The problem identified here of periodic broadcast of unnecessary query packets can be reduced by having an on demand method of broadcasting the queries. It is this on demand feature that is implemented in schemes developed in this research.

2.2.2.5 Dynamic Source Routing (DSR)

Dynamic Source Routing [26] is an on-demand routing protocol that is based on the concept of source routing. Mobile nodes using this protocol maintain route caches that contain the source routes of which the mobile node is aware. Entries in the route cache are continually updated as new routes are learnt. Each packet in this protocol has to have a list of all hops that the packet goes through before reaching the destination. DSR consults the routing cache to obtain paths for packets waiting to be transmitted. If there is no entry for the desired route, a route discovery process is initiated. In this case, the neighboring hosts record their addresses in the packet before passing it over if they don't have knowledge of the destination.

The two major phases in DSR are route discovery and route maintenance. Route discovery is initiated when a node wishes to send a packet to a destination whose route is not available in its routing cache. Route maintenance here is done using error packets. Every node is responsible for successful delivery of packets to the neighbors. If it detects that the packet was not delivered, for example by relying on link layer protocol or passive acknowledgement, it sends an error packet to the source for another route discovery to be initiated. Since DSR's successful packet delivery is directly related to generation of control messages, which increases with increase in speed, this protocol is not suitable in high mobility environments.

The algorithm begins with establishment of a need to send data packets. It then consults node's cache for an entry of a route to the destination and uses such a route if it exists otherwise it broadcasts discovery packets. This packet is refereed to as route request (RREQ) packet. A node receiving the packet checks whether it has a route to the destination. If a route does not exist, it adds its own address to the route record of the packet and forwards the packet along its ongoing links. The number of routes propagated is limited by sending only new packets with unavailable destinations. If a route to destination is known, the node appends the packet with the entire route to destination and generates a route reply packet (RREP). If it is the destination, it copies the route on its cache. If the destination node has a better route to the source, it may use it, otherwise if symmetric links are supported, it initiates its own route discovery and piggybacks the route reply on the new route.

This protocol uses route error packets and acknowledgements for route maintenance. Error packets are generated at a node when the data link layer encounters a fatal transmission problem. On receipt of error packets, the hop in error is removed from the nodes route cache and all routes containing the hop are truncated at that point.

Acknowledgements are used to verify the correct operation of the route links. This may however lead to information loss since we only know of a link loss when there is transmission failure. Since this protocol uses unicast packets that require acknowledgements, it suffers from heavy overloads during transmission and maintenance.

2.2.2.6 Ad Hoc On-Demand Distance Vector (AODV)

The Ad hoc On-demand Distance Vector (AODV) [27] routing protocol is also an on-demand protocol (reactive) which borrows the basic operation principal from the DSDV. Basically, it minimizes the number of broadcasts by creating routes on demand as opposed to DSDV and DSR that maintain a complete list of periodically updated routes. Another notable difference from DSR is its use of hello messages for maintenance of list of neighbors. It checks its routing table for destinations when a node wants to send a packet and initiates a route discovery process ones the destination in unknown. Route discovery is done by broadcasting route request packets (RREQs) to the neighbors. Intermediate Neighbors forward this request till the destination is found. Intermediate nodes record the source and destination addresses of the packets together with its id and sequence numbers before forwarding it. Nodes use sequence numbers to update routes (deleting old routes) and avoid loops in the routes. Old routes are normally deleted when predefined periods are exceeded. Hello messages are used for maintenance of the networks backbone. This can however be done with lower layer protocols. When a node moves out of its position, an upstream neighbor detects this. A route message of type RREP with infinite metrics is generated by the neighbor to other upstream nodes so as to invalidate the route. If a source moves, it reinitiates a route request.

The algorithm of AODV begins with node broadcasting a route request (RREQ) to its neighbors as in case of DSR. This packet includes in its header a unique sequence number for looping avoidance. Intermediate nodes can reply if they have a route to the destination with equivalent or greater sequence number than that contained in the RREQ. When forwarding the RREQ, nodes record in their route tables the address of the neighbor from which the first copy of the RREQ is received, thereby establishing a reverse path. When RREQ reaches destination or an intermediate node with a fresh enough route, the node responds by uni-casting a route reply (RREP) packet back to the neighbor from which it first received the RREQ. As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their routing tables that point to the node from which the RREP came. In this case, the packet does not contain the entire route to the destination (as in DSR): rather it contains previous and next hops towards the destination. If a source node moves, it initiates a route discovery process. When an intermediate node moves, upstream nodes respond with link failure notification (RREP with infinite metrics). Periodic Hello (local broadcasts) messages are used for maintenance of local connectivity of known nodes and awareness of unknown. Knowledge of neighbors' location may also be achieved through listening to data retransmission. This may however not be accurate. These messages may take up bandwidth when there is much traffic and mobility is high (solution offered by our algorithm). Moreover, this information may not be fresh enough for sending data packets.

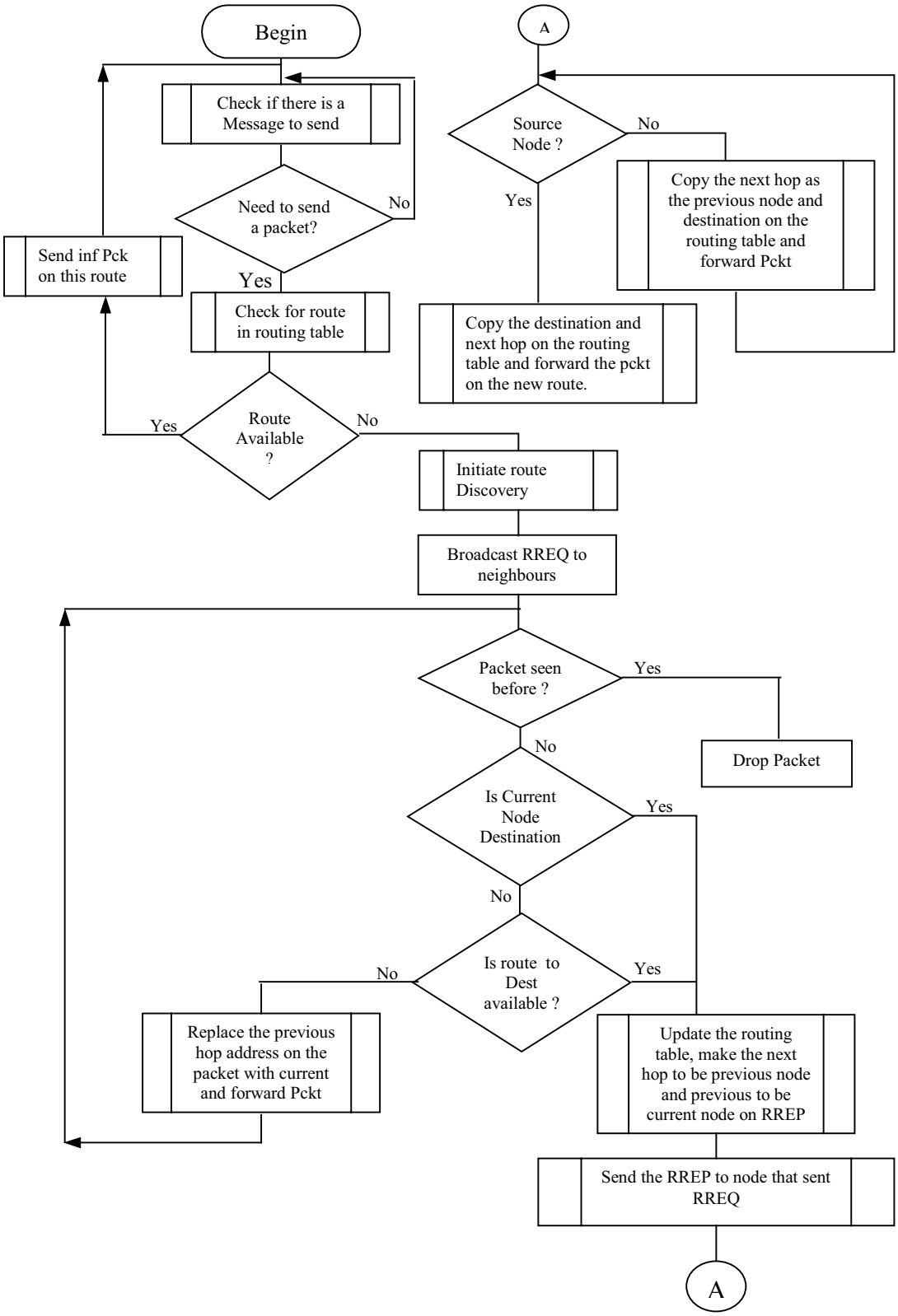


Figure 2.2 Algorithm forAODV

Issues of power saving and QoS are not addressed here. However security has been considered in a modified edition of the aodv protocol "secure on demand ad hoc routing protocol - saodv [28]". Figure 2.2 illustrated a schematic diagram of the general operation of the AODV routing protocol.

2.2.2.7 Adaptive Distance Vector Routing (ADV)

Adaptive Distance Vector (ADV) [13] is a distance vector routing algorithm that exhibits some on demand characteristics by varying the frequency and size of routing updates in response to the network load and mobility conditions. It starts with the common "distance vector algorithm" that uses sequence numbers to avoid loops and uses routing updates to learn and maintain routes. Routes are maintained only to active receivers in order to reduce the number of entries advertised.

This scheme uses an adaptive method of triggering partial and full route updating that obviates the periodic full updates. It broadcasts active receivers in the whole network and all nodes maintain routes to active receivers. This allows nodes to have updated routes to active destinations but may have a problem of overlay and congestion since all other nodes flag off their transmitters so as to wait for this active node transmission. This results in bandwidth waste.

2.2.3 Comparison of Reactive with Proactive Protocols

Most proactive protocols perform well in regard to optimization of the routes. However, this usually requires a lot of information in order to have precise information about the network. The advantage here is that changes in link states are usually reported as they occur. However as the network gets larger and the node mobility increases, the amount of control information could suffocate the network. As mentioned earlier, these control information should be maintained at lowest possible levels. Hierarchical/clustered structure of the network could be deployed to limit the amount of control traffic.
This gives rise to another problem of creating and maintaining these hierarchies if it is to be done in a dynamic manner.

On-demand protocols (reactive) differ in the way that they flood search packets and collect the route information, how they determine the metrics of link, how they find their neighbors, and how the maintenance of the route is done. Table 2.1 shows a comparison of the two classes of protocols. Comparison is done for nine metrics whereby the two classes are compared at both low and high node mobility.

From the analysis of these protocols, we see that proactive (shortest path) protocols provide good performance in terms of route acquisition, adaptation to changes and topology knowledge. However, this comes at the cost of high routing load. The on demand protocols suffer from sub-optional routes as well as dropped packets. However, they are significantly more efficient in terms of routing load. The multi-path, e.g. TORA, does not perform well in spite of maintaining multiple redundant paths. The overhead of finding and maintaining multiple paths and protocols sensitivity to the loss of routing

packets overweighs the benefits of multiple paths. Also the end-to-end delay performance is poor because of the loss of distance information. The routing load differentials between all routing protocols reduce with large number of peer-to-peer conversations in the network.

Metric	Proactive		Reactive	
	Low Mobility	High Mobility	Low Mobility	High Mobility
Route accuracy	High	Low	High	Moderate
Route acquisition speed	Instant	Low	Low	Moderate
Speed of adaptation to changes	High	Low	High	High
Topology knowledge rate	High	Low	Moderate	Moderate
Cache Optimization rate	Low	Low	High	High
Information loss rate	Low	High	Low	Moderate

Table 2.1 Characteristics' comparison of protocol classes at low and high node mobility.

Reactive routing protocols perform better under high mobility than proactive. High mobility results in more frequent link failures and the overhead involved in updating all the nodes with new routing information. This has more effects in proactive protocols than in reactive protocols where the routes are created only when required.

2.2.4 Hybrid Routing Approach

Hybrid routing protocols aggregates a set of nodes into zones in the network topology. Then, the network is partitioned into zones and proactive approach is used within each zone to maintain routing information. To route packets between different zones, the reactive approach is used. Consequently, in hybrid schemes, a route to a destination that is in the same zone is established without delay, while a route discovery and a route maintenance procedure is required for destinations that are in other zones. The zone routing protocol (ZRP), zone-based hierarchical link state (ZHLS) routing protocol, and distributed dynamic routing algorithm (DDR) are three hybrid routing approaches. The IMEP protocol is a modification of TORA that offers characteristics of a hybrid framework. The hybrid protocols can provide a better trade-off between communication overhead and delay, but this trade-off is subjected to the size of a zone and the dynamics of a zone. Furthermore, hybrid approaches provide a compromise on scalability issue in relation to the frequency of end-to-end connection, the total number of nodes, and the frequency of topology change. Thus, the hybrid approach is an appropriate candidate for routing in a large network.

2.2.4.1 Internet Manet Encapsulation Protocol (IMEP)

Internet Manet Encapsulation Protocol (IMEP) [30] is designed as an underlying layer that provides reliable delivery of messages to support upper layer protocols. Any upper layer can register with IMEP and get link status information from it. It maintains a neighbor list by periodically sending out a BEACON message, which is answered by a HELLO message. This protocol is designed to support many routing protocols in ad-hoc networks. Important features found in IMEP, making it suitable as a framework of other protocols includes its Link status sensing (allowing choice of uni- or bi-directional status), Control message aggregation and encapsulation, Broadcast reliability, Network-layer address resolution, and its ability to provide hooks for inter-router security authentication procedures.

IMEP provides router identification, interface identification, and addressing and defines its purpose to provide common interface to routing protocols. However, IMEP generates a lot of overhead, because its neighbor discovery algorithm generates a large number of HELLO messages per second that aims to provide reliable delivery. The protocol would greatly benefit from the link availability forecast algorithm that would help it reduce the overhead and be able to compete with the current protocols. This protocol was initially designed as a framework of protocol design from where TORA (Section 2.2.2.2) was designed. It has deliberately been place in the hybrid class for comparison purposes.

2.2.4.2 Zone based Hierarchical Link State (ZHLS).

Zone based Hierarchical Link State (ZHLS) [31] is a protocol that incorporates location information into a novel “peer to peer” hierarchical routing approach. The network under this protocol is divided into non-overlapping zones. Each zone maintains two types of routing messages: intra-zone messages which provide the node connectivity within a zone and inter-zone messages, which provides information about connectivity between different zones. Intra-zone messages are propagated locally within the zone while inter-zone messages are propagated globally within the whole network. This hierarchical characteristic reduces the amount of communication and storage overhead for routing. This is a very important property for large- scale wireless networks and in which scalability and scarce wireless bandwidth are major concerns. Node mobility is another important consideration. This normally has effect at zone level and at node level if the moving node is a gateway node (a node joining two zones).

This protocol does not have cluster heads and the zone level topological information is distributed to every node. The peer-to-peer characteristic avoids traffic bottlenecks, prevents single point of failure and simplifies node mobility. The zone messages are used for building the zone routing tables. A node wishing to send a packed only need to indicate the zone ID and the node ID of the destination in the packet header.

2.2.4.3 Distributed Dynamic Routing (DDR):

Distributed Dynamic Routing (DDR) [32] is a Global Position-Less (GPL) hybrid routing protocol. In principle, this protocol shares many characteristics as in the ZHLS mentioned above. The main idea here is to construct a forest from a network topology, where each tree of the constructed forest has to be optimal. Each tree then forms a zone. The network is then partitioned into non-overlapping zones as in ZHLS. Each node periodically computes its zone ID independently. Zones are interconnected through nodes not in their trees but in transmission range with nodes in their trees. Therefore the whole network is a set of interconnected zones. The size of the zone decreases or increases depending on some network features like node density, rate of network connection/disconnection, node mobility and transmission power. Mobile nodes can either be in a router mode or in a non-router mode regarding its position in the tree. This allows an efficient energy consumption strategy. Each node is assumed to contain routing information only to those nodes that are within its zone, and information regarding only its neighboring zones. This protocol has three main advantages over other known routing protocols: First, it provides different mechanisms to drastically reduce routing complexity and improve delay performance, second, it has a strong nature of infrastructure-less-ness – it does not even require physical location information and third, zone naming is performed dynamically and broadcast is reduced noticeably.

The algorithm consists of six cyclic time-ordered phases: preferred Neighbor Election, Forest Construction, Intra-Tree Clustering, Inter-Tree Clustering, Zone Naming, and Zone Partitioning all of which are executed based on the information provided by a beacon exchanged between two neighboring nodes. Basically, each node maintains two routing tables. These are the intra-zone routing table and the inter-zone routing table.

2.2.4.4 Zone Routing Protocol (ZRP)

Zone routing protocol (ZRP) [33] is a hybrid protocol for reconfigurable wireless networks (RWN) discussed above. Although it shares the same principal of zone forming with ZHLS, it differs in that it is not hierarchical and its zones are overlapping. ZRP dynamically adjusts itself with network radius. Therefore the zone radius is the most important parameter of this protocol. Normally, a network with high node density will have a larger radius while that with low density will have a lower radius. It is aimed at reducing costs of frequent updates in the constantly changing network topology by limiting the updates to the immediate neighborhood of the change. The routing is flat rather than hierarchical reducing organization overheads allowing optimal route discovery and reduced network congestions. This protocol adapts to mobility patterns of the mobile users. ZRP offers proactive nature through its IntEr-zone Routing protocol (IERP) and reactive nature in its IntrA-zone Routing Protocol (IARP). Lower node mobility favors larger radius (more proactive – IARP) while high node mobility favor low radius (reactive - IERP). ZRP uses a Neighbor Discovery Protocol (NDP) for neighborhood discovery.

2.2.4.4.1 Neighbor Discovery Protocol (NDP)

The NDP helps nodes identify their neighbors through broadcasting hello beacons. Upon receipt of a beacon, a node records the beacons source ID in its neighbors table that it scans at regular intervals to check the status of its neighbors. If no beacon is received from that neighbor in a maximum predefined interval, the neighbor is considered lost. If the beacon is received and the neighbor is previously unrecorded, it is considered as found.

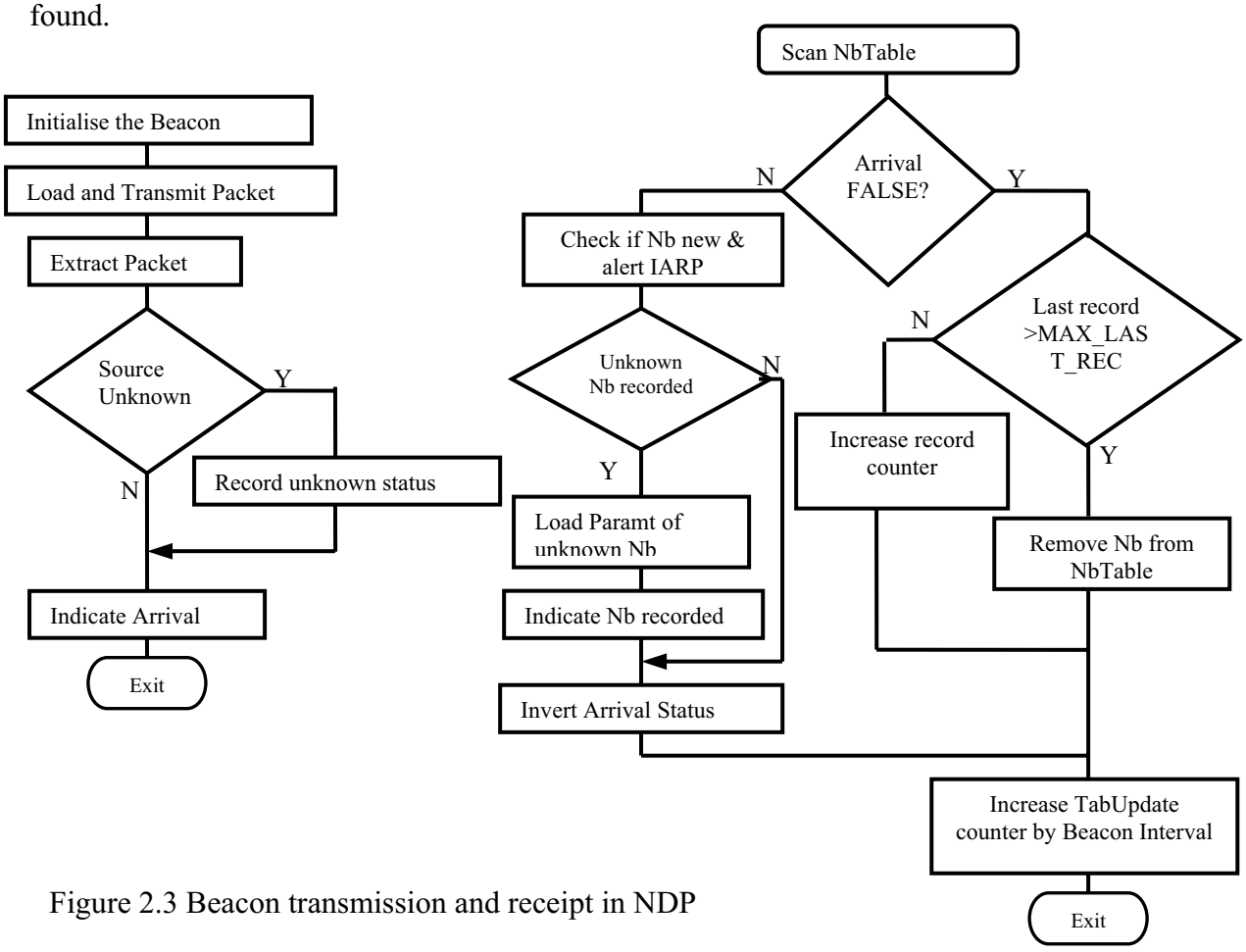


Figure 2.3 Beacon transmission and receipt in NDP

When a neighbor is either lost or found, the IERP is notified of this new link status. This protocol is of particular importance in this study since the research is based on efficiency in topology discovery schemes. This will be more obvious as its features are extracted from the AODV and adjustments made on the NDP in the zone routing protocol in the later chapters. Figure 2.3 illustrates a schematic diagram of the general operation of beacon transmission and receipt in NDP.

2.2.4.4.2 IntrA-zone Routing Protocol (IARP)

The IntrA zone Routing Protocol [34] is used by nodes to compute intra-zone routes based on the link status of each routing zone node. A node may receive link status updates either from IARP link status packet or from an interrupt generated by the NDP.

Link states are maintained in a link state table.

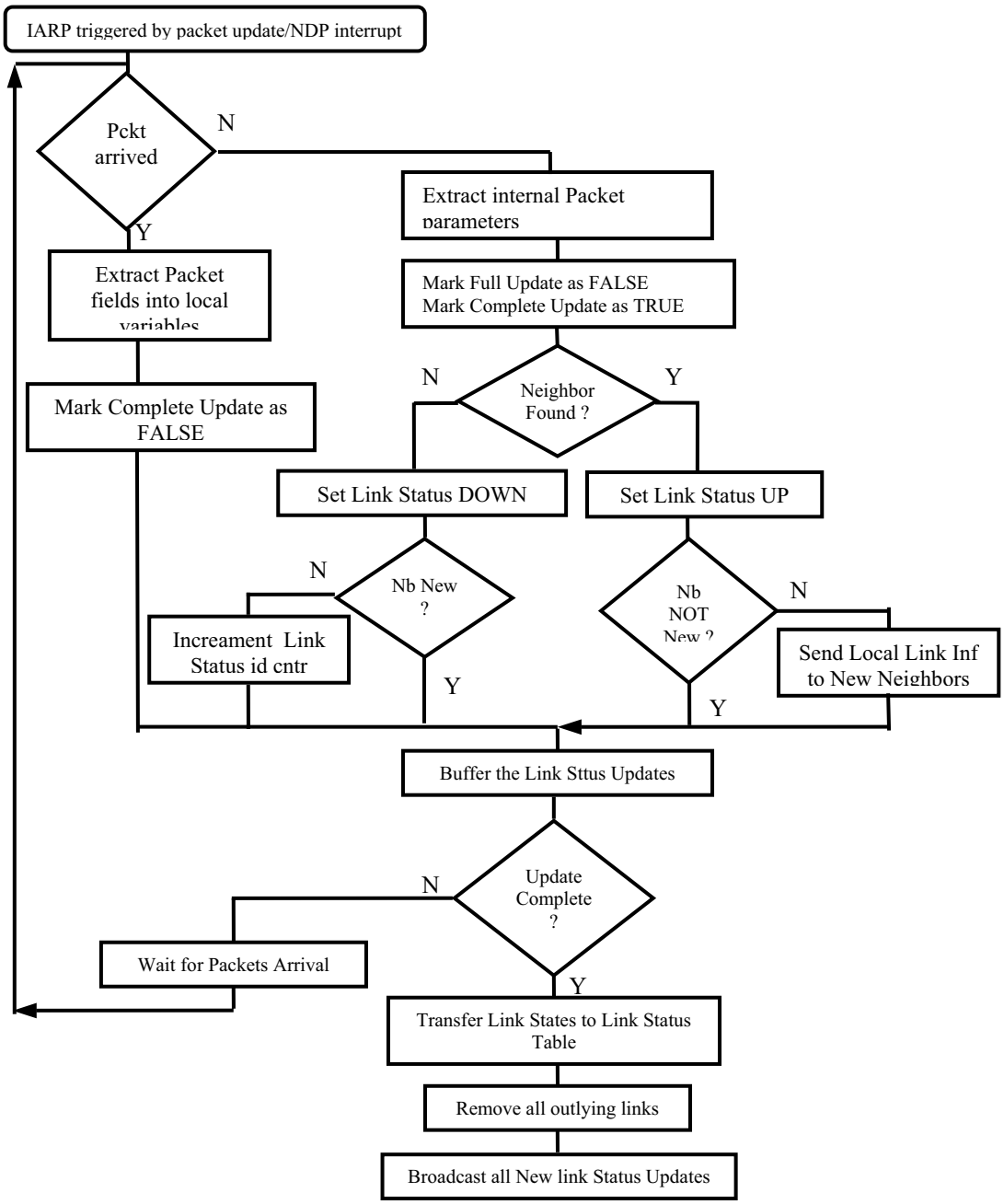


Figure 2.4 IARP - Routing Table Update

When all pending link states have been received, the route table is recomputed using a minimum spanning tree algorithm. The link state table is then updated to remove links that lie outside the routing zone. Newly received link-state updates for link sources within the node's routing zone are forwarded to all the nodes neighbors. In addition, any new neighbor discovered by the node is sent the link states of all nodes that lie inside the

neighbor's routing zone. Figure 2.4 illustrates the schematic flow of operation in the updating of the IAPR routing table.

2.2.4.4.3 IntEr zone Routing Protocol (IERP)

The IntEr zone routing protocol [35] is responsible for discovering routes to the hosts beyond a node's routing zone. A route request is triggered at the network layer when a data packet is to be sent to a destination that is outside the nodes zone. This message is assigned a query ID that is unique to the source node. The combination of the source ID and the query ID uniquely identifies a route query in the network. After recording the source query and ID in the request packet, the packet is broadcasted to all peripheral nodes.

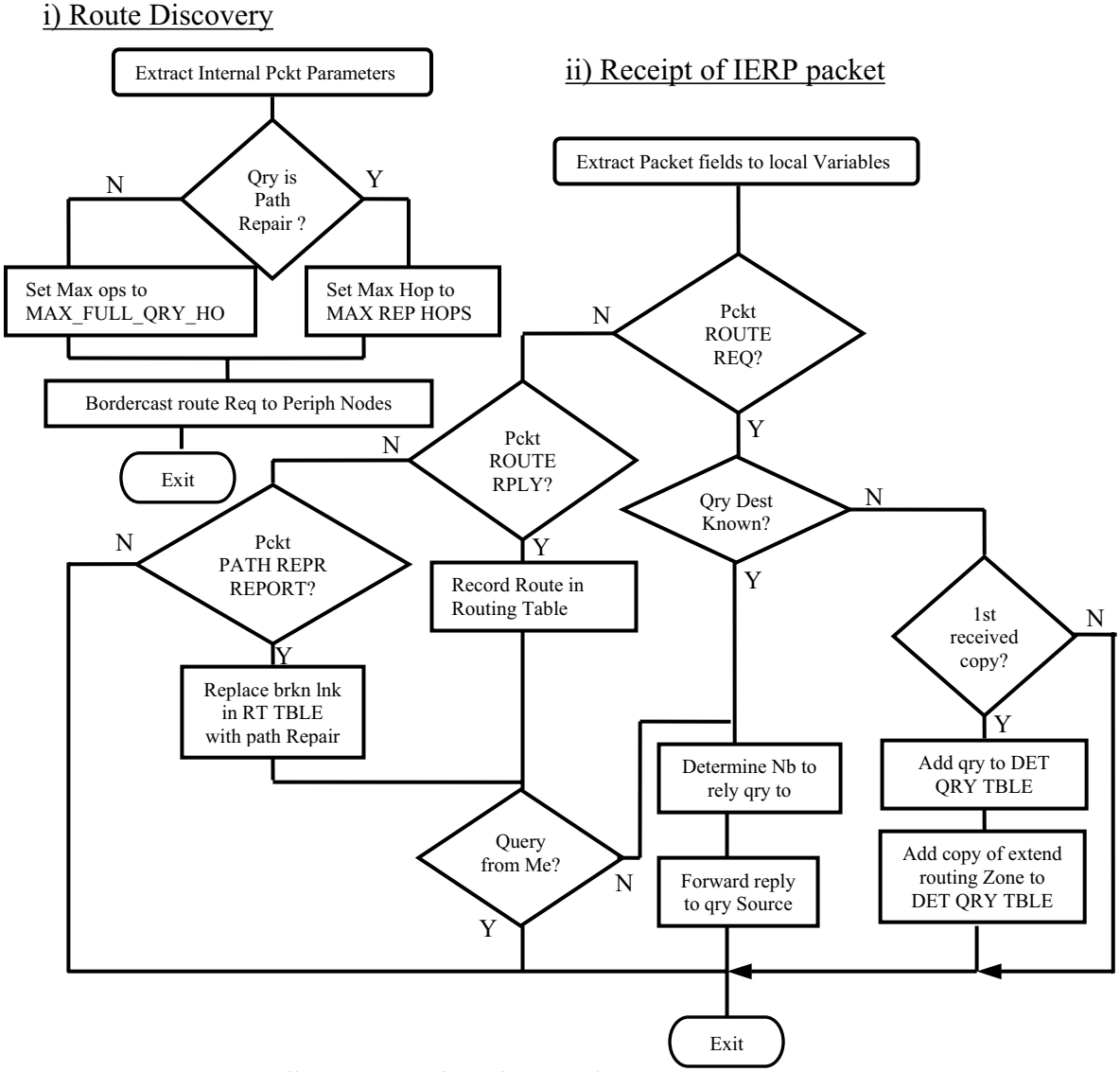


Figure 2.5 IERP - Route discovery and Packet receipt

When a node receives a route request packet, the query ID, query source, broadcasting node, and previous hop are recorded in the detected queries table. The node then searches its route table to see if the requested destination lies within the routing zone. If so, the node responds with a route reply returned to the query source, along a path specified by the previous hop information cached in the detected queries table. If the destination does not belong to the node's routing zone, the node forwards the route request down its bordercast tree to the target peripheral nodes.

The protocol employs a combination of advanced query detection and early query termination techniques. These techniques are employed to avoid possible excess traffic by the bordercasting mechanism. In these techniques, a node restrains from forwarding route request to a bordercast recipient if the recipient lies within a previously queried zone or if the current node has already relayed a query packet to that recipient. IERP is also responsible for the repair of broken routes. On detecting a broken route, IERP is immediately notified and attempt a repair though a restricted route discovery method. Upon receipt of route repair reply, all routes containing the broken link are updated. If no route reply is received within a reasonable duration of time, the routes containing the broken links are removed. The route discovery and packet receipt in the IERP are schematically represented in figure 2.5.

2.2.4.4.4 Bordercast Resolution Protocol (BRP)

The Bordercast Resolution Protocol (BRP) [36] provides the border-casting packet delivery service used to support network-querying applications. It uses a map of an extended routing zone, provided by the local proactive IntrA-zone routing protocol (IARP), to construct the border-cast (multicast) trees, along which query packets are directed. It acts as a guide for the requests of the global reactive IntEr zone Routing Protocol (IERP). This protocol employs special query control mechanisms to steer requests away from areas of the network that have already been covered by the query. The combination of Multicasting and zone based query control makes border-casting an effective and tunable service that is more suitable than flood searching for network probing application like route discovery.

2.2.5 Comparison of the reactive, proactive and hybrid types

Table 2.2 shows a summary of the results from the graphs on comparison of reactive proactive and the hybrid routing protocols. Performance has been shown for low and high packet rates in figure 2.2 and performance for low and high Mobility in figure 2.3.

2.3.0 Routing Overheads in RWAdhoc Nets

Routing Overheads (RO) are packets that are responsible for route's creation and maintenance in a wireless network. In these networks, these packets cannot be done without since nodes in the network can only know existence of their neighbors through exchange of these packets .The overheads are therefore crucial in any routing

environment. The packets exist in various forms and sent using different methods. The ones used in wireless networks can be grouped in two main groups: Single and multi-hop groups.

	Proactive		Hybrid		Reactive	
	Low Rate	High Rate	Low Rate	High Rate	Low Rate	High Rate
Routing Overhead	L	L	L	M	L	H
Throughput	H	L	M	M	M	M
Delay	L	L	M	M	M	H
Packet delivery ratio	H	L	M	M	H	M
Route acquisition	H	M	M	L	M	L
Route reliability	H	M	M	L	M	L

Key: L = Low, M = Moderate, H = High

Table 2.2 Performance of protocol classes with low and high packet rate at constant speeds

	Proactive		Hybrid		Reactive	
	Low Mobility	High Mobility	Low Mobility	High Mobility	Low Mobility	High Mobility
Routing Overhead	L	L	L	H	L	H
Throughput	H	L	H	M	M	M
End-to-End Delay	L	H	L	H	M	M
Packet delivery ratio	H	L	H	M	M	M
Route acquisition	H	L	H	M	M	M
Route reliability	H	L	H	M	M	M

Key: L = Low, M = Moderate, H = High

Table 2.3 Performance of protocol classes at low and high node mobility and constant transmission rates

In the single hop group, packets are sent only to the neighbors. These can be neighborhood discovery packets, hello messages, negotiation packets, e.g. request to send (RTS), clear to send (CTS) and acknowledgement (ACK), among other possible one-hop packets. Neighborhood and hello packets are used for discovering or confirming nodes neighborhood. When a node receives these packets, it gains new knowledge of the sender's existence and can use it to ensure sending information to reachable destinations and minimize loosing packets otherwise sent to unreachable destinations. These packets can be transmitted on demand or periodically depending on the protocol and their use. For example, hello messages are normally transmitted periodically in the current protocols while negotiation packets are transmitted only when required. As illustrated later, periodic transmission may have negative effect on one kind of protocol class while on demand mode may have undesired consequences on the other protocol class. A hybrid of the two would therefore yield better results.

Multiple hop packets include the route discovery packets, route maintenance packets, and error packets, among others. Route discovery packets can be route request packets, route reply packet, gratis packets etc. These follow multiple hops being forwarded by intermediate nodes towards their destinations. They are normally flooded in the whole network since the sender does not initially know where the destination is.
All the above packets are vital in the normal running of a network but their applications vary depending on the protocol they are being used with. For example, Ad hoc On demand Distance Vector (AODV) protocol generates more request packets while Dynamic Sequence Routing (DSR) protocol generates more reply and delivery acknowledgement packets. Their implementation is however basically similar. This allows formulation of a general method of handling them according to hop quantity.

2.3.1 Effects of RO

Due to variation in application of the routing overhead packets in different protocols, these packets have different effects on the protocol in use. As mentioned in the previous section, proactive protocols generate less neighbor management packets but suffer inaccuracy of the neighborhood and consequently of the whole network if the mobility of the nodes increases. Increasing the frequency of the updates would result into heavy traffic around the node, which causes overloading of the node and its inability to handle information traffic appropriately. On the other hand, reactive protocols suffer the reactive updating due to the frequent updates triggered by the frequent network changes. These networks also suffer from inability to discover neighbors that are inactive, as they consider them unreachable and remove them from the list of neighbors. These inactive neighbors would otherwise be useful if they participate in routing procedures.

When routing packets are flooded in a network, they travel through the entire network. The only limiting factor is the set time to live or fragmentation of the network. They therefore travel to sections of the network where they are not needed. This leads to processing of packets that will never be used resulting to improper use of the scarce bandwidth, waste of much needed energy and improper use of network resources. These packets also cause congestion of the network. They cause delays in delivery of information and reduction of delivery ratio as information packets are dropped due to filling up of queues. Algorithms are therefore needed that control and properly manage the way these packets are sent.

2.3.2 Controlling RO

In order to control these routing overhead packets, it is necessary to understand their origin, use and application. It is also important to understand their effects and consequences on various protocols. As mentioned in the section of related areas (1.4), different protocols in the literature partially address this problem with suitability only on the particular protocols in which they are used. It would however be beneficial common characteristics of these packets were identified and a common algorithm that would be used as a base in designing routing protocols derived. The common characteristics

identified in this research are the hops quantity experienced by the packets and the methods of broadcasting the packets.

In order to reduce the single hop type, designing of a hybrid algorithm between periodic updates and on demand has been suggested. An intelligent algorithm that uses available parameters for making decision on when to send the packets is desired. On the multiple hops type an algorithm that reduces the flooding zone has been developed. This relieves non- participating nodes from handling packets unnecessarily allowing more productive use of their resources. The method suggested here is based on location and other mobility parameters.

2.4 Other Issues in Ad Hoc Routing

2.4.1 Security considerations of Ad-hoc Routing Protocols

Although Security in ad hoc routing [37] is, with reference to this dissertation, a separate and possibly a future research topic, it is worth mentioning some important issues that need attention.

Achieving security in such networks is a challenging task mainly due to three reasons:

i) the dynamic topology and membership – a network topology of Reconfigurable Wireless Ad hoc network is very dynamic as mobility of nodes is very random and rapid, emphasizing the need for secure solution to the dynamics,
ii) Vulnerable wireless link – passive/active link attacks like eavesdropping, spoofing, denial of service, masquerading, impersonation are possible and
iii) Roaming in dangerous environments – Any malicious node or misbehaving node can create hostile attack or deprive all other nodes from providing any service.

Nodes within nomadic environment therefore require secure communication link to communicate. They therefore need identification with each other before such a link can be secured. They exchange identifications and credentials, which are authenticated and protected for recognition by the receiving node. In order to guarantee that the delivered identity and credentials are not compromised, it is essential to provide a security architecture to secure Reconfigurable Wireless Ad hoc networking.

In most cases, identification problem leads to privacy problem. This is when mobile nodes are not willing to reveal their identifications to other mobile nodes from privacy point of view. Any compromised identity leads attackers to create privacy threat to user devices. It is unfortunate that current mobile standards do not provide any location privacy and in some cases revealing identity is inevitable to generate communication link. This calls for a seamless privacy protection to harness the usage of Reconfigurable Wireless Ad hoc networks. The main challenges in Reconfigurable Wireless Ad hoc security are:

a) Secure routing:

Since each node in a Reconfigurable Wireless Ad hoc network acts as a router, the routing protocols supported in Reconfigurable Wireless Ad hoc networks are vulnerable to attacks. These attacks can be classified into passive and active attacks. In passive attacks, the attacker does not disrupt the operations of the routing protocol but only tries to discover valuable information by listening to the routing traffic. It is normally impossible to detect such an attacker thus difficult to put a defence against such an attack. In the second class of attack, active attack, the attacker injects arbitrary packets into the network. The aims may be to attract packets destined to other nodes for analysis or disabling the network. It is sometimes possible to detect such an attacker. It however remains a great threat especially when huge amounts of money are involved such as in commercial and military applications.

Proposed routing solutions are capable to operate with dynamic topology but in terms of security measures they provide partial or no solution (see table 2.4.1). Implementation of secure routing protocol is also a challenge.

b) Link level security:

Spoofing is the main type of attack that can affect wireless communication environments. As there is no protection like firewall or access control in Reconfigurable Wireless Ad hoc environments, any node can become vulnerable to attacks coming from any direction or from any node. This leads to susceptibility or other attacks like tampering with nodes credentials, leaking of confidential information or impersonation.

c) Key management:

Cryptography is probably one of the most common techniques used in Reconfigurable Wireless Ad hoc networks. This involves public key encryption of digital signature. These mechanisms are supported through centralised key management where trusted certificate authority (CA) provides public key certificates to mobile nodes so that the nodes can provide mutual trust between themselves. This kind of mechanism breaks the distributed nature desired in Reconfigurable Wireless Ad hoc networks. Further more, any tampering with the CA can easily compromise the security of the entire network. This can be compared to the undesired point of failure experienced by centralised type of networks. The proposed mechanisms for identification therefore only provide a partial solution since they are vulnerable and are not able to scale. There is also difficulty in achieving the goal of proper management and safekeeping of small number of cryptographic keys in Reconfigurable Wireless Ad hoc networks due to random mobility of nodes where continuous connectivity is not maintained.

2.4.2 Quality of Service

Providing different quality of service levels in a constantly changing environment is also another major challenge. An adaptive QoS mechanism needs to be devised over the

existing resource reservation mechanism where applications request QoS by specifying the minimum level of service they are willing to accept and the maximum level of service they are able to use rather than just specifying a fixed value of a particular parameter. Qos can both be applied at application layer and network layer. At application layer, it is necessary to check how well user expectations are qualitatively satisfied, the arrival patterns, sensitivity to delivery delays and application level implementation, i.e. application level specifications and encodings. At network level, parameters that need to be checked are: the bandwidth utility, i.e. the rate at which the applications' traffic must be carried by the network, latency, i.e. the delay that an application can tolerate in delivering a packet data, jitter i.e. the variation of latency and loss - which is the percentage of lost data.

Qos in RWAN differs from that of wired networks in that there is no core and edge distinction since nodes become homogeneous in QoS provision, the medium is shared as opposed to point-point in case of wired network, Low bandwidth capacity which is basically in the range of 2Mbps – 11Mbps as opposed to the gigabit router in wired networks, node mobility resulting in frequent (inevitable) QoS breaks that require recovery among other features. In order to cater for these challenging features, it is desired that routing protocols become QoS aware. One possible way of making routing protocols QoS aware is enhancing QoS aware MAC layer and QoS aware routing in cross-layer integration. This is however a possible future work research topic and not dealt with in this dissertation.

2.4.3 Medium Access Control

The Medium Access Protocol (MAC) is critical to achieving a statistically equitable distribution of available capacity between contending users [38]. This is also important for ensuring that the QoS requirements of different users are satisfied. The lack of a centralized mechanism such as a base station or an access point is an issue to be considered while designing the Medium Access schemes for Reconfigurable Wireless Ad hoc Networks. A scheme like Busy Tone Multiple Access [39] would therefore not be suitable. Suitable schemes should be distributed in nature and be efficient enough to work in the Reconfigurable Wireless Ad hoc environment where each node typically knows very little about other nodes, when compared to nodes in networks where centralized control points exist. While taking care of the hidden and exposed terminal problems the MAC protocols should be efficient and robust so as to exchange data packets between the nodes without collisions. If the networks need to exchange real time traffic, the MAC protocol must provide support to upper layers by keeping the delay between packet transmissions to a minimum bounded value thereby guaranteeing a minimum level of QoS.

2.4.3.1 Medium Access Control protocols

MAC protocols were extensively studied in the early 70's, mainly for single shared channel applications [40]. However, due to the possibly large size of Reconfigurable Wireless Ad hoc Networks, much larger than the transmission range of a single

transmitter, single shared channels would not perform in this environment. A fair MAC sharing scheme [41] is more suitable for ad hoc networks. Hidden channel and exposed terminal problems needs to be addressed by MAC protocols.

The design of a good wireless MAC protocol has to address challenges raised by (i) mobility of the nodes ad (ii) unreliable time varying channels. Mobility affects the MAC protocol because the set of users competing for the capacity of the medium keeps changing. This makes it difficult to allocate bandwidth in equitable version. Time varying effects such as fading and interference also make it difficult to administer medium access control on the channel. Traditional protocols like "ALOHA" fails to deliver due to packets collision and low optimization. This has been improved by use of time slots to achieve only about 36% [40].

In Carrier Sense Multiple Access (CSMA), every user senses the channel before transmission. It however suffers the hidden terminal since not all nodes in the network can her each other. Collision at the receiver is also possible in CSMA since channel is sensed only at the transmitter. This is the exposed terminal problem. An improvement to this protocol has been provided by introduction of the RTS/CTS (Request To Send / Clear To Send) dialogs. Such mechanisms are found in protocols like MACAW (Media Access Protocol for Wireless LANs) [42] and FAMA (Floor Acquisition Multiple Access) [43] that introduces the Collision Detection Multiple Access (CDMA) mechanism. Most of the wireless LAN protocols do not address, or insufficiently address, the central issue of a large-scale mobile network, namely that the connectivity between nodes may be highly unstable where assumption is made that all the nodes will hear the RTS/CTS dialogs. This assumption is invalid where nodes are highly mobile resulting to high loss in packets due to collision.

The requirement of a protocol is that it should on one hand, resolves the collision based on the state of the channel at the receiver while providing a constant indication of the status of the channel. This is so that when a mobile node migrates within the range of transmitting/receiving node, the mobile nodes transmission does not interfere with the transmission in progress. The Dual Busy Tone Multiple Access (DBTMA) [44] protocol that is a hybrid of CSMA and RTS/CTS schemes supports these characteristics. A comparison between DBTMA and RTS/CTS schemes shows that the prior improves the performance of the later by more than doubling the network capacity. DBTMA provides improved channel capacity while avoiding most of data collisions.

2.5 Summary

From the analysis of the protocols discussed, hybrid routing protocols are more promising in achieving the goals of Reconfigurable Wireless Ad hoc networking but require a number of modifications. Among the modifications required are the reduction of traffic overheads mostly caused by control packets and better knowledge of the network topology. Traffic overhead is one of the biggest challenges in the current protocols suitable for high mobility scenarios. More efficient route selection and maintenance algorithms are also needed. These are the main issues addressed in this dissertation. It is

worth mentioning that most of the current routing protocols have not addressed the issues of QoS and Power saving options. Power saving options are mostly found in Sensor networks protocols [40], [41], [42], [43]. The schemes developed in this research lead to savings in nodes power and energy. Only a handful of protocols have addressed the security issue. Table 2.2 gives a comparison of some competing Reconfigurable Wireless Ad hoc routing protocols discussed in this chapter while fig 2.6 shows a summary of protocol classes.

	Proactive			Reactive				Hybrid		
Property	DSDV	WRP	OLSR	TORA	SSR	DSR	AODV	IMEP	DDR	ZRP
Loop freedom	Y	Y	Y	SL	Y	Y	Y	SL	Ps	Y
Uni-directional link	N	N	Ps	N	N	Y	N	Y	Ps	N
Periodic b-cast	Y	Y	Y	N	N	N	Y	Y	Y	Y
Distributed	Y	Y	Y	Y	Y	Y	Y	Y	P	Y
Multiple routes	N		N	Y	N	Y	N	Y	Ps	N
QoS	N	N	N	N	N	N	N	N	Ps	N
Power Saving	N	N	Ps	N	N	N	N	N	Y	N
Security	N	N	N	Y	N	N	Ps	Y	Ps	N

Key: Y = Yes, N = No, SL = Short Lived, P = Partial, Ps = Possible, F = Fast, S = Slow, M = Moderate, H = High, L = Low

Table 2.4 Summary of comparisons of routing protocol classes.

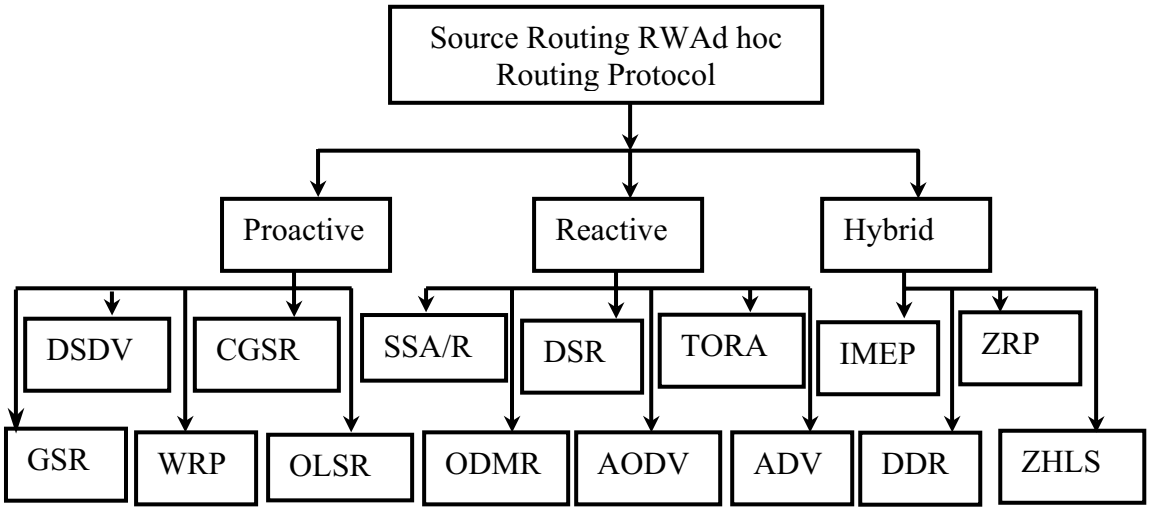


Figure 2.6 Time dependant route establishment classification of Reconfigurable Wireless Ad hoc Routing protocols

2.6 Performance Evaluation of "winning" Routing Protocol

2.6.1 Simulation Models

2.6.1.1 Methodology

2.6.1.1.1 The MAC layer specifications

The ns-2 based simulation model used in this dissertation uses extensions made by the Monarch group to include an implementation of the 802.11 MAC layer protocols. This is a Distributed Coordination Function of IEEE 802.11 for wireless LANs. An unslotted carrier sense multiple access (CSMA) technique with collision avoidance (CSMA/CA) is used to transmit the date packets. It uses a shared media radio with a normal bit-rate of 2.0 Mbps and a nominal range of 250 meters. This is a radio model that uses characteristics similar to a commercial radio interface, Lucent's WaveLAN [49].

The MAC 802.11 extension uses RTS/CTS prior to each DATA transmission for reduction of the well-known hidden terminal problem. The protocol maintains a send buffer of 64 packets. It contains all data packets waiting for route, such as packets for which route discovery has started, but no reply has arrived yet. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 seconds. All packets, both (data and routing) sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has a maximum of 50 packets and it maintains as a priority queue with two priorities each served in FIFO order. Routing packets gets higher priority than data packets.

In this simulation model, the routing protocol has a complete overview of all packets transmitted and forwarded by each node (for simulation purposes).
All the RREQ packets are treated as broadcast packets by the underlying MAC layer. The RREP and RERR are uni-casted between participating nodes while the Hello packets are uni-casted between the participating nodes but listened by other neighbors promiscuously (in a passive manner). Our modified protocol uses this promiscuous listening and the feedback information from the MAC layer to detect link breakage.

2.6.1.1.2 Traffic and Mobility models

Setdest tool is used in this research for generating Network topology and movement models while Cbrgen tool is used for generating traffic models. These tools are part of the CMU Monarch group's extension packages to the ns-2. The CBR (Constant Bit Rate) data stream with packet size of 512 kbytes is used as application layer traffic in our simulations. The communication source destination node pairs (nn/scrs) are randomly distributed over the network by the cbrgen tool. The mobility model provided uses the random waypoint model [50]. Each of the nodes starts its movement towards a randomly selected position (within the grid) at a randomly selected speed (within set limits). Once the destination is reached, the node pauses for a predetermined duration, before it randomly selects a new destination.

2.6.2 Performance Metrics

Four important performance metrics were evaluated. These were the packet delivery fraction, the average throughput, and routing overhead and average end-to-end packet delay. These metrics are somehow related to each other. The longer the paths, the higher the probability of packets drops. Therefore with a lower delivery fraction, samples are usually biased in favor of smaller path lengths and thus less delay.

2.6.2.1 Packet delivery fraction

Packet delivery fraction is the ratio of data packets delivered to the destinations to those generated by the CBR sources. This is an important metric for best effort traffic.

2.6.2.2 Throughput

This is the amount of information delivered (error free) to destinations per unit time interval. This may be computed in packets per unit time or in bits per unit time. This metric is affected by factors like routing overhead and delay, which are explained in the next sub-section. The general formula for calculating throughput is:
Throughput = (0.75 *W *MSS)/RTT where, W = window length,
MSS (Maximum segment size) = packet size without header and
RTT = round-trip-time.

2.6.2.3 Routing overhead

This is the number of routing packets transmitted within a predetermine interval. It is also important to determine the normalized routing load, which is the number of routing packets transmitted per data packet delivered to the destination. Each hop-wise transmission of the routing packet is counted as one transmission. In order to justify our simulation, it is also necessary to determine the percentage of the overall transmitted packets that control packets take. This will help in determining weather the control packets are taking up bandwidth that could otherwise be used by the information packets.

2.6.2.4 Average end-to-end Packet Delay

This gives the average time that it takes to transfer packets from source to destination. This may be based on routing packets (protocol performance) or information packet (efficiency of the routing protocol).

2.6.2.5 Assumptions made in the implementation

Due to the current nature of the implementation tool in use, a number of assumptions have been made. This is due to omission of some of some real life conditions in the tool. An assumptions hereby made is that the speeds of nodes are assumed to be uniform throughout the simulation. This may not be practical in real life situation as nodes may

change speed unpredictably from high to low and vise versa. The model however does not deviate too much from expectation since the selection of destinations before change of direction is random. Moreover, we normally have destinations selected by nodes prior to departure unless something unique happens on the way. The other assumption made is that nodes always receive messages successfully (weak signals are hereby not taken in to consideration). In real life situation, energies may be lost along the way due to weaker node signals.

2.6.3 Implementation proceedings

2.6.3.1 CMU's wireless extensions to ns-2

CMU's wireless extension to ns-2 [54] (incorporated in the ns-2.1b9a) provides the implementation of some MANETs routing protocols. For simulation purposes, necessary modifications on the OLSR, DSR and AODV codes were done on their existing versions on the ns-2.1b9a version. For visualization of the node movements under various protocols, choice of visualization tool was necessary. "Nam", which is the basic visualization tool used for ns-2 but it does not support Reconfigurable Wireless Ad hoc simulations. Therefore, Ad-hockey, which is a Perl/Tk program that supports the visualization of ad-hoc simulations was adopted. An older version of this tool however requires compatibility with the version of Perl/Tk used. Perl/Tk modules developed after version 5.3 do not work with this version of ad-hockey tool. The newer version of the tool was used with the Perl/Tk version 5.3 for this compatibility. This also required Tk800.015.

2.6.3.2 Generating traffic and mobility models

Random traffic connections of TCP and CBR can be set up between mobile nodes using the cbrgen.tcl traffic-scenario generator script. This generator script is available under ~ns/indep-utils/cmu-scen-gen. It can be used to create TCP and CBR traffic connections between wireless mobile nodes. The command used for this purpose was of the form:

ns cbrgen.tcl [-type cbr |tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]

For the simulations carried out, traffic models were generated with from 10 to 100 nodes using cbr traffic sources. The sources were varied in the range between 8 and 80 while the rates were selected between 4 and 88 kbps.

The node-movement generator is available under ~ns/indep-utils/cmu-scen-gen/setdest directory and contains the setdest{.cc, .h} and the Makefile. The necessary command for the mobility scene generation is:

./setdest [-n num_of_nodes] [-p pausetime] [-s maxspeed] [-t simtime] \ [-x maxx] [-y maxy] > [outdir/movement-file]

Mobility models were created for the simulations using up to 100 nodes with low pause times (0s and 1s were used). Maximum speeds were varied between 10 m/s to 40 m/s, topology boundary of 500 x 500 and simulation time of 100 seconds were used.
Nb: Simulations could have been done with more complex scenarios using longer simulation times (greater than 100 sec). These simulations take much longer to complete since the trace files generated were as large as 50mb. Such files would be very difficult to analyze especially for delay calculations.

2.6.3.3 Simulations' configuration code

This involves writing the TCL code to set up the wireless simulation components. There are: network components types, parameters like the type of antenna, the radio-propagation model, the type of ad-hoc routing protocol, traffic models and node movement models used by mobile nodes etc. The documented code is available at the appendix.

2.6.3.4 Parsing the simulation trace files

After each simulation, trace files recording the traffic and node movements are generated. These files needed to be parsed in order to extract the information needed for performance metrics measurements. The new trace formats look like:

s –t 2.556838879 –Hs 1 –Hd -2 –Ni 1 –Nx 338.88 –Ny 124.44 –Nz 0.00 –Ne -1.000000 –Nl AGT –Nw --- -Ma 0 –Md 0 –Ms 0 –Mt 0 –Is 1.0 –Id 2.0 –It cbr –Il 512 –If 0 –Ii 0 –Iv 32 –P cbr –Pi 0 –pf 0 –Po 2

Here we see that the packet was sent (s) at (t) 2.556838879 sec, from the source node (Hs) 1 to the destination node (Hd) 2. The sourse node id (Ni) is 1, its x-cordinate (Nx) is 338.88, its y-coordinate (Ny) is 124.44, its z-coordinate (Nz) is 0.00, energy level (Ne) is1.000000, the trace level (Nl) is AGT (Agent) and the node event (Nw) is blank. The MAC level information is given by the duration (Ma) 0, destination Ethernet address (Md) 0, the source Ethernet address (Ms) 0 and the Ethernet type (Mt) is 0. The IP packet level information like source address.source port number is given by (Is) 1.0 while the destination address.destination port number used was (Id) 2.0. The Packet type (-It) was cbr, packet size (-Il) was 512 bits, the flow id (-If) was 0, the packet id (Ii) was 0, the packets time to live (ttl) value (-Iv) was 32, the Information about the CBR application presented by the tags (-Pi) for sequence number was 0, -pf for the number of times the packet has been forwarded was 0 and -Po for the optimal number of forwards was 2.

2.6.4 Analysis of Results

Understanding of results analysing tools is necessary. This analysis also requires basic knowledge of a script language like Perl (which was used for analysis) and the ad hockey tool (which was used for visualization of the mobility).
In order to justify the experiments, it is necessary to have comparison of with existing results. The results were compared with those obtained in the performance comparison for current routing protocols' papers [51].

2.6.4.1 Evaluation of packet delivery fraction

The number of sent packets that have the trace form:
/^s*-Nl AGT.*-Is (\d{1,3})\.\d{1,3} –Id (\d{1,3})\.\d{1,3}.*-It cbr.*-Ii(\d{1,6})/
AGT =>Agent Level Trace was first calculated.

Then the number of received packets of the form:
/^r –t (\d{1,3}\.\d{9}).*-Nl AGT.*-Is (\d{1,3}\.\d{1,3} –Id (\d{1,3})\.\d[{1,3}.*-It cbr.*-Ii (\d{1,6})/ was obtained.

Packet delivery ratio was then calculated from the formula:
Packet delivery fraction (pdf %) = (received packets/sent packets)*100 %

2.6.4.2 Evaluating throughput

The number of packets having the trace form:
m/^r.*-Nl AGT.*-It cbr –Il (\d{1,4}/) was first calculated and a count of the number of bytes received kept . The bytes count was necessary for throughput in bytes per second. The total number of bytes received could also be obtained by multiplying the number of packets recorded by 532. Throughput was then evaluated as (total-number-of-bytes-received *8)/(900*1024) Kb/sec.

2.6.4.3 Evaluating Routing Overhead

For calculation of routing overhead, packets of the following form were extracted from the trace file:
/^[s|f].*-Nl RTR.*-It (?:AODV|DSR|OLSR) –Il
(\d{1,4})/)

This gave the total number of packets involved in routing process. If this was compared with the values of total counted sent packets, a fraction of routing packets to the overall packets transmitted would have been obtained.

2.6.4.4 Evaluating Average End-to-End packet delay

A program was written in script language to search for packets of the form

/^s –t (\d{1, 3}\.\d[{9}).*-Nl AGT.*-Is (\d{1, 3})\.\d{1,3}-Id (\d{1, 3})\.\d{1, 3}.*-Ii (\d{1, 6})/), which were the packet sent at the AGT level. The value of the time the packet was released, the source ($src) and destination ($dst) addresses and the packet id (pkt:id) were noted.

The program then searched for the packet of the form:
/^r -t (\d{1, 3}\.\d{9}).*Nl AGT.*-Is $scr\.\d{1, 3} –Id $dst\.\d{1, 3}.*-It cbr.*Ii $pkt_id/) and recorded the delay as the difference between the time of sending and time of receipt. The counter for the number of packets processed was then incremented.
These two searches were repeated until to the end of the file was reached, each time accumulating the total delay. This gave the total delay and total number of packets. Dividing the two gave the average delay.

2.6.5 Performance of different Protocols

Performance of each of the routing protocols was recorded and compared with the other protocols. First, rate of packets delivery (load) was varied from 4 to 88 packets per second and the performance parameters checked. The speed was fixed at 20 m/sec while the number of communicating pairs fixed at 100/ 80.

The mobility of the nodes was then varied and packet delivery (network load) fixed at a workable value (8 pckts/sec) while other metrics remained constant. Finally the network load was fixed at 8 packets/ sec, the mobility fixed at 20 m/s and the number of communicating node pairs varied from 10/8 to 100/80 with intervals of 10.

2.6.5.1 Varying the Mobility (Speed) of nodes

While fixing the communicating pairs at 50 nodes each having 40 connections the mobility (speeds) of the nodes were varied. Performances of the various routing protocols were then compared.

2.6.5.1.1 Packet delivery comparison

Here the number of packets delivered at the destination and those generated at the source were counted. The delivery fractions for standard protocols were computed as the ratio of the packets delivered to the packets generated. The graph of figure 2.7 shows the variation of the packets received with the mobility for AODV and DSR. This graph indicated that AODV performs better than the DSR counterpart at high mobility (beyond 10 m/s).

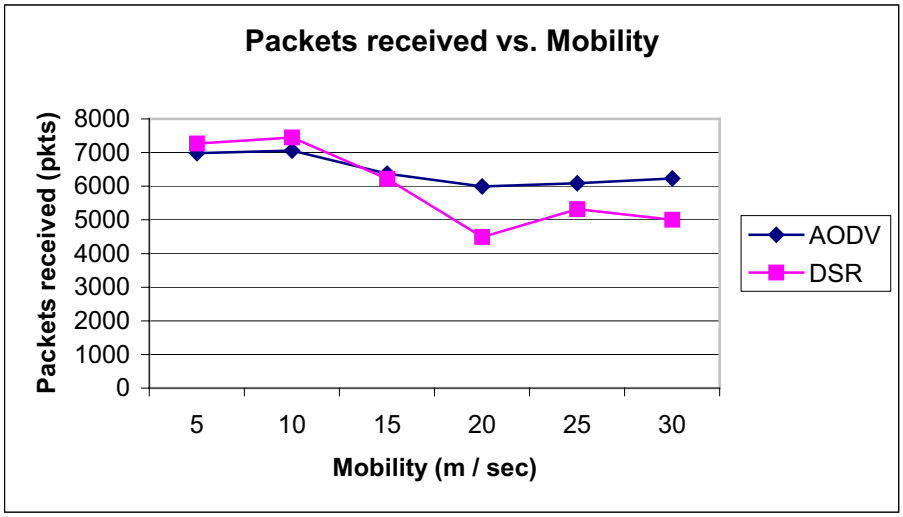


Figure 2.7 Variation of packets received with mobility for fixed communication pairs (50/40)

2.6.5.1.2 Throughput comparison

For evaluation of the throughput, the method in 2.6.4.2 was adopted. A graph of the comparisons between AODV and DSR was obtained. As shown in figure 2.8, the AODV again performed better (higher throughputs) than DSR at higher speeds.

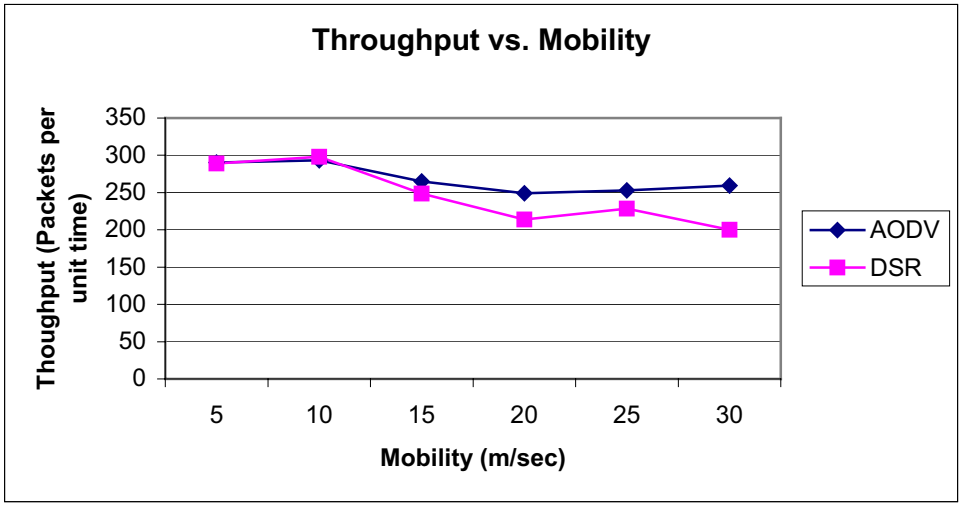


Figure 2.8 Variation of throughput with Mobility at fixed communication pairs (50/40)

2.6.5.1.3 Routing overhead comparison

In this case, the routing overheads of standard routing protocols were compared. From the graph of figure 2.9 it is evident that the AODV generated much more routing overheads than the DSR. This was as a result of the many route requests and hello messages generated by the protocol.

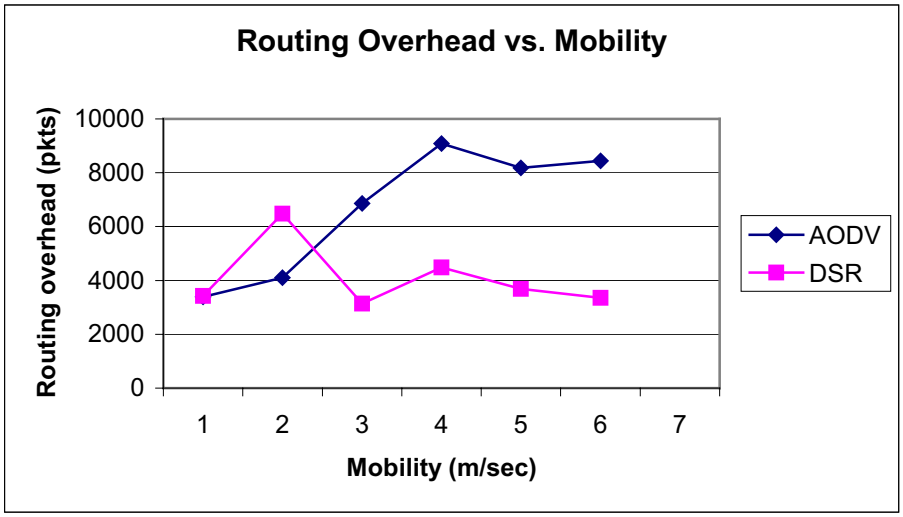


Figure 2.9 Variation of routing overheads with low mobility for fixed communicating pairs (50/40)

2.6.5.2.1 Packet delivery comparison

Similar trend to that of changing mobility was observed with increase in rate of packet transmission (network load). Again as shown in figure 2.10, DSR outperformed AODV.

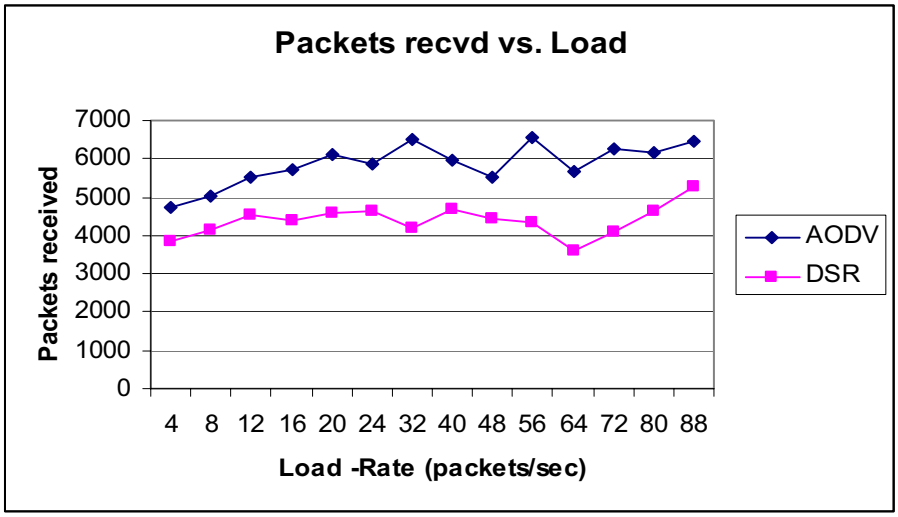


Figure 2.10 Variation of packets received with increased network load at average speed of 20 m/s.

2.6.5.2.2 Throughput against load

Throughput for AODV was still better than that of DSR despite the higher initial traffic transmissions. The change in throughput did not increase much due to the fact that throughput is not directly determined by the rate of generation of packets rather the topology and ability to find routes. However for very large traffics, congestion may affect the throughput.

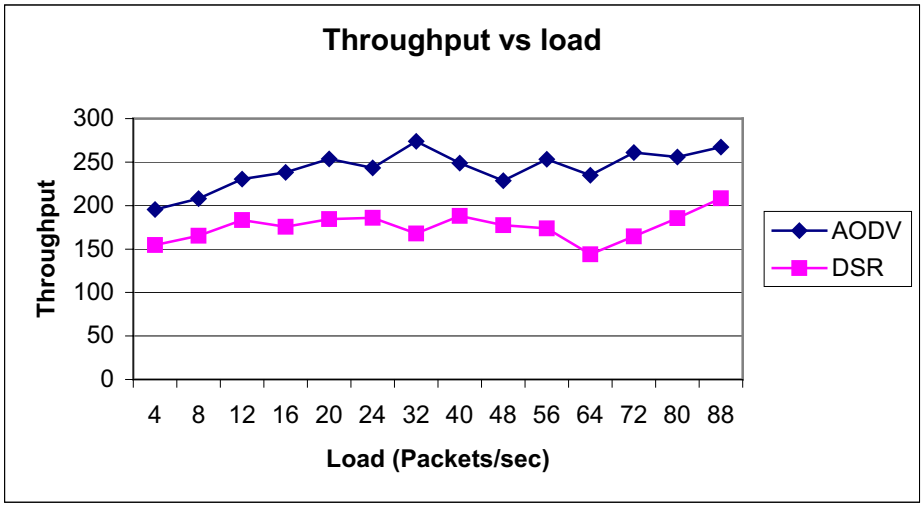


Figure 2.11 Variation of throughput with network load at an average speed of 20m/s

2.6.5.2.3 Routing Overhead vs Transmission rate

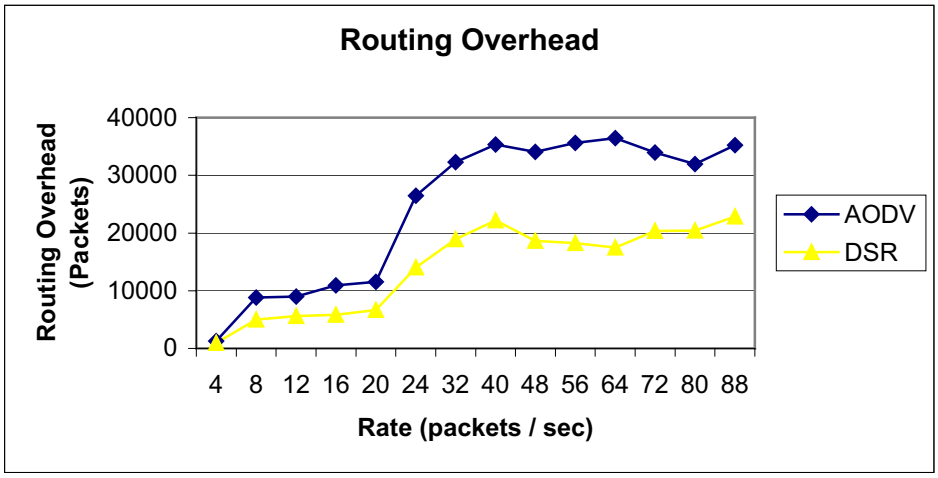


Figure 2.12 Variation of routing overhead with increased network load at an average speed of 20 m/s

Figure 2.12 sows that the routing overheads in AODV are higher than those in DSR and difference gap increases with increase in rate of transmission. The main contributors to the overheads in this case are the hello messages and the route request for route discovery and updates.

2.7 Chapter summary and Conclusions

From the analysis of the existing routing protocols and protocol schemes, it is evident that we are still far from achieving the intended goals of Reconfigurable Wireless Ad hoc networks. Excess routing overheads have been identified as an obstacle to achieving some of these goals. However, the results and progress made in the existing research is very promising. A desirable Reconfigurable Wireless Ad Hoc routing protocol is one that

combines various properties suggested by various research groups found in some of the recent routing protocols. Such a protocol should be proactive in the sense that it should maintain the network connectivity and also be able to react to dynamic network changes. However, the current hybrid routing protocols fall short of this expectation. Some of the desired characteristics may be gotten from various protocols that are designed to offer specific functions for the desired applications. These protocols include the DSDV, OLSR, TORA, IMEP, DSR and AODV as pro- and reactive protocols, ADV, DDR and ZRP as protocol frameworks, among other protocols. Notable modifications could however be:

i) On faster and more efficient network discovery mechanisms,
ii) Minimization of control packets especially when the network becomes more dynamic,
iii) GPS independence whereby the protocol would use methods such as triangulation based techniques with modification of pulse synchronizations and
iv) Increase scalability of highly dynamic (and large-scale) networks with high node mobility.

Other notable characteristics are that it should support efficient routing and robustness against link failures and intrinsically support services and topology discovery, the latter of which can be supplied separately in other protocols. The schemes suggested in this dissertation specifically address the routing overhead management while making the protocol use intelligence in making broadcast decisions. The ideas suggested can be implemented on a wide variety of existing protocols. While this is so, we strongly support the use of hybrid frameworks similar to the DDR that would make such an implementation easier.

Comparison between the proactive and the reactive protocols showed that both protocol groups have both advantages and disadvantages over each other under different conditions. There is therefore no outright winner between the two groups. In order to leap the benefits offered by each of the protocols, it would be advisable to support a hybrid protocol that would take advantage of each of the individual protocol and offer modifications that will fit on both cases in one hybrid protocol group.

In order to check the effects of routing overheads on routing protocols, a comparison was done on leading protocols. Simulations were done on the different protocol types and results compared. Excess routing overheads were found to have negative effect on the efficiency of the tested protocols. This was also reflected in the performance of the protocols. The reason for doing these simulations was to highlight a typical problem that can be solved by the developed schemes. In this case the AODV outperforms the DSR in many areas but faces a problem of the overheads due to inability to manage them. If the overheads can be managed, its advantages over other protocols would allow it become more universal and allow more development of the scheme.

From the above analysis, it would be worth investing in designing schemes that would work in both reactive and proactive protocol group and improve the general efficiency and performance of the protocols.

CHAPTER THREE

3.0 MOBILITY BASED ROUTING OVERHEAD MANAGEMENT SCHEMES

3.1 Introduction

The routing overheads generated by the RWAdhocNet routing protocols have to be managed in order to allow the network meet its objectives. If unchecked, the packets can cause degradation of the network when they are generated frequently. In this thesis, reactive and proactive routing protocols have been selected for demonstration of their effects. The ideas for the prevailing solution are however based on mobility and location parameters thus the topic "mobility based routing overhead management". While these packets may have negative effect in proactive protocols as discussed in chapter two, the situation is more apparent in reactive network protocols where the rate of generation of the packets is dependent on, among other factors, the rate of topology change and speeds of movement of participating nodes. Schemes that will manage generation of these packets are therefore needed. It is desirable that this is tackled from two fronts: first, at node level with its neighbors and second, at network level. At node level, we need to generate these packets only when it is very crucial so that the node will have more bandwidth available for information transmission and reduce interference. This will thus relieve the neighborhood of unnecessary overhead during topology maintenance and guarantees connectivity of the network whenever possible. We suggest a scheme that we call "*link-availability forecast*" that is based on history and prediction of node movement and motion parameters. The packets targeted at this level are the neighborhood maintenance packets (notably the hello messages). We shall see in the section 3.1.1 that, these messages have different impacts on proactive and reactive protocols. At network level, we need to reduce the route search fields so that the competition for the limited bandwidth among routing packets originating from different nodes is reduced. For this level we suggest schemes of reducing search and forwarding zones giving possible modifications. These schemes are also based on nodes' location and history of movement.

3.1.1 Effect of routing overheads in pro- and reactive routing protocols

3.1.1.1 Proactive scenario

In proactive scenarios, overheads may result from hello messages or route discovery/maintenance packets. The routing packets and hello messages alike are normally sent periodically. In case of node movement, hello messages may lead to inaccurate prediction of neighbours' status. One way of solving this would be reducing the intervals of sending the messages. This may have a side effect of increase in overheads in the nodes' neighbourhood. If these messages were sent intelligently, i.e. only when there is a risk of losing a link, the image of the network would be more accurate and the congestion due to their overheads can be avoided. This is the basic goal of the link availability forecast scheme described in section 3.2. We still need the periodic nature of sending these packets so that the nodes, which may have died due to some reason, are identified and inactive nodes noted. The final status would be incorporation of reactive and proactive way of sending the hello messages. We would send a hello message when

we sense the link is about to break and reset the counter for periodic mode. If a node is not moving or is not experiencing link breakage, it continues the periodic mode of sending the hello messages. The overheads resulting from the route discovery packets are due to the magnitude of the messages sent out since each node requires knowledge of the entire network. They will therefore be competing for limited bandwidth.

3.1.1.2 Reactive scenario

Periodic hello messages have proven not very suitable for these scenarios. This is due to the fact that they are needed very frequently and normally lead to over-clouding of the network portion. This is because reactive protocols need to send other control packets like the route request and route reply packets very frequently. Recent protocols have opted to use link layer notification for knowledge of link status. Use of hello messages would help in prevention of losing information before the link layer detects a lost link. This is because the broken link is detected before any information is released. Since hello messages are not flooded like the other control packets, they do not have much negative effect on the entire network. They also have a positive effect since the participating nodes set their own modes of sending the packets independent of each other. The fact that one node may break links faster than the other participating nodes calls for independent way of detecting its own link breakages. It would be desirable in reactive scenarios to set a minimum time for sending hello messages, which would depend on the network pattern and a maximum time to take care of possibly dying nodes. We would then select when to send the hello message if the predicted link breaking time lies between the minimum and maximum times. Reduction of routing overheads can be done more effectively by reduction of flooding areas of routing packets (route requests, replies, acknowledgements etc.).

The above two scenarios suggest to us that a hybrid scenario would be more appropriate for effective reduction of routing overheads in ad hoc routing protocols. This is because even an ad hoc network with high mobility characteristics still has nodes that are not highly mobile since a node may move fast for a short while and change to low speed at some point of time. Auto-configurable routing protocols that can change between pro- and reactive modes would be the better solution to overhead controls.

3.2 Link Availability Forecast Schemes

For nodes in a wireless network to remain in communication, it is necessary for them to be constantly aware of each other's existence. To have such knowledge, they need to be constantly exchanging information about their existence and capabilities. The commonly used method of doing this is broadcasting of hello messages. This is when a node periodically broadcasts packets containing their "*ids*" to its neighbors. This method works well in proactive protocols where the network topology does not change too frequently. However in reactive protocols, the information delivered by these packets is normally obsolete and may mislead the nodes receiving them. This may in turn lead to loss of information packets being sent to nodes that might no longer be reachable. These nodes may have moved away from previously known locations. To avoid such losses, notification of the link status by the underlying link layer may be called for. While this

creates awareness of invalid routes, it does not solve the issue of finding correct routes. Another suggestion to the solution would be increasing the frequency of updates by the hello packets. This, while trying to match the frequent network topology changes, would result to excess packets in the neighborhood thus increasing competition for the wireless' limited bandwidth. As a better solution to this, we suggest a method of forecasting availability of a link between communicating nodes and only send hello messages when it is absolutely necessary to do so. We call this "link availability forecast". This will reduce unnecessary broadcast of the packets when these packets are not very necessary resulting to savings in bandwidth and reduction of network congestion. Link availability is however determined by not only the distance of separation, but also other factors like transmission energy, obstruction etc. We discuss suggestions based on motion parameters (physical) and energy in the node and deduce a general way of computing more than one parameter for the decision of link availability.

3.2.1 Motion parameters based Link availability forecast

Link availability condition will allow completion of transmission of information between two communicating nodes. This will be possible if the two nodes are at what we refer to as "safe distance". Two nodes are at a "safe distance" from each other if their separation does not exceed a distance that can guarantee any communication task. We define safe distance r_s with the following equation:

$r_s \mid R 4 (v_1 4 v_2) 1 (t 2 t')$ ……(1)

Where R is the shorter transmission range of the two nodes,

v_i (i=1,2) is the maximum velocity of the node i,

t is the maximum network latency (delay), and

t` is the time needed for a split or merge to be completed.

It is desirable that a routing algorithm has the capability of choosing a route that will guarantee safe and complete delivery of a packet. This can be improved if we can be able to forecast availability of a link between participating nodes. Safe distance concept is hereby viewed as a tool that can increase the strength of such forecast. With such a forecast, the frequency of sending control packets is reduced while chances of dropping information packets once a link is detected as broken by the link layer application are minimized. Nodes' motion parameters can be obtained using various methods. Two cases are considered in this research. These are fully GPS dependent and partial GPS dependent cases. GPS systems are considered in this research due to their high accuracy and precision. Since GPS works in out-door applications, some algorithms have been introduced that can work in situations where GPS is temporarily not available (indoor applications). In the GPS dependent case, the nodes will be fitted with GPS receivers so as to obtain accurate position on nodes. In the second case, at least one node is chosen as the GPS locator while the others approximate their locations relative to the GPS locator using calculations and correcting their locations when in transmission range with a GPS locator. The nodes may also have their own GPS receivers and calculate their current locations using last known GPS location. Other GPS free schemes suggested in the literature [55] can also be used.

Node movement can be considered to consist of a sequence of random lengths called *epochs.* Within these epochs, we assume each node to be moving at a certain average speed *v* in the direction ď (0 Ω ď < 2´). Consequently, during an epoch *i*, of a duration T_i, node *n* moves for a distance v_nT_i in a straight line at an angle $ď_n$. The number of epochs during an interval of time *t* is the discrete random process *N(t).* Figure 3.1 illustrates the movement of a node over several epochs from position n_1 to n_2 for a displacement R_i and angle of displacement $ď_i$.
For a single node *n* during a time *t* we can define the mobility vector $R_n(t)$ as the sum of epoch mobility vectors:

$$\overset{C}{R}_n(t) \mid \overset{C}{\text{——}}R_n \quad \text{……..(2)}$$

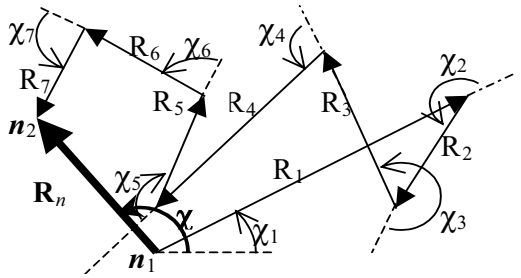
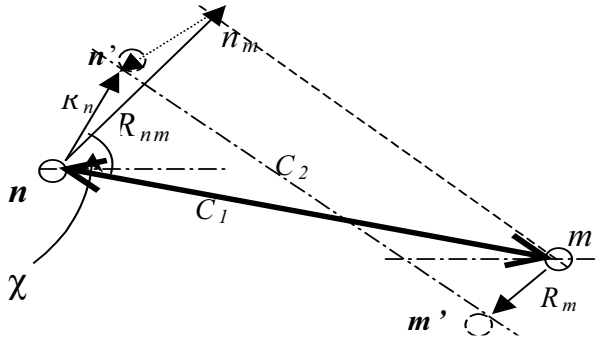


Fig 3.1 A mobile node moving randomly in epocs

For two nodes *n* and *m* moving independent of one another, we can use relative motion concept and calculate displacements of the nodes relative to each other. We will then be able to approximate their separation after a certain period of time. This will help us approximate the separation of two mobile nodes of an ad hoc network scenario.
We choose one of the nodes to be the reference point and move the other node relative to this point. For each movement of the reference node, the other node's movement is translated to an equal movement in the opposite direction. The result is the equivalent random vector $\overset{C}{R}_{nm}$, characterized by the following definition:

$\overset{C}{R}_{nm}(t)$ *is the equivalent random mobility vector of node n with respect to node m defined by fixing n's frame of reference to m's position and moving n with respect to that point.*

Figure 3.2 shows the two nodes *n* and *m* moving over an interval *t*.



Initially the two nodes are at a separation C_1 from each other. Node *n* moves with a random mobility vector $\overset{C}{R}_{nm}(t)$ from position *n* to n_m (the new position of *n* with respect

to *m*). It can be shown that $\overset{C}{R}_{nm}(t)$ is approximately Raleigh distributed over (0, 2´). The relative motion can also be obtained from the relation of individual mobility vectors as:

$$\overset{C}{R}_{nm}(t) \mid \overset{C}{R}_{m}(t) 4 \overset{C}{R}_{n}(t) \quad \text{.....(3)}$$

The value of the separation C_2 can now be obtained (through simple triangulation), since the value of C_1 is already known.

$$C_2^2 \mid \left|\overset{C}{R}_{nm}\right|^2 2\, C_1^2\, 2\, 2\left|\overset{C}{R}_{nm}\right| 1\, C_1\, 1 \cos\chi \quad \text{.....(4)}$$

For a link between the nodes *n* and *m* to be available, the value of C_2 should be less than or equal to the *safe distance* r_s previously defined, i.e. the inequality $C_2 <= r_s$ should hold.

In order for a node to keep information about the location of its neighbors, a node will transmit when it discovers that its movement may lead to a link failure. It checks its speed and its change in direction of motion (i.e. when its direction of motion changes by ϕ/2 or more away from its neighbor). Fig 3.3 shows possible locations at which a node transmits its parameters and adjusts its knowledge about its neighbors.

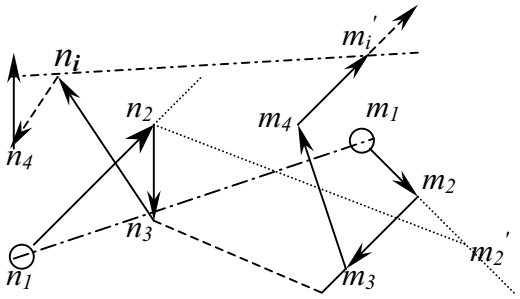


Figure 3.3 Nodes transmit when their motion may lead to link failure. Indicated by dotted line

At n_2 node *n* will turn at an angle less than ϕ/2 away from the known direction of *m*. It therefore does not transmit to *m,* but computes *m's* position as m_2'. This location is safe. At n_3, the turned direction is more than ϕ/2 and *n* transmits its parameters. It then adjusts *m's* accurate position, velocity and direction after getting a reply from it. Please note that at a point like m_i', *m* may transmit, not having changed in velocity or direction, but as a result of expiration of the 'hello message' time. The algorithm of how the messages of the link status are advertised is illustrated in form of flow chart in the figure 3.4.

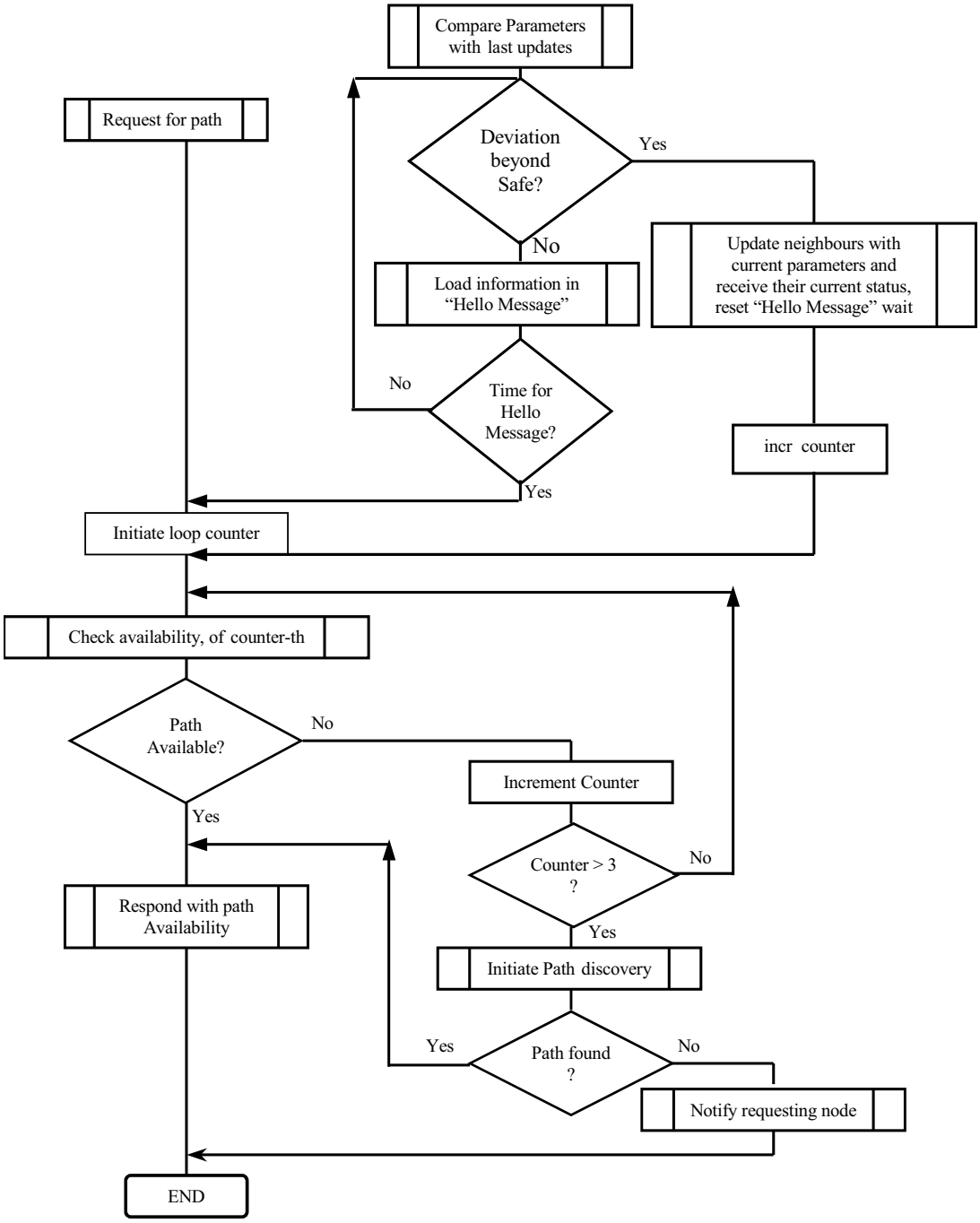


Fig 3.4 Algorithm for Link- Availability

3.2.1.1 Fully GPS location dependent link forecast

In this case, nodes that move in an area where GPS can fully be applied (e.g. battle fields in section 5.2.4) are considered. Such nodes can be fitted with GPS antennas and obtain their location information directly from the GPS satellites. This information is then loaded on the control packets sent to the neighbors. Fully dependant forecast gives more accurate

location results but may fail in situations like disaster areas and other indoor application areas like conferences. The scheme hereby proposed uses the partial GPS location based link availability forecast.

Method 1 (Using coordinates from GPS)

With this method, a node relies entirely on GPS readings to know its position and uses intelligence from *link availability forecast* to obtain the approximate location of the neighbor. This information is obtained from the last hello message sent by that neighbor. Figure 3.5 illustrates such a situation.

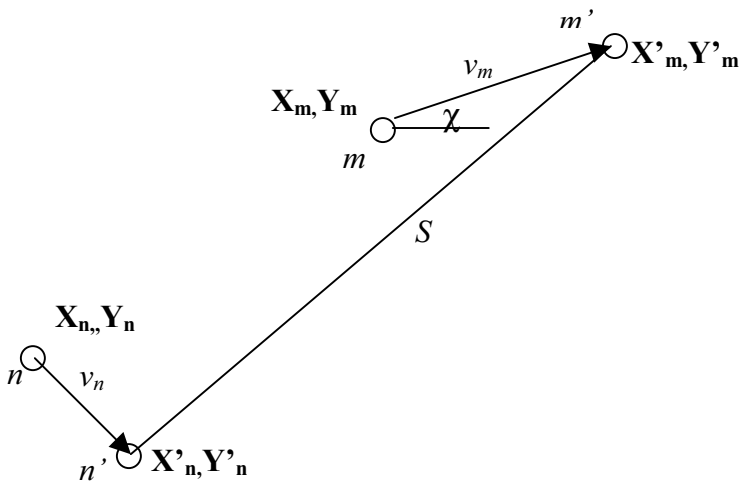


Figure 3.5 Neighborhood discovery

In this method, the node does not need to store its current location since it will not need it in the future. It however records it in the hello packet since the neighbors will need it for the prediction of its future location. When however a node receives a hello message, it records or adjusts the neighbors' information on its neighbors' list.
Since the node knows the neighbors' speed *v* and direction of movement χ (about x-axis), the predicted location of the neighbor after a duration *t* is computed using simple trigonometric formula:

$X'_m = X_m + vt \sin\chi$
$Y'_m = Y_m + vt \cos\chi$ ……(5)

The distance of separation *S* can now be obtained from the equation

$S = \sqrt{(Y'_m - Y'_n)^2 + (X'_m - X'_n)^2}$ ……..(6)

The mobile node will be checking the separation from the neighbor after every *t* seconds and if the neighbors' separation is found to be less than a predetermined safe distance, a hello message is uni-casted to the respective neighbor. Other neighbors whose separation from the transmitting node is safe will listen to this message promiscuously

(process it without response). If however the other neighbors discover that they are at a risk of loosing link with either the sender or the receiver they may send a hello message to that neighbor.

This distance can also be obtained from relative motion where we make one node stationary and the other moves relative to this point.
If we fix the location of the neighbor and move the current node relative to the last known location of the neighbor, the resultant motion vector is the vector sum of the two relative vectors. This is illustrated in Figure 3.6.

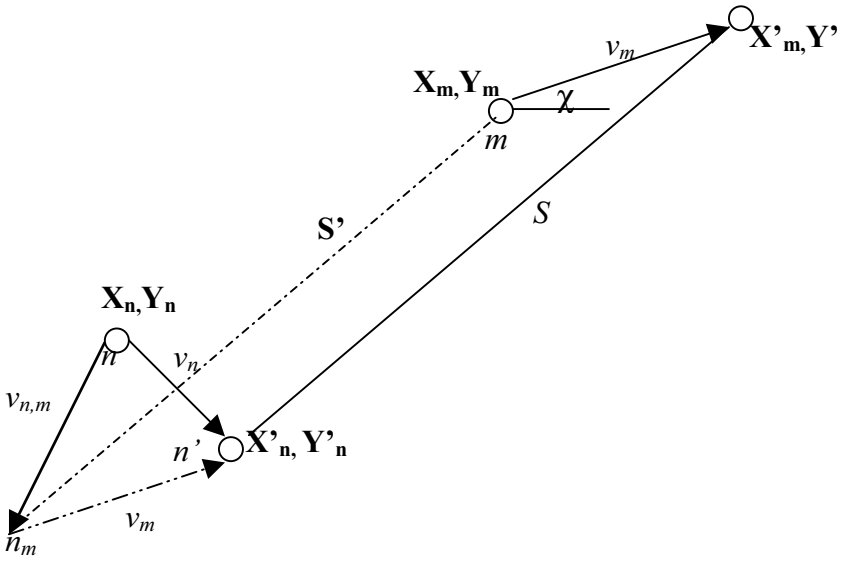


Fig 3.6 Using relative motion for neighborhood discovery

The resultant relative velocity of n with respect to m, which is denoted as $v_{n.m}$ can be obtained from relative motion vector equation,

$v_{nm} = v_n + v_m$.....(7)

Since we are interested in the displacement vector of n_m and we already know the displacement vector of n' we simply add the location coordinates of n' to the calculated displacement vector of m.
The x component of v_{nm} is $X'_n + vt\cos\chi$
The y component is $Y'_n + vt\sin\chi$
The separation S is now obtained relative to the last known position of the neighbor (m) using the formula (6):

$$S' = \sqrt{(Y'_m - Y'_n - vt\sin\chi)^2 + (X'_m - X'_n - vt\cos\chi)^2} \quad \text{.....(8)}$$

3.2.1.2 Partial GPS location dependant link forecast (Using coordinates calculations)

In this method, we assume that the node does not have full access to the GPS coordinates and calculates its coordinates from the last known coordinates from GPS. Such a method

would be suitable in disaster stricken areas or rescue operation scenarios where rescue team members continuously enter and leave collapsed buildings (section 5.2.1). When inside a building, the nodes use last known coordinates and estimate their locations through computation and readjust accurate location when the node leaves the building.
The method will therefore use the formulas in the first method with replacement of the current nodes' coordinates with the calculated ones when GPS information is not available. In this case, it would be necessary for the node to keep record of both its neighbors' last known location and direction of movement. Its location can be obtained through successive accumulation of epoch movements explained earlier in formula (2) i.e. $\overset{C}{R}_n(t) \mid \overset{k}{\underset{1}{\longrightarrow}} \overset{C}{R}_n$, where k is the number of epochs from the last known GPS location.

It is worth noticing that the accuracy of the location degrades with increase in k. This is also expected with longer durations of inaccessibility to GPS location.
Other GPS-free positioning algorithms [55] can also be applied in this scheme. A location dependent positioning algorithm is more preferred in this case.

3.2.2 Energy based Link Availability forecast

In energy-based forecast, the node will check the energy of the previous hop that will be obtained from the packet and the radio propagation module. When a node sends a hello packet, it includes the remaining node energy in its broadcast. We also propose that a field be included in the hello message for the rate of consumption of energy calculated from nodes history of energy consumption. This would be necessary because nodes at different regions may be using up energy faster than in other regions and a wrong indication of remaining energy may be experienced by a neighbor. MAC layer would only give us energy of the neighbor at that particular time but this may not be accurate after a short while. Therefore the energy value is computed with considerations of the expected energy after a predetermined duration of time. When the hello message is received, this value is recorded together with other neighbor parameters. This value is to be used in considering the availability of a link within the time that the link is to be considered available. If this value reaches a minimum before the time-out for the neighbor to become unavailable, a hello message is uni-casted for confirmation of the energy values of the corresponding neighbor. The other neighbors listen to this uni-casted message promiscuously. Adjustments may be made by these neighbors if found necessary.

3.2.3 Hybrid parameter-based approach

In this case, we consider the two values simultaneously using a suitable formula. The safe distance and the energy values are weighted and if one of them reaches its critical value, hello message is automatically triggered. We use the following scenario to develop a suitable formula for the two parameters:

Let t_d be the time consumed since the last communication about the distance of the neighbor, t_e the time consumed since the last communication on energy by the neighbor, t_{othr} time consumed since last communication for other parameter. Let also t_{cd} be the maximum time a link is expected to rust, t_{ce} the maximum time the nodes are expected to

have enough energy to successfully deliver a packet, and t_{cothr} the maximum time for another parameter to be reached. We then define a *safe link* as a coefficient S_{lnk}, which defines the fraction of the summation of the time factor of distance component divided by energy time component to the product of the same distance and time component (see formula (9)).

$$S_{\ln k} = \frac{\sum_{1}^{k} \frac{t_{c_{xi}}}{t_{xi}}}{\prod_{1}^{k} \frac{t_{c_{xi}}}{t_{xi}}} \geq k \quad \text{(9)}$$

Where $t_{c_{xi}} = \{t_{cd}, t_{ce}, t_{\text{coth}}\}, t_{xi} = \{t_d, t_e, t_{othr}\}$ and k is the number of parameters being considered. In our case, we consider two parameters i.e. distance and energy, therefore our k is 2.

3.3 Integration of Link availability into Ad Hoc Routing Protocols

3.3.1 AODV with Link Availability forecast (AODV_LA)

AODV_LA is a modification of the AODV (section 2.2.2.6) that handles the communication packets in a different manner to include intelligence of the *link availability forecast* in the protocol. This modification is on the sending of the hello messages and the way a mobile node identifies its neighbors' properties.

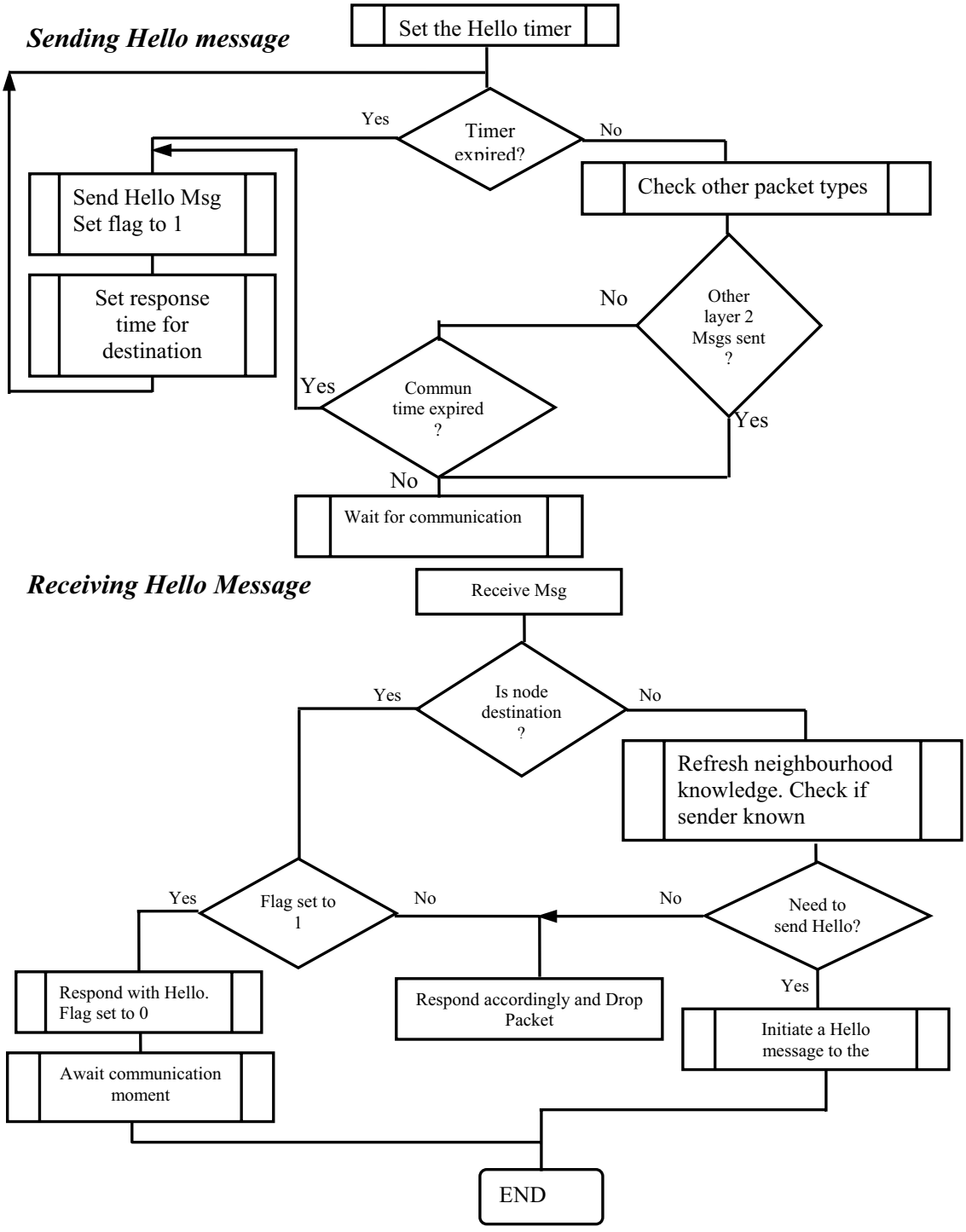
In AODV, a mobile node sends a hello message periodically, which involves setting of timers to be triggered when the time set for the next hello message expires. These messages may take up useful bandwidth when mobility becomes high. Moreover, the accuracy of fresh routes may be compromised. In our scheme, hello messages are sent after intelligently establishing the need to do so. The scheme performs simple calculations in order to establish when it is necessary to send the message packets. We use the idea of location forecast explained in section 3.2 to estimate the parameters needed for our estimation. The protocol periodically checks the neighbor list and does necessary computation for its updates. Updates on this table are done in two ways: first, when a hello message is received from a neighbor and second, when a computation is done after expiry of a set update period. Such updates may or may not trigger hello messages depending on the outcome of the computations, i.e. if the computation establishes that the link is unsafe, it sends a hello messages and refrain from doing so otherwise.

3.3.1.1 Hello messages in AODV_LA

In order to pre-determine when to send a hello message, a node sets a timer to predicted time it would take for a neighbor to be out of reach. This is obtained from the calculations in the above method (formula (9)). The node will then send a hello message to confirm the neighbor's unavailability. This message will have a destination address of the node in question but maintains a hop count of zero, time to live (TTL) = 1 and a flag set to 1 (a request hello message). When the node in question receives this message, it responds with a similar hello message to the sender but resets the flag to 0 (reply to a hello message).

Other neighboring nodes listen to these messages in a promiscuous manner but will adjust their knowledge of these nodes. If there is no response from a requested node, a sender generates a RERR message and deletes the node from neighbors list. If it receives a response, it adjusts the neighbor's location and other motion parameters. This scheme avoids sending hello messages when they are not needed thus saving bandwidth by the drastic reduction of control messages. Figure 3.7 illustrates this algorithm.

Fig 3.7 Algorithm for AODV_LA Hello



3.3.2 ZRP with link Availability (ZRP_LA)

Similar modifications to those on AODV have been done to the ZRP (section 2.2.4.4). Modifications are easier here because we already have the NDP, which is adjusted to implement our scheme. Having done modifications on the AODV, it would be easier to do similar modifications since the IERP can be replaced by AODV. In fact the one can directly use AODV as the inter zone routing protocol of the ZRP. It follows that it would be convenient to use the AODV for IERP and modify the NDP to accommodate the link availability scheme. A protocol like OLSR (section 2.2.1.5) can be used for the IARP.

3.3.2.1 Node Discovery Protocol with Link Availability forecast (NDP_LA)

The Cornell Wireless Networks Laboratory extracted NDP from the ns-2 implementation of ZRP. Essentially in the ZRP, for the discovery of neighbors, nodes advertise their presence to their neighbors by periodically transmitting a HELLO beacon. This will be done on demand in our scheme. Upon receipt of the beacon, a node records the beacon source in its *neighbor table*. Each node scans its *neighbor table* at regular sampling intervals to check the status of each of its neighbors. If no beacon was received from a neighbor during the previous MAX_LAST_RECORDED sampling intervals, the neighbor is considered lost.
In this work, the conclusion is not drawn immediately but confirmation is first made if essential. If however a beacon is received from a previously unknown neighbor, the neighbor is considered found. When a neighbor is either lost or found, IARP is notified of the new link status.

In the extraction of the NDP, the classes isolated for the functions of the topology discovery protocol are basically the hello (beacon) related timers and the neighbor related timers. We then add the functionalities of the link forecast on the *beacon* (Hello) packets, rescheduling them as required and adding the parameters needed for the implementation of our algorithm. In Section 3.3.3.2 (hello messages in NDP_LA) we explain further how this can be done.

3.3.2.2 Hello messages in NDP_LA

Since our main aim is to make better use of the Hello messages and offer savings in terms of overhead and bandwidth, we change the mode of sending the beacon from periodic to reactive (i.e. in response to critical demand).

While the main aims of any protocol are efficiency, quick and guaranteed delivery of information packets, it would be helpful if the protocol adopted an algorithm for uniform load distribution. Within such an algorithm, we would consider the nodes that are active in the transmission of information packets and give them priorities in sending the beacons. We may however consider sending of a beacon (hello message) as not very necessary if a node is not active in information transmission. Here a sleep status would be essential. This would give further savings in bandwidth and battery power. In such a case a node will check a few parameters before sending the message. These include whether it

has sent/forwarded any information packet in the last MAX_LAST_RECORDED. If not, it advises the neighbors on its unwillingness to do any topology updates and goes to a partially active state "*sleep*". When a node goes to partially active state, it will still transmit its presence in the neighborhood but this will be done proactively and listened to promiscuously by the neighbors. It can however be woken from "sleep" by a neighbor that cannot find a route to a destination and has this node in its inactive neighbors list. Figure 3.8 describes this algorithm (nb: the "*sleep*" condition is yet to be implemented in the simulator).

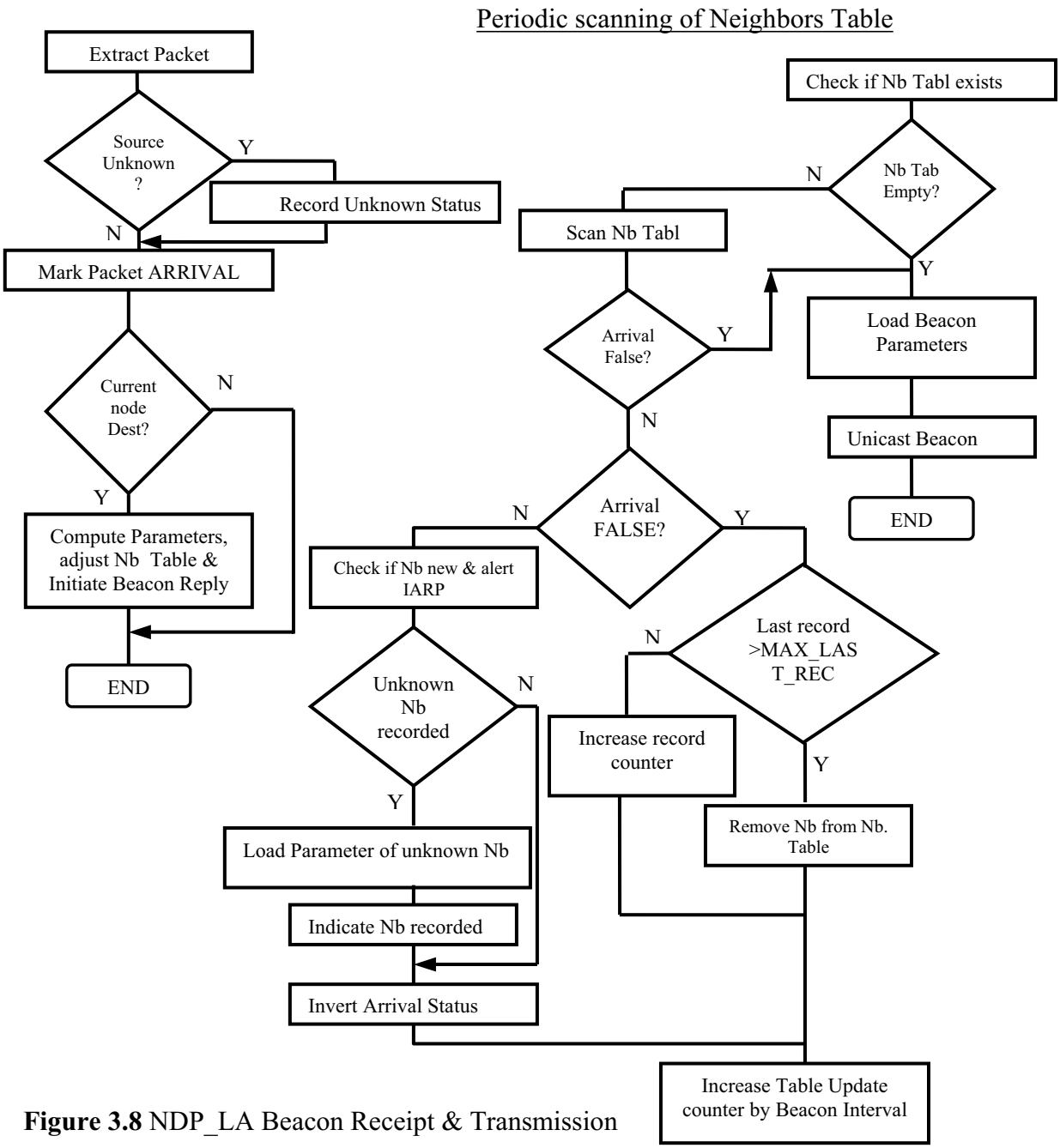


Figure 3.8 NDP_LA Beacon Receipt & Transmission

3.4 Location guided (network level) Routing Overhead management schemes

Although still not optimal, flooding is an indispensable message dissemination technique for network-wide broadcast within *mobile Reconfigurable Wireless ad hoc networks* (mobile RWAdhoc NETs). As such, the plain flooding algorithm provokes a high number of unnecessary packet rebroadcasts, causing contention, packet collisions and ultimately wasting precious limited bandwidth. Studies have been undertaken to optimize flooding using a deterministic approach. Examples are the LAR [6], Greed [7] and other "geocasting" schemes. Because of the highly dynamic and mobile characteristics of mobile RWAdhocs, more efficient algorithms need to be developed.

Due to the ever-changing topology of mobile RWAdhoc nets, broadcasting is a fundamental communication primitive, essential to ad hoc routing algorithms for route discovery. The usual approach for broadcasting is however through flooding. Flooding is well suited for mobile RWAdhoc nets as it requires no topological knowledge. It constitutes in each node re-broadcasting a message to its neighbors upon receiving it for the first time, as we shall see later in section 3.4.1. Although straightforward, flooding is far from optimal and generates a high number of redundant messages. Besides research mentioned above, more effort has been devoted to defining MAC and routing algorithms adapted to mobile RWAdhoc nets, than to flooding. Since flooding is a low-level primitive, optimising it will drastically improve the overall performance of mobile RWAdhoc networks.

3.4.1 Flooding

In *flooding,* a sender broadcasts data packets to all its neighbors. Then, each node receiving the data packets forwards them to its neighbors. Thus, flooding provides potentially lower reliability of data delivery because it uses broadcasting which creates significantly high overhead causing network congestion. One of the advantages of flooding is to deliver packets to the destination on multiple paths. From this point of view flooding is reliable. Flooding may be more efficient than other protocols when rate of information transmission is low enough that the overhead of explicit path generation/selection incurred by other protocols is relatively high (e.g. when nodes transmit small data packets relatively infrequently). Particular routes may be selected by the other protocols more frequently and weakening of that particular section of the network – broadcasting may offer more and probably better alternatives. This approach uses sequence numbers to avoid the possibility of forwarding the same packets more than once. Data packets will reach destination provided that the destination is reachable from the sender, and the destination node does not forward the packets. Some of the drawbacks encountered by flooding include high routing overhead and lower reliability of data delivery due to the broadcast behaviour of flooding. However there is the advantage of simplicity and availability of multiple paths to destination from which a node can choose. The schemes suggested in this research are not replacements of flooding, rather an improvement of flooding as we actually do flooding but to limited areas.

3.4.2 Cone-shaped route search field definition scheme

We suggest a scheme of handling the traffic overhead that has a similar approach to LAR [6] but reduces the field by defining the direction field as a cone section between the source and destination. Our scheme also defines the expected destinations location differently.

Before the source node floods a route request to a previously known destination, it calculates the destination's possible area of location as last known location plus a location error. Location error will be obtained from last known average speed and the time elapsed since last update i.e. $L_{appx} = L_{\text{last known}} \partial R$. Now $R = v_{last\ known} \Delta t_{ellapsed}$.
When a forwarding node receives this route request packet, it first compares its location with the approximate destinations location to know whether it is in the likely direction of the destination. If it lies within the tolerance, it checks whether it has more recent information of the destination (from its time elapsed since last update from that destination). If it has more recent information, it computes the possible location of the destination and replaces the parameters placed by the source. It then re-broadcasts the packet towards the destination. If it has no current information about the destination but lies in the direction of the destination, it forwards the packet with parameters from the source. If it does not lie in the destination's direction and has not heard from the destination, it simply drops the packet. Figure 3.9 shows how the source broadcasts the request packets that pass through the intermediate nodes towards the destination.

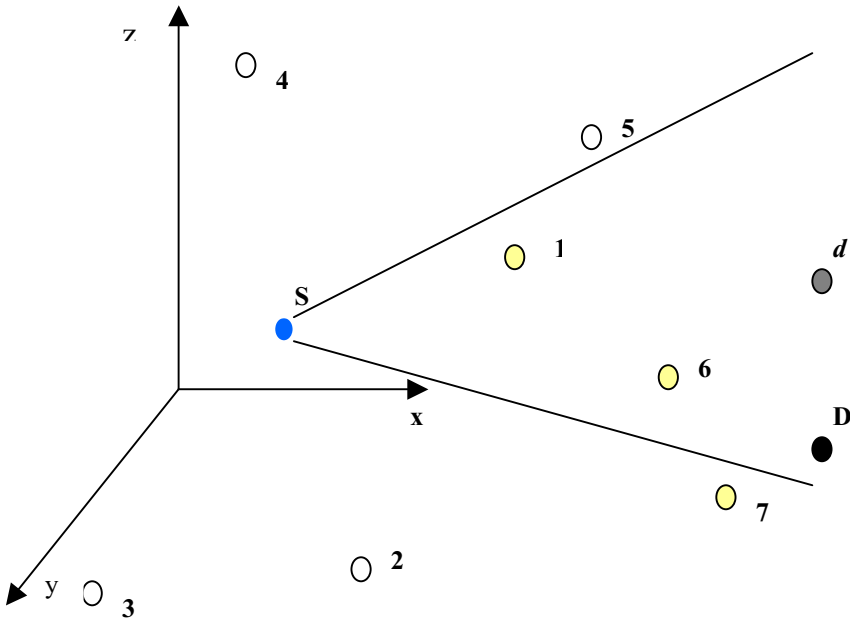


Figure 3.9 Source estimates destinations location

In figure 3.9, the source node S broadcasts a request packet towards '*d*'(last known location of the destination node currently at D). Nodes 1 and 6 are within the estimated field of direction of *d* and therefore forward the request packet towards *d*. Note that node 7 also forwards the packet although it is not within the direction field of *d*. This is because it has more recent information about *d* than the preceding intermediate nodes. Therefore it is an intermediate node towards *d*. Nodes 2, 3, and 4 may hear the broadcast from S if they are within its transmission radius but drop the packet since they are neither in the direction field of s nor do they have more recent information of *d*.

As seen in the figure 3.9, the number of broadcasted packets in the scheme is greatly reduced allowing the nodes not in the direction of the destination (2,3 and 4) not to participate in the route discovery and possibly handle other tasks.

So as to have the conical direction field, we use the fact that the further the destination from the source, the longer it takes to convey information to the source and the higher the possibility of it having deviated further from the last known location. Therefore an intermediate node will determine the direction field based on its current distance from line joining the source and destination (see figure 3.10)

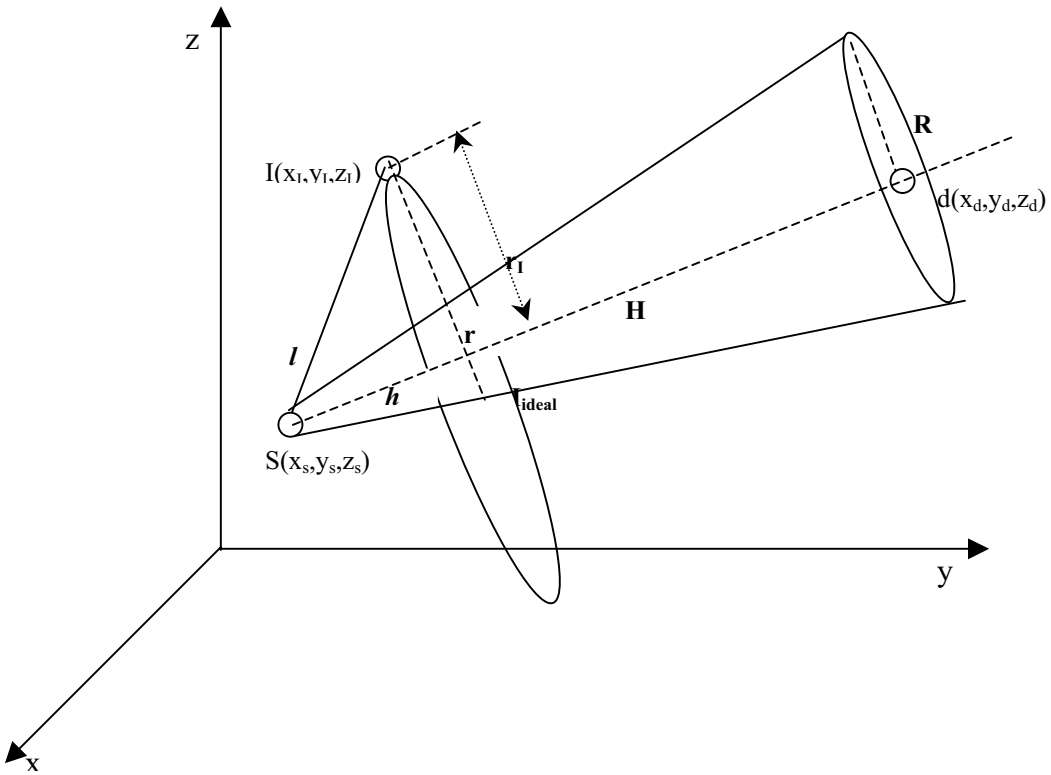


Figure 3.10 Direction field of node d from node S

In the figure 3.10, $I(x_I, y_I, z_I)$, $S(x_s, y_s, z_s)$ and $d(x_d, y_d, z_d)$ are the known locations of the nodes I, S and *d* respectively. *h* is the distance of the intermediate node *I* from the source *S* along the centre line wile *l* is the actual distance of I from S. I_{ideal} is the equivalent location of node *I* along the line joining *S* and *d* (last known location of D). $\mathbf{r_I}$

is the distance of I from the point I_{Ideal} while **r** is the maximum allowed deviation from I_{real}. **R** is the maximum deviation from location d while **H** is the last known distance of d from S.

From relations: $\frac{r}{R} \mid \frac{h}{H}$, it follows that $r \mid \frac{h * R}{H}$ …….(10)

For simplicity, we will consider the nodes to be located in a three dimensional plane.
From vector geometry, if we know the vectors of three sides of a triangle, we can calculate the angle between two adjacent sides. Figure 3.11 illustrates this angle and the associated vectors.

Using vector coordinates, $I \mid \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \mid /x_I, y_I 0,$

$S \mid \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \mid /x_s, y_s 0$ and $d \mid \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \mid /x_d, y_d 0$ …….(11)

$Cos\chi \mid \frac{I.d}{\|I\|.\|d\|} \mid \frac{/I 4 S0 * /d 4 S0}{\|I 4 S\|.\|d 4 S\|}$ …….(12)

(see figure 3.11)

$Cos\chi \mid \frac{b}{l} \bullet \frac{/I 4 S0 * /d 4 S0}{\|I 4 S\|.\|d 4 S\|} \mid \frac{b}{l}$

$b \mid \frac{l 1 /I 4 S0 * /d 4 S0}{\|I 4 S\|.\|d 4 S\|}$

…….(13)

But $\|I 4 S\| \mid l$ and $\|d 4 S\| \mid H$

Substituting, we get $b \mid \frac{/I 4 S0 * /d 4 S0}{H}$ ……..(14)

From similar triangles,

$\frac{b}{H} \mid \frac{r}{R} \bullet r \mid \frac{R 1 b}{H} \mid \frac{R 1 /I 4 S0 * /d 4 S0}{H 1 H}$

$\mid \frac{R}{H^2} 1 /I 4 S0 /d 4 S0$…….(15)

Now $/I 4 S0 * /d 4 S0 \mid \begin{pmatrix} i_1 4 s_1 \\ i_2 4 s_2 \end{pmatrix} * \begin{pmatrix} d_1 4 s_1 \\ d_2 4 s_2 \end{pmatrix}$

$\mid /i_1 4 s_1 01 /d_1 4 s_1 02 /i_2 4 s_2 01 /d_2 4 s_2 0$

Therefore replacing polar coordinates with corresponding planer coordinates,

$r \mid \frac{R}{H^2} 1'' /x_I 4 x_S 0 x_d 4 x_S 02 /y_I 4 y_S 0 y_d 4 y_S 0 €$…….(16)

r_I can easily be obtained from the equation: $r_I = \sqrt{l^2 - b^2} = \sqrt{l^2 - \left(r - \frac{H}{R}\right)^2}$ .....(17)

Now from previous argument, if $r_I > r$, the node I is outside the direction field of the destination node d. The node then checks if it has more recent information about d than the preceding node and decides whether to forward the packet or drop it.

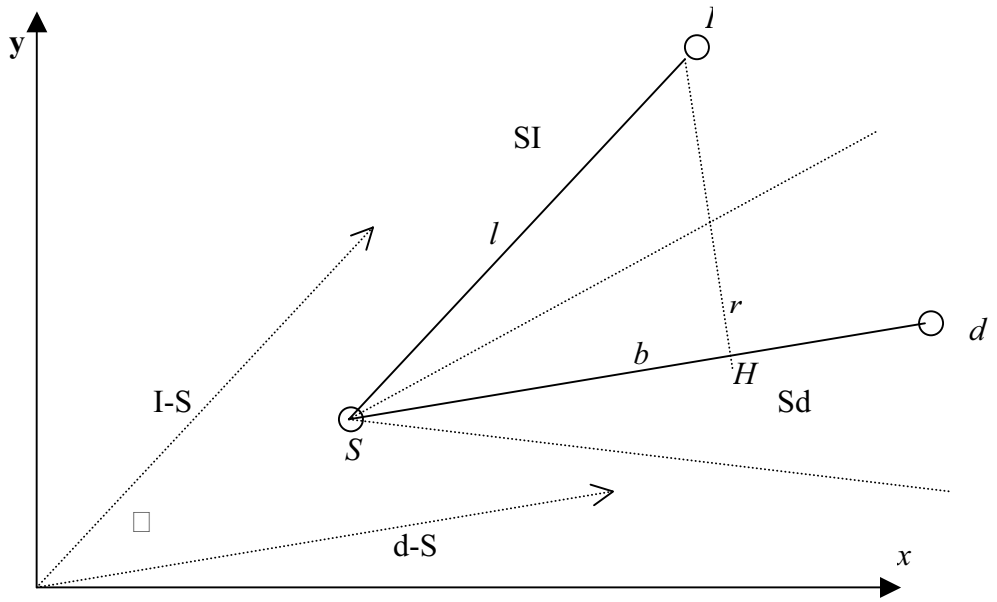


Figure 3.11 Planer vector representation of the nodes

The value of ***R*** can be determined using various methods. For simplicity, we have used average speed ($v_{last\ known}$) and time elapsed ($t_{elapsed}$) for the approximation.

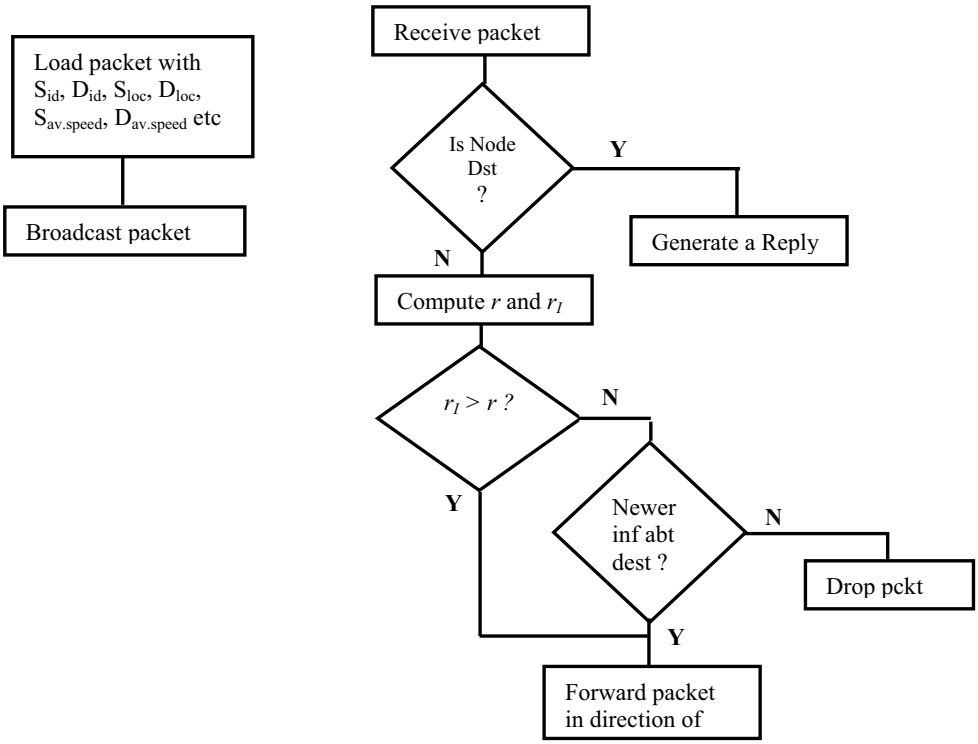


Figure 3.12 The Schematic diagram of the scheme

A more realistic estimation of R would be considering history of movement of the destination node and take average deviations from d in all directions i.e. $\in x$, $\in y$ and $\in z$. The diagram in figure 3.12 explains this scheme.

3.4.3 Network level routing overhead management scheme

3.4.3.1 Algorithm of the scheme

A scheme of handling the traffic overhead based on history of movement of destination nodes is hereby suggested. This scheme involves first tracking the destination mobile node, and then trying to find a better route to the source. The method significantly reduces the area of packets broadcast, but at a cost of time required for tracking the destination node. However the tracked route is always available and updated reactively. We expect the search time not to differ much from the traditional methods since the time for locating the destination is compensated by the savings on the time we would require to send the reply before an information packet can be placed on the route (the scheme places the information immediately after the reply packet). Every node in the network will keep a route to its destinations, which is marked as *varied, expired* or *invaried.* When a route is marked as *invaried* it is deleted after some predetermined period of time. *Varied* routes will be the ones that have been used before the expiration time while *expired* route will be the ones that have not been used from the expiration time onwards. The decision of deleting routes is made when a source node doesn't receive any information about the destination node after a predetermined time. The value of this will depend on the size of the network and the average time required for a packet to traverse from that node to the extreme end of the network.

When a source has a packet to send to a destination, it first checks if it has a varied route and uses it if available. If not, it checks if it has an expired one, whereby it sends the tracking packet using this route. If it does not have a possible route to the destination, it uses the common flooding algorithm. In order for the source node to initiate a route search, it sends a packet to the destination whose location is known. This packet is hereby called "*DestSearch"* packet. This packet is similar to the DSR's *route request* packet in that, it carries information of all the nodes traversed on the way to the destination. On arrival of the *DestSearch* packet at the destination, the destination node initiates a request packet towards the source. The area of route search is a union of two regions: (i), the area enclosed by the diameter of the destination node's transmission range and diameter of the source's maximum displacement (from calculations of its velocity) plus its transmission radius and (ii), the area between the line joining two nodes and the track followed by the *DestSearch* packet. Figures 3.13 show the first region, which is covered by the DestSearch packet (a route already known to the source), while 3.15 illustrate the union of the two regions.

Figure 3.15 shows the area between the source's (S) and destination's (D's) diameters. This is the region enclosed by the diameter resulting from the transmission radius of S plus its expected displacement from its last known location (carried by the DestSearch packet) the transmission radius of D plus its displacement and the route

formed by the Destsearch packet. The union of these regions gives the region of route search by the reverse request packet from the destination D towards the source S. The parameters needed for determination of the existence of a node in the pre-defined are of search are shown in figure 3.16. This figure shows that in order for an intermediate node to forward the reverse packet, it should either be in the transmission area of the sender or the destination, or lie in either of the two defined zones.

In order for an intermediate node to know whether it can forward a reverse packet, it does some computation of its location relative to the prescribed zones. We suggest two methods of doing this but will use one of them for our experiment.

One method would be to calculate the distance of the node from the line joining the centres of the source node and the destination node. If the distance is beyond the tangent touching both circles, the node is out of the zone. One circle is formed by the sum of the radius of transmission of the source and its displacement from last known location while the other is formed by the sum of the radius of the destinations and displacement from its last known location.

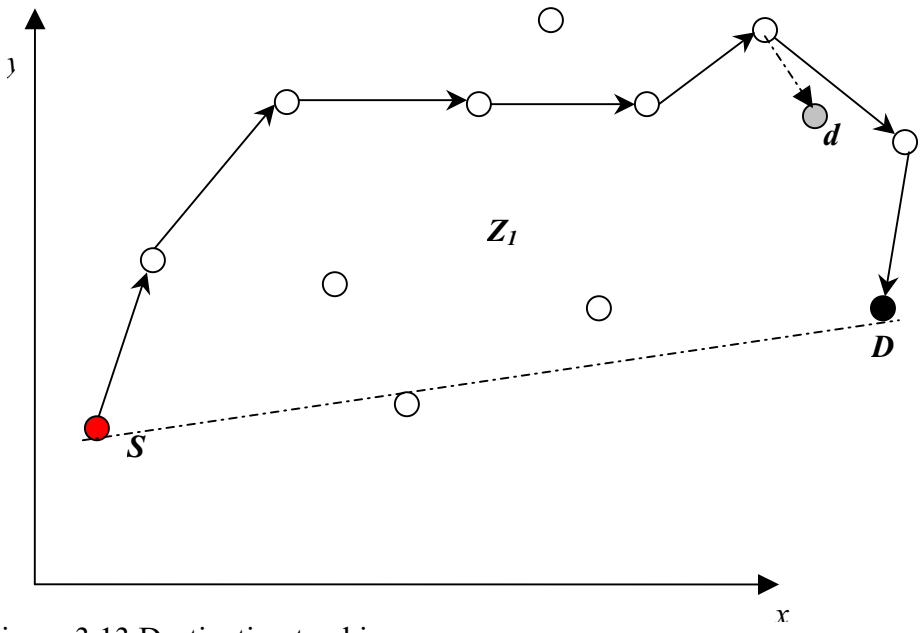


Figure 3.13 Destination tracking

In figure 3.18, we show the parameters needed by an intermediate node ***I*** to establish whether it lies within the region $\boldsymbol{Z_2}$ using the first method. This test is carried out before the test for the existence in the region $\boldsymbol{Z_1}$. It is only after the test for Z_2 fails that an intermediate node tests for its existence in $\boldsymbol{Z_1}$. We use simple trigonometric rules for the first test and use information carried in the DestSearch packet to perform the second test.

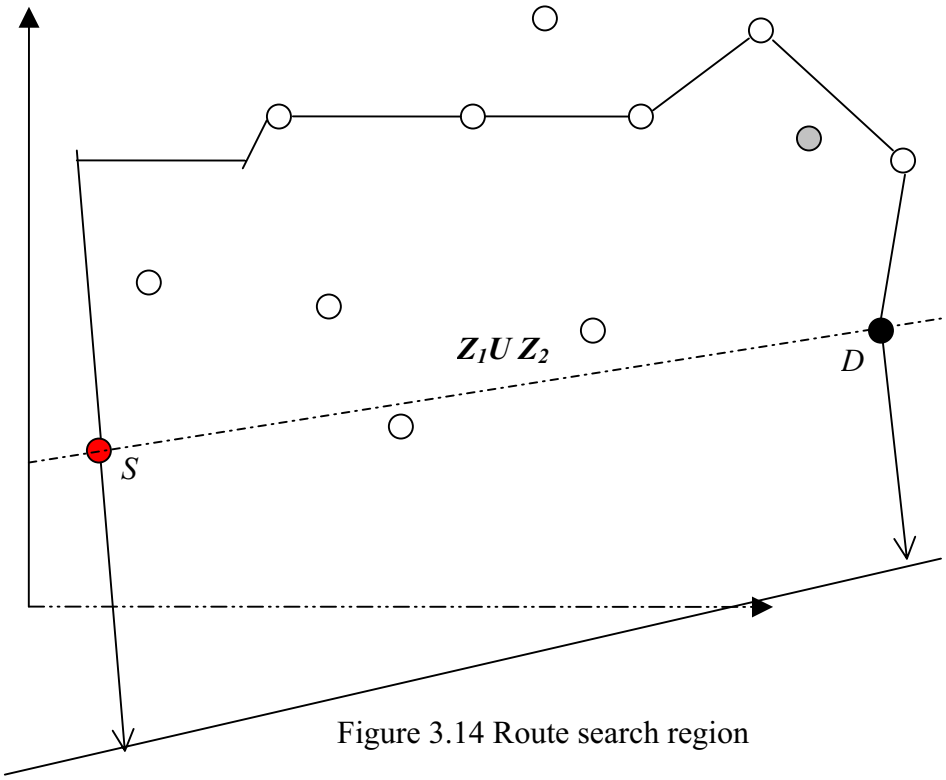


Figure 3.14 Route search region

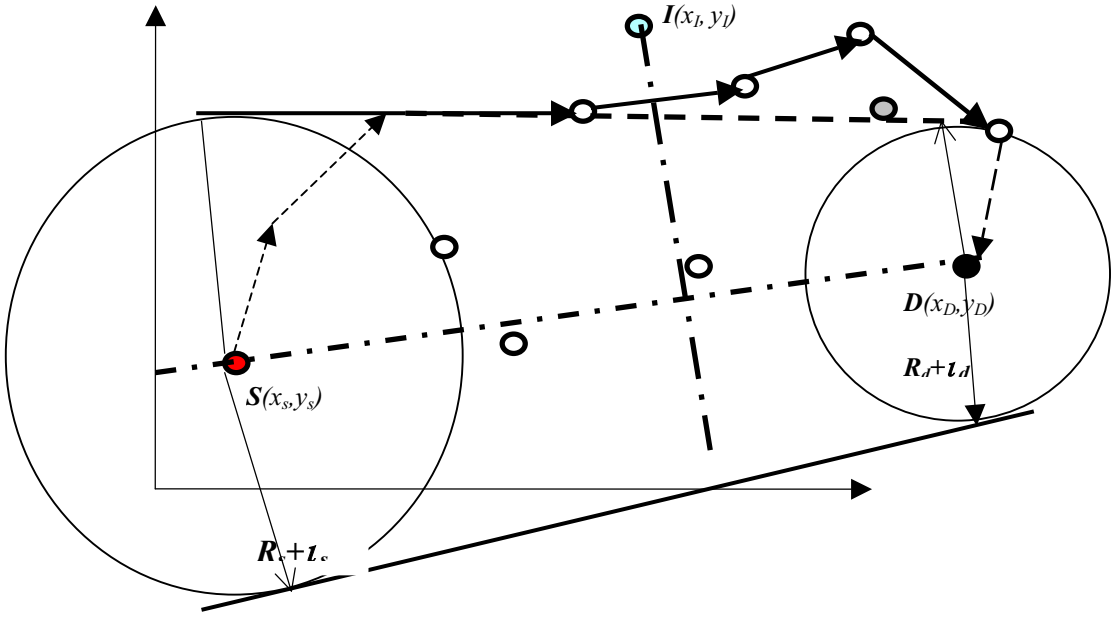


Figure 3.15 Defining the Region of packet forwarding

From figure 3.17, the parameters needed for the test for Z_2 are r_I and l, whose magnitudes are to be evaluated and compared.
The coordinates of ***S, I*** and ***D*** form a ***a-b-c*** triangle. The values of ***a, b*** and ***c*** can be obtained from:

$a = \sqrt{(X_I - X_S)^2 + (Y_I - Y_S)^2}$, $b = \sqrt{(X_D - X_S)^2 + (Y_D - Y_S)^2}$ and $c = \sqrt{(X_D - X_I)^2 + (Y_D - Y_I)^2}$ .....(17)

From the area of the triangle,

$r_I = \frac{2}{b}\sqrt{S(S-a)(S-b)(S-c)}$

$S = \frac{1}{2}(a + b + c)$. .....(18)

The values of h and *l* can be gotten from Pythagoras's theorem and equivalent triangles.

$c^2 = r_I^2 + h^2 \Rightarrow h = \sqrt{c^2 - r_I^2}$

$l = \frac{h(R - r)}{H} = \frac{\sqrt{c^2 - r_I^2}(R - r)}{b}$ (*H=b*).....(18)

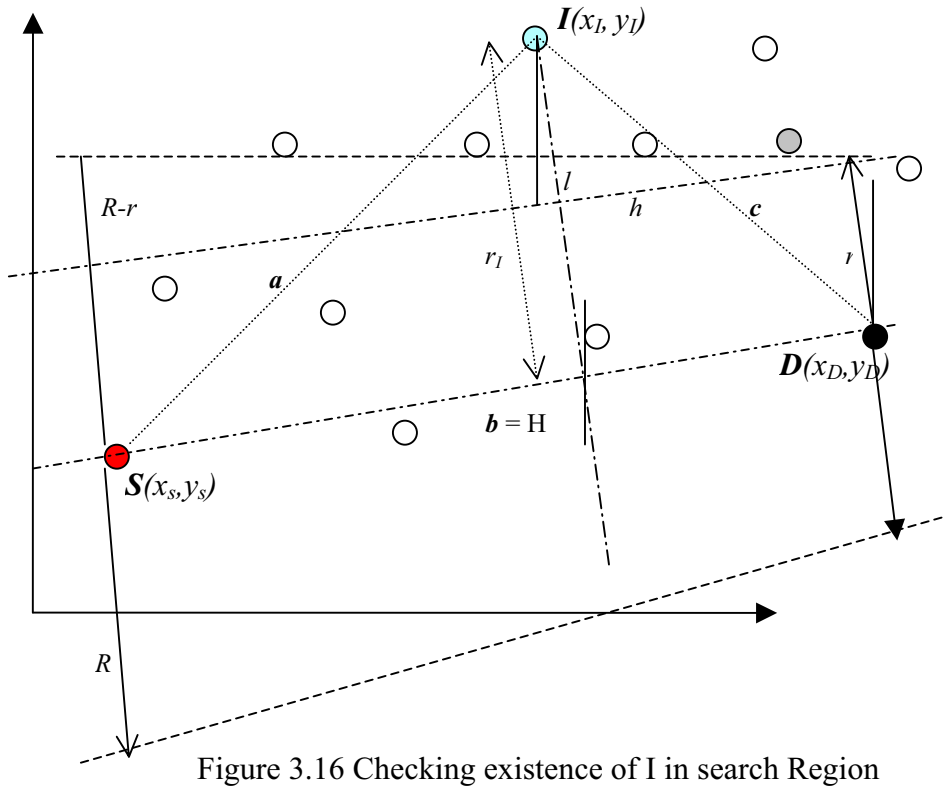


Figure 3.16 Checking existence of I in search Region

In the second method, we need to determine the four coordinates where two lines, each passing through the diameter of one of the two circles and perpendicular to the line joining the two radii intersect the circles. We then determine the equations of the four lines that enclose the region between the two circles. Two of the equations are the equations of the parallel lines crossing the two circles. The other two are the equations of the lines joining the points of intersection with the circles without closing. We then use inequalities to determine the region enclosed by the four lines.

With known values of r_I and *l*, the first test can now be done as: If $r_I <= r+l$ the node is within the region $\boldsymbol{Z_2}$ otherwise we proceed to the second test.

As for the second method, the equations of the lines needed are of the form

$y \mid mx 2 b$

We need to determine the values of *m* and *b* for each equation.

We have the values of the centres of the two circles, which help us determine the gradient m_1 between source and destination. Figure 3.17 illustrates the lines needed and the section enclosed by the four lines.

The values of the gradients of the two lines passing through the diameters of the two circles are both $4\frac{1}{m_1}$ (perpendicular to line DS).

To obtain the points of intersection between the circles and the lines passing through them, we use the equations $/x 4 x_0 0^{\circ} 2 /y 4 y_0 0^{\circ} \mid r^2$ for the circle of radius *r* and centre (x_0, y_0) and the line $y \mid mx 2 b = b 4 \frac{1}{m_1} x$

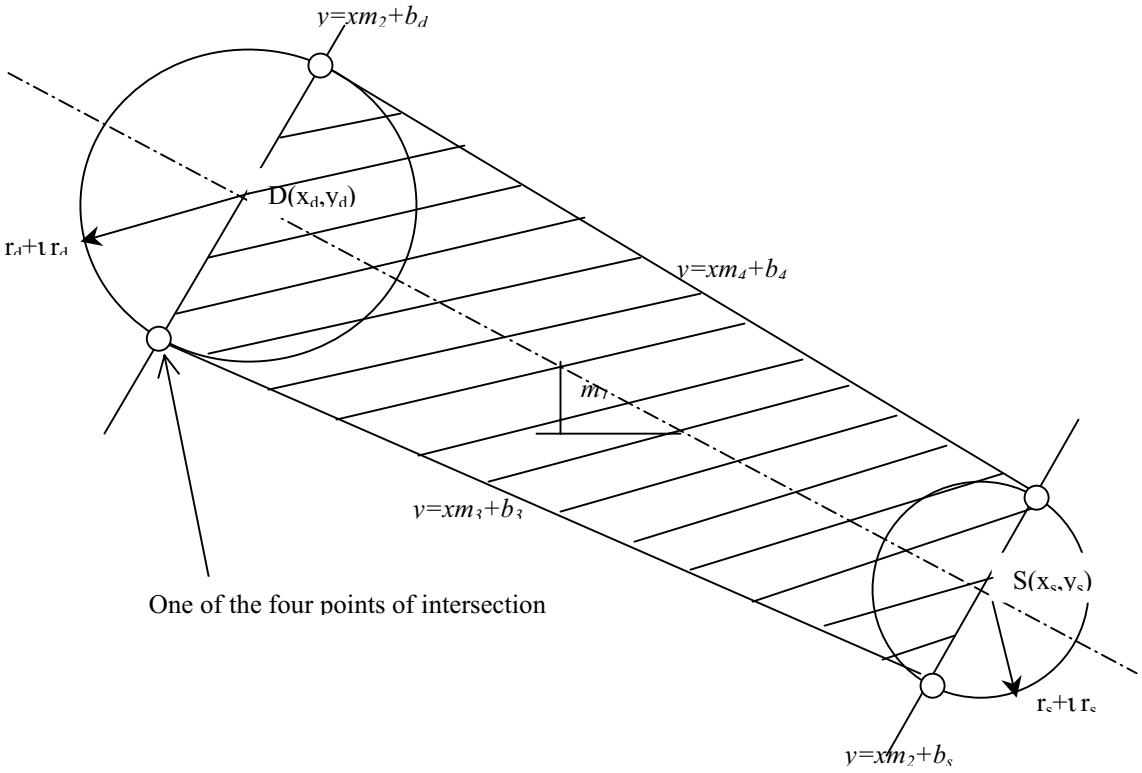


Figure 3.17 Determining existence in zone2 using inequality linear equations

The two pints on each circle have the coordinates:

$$x \mid \frac{my_0 2 x_0 4 mb}{m^2 2 1} \partial \frac{\sqrt{r^2 4 /\!/ y_0 4 mx_0 4 b0/\sqrt{1 2 m^2}0^{\circ}}}{\sqrt{1 2 m^2}} \quad \text{.....(19)}$$

$$y \mid \frac{m^2 y_0 \; 2 \; m x_0 \; 2 \; b}{m^2 \; 2 \; 1} \partial \frac{m \sqrt{r^2 \; 4 \; \text{®}\text{®}\text{©}\text{©}\text{™}\text{™} y_0 \; 4 \; m x_0 \; 4 \; b \; / \sqrt{1 \, 2 \; m^2}}^{2}}{\sqrt{1 \, 2 \; m^2}} \quad \text{.....(20)}$$

We now substitute the values of m, r, b, x_0, and y_0 taking into consideration the following:

If we consider the circle formed by the destination,

$m \mid 4 \frac{1}{m_1}$, r is transmission radius plus displacement of the destination from last known location (can be obtained from values of speed and time elapsed, carried by the reverse packet), $b \mid y_d \; 2 \; \frac{x_d}{m_1}$ where (x_d, y_d) is the centre of the circle and $x_0 = x_d$, $y_0 = y_d$

After obtaining the coordinates of the point of intersection, we determine the equations of the lines passing through these pairs of points on different circles. Their gradients can easily be obtained from the two known coordinates on each line.

For the test for zone Z_1, the receiving node (destination or intermediate) node uses the information carried in the packet. It first ensures that it is within maximum and minimum coordinates of the route map. These are determined by the destination node and loaded into the packet. It then compares its coordinates with consecutive pairs of locations carried by the *DestSearch* packet. It finally compares its gradient to the location on the packet that is closest to the line joining destination and source with the gradient of the line joining the two consecutive locations between which it lies. If its gradient is less, it is within the region otherwise it is not and drops the packet. Figure 3.18 illustrates this. It is important here to note the sign of the gradient so as not to miss the location of the intermediate node relative to the region.

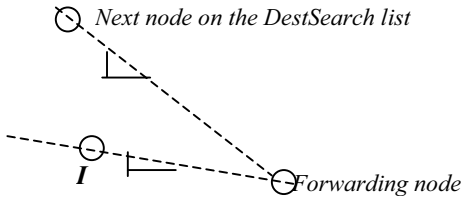


Figure 3.18 gradients comparison

In this scheme, we would simply have used the known location of the destination and the transmission radii of the nodes and their speeds of movement to initiate the request but since the information of the destination node's location may not be within allowed accuracy, we won't risk using it and prefer requesting from the destination for more recent information. Moreover we would have to broadcast the request again within the same region with extensions of the search zone as in LAR if the node happens to be outside region $\boldsymbol{Z_1}$ that would cost us more bandwidth utility, more time to get the route and more congestion in the region. Our scheme guarantees reachability of the information packets to the destination since we already have a route to the destination, which would be

used in extreme situation (i.e. if we fail to get a route within the time a packet would use to reach the destination using the known route). *Nb*: We are trying to look for a better route than the one we already know.

One reason why the region $\boldsymbol{Z}_1$has been included is that in case of failure of convergence using region $\boldsymbol{Z}_2$, it is more likely for mobile nodes in an ad-hoc scenario to revisit areas that they have been before than new areas. The latter is taken care of by the use of the transmission radius plus the possible displacement away from the last known location. Figure 3.19 shows the schematic scheme of our suggested scheme.

3.4.4 Modifications to the scheme

A modification to this scheme would be availing destination information to the source on establishment of a route and availing source's information to the destination. The source would then send the requests towards the destination using the algorithm used by the reverse request message. When an intermediate node receives the message, it responds with an ordinary reply if it has a fresh route to the destination or forwards the packet if it is either having an old route to the destination whereby it indicates in the packet that it has forwarded on an old route, or it is within zone2. When the destination receives the message, it responds with an ordinary reply if it receives the packet from a intermediate node or source node. It however responds with a reverse request if it receives the packet from an old route. If however it receives a request from an intermediate node that is within zone2, it discards any messages that arrive later for the same route. This would save the algorithm the delays caused by the wait of the destination packet by the source in the proposed scheme. The implementation of this alternative is however dependent on the speed of movement of the nodes. It is suitable for low and moderate speed nodes whereby nodes would take long to travel a distance greater than their transmission radii. Results of this are shown in chapter 4.

3.5 Integration of RO management scheme on typical Ad hoc Routing protocols

As mentioned earlier, this scheme does not specify a particular routing protocol although more beneficial to hybrid routing framework. Although not bug free, the AODV has been chosen for implementation as it has a tested ns implementation. ZRP or DDR would have been a better options for this integration, but since their implementations on ns is either unavailable or not fully tested, the better option was AODV.

3.5.1 AODV with RO (AODV_LA) management scheme

In order to have a reflection of the working of the suggested scheme, a few functions were added to the AODV code and linked to the protocol. The functions were also linked to the otcl front end. These functions were: (i) for sending the destination search packet, (ii) for sending a reverse route search after receiving the destination search packet and other functions for testing the existence of a node in the two zones discussed in section 3.4.3. Other added functions to AODV are the path cache (similar to the one used by DSR), added fields on the node data like old route expiration, node location and speed etc.

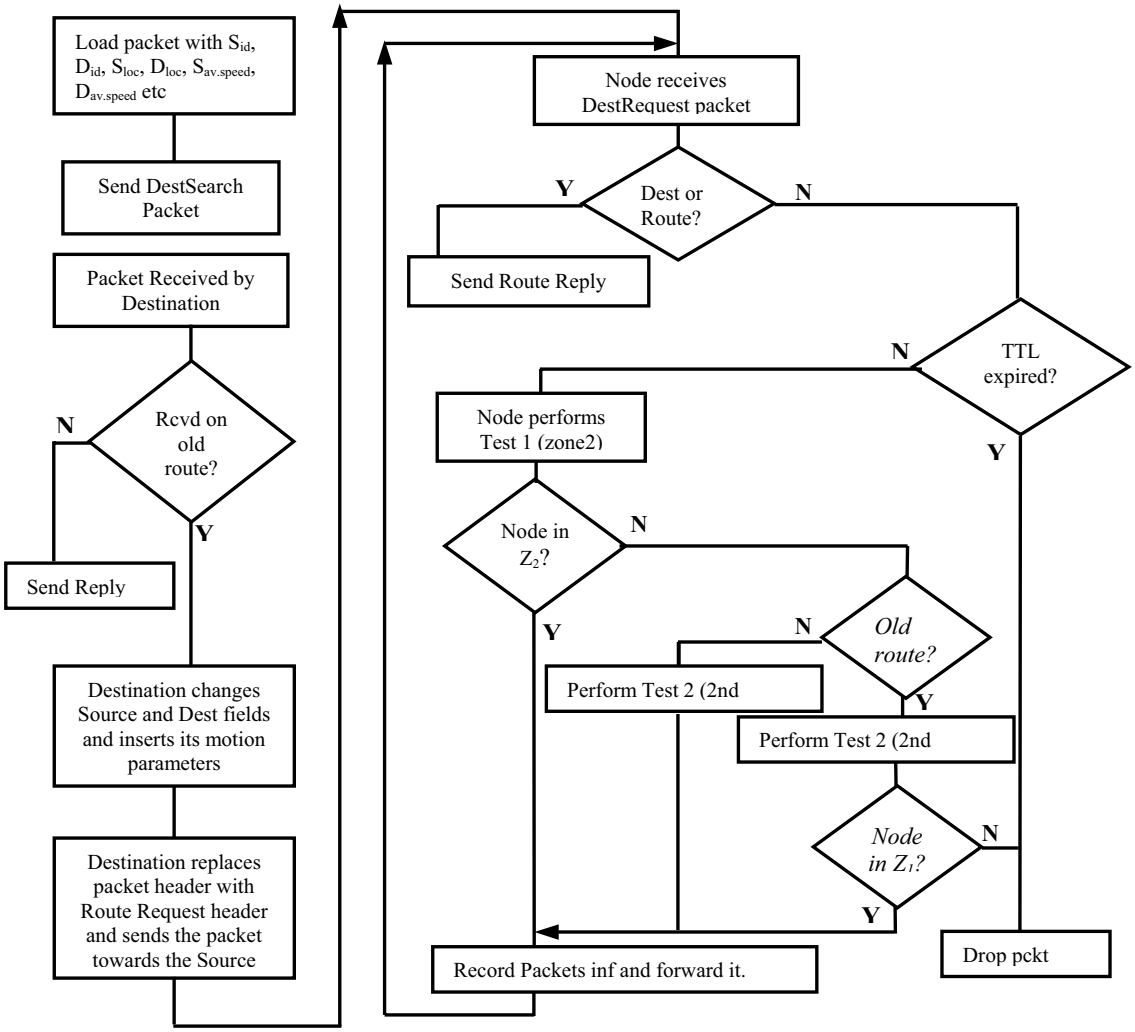


Figure 3.19 Schematic diagram of the algorithm of destination search and limited flooding of request packets

The node initially behaves as in pure AODV for initial knowledge of their destinations. Once a node has received information of a new node, it keeps this information for an extended duration of time after the time for a route to this destination expires. This extension is what is referred to as old route expiration time. It is during this extended time that the node can send a destination search packet. This packet is sent and forwarded like an ordinary information packet. It has the ordinary request packet format with additional information about the route that it passes towards the destination. It also has a destination address instead of the IP_BROADCAST information.

3.5.1.1 Basic algorithm of AODV_RO

At topology initialisation, a node broadcasts a route request (RREQ) to its neighbors as in pure AODV. When a route to a destination expires, it is flagged as route under repair and a destination search request packet (DEST_SEARCH) is sent to the destination. As this

packet travels towards the destination, it collects the map to be used for the checking of existence of a node in zone1.

When an intermediate node receives this packet, it checks whether it is the packet's destination and generates a reverse request packet (REVREQ) if it is the destination. If it is not the destination and it has a route to the destination, it forwards it. Otherwise it drops the packet. The reverse request packet contains, in addition to information contained by an ordinary request packet, the location of the original destination and its speed, time of sending the reverse request, and the route map gathered by the destination search packet.

When an intermediate node receives this packet, it first tests if it is the reverse packets destination. If it is, it treats this as a reply to its request and forwards its information packet using this route. It also resets the route expiration time since this is a fresh route. If the intermediate node is not the reverse packet's destination, it tests for its existence in zone2 and or zone1. If it is in either of the two zones, it forwards the packet towards the reverse packet's destination or drops the packet if it is in neither of the zones. When it forwards the packet, it creates a reverse route to the original destination (reverse packet's source).

Since many nodes will receive and forward the reverse search packet, the reverse route created by these nodes is treated similar to the reverse route created in a pure AODV protocol when a node receives a route request packet. This reverse route is deleted after route request time out is reached. A flag is used in this scheme to differentiate between when a route is under ordinary repair or if it's undergoing a destination search process. This flag is sent to "on" whenever a node initiates a destination search packet and "off" when a route is found or search time expires. Therefore when the route flag is set to route under repair and the destination search flag is set to "off", the route is under ordinary route repair. Otherwise it is under destination search.

If a destination search packet source does not receive a reverse route packet within the old route expire time, it flags down the route. This route is deleted during the next route purging routine. The node will then use ordinary flooding to establish a route to a deleted or flagged down route. Figure 3.20 illustrates a schematic flow of events in an AODV protocol with the RO scheme implementation.

3.5.1.2 Main alterations on AODV packet

The main alteration on the AODV packet is the packet size. An extra field (a list) has been added to the request packet. This list stores the intermediate nodes' movement parameters, namely the location, the node velocity and the node's transmission radius. This packet is labelled destination search packet when first sent, and destination reverse packet when replied by the destination or intermediate node. A node that receives the destination search packet appends its motion parameters before forwarding or replying to it.

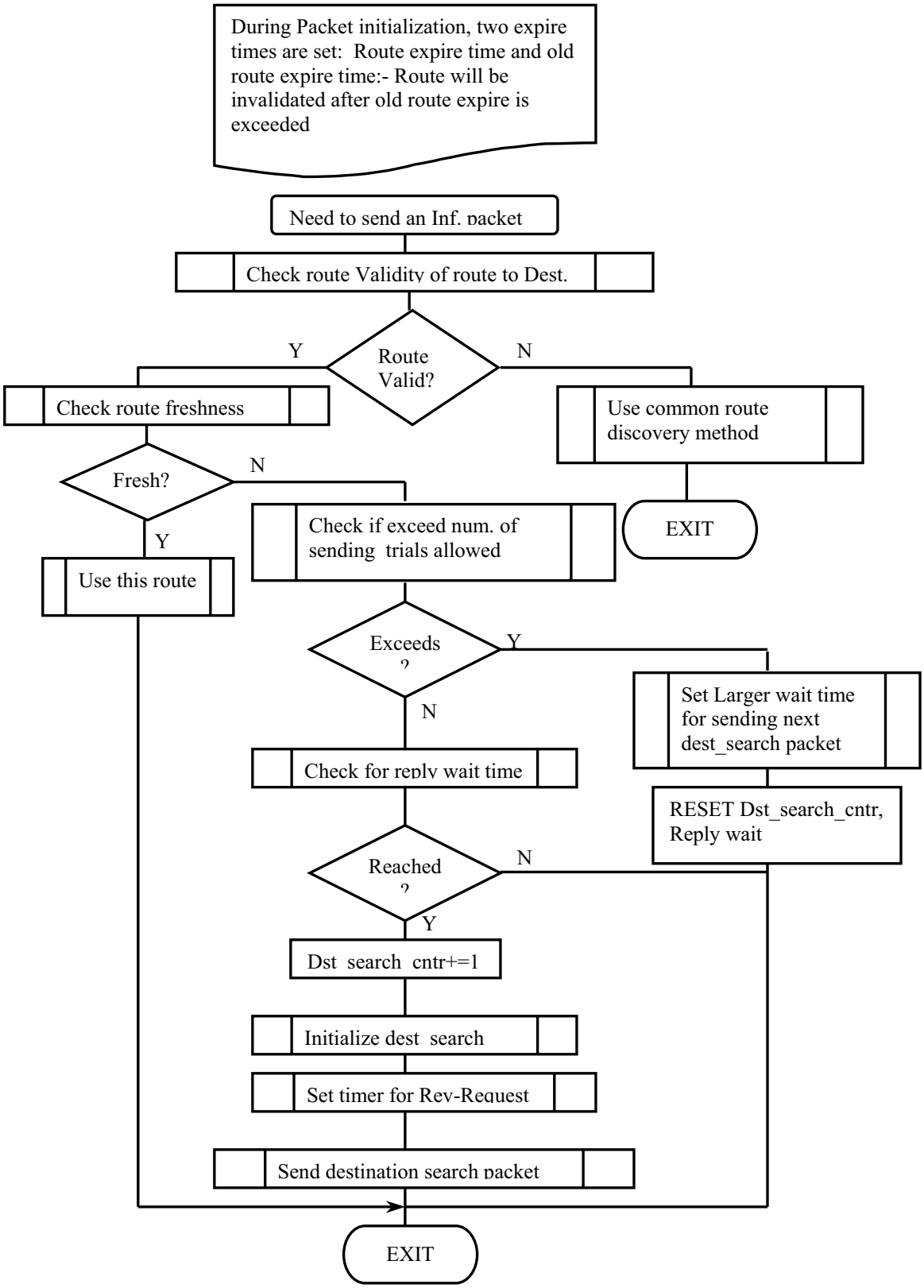


Figure 3.20 Implementation of RO scheme on AODV

When a node receives the reverse request packet, it checks whether it is the destination of that packet and responds with a reply and sends any awaiting packets to the destination. If it is not the intended destination, it checks if it is in the zones defined by the accumulated path on the packet and drops the packet if it is not within the two zones.
Figure 3.21 illustrates a schematic diagram of how the node responds when it receives a destination search packet while figure 3.22 shows how it will respond when it receives a destination reverse packet.

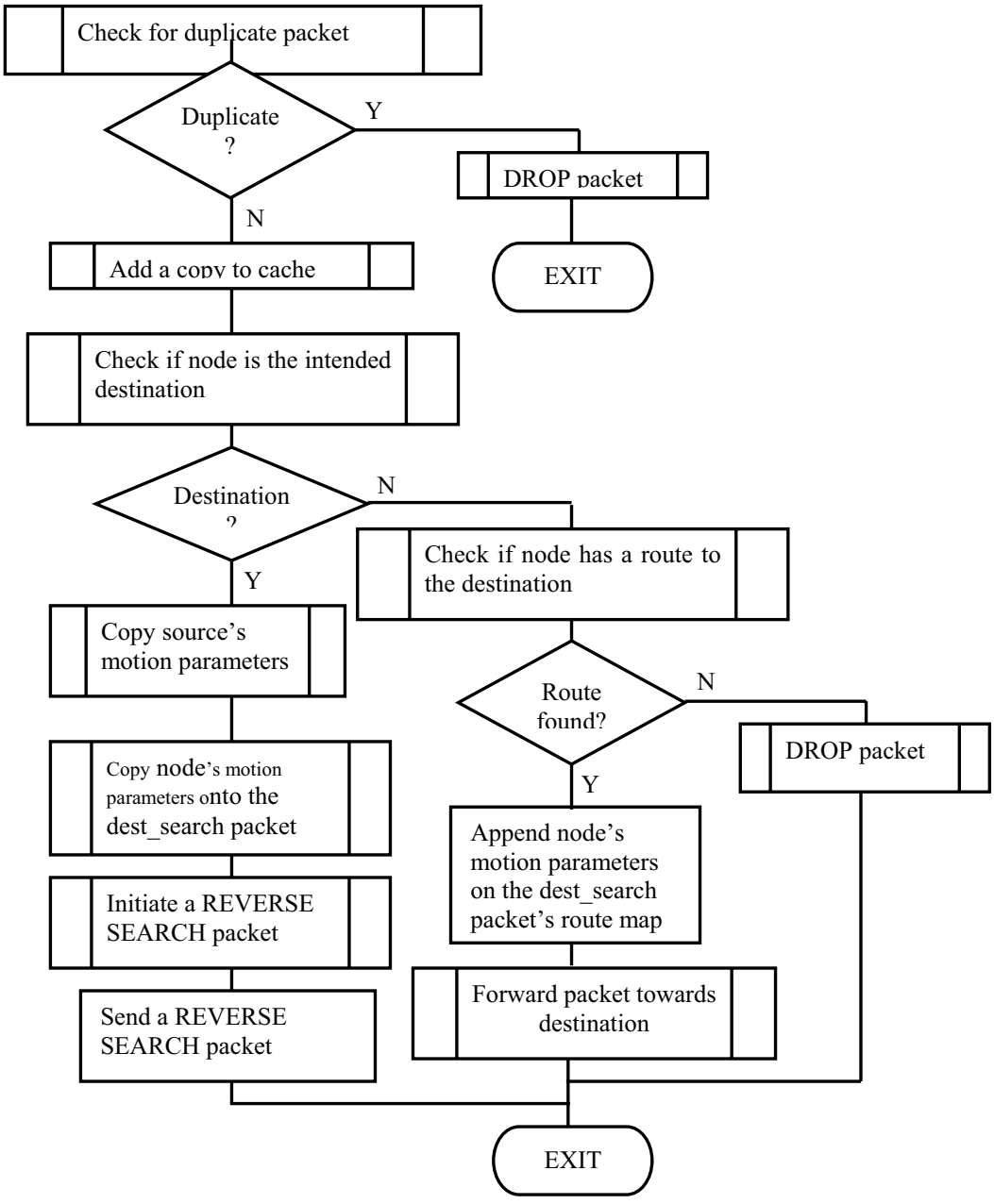


Figure 3.21 Receiving of DESTINATION SEARCH packet

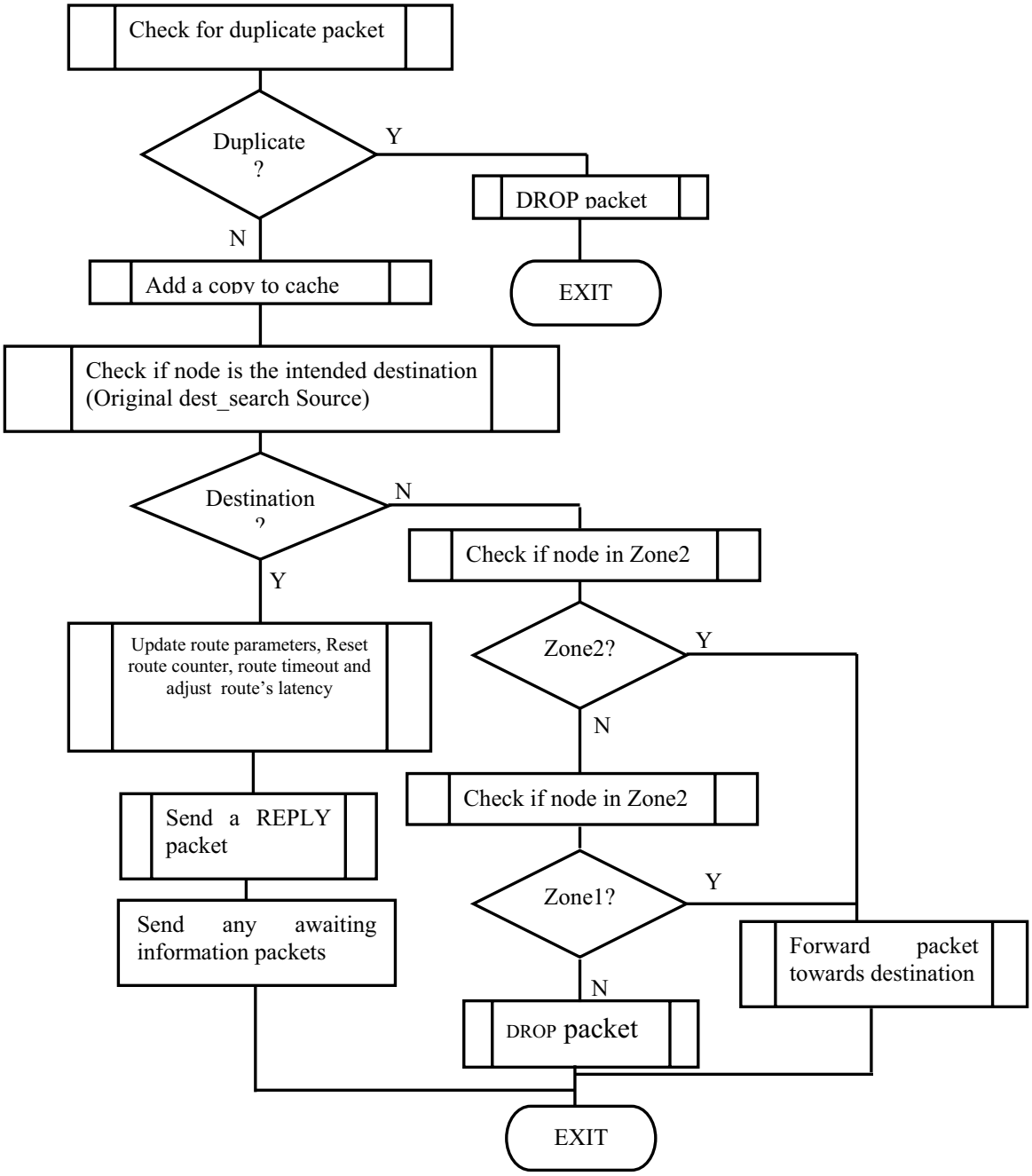


Figure 3.22 Receiving of REVERSE SEARCH packet

3.6 Route Reliability Forecast and Topology maintenance schemes

3.6.1 Route selection strategies

When a node sends out for a route search, it receives a number of replies that come from the destination through different routes. Route selection can be done using different methods. One method is route selection by the destination, whereby the destination chooses the request packet to respond to (normally the first request to reach the

destination). Another way is to allow only the source to select the route from the replies that it receives from the destination. In this case the destination replies to all the requests that it receives. The former is the most commonly used method.

The route selection criteria may be determined by different factors. This is dictated by the application of the routing protocol. For example, route selection may be based on shortest path suitable for real time application. It may also be based on energy usage, which is suitable for sensor network application.

Another factor that may be considered is network congestion whereby a node chooses the less congested route. Such a criterion would be suitable in large-scale networks. Node mobility can also be used as a base for route selection.

It had previously been argued that incorporating many criteria would not be economical in terms of memory use and processing time. However, current technology has made great progress towards alleviating this problem. Nodes are no longer restricted to space and speed as processors have become faster and memory increased. This gives a possibility of using an algorithm, that auto reconfigure itself so as to select the appropriate scheme to use. It is therefore suggested in this work that an algorithm that incorporates multiple properties being used since it offers greater benefit to the routing protocol. This would allow the protocol to be more reliable as it would adjust itself to various requirements.

3.6.2 Reliability determined by multiple properties

For reliability of a route, the method hereby supported is the one discussed in section 3.2.3. In the formula, a third property has been added. This is the minimum link time on the route. The value of k is therefore 3.

The distance dependant time factor is the minimum time taken by a round trip while the link stability time is the minimum from all the link times along the route travelled by the *DestSearch* packet. The most reliable route will be the one with the lowest value of S_{lnk}.

3.6.3 Topology maintenance

In order for a topology to be maintained in a wireless mobile reconfigurable network, the properties mentioned in section 3.6.2 among others are essential for consideration. The most important one is fair distribution of routes. Routes need to be fairly distributed so as to avoid congestion of the network on particular route that may lead to energy drying up on a particular section of the network and subsequent fragmentation. Such congestions may also cause delays due to long waiting on overloaded nodes or loss of information. The overloaded nodes may drop information packets as their waiting queue gets filled up. The formula derived in section 3.2.3 helps in ensuring that we don't over-cloud one section of the network. The weights (tuples) to be used in the formula are however dependent on the application of the protocol. Some networks may be more sensitive to

time while others may require more energy on the routes used. The parameters should therefore be adjusted to suit particular applications.

3.7 Chapter Summary and Conclusions

In this chapter, two main schemes have been developed aimed at improving the efficiency and scalability while enhancing the performance of reconfigurable wireless ad hoc routing protocols through management of routing overheads. These are neighbourhood reduction of overheads, referred to as "link availability forecast" and network level overhead reduction, which is referred to as "location guided routing overhead management scheme".

The link availability forecast scheme targets the hello messages and uses the messages to provide mobility information needed by the algorithm. In this scheme, hello messages are broadcasted only when a node discovers that there is a risk of link breakage with an active neighbour. This allows the protocol to save bandwidth and resources by reduction of hello message broadcasts. Instead of the traditional periodic scheduling of the hello messages, the scheme operates in a hybrid manner in that it schedules long intervals for periodic updates and sends messages reactively when there is risk of loosing a link on an active route. In this way it updates the active nodes timely and updates inactive nodes periodically keeping the network connected.

The second scheme, location guided, targets the destination request messages and uses broadcast messages for carrying the mobility parameters needed by the scheme. This scheme allows the source nodes to include their location and other movement parameters in the request packets, which are recorded by intermediate and destination nodes for future use. This scheme is purely reactive in that it is triggered by the need to send information to certain destinations. Instead of the common flooding method that sends requests to the whole network, this scheme makes use of destinations' locations relative to the source. After the initial knowledge of a destinations location through common flooding, a node keeps this location and estimates the destinations future location using parameters passed to it during a previous communication. This location allows the node to decide the region that the path to the destination is most likely to be. If however the estimated zone fails to give a route to the destination, common flooding is used. This offers savings in energy otherwise consumed by intermediate nodes in the areas that the route is not likely to be found and reduce congestion in those regions thus saving bandwidth.

The two schemes have been implemented separately in order to view the individual impact of each scheme on the existing algorithms. The other reason for individual implementation is that some protocols use hello messages for neighbourhood discovery while other methods use techniques like MAC layer notification. The neutral nature of these schemes allows them to be implemented on many protocols as they are developed on the basics of routing in ad hoc in general. Evidence of the effects of the two schemes is derived in the chapter 4. This chapter also demonstrates the effect of a combination of the two schemes in one protocol.

CHAPTER FOUR

4.0 EVALUATION OF PROPOSED SCHEMES

Evaluation was done by comparing the performances of a current protocol and the protocol with implementations of the proposed schemes. Analysis was based on various metrics that include: (i) Number of control packets (overhead) in the network, (ii) Packets delivery fraction (ratio) (iii) Data packets delay (latency), (iii) Throughput, (iv) Average delay, and (v) Scalability. The AODV has been chosen as a typical reactive routing protocol for this illustration. Comparison has been done using pure AODV that uses flooding method, AODV with Link availability implementations (AODV_LA), AODV with network level routing overhead reduction implementation (AODV_RO) and a combination of the two schemes (AODV_LARO).

4.1 Simulation Environment

The simulation tool used for testing the performance of the scheme is the Network Simulator (ns-2) [54] from UC Berkeley University. Mobility extension to ns that was developed by the CMU Mornach project at Carnegie Mellon University [54] has been used to simulate mobile wireless radio environment. The simulator and the mobility extensions are explained in more detail in appendix I section of this dissertation. To justify the working of the new schemes, modifications were done on the reactive AODV routing protocol.

4.2 Simulations Methodology

The simulation methodology used was the same as that used in the validation in section 2.6.

4.2.1 Mobility model

As in the previous experiment (chapter two), the "Setdest" tool was used for generating the nodes' movement patters and "Cbrgen" tool for generating the traffic. The Cbrgen tool was modified to the suitability of the experiment. This was done by changing the timing of traffic being generated. Instead of the traditional continuous flow of traffic after start off, the program was adjusted to generate the traffic using random intervals selected between the start and end of the simulation. The simulation time was divided into four equal portions within which traffic was generated. In each of the intervals, the start of the traffic generation was chosen as the start of the preceding quarter, while the stop time was chosen randomly between the start and the end of that particular quarter. The reason for doing this was to model possible unpredictable intervals of traffic generation since in real application traffic can be generated randomly and at random intervals. Four equal portions were however chosen conveniently only for data generating purposes. Otherwise random intervals could also have been chosen.

The working parameters were selected as follows:

Average speed was chosen between 2 and 16 m/s, pause time was chosen to be approximately zero to allow aggressive change in mobility patters, queue length was set at 50 packets, Two choices were made for the grid. One being 1000 by 700 m and another 1500 by 1000 m. Simulation iterations were repeated with 20 seconds for the first trial, 40 seconds for the second trial and 60 seconds for a third trial. The number of nodes chosen was between 10 and 50 with between 8 and 40 CBR-Sources for the first grid, and between 20 and 60 nodes with between 8 and 48 CBR-Source connections for the second grid. The traffic transmission rate was chosen between 2 and 16 packets per second. Each set of parameters was tested on the pure AODV, AODV_LA (with link availability), AODV_RO (with network level scheme) and AODV_LARO (with a combination of the two schemes) respectively. The actual values used can be found in figures 4.1 to 4.12.

4.2.2 Scenario

Using randomly generated traffic and movement models, different scenarios were set up and observation of the different protocol implementations noted. Since different scenarios gives different results for the efficiency of the protocol under investigation, determination of the best protocol in terms of efficiency and performance would only be justified after a large number of iterations. This is because each scenario has a different topology. The topology plays a great role especially for the second scheme (network level scheme). One scenario may have nodes concentrated more at one section of the grid making the scheme have no effect neither on efficiency nor the performance. In fact this may lead to more traffic and more delays as a result of the required computations. This is however a rare case. More simulations would therefore yield even better reflection of the expected results.

Four sets of experiments were carried out so as to have a wide analysis of the effect of the two schemes. These are:

(i) Varying the node density in the network (nn).
In this experiment, the average speed (node mobility) was fixed at 10 m/s and transmission rate set at 8 packets per second. The node density was increased with steps of 5 from 10 to 50 for the first grid and steps of 10 nodes from 20 to 80 nodes for the second grid choice. The node to CBR-Sources ratio was maintained at 0.8.

(ii) Changing the maximum average speed of the node movement (mobility).
Here, the node density was maintained at 30 nodes with 24 CBR-sources for the first grid choice and 60 nodes with 48 CBR-Sources for the second grid choice. Transmission rate was fixed at 8 packets per second in both cases. Mobility was increased with steps of 2 from 2 to 16 m/s.

(iii) Varying the number of communicating pairs (CBR-sources).
In this case, the number of nodes (node density) was fixed at 30 and 60 nodes for the first and second grid choices respectively, the average speed fixed at 10 m/s while the transmission rate was kept at 8 packets per second. The number of communication pairs was increased with steps 2 from 10 to 24 CBR-sources for the first grid choice and with steps of 4 from 20 to 48 CBR-sources for the second grid respectively.

(iv) Changing the rate of traffic flow (transmission rate or traffic load) for different simulations.

Here, the average maximum velocity was set at 10 m/s and the node density set to 30 nodes with 24 CBR-sources for the first grid choice and 60 nodes with 48 CBR-sources for the second grid choice. The transmission rate was increased with steps of 2 from 4 to 14 packets per second.

For the comparison of the performance of the routing protocols, the metrics used were: Average throughput, delivery ratio and average-end-to-end delay. The efficiency was determined conclusively from the traffic encountered congestion reduction.

4.3 Simulation results

As this research is concerned with the routing overheads, three types of overheads are analysed. These are the request packets, the hello messages, the reply packets and error messages generated when a node becomes unreachable. It is important to note here that not all the types of overheads are analysed. The reason being that, the schemes hereby developed do not have direct effect on them all. They however play an important role in the efficiency and performance of the protocols in general. This has thereby been left as a future work analysis.

The experiments were first done with a grid of 1000 by 700 m for a simulation time of 20 seconds each set having 12 iterations. While the simulation time and number of iterations were low, it gave an impression of the expected results. The experiment was repeated with a larger grid (1500 by 1000 m) for a simulation time of 40 seconds, again with 6 iterations per set. This gave a better impression of the expected results. The two experiments were done again with a longer duration (60 seconds) each having ten iterations. Simulation results were extracted in tabular form and graphs for different sets of experiments obtained for each separate scheme and a combination of the two schemes. The different results for the different schemes are detailed in the appendix sections. The results for the three trials (20 seconds, 40 seconds and 60 seconds) showed similar trends. The only differences were the magnitudes of the parameters and margins between the different protocols. Similarly, the trends for the 1000 by 700 m grid were similar to the 1500 by 1000 m grid's experiments again with differences in magnitudes and margins between different protocols. Only one set of the results (1500 by 1000 m grid) is shown in this chapter for demonstration purposes. Other selected tables and graphs are included in the appendix section.

4.3.1 Link Availability forecast scheme

The results in this sub-section show the variation of hello messages for pure AODV compared with AODV_LA. Graphs for four sets of experiments where different parameters were varied were drawn and the trends of the graphs explained.
Since this scheme deals with node level broadcasts, results extracted were for hello messages (which were the targeted node level broadcasts) and performance metrics. Following are the results for the different parameters.

(i) Varying the node density

Figure 4.1 shows a general increase in hello messages with increase in node density as expected. The graph however showed a surprising uniform trend for pure AODV. The link availability scheme significantly reduced the number hello messages. This was due to the fact that the messages were only broadcasted when a link was at a risk of breaking and only for active routes. If however a link breaks for an inactive route, no hello message was broadcasted. The reduction is therefore at two levels: one from prediction of link breakage and the other from broadcasting only for active routes. The graph shows a 95 % reduction in hello messages. The gap however widened with increase in the node density. It is worth mentioning here that the % reduction is only for the hello messages that contribute to the overheads (there are other typed of overheads).

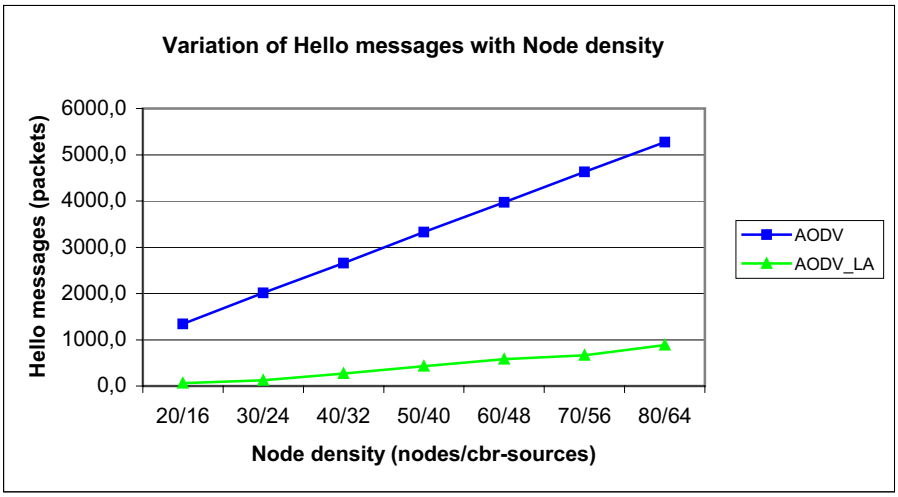


Figure 4.1 Node density against hello messages generated

(ii) Varying the mobility

Figure 4.2 shows that the mobility on the nodes has no effect on the amount of hello messages for AODV. This is because the hello messages are broadcasted periodically. The link availability scheme however is only slightly affected by the mobility since higher mobility means more link breakages, resulting in more hello messages being broadcasted. This effect does not come anywhere near the huge volumes encountered with pure AODV.

(iii) Varying the CBR-sources

Figure 4.3 shows a similar trend with the mobility graph. Again, the periodic nature of hello messages in AODV results in the constant amount of hello messages generated. The link availability scheme however is only slightly affected by the number of CBR-sources since higher number of sources means more routes and greater need for acknowledgement of link failures.

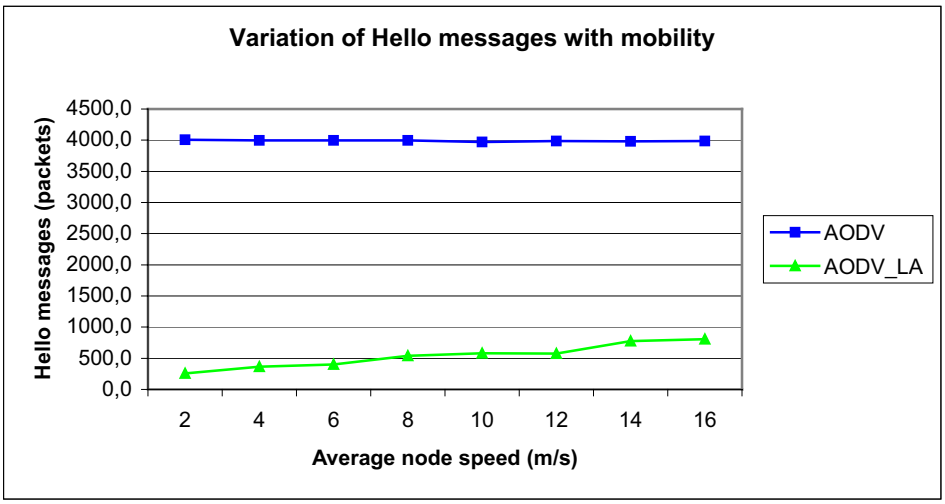


Figure 4.2 Average node speed against hello messages generated

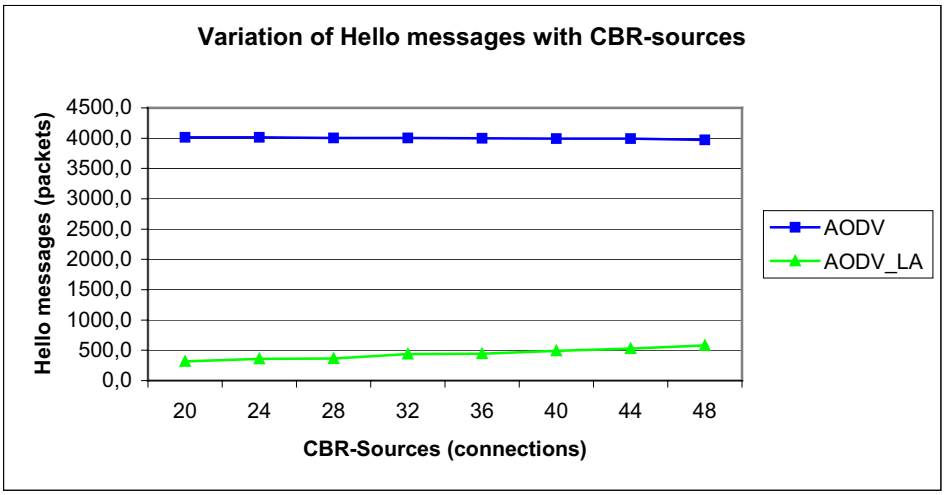


Figure 4.3 CBR-Sources against Hello messages generated

(iv) Varying the transmission rate

In this case (figure 4.4), the number of hello messages is not affected by the transmission rate in both pure AODV and AODV_LA (with link availability forecast). For pure AODV, the reason is that the broadcasting method is purely periodic. With the link availability scheme implementation, the sending is dependant on the topology of the nodes and their mobility patterns and not affected by the information traffic.

4.3.2 Network Level RO reduction scheme

Similar analysis to the ones for the link availability forecast implementation was done for the network level routing overhead management scheme where total route request messages were used in place of hello messages.
The total request messages used was the sum of the initial request messages through flooding plus the eventual requests directed to specific zones. It is worth noticing at this level that in case a route cannot be found using the scheme, the protocol resumes to general flooding.

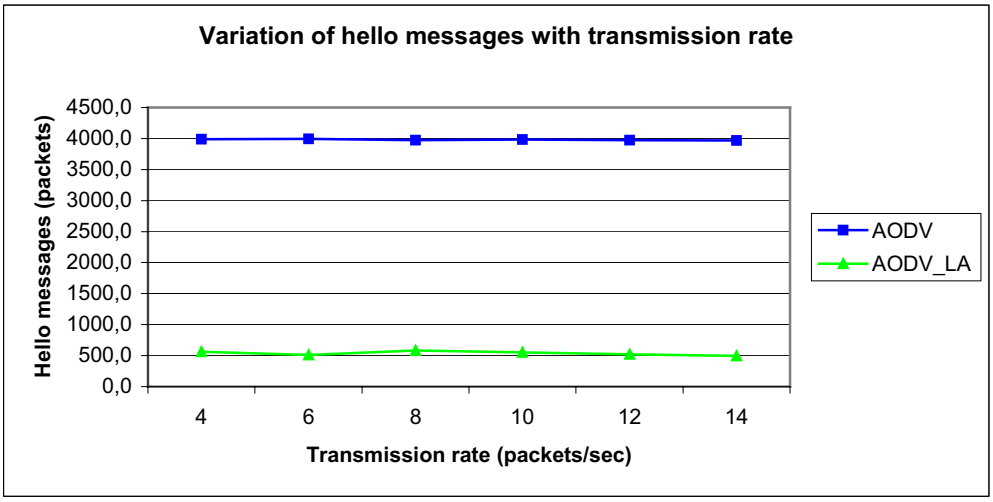


Figure 4.4 Transmission rates against Hello messages generated.

(i) Varying the Node density

As the graph in figure 4.5 indicate, while adapting a general trend of increase in request packets with increase in the node density, the network level scheme results to a 26% reduction at high node density. The margin between the protocol with and without the scheme increases with increase in density. There is however no significant savings at node density as low as 40 nodes.

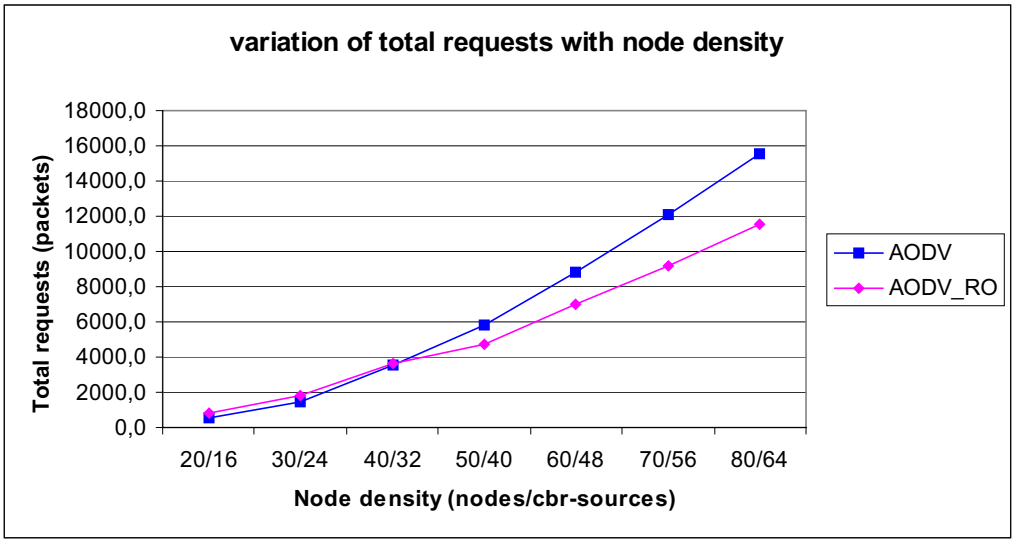


Figure 4.5 Node density against total route requests generated

(ii) Varying the mobility

As shown in the graph of figure 4.6, there is a general increase in request messages with increase in mobility. This is because request messages are broadcasted on demand either due to availability of information messages to send to a destination whose location is not known or when a route to a known location has been invalidated due to link breakage or aging. In this case the increase is mainly due route invalidation due to braking of links due to high node mobility and aging of routes.

The network level scheme however has fewer requests due to the reduction of broadcast zone offered by the scheme. It gives a reduction of over 20 % for velocities above 8 m/s. The trend (%) however increases with increase in mobility.

(iii) Varying the CBR-sources

The graph in figure 4.7 indicates, the number of request packets broadcasted in both pure AODV and AODV_RO increased with increase in the CBR-Sources. The network level scheme reduces the number of requests especially with many connections. There is a general 19.5% reduction with the proposed scheme. The increase in the difference at high CBR-Sources is probably due to high general increase in the overall overhead traffic.

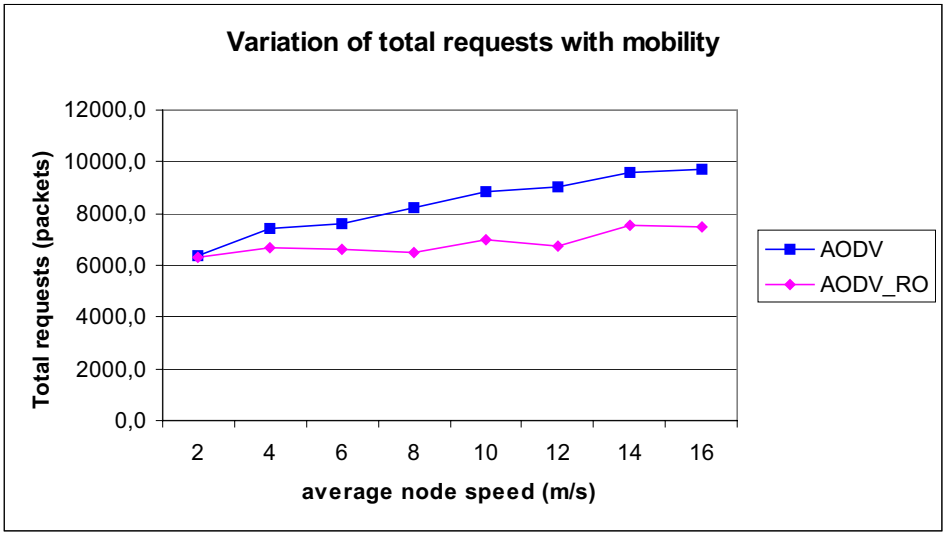


Figure 4.6 Average node's speed against total route requests generated

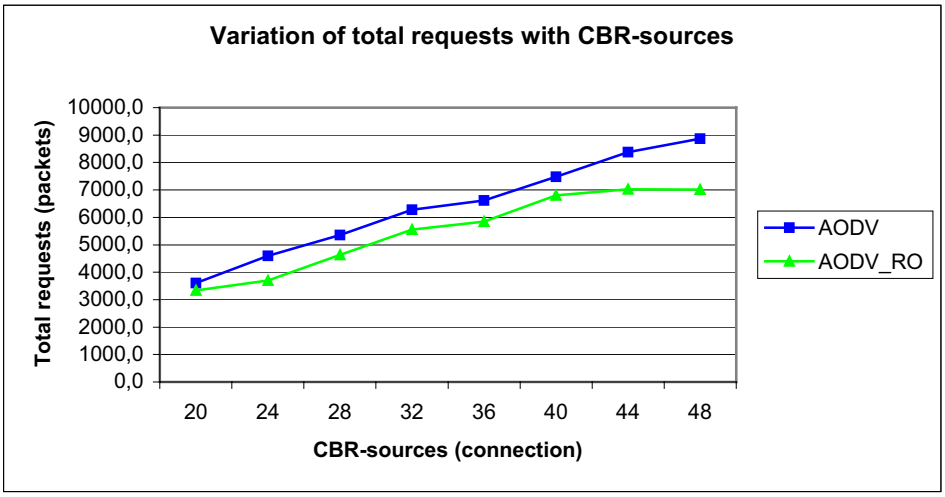


Figure 4.7 CBR-Sources against total route requests generated

(iv) Varying the transmission rate

As the graph in figure 4.8 shows the amount of requests generated changes slightly with increase in the transmission rate. This is because the increase in rate will not greatly determine the need for new route establishment unless of course a link breaks before a transmission is complete this results in the slight increase shown in the graph. However, the network level scheme results in fewer requests than the pure protocol. There is a general 22% reduction in routing requests.

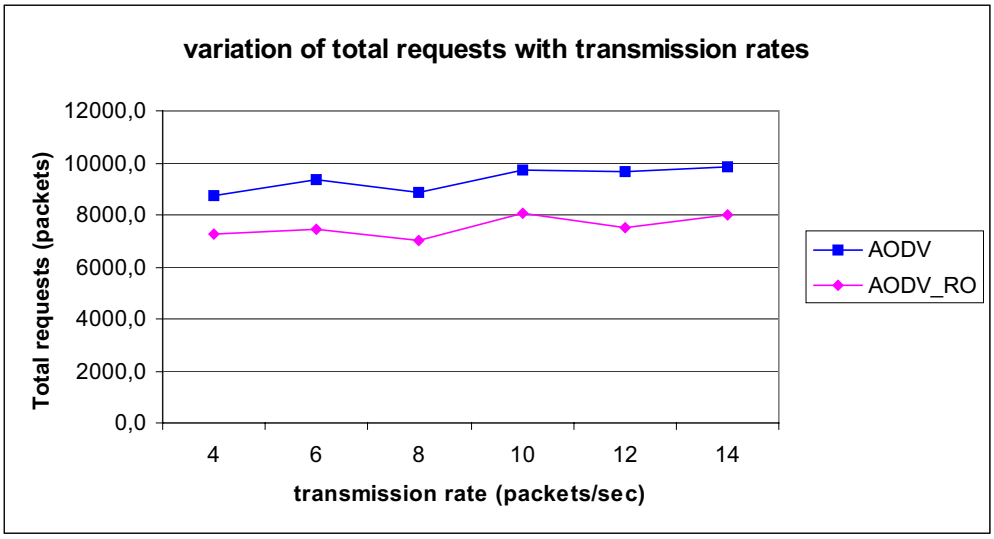


Figure 4.8 Transmission rates against total route requests generated

4.3.3 Combination scheme

For the combination of the link availability and network level overhead management scheme, the results for separate schemes and the overall routing overheads are shown. The results show a combined effect by the node level and network level schemes. Detailed effects for the different experiments follow.

(i) Varying the node density

As shown in the graph of figure 4.9, a combination of the two schemes shows the highest reduction in overall overheads. This indicates a double effect by the combination of the two schemes. The range of reduction increases with increase in node density. The difference is as high as 38.6% reduction at high node density.

(ii) Varying the mobility

Again as shown in the graph of figure 4.10, there seems a double effect by the combination scheme as compared to the individual schemes. The combination results to a general 40.3% reduction on the overall overheads.

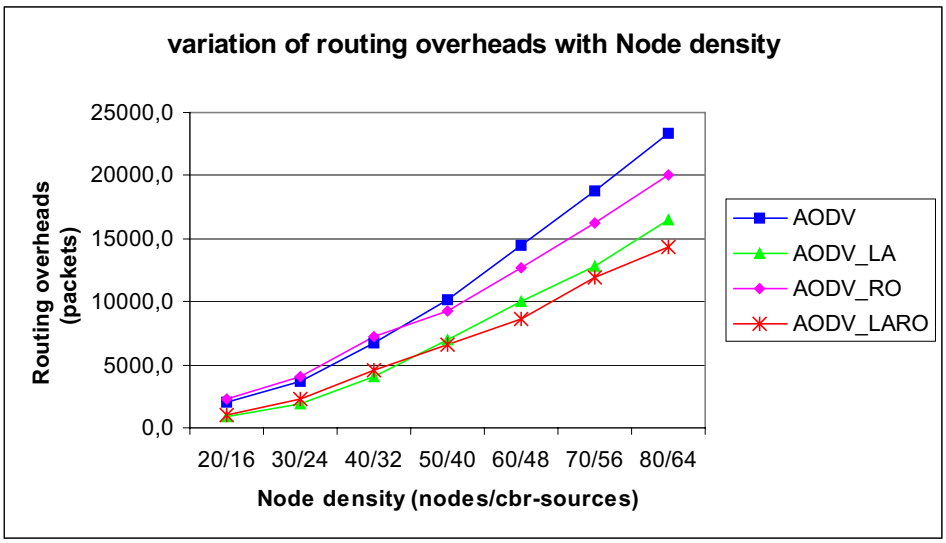


Figure 4.9 Node density against overall protocol overheads generated

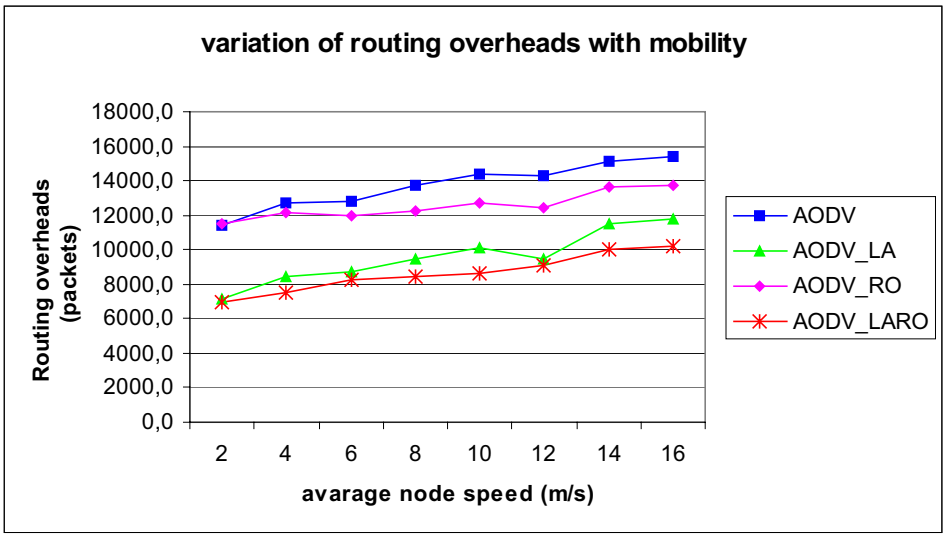


Figure 4.10 Average node speed against overall protocol overheads generated

(iii) Varying the CBR-sources

The graph of figure 4.11 demonstrates that the reduction in the overall routing overheads is mainly due to the reduction in node level broadcasts. There is a general reduction of 35.5 % compared with pure AODV.

(iv) Varying the Transmission rates

Again as in the case of CBR-sources figure 4.12 indicates that the reduction in overall overheads is mainly due to the link availability scheme. As high as 37% reduction is achieved by the combination scheme.

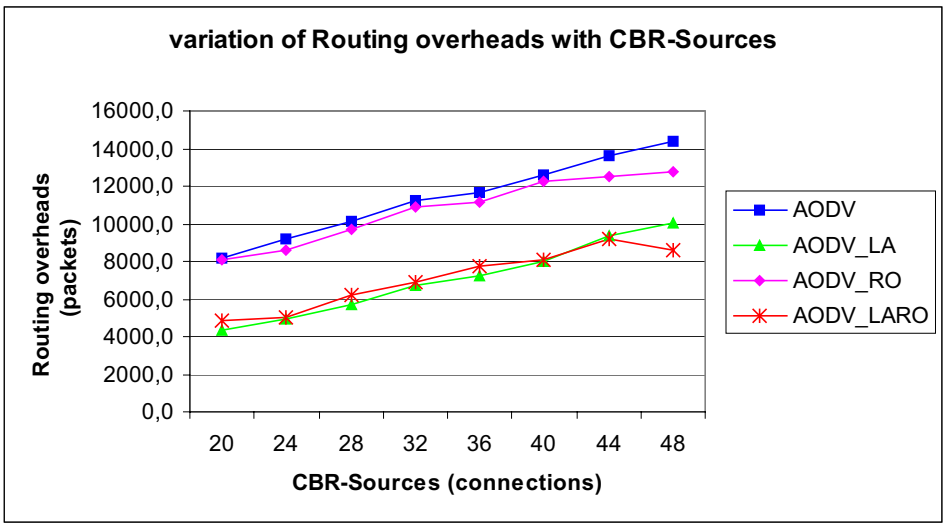
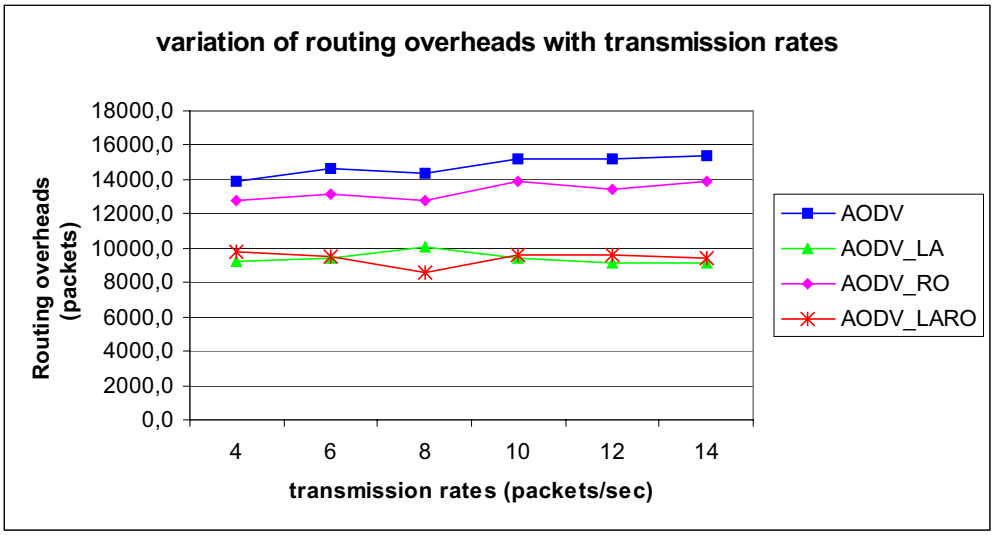


Figure 4.11 CBR-Sources against overall protocol overheads generated



4.12 Transmission rates against overall protocol overheads generated

4.3.4 Efficiency and Performance

(i) **Efficiency**

The efficiency here is determined as the degree in which the protocol is able to achieve reasonable performance with minimum consumption of available resources. The parameters considered in this case are the bandwidth use and energy consumption.

(a) Bandwidth

Bandwidth is one of the special and critical limitations encountered by reconfigurable wireless ad hoc networks (RWAdHoc). Although the availability of bandwidth depends on the medium access method in use (time sharing, frequency multiplexing code use etc) efficient use of the available bandwidth is vital. The RWAdHoc is very sensitive to the scheme used for bandwidth distribution. The most a node can do in order to determine the availability of bandwidth in its surrounding is to listen to the

medium and calculate the number of transmissions it hears per unit time. This way it can know the bandwidth use in its environs. It follows that all nodes should be made responsible for the use of the medium. Reduction of non-vital transmissions is a place to start. The suggested schemes reduce the local transmissions (hello messages) and network-wide transmissions resulting in more efficient use of bandwidth by the participating nodes. The lower the traffic around a node the higher the availability of bandwidth for information transmission.

(b) Energy use

Mobile nodes in ad hoc networks rely mainly on battery power for their operation. The energy available in such nodes is however limited and should be conserved under all costs. Packets transmissions and receipting are the main energy consumers. While computation also uses energy, the amount of energy used for computation is minimal as compared to the transmission and receiving energies.

The link availability scheme achieves a reduction in the number of neighborhood transmissions therefore saving energy otherwise used for transmission by host node and receiving energy by the neighbor for future use.

If a node transmits a packet to a neighbor and the neighbor drops the packet due to duplication, this results to wasted energy (transmission by the sender and receiving by the receiver). This is what happens in flooding schemes. There are many packets that are dropped by intermediate and destination nodes due to duplication. The network level scheme reduces the flooding of request packets by specifying the zones of transmission. The reduction of traffic in the scheme gives evidence that the scheme succeeds in avoiding transmissions to areas that do not benefit from these packets. A general network energy saving is achieved by the two schemes.

(ii) **Performance**

Table 4.1 shows the performance for two conditions of different scheme implementations. One is for the 1500 by 1000 grid with 60 nodes having 48 CBR-Sources, while the other is for 1000 by 700 grid with 30 nodes having 22 CBR-Sources. The speed used was 10 m/s for the first case and 8 m/s for the second case. Both experiments used transmission rate of 8 Packets/sec. Simulation time chosen was 60 seconds for each of the ten iterations of the first experiment and 20 seconds for each of the six trials of the second experiment. The reason for showing the two cases of the experiments is that the delivery ratio values on the first case appear too low. This is not the actual reflection as the packets counted at the receiving are the packets that make it to the destination within the 60 seconds. Other packets that don't make it in time to the destination are lost but are actually held up in the queues along their route towards the destinations. A Further experiment was done to show the contribution of each of the studied packets to the total overheads. Results of this are shown at the appendix section (Appendix II B). To indicate the improvement offered by the schemes, only the graphs for the first experiment are represented in figures 4.13, 4.14 and figure 4.15.

(a) Throughput

As the graph in figure 4.13 indicates, there was an increase in throughput for the three implementations. AODV_LA offered the best throughput due to its aggressive reduction of the overheads associated with hello messages at the node level. This is mainly associated with the savings in the bandwidth thus more packets could be delivered per unit time.

Protocol	Throughput (kbits/sec)	Del.Ratio	Delay (sec)
AODV	219,29	0,383576	0,938121
AODV_LA	224,39	0,393051	0,791259
AODV_RO	223,64	0,391523	0,933263
AODV_LARO	223,70	0,392470	0,894393

Table 4.1a Performances for 60 nodes

Protocol	Throughput (kbits/sec)	Del.Ratio	Av.Dely (sec)
AODV	242,9328	0,681470	0,527925
AODV_LA	246,3617	0,688629	0,325462
AODV_RO	250,3448	0,699750	0,563313
AODV_LARO	261,3935	0,731123	0,392693

Table 4.1b Performances for 30 nodes

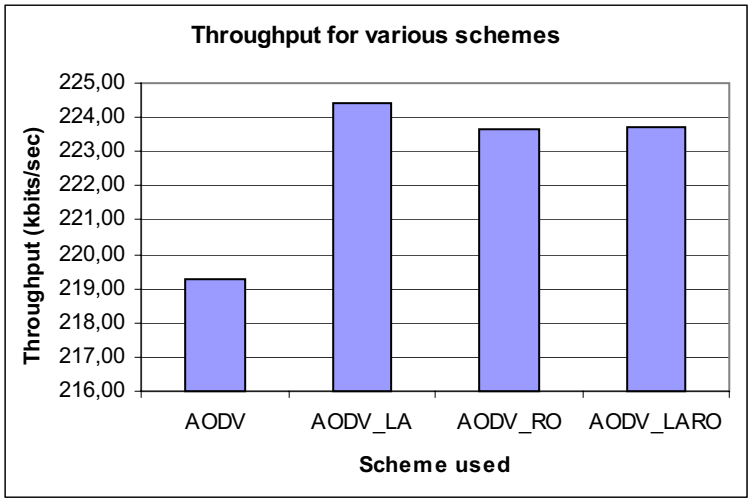


Figure 4.13 throughputs

(b) Delivery ratio

As seen in the graph of figure 4.14, the AODV_LA implementation again showed the best delivery ratio although the margin between the implementations was small. However, both link availability and network wide overhead reduction schemes showed better delivery ratio than pure AODV. The combination scheme showed an average effect that lies between the two schemes.

(c) Average delays

Similar indication to the case of delivery ratio was shown for the average delay. Figure 4.15 illustrates this trend. As expected for the network wide overhead reduction, the delay was more than for pure AODV mainly due to the computation. Since the delay taken was for the whole network, it would includes cases of the scheme failing and takes the delay as the time between the first packet sent to the predefined zone till a route is successfully found.

(iii) **Scalability**

An algorithm or scheme can be said to be scalable based on its performance versus network density (or size). For a reasonably large size of a network with certain mobility patterns and traffic loads if two protocols achieve similar throughput, delivery ratio or delay, but one displays significantly more control packets, the one with less traffic is considered to use its control packets more efficiently. However, if one of the protocols gives more improvement with increase in the network size or traffic aggression, the protocol is said to be more scalable. In these experiments, the two schemes scale better than the pure AODV for higher node densities. The network level scheme scales better at higher mobility than the node level scheme.

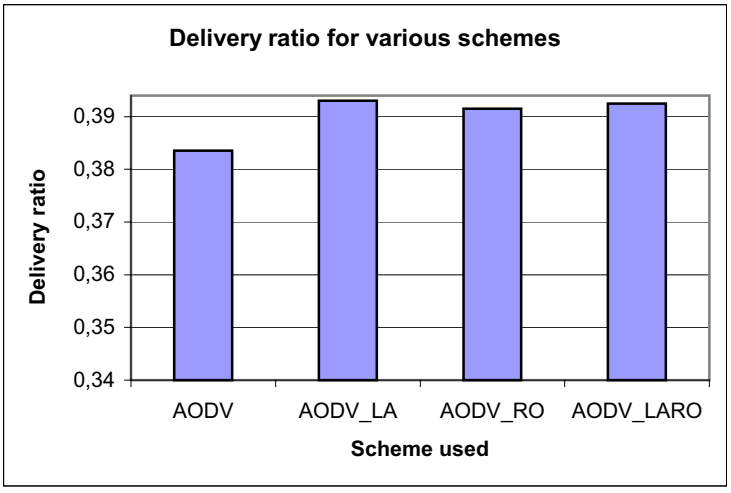


Figure 4.14 Delivery ratios

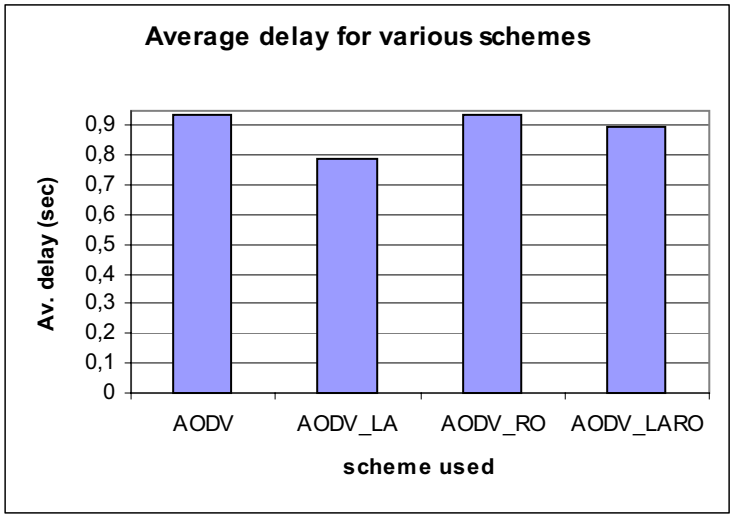


Figure 4.15 Average delays

It is therefore just to say that the two suggested schemes improve the scalability of the routing protocol. It's worth mentioning at this point that scalability depends on many other factors e.g. choice of routes, congestion parameters, network diameter, bandwidth utility, propagation model where deployment (in or out-door) and morphology are considered, mobility model among other parameters. Most of these parameters have been satisfied by the suggested scheme as the experiments carried out included medium to large size networks with different mobility patterns. The scalability of a routing protocol is normally confused with the scalability of the network. The main difference is that protocol scalability is bounded by the network scalability i.e. to say protocol scalability cannot go beyond what a network can. However, network performance can be improved by scalability of the protocol.

4.3.5 Accuracy of results

For the accuracy of results in the simulation, two factors were considered: (i), the reliability of the simulation tool and (ii), accuracy of the algorithm. For the reliability of the simulation tool, evidence was based on the wide use of the tool in the research field. The reliability can more accurately be determined by comparison of the simulation results with those of real life implementation. For the second factor, an analysis of location accuracy offered by the schemes and the statistical study of results were used.

The accuracy of location estimation by the nodes was sampled with ten sets of values taken from the link availability forecast scheme. The exact location was obtained from a tool "god", that oversees the whole simulator while estimates were obtained from the forecast results. From all the readings taken for each set, the predicted readings that coincided with the exact readings were noted as exact locations. The readings that deviated from the exact readings but within the nodes transmission range were considered as approximate location. Although there were some inaccuracies, there were no locations that were out of transmission range as table 4.2 indicates. Table 4.2 shows the iteration number for the experiment in the first column, the number of location readings taken in the second column, the fraction of all the readings that gave exact location predictions in the third column, the fraction of inaccurate but within limit (transmission range of the exact node location) while the last column shows the predicted locations that were out of nodes transmission range that could result to miss-information to the sender and subsequent loss of data.

To obtain the confidence limit of the results, one set of results was used. Results for six iterations on the routing overheads with a grid of 1500 by 1000, mobility of 10 m/s, transmission rate of 8 packets per second and 60/48 node/cbr-source values ran for 40 seconds were used. For each of the scheme, the relevant overhead parameters were used and the confidence limit obtained and tabulated as shown in table 4.3. Sample means were calculated from the samples, then standard deviations and confidence interval obtained at 95% confidence limit.

Determining the accuracy of Link forecast scheme on location estimates
scheme used: AODV_LA, grid: 1500by1000, nn/cbr-src 60/48, velocity: 10m/s, Trans Rate 8 pck/sec, Time: 60 sec

Iteration	No. of locations	fraction of exact locations	Inaccurate but within limit	Locations out of limit
1	50601	0,976917	0,023082	0,000000
2	40439	0,986078	0,013922	0,000000
3	47637	0,987279	0,012721	0,000000
4	43241	0,971092	0,028908	0,000000
5	76003	0,988619	0,011381	0,000000
6	30909	0,988482	0,011518	0,000000
7	65946	0,980469	0,019531	0,000000
8	43072	0,980312	0,019688	0,000000
9	50818	0,979082	0,020918	0,000000
10	37287	0,980342	0,019658	0,000000
Mean		0,981867	0,018133	
	Std Dev	0,005690329	0,005690259	
99 % Confidence		2,25521E-05	2,25518E-05	

Table 4.2 Accuracy of location estimates in Link availability forecast scheme

Protocol	Iteration/ parameter	1	2	3	4	5	6	mean	std dev	Cfd(95%)
AODV	Overheads	14593	14335	14527	14296	14023	14667	14407	237,3	189,9
AODV_LA	Hello msgs	597	550	597	606	563	577	581,7	22,1	17,7
AODV_RO	Requests	6978	6672	7352	6586	6353	8170	7019	661,6	529,4
AODV_LARO	Tot Ovhds	9711	8158	8734	8011	7216	9798	8605	1014,4	811,7

Table 4.3 Confidence limits approximation for routing overheads generated

A similar procedure was repeated for the performance parameters and the confidence obtained for pure AODV and the two schemes' implementations. In this case ten iterations were used. Table 4.4 shows a summary of results obtained for the performances. The table shows mean values taken for ten samples, their standard deviations and the confidence at 95% limit. The degree of freedom used was n-1=9.
In the three tables, the formulae used for standard deviation is:

$$s = \sqrt{\frac{n\sum x^2 - \left(\sum x\right)^2}{n(n-1)}} \quad \text{..(19)}$$

where "*s*" is the standard deviation, "*x*" represents the deviation from the sample mean and "*n*" is the number of samples.

For the calculation of the confidence interval, if we assume alpha (ζ) equals 0.05, we need to calculate the area under the standard normal curve that equals (1 - alpha), or 95 percent. This value is ± 1.96. The confidence interval is therefore:

$$\bar{x} \pm 1.96\left(\frac{\omega}{\sqrt{n}}\right) \quad \text{...(21),}$$

where ω is the standard deviation and n is the number of samples.

	Scheme	Mean values	Std Dev	Confd(95%)
	aodv	219,29	22,08	13,69
	aodv_la	224,39	20,94	12,98
	aodv_ro	223,64	20,40	12,65
Throughput (kbits/sec)	aodv_laro	223,70	19,92	12,35
	aodv	0,383576	0,038708	0,023991
	aodv_la	0,393051	0,036229	0,022455
	aodv_ro	0,391523	0,036077	0,022360
Del. Ratio	aodv_laro	0,392470	0,035224	0,021831
	aodv	0,938121	0,206660	0,128087
	aodv_la	0,791259	0,104286	0,064636
	aodv_ro	0,933263	0,173785	0,107711
Av. Delay (sec)	aodv_laro	0,894393	0,124137	0,076939

Table 4.4 Confidence limits approximations for the performance metrics

4.4 Analysis of Results

These results indicated that the link availability forecast results in greater gains than the network wide scheme. The node level scheme improved the overheads due to hello messages by over 90% with all configurations tested. The improvement however increased from low node density to high node density but remained the same for different speeds and CBR-Source connections. The constant difference is because sending of hello messages is not directly dependent on the speed or rate of traffic flow. The small changes in the graphs are however as a result of higher link breakages as a result of high speeds and congestion in case of high CBR-sources resulting in possible failure to have accurate updates. The observations were as a result of the theoretical predictions that reactive scheduling of hello messages would result in more effective use of hello messages. The drastic reductions in overheads result in efficient use of resources (e.g. bandwidth and energy) and reduction of congestion around the nodes neighbourhood.

As for the network level scheme, the improvement is smaller than in the link availability forecast scheme. However there is a general reduction in the request packets of about 20 %. With increase in node density, the improvement is a result of less nodes participating in route establishment. With higher speeds of movement, there are greater chances of link breakages and route invalidation resulting in more frequent demand for route establishment. Similarly, with increase in CBR-Sources, more link breakages result in a possibility of more routes being invalidated with time. The amount of request messages is not directly dependent on the rate of transmission of information packets as figure 4.8 shows. This is because, the number of requests depend on the demand for routes and/or topological structure of the network.
Figures 4.9, 4.10 and 4.11 show a repeat of the trends of the independent node level and network level schemes whereby the two effects have been combined. The graphs indicate better efficiency as a result of the combination of the two schemes on increase in node density and average node speed. The combination scheme however performs similar to the links availability scheme implementation for changing CBR-Sources and transmission rates.

On the performance issues, there is notable improvement offered by the two schemes, in all the three metrics. Although more improvements were expected for the throughput and delivery ratio the schemes seem to favour efficiency more than performance. The reason for not achieving more performance gains is probably due to the larger packets than in the pure AODV being transmitted. This however does not compromise the performance and more so the scalability of the schemes. For the network wide protocol, stronger evidence would have been obtained if the coverage area were significantly larger than the transmission radius so that the forwarding zone becomes more relevant. Even more intensive results would also have been obtained if the simulation time used was larger (100 seconds with a about 20 iterations would have been more ideal) but this would require large volumes of hard disk space, which was not available.

The accuracy of the results shows that the implementation of the algorithm meets the expectations of the anticipated results with acceptable level of confidence. The algorithm showed over 98% accuracy in obtaining the exact locations and 1.8% inaccurate locations, which were acceptable. There was no case of locations that could lead to misleading the nodes and result to link failure. As table 4.2 indicate, the variation of the routing overheads shows a 95% confidence interval of 14407∂189.9 for aodv, 581∂17.7 for aodv_la, 7019∂529.4 for aodv_ro and 1014∂811.7 for aodv_laro from a randomly selected set of results. The confidence intervals for the throughputs are indicated in table 4.3.

4.5 Chapter Summary and Conclusions

The results of the experiments offer a justification on the proposal on the impact of the reduction of management of routing overheads. The node level scheme succeeded in reduction of the overheads due to neighbourhood broadcasts. This scheme has indicated a reduction of over 90% of the broadcasts due to hello messages. The high percentage shows that most these packets are really not useful to the functioning of the network. Only a portion of them is necessary. This is evident, as the performance is not degraded by the drastic reduction of the packets. If anything most performance metrics are improved. The results also showed that the link availability scheme reduced the effect on the hello messages generated by increase in node density. The increase in average node speed and CBR-Source connections had no effect on the amount of hello messages in pure AODV, however these has affect on the suggested scheme since high speeds would result in more frequent link changes and thus more need to know the status of neighbours while more CBR-Sources means more knowledge of neighbourhood needed. The number of hello messages was not affected by the rate of transmission in both pure and link availability implementation.

The network level scheme manages the request packets and directs them to specified zones instead of flooding them to the entire network. An overall reduction in request packets of over 20% was realised. There was a general increase in route requests in both pure AODV and network level scheme. However at higher node density, the network implementation out performed the pure AODV. A similar trend was shown by the average node speed experiment. The changes are however not as gradual. Although network level scheme performs better than AODV with increase in CBR-Sources, the difference is almost constant. There is however a break in trend at

high CBR-Sources. The two schemes behaved similarly with increase in transmission rates with network scheme generating less request messages.

Although the node level scheme showed more reduction in hello messages, its application is limited to protocols that rely on single hop broadcast for neighbourhood establishment and maintenance of local connectivity. The network level however has the advantage of wider application since most of the available protocols rely on route establishment packets for sending and receiving information packets. The graphs (figures 4.9 – 4.12) on overall routing protocol show the effects of the targeted message types on the overheads. In these graphs the link availability scheme showed more effects between the two schemes.

A combination of the two schemes has also shown better results across the board. Although the trends of the individual schemes have been repeated, the effects are less mild. This is because of the summation of the two. The combination effect is however an improvement on both schemes.

Although the simulation tool does not reflect all real world features, it gives a good impression of the functionality of the protocols and possible implementation.
The suggested schemes have been designed to adapt to most of the common issues in the reconfigurable ad hoc protocols. They can however be used on a wide range of protocols with minimum modification. As a conclusion, the two schemes have shown improvements on the overall routing overheads resulting in improved efficiency and performance of the tested routing protocols.

CHAPTER FIVE

5.0 IMPLEMENTATIONS, APPLICATION CLASSES AND AREAS

5.1 Implementations of some Ad Hoc routing Protocols

Current routing protocols have been implemented in various fields. Only the most common protocols and their implementations are mentioned below:
TORA/IMEP has found implementation at a "TORA/IMEP project" in the University of Maryland. It has been implemented on ns2 simulation tool [56] running on a Linux platform.

Other than AODV's implementation on ns simulation at the Uppsala University, AODV for IPv6 at University of California [60] and Kernel AODV at NIST in OPNET [58], [61], AODV has found implementation on the NovaRoam routers [57]. Nova claims to exceed 802.11(WiFi)'s communication range with this implementation. DSR also has ns implementations and has been implemented by the picoNet project for IPv4 on Linux 2.4 kernel [62]. DSR has also found implementations in the Monarch project at the Rice University [63], [66]. OLSR has been implemented by the hipercom (high Performance Communication) group of the French National Institute for research in Computer Science and Control (NIRA) [64]. It has also been implemented on the Widows 2000 and pocket PC at the University of Valencia, Spain [65]. ZRP that was initially developed at the University of Cornell has been implemented on Opnet and ns2 simulation tools [67]. It has also found application by the Mesh Networks group in their Mobile Broadband solution [68].

5.2 Application classes and areas

Ad hoc networks can be quite useful especially due to their ability to be instantly deployed and resilient to changes. These characteristics find beneficial applications in Virtual classrooms, Personal Area Networks (PAN), Embedded Computing Applications (ECA), Inter-Vehicle Communications (IVC) and automotive to PC interactions, Military communications and emergency search and rescue operations where communications are disabled. They also find application in instant network infrastructure to support collaborative computing in temporary or mobile environments such as meetings: where business associates share information during a meeting, lectures: where students use laptops to participate in an interactive manner and other envisioned applications. They will also be used to extend the usefulness and flexibility of the current Internet. Furthermore, ad-hoc networks have the potential to serve as a ubiquitous wireless infrastructure capable of interconnecting many thousands of devices with a wide range of capabilities and uses.

As envisioned by the IETF's MANET working group [64], ad hoc routing protocols, which interoperate with the standard Internet, could bring us closer to realizing the original goal of the Internet "an interoperable internetworking capability over a heterogeneous networking infrastructure" – quote from S. Corson [70]. This also calls for

a routing protocol that supports different types of networking interfaces. The routing protocol here proposed will find practical application in large-scale networks such as those of embedded systems, wireless sensor networks, metropolitan transport networks among others.

It is desired that routing protocols be categorised into application classes due to the fact that different protocols have different properties and are intended for different applications. If we take the routing example at hand, hierarchical shortest path routing is aimed at producing shortest routes within clusters. However, if we consider routing in thousands of power-constrained devices like sensors, the goal of a clustering algorithm is to maximise the network's overall lifetime by use of energy conserving clustering constraints. Considering yet another example where the goal is to minimise the processing time at the work nodes, we find that this can be categorised together with the one of minimizing the energy consumption and can be placed in the same class. In this section, application classes for the protocols are highlighted. The type of protocol hereby supported falls into three application classes: (i), local resource-preservation in dynamic networks as one of the goals is to minimise computation with suitable application in sensor networks, (ii), balanced communication cost versus resource consumption in dynamic scenarios aimed at conserving the bandwidth through reduction of signalling traffic between nodes while trying to achieve fast network start-up (topology discovery), and (iii), data-centric application in highly dynamic network where information dependant dissemination is a principal focus.

5.2.1 Emergency Services and Rescue operations

Till recent times ad hoc networks were mostly used for military applications. But due to their various advantages, lots of applications are soon anticipated to emerge in the commercial markets. One of these is Bluetooth technology that essentially uses the concept of personal ad hoc networks. In fact the establishment of the Internet Engineering Task Force, Mobile Ad Hoc Networks (MANET) Working group, which works on IP routing in ad hoc networks shows that there is a widespread interest about these networks in the scientific community as well as the industry.

Ad Hoc Network Applications have the capability of deployment without an existing infrastructure that makes them very suitable for application in emergency situations. These could be areas stricken by natural disaster like earthquake, collapsed building due to weaknesses or as a result of war, destroyed infrastructure due to fire or other calamities. Ad hoc networks can therefore be suitable in search and rescue operations, policing and fire fighting.

Mobile nodes will carry networking equipment in support of routing operations for the times when the Internet has not been impaired. With the techniques discussed in this thesis, emergency mobile units can greatly extend of the usefulness of the networking equipment during times of loss of infrastructure support. For instance, police squad cars and fire fighting equipment can cooperate to form an ad hoc network in places with restricted resources. The schemes would allow better use of resources.

5.2.2 Application in Sensor Networks

As one of the most convincing application of ad hoc networks, sensor networks share the similar definition, but put more stringent resource conditions at each sensor node. Researches in ad hoc networks have led effective means to implement sensor networks and allow sensor nodes to enjoy greater mobility and connectivity.

As planting of sensors in the environment become more and more common, ad hoc networking is becoming very useful. Sensors connected to simple processors can easily be added to the environment. A dynamic networking system, such as ad hoc sensor network, will allow the addition and integration of new sensors into the system. Since each sensor acts as a hub, the range of the environment that can have the sensors is tremendously increased because they do not have to be centered on some sort of central station. Along with this using energy efficient protocols will help increase the life span of the equipment in the field. Looking at one of the design challenges in wireless sensor networks, flooding is a well-accepted mechanism to route a packet to all nodes in the network. When the flood message is generated by a base-station, nodes can use the reverse path to return data. In densely deployed wireless networks, flooding causes massive redundancy, resulting in wasted energy and channel resources. Reduction of this overhead would go hand in hand in reducing this waste.

Following are typical examples of reconfigurable Ad hoc sensor networks:

i) Self-organizing wireless sensor networks

These may be built from sensor nodes that may spontaneously create impromptu network, assemble the network themselves, dynamically adopt to device failure and degradation, manage movement of sensor nodes and react to changes in task and network requirements.

ii) Scalable coordination of wireless robots

These form a multi-hop wireless network. They aimed for scalable, application driven wireless network service using mobile nodes.

iii) Distributed Surveillance Sensor Network

These are sensors used underwater for collecting information. They mate with underwater docking stations, which provide energy and an opportunity for mass data transfer that would be impractical with acoustical means.
Usefulness of the proposed would mainly be in terms of energy and resources savings as well as scalability improvements. The second scheme would be particularly important as would take advantage of the area of coverage reduction.

5.2.3. Commercial and civil applications

It is believed that any commercially successful network application is a potential candidate for useful deployment with the nodes that can form an ad hoc network. This may extend to mail and web server applications. Below are other possible commercial applications.

5.2.3.1 Conferencing

Mobile conferencing is a typical prototype that requires Global Systems for Mobile telecommunications (GSM), Universal Mobile Telecommunications Systems (UMTS 2) or ad hoc networks. When mobile computers gather outside the normal office environment, business infrastructure is missing while the need for collaborative computing increases. Given that most of today's projects are computerized, the need for ad hoc becomes imminent. The need for ad hoc establishment goes beyond missing infrastructure to existing Internet. This requirement arises when there is a likely requirement of overhead in the infrastructure links, for example if there is needed routing between widely separated offices environments. This type of application can also be extended to meeting rooms and virtual classrooms.

5.2.3.2 Civilian environments

As wireless computers become popular in homes, the need for ad hoc corroboration between these computers becomes clearer. Such computers are at times taken to and from offices and taken out on business trips. Such computers may not have topology related IP addresses. It would therefore be desirable to allow such computers to ad hoc in the home although the home maintains a different subnet that may be permanent. Such ad hoc would reduce user intervention for configuration and address allocation tasks. Ad hoc networks would allow reachability to all nodes at the home regardless of their usual point of attachment, which would otherwise be indicated by network prefix that is part of every IP address.

Another envisioned area of application is the Automotive and Personal Area Networks. For application in automotive we consider interior and exterior communication devices. Interior devices may include under the hood accessories like breaks, transmission, engine etc. Passenger cabin accessories like dashboard, seatbelts, airbags can also benefit from ad hoc features. Mobile devices carried by passengers like cell phones, laptops, earphones, wristwatches etc can also participate in the ad hoc scenario. External automotive communications like satellite radio, positioning information and notification of traffic information can also benefit a lot from ad hoc networks.

Miniaturization has allowed the development of many types of portable computer equipment in personal area networks. Many of these devices can take advantage of some sort of network connection, possibly a local LAN or an Internet connection. Current technology forces the portable devices to be within range of some sort of wireless hub.

This drastically reduces the range and mobility of the system. If portable devices were equipped with ad hoc structure and the density of these devices is good enough, this would allow users to have some sort of network connection at all instances.
Further application of ad hoc in civil environments includes Taxi Networks, Sports stadiums boats and small aircrafts

5.2.3.3 Embedded Computing

As mobile machines increase in today's world, the need for intelligence of these machines increases. To have such intelligence, these machines are required to process a lot of environment information that requires incorporation of high processing power into these machines. These machines would need to react to environmental changes in which they are situated and will themselves cause changes to the environment in a predicted manner. Such machines are referred to as ubiquitous computers. A first step towards realization of these ideas is found in today's laptops and PDAs, which have been loaded with wireless ports. These ports are currently being used for simple tasks like data synchronisation between machines owned by the same person.

While Ubiquitous computing creates hope for a brighter future, security conditions are becoming harder and more challenging. Security considerations must be taken into account to prevent intrusions into our privacy and protect against possibility of impersonation [1]. Ubiquitous environments can comfortably be handled by wireless LAN computing like Bluetooth. However when we need temporary interaction with other LANs unknown to the hosts, it calls for formation of ad hoc. For example, there will reach a stage that people will carry computer devices with them that will provide them with environment information like weather, traffic information, location of entertainment sites etc without any configuration. Infrastructure elements that will make such information available by way of standard TCP/IP client-server application could operate in dual mode so that they participate in ad hoc networks as well.

5.2.4 Military Applications

Military equipment now routinely contains some sort of computer equipment. Ad hoc networking allows the military to take advantage of commonplace network technology to maintain an information network between the soldiers, vehicles, and military information head quarters. Rapid deployment of ad hoc nodes makes them particularly important for military applications.

With the assistance of ad hoc networks, Unmanned Ground Vehicles (UGV) can perform scout/reconnaissance missions prior to main body movement, breach and/or clear hazardous areas and facilitate communication of main body by serving as relay stations (form an Ad hoc network). These vehicles can also work as relay stations whereby manned or unmanned vehicles collaborate to form an Ad hoc Network**,** occupy key terrain with optimal transmission characteristics, move to successive locations as main body movement progresses and provide robust and reliable information connectivity to main body elements.

5.2.5 Other envisioned application areas

It is envisioned that once ad hoc networks become conveniently available though deployment of dynamic routing protocols, application will extend to localised areas to wide areas like college campuses whereby students and faculty members communicate in an ad hoc manner. This will include messaging and browsing of information in the University library or the Internet.

Hospital application is another area in focus. Doctors and nurses may need to have access to administrative infrastructure even when not within the hospital. Similarly visiting paramedics will need to confer with patients at their residential areas or at sites of accidents and transfer data to and from the hospitals.

With presence of ad hoc technology, when people are within reach of each other, local communications will only depend on local communication channels and transmission technologies and protocols. Finally the rise of wireless cellular telephony, combined with ad hoc networks, could provide a new paradigm of public use of the public airwaves.

5.3 Chapter Summary

In this chapter, a number of existing and possible applications were mentioned. There were grouped into application classes and areas. The classes discussed are the emergency services and rescue operations, ad hoc sensor networks, commercial applications, military applications and other envisioned application areas. The characteristics of the proposed schemes particularly reduction of overheads has desired benefits to all the application areas discussed.

CHAPTER SIX

6.0 CONCLUSIONS AND FUTURE WORK

6.1 Summary and accomplishment

This dissertation covers a research carried out in the field of ad hoc networks. It gives the milestones in the introduction chapter where a brief introduction of the topic area is given, the problem statement and motivation states and boundaries (scope of coverage) of the research detailed. In the first part of the Introduction, the broad area of reconfigurable wireless ad hoc networks is described, narrowing down to the main research topic, which is the management of routing overheads in ad hoc routing protocols. In the scope subsection, the specific types of overheads dealt with in this dissertation are mentioned and the limitations defined according to available resources.

A general overview of the research area is outlined, whereby the area of routing in the ad hoc networks is given. The general characteristics and features of reconfigurable ad hoc routing protocols are outlined including the analysis if the different classifications. Detailed pros and cons of each of the mentioned class are also given.

The routing overhead management, which is the area of research is discussed, giving details of the some types of overheads available and their effects on the functioning of the network. A comparison of performance of some common routing protocols through simulations is shown, pointing out the areas that need attention. Experiment results highlight the effects of routing overheads in these protocols.

A detailed description of the two derived schemes for managing the routing overheads is given whereby weaknesses in the current protocols are exposed. The two schemes are the node level routing overheads scheme referred to as “link availability forecast” and the network level scheme referred to as the “location guided or destination search” overhead reduction scheme.

The link availability forecast scheme targets the node level single hop broadcasts. Although the scheme can be used on any single hope broadcast, this dissertation uses the hello messages as an example of such packets. The scheme makes use of history of node movement to predict its future location and based on this makes intelligent decision on when it is necessary to broadcast the single hop messages. The accomplishment made by this scheme is that the messages that are traditionally sent purely proactively are sent reactively when there is a risk of link breakage and periodically when there is no activity for a long time interval. The scheme results in a reduction in the overheads due to the node level broadcasts (hello messages). This also saves the node from sending packets that are not required by the neighbors and thus save both the broadcasting node and the receiving energy and bandwidth for around the node.

The network level overhead management scheme however targets the route request broadcasts. These are the broadcasts that are normally flooded throughout the

network by every node in the network that requires a route to a destination. This scheme also uses the location information of a previously known node to predict its location in future. It uses this information to define a possible zone of existence of a route to the destination and uses it as a search region for a route to the destination. The scheme accomplishes reductions in the number of overall route packets broadcasted in the network since the packets are sent only to prescribed zones.

The schemes described in chapter three are validated in chapter four through simulations. The tool used for the simulation is the ns2 simulation tool. The two schemes are implemented on a selected common routing protocol and evaluation done. Different parameters that affect changes in the amounts of overheads and performances are used on the two schemes' implementations. Scenarios such are changing node mobility, changing CBR-Sources, changing transmission rates and changing the node density are used for the evaluation. The amounts of hello messages in case of link availability forecast and total route requests messages are noted with changes in the parameters mentioned above. Graphs and tables for the overheads changes and performances for various implementations are obtained and explanations for certain behaviors explained. A combination effect of the two schemes in one protocol is also tested and the results analyzed. The two schemes accomplished the anticipated objectives or overhead management and increased efficiency and performance of the routing protocol used.

Various implementations and application areas of the ad hoc networks where the scheme finds suitability are explained. In this area, details of the emergency and rescue operations, the application in sensor networks, commercial applications military applications and other envisioned application areas are given.

6.2 Conclusions

In this dissertation, it has been identified that while routing overheads are important for the operation of a reconfigurable wireless ad hoc network, they at times cause undesired congestions, uneconomical bandwidth utility and energy use. This particularly happens when the overheads are sent to areas that don't benefit directly from them. The method of sending the packets and the mode of broadcasting the packets has been analyzed and changes that would make the overheads more useful effected. The overheads has been handled both at node and network levels whereby the node level broadcasts have been broadcasted in a semi- reactive and proactive manner rather than the traditional periodic while the network broadcasts have been forwarded to predetermined zones rather than the traditional flooding. In this way, there has been savings in bandwidth around the participating nodes, savings in energy in the nodes, general improvement in scalability and efficiency while performance has been enhanced.

The node level scheme hereby derived is referred to as the link availability forecast since it utilizes the features of prediction of availability of a link between communicating nodes in the future according to their movement history.

The network level scheme is also based on node mobility. In this scheme the location of a destination node at a later time is predicted and based on this prediction, a source node computes the zone between itself and the destination where a path is most likely to be. This scheme however takes over after an initial establishment of a destinations location using the flooding method or other convenient scheme. In case the scheme fails, the protocol adjusts itself to the normal flooding.

Simulation results indicate that the two schemes actually leads to a general reduction in the traffic overheads due to the reduction/better management of the two types of overheads analyzed. There are other types of overheads that contribute to the routing overheads. This research is confined to the overheads due to hello messages (neighborhood establishment messages) and the route request messages. This research however leads to the conclusion that better management of routing overheads would lead to savings in bandwidth and energy, better efficiency, higher scalability and would enhance the performance of both the routing protocol and the network. It therefore opens and stresses the importance of further analyses of the overheads as a future work topic.

6.3 Future Work

As mentioned in the conclusions sub section, a further investigation on the other types of overheads for further improvement of the protocol and network characteristics would be important. Other issues that also need attention are the simulation tool improvement that is going on in the research area so as to reflect more of the real world issues. While this is a rather new research area, a lot of improvements have however been made and the results are very promising.

REERENCES

[1]. A new Protocol for the Reconfigurable Wireless Networks (RWN) by Zygmunt J. Haas of Cornell University. 1999.
[2]. Some Computer Science Issues in Ubiquitous Computing by Mark Weiser. March 1993
[3]. Bandwidth efficient Multicasting Routing for Multi-hope Ad Hoc Networks by Tomochika O of Hitachi ltd, Jaime B. K. of California State Univ. and Tatsuya S of Univ. of California. 2000.
[4]. Link Availability models for mobile ad-hoc networks. Technical report TR99-07 by A.B. McDonald and T. Znati of University of Pittsburgh. May 1999
[5]. A Path Availability Model for Wireless Ad-Hoc Networks by A. Bruce McDonald and Taieb Znati of University of Pittsburgh. September 1999
[6]. Location Aided Routing Protocol (LAR) in Mobile Ad Hoc Networks by Young-Bae K. and Nitin H. V of Texas A &M University
[7]. Greedy Parameter Stateless Routing Protocol (GPSR) by Brad Karp and H. T. Kung of Harvard University
[8]. Optimised Link State Routing Protocol (OLSR). Internet Draft by Thomas C., Phillipe J., Anis of Rocquen court, France. October 2001.
[9]. Ad Hoc Routing – an Overview by Hanna- Maijia of Helsinki University of Tech. May 2000.
[10]. Ad Hoc Networks and Routing Protocols: An Overview by Aaron Fabbri. May 2000.
[11]. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks by Elizabeth M. Royer Univ. of California and Chai-Keong of Georgia Inst. Of Tech. April 1999.
[12]. Classification of Ad Hoc Routing Protocols by Petteri K. of Finnish Defense forces, Navel Academy
[13]. Scalable routing protocols for Mobile Ad Hoc Networks by Xiaoyan H., Kaixin Xu, and Mario Gerla of University of California
[14]. A Taxonomy of Routing Protocols in Mobile Ad Hoc Networks by Laura M. F. of Swedish Institute of Computer science 1999.
[15]. Routing Techniques in Wireless Ad Hoc Networks – Classification and Comparison by Xukai Z, Byrav R of University of Nebraska – Lincoln and Spyros M. of Florida Atlantic University
[16]. Destination - Sequence Distance Vector (DSDV) by Guoyou He of Networking Laboratory. Helsinki University of Technology. 1999.
[17]. Global State Routing (GSR): A New Routing Scheme for Ad Hoc Networks by Tsu-Wei Chen and Mario Gerla. Proc. IEEE ICC'98
[18]. Link State Routing Protocols (LSP) by Dr. Rockey K. C. Chang. Nov 2002
[19]. A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks by Mario Jao-Ng, student Member, IEEE and I-Tai, Senior Member. IEEE Journal. August 1999.
[20]. Fisheye State Routing Protocol (FSR)
[21]. Routing in Cluster Multi-hop, Mobile Wireless Networks with fading channel (CGSR) by C. C. Chiang, Proc. IEEE SICON'97. April 1997.

[22]. An Efficient Routing Protocol for Wireless Networks (WRP) by S. Murthy and J. J. J. Garcia-Lana-Aceves, ACM Mobile Networks and Applications. Oct. 1996.
[23]. Temporally Ordered Routing Algorithm (TORA) IETF MANET Internet Draft by Vincent D. Park and M, Scott Corson. Flarion Technologies – NJ. July 2001.
[24]. Signal Stability Based Routing (SSR) for Ad Hoc Mobile networks by Rohit Dube, Cynthia D. Rais Kuang-Yeh Wang and Satish K. Tripathi. Mobile Computing and Multimedia Laboratory, Dept. of Comp. Sci., Univ. of Maryland, MD. December 1996.
[25]. On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks (ODMR) by Sung-Ju lee, William Su, Mario Gerla of University of California, Los Angeles. 2001.
[26]. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks by David B. Johnson, David A. Maltz and John Broch. Dept. of Comp. Science Carnegie Mellon University, Pittsburgh, PA. 2000.
[27]. Ad Hoc On-Demand Distance Vector (AODV) Routing by Charles E. Perkins, Elizabeth M.Royer, Samir R. Das., draft-ietf-manet-aodv-11.txt. June 2002
[28]. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing draft-guerrero-manet-saodv-00.txt by Manel Guerrero Zapata of Nokia Research Center. August 2001
[29]. An Adaptive Distance Vector (ADV) Routing Algorithm for Mobile Ad Hoc Networks by Rajendra V. Boppana Univ of Texas and Satyadeva P Konduru of Nexsi Coro SanJose CA.
[30]. An Internet MANET Encapsulation Protocol (IMEP) Specification draft-ietf-manet-imep-spec-01.txt by M. S. Corson, UMD, S. Papademetriou, UMD, V. Park, NRL, and A. Qayyum, INRIA. August 7, 1999
[31]. Zone Based Hierarchical Link State Routing (ZHLS) by Mario Joa-Ng. Ph D thesis Polytechnic University, NY. June 1999.
[32]. Distributed Dynamic Routing (DDR) Algorithm for Mobile Ad Hoc Networks by Navid Nikaein, Houda Labiod and Christian Bonnet of Institute Eurecom. 2000.
[33]. The Zone Routing Protocol (ZRP) for Ad Hoc Networks <draft-ietf-manet-zone-zrp-02.txt> by Zygmut J. Haas and Marc R. Pearlman of Cornell University. June 1999.
[34]. Intrazone Routing Proitocol (IARP) for Ad Hoc Networks <draft-ietf-manet-zone-iarp-01.txt> by Zygmunt J. Haas, Marc R. Pearlman and Prince Samar of Cornell University. June 2001
[35]. Interzone Routing Protocol (IERP) for Ad Hoc Networks <draft-ietf-manet-zone-ierp-01.txt>. June 2001.
[36]. Bordercast Resolution Protocol (BRP) for Ad Hoc Networks <draft-ietf-manet-zone.txt> by Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar of Carnell University. June 2001.
[37]. Security Considerations of Ad Hoc Routing Protocols
[38]. Some Challenges and Design Choices in Ad Hoc communications By Zygmunt J. H. of Cornell University and Siamak Tabrizi of Air Force Research Laboratory, Rome. 1999.
[39] Packet switching in radio channels Part II IEEE Trans comm. Vol COM-23 by F. A. Tobagi and L. Kleinrock . 1975
[40] Medium Access Control in a Network of Ad Hoc Mobile Nodes with Heterogeneous Power Capabilities by Neeraj P. Srikanth V.K., and Son D of University of California.

[41] Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks by Ken Tang and Mario Gerla of the wireless Adaptive Mobility Laboratory, University of California, Los Angeles. October 1999.
[42] "MACAW": Media Access Protocol for wireless LANs, SIGCOMM by V. Bharghavan, A. Demers, S. Shenker and L Zhang. 1994
[43] Floor Acquisition Multiple Access (FAMA) for Packet Radio Networks, in SIGCOMM'95 ACM by C. L. Fullmer and F. A. Tobagi. 1995
[44] Dual Busy Tone Multiple Access (DBTMA) – "Performance evaluation" in IEEE VTC'99, Hauston TX, by Z. J. Haas and J. Deng. May 1999.
[45]. Energy efficient Communication Protocol for Wireless Micro-sensor Networks by Wendi R. H., Anantha C., and Hari B. of Massachusetts Inst. Of Tech. Jan 2000.
[46]. Wireless Distributed Sensor Networks for In-Situ Exploration of Mars by Craig Ulmer, Sudhakar Y. of Georgia Inst. Of Tech. And Leon A. of California Inst. of Tech. 2000.
[47]. Distributed sensor processing over an Ad Hoc wireless network: Simulation framework and performance criteria by Robert E. Van & Leonard E of wireless Communications technologies group NISST Maryland. 2001.
[48]. Timeliness and Reliability of Large-Scale Networks: A Dynamically forming, Self-Organizing Hierarchy by Daniel Mosse of Univ. of Pittsburgh. 2000.
[49]. Lucent's WaveLAN
[50]. Measurements and analysis of the error characteristics of an in building wireless network by David Eckhardt and Peter Steenkiste. ACM SIGCOMM. Oct '96
[51]. Performance Comparison of two on-demand protocols for Ad Hoc Networks by Samir R. D., Charles E. P. of University of Texas and Elizabeth M. R. of University of California. 2000.
[52]. Information Centric, Auto-configuration Addressing and Routing Protocols for Large-Scale Networks by Mario Jao-Ng Mathew Cheng and others of Applied research, Telcordia Technologies Inc. 2001.
[53]. Ad Hoc Networking by Charles E. Perkins. 2001
[54]. The CMU Monarch Project's wireless and Mobility Extensions to ns by David Johnson, Josh Broch, Yih-Chun Hu, Jorjeta Jetcheva and David A. Maltzu of Carnegie Mellon University. http://www.monarch.cs.cmu.edu/.
[55] GPS-free positioning in mobile ad hoc networks by Srdjan Capkun, Maher Hamdi, Jean-Pierre Hubaux of Institute for computer science (ICA), Communications Systems Department (CSD), Swiss Federal Institute of Technology (EPFL), Switzerland. 2001
[56] TORA/IMEP, http://www.cshcn.umd.edu/tora.shtml
[57] Embedded AODV & TORA, http://www.nova-eng.com/novaroam.html
[58] AODV-UCSB, http://moment.cs.ucsb.edu/AODV/aodv.html#Implementations
[59] AODV-UU, http://www.docs.uu.se/docs/research/projects/scanet/aodv/
[60] HUT AODV for IPv6, http://www.tml.hut.fi/~ajtuomin/manet/aodv/
[61] AODV Kernel, http://w3.antd.nist.gov/wctg/aodv_kernel/
[62] picoNet - DSR implementation, http://piconet.sourceforge.net/
[63] Monarch implementation of DSR, http://www.monarch.cs.rice.edu/dsr-impl.html
[64] INRIA implementation of OLSR, http://hipercom.inria.fr/olsr
[65] UPV implementation of OLSR for Windows NT/CE, http://reptar.grc.upv.es/~calafate/olsr/olsr.htm

[66] NRL modifications to INRIA OLSRv3, http://pf.itd.nrl.navy.mil/projects/olsrv3/
[67] ZRP implementation, http://wnl.ece.cornell.edu/wnlprojects.html
[68] ZRP's implementation in mesh networks, http://meshnetworks.com
[69] Mobile Ad Hoc Networks (manets) http://www.itef.org, *ftp://ftp.isi.edu/in-notes/rfc2501.txt*
[70] Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations by S. Corson [*University of Maryland*], J. Macker [*Naval Research Laboratory*] January 1999

APPENDIX I

ABBREVIATIONS

- ADV – Adaptive Distance Vector
- ALOHA – This is a packet-based radio media access protocol using a "*transmit at will*" method developed by the University of Hawaii. It means *Hello* or *Welcome* in Hawaii language.
- AODV - Ad hoc On-demand Distance Vector
- AODV_HLA - Ad Hoc on-Demand Distance Vector with Hybrid Link Availability forecast
- AODV_LA – Ad Hoc on-Demand Distance Vector with Link Availability forecast
- AODV_RO - Ad Hoc on-Demand Distance Vector with network level Routing Overhead management
- AODV_LARO. - Ad Hoc on-Demand Distance Vector with a combination od link availability and network level overhead management schemes
- BRP – Bordercast Routing Protocol
- CDMA – Collision Detection Multiple Access
- CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance
- CGSR – Cluster Head Gateway Switch Routing Protocol
- DBTMA – Dual Busy Tone Multiple Access
- DDR – Distributed Dynamic Routing
- DSDV – Destination Sequence Distance Routing
- DSR – Dynamic Source Routing
- GSR – Global State Routing
- FAMA – Floor Acquisition Multiple Access
- IARP – Intra Zone Routing Protocol
- IERP – Inter Zone Routing Protocol
- IMEP - Internet Manet Encapsulation Protocol
- LAN – Local Area Network
- NDP – Neighbor Discovery Protocol
- NIST – National Institute of Standard and Technology
- NS – Network Simulator
- MAC – Media Access
- MACAW – Media Access Protocol for Wireless LANs
- ODMR – On Demand Multicast Routing Protocol
- OTcl – Object oriented Tcl language
- PAN – Personal Area Networks
- RTS/CTS – Request To Send / Clear To Send
- RWN – Reconfigurable Wireless Networks
- SSR – State Stability Routing Protocol
- TDMA – Time Division Multiple Access
- TORA – Temporally ordered Routing Algorithm
- WEP – Wired Equivalent Privacy
- WiFi – Wireless Fidelity

- WRP – Wireless Routing Protocol
- ZHLS – Zone Hierarchical Links State
- ZRP – Zone Routing Protocol
- ZRP_HLA – Zone Routing Protocol with Hybrid Link Availability forecast
- ZRP_LA – Zone Routing Protocol with Link Availability forecast

APPENDIX II

SIMULATION TOOL

Simulation Platform

To build the ns simulator, one needs a computer and a C++ compiler. The simulator was developed on several kinds of Unix (FreeBSD, Linux, SunOS, Solaris). It therefore installs smoothly on Unix based OS. It should however run on a Posix-like computer, possibly with some small modifications. Ns also builds and run under Windows although not as stable as in the Unix systems.
Different Linux freewares were installed on different machines. The one found most convenient for starting off was SuSE 7.3, which was downloaded from a German mirror site and later upgraded to 8.0. Simple scenarios should run on any reasonable machine, but very large scenarios benefit from large amounts of memory. Ns is fairly large. The allinone packages require about 250MB of disk space to build. However building from pieces can save some disk space. This is the best installation if the tool is to be used by multiple people.

The NS-2

Network Simulator 2 is an object oriented, discrete event driven network simulator which is a result of on going effort of research and development basically administered by researchers at UC Berkeley. It is targeted at network research that provides substantial support for simulation of both wired and recently wireless and ad hoc networking protocols. It is a free ware simulation tool that is available both by HTTP (at http://www.isi.edu/nsnam/dist/), by anonymous FTP (at ftp://ftp.isi.edu/nsnam) and by anonymous CVS. The ftp part of ns is mirrored in Europe at ftp://ftp.ee.surrey.ac.uk/pub/mirrors/ftp.isi.edu/dist.
The simulator allows simulation scripts (scenarios) to be written in script-like programming language, OTcl. The user can therefore write OTcl scripts that define the network (number of nodes and links), the traffic in the network (source, destination, type of traffic and which protocol is to be used. More complex functionality relies on C++ code that either comes with ns-2 or supplied by the user. Although it is fairly easy to use once one gets to know the simulator, it is quite difficult for a first time user [], because there are few user friendly manuals. Even though there are a lot of documentation written by the developer which has in depth explanation of the simulator, it is written with the depth of a skilled NS user.
Most network components can be configured in detail, and models for traffic patterns and errors can be applied to a simulation to increase its reality.
Simulations in ns-2 can be logged to trace files, which include detailed information about packets in the simulation and allow for post-run Data processing. This may be calculation of network efficiency metrics like delay, throughput, etc. below is an overview of how a simulation is done.

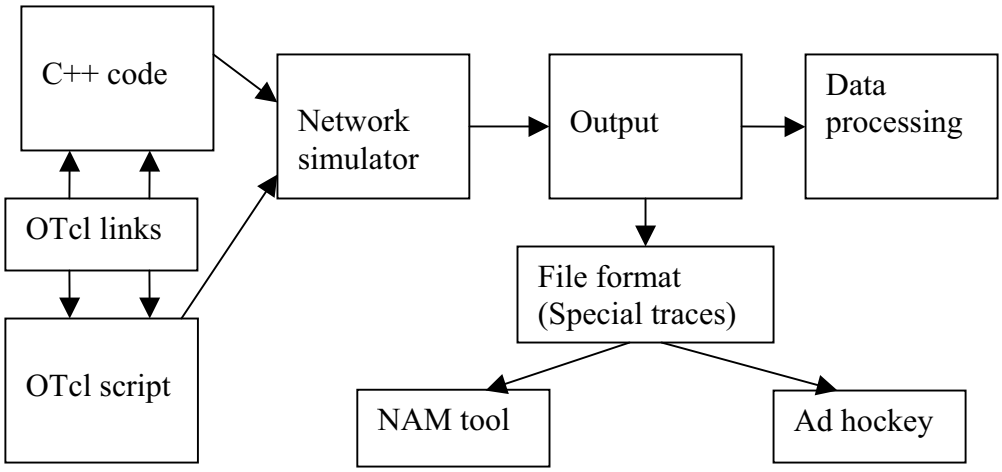


Figure ii-1 Ns-2 overview

Ns-2 also allows generation of output files that can be converted into special trace files to be used by NAM (Network Animator), a visualization tool that is part of ns-2 distribution or other visualization tools like the Ad hockey. This subchapter provides a brief overview of the ns-s simulator and illustrate use of some of it network components especially those important in porting of the AODV_LA and ZRP_LA described in the later chapters. The version described here is 2.1b9.

Installation of the Ns-2

As explained in section 3.1, the simulator can be installed as all-in-one where all the packages are installed at once or piece-wise. The all-in-one versions are the best for a user who does not need detailed knowledge of the simulation (no modifications needed to the simulator) and where memory space is not a real issue. The piece-wise installation is for deeper knowledge of what goes on in the simulator and memory saving, especially if the tool is to be used by more than one person or where huge simulations are to be expected. This is the installation that was used for the purpose of this dissertation.
To have ns-2 running, one needs at least these three pieces as minimum: Otcl, Tclcl and Ns2. To visualise the simulation, nam-1 is required. Other visualisation tools like the ad-hockey may be used.
If the network file system does not support tcl/tk 8.0 (or there is multiple tcl/tk installations that confuse the system), tcl8.0 and tk8.0 are required. If tcl/tk 8.0 are already installed, they can be used to install otcl, tclcl, ns-2 and nam.
After installation is complete, it is necessary to ascertain that the path points directly to the directories with ns and nam executables. Then, TCL and TK environment variables are set followed by configurations of the individual packages. When doing configurations certain options need to be set. These are:
--with-tcl=/path/ --with tcl-ver=8.0 –with-tk=/path/ --with-tk-ver=8.0
Our ns-2 simulator was installed from the website http://www.isi.edu/nsnam/ns/

This was a bit lengthy and time consuming process. It involved downloading and setting up a number of packages.

Understanding the Basics of Ns-2 simulations.

A tutorial for the basic ns-2 simulation is available at the website: http://www/isi.edu/nsnam/ns/tutorial/index.html. This tutorial helps in the basic understanding of how the simulator works. The tutorial is however more helpful to someone trying to simulate already code scripts. There is however no documented literature of how one would develop a new protocol on the simulator. It therefore involved some basic knowledge of the OTcl language and strong background in C++ programming language.

The OTcl/C++ environment

The C++ and OTcl programming languages allow increase in the flexibility and efficiency of the ns-2. C++ is mainly for event handling and per-packet processing; tasks for which OTcl would be too slow. OTcl us used for simpler routing protocols, general ns-2 code and simulation scenario scripts. The use of OTcl for simulation scenario scripts allows the user to change parameters of a simulation without having to recompile any source code. These two programming languages are tied together in the sense that the C++ objects can be made available to the OTcl environment (and vice versa) through an OTcl link as shown in the figure above. This link creates OTcl objects for C++ objects and allows variables of C++ objects to be shared as well. In addition it offers access to the OTcl interpreter from C++ code. Network components can therefore be configured from the simulation scenario scripts allowing transparency to the user who has some basic programming knowledge of the two programming languages.

Network Components

To help understand the basic network components we look at a partial OTcl class hierarchy of NS2. see figure 3.2 below.
The root of the directory is the TclObject class that is the super-class of all OTcl library object (scheduler, network components, timers and other objects including NAM related ones). An ancestor class of TclObject, NsObject is the superclass of all basic network component objects that handle packets, which may compose compound network objects such as nodes and inks. The basic network components are further divided into two sub-classes, connector and classifier based on the number of possible output data paths. The basic network objects that have only one output data path are under the connector class and switching objects that have multiple output data paths are under the classifier class.

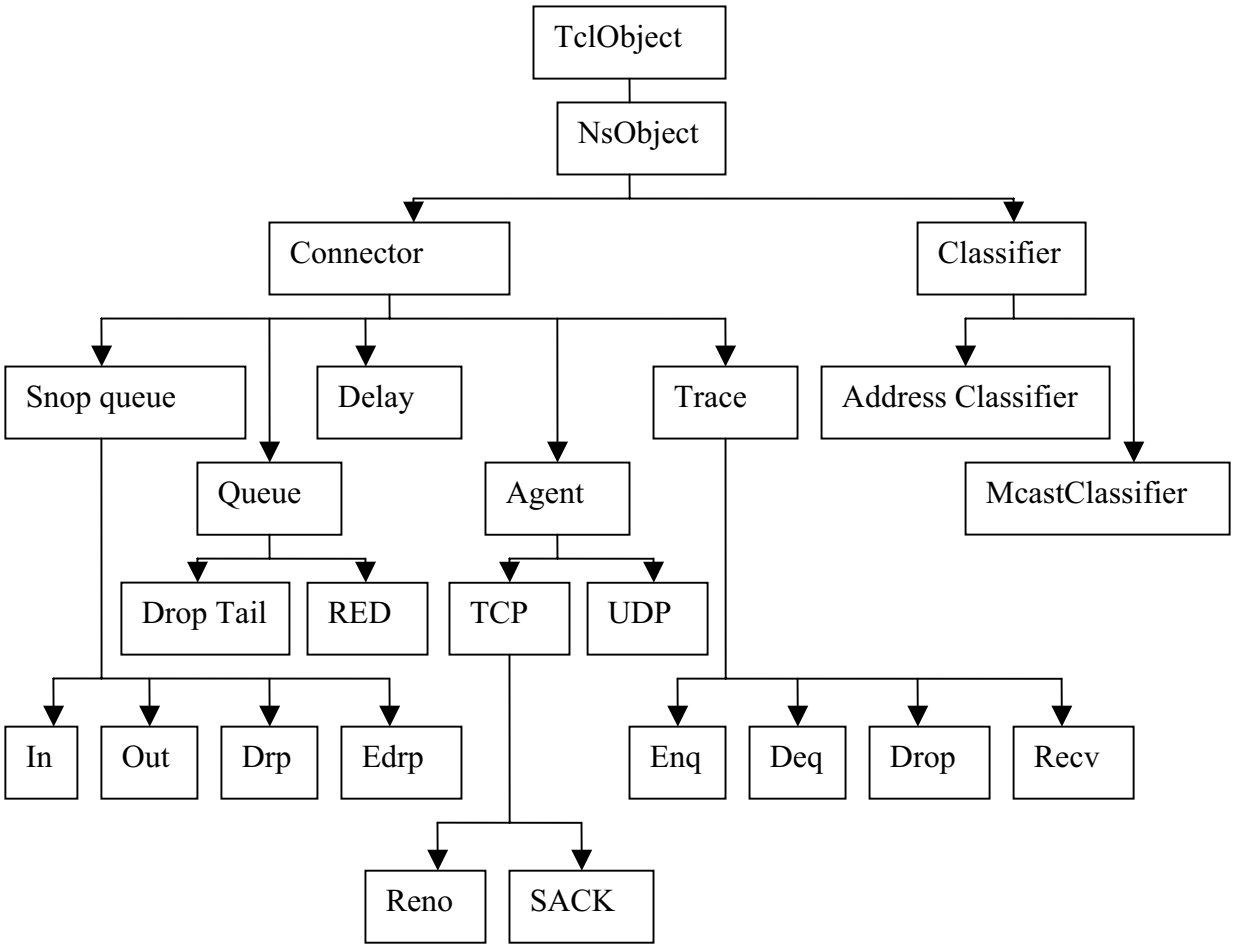


Figure ii-2 Class hierarchy of Network Components

Mobile Networking

Mobile networking in ns-2 is based on the mobility extensions developed by the CMU Monarch group. These extensions introduce the notion mobile nodes connected to wireless channels and allow for simulation of wireless network and ad hoc networks.

Mobile Nodes

A mobile node is a node with extra functionality to allow adaptation to mobile networking. An extension made in the ns-2 based on a shared media model allows all mobile nodes to have one or more network interfaces that connect to a channel. A channel represents a particular radio frequency with a particular modulation and coding scheme. Basically when a packet is sent (put in the channel), it is received (copied to all mobile nodes) connected to the same channel. This is determined by the radio propagation models, based on the transmitter range, the distance that the packet has travelled and the amount of bit errors.

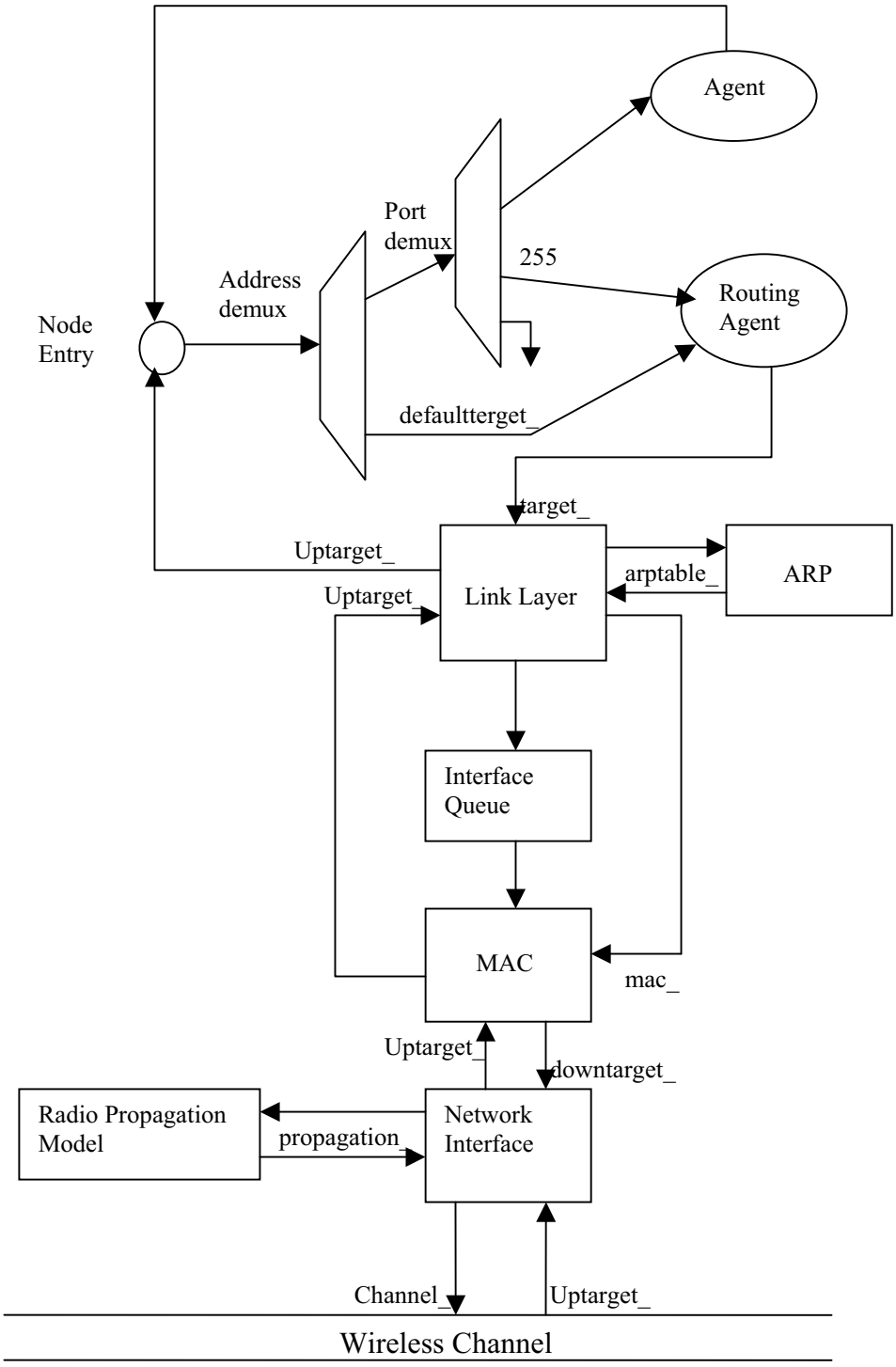


Figure ii-3 Schematics of a Mobile Node

For packets processing, each mobile node makes use of a routing agent for calculation of routes to other nodes in the network. Packets are sent from the application and are received by the routing agent. The agent decides the path that the packet must follow in order to reach the destination and stamps it with this information. It then sends this packet down to the link layer. The link layer level uses an address resolution protocol (ARP) to decide the hardware address of the neighbouring node and map ip addresses to their correct interfaces. When this information is known, the packet is sent down to the interface queue and awaits a signal from the Multiple Access Control (MAC) protocol. When the MAC layer decides that it is ok to send the packet, it fetches the packet from the interface queue and hands it over to the network interface. The network interface then hands the packet over to the Radio channel. The packet is then copied and delivered to all network interfaces at the time at which the first bit of the packet would begin arriving at the interfaces in the physical system. The Propagation model uses transmit and receive stamps to determine the power with which the interface will receive the packet. The receiving network interface then uses these properties to determine if they successfully received the packet and send is to the MAC layer if appropriate. If the MAC layer receives the packet error- and collision- free, it passes the packet to the nodes entry point. From there, it goes to the address demultiplexer which determines the destination of the packet. If the packet is addressed to this node, it is handed to the port demultiplexer, otherwise it uses the default address and sent to the routing agent for further processing. The procedure then repeats. Fig 3.2 shows a schematic view the mobile node and of how the packet is processed by the node.

Packets

Packets are the fundamental units of exchange between objects in a simulation. They are built up of packet headers, corresponding to different protocols that may be used and packet data. An ns packet is composed of stack headers, and an optional data space. See figure 3.3. A packet header format is initialised when a simulator object is created, where a stuck of all registered headers such as common header that is commonly used by any objects as needed, IP header, TCP header, RTP header (UDP uses RTP header) and trace header is defined and the offset of each header in the stuck is recorded.
Access to different packet headers and data portions of a packet is made available through access methods. New protocols may add their packet header types to the available ones and unused packet headers can be turned off to save memory during simulations. Packet delivery in ns-2 is built on the concept of network objects (components) interacting with each other. Packets are generated by agents or traffic generators and delivered through links from one node to another. In each step of the packet delivery process, a network object sends the packet using a send method, which invokes the rec (receive) method of another network object. For a packet delivery to take place, a reference to the receiving network object is needed. However all plumbing work involving network objects is performed in the simulation scenario script, and hence, all references to the network object are known.

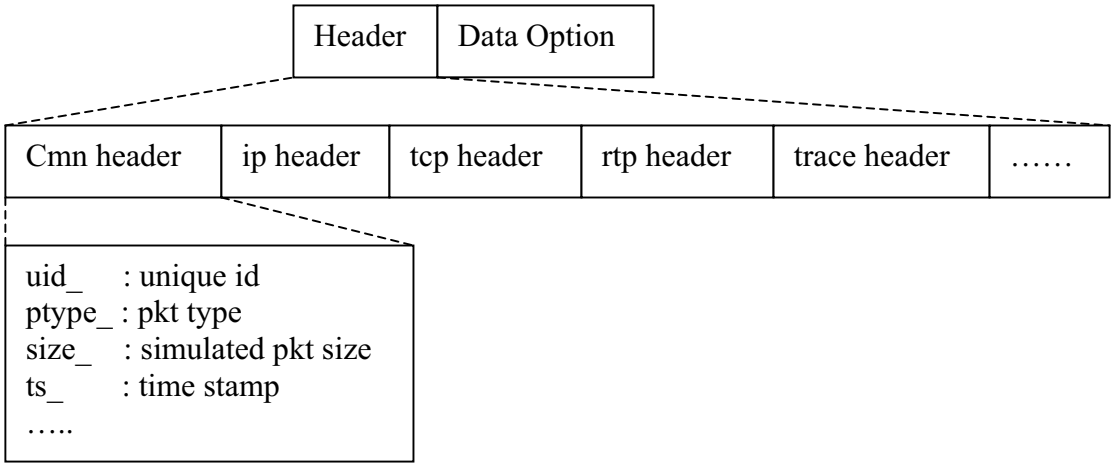


Figure ii-4 NS Packet format

Timers

Timers are used by agents and other objects in ns-2 for keeping truck of delays and performing periodic execution of the program code. They may either be implemented in OTcl or in C++, and rely on the scheduler of ns-2 for their notion of time.
Since timers are used by agents, it is very common to see constructors of timer class taking a pointer to an agent as an argument. This pointer can be stored by the timer and later dereferenced to gain access to that agent class. Similarly timer classes are often made friends of an agent class to allow them to access protected methods which otherwise could be inaccessible.

Agents

Agents represent end-points where packets are generated or consumed, and are used for implementing protocols at various layers. Routing agents and traffic source/sinks are good examples of agents that are frequently used in simulations. The OTcl class Agent and the C++ Agent class together implement agents in ns-2.
Each agent holds certain amount of information, mainly to be able to assign default values to packet fields when generating packets and to identify itself.
Agents are normally attached to nodes by installing an agent to the port classifier of the node. After attaching an agent to a node, the agent will receive all packets designated for it. The delivery of packets to agents is done by the pert demultiplexer of the node.
Routing agents are usually installed as the default target of the address demultiplexer of the node to perform forwarding of packets.
To allow agents attached to different nodes to communicate offers a command for connecting them with each other. However, the nodes to which the agents are attached need also be connected to each other e.g. using a duplex link.
When creating a new agent, e.g. a routing agent, a number of steps have to be followed in order to make the agent available to the ns-2 environment. The most fundamental steps

involves decision of the inheritance structure of the agent (this is achieved by use of the Agent class), definition of a procedure for packet reception, definition of any necessary timers for its operation, definition of OTcl links to allow agent usage from the OTcl environment of ns-2, incorporating the source code of the agent into the source code tree of ns-2 and recompilation of ns-2 to the changes to take effect.

Mobility extensions
Node Mobility

Each node is an independent entity that is responsible for computing its own position and velocity as a function of time. Nodes move around according to a movement pattern specified at the beginning of the simulation.

MAC 802.11

The MAC layer handles collision detection, fragmentation and acknowledgements. This protocol may also be used to detect transmission errors. 802.11 is a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol. It avoids collision by checking the channel before using it. It uses positive acknowledgement messages to ascertain successful delivery of packets. 802.11 also supports power saving and security. Power saving allows packets to be buffered even if the system is asleep. Security is provided by an algorithm known as Wired Environment Privacy (WEP) which supports authentication and encryption.
One of the most important features of the 802.11 is the ad-hoc mode, which allows users to build wireless LANs without an access point (infrastructure).

Radio Propagation Models

Radio propagation models are used for calculating the received signal power of packets in wireless simulations. Based on the received signal power and thresholds, different actions may be taken for a packet. The signal may be too weak or the packet may be marked as containing errors and rejected by the MAC layer. Otherwise it may have been received correctly. Three propagation models are currently being implemented by ns-2; free space model, the two-ray ground reflection model and the shadowing model. In our simulations, we use the two-ray ground reflection model. This model considers not only the line of site path between two nodes but also ground reflection. The model has shown to give accurate predictions of the received power at long distances than the free space model. It uses the formula below to calculate the received signal power at a distance *d* from the transmitter.

$$P_r(d) = (P_t G_t G_r h_t^2 h_r^2)/(d^4 L)$$

In the formula, P_t is the transmitted signal power, G_t is the antenna gain of the transmitter, G_r is the antenna gain of the receiver, *L (L Ó 1)* is the system loss, h_t is the

height if the transmitting antenna and h_r is the height of the receiving antenna. Since the model doesn't give accurate results for small d, the two-ray reflection model in ns-2 calculates a cross over distance d_c for automatically switching between free space model and two-ray ground reflection models. The formula used is shown below:

$d_c = (4\acute{} h_t h_r)/\lambda$ where λ is the wavelength.

When $d < d_c$. The free space model formula used is show below:

$P_r(d) = (P_t G_t G_r \lambda^2)/((4\acute{})^2 d^2 L)$ is used.

When $d > d_c$, the two-ray ground reflection model is used instead, since the two methods give the same results for $d=d_c$. In ns-2 simulations, it is common to select $G_t=G_r=1$ *and* $L = 1$.

APENDIX III

MORE SIMULATION RESULTS

A. 1000 by 700 grid with 20 Seconds simulation time

Reducing the Node Density with steps of 5 from 50/40 to 10/8.
The grid is: 1000 by 700 ,Velocity is: 10 m/s, transmission Rate is 8 packets/sec, Seed is: 1
and simulation time is: 20 sec

Node Density	Protocol	Routing Overheads							Performance		
		Total	Requests	Dest Search	Total Rqsts	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
10/8	AODV	491,5	130,3	0,0	130,3	339,5	20,2	1,5	94,62	0,73414	0,443296
	AODV_LA	230,8	155,0	0,0	155,0	26,7	33,3	15,8	90,61	0,708097	0,502062
	AODV_RO	508,7	109,8	32,3	142,2	340,7	21,7	4,2	90,02	0,700415	0,440766
	AODV_LARO	267,8	116,5	61,0	177,5	30,2	35,5	24,7	88,77	0,687474	0,417862
15/12	AODV	826,2	257,8	0,0	257,8	509,7	49,2	9,0	169,71	0,834089	0,247945
	AODV_LA	453,5	305,8	0,0	305,8	48,2	68,8	30,7	176,78	0,869713	0,348995
	AODV_RO	830,8	177,5	90,7	268,2	506,8	44,8	11,0	166,87	0,824934	0,262372
	AODV_LARO	518,0	188,8	150,3	339,2	56,3	69,7	52,8	169,16	0,82316	0,328719
20/16	AODV	1113,3	363,5	0,0	363,5	675,5	60,3	14,0	241,27	0,863862	0,223595
	AODV_LA	608,8	428,0	0,0	428,0	68,7	74,2	37,8	241,48	0,86536	0,271155
	AODV_RO	1190,8	231,7	186,7	418,3	675,2	68,5	28,8	233,13	0,83834	0,278766
	AODV_LARO	741,0	240,8	276,8	517,7	71,7	91,8	59,8	251,83	0,903597	0,243615
25/20	AODV	1712,3	726,8	0,0	726,8	839,7	110,8	35,0	257,41	0,792842	0,52872
	AODV_LA	1055,7	758,2	0,0	758,2	93,3	116,5	87,7	264,03	0,816049	0,47214
	AODV_RO	1676,0	436,5	253,3	689,8	838,8	101,2	46,2	256,79	0,790015	0,549042
	AODV_LARO	1182,2	457,0	405,3	862,3	92,8	137,7	89,3	267,42	0,816202	0,458429
30/24	AODV	2438,0	1211,7	0,0	1211,7	1007,7	152,8	65,8	281,76	0,70104	0,378932
	AODV_LA	1409,2	1051,2	0,0	1051,2	111,8	138,3	107,8	297,21	0,743322	0,325386
	AODV_RO	2319,3	558,3	515,0	1073,3	1008,3	138,0	99,7	289,48	0,721937	0,398451
	AODV_LARO	1499,3	532,2	538,2	1070,3	117,7	150,5	159,2	278,16	0,692076	0,343684
35/28	AODV	3322,8	1841,0	0,0	1841,0	1164,0	224,5	93,3	287,27	0,609106	0,623038
	AODV_LA	1999,3	1518,0	0,0	1518,0	151,0	195,5	134,8	292,70	0,620723	0,593737
	AODV_RO	3259,0	896,5	766,0	1662,5	1169,2	211,7	215,7	287,34	0,609854	0,614066
	AODV_LARO	2314,8	997,7	851,2	1848,8	156,7	224,0	168,7	286,68	0,60438	0,605935
40/32	AODV	4248,2	2513,8	0,0	2513,8	1333,0	255,8	145,0	300,88	0,548824	0,951843
	AODV_LA	2555,5	1981,0	0,0	1981,0	173,3	230,8	170,3	293,95	0,535821	0,910849
	AODV_RO	4101,0	1161,0	1082,2	2243,2	1328,2	255,8	273,8	289,97	0,531574	0,949361
	AODV_LARO	2858,3	1118,7	1056,7	2175,3	193,0	269,2	220,8	300,67	0,55036	0,825731
45/36	AODV	5399,8	3464,3	0,0	3464,3	1483,2	342,2	193,5	295,02	0,478639	0,935108
	AODV_LA	3334,7	2689,8	0,0	2689,8	200,3	259,7	184,8	278,75	0,45127	1,127028
	AODV_RO	5036,2	1439,0	1440,3	2879,3	1489,0	307,2	360,7	289,28	0,466374	1,055205
	AODV_LARO	3431,2	1416,5	1262,7	2679,2	224,2	317,3	210,5	292,05	0,473524	0,99281
50/40	AODV	6538,5	4303,8	0,0	4303,8	1656,7	377,5	200,5	315,74	0,466664	1,249979
	AODV_LA	3917,2	3261,0	0,0	3261,0	234,2	252,5	151,5	314,97	0,466689	1,216211
	AODV_RO	5854,3	1939,8	1549,5	3489,3	1649,3	320,5	395,2	299,67	0,443406	1,184885
	AODV_LARO	3743,3	1702,7	1294,0	2996,7	235,2	297,3	214,2	340,47	0,503174	1,103137

Changing the Velocity (mobility) with steps of 2 from 16 to 2m/s.

The grid is: 1000 by 700 ,Velocity is: Transmission Rate is 8 packets/sec, Node/cbr_source is 30/24,
Seed is: 1 and simulation time is: 20 sec

Mobility	Protocol	Routing Overheads							performance		
		Total	Requests	Dest Search	Total Rqst	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
2	AODV	2159,5	1047,7	0,0	1047,7	1011,5	88,7	12,5	275,73	0,698536	0,253935
	AODV_LA	1187,3	946,2	0,0	946,2	73,3	91,5	76,8	276,36	0,690129	0,322827
	AODV_RO	2185,8	800,2	257,2	1057,3	1010,5	89,5	28,3	279,47	0,681634	0,371092
	AODV_LARO	1320,8	791,2	283,8	1075,0	71,5	102,3	72,0	281,59	0,698026	0,299936
4	AODV	2351,0	1132,0	0,0	1132,0	1005,8	162,7	50,5	279,61	0,700329	0,401918
	AODV_LA	1412,7	1052,5	0,0	1052,5	95,0	151,7	113,5	292,84	0,732043	0,502174
	AODV_RO	2389,3	635,3	497,0	1132,3	1010,5	148,3	98,2	285,67	0,712526	0,505935
	AODV_LARO	1545,0	579,7	535,0	1114,7	101,2	179,0	150,2	298,42	0,74102	0,389522
6	AODV	2340,7	1176,7	0,0	1176,7	1009,8	115,2	39,0	277,19	0,696291	0,346626
	AODV_LA	1325,7	1024,2	0,0	1024,2	94,8	114,7	92,0	282,00	0,706092	0,200073
	AODV_RO	2271,7	692,0	377,8	1069,8	1011,2	108,7	82,0	275,49	0,687164	0,267504
	AODV_LARO	1491,5	715,2	456,7	1171,8	101,2	128,3	90,2	290,35	0,724397	0,263224
8	AODV	2438,0	1211,7	0,0	1211,7	1007,7	152,8	65,8	281,76	0,70104	0,378932
	AODV_LA	1409,2	1051,2	0,0	1051,2	111,8	138,3	107,8	297,21	0,743322	0,325386
	AODV_RO	2319,3	558,3	515,0	1073,3	1008,3	138,0	99,7	289,48	0,721937	0,398451
	AODV_LARO	1497,7	532,2	538,2	1070,3	117,7	150,5	159,2	278,16	0,692076	0,343684
10	AODV	2438,0	1211,7	0,0	1211,7	1007,7	152,8	65,8	281,76	0,70104	0,378932
	AODV_LA	1409,2	1051,2	0,0	1051,2	111,8	138,3	107,8	297,21	0,743322	0,325386
	AODV_RO	2319,3	558,3	515,0	1073,3	1008,3	138,0	99,7	289,48	0,721937	0,398451
	AODV_LARO	1499,3	532,2	538,2	1070,3	117,7	150,5	159,2	278,16	0,692076	0,343684
12	AODV	2252,7	1207,3	0,0	1207,3	1006,2	158,8	60,7	289,14	0,720518	0,32249
	AODV_LA	1501,3	1191,5	0,0	1191,5	122,8	156,0	98,3	299,84	0,751844	0,310566
	AODV_RO	2189,8	536,8	573,3	1110,2	1002,5	138,7	105,2	287,27	0,720222	0,357699
	AODV_LARO	1647,5	547,8	644,7	1192,5	136,7	178,7	139,7	298,00	0,746726	0,368566
14	AODV	2517,5	1123,5	0,0	1123,5	1001,0	157,0	69,3	275,14	0,687507	0,390447
	AODV_LA	1619,2	1212,5	0,0	1212,5	139,3	160,8	106,5	270,30	0,661436	0,370395
	AODV_RO	2508,8	569,8	619,5	1189,3	1001,5	157,0	161,0	266,21	0,662069	0,336814
	AODV_LARO	1636,7	582,3	712,2	1294,5	157,5	187,2	164,2	280,03	0,697009	0,361772
16	AODV	2511,8	1257,2	0,0	1257,2	1003,2	166,8	84,7	272,89	0,682838	0,325982
	AODV_LA	1732,2	1315,7	0,0	1315,7	155,2	166,7	94,7	280,58	0,698869	0,315844
	AODV_RO	2479,8	632,3	560,0	1192,3	1001,8	147,8	137,8	271,37	0,677203	0,404409
	AODV_LARO	1716,2	555,2	672,8	1228,0	155,3	181,7	151,2	289,45	0,720212	0,333143

Changing the Transmission Rates with steps of 2 from 2 to 12.

The grid is: 1000 by 700 ,Velocity is: 10 m/s, Node/cbr_source is 30/24, Seed is: 1 and simulation time is: 20 sec

Transm Rates	Protocol	Routing Overheads							Performance		
		Total	Requests	Dest Search	Total Rqsts	Hello	Replies	Errors	Throughput	Del Ratio	Av Delay
2	AODV	1891,3	742,0	0,0	742,0	1013,8	108,3	27,2	95,56	0,899052	0,038874
	AODV_LA	1157,3	900,0	0,0	900,0	101,0	126,7	59,7	91,65	0,860897	0,067841
	AODV_RO	1968,7	544,0	379,7	923,7	1012,8	122,2	60,0	96,49	0,907088	0,111441
	AODV_LARO	1529,7	573,8	556,3	1130,2	123,3	170,0	106,2	93,79	0,886496	0,117471
4	AODV	2206,8	1018,2	0,0	1018,2	1011,5	147,0	30,2	177,99	0,873632	0,182988
	AODV_LA	1427,3	1061,2	0,0	1061,2	128,7	147,0	90,5	171,86	0,839292	0,142963
	AODV_RO	1989,2	545,0	396,7	941,7	1015,2	118,7	80,3	175,57	0,860685	0,218805
	AODV_LARO	1697,2	559,0	682,0	1241,0	152,3	185,3	118,5	170,93	0,833721	0,154223
6	AODV	2025,8	853,2	0,0	853,2	1010,2	118,7	43,8	241,79	0,798834	0,178656
	AODV_LA	1426,5	1040,0	0,0	1040,0	112,7	147,2	126,7	244,21	0,807122	0,107615
	AODV_RO	2120,8	566,0	354,5	920,5	1008,3	119,0	73,0	245,08	0,813749	0,215350
	AODV_LARO	1473,8	593,8	491,7	1085,5	119,3	161,3	107,7	251,11	0,828602	0,237038
8	AODV	2438,0	1211,7	0,0	1211,7	1007,7	152,8	65,8	281,76	0,701040	0,378932
	AODV_LA	1409,2	1051,2	0,0	1051,2	111,8	138,3	107,8	297,21	0,743322	0,325386
	AODV_RO	2319,3	558,3	515,0	1073,3	1008,3	138,0	99,7	289,48	0,721937	0,398451
	AODV_LARO	1499,3	532,2	538,2	1070,3	117,7	150,5	159,2	278,16	0,692076	0,343684
10	AODV	2740,3	1497,7	0,0	1497,7	997,7	163,7	101,5	281,10	0,563304	0,798413
	AODV_LA	1525,3	1119,0	0,0	1119,0	123,7	152,0	152,0	273,72	0,550742	0,546015
	AODV_RO	2414,7	646,5	506,3	1152,8	1001,5	131,8	128,5	276,63	0,554870	0,671811
	AODV_LARO	1678,0	660,5	573,2	1233,7	131,7	158,7	154,0	303,27	0,608586	0,622225
12	AODV	2699,8	1446,2	0,0	1446,2	992,2	157,0	104,5	294,99	0,494428	0,903661
	AODV_LA	1518,5	1113,7	0,0	1113,7	127,2	142,5	135,2	291,63	0,489801	0,693224
	AODV_RO	2243,0	675,3	575,7	1251,0	995,8	138,5	191,0	299,63	0,501563	0,991401
	AODV_LARO	1530,5	637,3	492,7	1130,0	115,5	136,2	148,8	297,03	0,498511	0,879979
14	AODV	2550,0	1276,2	0,0	1276,2	999,2	150,7	124,0	325,12	0,469147	0,962506
	AODV_LA	1348,7	977,2	0,0	977,2	104,8	120,5	146,2	320,00	0,461295	1,025927
	AODV_RO	2323,5	599,7	443,2	1042,8	991,7	136,2	152,8	346,46	0,498856	0,927199
	AODV_LARO	1478,0	656,5	407,7	1064,2	110,5	128,3	175,0	352,73	0,508487	0,930861

Reducing the CBR sources with steps of 2 from 24 srcs down to 10.			

The grid is: 1000 by 700 , number of nodes is: 30			
Velocity is: 8 m/s, Transmission rate is: 8.0, Seed is: 1 and simulation time is: 20 sec			

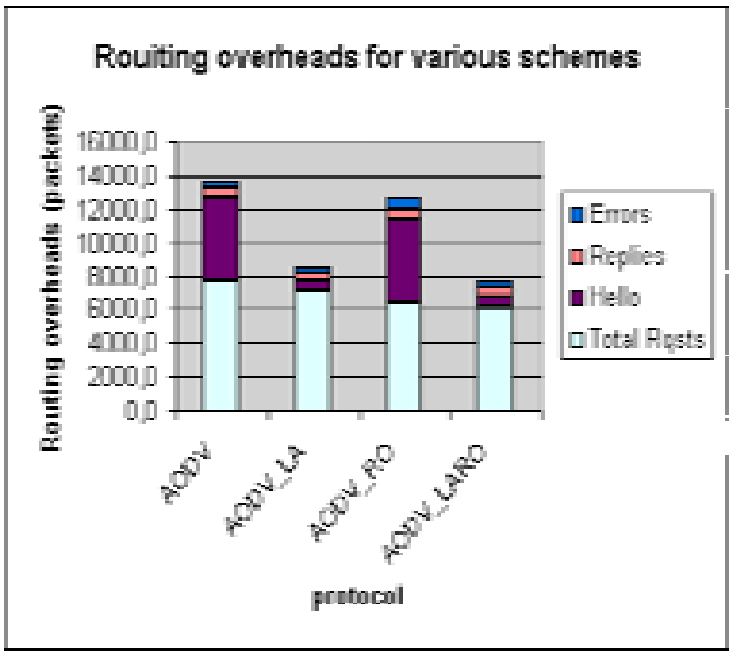
CBR Sources	Protocol	Routing Overheads							Performance		
		Total	Requests	Dest Search	Total Rqst	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
24	AODV	2438,0	1211,7	0,0	1211,7	1007,7	152,8	65,8	281,76	0,70104	0,378932
	AODV_LA	1409,2	1051,2	0,0	1051,2	111,8	138,3	107,8	297,21	0,743322	0,325386
	AODV_RO	2319,3	558,3	515,0	1073,3	1008,3	138,0	99,7	289,48	0,721937	0,398451
	AODV_LARO	1497,7	532,2	538,2	1070,3	117,7	150,5	159,2	278,16	0,692076	0,343684
22	AODV	2459,5	1241,3	0,0	1241,3	1004,8	151,5	67,8	242,93	0,68147	0,527925
	AODV_LA	1469,7	1099,7	0,0	1099,7	114,8	139,2	116,0	246,36	0,688629	0,325462
	AODV_RO	2265,3	582,2	439,7	1021,8	1003,2	122,8	117,5	250,34	0,69975	0,563313
	AODV_LARO	1494,3	531,2	537,7	1068,8	128,5	155,5	141,5	261,39	0,731123	0,392693
20	AODV	2203,3	1023,5	0,0	1023,5	1002,8	130,2	46,8	233,93	0,71918	0,372593
	AODV_LA	1504,3	1119,7	0,0	1119,7	127,0	154,0	87,3	227,83	0,699816	0,308229
	AODV_RO	2202,0	555,7	422,2	977,8	1006,5	130,7	87,0	237,60	0,729549	0,596757
	AODV_LARO	1486,5	526,7	543,3	1070,0	119,5	153,0	140,7	240,68	0,736963	0,391748
18	AODV	2180,0	984,0	0,0	984,0	1009,0	131,5	55,5	221,29	0,729895	0,486353
	AODV_LA	1330,0	973,2	0,0	973,2	123,2	139,3	94,3	229,49	0,753938	0,299807
	AODV_RO	2063,8	454,8	388,0	842,8	1006,5	123,8	90,7	217,13	0,717188	0,548942
	AODV_LARO	1357,0	464,8	498,7	963,5	123,5	143,3	126,7	224,44	0,7408	0,346988
16	AODV	1965,3	818,8	0,0	818,8	1005,7	105,8	35,0	213,60	0,770561	0,469536
	AODV_LA	1157,5	831,7	0,0	831,7	113,8	122,5	89,5	218,90	0,787273	0,217425
	AODV_RO	1870,3	390,3	306,0	696,3	1008,8	93,5	71,7	213,77	0,76583	0,436526
	AODV_LARO	1173,2	411,0	417,7	828,7	106,7	123,3	114,5	211,93	0,764074	0,29162
14	AODV	1853,8	732,7	0,0	732,7	1010,2	83,3	27,7	194,27	0,819514	0,281456
	AODV_LA	963,7	662,3	0,0	662,3	89,8	89,8	121,7	178,44	0,752397	0,192704
	AODV_RO	1890,3	389,2	323,7	712,8	1011,8	87,0	78,7	184,33	0,780595	0,44409
	AODV_LARO	1107,7	362,0	437,7	799,7	103,7	110,8	93,0	195,59	0,824474	0,305935
12	AODV	1779,0	672,5	0,0	672,5	1012,5	72,7	21,3	168,78	0,83087	0,23909
	AODV_LA	972,7	694,5	0,0	694,5	97,7	101,7	78,8	166,18	0,819597	0,147622
	AODV_RO	1696,0	306,3	255,7	562,0	1017,8	70,5	45,7	172,62	0,847618	0,33098
	AODV_LARO	891,2	291,2	326,8	618,0	94,3	91,3	87,5	168,36	0,827364	0,264674
10	AODV	1623,3	523,8	0,0	523,8	1015,7	68,0	15,8	132,52	0,851997	0,191776
	AODV_LA	824,8	580,5	0,0	580,5	97,0	88,2	59,2	137,57	0,880965	0,164329
	AODV_RO	1655,8	289,5	233,3	522,8	1014,3	70,3	48,3	129,36	0,8217	0,240368
	AODV_LARO	1011,7	299,3	400,2	699,5	97,3	123,2	91,7	135,04	0,865194	0,236521
8	AODV	1571,3	482,8	0,0	482,8	1013,2	60,3	15,0	109,00	0,852864	0,101914
	AODV_LA	743,8	554,2	0,0	554,2	76,5	75,3	32,5	117,34	0,914285	0,333772
	AODV_RO	1591,8	290,3	197,0	487,3	1014,2	59,2	31,2	104,43	0,815016	0,137352
	AODV_LARO	907,8	321,2	335,0	656,2	84,0	100,2	67,5	112,50	0,869649	0,214106

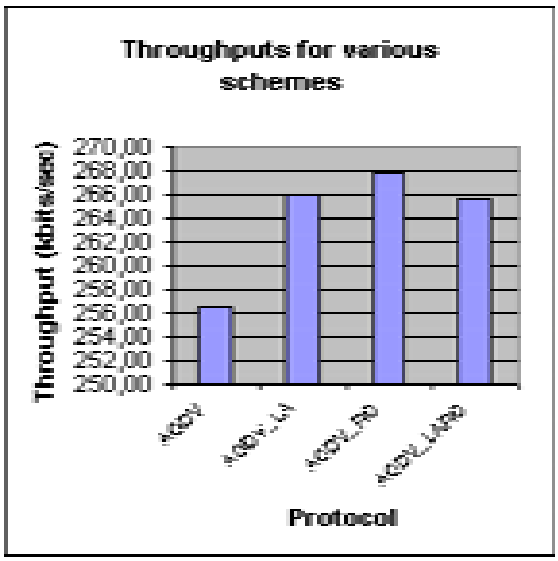
B. 1000 by 700 grid with 60 Seconds simulation time

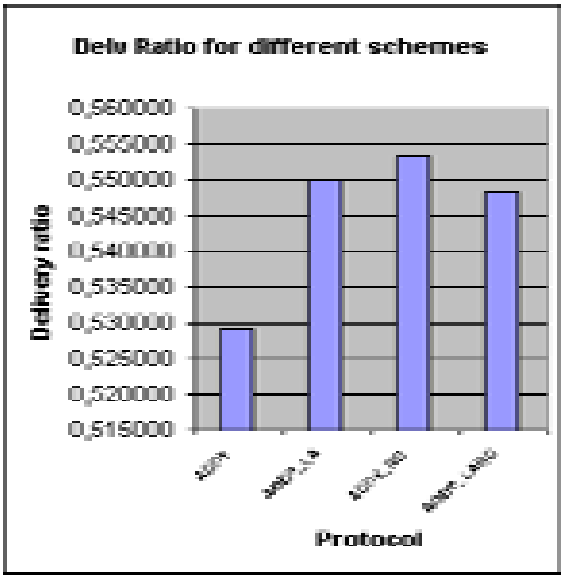
The grid is: 1000 by 700 , number of nodes is: 50 with 40 CBR-Sources

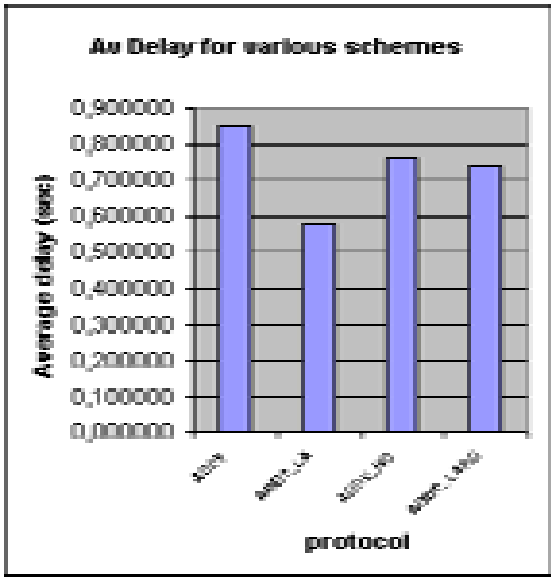
Velocity is: 8 m/s, transmission Rate is 8 packets/sec, Seed is: 1 and simulation time is: 60 sec

Protocol	Routing Overheads							Performance		
	Total	Requests	Dest Search	Total Rqsts	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
AODV	14364,9	7659,6	0,0	7659,6	4967,4	604,7	433,2	256,58	0,529266	0,852594
AODV_LA	8535,5	7052,2	0,0	7052,2	640,1	543,7	299,5	265,92	0,549876	0,581463
AODV_RO	12738,0	2303,2	4131,3	6434,5	4964,7	584,0	754,8	267,90	0,553462	0,760535
AODV_LARO	7755,6	2186,7	4021,2	6207,9	529,0	574,5	412,9	265,61	0,548441	0,742038









C. 1500 by 1000 Grid with 40 seconds simulation time

Changing the Node density from 20/16 to 80/64 nodes/cbr-sources

The grid is: 1500 by 1000 ,Velocity is: 10 m/s, Transmission rate is: 8.0, Seed is: 1 and simulation time is: 40 sec

Node Density	Protocol	Routing Overheads							Performance		
		Total	Requests	Dest Search	Total Rqsts	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
20/16	AODV	1987,0	588,8	0,0	588,8	1343,3	37,5	17,3	58,67	0,488629	0,961851
	AODV_LA	827,2	644,8	0,0	644,8	64,3	63,8	54,2	61,87	0,514950	0,948999
	AODV_RO	2223,8	673,5	115,2	788,7	1344,8	48,7	41,7	60,11	0,497029	0,660144
	AODV_LARO	1032,8	632,2	180,7	812,8	63,3	72,0	84,7	61,14	0,512388	0,836882
30/24	AODV	3685,3	1486,8	0,0	1486,8	2012,7	122,8	63,0	74,53	0,436337	1,340665
	AODV_LA	1932,7	1569,2	0,0	1569,2	121,3	149,7	92,5	74,47	0,435124	1,348929
	AODV_RO	4060,7	1405,5	383,3	1788,8	2012,2	131,8	127,8	82,70	0,486720	1,337728
	AODV_LARO	2292,0	1452,8	413,8	1866,7	130,5	160,2	134,7	81,24	0,477410	1,242045
40/32	AODV	6771,8	3510,2	0,0	3510,2	2657,2	374,8	229,7	100,78	0,420734	0,922646
	AODV_LA	4001,3	3108,5	0,0	3108,5	272,5	386,3	234,0	101,08	0,422719	0,814998
	AODV_RO	7177,8	2489,3	1185,7	3675,0	2669,7	382,5	450,7	105,37	0,441566	1,040639
	AODV_LARO	4526,7	2226,3	1281,7	3508,0	275,0	379,8	347,2	107,99	0,452139	1,048211
50/40	AODV	10209,5	5828,3	0,0	5828,3	3325,3	662,7	393,2	131,88	0,443699	0,859742
	AODV_LA	6949,7	5422,5	0,0	5422,5	432,3	654,5	441,8	130,99	0,441334	0,871337
	AODV_RO	9277,7	2572,3	2143,2	4715,5	3321,3	541,8	699,0	135,80	0,457334	0,922809
	AODV_LARO	6630,5	2712,5	2387,0	5099,5	409,2	609,0	512,8	134,81	0,455223	1,005317
60/48	AODV	14406,8	8863,3	0,0	8863,3	3971,3	972,7	599,5	135,24	0,385880	0,951475
	AODV_LA	10071,5	8001,8	0,0	8001,8	581,7	917,0	571,0	134,34	0,382775	1,051442
	AODV_RO	12744,5	3789,2	3229,3	7018,5	3971,8	787,7	966,5	147,52	0,420766	0,963162
	AODV_LARO	8604,7	3414,2	3235,8	6650,0	546,5	794,2	614,0	144,97	0,413903	0,992282
70/56	AODV	18750,8	12113,5	0,0	12113,5	4626,3	1137,2	762,5	153,65	0,379678	1,098916
	AODV_LA	12754,7	10535,0	0,0	10535,0	665,7	939,8	619,7	155,28	0,383646	0,958100
	AODV_RO	16207,3	4669,3	4504,5	9173,8	4647,5	950,8	1435,2	161,69	0,399583	1,104973
	AODV_LARO	11960,3	4794,0	4614,3	9408,3	737,5	1006,8	807,7	160,83	0,397297	1,084255
80/64	AODV	23358,0	15580,8	0,0	15580,8	5276,2	1445,8	1055,2	135,42	0,305478	1,156424
	AODV_LA	16480,5	13581,2	0,0	13581,2	883,0	1251,8	764,5	143,09	0,322446	1,147599
	AODV_RO	20016,5	6255,2	5257,3	11512,5	5301,0	1157,2	2045,8	148,78	0,335377	1,173719
	AODV_LARO	14346,0	5657,7	5602,7	11260,3	898,3	1207,0	980,3	147,42	0,332685	1,127147

Changing the mobility from 2 to 16 m/s

Reducing the Mobility with steps of 2 from 16 to 2.

The grid is: 1500 by 1000 , number of nodes is: 60 with 48 CBR_Sources

Transmission Rate is 8 packets/sec, Seed is: 1 and simulation time is: 40 sec

Mobility	Protocol	Routing Overheads							Performance		
		Total	Requests	Dest Search	Total Rqsts	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
2	AODV	11415,3	6363,7	0,0	6363,7	4007,2	725,2	319,3	163,20	0,465525	0,914505
	AODV_LA	7113,8	5843,7	0,0	5843,7	255,5	668,7	346,0	158,62	0,453367	0,848159
	AODV_RO	11497,5	4400,2	1892,7	6292,8	4011,5	619,3	573,8	170,82	0,487445	1,213639
	AODV_LARO	6931,7	3942,3	1715,8	5658,2	277,3	614,0	398,8	158,34	0,451419	1,053840
4	AODV	12721,2	7432,0	0,0	7432,0	3994,8	883,8	410,5	139,81	0,399395	1,224410
	AODV_LA	8452,5	6850,7	0,0	6850,7	366,7	823,5	411,7	134,27	0,382906	1,267204
	AODV_RO	12119,0	4246,3	2433,3	6679,7	3996,2	700,7	742,5	146,15	0,417292	1,291738
	AODV_LARO	7545,2	3695,5	2338,7	6034,2	342,3	685,0	483,7	146,22	0,417702	1,205699
6	AODV	12812,7	7631,8	0,0	7631,8	3995,0	789,8	396,0	143,68	0,409253	1,080701
	AODV_LA	8764,2	7139,8	0,0	7139,8	401,8	790,0	432,5	139,63	0,398808	1,065989
	AODV_RO	11979,8	4156,3	2483,2	6639,5	3830,0	676,7	667,0	149,95	0,428440	1,242328
	AODV_LARO	8301,2	4109,8	2549,0	6658,8	409,7	733,2	499,5	148,51	0,425166	1,226298
8	AODV	13753,5	8217,3	0,0	8217,3	3994,7	984,8	563,8	138,32	0,394226	0,863776
	AODV_LA	9502,5	7580,0	0,0	7580,0	536,8	861,7	832,3	143,10	0,407350	0,893029
	AODV_RO	12283,7	3566,2	2946,7	6512,8	3994,3	789,7	986,8	143,95	0,410937	0,974980
	AODV_LARO	8468,7	3302,2	3205,8	6508,0	497,5	846,5	616,7	143,36	0,409162	1,040284
10	AODV	14406,8	8863,3	0,0	8863,3	3971,3	972,7	599,5	135,24	0,385880	0,951475
	AODV_LA	10071,5	8001,8	0,0	8001,8	581,7	917,0	571,0	134,34	0,382775	1,051442
	AODV_RO	12744,5	3789,2	3229,3	7018,5	3971,8	787,7	966,5	147,52	0,420766	0,963162
	AODV_LARO	8604,7	3414,2	3235,8	6650,0	546,5	794,2	614,0	144,97	0,413903	0,992282
12	AODV	14294,2	9001,8	0,0	9001,8	3985,5	777,3	529,5	132,49	0,379268	0,940120
	AODV_LA	9422,0	7634,3	0,0	7634,3	577,2	734,2	476,3	136,51	0,389806	0,791595
	AODV_RO	12475,5	3511,8	3242,7	6754,5	3983,3	724,3	1013,3	142,49	0,406438	0,895497
	AODV_LARO	9124,7	3385,3	3809,2	7194,5	591,7	787,2	678,3	146,49	0,419115	0,991621
14	AODV	15161,3	9558,5	0,0	9558,5	3978,8	975,5	648,5	129,44	0,370391	0,842298
	AODV_LA	11531,7	9194,8	0,0	9194,8	775,2	931,8	629,3	138,67	0,395752	0,755297
	AODV_RO	13668,8	3942,3	3627,7	7570,0	3988,3	852,7	1257,8	137,34	0,392275	0,811485
	AODV_LARO	10044,0	3964,3	3742,3	7706,7	726,0	878,3	733,0	141,21	0,402002	0,813142
16	AODV	15381,5	9691,7	0,0	9691,7	3983,8	961,3	744,7	118,90	0,339531	1,031496
	AODV_LA	11804,5	9345,2	0,0	9345,2	808,5	972,5	678,3	129,14	0,368987	0,816635
	AODV_RO	13687,0	3701,3	3773,8	7475,2	3980,8	882,8	1348,2	122,16	0,349559	1,000287
	AODV_LARO	10239,2	3634,5	4032,0	7666,5	797,0	896,5	880,2	129,53	0,369026	0,985253

Changing the transmission rates from 4 to 14 packets per second

Increasing the Transmission Rates with steps of 2 from 4 to 14.

The grid is: 1500 by 1000 , number of nodes is: 60 with 48 connections
Velocity is: 10 m/s, Seed is: 1 and simulation time is: 40 sec

		Routing Overheads							Performance		
Rate	Protocol	Total	Requests	Dest Search	Total Rqsts	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
4	AODV	13940	8751	0	8751	3988	860	341	163,20	0,542388	0,625607
	AODV_LA	9193	7452	0	7452	560	776	405	158,62	0,530349	0,705560
	AODV_RO	12817	4196	3092	7288	3988	685	856	170,82	0,477128	0,835281
	AODV_LARO	9753	3901	3902	7802	566	835	550	158,34	0,565444	0,809704
6	AODV	14686	9335	0	9335	3994	904	453	139,81	0,447177	0,898524
	AODV_LA	9401	7699	0	7699	511	743	447	134,27	0,448393	0,805781
	AODV_RO	13141	4354	3121	7475	3996	690	980	146,15	0,459173	0,938873
	AODV_LARO	9540	4174	3517	7692	524	747	577	146,22	0,472415	0,845565
8	AODV	14407	8863	0	8863	3971	973	600	143,68	0,385880	0,951475
	AODV_LA	10072	8002	0	8002	582	917	571	139,63	0,382775	1,051442
	AODV_RO	12745	3789	3229	7019	3972	788	967	149,95	0,420766	0,963162
	AODV_LARO	8605	3414	3236	6650	547	794	614	148,51	0,413903	0,992282
10	AODV	15160	9696	0	9696	3982	905	576	138,32	0,313187	1,251226
	AODV_LA	9402	7574	0	7574	551	751	526	143,10	0,330954	1,210661
	AODV_RO	13896	4712	3345	8058	3988	692	1159	143,95	0,325784	1,141860
	AODV_LARO	9612	4471	3310	7782	508	689	634	143,36	0,332302	1,317161
12	AODV	15209	9683	0	9683	3974	880	672	135,24	0,273105	1,227072
	AODV_LA	9105	7314	0	7314	523	719	532	134,34	0,277274	1,358170
	AODV_RO	13446	4429	3100	7529	3985	665	1267	147,52	0,290325	1,360703
	AODV_LARO	9572	4363	3307	7670	540	695	676	144,97	0,285754	1,411690
14	AODV	15370	9866	0	9866	3966	874	665	132,49	0,239970	1,249535
	AoDV_LA	9136	7448	0	7448	494	728	466	136,51	0,239032	1,366996
	AODV_RO	13927	5022	2980	8002	3974	663	1288	142,49	0,274845	1,410657
	AODV_LARO	9423	4448	3129	7577	510	664	672	146,49	0,278062	1,223481

Changing the CBR-Sources from 20 to 48 souces

Reducing the CBR_Sources with steps of 4 from 48 to 20.

The grid is: 1500 by 1000 , number of nodes is: 60

Velocity is: 10 m/s, transmission Rate is 8 packets/sec, Seed is: 1 and simulation time is: 40 sec

CBR Sources	Protocol	Routing Overheads							Performance		
		Total	Requests	Dest Search	Total Rqsts	Hello	Replies	Errors	Throughput	Delv Ratio	Av Delay
20	AODV	8136	3605	0	3605	4014	330	188	84,55	0,611413	0,787502
	AODV_LA	4324	3432	0	3432	317	331	243	84,83	0,610447	0,791865
	AODV_RO	8120	1919	1421	3340	4015	306	461	87,95	0,635046	0,959294
	AODV_LARO	4841	2020	1802	3823	339	360	321	91,30	0,661235	0,833991
24	AODV	9224	4594	0	4594	4013	396	223	100,27	0,582841	0,757805
	AODV_LA	4928	3941	0	3941	361	363	263	102,42	0,597651	0,762115
	AODV_RO	8568	2166	1529	3694	4016	333	525	106,59	0,622194	0,886156
	AODV_LARO	4996	1963	1962	3925	355	395	328	108,96	0,635607	0,797346
28	AODV	10137	5360	0	5360	4006	489	267	108,17	0,528929	0,792773
	AODV_LA	5700	4588	0	4588	365	430	317	110,38	0,538542	0,666748
	AODV_RO	9716	2764	1869	4633	4010	446	628	111,05	0,540561	0,785319
	AODV_LARO	6202	2488	2364	4852	2364	495	450	110,42	0,539595	0,948696
32	AODV	11252	6277	0	6277	4005	617	357	111,55	0,467117	0,809256
	AODV_LA	6733	5365	0	5365	440	563	366	114,29	0,477751	0,904884
	AODV_RO	10855	3214	2343	5557	4009	529	761	117,56	0,491008	0,897499
	AODV_LARO	6899	3011	2460	5472	424	516	488	122,26	0,512684	0,999614
36	AODV	11644	6614	0	6614	4000	674	357	128,49	0,474192	0,852682
	AODV_LA	7192	5753	0	5753	446	571	438	120,40	0,444125	0,826496
	AODV_RO	11186	3440	2407	5847	4002	536	801	126,30	0,467464	1,001733
	AODV_LARO	7752	3399	2811	6210	437	573	532	119,10	0,440407	0,874690
40	AODV	12591	7478	0	7478	3993	736	384	124,46	0,420466	1,059623
	AODV_LA	8025	6469	0	6469	490	633	433	122,49	0,412652	0,879733
	AODV_RO	12270	3986	2815	6801	3995	627	847	127,84	0,431200	0,977124
	AODV_LARO	8123	3513	2906	6419	474	644	587	127,64	0,429862	0,860212
44	AODV	13633	8380	0	8380	3994	797	462	130,98	0,399094	1,030641
	AODV_LA	9393	7642	0	7642	526	737	488	127,63	0,387791	0,999599
	AODV_RO	12527	4103	2919	7022	3991	635	879	128,57	0,392194	1,210695
	AODV_LARO	9234	4077	3309	7386	510	720	615	130,65	0,396984	1,086479
48	AODV	14407	8863	0	8863	3971	973	600	135,24	0,385880	0,951475
	AODV_LA	10072	8002	0	8002	582	917	571	134,34	0,382775	1,051442
	AODV_RO	12745	3789	3229	7019	3972	788	967	147,52	0,420766	0,963162
	AODV_LARO	8605	3414	3236	6650	547	794	614	144,97	0,413903	0,992282

D. Confidence Limits Calculations for the various implementations for the 1500x100 Grid with 10 iterations of 60 seconds each

Protocol	Throughput	sq dev (thrpt)	Del Ratio	sq dev (del ratio)	Av Dely	sq dev (av del)
AODV0	222,43	9,85	0,388849	0,000028	0,697476	0,057910
AODV1	200,40	356,84	0,348543	0,001227	1,103840	0,027463
AODV2	198,95	413,91	0,349241	0,001179	0,880263	0,003348
AODV3	214,32	24,67	0,377810	0,000033	1,319135	0,145172
AODV4	220,63	1,79	0,386741	0,000010	1,159765	0,049126
AODV5	211,00	68,75	0,367615	0,000255	0,909289	0,000831
AODV6	218,62	0,45	0,382093	0,000002	0,715405	0,049602
AODV7	277,71	3412,44	0,485728	0,010435	0,805955	0,017468
AODV8	219,52	0,05	0,383323	0,000000	0,773087	0,027236
AODV9	209,34	99,09	0,365812	0,000316	1,016993	0,006221
	SUMM OF SQUARE DEVIATION:	**4387,84**		**0,013485**		**0,384377**
	AVERAGE THROUGHPUT:					
	219,29		0,383576		0,938121	
	SAMPLE SIZE	10		10		10
	DEGREE OF FREEDON (n-1):	9		9		9
	STANDARD DEVIATION:	22,08		0,038708		0,206660
	SIGNIFICANT LEVEL	0,05		0,05		0,05
	CONFIDENCE	**13,69**		**0,023991**		**0,128087**

Protocol	Throughput	sq dev (thrpt)	Del Ratio	sq dev (del ratio)	Av Dely	sq dev (av del)
AODV_LA0	245,84	460,24	0,429072	0,001298	0,642652	0,022084
AODV_LA1	236,91	156,68	0,417215	0,000584	0,768132	0,000535
AODV_LA2	211,69	161,22	0,372809	0,000410	0,779662	0,000134
AODV_LA3	199,71	609,16	0,349691	0,001880	0,920228	0,016633
AODV_LA4	215,64	76,54	0,378316	0,000217	0,993456	0,040884
AODV_LA5	214,12	105,53	0,373656	0,000376	0,827552	0,001317
AODV_LA6	220,14	18,03	0,388191	0,000024	0,671384	0,014370
AODV_LA7	264,75	1629,26	0,462346	0,004802	0,783638	0,000058
AODV_LA8	235,38	120,85	0,410114	0,000291	0,751510	0,001580
AODV_LA9	199,71	609,16	0,349099	0,001932	0,774378	0,000285
	SUMM OF SQUARE DEVIATION:	**3946,69**		**0,011813**		**0,097880**
	AVERAGE THROUGHPUT:					
	224,39		0,3930509		0,7912592	
	SAMPLE SIZE	10		10		10
	DEGREE OF FREEDON (n-1):	9		9		9
	STANDARD DEVIATION:	20,94		0,036229		0,104286
	SIGNIFICANT LEVEL	0,05		0,05		0,05

	CONFIDENCE	**12,98**		**0,022455**		**0,064636**

Protocol	Throughput	sq dev (thrpt)	Del Ratio	sq dev (del ratio)	Av Dely	sq dev (av del)
AODV_RO0	235,59	142,78	0,412825	0,000454	0,966676	0,001116
AODV-RO1	238,71	227,00	0,419371	0,000776	0,825396	0,011635
AODV_RO2	222,01	2,65	0,388318	0,000010	0,871482	0,003817
AODV_RO3	194,79	832,40	0,340338	0,002620	1,226745	0,086132
AODV_RO4	212,32	128,27	0,371680	0,000394	1,155092	0,049208
AODV_RO5	193,20	926,87	0,338217	0,002841	0,695388	0,056584
AODV_RO6	227,69	16,42	0,396670	0,000026	1,027501	0,008881
AODV_RO7	258,87	1240,75	0,453960	0,003898	1,008490	0,005659
AODV_RO8	236,91	175,97	0,414758	0,000540	0,794075	0,019373
AODV_RO9	216,33	53,41	0,379089	0,000155	0,761783	0,029405
	SUMM OF SQUARE DEVIATION:	**3746,52**		**0,011714**		**0,271812**
	AVERAGE THROUGHPUT:					
	223,64		0,391523		0,933263	
	SAMPLE SIZE	10		10		10
	DEGREE OF FREEDON (n-1):	9		9		9
	STANDARD DEVIATION:	20,40		0,036077		0,173785
	SIGNIFICANT LEVEL	0,05		0,05		0,05
	CONFIDENCE	**12,65**		**0,022360**		**0,107711**

Protocol	Throughput	sq dev (thrpt)	Del Ratio	sq dev (del ratio)	Av Dely	sq dev (av del)
AODV_LARO0	250,00	691,44	0,438341	0,002104	0,869933	0,000598
AODV_LARO1	231,36	58,70	0,406079	0,000185	0,937016	0,001817
AODV_LARO2	218,27	29,49	0,385365	0,000050	1,093283	0,039557
AODV_LARO3	206,98	279,62	0,363338	0,000849	1,026062	0,017337
AODV_LARO4	227,07	11,33	0,397964	0,000030	1,024419	0,016907
AODV_LARO5	193,13	934,90	0,338749	0,002886	0,768959	0,015734
AODV_LARO6	219,87	14,73	0,384932	0,000057	0,885568	0,000078
AODV_LARO7	260,67	1366,26	0,458465	0,004355	0,822976	0,005100
AODV_LARO8	218,55	26,56	0,382413	0,000101	0,711605	0,033412
AODV_LARO9	211,14	157,90	0,369052	0,000548	0,804113	0,008151
	SUMM OF SQUARE DEVIATION:	**3570,93**		**0,011166**		**0,138690**
	AVERAGE THROUGHPUT:					
	223,70		0,392470		0,894393	
	SAMPLE SIZE	10		10		10
	DEGREE OF FREEDON (n-1):	9		9		9
	STANDARD DEVIATION:	19,92		0,035224		0,124137
	SIGNIFICANT LEVEL	0,05		0,05		0,05
	CONFIDENCE	**12,35**		**0,021831**		**0,076939**